Wireless Multi Hop Access Networks and Protocols

Nilsson Plymoth, Anders

2007

Wireless Multi Hop Access Networks and Protocols

Anders Nilsson Plymoth

LUND INSTITUTE OF TECHNOLOGY
Lund University
Department of Electrical and Information Technology
Lund University

ISSN 1101-3931
ISRN LUTEDX/TETS–1084–SE+194P

Printed in Sweden
Lund 2007

To Amelie

This thesis is submitted to Research Board FIME - Physics, Informatics, Mathematics and Electrical Engineering - at Lund Institute of Technology, Lund University in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Engineering.

Contact information:
Anders Nilsson Plymoth
Department of Electrical and Information Technology
Lund University
P.O. Box 118
SE-221 00 LUND
Sweden

Phone: +46 46 222 03 67
Fax: +46 46 14 58 23
e-mail: andersn@eit.lth.se

ABSTRACT

As more and more applications and services in our society now depend on the Internet, it is important that dynamically deployed wireless multi hop networks are able to gain access to the Internet and other infrastructure networks and services. This thesis proposes and evaluates solutions for providing multi hop Internet Access. It investigates how ad hoc networks can be combined with wireless and mesh networks in order to create wireless multi hop access networks. When several access points to the Internet are available, and the mobile node roams to a new access point, the node has to make a decision when and how to change its point of attachment. The thesis describes how to consider the rapid fluctuations of the wireless medium, how to handle the fact that other nodes on the path to the access point are also mobile which results in frequent link and route breaks, and the impact the change of attachment has on already existing connections.

Medium access and routing protocols have been developed that consider both the long term and the short term variations of a mobile wireless network. The long term variations consider the fact that as nodes are mobile, links will frequently break and new links appear and thus the network topology map is constantly redrawn. The short term variations consider the rapid fluctuations of the wireless channel caused by mobility and multi path propagation deviations. In order to achieve diversity forwarding, protocols are presented which consider the network topology and the state of the wireless channel when decisions about forwarding need to be made. The medium access protocols are able to perform multi dimensional fast link adaptation on a per packet level with forwarding considerations. This i ncludes power, rate, code and channel adaptation. This will enable the type of performance improvements that are of significant importance for the success of multi hop wireless networks.

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor Prof. Ulf Körner, for his support and constant belief in my research. Lars Reneby for introducing me to the department and supporting me in my initial research activities. A few of the papers written as basis for this thesis have been done in cooperation with other friends and colleagues including Charles Perkins, Ryuji Wakikawa, Jari Malinen, Antti Touminen, J.J Garcia-Luna-Aceves, Marco Spohn, Ali Hamidian and Per Johansson. Thank you all for your contributions and support. Finally, Amelie, my family and all my other friends for your support and constant encouragement.

Lyon, France, October, 2007 Anders Nilsson Plymoth

CONTENTS

1. CHAPTER I

Introduction

The enormous increase of mobile computing and the number of communication devices, such as cell phones, laptops and personal digital assistants, is driving a revolutionary change in our information society. We are moving towards a new computing age where a user, at the same time, utilizes several electronic platforms through which he can access all the required information he needs. The nature itself of the ubiquitous users and devices makes wireless networks and technologies the easiest solution for their interconnection needs. As a result, wireless computing has been experiencing exponential growth for the past decade.

The future Internet is likely to be fundamentally different than the Internet today because it will be dominated by the many mobile devices, that all have very diverse computational resources. Today, the number of mobile devices is growing very rapidly, and it is expected that the mobile device population of the Internet will soon contain well over several billion wireless devices.

1.1 Ad Hoc Networks

Mobile ad hoc networks are networks that are formed dynamically by an autonomous system of nodes that are connected via wireless links without using the existing network infrastructure or central administration. The nodes are free too move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Mobile ad hoc networks are infrastructureless networks since they do not require any fixed infrastructure, such as a base station, for their operation. In general, routes between nodes in an ad hoc network may include multiple hops, and hence it is appropriate to call such networks as multi-hop wireless ad hoc networks. Each node will be able to communicate directly with any other node that resides within its transmission range. For communication with nodes that reside beyond this range, the node needs to use intermediate nodes to relay the messages hop by hop.

Ad hoc networking does have some networking challenges. Some of these are the traditional problems of wireless communication and networking:

- The wireless medium has no absolute or observable boundaries outside which stations are known to be unable to correctly receive data.
- The wireless channel is unprotected from outside channels.

- The wireless medium is significantly less reliable than wired media.
- The channel has time-varying and asymmetric propagation properties.
- Hidden terminal and exposed terminal phenomena may occur that degrade performance

1.1.1 Research Issues

Because of these problems, the multi-hop nature and the lack of fixed infrastructure, ad hoc networks have some specific constraints that make research in this field quite challenging:

Autonomous. Ad hoc networks does not depend on any established infrastructure or centralized administration. Each node operates in distributed peer-to-peer mode, acts as an independent router and generates independent data. Network management has to be distributed across different nodes, which brings added difficulty in fault detection and management.

Multi-hop routing. No default router is available, and every node acts as a router and forward packets in order to enable information sharing between mobile hosts.

Dynamically changing network topologies. In mobile ad hoc networks, because nodes can move arbitrarily, the multi-hop network topology frequently and unpredictably changes, resulting in route changes, frequent network partitions, and possibly packet losses.

Variation in link and node capabilities. Each node may be equipped with one or more radio interfaces that have varying transmission and receiving capabilities and operate across different frequency bands. This heterogeneity in radio capabilities may result in asymmetric links. In addition, each mobile node might have different software and hardware configurations, which result in processing capability variations. Designing network protocols and algorithms for this heterogeneous network can be complex, requiring dynamic adaptation to the changing conditions, such as power and channel conditions, traffic load and distribution variations, congestion etc.

Energy. Because batteries typically carried by each mobile node have limited power supply, processing power is limited, which in turn limits services and applications that can be supported by each node. This becomes a bigger issue in mobile ad hoc networks because, as each node is acting as both an end system and a router at the same time, additional energy is required to forward packets from other nodes.

Network scalability. Currently, most network management algorithms were designed to work on fixed or relatively small wireless networks. Many mobile

ad hoc network applications involve large networks with tens of thousands of nodes, as found for example, in sensor networks and tactical networks. Scalability is critical to the successful deployment of these networks. The steps toward a large network consisting of nodes with limited resources are not straightforward, and present many challenges that are still to be solved, such as: addressing, routing, location management, configuration management, interoperability, security etc.

1.2 IEEE 802.11 networks

In 1997, the IEEE adopted the first wireless local area network standard, named IEEE 802.11 [1], with data rates up to 2 Mbps. Since then, several task groups have been created to extend the IEEE 802.11 standard. Task groups 802.11b, 802.11a and 802.11g have completed their work by providing three relevant extensions to the original standard which are often referred to as Wireless Fidelity (Wi-Fi). The 802.11b task group produced a standard for WLAN operations in the 2.4 GHz band, with data rates up to 11 Mbps and backward compatibility. This standard, published in 1999, has become a huge success and is supported by most laptops today and newer pdas. 802.11g is a high-speed extension to 802.11b and supports data rates up to 54 Mbps. Because 802.11g is backward compatible with 802.11b, 802.11g have become a big success, and have now more or less replaced 802.11b as the major 802.11 physical layer standard. Almost all laptops sold today support 802.11g. The 802.11a task group created a standard for WLAN operation in the 5 GHz band, also with data rates up to 54 Mbps. But 802.11a never became a big success, mostly because it wasn't compatible with the original standard, nor 802.11b. Among the other task groups, it is worth mentioning the task group 802.11e that enhances the MAC with QoS features to support voice and video over 802.11 networks.

The IEEE 802.11 standard defines two operational modes for WLANs: *infrastructure-based* and *infrastructure-less* or *ad hoc*. Network interface cards can be set to work in either of these modes but not in both simultaneously. Infrastructure mode resembles cellular infrastructure-based networks. It is the mode commonly used to construct the so-called Wi-Fi hotspots, i.e., to provide wireless access to the Internet. In the ad hoc mode, any stations that are within the transmission range of each other, can after a synchronization phase, start communicating. No AP is required, and the ad hoc network can be created dynamically, on the fly, without any central administration.

1.3 Routing in Ad hoc Networks

In contrast to infrastructure based networks, all nodes are mobile and can be connected dynamically in an arbitrary manner. All nodes therefore behave as routers and take part in the discovery and maintenance of routes to other nodes in the net-

work. Ad hoc networks are very useful in emergency search-and-rescue operations, meetings or conventions in which persons wish to quickly share information, and data acquisition operations in inhospitable terrain.

Routing protocols for ad hoc networks can be divided into two main categories: reactive or proactive, sometimes also called on demand and table driven protocols, depending on how and when the routes are discovered. In proactive routing protocols routes are constantly maintained and updated, assuring that a route is always available when needed. In reactive protocols, routes are discovered and maintained when they are needed, introducing a route discovery latency. When routes are no longer needed, they are removed from the routing table.

Both categories have their advantages and disadvantages. Proactive protocols have the advantage of always having an available route, if one exist, and therefore typically experience a lower delay than reactive protocols do. Proactive protocols also have the advantage of knowing the network topology and the number of nodes in the network. Reactive protocols on the other hand, doesn't have to maintain routes that isn't being used, thereby saving scarce energy resources as many nodes are likely to be battery operated.

This thesis mainly use, analyze and compare two different routing protocols: the Ad hoc On demand Distance Vector (AODV) routing protocol [2] and the Optimized Link State Routing (OLSR) protocol [3].

AODV is a reactive protocol that initiates route discovery whenever a source needs a route, and maintains this route as long as it is needed by the source. Each node also maintains a monotonically increasing sequence number that is incremented whenever there is a change in the local connectivity information for the node. Route Discovery follows a *Route Request* (RREQ), *Route Reply* (RREP) query mechanism. In order to obtain a route to another node, the source node broadcasts a RREQ packet across the network, and then sets a timer to wait for the reception of a reply. Nodes receiving the RREQ can respond if they are either the destination, or if they have an unexpired route to the destination. If these conditions are met, a node responds by unicasting a RREP back to the source node.

OLSR is a proactive protocol that is an optimization of the pure link state algorithm adapted to the requirements of a mobile wireless network. The key concept used in the protocol is that of multipoint relays (MPRs). MPRs are selected nodes (by their one hop neighbors) which forward broadcast messages. The use of MPRs reduces the size of the control packets by declaring only a subset of links towards its neighbors, the MPRs. It also minimizes flooding of the control traffic by only using the selected MPRs to diffuse the control information. All other neighboring nodes receive the information, but do not rebroadcast it.

1.4 Medium Access Control

A medium access control (MAC) protocol moderates access to the shared wireless medium by defining rules that allow devices to communicate with each other in an

orderly and efficient manner. MAC protocols decide what device should be allowed to transmit and access the physical medium, at any give time. In a wireless environment, if two nodes transmit at the same time, they will cause interference for each other that may result in the loss of data. A common solution to this problem is to only allow one single node to transmit on the channel at the same time, thus enabling successful transmissions and preventing collisions from occurring. MAC protocols therefore play a crucial role in wireless networks by ensuring efficient and fair sharing of the scarce wireless bandwidth. .

1.4.1 Wireless MAC Issues

The unique properties of the wireless medium make the design of MAC protocols very different from, and more challenging than, wireline networks. Some of the unique properties of wireless systems and their medium are:

Half-Duplex Operation: In wireless systems it is very difficult to receive data when the transmitter is sending data. This is because when a node is transmitting data, a large fraction of the signal energy leaks into the receive path. This is referred to as self-interference. The transmitted and received power levels can differ by several orders of magnitude. The leakage from the transmitted signal typically has much higher power than the received signal, which makes it impossible to detect a received signal while transmitting data. Hence, collision detection is not possible while sending data. Due to the half-duplex mode of operation, the link needs to be multiplexed in time (TDM), frequency (FDM) or by code (CMD). As collisions cannot be detected by the sender, all proposed protocols attempt to decrease the probability of a collision using different collision avoidance principles.

Time Varying Channel: Radio signals propagate according to three mechanisms: reflection, diffraction, and scattering. The signal received by a node is a superposition of time-shifted and attenuated versions of the transmitted signal. As a result, the received signal power varies as a function of time. This phenomenon is called multipath propagation. The rate of variation of the channel is determined by the coherence time of the channel. Coherence time is defined as time within which the received signal strength changes by 3 dB. When the received signal strength drops below a certain threshold, the node is said to be in fade. Handshaking is a widely used strategy to mitigate time-varying link quality. When two nodes want to communicate with each other, they exchange small messages that test the wireless channel between them. A successful handshake indicates a good communication link between the two nodes.

Carrier Sensing: Carrier sensing is a function of the position of the receiver relative to the transmitter. In the wireless medium, because of attenuation and multipath propagation, signal strength decays more or less according to distance. Only nodes within a specific radius of the transmitter can detect the carrier of the channel. This location-dependent carrier sensing results in three types of nodes in protocols that use carrier sensing. *Hidden Nodes*: A hidden node is one that is within the range of the intended destination but out of range of the sender. Hence,

hidden nodes can cause collisions on data transmission. *Exposed Nodes*: Exposed nodes are complementary to hidden nodes. An exposed node is one that is within the range of the sender but out of range of the destination. If the number of exposed nodes are not minimized, the bandwidth is underutilized. *Capture*: Capture is said to occur when a receiver can cleanly receive a transmission from one of two simultaneous transmissions, both within its range. When two nodes transmit simultaneously, the signal strength received from one node may be much higher than that of the other, and can be decoded without errors despite the presence of the other transmission. Capture can result in unfair sharing of bandwidth with preference given to nodes closer to the transmitter. Wireless MAC protocols need to ensure fairness under such conditions.

1.4.2 Wireless MAC protocols

The most popular wireless MAC layer protocol used today is the IEEE 802.11 DCF [1]. 802.11 as explained above, is being used in almost every laptop computer as a wireless LAN technology. DCF is the most commonly used medium access technonology defined by the 802.11 specification,

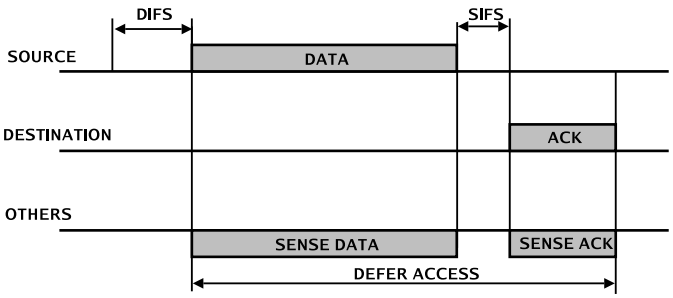


Fig. 1.1: DCF basic access

DCF is based on CSMA/CA, and it provides asynchronous access for best effort service. The basic operation of the DCF is illustrated in Figure 1.1. If a station generates a frame to transmit when there is no ongoing backoff procedure, it checks the medium status to see if it is idle. If the medium is sensed to be idle, the station immediately proceeds with its transmission after an idle interval equal to DCF Inter Frame Space (DIFS); this is often referred to as an immediate access. If the medium is sensed to be busy, the station defers its access until the medium is determined to be idle for a DIFS interval, and then it starts a backoff procedure. A backoff procedure starts by setting its own backoff timer by uniformly choosing a random value from the range [0,CW], where CW is the current contention window size, and its size is an integer value within the range of CW_{min} and CW_{max}. The backoff counter is decreased by a slot time as long as the channel is sensed idle, while it remains frozen when the channel is sensed busy. The backoff countdown is resumed after the channel is sensed to be idle for a DIFS interval. When the back-

off counter reaches zero, the station starts its data frame transmission. If the source successfully receives an acknowledgment (ACK) frame after a Short Inter-Frame Space (SIFS) idle period, the transmission is assumed to be successful. After a successful transmission, the source resets its contention window to the minimum value CW_{min}, and performs another backoff process irrespective of whether it has another frame to transmit or not. This process is often referred to as post backoff, and it prevents a station from performing consecutive immediate accesses. On the other hand, if a frame transmission fails, the current contention window size is doubled with the maximum value CW_{max}. The station attempts to transmit the frame again by selecting a backoff counter value from the increase contention window. After the number of failures reaches a retry limit, which is 4 by default, the station drops the frame.

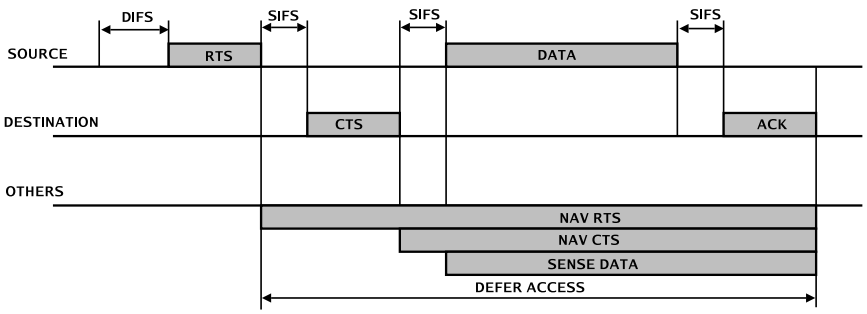


Fig. 1.2: 802.11 RTS CTS Handshake

The RTS/CTS access method is provided in IEEE 802.11 as an option for reducing the collisions caused by hidden terminal problems. When a station needs to transmit a data frame longer than the rtsThreshold, it follows the backoff procedure as in the basic mechanism described before. After that, instead of sending the data frame, it sends a special short control frame called a Request-To-Send (RTS). This frame includes information about the source, destination, and duration required by the following transactions (CTS, DATA, and ACK transmission). Upon receiving the RTS, the destination responds with another control frame called a Clear-To-Send (CTS), which also contains the same information. The transmitting station is allowed to transmit data if the CTS frame is received correctly. All other nodes overhearing either RTS and/or CTS frames adjust their Network Allocation Vector (NAV) to the duration specied in the RTS/CTS frames. The NAV contains the duration for which the channel will be unavailable and is used as virtual carrier sensing. Stations defer transmissions if either physical or virtual sensing indicates that the channel is busy. Nevertheless, if a receiver's NAV is set while the data frame is received, DCF allows the receiver to send the ACK frame.

1.5 Multi-hop Internet Access

Most of the research presented in this thesis relates to typical IEEE 802.11 ad hoc networks. Much of this work focuses on different methods for providing Internet Access to *multi-hop* ad hoc and mesh networks. Multi-hop networking isn't currently directly supported in any of the two modes in 802.11. In order to have networking operating in a multi-hop manner, we need a routing protocol that establishes routes between the communicating nodes in the wireless network. In order to have a multi-hop network, we have to operate in 802.11's ad hoc mode. This is because the infrastructure mode uses a coordination and association function that only extends to nodes within direct communication range.

So, suppose we have a multi-hop ad hoc network, and we want to obtain Internet Access? Well, for this to work, a few different things need to be solved. First of all, the general view of an ad hoc networks is that it is a stand alone network, isolated from any external networks. It should be possible to establish the network with little or no configuration and it should be able to dynamically reconfigure itself. The whole ad hoc networking concept sort of assumes independency and flexibility. This is now, however, slowly starting to change. As the Internet is becoming a more and more integral part of our daily life, a free stand alone ad hoc network seems less useful. If at least one of the nodes in the ad hoc network is within communication range of an Internet access point, why not use that node to access the Internet? In order for a multi hop ad hoc access network solution to work, we have the following constraints and considerations:

- An access point is needed, that acts as a gateway between the ad hoc network and the Internet.
- The access point needs to be able to operate in ad hoc mode, as nodes in the network communicates in ad hoc mode.
- Nodes in the network must be able to discover, identify and differentiate between access points and common nodes. An important property here is how a node can discover an access point. This can either be done proactively, where the access points announces its presence periodically to the network, or reactively where a node may start a discovery process when access to the Internet is needed.
- When more than one access point is available, the node should use a policy to choose the most appropriate access point. This policy can depend on either performance or organizational aspects, or both.
- Nodes must be able to configure an address that is accessible from the Internet. While a node may already have a fully functional address for the ad hoc network, this address is not necessarily accessible, or allowed in the global Internet.

- Correspondent nodes in the Internet communicating with an ad hoc node should regard the ad hoc destination address as any other destination address.

- The network needs to support both macro and micro mobility. Micro mobility is the case where a nodes moves from one access point to another within the same domain. Macro mobility is the case where the node moves to a new access point that is operated and maintained by a different entity, where a new and different addressing policy is enforced that requires the node to configure a new address. This mobility may require both rerouting and readdressing, should be as fast as possible, and be transparent to applications running in the node.

- Nodes must be able to discover and maintain routes to the access point. This maintenance should be synchronized between the routing process and the mobility process. When a mobility decision has been made where a node moves its association to a new access point, routing to and from the access point should be updated accordingly.

- Nodes in the ad hoc network without an Internet configured address should be able to communicate with other nodes in the ad hoc network as normal. The ad hoc network should be configured as a normal ad hoc network, with the extended feature of having Internet access.

- The access gateway should only forward packets destined to the Internet, and not those packets with a destination inside the ad hoc network.

1.6 Mesh networks

Wireless mesh networks is an area that has been receiving a lot of attention within the last few years. They can be considered as wireless networks where each node can function both as an access point, and as a router responsible for forwarding traffic from other parts of the network. Ad hoc networks is therefore a type of wireless networks that is closely related to mesh networks. The main difference between mesh and ad hoc networks is that ad hoc networks are constructed by user terminal nodes, and these user nodes are expected to be highly mobile. This mobility aspect has led to extended amount of research performed on the topic of mobile routing, because as the user nodes move around in the network, the network topology will constantly be changing. Within this topic, IETF has had a very important role, resulting in the standardization through experimental RFCs, of a couple of routing protocols.

While ad hoc networks are still waiting for a killer application into the commercial market, mesh networks devices are already available through different manufactures. Mesh networks are built as cost effective wireless access networks, and have been deployed in some cities as wireless MAN networks.

Mesh networks are also considered a promising technology for emergency, search and rescue operations. This is because mesh networks can rapidly be deployed and be made to need little or no configuration. With well configured mesh devices, the only thing an emergency team needs to do to get a fully operational access networks, is to bring the devices to a location, turn them on, and distribute them in the area. This is especially useful in areas where very little communication infrastructure is currently available, or where the communication infrastructure has been destroyed.

1.7 Diversity forwarding

Diversity forwarding is a concept that tries to utilize the diversity of the network. In multi hop networks, it is often possible to find more than one path between a source and a destination. Sometimes it also possible to find multiple paths of the same lengths, that can be used to create redundant routing paths. Diversity forwarding can be seen as combination of these approaches, where the exact path between a source and destination is not determined beforehand, but that each next hop is determined by each forwarding node. In normal single or multi path ad hoc routing, the path is either determined by the initiating source, or by a routing protocol with a consistent view of the network topology among all the nodes. With diversity forwarding, the forwarding decision is made by each forwarding node based on the current state of the available forwarding links, or the current state of available nodes. For example, if one of the available links that is available for routing between a source and destination is in a bad state due to fading or interference, some other link with more favorable conditions could be chosen. Similarly, if one of the available forwarding neighbors experience high delays because of contention or congestion, some other neighbor could be picked to forward the packet.

This concept has been discussed in a few recent papers. To my knowledge, the first work about diversity forwarding is the Selection Diversity Forwarding (SDF) scheme, presented in [4]. Here a node first multicasts a data packet to a set of candidate nodes, and then the forwarding decision is made based upon responses from the candidates. A similar and sort of reverse idea was later developed in [5] [6], where a small probe, or RTS packet is multicasted to a set of receivers, and the candidate that responds first is chosen as the next hop. Similary, [7] [8] first transmits a probe, but they wait for *all* receivers to respond, before choosing the candidate with the best current radio conditions.

When a diversity protocol queries the set of candidates, it may from the probe messages not only learn which of the candidates that are available, but also the *Channel State Information* (CSI). This information can enable the transmitter to determine which of the available date rates that will be best suited for the current channel radio conditions. In [9] a rate adaptive MAC protocol called Reveiver-Based AutoRate (RBAR) is presented that changes the modulation scheme and thus the data rate based upon the current radio conditions. In an other aspect, [10]

presents a MAC protocol that performs power control that takes into account both the current radio conditions, and the location of neighboring nodes. Link state routing protocols have one significant advantage over distance vector protocols, and that is that link state protocols have an overview of the topology of the network, while a distance vector protocol only see the distance to the final destination through the next hop. This wider view enables a diversity forwarding protocol to make more dynamic routing decisions, as it not only sees the shortest path to a destination, but *all* paths to a destination. This wider view also enables other protocols, such as MAC protocols, or applications to make wiser decisions.

In a diversity forwarding protocol, the task of the routing protocol is to provide the MAC protocol with the set of candidates that it determines should be evaluated.

1.8 Dynamic Code Division Multiple Access

Many wireless systems today use different spread spectrum techniques on the physical layer in order to combat interference and noise. Spread spectrum basically means that the transmitted signal is spread over a frequency band in such a way that it occupies a bandwidth much greater than that which is necessary to send the information. This results in the signal being much less sensitive to interference. The bandwidth is spread by means of a code which is independent of the data that is to be transmitted. The use of an independent code and synchronous reception allows multiple users to access the same frequency band at the same time.

In order to protect the signal, the code used is pseudo-random. It appears to be random, but is actually deterministic, so that the receiver can reconstruct the code for synchronous detection, and since the receiver knows how to generate the same code, it correlates the received signal with the code in order to extract the data.

802.11 as discussed, uses a spread spectrum technique called Direct Sequence Spread Spectrum, DSSS. Here the digital data is directly coded at a much higher frequency than the signal and the bits themselves. The used spreading code is predefined by the 802.11 specification, and since the receiver knows how to generate the same code, it can correlate received signals with that code in order to extract data.

Code Division Multiple Access, CDMA, is a MAC technique that allows multiple users to access the medium at the same time through assignment of unique user codes. In centralized systems such as cellular networks, codes are assigned by the network itself. In ad hoc networks, no central entity is available that can assign codes, and many of the ad hoc and mesh solutions used today is constructed with 802.11 devices.

1.9 Thesis and contribution

This thesis presents architecture solutions and constraints that tries to optimize the performance of hybrid multi hop access systems. Example of issues discussed in

this thesis are how to discover an access point and configure a correct address, how multiple access points should be handled; when and how should a node switch from one access point to the other? What impact does mobility have on the system? How should routing to and from access points, as well as within the network, be achieved and optimized? Should the amount of traffic be considered; will the access point be a hotspot and cause congestion in a specific part of the network? These questions will be answered in the following chapters.

This thesis also presents MAC and routing protocol solutions that enables diversity forwarding by querying the state of available candidates and links prior to the transmission of a data packet. Two link state diversity routing protocols are presented, one on demand and one proactive. The presented MAC protocols all support rate adaptation and power control in addition to providing diversity forwarding support. The final decision on which next hop to forward the packet to, is performed by the routing protocol, based on radio, MAC and network conditions. One MAC protocol also supports the use of node specific CDMA code assignment, with fast link adaptaion and dynamic channel assignment.

1.10 List of papers

This section list the papers on which this thesis is based.

1.10.1 Paper I

Performance Analysis of Traffic Load and Node Density in Ad hoc Networks
Anders Nilsson
Proceedings of The Fifth European Wireless Conference 2004, Barcelona, Spain, February 2004

1.10.2 Paper II

Internet Connectivity for Mobile Ad hoc Networks
Charles E. Perkins, Anders Nilsson, Ryuji Wakikawa, Jari T. Malinen, Antti J. Tuominen
Wireless Communication and Mobile Computing Journal, Volume 2, Issue 5, August 2002

1.10.3 Paper III

A Simulation Study of Internet Access in IPv6 Ad hoc Networks
Anders Nilsson, Charles. E. Perkins, Antti. J. Tuominen, Ryuji. Wakikawa, Jari. T. Malinen
Presented at the AODV Next Generation (AODVng) 2002 Workshop, Lausanne, Switzerland. A shorter version is published in ACM SIGMOBILE Mobile Computing and Communications Review, Volume 6, Issue 3, July 2002

1.10.4 Paper IV

Routing in Hybrid Ad hoc Networks using Service Points
Anders Nilsson, J.J Garcia-Luna-Aceves, Marco. A. Spohn
Proceedings of IEEE 58th Vehicular Technology Conference, VTC 2003-Fall. Orlando, FL, USA, October 2003

1.10.5 Paper V

Micro Mobility Performance in Internet Access Ad Hoc Networks
Anders Nilsson, Ulf Körner
Proceedings of World Wireless Congress 2004, WWC04, San Francisco, CA, May 2004

1.10.6 Paper VI

Micro Mobility and Internet Access Performance for TCP connections in Ad hoc Networks
Anders Nilsson, Ali Hamidian, Ulf Körner
Proceedings of Nordic Teletraffic Seminar 17, Oslo, Norway, August 2004

1.10.7 Paper VII

Performance of Routing and Wireless Aware Transport Layer Connections in Micro Mobility Ad Hoc Networks
Anders Nilsson, Ulf Körner
Proceedings of World Wireless Congress 2005, WWC05, San Francisco, CA, May 2005

1.10.8 Paper VIII

Cross layer routing and medium access control with channel dependant forwarding in wireless ad hoc networks
Anders Nilsson, Per Johansson Ulf Körner
Lecture Notes in Computer Science , Vol. 4396. Springer Verlag. Computer Communication Networks and Telecommunication. 2007, IX, 271 p., Softcover ISBN: 978-3-540-70968-8.

1.10.9 Paper IX

Urban Mesh ad hoc networks and diversity forwarding
Anders Nilsson
Proceedings of 7th Scandinavian Workshop on Wireless Ad-hoc Networks (ADHOC ´07) Stockholm, Sweden, May 2007.

1.10.10 Paper X

Urban Mesh and Ad hoc Mesh Networks
Anders Nilsson Plymoth, Per Johansson Ulf Körner

To appear in ACM International Journal of Network Management, Volume 18 Issue 1, January 2008

BIBLIOGRAPHY

[1] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification, IEEE Std 802.11-1997*. The Institute of Electrical and Electronics Engineers, New York, 1997.

[2] C. Perkins. Ad-hoc on-demand distance vector routing. In *Second IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[3] P. Jacquet, P. Muhlethaler, T Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *IEEE International Multi Topic Conference*, 2001.

[4] P. Larsson. Selection diversity forwarding in a multihop packet radio network with fading channel and capture. *ACM SIGMOBILE Mobile Comnputing and Communication Review*, 5(4):47–54, 2001.

[5] Shweta Jain and Samir R. Das. Exploiting path diversity in the link layer in wireless ad hoc networks. In *Proceedings of the 6th IEEE WoWMoM Symposium, Taormina, Italy*, June 2005.

[6] J. Wang, H. Zhai, Y. Fang, and M. C. Yuang. Opportunistic media access control and rate adaptation for wireless ad hoc networks. In *Proceedings of the IEEE Communications Conference (ICC'04), Paris, France*, June 2004.

[7] P. Larsson and N. Johansson. Multiuser diversity forwarding in multihop packet radio networks. In *Proceedings of the 2005 IEEE Wireless Communications and Networking Conference*, volume 4, pages 2188– 2194, March 2005.

[8] M. Souryal and N. Moayeri. Channel-adaptive relaying in mobile ad hoc networks with fading. In *Proceedings of the IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, California*, pages 142–152, September 2005.

[9] G. Holland, N. Vaidya, and P. Bahl. A rate adaptive mac protocol for multi hop wireless networks. In *Proceedings of the Seventh Annual ACM/IEEE Internation Conference on Mobile Networking (MobiCom'01), Rome, Italy*, October 2001.

[10] A. Muqattash and M. Krunz2006. A single-channel solution for transmission power control in wireless ad hoc networks. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, Tokyo, Japan*, May 2004.

2. CHAPTER II

Performance Analysis of Traffic Load and Node Density in Ad hoc Networks

2.1 Introduction

Ad hoc networks are multihop wireless networks consisting of mobile hosts communicating with each other through wireless links. These networks are typically characterized by scarce resources (e.g. bandwidth, battery power etc), lack of any established backbone infrastructure and dynamic topology. A challenging but critical task that researchers have tried to address over the past few years have been development of routing protocols that suit the characteristics of ad hoc networks.

Several such routing protocols for ad hoc networks have been developed and evaluated [1], [2], [3]. These evaluations mainly focus their performance evaluations upon determining the throughput, packet delivery ratio and overhead of the different protocols. However, since many of the devices used in ad hoc networks are battery operated, they also need to be energy conserving so that battery life is maximized. Thus, when new routing protocols are being developed, these considerations should be taken into account.

In the 70s Kleinrock et al. theoretically studied the performance of Packet Radio Networks and tried to determine the optimum transmission radius. Their results were summarized in their paper "Optimum Transmission Radii for Packet Radio Networks" which was published in 1978 [4]. The paper provides an analytical analysis that explore the tradeoff between increased transmission radius, which result in fewer hops between source and destination, and the effective bandwidth lost at each node as a result of the increase in transmission range. The paper shows that the optimum number of neighbors for a given node is 6, and concludes that a node's transmission radius should be adjusted so that it has six neighbors.

In [5] Royer et al. explore the nature of the transmission power tradeoff in mobile ad hoc networks to determine the optimum node density for delivering the maximum number of data packets. They conclude that there does not exist a global optimum density, but rather that, to achieve this maximum, the node density should increase as the mobility rate of nodes increases. Their simulations were aimed at determining the maximum throughput of the network and therefore the traffic load upon the network was adjusted so that saturation occurred.

This paper examines how the traffic load upon the network and the transmission power affect the overall performance of the network. While increasing the

transmission radius, i.e. the node density, does reduce the available bandwidth, it may also be important to study how the optimum node density varies with different traffic loads and mobility rates. To investigate this, the reactive Ad Hoc On-Demand Distance Vector (AODV) routing protocol [6] is used for routing packets in the network. It is likely that different routing protocols will have different route characteristics, but the results obtained here can be generalized to most on-demand protocols. To make a comparison against more proactive routing protocols, the simulated scenarios were also run with the Optimized Link State Routing (OLSR) protocol [7]. The remainder of this paper is organized as follows. Section 2.3 briefly describes the basic mechanism of AODV's unicast routing. Section 2.4 describes the OLSR routing protocol. Section 2.5 describes the simulation model and environment. Section 4.2 discusses related work and Section 5.7 concludes the paper.

2.2 Related Work

Royer et al. performed a related study in [5]. In this work they varied the transmission power in order to determine the optimum node density for delivering the maximum number of data packets. Their simulations were aimed at determining the maximum throughput of the network and therefore the traffic load upon the network were adjusted so that saturation occurs. They concluded that there does not exist a global optimum density, but rather that, to achieve this maximum, the node density should increase as the mobility rate of nodes increases.

An investigation to determine the critical transmission range were performed in [8]. In this work the authors investigate the minimum transmission range of the transceivers that is required to achieve full network connectivity. They present an algorithm to calculate this minimum transmission range, and then study the effect of mobility on that value.

In [9], the authors study the problem of adjusting the transmission power in order to find a balance between the achieved throughput and power consumption. Algorithms are presented which adaptively adjust the transmission power of the nodes in response to topological changes, with the goals of maintaining a connected network while using minimum power. Through simulation, they show that an increase in throughput, together with a decrease in power consumption can be achieved by managing the transmission levels of the individual nodes.

2.3 Ad hoc On-Demand Distance Vector Routing

The Ad hoc On-Demand Distance Vector (AODV) routing protocol is a reactive protocol designed for use in ad hoc mobile networks. AODV initiates route discovery whenever a source needs a route, and maintains this route as long as it is needed by the source. Each node also maintains a monotonically increasing sequence number that is incremented whenever there is a change in the local con-

nectivity information for the node. These sequence numbers ensure that the routes are loop-free.

2.3.1 Route Discovery

Route Discovery follows a *Route Request* (RREQ), *Route Reply* (RREP) query mechanism. In order to obtain a route to another node, the source node broadcasts a RREQ packet across the network, and then sets a timer to wait for the reception of a reply. The RREQ packet contains the IP address of the destination node, the sequence number of the source node as well as the last known sequence number of the destination. Nodes receiving the RREQ can respond if they are either the destination, or if they have an unexpired route to the destination whose corresponding sequence number is at least as great as that contained in the RREQ. If these conditions are met, a node responds by unicasting a RREP back to the source node. If not, the node rebroadcasts the RREQ. In order to create a reverse route from the destination back to the source node, each node forwarding a RREQ also create a *reverse route entry* for the source route in its routing table.

As intermediate nodes forwards the RREP towards the source node, they create a *forward route entry* for the destination in their routing tables, before transmitting the RREP to the next hop. Once the source node receives a RREP, it can begin using the route to send data packets.

If the source node does not receive a RREP before the timer expires, it rebroadcasts the RREQ with a higher time to live (TTL) value. It attempts this discovery up to some maximum number of attempts, after which the session is aborted.

2.3.2 Route Maintenance

Nodes monitor the link status to the next hops along active routes. When a link break is detected along an active route, the node issues a *Route Error* (RERR) packet. An active route is a route that has recently been used to send data packets. The RERR message contains a list of each destination which has become unreachable due to the link break. It also contains the last known sequence number for each listed destination, incremented by one.

When a neighboring node receives the message, it expires any routes to the listed destinations that use the source as of the RERR message as the next hop. Then, if the node has a record of one or more nodes that route through it to reach the destination, it rebroadcasts the message.

2.4 Optimized Link State Routing

The Optimized Link State Routing (OLSR) protocol is an optimization of the pure link state algorithm adapted to the requirements of a mobile wireless network. The key concept used in the protocol is that of multipoint relays (MPRs). MPRs are selected nodes (by their one hop neighbors) which forward broadcast messages

during the flooding process. This technique lets OLSR substantially optimize the standard Link State algorithm in two ways:

Firstly it reduces the size of the control packets by declaring only a subset of links towards its neighbors, the MPRs. Secondly it minimizes flooding of the control traffic by only using the selected nodes to diffuse the control information. All other neighboring nodes receive the information, but do not rebroadcast it.

All nodes select its set of MPRs such that the set covers all of the nodes that are within two hops away. The OLSR protocol relies on this selection when calculating the routes to all the other known nodes. To achieve this, each node periodically broadcasts information about their one hop neighbors that have chosen it as a multipoint relay node. Each receiving node then uses this information to calculate a route to all other nodes in the network. These routes will be a sequence of hops consisting of MPR nodes between the source and destination node.

2.5 Simulation Model

The simulation platform used for evaluating the proposed approach is GloMoSim [10], a discrete-event, detailed simulator for wireless network systems. It is based on the C-based parallel simulation language PARSEC [11].

In our experiments, the MAC layer is implemented using the default characteristics of the distributed coordination function (DCF) of IEEE 802.11 [12]. This standard uses Request-To-Send (RTS) and Clear-To-Send (CTS) control packets to provide virtual carrier sensing for *unicast* data transmissions between neighboring nodes. A node wishing to unicast a data packet to its neighbor broadcasts a short RTS control packet. When its neighbor receives the packet, it responds with a CTS packet. Once the source receives the CTS, it transmits the data packet. After receiving this data packet, the destination sends an acknowledgement (ACK) to the source, signifying reception of the data packet. The use of the RTS-CTS control packets reduces the potential for the well known hidden-terminal problem. *Broadcast* data packets and RTS control packets are sent using CSMA/CA [12].

Two-Ray Path Loss with threshold cutoff is used as the propagation model. This model uses the Free Space Path Model for near sight and Plane Earth Path Loss for far sight. For a distance *r*, the Free Space model attenuates the signal as $1/r^2$ and the Plane Earth model as $1/r^4$. If the received power level of a packet is below the noise level plus the specified Signal to Noise Ratio (SNR) threshold, a collision is detected.

The data rate for the simulations is 2 Mbits/sec.

2.5.1 The Mobility Model

The mobility model used for the simulations is the Modified Random Direction model [5]. Each node randomly selects a direction in which to travel, where a direction is measured in degrees. The node then randomly selects a speed and destination along the direction and travels there. Once it reaches the destination,

it remains stationary for some pre-defined pause time. At the end of the pause time, a new direction and speed is selected, and movement is resumed. If a node reaches a border of the simulation area, it is bounced back. This model avoids the inherent problems of the popular *random waypoint model* [5, 13] and results in a uniform node distribution as well as causing continuous changes in the topology of the network. The pause time in the simulations is set to 10 seconds and the speed varies between 0 and 10 m/sec.

2.5.2 Simulation Setup

Four different node mobility's between 0 m/s and 10 m/s are modeled. The average number of neighbors in each simulation is varied by adjusting the transmission range. This is typically done by increasing the transmission power of each individual node.

The total amount of traffic injected into the network is varied between 82kbps and 1Mbps. This is done by varying the number of sources in the network and the number of 512-byte data packets sent per second. The type of traffic injected into the network is 10 short-lived CBR sources spread randomly over the network. When one session ends, a new source-destination pair is randomly selected. Thus the input traffic load is constantly maintained.

Each mobility/transmission range/traffic load combination is run for 6 different initial network configurations, and the results are averaged to produce the data points. All in all the total number of simulations run to produce the data points in this study are around 3200. Each simulation simulates 300 seconds and models a network of 100 nodes in a 1000 X 1000 m area.

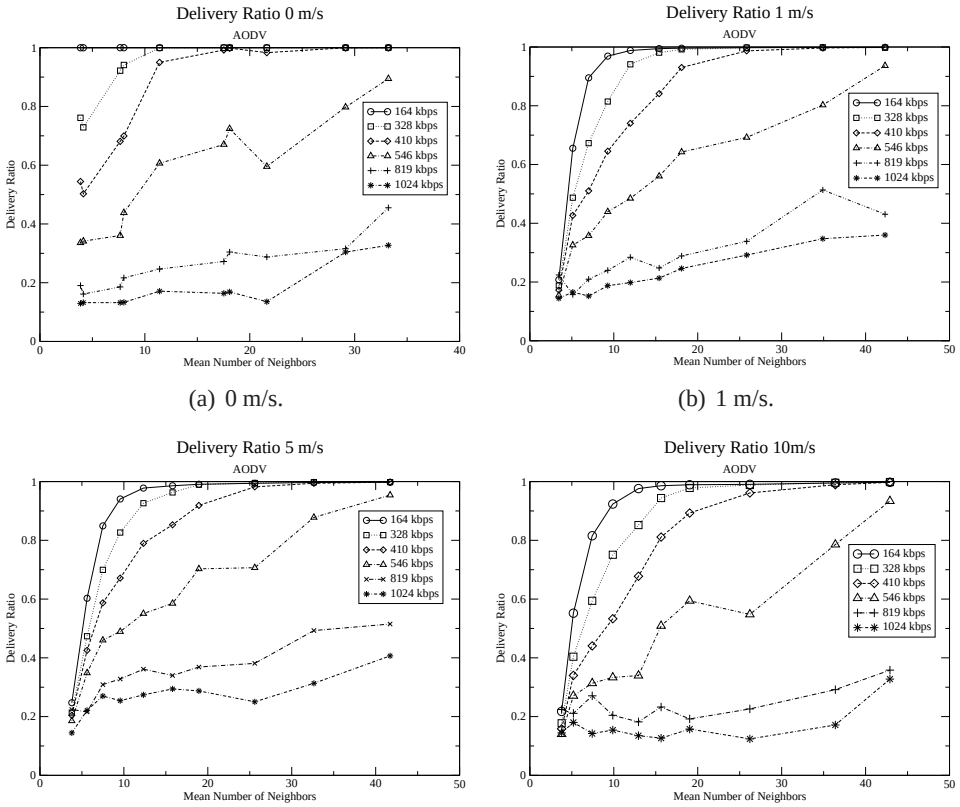
2.6 Results

2.6.1 Delivery Ratio

The delivery ratio is defined as the ratio between the number of packets delivered to a destination to those generated by the sources. This metric illustrates the effectiveness of best effort routing protocols, such as AODV and OLSR, for delivering packets to their intended destination.

The delivery ratio when AODV is used as the routing protocol is shown in Figure 2.1. Four different mobility rates and their graphs are illustrated in the subfigures. The figure shows that for small node densities and lower connectivity, fewer data packets are delivered due to lack of a route. However, when nodes are mobile and the connectivity increases, the delivery ratio rapidly increases for small traffic loads, until the curves level off. For small traffic loads it is therefore possible to find an optimum number of neighbors where almost all packets are delivered. This optimum value does however, depend on both the traffic load and the mobility rate. As mobility increases the optimum value shifts to the right. The faster nodes move, the more frequently link breaks occur. Hence, even though the effective

bandwidth seen at individual nodes suffer due to increased transmission power and collisions, the delivery ratio still increases compared to sparser densities. This is because link breaks are less frequent and routes are maintained for longer periods of time.



(a) 0 m/s. (b) 1 m/s.

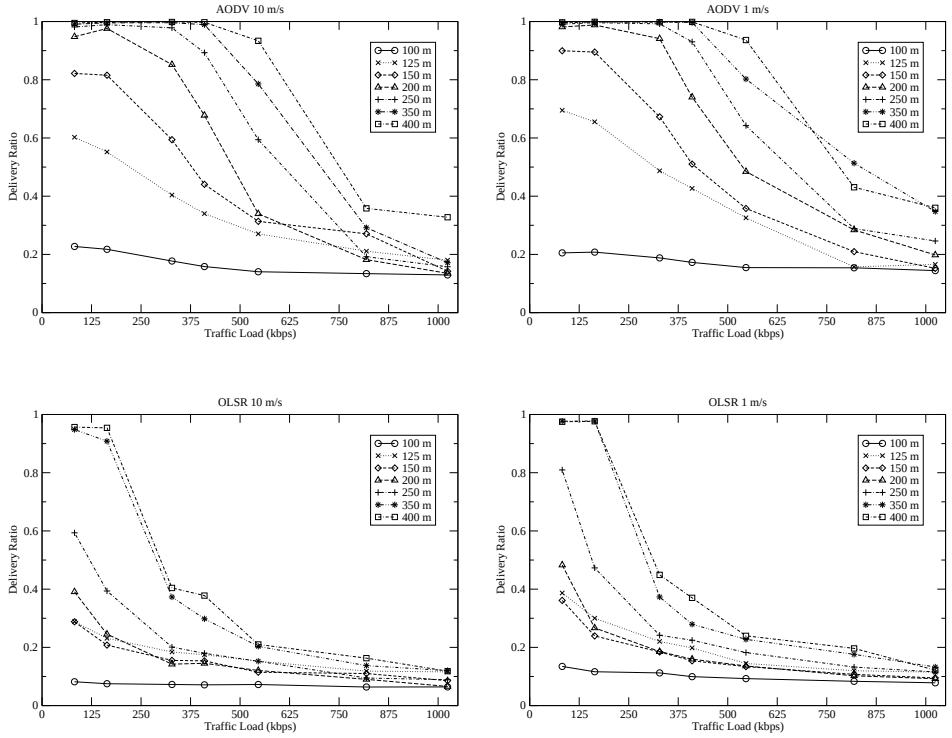
(c) 5 m/s. (d) 10 m/s.

Fig. 2.1: Delivery Ratio vs Mean Number of Neighbors for AODV

As the amount of traffic increases, the rate of increase becomes slower until it is almost linear. This occurs as a result due to the increased number of collisions, as well as reduced channel access. For these higher traffic loads it is therefore more difficult to find an optimum node density.

It should also be noted that when the transmission range is increased, thus increasing the node density, the mean number of hops between a source and destination decreases. This also have a positive effect on the delivery ratio.

Figure 2.2(a) illustrates the relationship between the traffic load and the delivery rate for different transmission ranges. Two mobility rates, 1 m/s and 10 m/s have been used in this setup. As the transmission range of a node is increased, the mean number of neighbors is also increased. It should be noted that the transmission ranges denoted here is the ideal transmission range when we have no interference. As the number of neighbors increase so does the interference, resulting in



(c) 10 m/s for OLSR. (d) 1 m/s for OLSR.

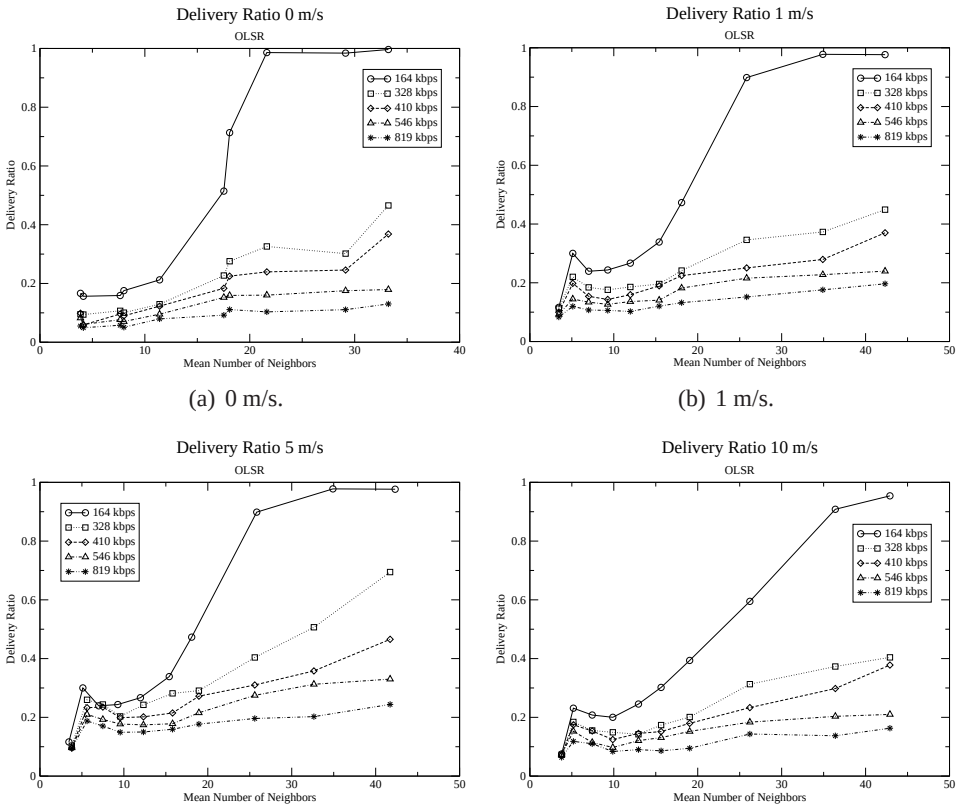
Fig. 2.2: Delivery Ratio per Traffic Load and Transmission Range

more collisions and retransmissions at the MAC layer. The effective transmission range is therefore lowered. These effects are studied in section 2.6.2.

In figure 2.2(a) and figure 2.2(b), AODV is used as the routing protocol. The figures show that as the traffic in the network is increased, the delivery rate becomes lower. For the higher transmission ranges it is possible to sustain a very high delivery rate up to a certain point where the delivery starts to decline. For higher transmission ranges it therefore seems possible to find an optimum traffic load with respect to the delivery ratio. However, for very sparse networks the delivery ratio seems to be fairly independent upon the amount of traffic in the network. This is due to both the lower connectivity as well as the higher probability for channel access. Because of the lower connectivity, it is also harder to establish a route and the delivery ratio is therefore quite low.

Figure 2.3 shows the delivery ratio when OLSR is used as the routing protocol. The figure illustrate that OLSR can achieve very high delivery rates for small traffic loads and dense networks. There are two reasons as to why OLSR performs better for dense networks.

Firstly, the network connectivity is higher for denser networks and the probability for an available route is therefore also higher.



(a) 0 m/s. (b) 1 m/s.

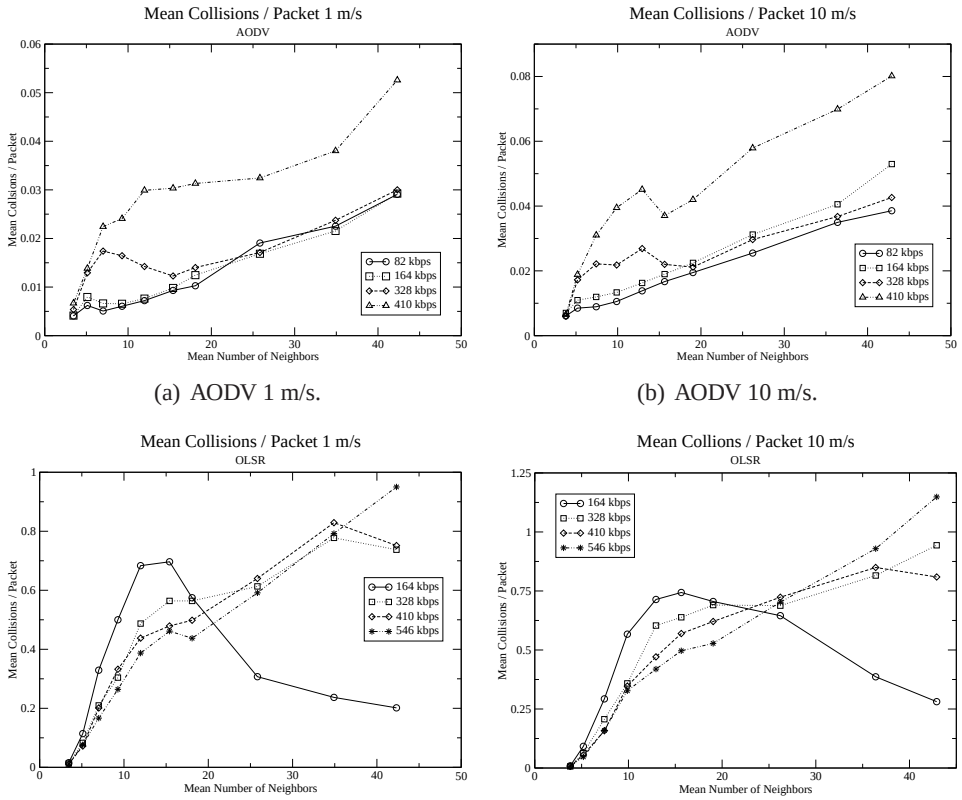
(c) 5 m/s. (d) 10 m/s.

Fig. 2.3: Delivery ratio vs Mean Number of Neighbors for OLSR

Secondly, as the network becomes denser, fewer MPRs are selected. As only MPR nodes will relay link state update messages, the control overhead will drop quickly.

For higher data rates the delivery ratio for OLSR is only slowly increasing. Although fewer MPRs are being selected, the contention for channel access also becomes greater.

Figure 2.2(c) and figure 2.2(d) illustrates the relationship between the traffic load and the delivery rate when OLSR is used as the routing protocol. We can see the same indications as we could when AODV were used. For higher transmission ranges it is possible to sustain a higher delivery ratio up to a certain point, after which the ratio rapidly drops. The difference between AODV and OLSR seems to be that the drop comes a bit earlier for OLSR than it does for AODV. The decline in delivery ratio is also faster for OLSR than for AODV.



(a) AODV 1 m/s. (b) AODV 10 m/s.

(c) OLSR 1 m/s. (d) OLSR 10 m/s.

Fig. 2.4: Mean Number of collisions per delivered packet

2.6.2 Collisions

Figure 2.1 and figure 2.3 seems to indicate that denser networks have better delivery ratio. If this is correct, the optimum network design choice would be to make the network as dense as possible. However, as we can see in figure 2.4, the number of collisions also increases with increasing network density. Figure 2.4 shows the mean number of collisions at the radio layer per delivered packet. This ratio is an indication of the energy cost needed in order to deliver a packet. More collisions at the radio layer typically means that energy has been wasted because the signal could not be received.

Here we see that although denser networks have higher delivery ratios, the price for actually delivering the packets becomes higher. Because more collisions means that additional control information at the MAC layer might need to be sent, more energy have to be spent for delivering the packets.

Because the mobile nodes in an ad hoc network are typically battery operated, although performance can be improved with density, it is not optimum from a energy point of view.

There are also some interesting variations in the displayed graphs. In fig-

ure 2.4(c) and figure 2.4(d) OLSR have been used as the routing protocol. For small traffic loads the number of collisions increases up to a certain point where it levels off and then starts decreasing. The reason for this is the same as explained earlier. As the network becomes denser, fewer MPRs will be selected and the control overhead will therefore be lower. As a result of this, fewer collisions occur. But as the traffic load is increased, the contention for channel access will increase, again causing more collisions to occur. These results are a bit surprising because OLSR was designed to work better in denser networks. The reason for this lies in the large number of nodes in the simulated network, 100 nodes. This means that the size of the link state update messages will be large, as well as many due to topology changes. The result of this is that many broadcasted RTS and update messages will collide. Similar observations were made in [14] after the publication of our study, where a 100 node network was also simulated. They conclude that channel contention and routing overhead cause the MPRs to be saturated. This problem has been further recognized by the work in [15], which is partially conducted by one of the creators of OLSR. They propose a new mechanism to detect link disconnections, in combination with link buffering and packet restoration. Here link breaks are also detected if no CTS or no ACK is received. After a link break, all routes using the broken link is invalidated, and the neighbor and routing tables are updated. If a packet is received using an invalidated route, it is stored in the link buffer until the route is restored through topology updates. This is similar to the route repair procedure of AODV, and it would be an interesting future study to see how this version of OLSR performs for the scenarios of this study.

It is interesting to see that AODV also displays variations, but for higher traffic loads. See figure 2.4(a) and figure 2.4(b). The mean number of packet collisions here rapidly increases with node density up to a point where it levels off or starts decreasing. For even higher node densities the number of collisions again starts to increase. The explanation for this can be found in the way AODV flood request messages. When a node needs a route it broadcasts a RREQ to its immediate neighbors. If the receiving neighbor is unaware of the requested destination address, it rebroadcast the RREQ. However, if the neighbor does know of a route to the destination, it unicasts a RREP back to the requesting node. As the network becomes denser, the probability for a neighbor to have an available route increases. This is the point where the curves level off or starts decreasing. But more neighbors also means that more packets have to be rebroadcasted, increasing the number of collisions. At some point the positive effect of neighbors having available routes will be drowned by rebroadcasts by other neighboring nodes. The number of collisions will then again start to increase. For lower traffic loads these effects are less distinct.

It should also be noted that the scale of the figures are different. The number of collisions that occur when OLSR is used for routing is higher than for AODV.

2.7 Conclusion

With the increasing popularity of mobile networking, it is important to understand the characteristics of these networks so that they can be tuned to achieve optimum performance. A key component for determining the network connectivity is the transmission power. For wireless transmission, a tradeoff exists between increasing the number of neighbors and decreasing the effective bandwidth available to individual network nodes.

It has been shown that it is desirable to increase the node density and transmission power in order to achieve high delivery of data packets to their destinations. Moreover, the optimum connectivity level of the network does not only depend upon the mobility of the nodes, but also upon the traffic load on the network. In sparser networks it is possible to achieve high delivery rates up to a certain point where it starts to decline. When the transmission power of the individual nodes is increased, the delivery rate will also increase in a rate that is dependent upon the traffic load in the network. For lower traffic loads the increase in delivery is quite fast. As the traffic gets higher, the rate of this increase becomes slower. Although denser networks can generally achieve a higher delivery ratio, the cost will also be higher as more collisions occur which consume more power and channel bandwidth.

The conclusion we can draw from this study is that when the behavior, capacity and performance of a wireless ad hoc network is to be determined, the amount of traffic expected in the network, as well as the node density needs to be taken into account.

BIBLIOGRAPHY

[1] T. Clausen, P. Jacquet, and L. Viennot. Comparative study of routing protocols for mobile ad hoc networks. In *Med-hoc-Net*, september 2002. Sardaigne.

[2] J. Broch, D.A. Maltz, David B. Johnson, Y. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc networking protocols. In *Proceedings of the Fourth Annual ACM/IEEE Internation Conference on Mobile Networking (MobiCom'98, pp 25-30*, October 1998.

[3] S.R. Das, C.E. Perkins, and E.M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedings of the IEEE Infocom, pp. 3-12*, March 2000.

[4] L. Kleinrock and J. Silvester. Optimum transmission radii for packet radio networks or why six is a magic number. In *Proceedings of the IEEE National Telecommunications Conference, Birmingham, Alabama*, pages 4.3.1–4.3.5, December 1978.

[5] E. Royer, P. Melliar-Smith, and L. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *Proceedings of the IEEE International Conference on Communications, Helsinki, Finland*, 2001.

[6] C. Perkins. Ad-hoc on-demand distance vector routing. In *Second IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[7] P. Jacquet, P. Muhlethaler, T Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *IEEE International Multi Topic Conference*, 2001.

[8] M. Sanchez, P. Manzoni, and Z.J. Haas. Determination of critical transmission range in ad hoc networks. In *Proceedings of the Multiaccess, Mobility and Teletraffic for Wireless Communications (MMT) Conference*, October 1999. Venice, Italy.

[9] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 403–413, March 2000. Tel Aviv, Israel.

[10] Lokesh Bajaj, Mineo Takai, Rajat Ahuja, Rajive Bagrodia, and Mario Gerla. Glomosim: A scalable network simulation environment. Technical Report 990027, 12, 1999.

[11] R. Bagrodia and R. Meyer. Parsec: A parallel simulation environment for complex system. Technical report, 1998.

[12] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification, IEEE Std 802.11-1997*. The Institute of Electrical and Electronics Engineers, New York, 1997.

[13] C. Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 80–91, June 2002. Lausanne, Switzerland.

[14] J. Lipman M. Abolhasan, T. Wysocki. Performance investigation on three-classes of manet routing protocols. In *Proceedings of Asia-Pacific Conference on Communications 2005*, October 2005.

[15] Goto, S. Yoshida, K. Mase, and T. Clausen. A study of link buffering for olsr. In *Proceedings of OLSR Interop & Workshop, San Diego, CA, USA*, August 2004.

3. CHAPTER III

Internet Connectivity for Mobile Ad hoc Networks

3.1 Introduction

The future Internet is likely to be fundamentally different than the Internet today because it will be dominated by mobile devices with diverse computational resources. Today, the number of mobile devices is growing very rapidly, and in the future the mobile population of the Internet is expected to contain well over several billion wireless devices. Current research in ad hoc networks is likely to further enhance the options for connectivity available to mobile wireless Internet devices.

With the continued growth of interest in ad hoc networks, it is inevitable that some of them will at least occasionally encounter nearby potential points of attachment to different type of networks, including the global Internet. With today's wireless hot spot technologies, for instance, IEEE802.11 [1], Bluetooth [2] will be enhanced with newer technologies such MIMO and mesh network (e.g. 802.11s), and will be become very familiar in our everyday life and enable Internet access from many locations within urban areas. Most such hot spots support IP addressable devices and should be enhanced to enable the construction of a wireless ad hoc network, perhaps in support of 3rd Generation Mobile Telecommunications (3G) services, future 4G services, and the Intelligent Transport System (ITS). 3G is a global development of communication standards and technologies, and ITS comprises an advanced information and telecommunications network for users, roads and vehicles. Since both services are or will be closely related to our daily life, they highlight the importance and necessity of global connectivity for mobile ad hoc networks. Because ad hoc networks do not have to rely on preestablished infrastructure, they can be deployed anywhere, for example, conference premises, in emergency areas and near network-reachable hot spots, wherever participants need to form a (often temporary) data communication network.

The point at which the attachment is to be made (i.e. the IP node with access to the global Internet) is called the *Internet Gateway*. The Internet Gateway (in this chapter, often shortened to just *gateway*, since no other kind of gateway will be important for our purposes) can offer global addressability and bidirectional Internet connectivity to every node in the ad hoc network that has a suitable path to the gateway. We would like to avoid placing any restriction on the mobility of the nodes in the ad hoc network and certainly avoid any restrictions engendered by addressability. This can be done in such a way that mobile wireless nodes can

migrate between wireless access points that have direct access to the wired Internet and wireless ad hoc networks that are isolated and do not have any such access to the infrastructure.

The Internet Protocol (IP) [4] and other associated protocols [5] have served the world very well even during the explosive growth that has taken place over the years of their existence. To support possibly billions of Internet-accessible nodes (e.g. cell phones, automobiles and PDAs), a protocol change on the network layer to IP version 6 (IPv6) [6] is under way. IPv6 will enable cost-efficient availability of permanent IP addresses for all these devices, and many more not yet imagined. Manageable and scalable support for routers that are themselves mobile is useful for maintaining the robustness provided by dynamic IP routing and for providing support for general network configurations that are not easily handled otherwise. These *mobile routers* can provide access to arbitrary network topologies with no specific restrictions on the depth of forwarding paths or on their connectivity to fixed and/or mobile parts of the routing fabric.

Problems identified in *router mobility* (packet routing with mobile routers in mobile networks) can be considered as special cases of routing by nodes in ad hoc networks, which are being standardized by the Mobile Ad hoc Networking (`Manet`) working group [7]. Currently, the `Manet` working group have released AODV [8], Dynamic Source Routing (DSR) [9], Optimized Link State Routing (OLSR) [10] and Topology Broadcast Based on Reverse-Path Forwarding (TBRPF) [11] as experimental RFCs to be used as base routing protocols in with ad hoc networks. Although these are all IPv4-based protocols, they are all easily extended to support IPv6. For the purpose of this paper, we describe how such reactive and proactive protocols can be considered, under the assumption that the IPv4 address fields are simply expanded to be long enough for IPv6 addresses. For the particular case of AODV, we rely on the already existing specification for the AODV for IPv6 (AODV6) [12] for detailed experimentation with the operation of our proposals with IPv6-based ad hoc networks. A comparison between AODV and AODV6 shows that the protocols are not quite identical, but the major change is that the fields in the message header have been rearranged for better alignment. Such changes have no effect on address autoconfiguration procedures or gateway discovery. We expect that our recommendations in this paper will be immediately useful for these other base routing protocols just mentioned, once their IPv6 versions are specified.

Dynamic routing protocol solutions for ad hoc networks (including those mentioned above) meet requirements such as

- multihop forwarding capability;
- *loop freedom* for all routing paths;
- elimination of the *counting to the infinity* problem, avoiding nonconverging metric-based routing scenarios in distance vector protocols [13];
- low processing and memory requirements and

- self-starting operation without the need for user intervention.

These requirements are relevant when a dynamic routing protocol must operate in arbitrary, rapidly changing network topologies. Routing protocols considered within `Manet` also fit related applicability requirements. Our solution for Internet connectivity works with the various `Manet` protocols, but there are certain differences that depend on the choice of base protocol. As a simple example, the choice of protocol will affect the names and message formats of various routing control signals.

Whenever any node of a mobile ad hoc network (called a *Manet* node) comes into contact with a node that has connectivity to the global Internet, cooperation between these two nodes can provide global connectivity for every other node in the ad hoc network. The cooperative node that has global connectivity is called the *Internet Gateway* and is treated by every node within the ad hoc network (i.e. every *Manet* node) as a default router. This conforms well with standard treatment for default routers, especially if the model for the ad hoc network routing protocol determines next hops for the various ad hoc host and network destinations. This much can work even if the gateway node does not run any other routing protocol except the ad hoc routing protocol (which is needed so that it can answer requests for its own address). In our discussion, since the gateway node also runs the base ad hoc network routing protocol, it is also considered to be a Manet node.

If, in addition, nodes in an IPv6 ad hoc network need to receive packets from the global Internet as well as transmit them, then the gateway has to take steps so that it will be a forwarding node along the path for packets transmitted from within the Internet toward a Manet node as destination. This means that the gateway node might have to provide reachability information for the addresses of every other Manet node. In the Internet, reachability information is given by way of routing protocols such as Open Shortest Path First Protocol for IPv6 (OSPFv6) [14] Routing Information Protocol (RIP) for IPv6 (RIPng) [15] or Border Gateway Protocol for IPv6 (BGP) [16]; the gateway node has to be assigned a routing prefix, and it has to have the ability to forward packets toward any node whose address has that routing prefix as the leading bits of its address. For all Manet nodes that have addresses conforming to the routing prefix(es) advertised by the gateway, this will work fine. Our method enables Manet nodes to autoconfigure addresses that conform to IPv6 infrastructure routing requirements.

However, some Manet nodes must receive packets that are sent to addresses that do not conform to the set of prefixes advertised by the gateway. The gateway node should not advertise reachability for those topologically incorrect addresses. Otherwise, host routes for those particular nodes would have to be injected into the distributed database consisting of the routing tables for the Internet infrastructure routers, an approach that is known to be unscalable, unmanageable and difficult to secure. Instead, the Manet nodes acquire topologically correct addresses that conform to the advertised prefix and use the acquired addresses to enable reception of the packets delivered to the topologically incorrect address.

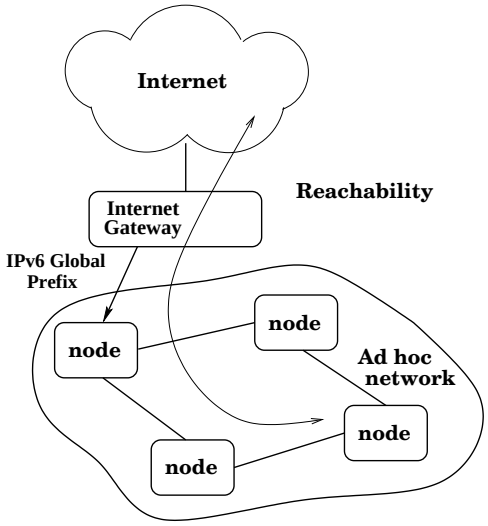


Fig. 3.1: Internet reachability from an ad hoc network.

Such persistent (‘*always-on*’) IP-address reachability is a very desirable property for Internet connectivity. Previous developments [17, 18] provide to wireless devices the same level of possibilities for Internet connectivity as wired computers, including such persistent connectivity wherever the physical links are available. This enables mobility regardless of the wireless link layer, and with all transport and application protocols. These solutions do not require any modifications to the network-layer routing infrastructure, in order to maintain the layered end-to-end communication model, backward compatibility and robustness of the Internet. Mobile IPv6 [18] provides this persistent reachability for IPv6 mobile nodes, by hiding the movement of a host away from its home domain. This combination of mobile node reachability and router mobility in IPv6, illustrated in Figure 3.1, is a likely requirement for future mobile Internet devices. If, as expected, the future Internet is mostly populated with mobile devices, scalability will be a key consideration for mobility management. For many small sensor devices, computational complexity has to be kept at a minimum to conserve battery power and avoid processing delays. Manual configuration is unacceptable for any such high-volume devices. Furthermore, protocol simplicity is also a requirement, since a complex solution would be difficult to develop on all the different network platforms that are going to be prevalent within the wireless Internet. Mobile IPv6 affords important advantages for making such wireless attachments, especially regarding router advertisement and address autoconfiguration.

We propose the following steps for managing ad hoc connectivity to the Internet:

1. If no address is currently configured, acquire a canonical site-local address.

2. If an advertisement is received, configure a globally valid IPv6 address ac-

cording to the information received in the advertisement.

3. When access *to* Internet is desired, but no globally unique IPv6 address is yet configured, solicit for Gateway information.

4. Set up a default route to Internet, and a host route to Gateway (if desired).

5. Form a topologically correct address using prefix information from the gateway.

6. When access *from* the Internet is desired, run Mobile IPv6 if the Manet node's persistent address is different than its topologically correct autoconfigured IPv6 address.

These steps are effective regardless of the base ad hoc network routing protocol in use nor on the routable scope of the address prefix advertised by the gateway. Therefore, the gateway could advertise a sitelocal prefix instead of a global prefix, without any change to the procedure; however, we typically imagine these operations being performed for global-scope routing prefixes. Furthermore, the addressability *from* the Internet for arbitrary IP addresses does not strictly depend on Mobile IP, but that is the method most likely to offer seamless connectivity. In this paper, we propose a method for connecting ad hoc networks to the Internet by way of Internet Gateways, then describe our experiments using AODV for IPv6 (AODV6) as the base protocol, followed by further experiments using Mobile IPv6.

The chapter is organized as follows: We first discuss the current problems of combining the existing IPv6 mobility protocols to allow node reachability in ad hoc networks, in Section 3.2. We list related work in Section 3.3. In Section 3.4, we describe our method for enabling the initial autoconfiguration of a canonical sitelocal address for a Manet node that does not have any existing IPv6 addresses at all. Then, in Section 3.5, we first present solutions for setting up node reachability to the fixed network from a node in an ad hoc network. This includes the operation by which nodes in the ad hoc network can acquire a default route, as described in Section 3.5.1. We describe how to route packets accurately on discovered routes through an Internet Gateway in Section 3.6. In Section 3.7, we discuss a solution for using Mobile IPv6 in an ad hoc network. One candidate ad hoc networking protocol for IPv6 is AODV for IPv6 (AODV6) [12]. It is described in Section 3.8. We apply our ideas for Internet connectivity to AODV6 as a case study in Section 3.8, after which we provide concluding observations for the designs in Section 3.9.

3.2 Internet Connectivity Basics

A wireless ad hoc network (mobile or not) has no preexisting infrastructure. It is formed on demand when two or more computers start to communicate with each other. To be able to work efficiently, nodes participating in the network must be

willing to forward packets destined to other nodes. The network is typically a graph with multiple hops between some end points owing to the limited transmission ranges of the individual wireless nodes.

Ad hoc mobile communication with the global Internet is problematic if only host-based routing is used. Each node, which is allowed to have an arbitrary IP address in an ad hoc network, would require a host route propagated to every router of the fixed network; clearly, this is an unscalable approach. A solution that hides the state space explosion caused by host routes is needed. Since we wish to design a protocol that can support ubiquitous mobility and connectivity support for all possible nodes and routers, including those moving to an ad hoc network from the fixed network, this IP address should be any globally routable (*unicast*) address. In this paper, we do not consider multicast addresses, which usually require specialized support from the routing protocol.

Ad hoc networks therefore should support efficient Internet connectivity, including mobility management. Our observation is that Mobile IP, considered as an access protocol, reduces the need for host route dissemination. By providing access with a topologically correct address and maintaining the address mapping of the mobile host/router only in binding caches of correspondent nodes, Mobile IPv6 reduces host route state maintenance to a small number of nodes.

When an IPv6 mobile node has become part of an ad hoc network, it may need to obtain a default route that it can use to transmit packets to destinations within the IPv6 Internet. A node in the fixed network finds a default router by means of router discovery. In IPv6, this is accomplished by way of Router Advertisement messages as specified in the Neighbor Discovery Protocol (NDP) [19]. Such advertisements include all the information that IPv6 nodes need to establish viable communications to the Internet by using the link carrying the advertisement. However, NDP cannot be used unchanged in mobile ad hoc networks since the messages used for next-hop default router discovery, the Router Advertisements and Router Solicitations, are used with link-local scope addresses and can only reach nodes one hop away (i.e. *neighbors*).

To simplify the process enabling a node to establish connectivity through an Internet Gateway in a multihop ad hoc network, we make some assumptions corresponding to some natural constraints:

- Gateways are routers located between the ad hoc network and the Internet, to provide Internet connectivity. No special assumptions are made, except that the gateways follow the protocol specifications in this chapter.

- The gateway advertises a topologically correct global routing prefix, so that packets transmitted anywhere in the Internet with the destination address belonging to that routing prefix can be routed toward the gateway. A node in an ad hoc network is allowed (but not required!) to have a preexisting arbitrary global IP address.

- A node that needs bidirectional connectivity to the Internet has to support a

discovery mechanism to find a globally reachable network prefix to be used in the ad hoc network.

- A manet node can use Stateless Address Autoconfiguration [20] for acquiring a topologically correct (routable) address within the ad hoc network. All IPv6 nodes are already required to support this procedure.
- Data forwarding within the ad hoc network, for packets initiated inside the ad hoc network, should use nonintrusive forwarding mechanisms that do not require any changes to nodes or routers (or routing protocols) in the fixed network.
- Any IPv6 node sending a packet to a Manet node does not have to have any knowledge that the node is in an ad hoc network.

Furthermore, a node that wishes to receive packets delivered to its persistent IPv6 address (if it has one) has to perform the following operations:

- learn from the gateway how to configure a topologically correct address and
- use this address as a *care-of address* with Mobile IPv6 so that this operation substitutes the router discovery part of Mobile IPv6 operation; the location update operations of Mobile IPv6 (using Binding Updates) remain unchanged.

In an ad hoc network, for a node to start communicating with other nodes in the Internet, the node must discover an Internet Gateway to obtain a globally routable prefix. This can happen as part of the boot-time procedures or it can happen as the mobile node migrates into an ad hoc network. The node makes use of the discovered prefix information to configure its network interfaces with globally routable IPv6 addresses. This discovery also provides IPv6 addresses of gateways and enables setting up default routes to the Internet through them, analogous to the way that IPv6 nodes on fixed networks configure default routers and prefix information from Router Advertisements. Gateways can unicast reply messages that include their own global prefix and IPv6 address. It is usually also important for gateways to discover routes toward such requesting nodes, since typically such nodes need bidirectional communications.

If a Manet node solicits prefix information, any intermediate node that has the requested information may supply it to the mobile node. As is typical with route discovery operations, such a requesting node may receive reply messages from multiple intermediate notes, each of which satisfies its request. The issues in selecting a particular default route are much the same as with IPv6 [19, 21] there are not yet well understood policies available to help with the decision process. For a Manet node, the gateway that is closest (as measured by hop count) would often be selected.

With the routing protocol in an ad hoc network running on the gateway, intermediate nodes can resolve the host route for the gateway by the exchange of routing

protocol messages. Subsequently, intermediate nodes can forward any packet to the gateway, which then can receive the packet and route it to its final destination. Care should be taken so that each intermediate node does not independently attempt to discover a route to the final destination address; this would result in unnecessary traffic overhead. Whether or not the final destination node exists inside the ad hoc network, intermediate nodes that do not have a host route for the destination would themselves forward the packet along their default route to the Internet. Although intermediate nodes may themselves obtain a default route by searching for the destination or for the gateways, these actions are obviously redundant. If intermediate nodes are involved with forwarding to default routes, we require those nodes to acquire their default route to the Internet at the same time the source node does, by means that are a very natural extension to the typical route discovery operation.

3.3 Related Work

In Mobile IPv4, there are some existing solutions for Internet connectivity to an ad hoc network, for example, MIPMANET [22], which includes the use of foreign agents. Mobile IPv6, however, does not define foreign agents. To be able to reach the Internet, mobile nodes using Mobile IPv6 need an Internet Gateway, which routes packets from nodes in the ad hoc network to nodes that reside in the Internet cloud (e.g. home agents and correspondent nodes) and *vice versa*. Lei et al. [23] propose another solution by modifying RIP [24] to work as a routing protocol among these nodes. RIP was not initially designed for being used in a wireless ad hoc network. The route table management was changed to allow the gateway node (for Reference 22, the Mobile IPv4 foreign agent) to have special status in the route table. A specialized route table manager was built to establish cooperative maintenance of host routes between arbitrary Manet nodes and routes to and from the IPv4 gateway. Cluster Gateways (CGs) have been proposed as a protocol-independent Internet access method [25]. A CG provides both a service access point and a Mobile IP foreign agent for ad hoc networks. If the CG works as a service access point, a gateway gives a kind of Network Address Translation (NAT) service for an ad hoc network. Otherwise, Internet access is given by Mobile IP operation, that is, triangle routing on Mobile IPv4. Every node in the ad hoc network has to register with the CG gateway to obtain Internet access services. However, they cannot get a globally routable address. Thus, both methods of CG are not appropriate when running Mobile IPv6 over an ad hoc network owing to the lack of a globally routable care-of address. The WINGS project [26] provides wireless Internet gateways over ad hoc networks. This effort targeted the ad hoc network routing protocol itself and did not initially focus on Internet connectivity and address assignments. However, they did have demonstrations interworking between WINGS-based ad hoc network and the Internet (for instance, by way of a satellite link and wired routers). They also demonstrated the WING protocol between two WING clients running the FreeBSD and the VIC Mbone tool.

3.4 Local Address Configuration

This section describes local address configuration for nodes which do not already have an IPv6 address when they join the ad net network. As in the IPv6 NDP, a local address can be configured whether or not router advertisements are present. We reserve the use of a canonical site-local prefix (MANET PREFIX) that is to be used in every ad hoc network for autoconfiguring an initial IPv6 address. This prefix has value fec0:0000:0000:ffff, with prefix length 64, which can also be written as fec0::ffff/64. We call any address formed with this special site-local routing prefix a manet-local address.

Every Manet node must configure a IPv6 manetlocal address, which can be used in further protocol operations (see Section 3.5.2). It chooses a candidate address from that routing prefix, which could be constructed by simply appending its own 64-bit IEEE address. The canonical prefix has site-local scope [27]; the site-local limitation prevents any opportunity for packets to leak into the Internet.

Unfortunately, choosing a candidate address is only the first step because the node already has to have a source IP address in order to check the uniqueness of its candidate IP address. For this purpose, the node also borrows another transient address to be used only for verifying the uniqueness of the candidate address. This transient address is chosen from the MANET INITIAL PREFIX, which is that obtained from the MANET PREFIX by adding 32 more zeroes to the former routing prefix, that is, fec0:0000:0000:ffff::/96. The configured actual address has to come from the part of the MANET PREFIX not overlapping with MANET INITIAL PREFIX. The transient address will only be in use for a few hundred milliseconds at most, so there is no significant danger of source address collisions even though the transient source address is used without verification of uniqueness. Any timeout for routes toward the transient address are set to be very short; furthermore, routes toward the transient address can be explicitly deleted after the uniqueness check, if the underlying protocol provides for such a mechanism.

After selection of a source IP address if necessary, the node performs a uniqueness check for the address to be configured, using a modified version of the IPv6 Neighbor Solicitation. Generally, the route discovery messages of the base routing protocol can be used with minimal modification to attempt to acquire a route to the candidate manet-local address; if a path can be acquired, then the address is already in use and must be discontinued as a viable candidate. In Reference [27], this modified protocol message is called the Address Request message (AREQ), and there is a corresponding message called the Address Reply message (AREP). If the selected address is already in use, then the AREQ message will be received by the node that is already using the address. This node will then send the AREP message in reply, causing the candidate address to be eliminated from consideration. This is very unlikely to ever happen if the candidate address was chosen wisely (or even at random), but if it does happen, a new candidate address and a new transient source address must be chosen, and the process started again. We have not designed this system to work with multicast addresses, but they are indis-

tinguishable from unicast addresses and not used elsewhere in this chapter. Anycast group members have to defend the anycast group address since it may be otherwise indistinguishable from a unicast address.

3.5 Obtaining Global Addresses

We may now presume that the Manet node has an address that has sufficient scope for use within the ad hoc network. To send packets to the Internet, a Manet node acquires information about an Internet Gateway and establishes appropriate routes to this gateway. The gateway may be allowed to distribute router advertisements periodically over the ad hoc network as a part of its NDP operation, requiring minimal changes to the current protocol. But, in most cases, the gateway cannot distribute the router advertisement across the ad hoc network because in wireless networks a link is not necessarily organized as a fully connected graph, as in wired networks. For example, consider a group of three nodes, where a Node A can hear two other Nodes B and C, but Nodes B and C can only hear Node A, and not each other. This leads to the well-known hidden-terminal problem. Every node is likely to have a different notion of the physical extent of their 'link'. Nodes B and C see two separate links, but it is reasonable for Node A to characterize its communications path to both Nodes B and C as just one link.

Because link-local packets must not be forwarded, it is not acceptable to use them for unicast in an ad hoc network, except for operations that are confined to a node's one-hop neighborhood (e.g. neighborhood sensing). Such operations are not considered in this chapter. Even if such periodic on-link advertisements were allowable, though, we would still prefer that they not be used because the cost of broadcasting packets periodically in an ad hoc network is very high. Every node has to process the packet and possibly to assist in its redistribution. This is expensive in terms of processing and bandwidth utilization and energy consumption. Still, for some scenarios and applications, a proactive solution might be more effective and utilize less energy. Such scenarios will be mentioned in later chapters.

According to these considerations, and as previously mentioned, a Manet node can request a router advertisement by some sort of solicitation and get back a reply or a modified router advertisement. Since the default router may now be multiple hops away, this also resembles a typical on-demand Route Discovery operation. The basic signaling of the global Internet access setup is illustrated in Figure 3.2.

We present two alternative solutions for requesting router information from the gateway. For both, all gateways must join the INTERNET GATEWAYS multicast group. The Manet node can acquire the necessary information either by extending the operations for route discovery that are typically present in the underlying routing protocol for the ad hoc network or by following the IPv6 Router Advertisement model of operation. We call the first of these two alternatives 'Gateway Request and Reply' and the second one 'Gateway Solicitation and Advertisement'.

The Global Router Request and Reply in Figure 3.2 must be set to be either

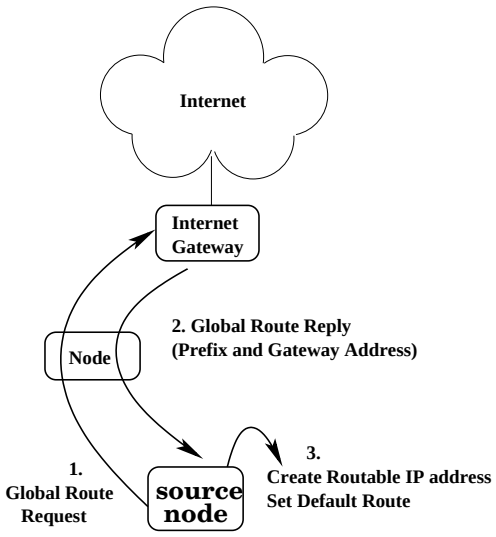


Fig. 3.2: Route request for address configuration.

Gateway Request and Reply or Gateway Solicitation and Advertisement, depending on the solution. In Section 3.5.1, we first explain, for the gateway discovery operations, the use of Gateway Requests and Replies. Afterwards we show the exchange of Gateway Solicitations and Advertisements.

Following the discovery operations, in Section 3.5.2, we will see how the Manet node creates a routable IP address, an operation that is common to both discovery solutions.

3.5.1 Internet Gateway Discovery

When a node performs address resolution in an ad hoc network, it needs to obtain a prefix with global (or, perhaps, site-local) scope from which to select a candidate IPv6 address. We describe this initial address configuration in this section, following the autoconfiguration protocol in the Internet Draft [27].

If a Manet node has no address at all when it joins the ad hoc network, it first configures an address as discussed in Section 3.4. While doing so, the node typically broadcasts an AREQ message, which will be received by all Manet nodes, including the gateway nodes within the ad hoc network.

For an address within the canonical site-local range MANET INITIAL PREFIX, a gateway can treat an AREQ as a request for routing information. In this case, the gateway will return a Gateway Advertisement to the requesting node. This has the effect of reducing the time required for the node to finish its autoconfiguration steps. This is also true if the gateway receives a Route Request (RREQ) from a node with an address within that site-local range.

The extensions in the following sections work for all protocols under consideration within the Manet working group, as well as all others known to the authors,

although the details are different in each case. For instance, if a backbone of nodes is to be established for selective broadcast of routing signals, a gateway node has to be sure it is reachable from one of the backbone nodes. If more than one gateway is available, a selection policy is needed to decide which gateway to use–perhaps by the number of hops or by some other priority. This selection policy is out of scope of this paper.

3.5.1.1 Gateway requests and replies

A node can use the route discovery mechanism of an ad hoc network routing protocol to obtain a global prefix and learn the Internet Gateway address. However, the base protocol has to be extended to allow the Internet Gateway to identify itself as having connectivity to the Internet and to allow the node originating the search to indicate that it is interested in finding such a gateway.

For this address resolution, we extend the route discovery scheme of the existing ad hoc network routing protocols and the NDP. For on-demand protocols, there is typically a RREQ message used to establish a route when one is needed. We extend the RREQ message to be useful with the special INTERNET GATEWAYS multicast address. Any gateway node can respond to such a RREQ by supplying a Route Reply (RREP) message in the underlying ad hoc routing protocol.

Proactive routing protocols should be extended to allow a gateway node to mark its advertisements with an indication that it belongs to the INTERNET GATEWAYS multicast group. Then, the path computation employed for selecting routes to a destination can also be used for obtaining a path to a gateway.

After sending the RREQ, the node should wait until all the gateways return a reply, for example, an AODV6 RREP. Getting a routing path to a gateway is not the same as getting a path to a general destination node because the request also has to carry with it the information that the desired destination is, in fact, a gateway and that prefix information should be included with the reply. For that case, we have defined an Internet-Global Address Resolution flag in the RREQ and reply messages of two on-demand ad hoc network routing protocols, DSR and AODV6. If the Internet Gateway finds the flag in a request from a Manet node, the gateway interprets this as the request for obtaining global prefix information and gateway addresses. Since intermediate nodes detect the new flag, they will rebroadcast the request over the rest of the ad hoc network. Therefore, the RREQ will reach the edge of the ad hoc network and be processed by the Internet Gateway.

The gateway does not further disseminate this route request.

First, the gateway checks the flag setting in the route request. If set, the gateway unicasts a RREP with the flag indicating the presence of the global prefix information and its own IPv6 address instead of a host route for the destination node. Each node that receives this RREP message relays it back to the source node (including the prefix extensions).

The INTERNET GATEWAYS multicast address is allowed to be the destination address. The requested address of the RREQ message can also be a global address for a subnet located outside the ad hoc network, for example, in the Internet. In this case, if the node owning the global address already resides within the ad hoc network, the requesting node is likely to receive both a default route with prefix information and a specific host route to the Manet node with the global address. The requesting node should, in the case of receiving two different replies, prefer the host route in order to avoid unnecessarily traversing the gateway node.

As mentioned previously, the gateway may also interpret the reception of an address resolution packet from a source address within the MANET INITIAL PREFIX as an implicit request for prefix information. The reply is then formulated in the same way as previously described when receiving a RREQ for a global address.

3.5.1.2 Gateway solicitation/advertisement

In this section, we describe an alternate method for acquiring gateway information, modeled on IPv6's Router Solicitation and Router Advertisement messages [19]. A Manet node sends an extended Router Solicitation, in order to prompt an Internet Gateway to generate an extended Router Advertisement. This advertisement contains the necessary information to configure topologically correct addresses, as well as auxiliary information for address lifetime and so on.

Because of the ambiguous scope of an ad hoc network link, we need some extensions for propagating these messages over multihop networks.

We created a new Manet (M) flag for both the Router Solicitation and Router Advertisement messages. In this paper, we call these new messages the Gateway Solicitations and Gateway Advertisements. If a receiving node finds this flag in either of these messages, it indicates that the message can be forwarded to a non-link-local address. Upon receiving the Gateway Solicitation, a gateway replies by sending a Gateway Advertisement message including its global prefix information and its own address. Note that we could have achieved the same result by defining a new Internet Control Message Protocol (ICMP) message type, instead of a new flag for the existing message type. If our approach is ever considered for further standardization, this alternative will no doubt get serious consideration.

A Manet node can solicit a Gateway Advertisement from available gateway by sending a Gateway Solicitation to the INTERNET GATEWAYS multicast address. The node may use an expanding ring search technique to disseminate the Gateway Solicitation message to the INTERNET GATEWAYS address using appropriate hop-limit values.

Non-gateway nodes in the ad hoc network also forward the solicitation if the hop limit has not already been reached. In addition, the intermediate nodes may need to set up a reverse path route to the requester, since the Gateway Advertisement messages will need to traverse the reverse path. This depends on the operation of the underlying protocol. DSR does not need such reverse-route setup (if the

Routing Table

Destination/Prefix	Next Hop Gateway
Defaul/0	Internet Gateway
Internet Gateway/128	Neighbor Forwarding the Reply

Fig. 3.3: Route request for address configuration.

solicitation carries a source route), but AODV6 does.

Whether the INTERNET GATEWAYS multicast address can be used as a broadcast address within the ad hoc network depends on the base routing protocol. If the node receiving a packet destined to the multicast address is not an Internet Gateway and if the hop limit allows it, the node must propagate the request ahead toward the INTERNET GATEWAYS address. Note that while this is a multicast address, no special multicast tree maintenance is needed, and the interior nodes should forward the request just as they would for any unicast destination address. If the routing protocol used within the ad hoc network does not support this, modifications may be needed for this to work [28]. Alternatively, the request could be broadcast at every node.

For the Dynamic Source Routing protocol [9], the draft [29] proposes a way of multicast and broadcast forwarding by using DSR's route discovery mechanism. In ad hoc networks running DSR, Gateway Solicitations and Advertisements can be exchanged between Internet Gateways and nodes by multicast and broadcast.

3.5.2 Address Configuration

After gateway discovery has taken place, the node has learned a global prefix, and possibly the address of an Internet Gateway serving the ad hoc network.

With this information, the node generates a global IPv6 address from the global prefix using its 64-bit interface ID. Since the node has already performed Duplicate Address Detection (DAD) for its Manetlocal address (as described in Section 3.4) before setting up the global address, a global address with a host number from this manet-local address is also unique. Many IPv6 nodes follow an analogous rule for link-local addresses, and we presume in this paper that all Manet nodes do the same. In the undesirable case, which is not yet prohibited by the base IPv6 specification, the Manet node may have to perform another DAD for this new address at the cost of additional start-up delay. Thus, we prohibit the undesirable behavior and require that all Manet nodes must acquire a manet-local address as described in Section 3.4.

3.5.3 Default Route Setup

Once a node has found a route to the Internet, it should set up a default route in its routing table, so that it can have a route for all the global addresses that are to be located in the Internet.

For routing protocols that do not maintain next hop information for the default route, a route table entry for the gateway needs to be kept in the route table along with information about the default route through the gateway. The node should set two routes into its routing table as shown in Figure 3.3. These entries should be held until the expiration of the lifetime provided in the reply to the global address resolution request and the router advertisement by the Internet Gateway. Before this lifetime expires, the node should refresh these routes and rerequest global Routing table prefix information from the Internet Gateway. This refreshment should be done periodically, either by the node individually or by the gateway for all such nodes collectively. The node can unicast the refreshment request to a specific gateway, or alternatively broadcast the request to the whole network again. The former method can allow the node to update its current Internet Gateway status and minimizes network congestion. The latter enables the node to quickly discover all Internet Gateways in an ad hoc network, some that may not have been previously available.

For routing protocols that can maintain just the next-hop default router, the information about the gateway may soon become inactive. After that point, the node only needs to keep the default route information (i.e. the next hop toward the gateway). No periodic refresh is needed.

If the node goes through an extended period during which Internet access is unnecessary, the default route to the Internet Gateway may expire. Whether or not the base routing protocol maintains the default route as a next hop, when the information is needed, it can once again be established on demand.

3.6 Internet Access Methods

When a Manet node has access to the Internet, the method for determining a route to a destination node could depend on whether that node is in the ad hoc network or it is reachable only by way of access to the general Internet. In the latter case, use of the default route is clearly needed, but otherwise it is likely to be better if a specific host route is available. Unfortunately, when a node wishes to send data to a destination, there may not be any good way to make the distinction. In fact, if the Internet Gateway (or some other Manet node!) is also serving as a home agent (see Section 3.7), even addressability within the ad hoc network is not enough to determine whether a host route to the destination can be obtained.

This section describes how to send packets on routes discovered through an Internet Gateway.

3.6.1 Route Discovery Algorithm

Whenever a Manet node is about to send a packet, it first refers to its routing table to obtain an appropriate route for the destination. If a default route or a route through the gateway to the Internet is obtained, then the Manet node has to decide whether a shortest route is required.

If the node gets the default route instead of host route, the node sends RREQs for the destination, because the destination node might be located in the same ad hoc network.

If the node gets the host route, the node sends packets toward the destination according to its host route.

If the node cannot find any route in the routing table, it starts gateway acquisition operation as described in Section.

If no gateway is found (i.e. no default route is available), the packet is dropped.

```
destaddr := destination_address;
route := rte_lookup(destaddr);
/* search route table */
if (route == null) {
   initiate_gateway_discovery();
   route := rte_lookup(destaddr);
   if (route == null) {
      drop_packet();
   return();
   }
}
if (route == default) {
   if (shortest_route_required) {
   route := route_discovery(destaddr);
   }
   send_packet(route);
}
else { /* ASSERT: route is host route */
   send_packet(route);
}
```

The above algorithm shows how to determine a route to the destination. If the node finds a host route during the route table lookup, it can start transmitting packets to the destination. The destination is typically located in the same ad hoc network. If the node does not find any routes (not even the default route), the node should start the Gateway Discovery operation as described in Section 3.5.

If the route table search returns the default route, and it is important to have the shortest route to the destination, the node should start the route discovery mechanism by sending RREQs for the destination address. If the node does not get any route replies, it proceeds as if the destination were on the Internet, external to the ad hoc network, and sends packets along the default route toward a gateway. Intermediate nodes should already have the default route or route to the gateway owing to some previous route discovery operation, so they will not send RREQs or

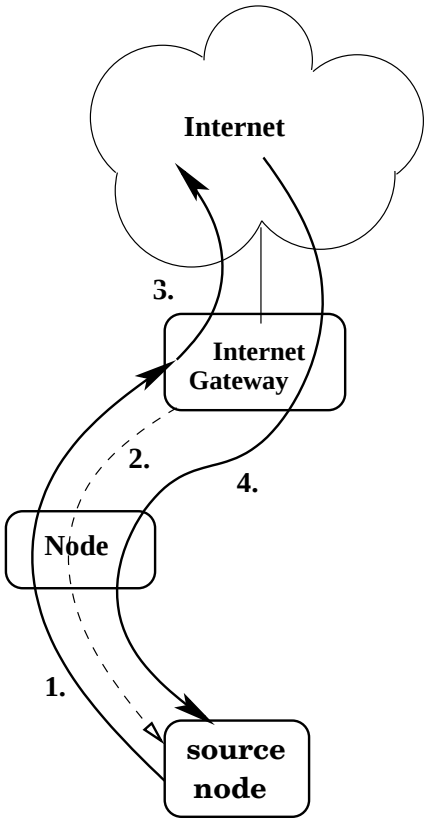


Fig. 3.4: Sending packets with routing header.

Route Errors (RERRs) for the destination. If the node does receive a RREP, it sets a host route for the destination and sends packets according to the received route. Note that the route may be a route to a particular subnet in the ad hoc network, in case the underlying protocol supports subnet addressability and route discovery for subnet prefixes.

3.6.2 Sending Data via Internet Gateway

Once the Internet Gateway has been discovered, the Manet node has a default route to the Internet. The exact nature of the default route depends on whether the underlying base routing protocol supports next-hop forwarding. When we propose that a routing header should be used specifying the gateway's address as an intermediate routing point toward the destination. DSR is such an underlying protocol; the proposed technique is a natural extension of the way that DSR already uses source routing, however, and does not place additional burden on the base routing protocol. As mentioned previously, we do not consider in this paper the problem of selecting from several possible default routes, when there are several gateways.

If the underlying protocol supports next-hop forwarding to the default router,

the node sends the packets to the global IPv6 destination address and relies on the next-hop routing in intermediate nodes. The sender node will not have to specify an explicit route to the Internet Gateway. Each intermediate node can decide independently how to route the packet efficiently out of the ad hoc network. The source of the packet in the ad hoc network does not typically require the address of the gateway, only the assurance that one of its neighbors is on a good path toward any of the possibly several gateways; either the first or the shortest path could be used.

When the routing protocol does not support such next-hop forwarding, the destination node should use an IPv6 routing header to make sure that the gateway node is listed as an intermediate routing point along the way to the destination. Using the routing header, packets are routed to the gateway as the first destination. For this to work, a node needs more information (the IPv6 address of the gateway) and exerts more control over the communication path. The sending node puts the gateway's address in the destination address of the IPv6 header and the final destination address in the routing extension header. When the gateway receives the packet, it will retrieve the actual destination from the routing header and insert it into the destination IP address field of the IPv6 header. More general formulations are possible for the routing header but are not covered within the scope of this paper.

As shown in Figure 3.4, the source node has a default route to the Internet Gateway. The steps in the figure are as follows:

1. The Manet node sends a packet through the Internet Gateway using a routing header. The gateway is typically then the overt destination of the packet.

2. If the Internet Gateway finds a host route for the destination that is faced toward the packet's incoming interface, the gateway returns a routing control signal to the source node, which we call a Route Update Request. This could be a routing protocol message, like the RERR message in DSR or AODV6, or the Gratuitous RREP in AODV6, which notifies the sender node that the destination node is located inside the ad hoc network and that the node should try to find a host route instead of using the default route for the node. Alternatively, the gateway could also send a ICMPv6 [30] Redirect Message [28].

3. Typically, the Internet Gateway will find a usable route for the final destination, so it forwards the packet toward the destination. Note that if the gateway were configured for operation within some larger domain that nevertheless did not offer Internet connectivity, this step could fail. In such cases, an ICMPv6 Destination Unreachable should be returned to the source Manet node.

4. When an Internet Gateway receives a packet from the Internet destined to a Manet node, this node can be reached without any special operation. The node already has a topologically correct global IPv6 address and the Internet Gateway routes the packet to the node along the host route or a source route maintained by ad hoc network routing protocols.

Suppose the destination is located within the ad hoc node network but that the source Manet node reaches the destination via a next hop serving as its default route toward the Internet. In this case any intermediate node that knows the host route for the destination may route the packets to it directly, without the knowledge of the source node. The source node should be notified that its packets are routed directly, instead of using the default route toward the gateway. For this purpose, if supported in the underlying ad hoc routing protocol, the intermediate node should send the source node a gratuitous RREP. This message indicates the current route table information at the intermediate node and enables the source node and each upstream node along the way to create an appropriate route table entry for the destination.

If the routing header is used, on the other hand, every packet is explicitly routed to the gateway. When the gateway detects that the destination is located inside the ad hoc network, it may optionally send a Route Update Request control message to the source. In either case, after receiving the control message, the Manet node may send a RREQ for the destination address and learn a new, more direct host route.

3.6.3 Route Examination/Determination

During communication, the network topology may change owing to node movement. To help update inaccurate routes, we present two methods for detecting the availability of a route. This subject is revisited in Section 3.7.2, after discussion of Mobile IPv6.

A gateway manages host routes in the routing table for the nodes in its ad hoc network because the gateway must often possess routes to nodes that need to receive packets from the Internet. When a packet arrives from the Internet, the gateway searches its routing table for the destination address of the packet's IPv6 header. If no route is found, and the underlying routing protocol is table-driven, then an ICMPv6 Destination Unreachable message is returned to the source of the packet. For on-demand protocols, the gateway initiates a route discovery operation for the destination. If no route is found, again an ICMPv6 message is returned to the sender.

From the other direction, whenever the gateway receives packets on its ad hoc network interface, it again searches its routing table for the destination address of the packet's IPv6 header. If a host or network route is found, which is routable within the ad hoc network, then the destination belongs to the ad hoc network. Therefore, the Internet Gateway can send a Route Update Request control message to the source node. Since this searching of the routing table occurs anyway during general forwarding operation on Internet Gateways, the extra overhead should be minimal. When the source node receives the routing control signal, it can initiate a new route discovery operation if needed.

A Manet node that receives ICMPv6 Destination Unreachable messages after sending packets to a destination based on a host route entry must invalidate that host route entry. If needed, the node can then discover the route by initiating a new

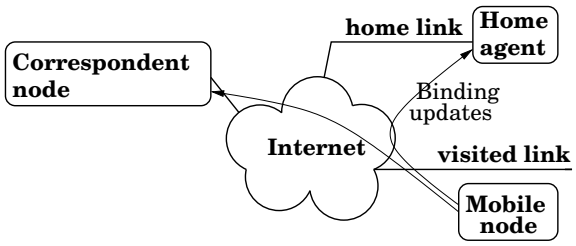


Fig. 3.5: Mobile IPv6 - sending binding updates.

route discovery operation. If the node receives ICMPv6 Destination Unreachable messages when the default route is used, the node should retry route discovery for the destination. If the node still does not receive any route replies, the node should discontinue route discovery and cease sending packets and RREQs for the destination for a while. This typically involves sending a signal (e.g. Destination Unreachable) to the application. The previous information about the route to the destination may nevertheless remain useful for some purposes, and so should be maintained temporarily if called for by the base routing protocol even if the route table entry is invalidated.

3.7 Mobile IPv6 Operation

If the mobile node has a persistent IPv6 address, it can be used as a Mobile IPv6 home address to provide always-on reachability from the fixed Internet. In this section, all Manet nodes are considered to be also mobile nodes running Mobile IPv6. Using the protocols we have defined, in conjunction with Mobile IPv6, the node's IP address remains accessible even when the mobile node moves between ad hoc networks connected to different points of the fixed network. Thus, mobility becomes transparent to applications, so that they can continue work without any modification. DNS name records do not need to be updated with new IP address information, so that the mobile node maintains its well-known identity from the point of the view of the rest of the Internet.

Mobile IPv6 does this by providing a means for any fixed network IPv6 address, the home address, to be reachable when the owner of the address, the mobile node (MN), is in a topologically incorrect place. When the mobile node arrives at a network other than its home link, it configures a care-of address for network access. The mobile node uses a Binding Update to register this address with its home agent (HA), a specialized router on the mobile node's home link. This agent then acts as a proxy for the home address of the mobile node, capturing packets sent to this address and encapsulating them for delivery to the registered care-of address (Figure 3.5).

Mobile IPv6 also provides route optimization by which a mobile node directly informs its communication end points, the correspondent nodes, about its location

by sending a Binding Update, similar to that sent to the home agent.

Route optimization allows for direct data communication between the mobile node and its correspondent nodes. The binding update maps the care-of address to the home address and establishes a binding cache entry for the mobile node into the correspondent node. Knowing this, the correspondent nodes can insert a routing header with a home address to data packets destined to the mobile node so that normal IPv6 routing will forward these packets directly to the care-of address of the mobile node. The mobile node can then easily supply the payload to its application, which is awaiting the data at a transport end point anchored with the home address.

Data packets originating from the mobile node contain a home address option specifying that these packets should be considered originating from a transport end point with the home address rather than the careof address in the IPv6 source address field of these packets.

3.7.1 Mobile IPv6 Operation on Ad hoc Networks

In Section 3.5, we have presented methods for enabling Manet nodes to configure a globally routable address. Once a globally routable address is configured, the node can initiate typical applications such as web browsing and DNS queries.

Mobile IPv6 uses neighbor discovery as part of its movement detection with the acquisition of a globally routable address. A mobile node uses the address built from the locally advertised prefix as its care-of address when performing a home registration. For a Manet node, the Internet Gateway replaces the local router and a Gateway advertisement replaces the Router Advertisement. The address configured from the Manet routing prefix from the Gateway advertisement is usable as a care-of address. When the base routing protocol is an on-demand protocol such as AODV6 or DSR, any Manet node using Mobile IPv6 should not expect to receive periodic Router Advertisements (or Gateway Advertisements), since for large ad hoc networks this periodic flooding is too expensive. For this reason, we expect that movement between separate ad hoc networks to be somewhat more time-consuming than movement between points of attachment to the fixed Internet.

If the prefix of the acquired address matches the statically known home network, a mobile node considers itself to be at home. Otherwise, if the prefix does not match the home prefix, the node performs a home registration using the global IPv6 address as the care-of address. If no home registration is needed, because the locally advertised prefix matches the routing prefix from the Manet node's home address, then we can say that the mobile node is at home in the ad hoc network. The mobile node would send packets to its home agent by way of a host or network route, just as it would with any other destination known to reside within the ad hoc network.

If a statically configured mobile prefix is known, dynamic home agent discovery may be necessary before the home registration. The home agent anycast address [18] can become routable within the ad hoc network by using the same

sort of route discovery actions as would be used with any other unicast destination in the ad hoc network.

It is a matter of policy, which should be selectable by the application [31], whether a mobile node should use its care-of address or home address for establishing its end-to-end communications with another application end point. Either of these addresses can be host-routed within the ad hoc network. Since the care-of address is topologically correct, packets to that address are more likely to stay within the ad hoc network. On the other hand, the home address is (as always) more likely to remain available to the communications partner, for example, if the mobile node moves back to the Internet or even to another mobile ad hoc network connected to the Internet at some other location.

3.7.2 Route Examination for Mobile IP

After a home agent forwards a packet to a mobile node by means of encapsulation, the mobile node normally sends a binding update to the correspondent node (i.e. the original sender) to create or update a binding cache entry associated with the mobile node's home address. Before this binding update is sent, the mobile node compares its network prefix value with the source address of the incoming packet. The mobile node can in this way learn whether the source node is located in the same ad hoc network. If this source node is local, the mobile node sends a RERR to force the sender node to discover the host route for the home address, instead of sending a binding update. The RERR message sent by the mobile node instructs the source node to get a new host route and establish a reverse-route path in intermediate nodes.

This operation could also be performed by the home agent. Upon receiving a packet from the correspondent source node, the home agent compares the prefix of the mobile node's care-of address with the incoming packet's source address. If the prefixes match, the mobile node is located in the same ad hoc network as the sending node. The home agent can then send an ICMPv6 Unreachable Message or ICMPv6 Redirect Message to prompt the correspondent source node to update its route.

3.8 AODV6 Case Study

In previous sections, we have presented methods by which Manet nodes can configure topologically correct addresses and use them for access to the global Internet via Internet Gateways. As a case study we have implemented our ideas using one of the existing ad hoc routing protocols, AODV6 (Ad Hoc On Demand Distance Vector protocol for IPv6). We begin this section by explaining the basic mechanism of AODV, which is exactly the same for both IPv4 and IPv6. The specific message formats for AODV6 are also presented. Afterwards, we explain the changes needed in the AODV protocol for providing Internet connectivity.

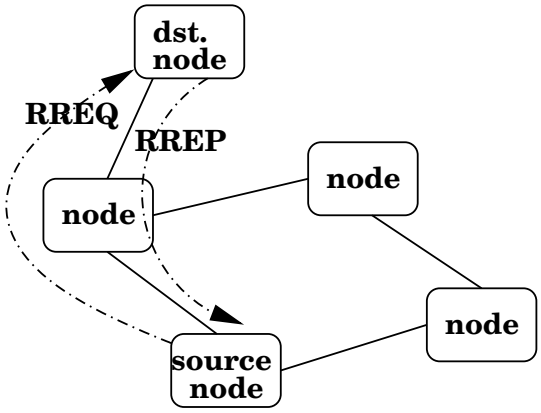


Fig. 3.6: Ad hoc On-Demand Distance Vector protocol.

3.8.1 AODV Description

AODV is an on-demand routing protocol that uses repeated route discovery to establish routes. A node that needs a route to some destination broadcasts a RREQ packet across the network. When either the destination or an intermediate node receives the RREQ, it responds by sending a RREP unicast back to the node as shown in Figure 3.6. Once the source node receives the RREP, it can begin using the route for data packet transmissions.

Routes in AODV are considered to be temporary and are marked as active during the time they are in use and seem to be capable of transporting data. When a route is no longer in use, it will expire and eventually be expunged from the route table, governed by the value ACTIVE ROUTE TIMEOUT (a few thousand milliseconds). In this way, the route table is modeled as a cache for routes. The improved delay characteristic of AODV is largely due to its careful maintenance of the cached route information. AODV does not often supply a stale route when one is needed for a new application between two Manet nodes. It is more likely to initiate the process of route discovery instead of using stale routes. Routes that have very recently been useful, however, are still kept available until a short time-out expires.

The route discovery operation itself (using RREQ and RREP as shown in Figure 3.6) requires that the node sending the RREP have a route back to the source of the RREQ. This reverse route could be cached at a large number of Manet nodes, since the RREQ is often flooded to every node in the ad hoc network. Such reverse routes have a much shorter time-out (REVERSE ROUTE TIMEOUT, on the order of a few hundred milliseconds) before they are expunged. Route maintenance in AODV makes use of RERR messages. When a link breaks in an active route, the node upstream of the break sends a RERR to each upstream neighbor (precursor) that was using that link to reach the destination. The RERR message lists each destination that is now unreachable owing to the loss of the link. When

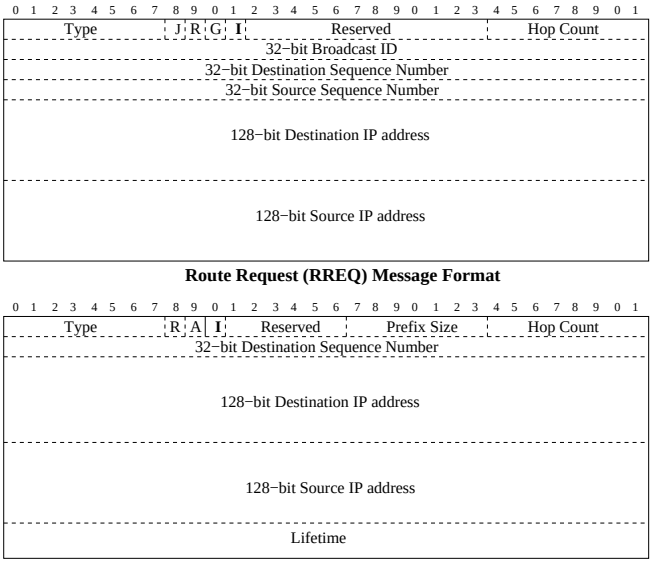


Fig. 3.7: AODV6 message formats (a) Route Request (RREQ) message format; (b) Route Reply (RREP) message format.

a source node receives a RERR, it may reinitiate route discovery if it still needs the route. AODV requires that such routes be maintained even after being invalidated, for long enough (DELETE PERIOD) to avoid supplying erroneous RREP information. The DELETE PERIOD is selected to be long enough to handle problems caused by Manet node reboots and other ways that protocol messages can go unanswered.

3.8.2 Internet Connectivity for AODV6

When a node issues a RREQ for validating a candidate global address, it can use an arbitrary address of scope larger than link-local from one of its interfaces as the source address in the IPv6 header. This can be a persistent, already allocated global address or a temporary address created with the MANET PREFIX (i.e. the manet-local address; see Section 3.4). For AODV6, Manet nodes take special action when installing reverse routes for a node initiating autoconfiguration. Such autoconfiguration packets will appear to emanate from a node with the source IP address within the address range MANET INITIAL PREFIX. Routes toward such autoconfiguring nodes should never be marked as active routes. Their lifetime should be initialized to the value REVERSE ROUTE LIFETIME.

The AODV6 node broadcasts the RREQ to the INTERNET GATEWAYS address. Figure 3.7 shows the modified AODV6 message formats. We have defined a flag, the Internet-Global Address Resolution flag (I), for the RREQ and RREP [28] messages. This flag indicates that the message is used for gateway discovery.

When an Internet Gateway receives a RREQ, it checks its routing table and

updates the reverse route to the node with the source address in the RREQ. If the Internet Gateway finds the I flag in the RREQ, the Gateway constructs a RREP with the I flag set, the prefix length used by the gateway, and its own IPv6 address and unicasts the RREP back to the requesting node. The IPv6 header field is built as in normal AODV6 operation. The global prefix information is derived from the Destination IP Address and Prefix Size fields in the RREP.

After a node acquires a topologically correct global IPv6 address, it deletes its temporary address that was formed from the MANET INITIAL PREFIX (see Section 3.4). A gratuitous RREP could be broadcast to create reverse routes toward the newly addressable Manet node in the intermediate nodes, but we have not implemented this. The Internet Gateway also updates its routing table entry with the address of the new Manet node.

If the node sends a packet to an Internet destination without a routing header, some intermediate nodes may generate RERRs, as specified by the AODV specification, because they do not have an active route to the packet's destination. To avoid such unnecessary RERRs, a default route can be maintained as an active route. When an intermediate node receives a packet for which it does not have a host route, it forwards the packet according to this default route. The intermediate node should also insert the previous hop as the precursor for the default route if it does not already exist in that list. Typically, however, the precursor list should already have the previous hop in the precursor list as a natural result of the original RREP message by which the default route was supplied.

If the node uses a routing header, the destination address in the IPv6 header should be the Internet Gateway IPv6 address. The intermediate nodes on the route path to the Internet Gateway have this host route and intermediate nodes do not have to generate RERRs for nonlocal outgoing packets.

3.8.3 Implementation

We have implemented AODV6 with Gateway Discovery as mentioned in Section 3.5.1, which enables global access. Our AODV6 routing daemon runs on LINUX platforms. We have not yet implemented the extended router advertisement nor solicitation of NDP; our Manet nodes discover Internet Gateways with extended RREQs and RREPs.

Our testing environment consists of three wireline connected AODV6 Manet nodes and an external correspondent node. One of the AODV6 nodes is the Internet Gateway with two network interfaces, one for the Internet and the other for the ad hoc network. One of the other two Manet nodes is the mobile node running Mobile IPv6. After acquiring a default route to both the Internet Gateway and a global IPv6 address, the mobile node sends a binding update to the home agent and to the correspondent node for route optimization. Manet nodes addressable using the gateway's advertised routing prefix can communicate with nodes in the Internet without Mobile IPv6. Since our AODV6 implementation uses a routing daemon running from user address space, while the routing table is maintained in

the kernel, we defined a new raw socket for interactions between the kernel and the daemon. If the kernel does not have an appropriate route for a destination, the kernel notifies the daemon through the raw socket to send RREQs across the ad hoc network.

Movement detection of Mobile IPv6 is triggered whenever the AODV6 daemon receives the new global IPv6 routable address. An IPv6 header includes the Home Address option and a Binding Update if it is needed. For implementations using a routing header, this requires use of two destination options after the routing header; more recently, the Binding Update has been modified to fit within a new header called the Mobility Header, but we have not yet implemented that new specification. Most current IPv6 and Mobile IPv6 implementations on Linux cannot carry two destination options after the routing header owing to an implementation limitation. We therefore extended the LINUX IPv6 implementation to store both options after the routing header in the IPv6 header.

3.9 Conclusion & Future Work

We have discussed the problems that we have encountered while attempting to connect nodes in an ad hoc network to the Internet with mobility support in IPv6 networks. We have presented solutions for address resolution, showed ways to gain Internet access by next-hop routing or by use of a routing header and have briefly described Mobile IPv6 operation in ad hoc networks. These problems include the following:

- site-local address acquisition and Duplicate Address Detection;
- acquiring a routing prefix from an Internet Gateway;
- establishing a default route and a host route toward the gateway;
- formulating a globally unique and topologically correct IPv6 address using the acquired routing prefix;
- soliciting gateway information whenever needed;
- when it is unknown whether a destination is present in the ad hoc network, determining whether to acquire a host route or using the default router;
- using the globally unique IPv6 address with Mobile IPv6;
- modifying the IPv6 ICMPv6 Router Solicitation and Advertisement messages to work across multihop networks;
- extending the route discovery mechanisms for on demand routing protocols to enable gateway discovery.

In most cases, we have discovered that the necessary extensions are quite natural. We have been able to formulate solutions for the above problems that work with the principal candidate routing protocols that are under consideration within the [Manet] (mobile ad hoc networks) working group of the IETF.

In our proposal, a Manet node with a need for global communication contacts an Internet Gateway by either sending a modified Router Solicitation, called Gateway Solicitation, or relying on a routing protocol route discovery functions. When the gateway receives one of these messages, it unicasts a response back to the requesting node, specifying its globally routable prefix and IPv6 address. The node then uses this information to configure an address that is globally reachable throughout the Internet. With Mobile IPv6, the mobile node can use this address as its care-of address and make a Binding Update to its Home Agent.

When sending packets to the Internet, the node can either use a routing header specifying the Internet Gateway as the first destination and rely on ordinary ad hoc routing to route the packet to the gateway or send the packets through the default route, relying on intermediate nodes to forward the packet toward the destination.

Our AODV6 routing protocol implementation uses an extra flag, called the Internet-Global Address Resolution flag, so that the node-gateway signaling can work as efficiently as possible. Along the way we fixed certain parts of the Linux IPv6 implementation, work that may be useful in many other contexts. We have shown that it is possible to implement connectivity between ad hoc networks and the Internet, with only slight modifications to the existing specifications.

In the future, we would like to revisit the problem of selecting between multiple Internet Gateways. In fact, it may be better to use multiple gateways simultaneously, depending on which one offers a shorter path to a particular Internet destination. This will require per-destination (or at least per-prefix) signaling. This is merely one instance of service selection for ad hoc networks, when a Manet node has the choice of several nodes offering a needed service.

We would also like to investigate ways to mark ad hoc networks as domains so that a mobile node could more easily distinguish between different ad hoc networks. This may involve borrowing some relevant ideas from OSPF. Finally, we would like to consider the possibility of using an ad hoc network as a transit network for foreign traffic, where both the source and the destination nodes are allowed to lie outside the ad hoc network.

BIBLIOGRAPHY

[1] IEEE 802.11 Committee, Alpha Graphics #35, 10201 N.35th Avenue, Phoenix AZ 85051, “Wireless LAN Medium Access Control MAC and Physical Layer PHY Specifications”, IEEE Standard 802.11-97, June 1997.

[2] J. Haartsen, “Bluetooth - The Universal Radio Interface for Ad Hoc Wireless Connectivity”, Ericsson Review No. 3, 1998.

[3] “Dedicated Short-Range Communication”, http://www.its-standards.net/, June 2002; The DSRC standards are still in the approval cycle.

[4] J. Postel,. “Internet Protocol”, Request for Comments (Standard) 791, Internet Engineering Task Force, September 1981.

[5] J. Reynolds, R. Braden, S. Ginoza, L. Shiota, “Internet Official Protocol Standards”, Request for Comments (Standard) 3000, Internet Engineering Task Force, November 2001.

[6] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification”, IETF RFC 2460, December 1998.

[7] S. Corson and J. Macker, “Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”, IETF RFC 2501, January 1999.

[8] C. Perkins, E. Royer, and S. Das. “Ad hoc On-demand Distance Vector (AODV) Routing”, IETF Internet-Draft, work in progress, January 2002.

[9] J. Broch, D. Johnson, D. Maltz. “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks”, Internet Draft, work in progress, Internet Engineering Task Force, February 2002;

[10] P. Jacquet, P. Muhlethaler, A. Qayyum. “Optimized Link State Routing Protocol”, Internet Draft, work in progress, Internet Engineering Task Force, draft-ietf-manet-olsr-06.txt, September 2001.

[11] R.G. Ogier, F.L. TemplinL, B. Bellur, M.G. Lewis. “Topology Broadcast Based on Reverse-Path Forwarding (TBRPF)”, Internet Draft, work in progress, Internet Engineering Task Force, draft-ietf-manet-tbrpf-05.txt, March 2002.

[12] C. Perkins, E. Royer, and S. Das. “Ad hoc On-demand Distance Vector (AODV) Routing for IP version 6”, IETF Internet-Draft, work in progress’, November 2000.

[13] J. Garcia-Luna-Aveces. “Loop-Free Routing Using Diffusing Computations”, IEEE ACM Transactions on Networking, Vol. 1, No. 1, February 1993.

[14] R. Coltun, D. Ferguson, j. Moy. “OSPF for IPv6”, Request for comments Comments (proposed standard), Internet Engineering Task Force, December 1999.

[15] G. Malkin, R. Minnear. “RIPng for IPv6”, Request for Comments (Proposed Standard) 2080, Internet Engineering Task Force, January 1997.

[16] Y. Rekhter, T. Li. “A Border Gateway Protocol 4 (BGP-4)”, Request for Comments (Draft Standard) 1771, Internet Engineering Task Force, March 1995.

[17] C.E. Perkins. “IP Mobility Support”, Request for Comments (Proposed Standard) 3220, Internet Engineering Task Force, December 2001.

[18] D. Johnson, C.E Perkins. “Mobility Support in IPv6”, Internet Draft, work in progress, Internet Engineering Task Force, March 2001.

[19] T. Narten, E. Nordmark, and W. Simpson, “Neighbor Discovery for IP Version 6 (IPv6)”, Request for Comments (Proposed Standard) 2461, Internet Engineering Task Force, December 1998.

[20] S. Thomson and T. Narten, “IPv6 Stateless Address Autoconfiguration”, Request for Comments (Proposed Standard) 2462, Internet Engineering Task Force, December 1998.

[21] R. Draves. “Default Address Selection for IPv6” Internet Draft, work in progress, Internet Engineering Task Force, June 2002.

[22] U. Jönsson, F. Alriksson, T. Larsson, P. Johannson and G. Maquire Jr,“MIPMANET - Mobile IP for Mobile Ad Hoc Networks. ” Proceedings of First Annual Workshop on Mobile Ad Hoc Networking and Computing, MobiHOC, August 2000.

[23] H. Lei, C. Perkins, “Ad Hoc Networking with Mobile IP.” Proceedings of 2nd European Personal Mobile Communication Conference, September 1997.

[24] G. Malkin. “RIP Version 2”, Request for Comments (Standards Track) 2453, Internet Engineering Task Force, November 1998.

[25] A. Striegel, R. Ramanujin and J. Bonney. “A Protocol Independent Internet Gateway for Ad-Hoc Wireless Networks”, fProceedings of Local Computer Networks (LCN) 2001, Tampa, Florida, November. 2001.

[26] J. Garcia-Luna-Aceves, C. Fullmer and E. Madruga and D. Beyer and T. Frivold. “WIRELESS INTERNET GATEWAYS (WINGS)”, Proceedings of IEEE MILCOM’97, Monterey, CA, pp. 1271-1276., November. 1997.

[27] C. Perkins, J. Malinen, R. Wakikawa, E. Belding-Royer and Y. Sun, “IP Address Autoconfiguration for Ad Hoc Networks”, IETF Internet-Draft, work in progress, November 2001.

[28] R. Wakikawa, J. Malinen, C. Perkins, A. Nilsson and A. Tuominen, “Global Connectivity for IPv6 Mobile Ad Hoc Networks”, IETF Internet-Draft, work in progress, November 2001.

[29] J. Jetcheva, Y. Hu, D. Maltz, D. Johnson. “A Simple Protocol for Multicast and Broadcast in Mobile Ad Hoc Networks”, Internet Draft, work in progress, Internet Engineering Task Force, draft-ietf-manet-simple-mbcast-01.txt, July 2001.

[30] A. Conta and S. Deering, “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.”, Request for Comments (Draft Standard) 2463, Internet Engineering Task Force, December 1998.

[31] X. Xinhua Zhao, C. Claude Castelluccia, M. Baker. “Flexible network support for mobility”, In 4th ACM Intl Conference on Mobile Computing and Networking (Mobicom 98), 1998.

4. CHAPTER IV

Routing in Hybrid Ad hoc Networks using Service Points

4.1 Introduction and Background

Mobile ad hoc networks (or multihop packet radio networks) consist of mobile nodes that communicate with each other over multihop wireless links. Each node in the network also acts as a router forwarding data packets for other nodes.

Table-driven or proactive protocols can become expensive in terms of control overhead, because each node in the network must maintain routing information for every other node, although the node only occasionally handles traffic destined for some of the nodes. To address the scaling problem of table-driven routing, on-demand routing protocols have been proposed for ad hoc networks. Nodes running such protocols set up and maintain routes to destinations only if they are active recipients of data packets. However, when routing information between only a few sources and destinations is constantly being maintained on-demand, it might be more attractive to use the proactive approach for these nodes, while on-demand routing is used between less accessed nodes. Similarly, if a large number of nodes frequently wants to exchange information with a few nodes, if might be more effective to proactively maintain these routes. This motivates the interest in a more hybrid approach to routing in ad hoc networks.

Recently a node-centric approach for routing in ad hoc networks was presented [1]. The idea here is that in many practical scenarios, certain nodes provide special services that are being requested throughout the network. For example, when ad hoc networks are wireless extensions of the Internet, these nodes may act as DNS servers, Internet Access points, web proxies or AAA servers. Services can also be local, for example locally stored data or database information. These nodes that host special services, and therefore have a higher likelihood of communicating with the rest of the network, are called *netmarks*.

The landmark hierarchy [2] is another node-centric routing approach that has been designed for proactive routing in large networks. Packets are not routed towards the destination address, but rather towards an area, the landmark radius, in which the destination is located.

Existing dynamic routing protocols for ad hoc wireless networks can be classified into two categories according to their design philosophy: *proactive* or *reactive* depending on how routes are computed and maintained. Proactive protocols maintain routes to every other node in the network independently of any data traffic

pattern. This has the advantage that when packets need to be sent, a route to the destination is typically available. Reactive or on-demand protocols, create and maintain routes only on a "as needed" basis. Thus, when a route is needed, some sort of *Query/Reply* search procedure is employed. When a route is no longer used, it eventually times out and is removed from the routing table.

Homogeneous ad hoc networks has been seen to suffer from poor scalability because most of the bandwidth is consumed by forwarding packets. Proactive protocols do not scale well because of the imposed control overhead. Reactive protocols are expected to behave better for larger networks. The use of route caching and local repair reduce the impact of flooding route requests, but there is still an overall $\emptyset(N)$ cost per-node.

LANMAR [3] is an interesting protocol that solves some the problems incurred by scalability. In LANMAR the problem of scalability is addressed by assuming that nodes normally move as a group. The network is grouped into logical subnets in which the members have a commonality of interests and are therefore likely to move as a group. A landmark is dynamically elected in each logical subnet and the route to the landmark is propagated throughout the network using a *distance vector* mechanism. Local routing within the group uses the proactive FSR [4] routing protocol.

In this paper we propose a novel hybrid routing approach that combines the reactive and proactive routing paradigms with node-centric and landmark routing by relying on netmarks for routing purposes. As in [1], we make the assumption of netmarks that provide special services being frequently requested throughout the network. Routes to and from the netmarks are therefore proactively maintained by all common nodes in the network. This is accomplished by letting common nodes run an advanced neighbor protocol that, in addition to maintaining links to neighboring nodes, also has the functionality of a service discovery protocol. This way, the services provided by the netmarks can be propagated throughout the network. Once a netmark has been found, nodes affiliate with the closest available netmark, and proactively maintain a route to this node.

The detailed solution is presented in Section 4.3, while Section 4.4 evaluates the performance of our approach, and Section 4.5 concludes this work.

4.2 Related Work

In [5], Xu et al propose a hierarchical extension to LANMAR by deploying a mobile backbone. This backbone is formed by introducing backbone nodes with powerful radio capabilities. This minimizes the number of hops in the network as well as lowering the performance bottleneck.

Hong et al proposes in [6] a solution to LANMARs problem of handling splitted groups. Because LANMAR is using an addressing scheme that consist of a Group ID and Host ID, a method is needed to dynamicly determine the Group ID. In this solution, a source unicast queries to landmarks to learn a destinations group

ID. It is still unclear how the performance of this scheme is affected when the group mobility assumption fails. Simulations are only made with group mobility and thus the need to do a group lookup is minimized.

[7] is yet another extension to LANMAR that enables the election of multiple landmarks in a group. This extension enable groups to be larger in size and thus minimizes the problem the original LANMAR had with isolated nodes.

ZRP [8] is another hybrid routing protocol that look upon the network in zones. Here each node maintains a zone, with a radius R. Routes to nodes within R hops is proactively maintained, while other routes is found on-demand.

4.3 Overview of Netmark Overlay Hybrid Routing

4.3.1 Netmark Overlay Routing

Netmarks announce their presence through periodic advertisements. This is an efficient way for common nodes to obtain internet connectivity from netmarks providing internet access. The affiliation process performed by common nodes can then be seen as a registration for Internet connectivity, and to become accessible from outside the local ad hoc network. Another objective of this protocol is scalability and good performance in networks with a large number of mobile nodes. An example where our approach might be applicable is a large metropolitan ad hoc network with a few special access points to Internet Services. Because mobile nodes will change their affiliation when a closer netmark becomes available, our protocol will also provide micro mobility functionalities.

The basic idea behind the routing part of the protcol, referred to as the *Netmark Overlay Routing Protocol* (NORP), is that the union of all the netmarks' routing tables covers every single node in the network. This is accomplished by not only letting the common nodes maintain routes to the netmark, but also by letting the netmark maintain routes to the common nodes. This information can then be used to locate and learn to which netmark a certain destination node is affiliated.

A virtual infrastructure is built to form an overlay network on top of the normal physical network to achieve efficient communication between netmarks. Each link in the virtual infrastructure can be viewed as a unicast tunnel in the physical network. All end to end data communications is made in the underlying physical network, while all control signaling is made in the overlay. The overlay can also be seen as an entity containing information about the approximate location of all common nodes, that is, their affiliated netmark.

In NORP, we let each netmark also perform the role of a landmark. When a node needs to send a packet to a destination for which no route is known, the node uses a simple discovery procedure to query the different landmarks. First a *location request* (LREQ) is unicasted to the affiliated netmark of the sending node. When the netmark node receives the LREQ, and the destination is unknown, the request is broadcasted in the virtual overlay to the other netmarks in the network. Once the request reaches the netmark to which the destination is affiliated, a *location*

reply (LREP) that includes the destination netmark is issued and unicasted back to the requesting node. The requesting node is now able to use the landmark routing approach, and route packets toward the netmark where the destination is affiliated. If a node is unaffiliated or no netmark is available, the protocol becomes a pure reactive protocol and routes are found on-demand.

If nodes are equipped with GPS devices, this protocol is easy to modify so that it can provide GPS coordinates through its locations services. This would enable the protocol to rely on geographical forwarding, rather than landmark forwarding.

To achieve landmark routing between different netmarks, each node in the network has a topological table, *TT*, containing every landmark in the network. The *TT* structure is a simple *distance vector* table including the destination landmarks, the next hop, the hop count and a sequence number timestamp. This table is periodically broadcasted to all one-hop neighbors, making the topological information eventually available throughout the network.

Creating a virtual infrastructure over a flat network reduces the number of nodes involved in routing, resulting in better energy consumption. Furthermore, because only routes to frequently accessed nodes are being maintained, while routes to other nodes are found on-demand, the control overhead is reduced and becomes more scalable.

The primary objectives of NORP are:

- To proactively maintain routes to frequently accessed service nodes.
- To broadcast discovery packets only when necessary.
- To disseminate information about changes in local connectivity to those neighboring mobile nodes that are likely to need the information.
- To distinguish between local connectivity management and general topology maintenance.
- To be scalable and perform well in networks with a large number of mobile hosts.

NORP exhibits some similarities with LANMAR but the main difference is that no assumption is made about group mobility. Neither does NORP rely on a specific addressing system. Because the basic routing process is completely flat, any type of addressing scheme can be used. In addition to that, NORP has been designed to optimize the performance between common nodes and netmarks, due to the special role of the latter.

4.3.2 Proactive Route Maintenance to Netmarks

Neighbor protocols for ad hoc networks are designed to exchange node information for determining which nodes are “alive” and reachable. One common type of neighbor protocols are periodic broadcast protocols. In these type of protocols, *Hello* packets are broadcasted with some, possibly variable, frequency. The

frequency may vary depending on network load, node mobility or some other criteria. *Hello* packets can be further extended to convey information about the locally known one-hop neighborhood, which allows each node to build its own two-hop neighborhood.

In NORP, a periodic neighbor protocol is used to create and maintain routes between netmark and common nodes. Hello packets contain a *netmark field* stating which netmark the advertising node is affiliated with.

Hello packets also contain the topological table, *TT*. However, the whole table is not transmitted in every update. What entries that are included depends on how far away they are from the broadcasting node; that is, they are included by using fisheye principles [4].

Netmarks, as well as common nodes, send Hello packets to inform their neighbors about their presence. When a node receives a Hello packet, it assumes the presence of a netmark in its neighborhood, and creates a route towards the netmark by indicating the sending source as the next hop. At the routing layer, if a node does not receive a Hello packet for some predefined interval of time, then the node can assume that the link to this neighbor is down. Because all common nodes in the network are announcing their affiliated netmarks, every node will also know about the path to a netmark.

By adding a netmark field to the Hello packet we achieve the goal of having routes between common nodes and netmarks. But it does not suffice that a common node have a forward path to the netmark. In order to accomplish the location search procedure described in section 4.3.1, netmarks also need reverse paths to the common nodes. In order to achieve this, nodes also include their next hop towards the netmark and a list of all the other nodes that is relying on it for forwarding packets towards the netmark. This list is called the *downstream tree*.

In Fig. 4.1, the downstream tree of *b* consists of node *c*, *d*, *e* and *f*. These nodes will therefore be included in the *neighbor field* of *b's* Hello packets.

When a node receives a Hello packet from a neighbor, it checks the *netmark field* to determine if this neighbor is affiliated with the same netmark as the node itself. If it is, it also checks the *next hop* field to determine if this neighbor relies on the node for forwarding packets towards the netmark. If this is the case, the node adds the list of neighbors indicated in the *neighbor field* to its downstream tree.

Consider Fig. 4.1 as an example on how the downstream forwarding tree is built:

1. Netmark ***a*** will start the process by sending a Hello indicating itself as a netmark.

2. When ***b*** learns of the netmark ***a***, it advertises information about this netmark in its next Hello packet.

3. ***c*** and ***d*** learns about netmark ***a*** through the Hello advertisement sent by common node ***b***. They can now start using the paths c-b-a and d-b-a respectively to reach netmark node ***a***.

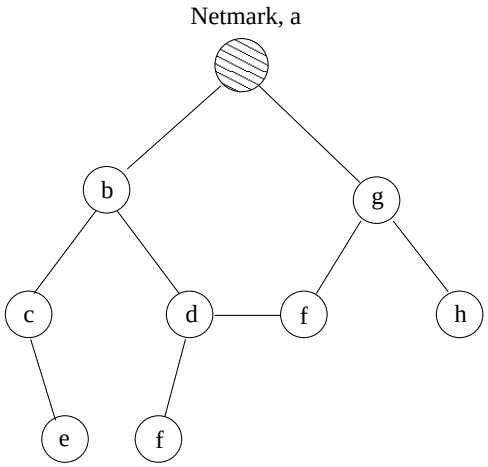


Fig. 4.1: Source Tree at Netmark

4. ***c*** and ***d*** now respectively re-advertise reachability to netmark ***a*** and that their next hop on this path is node ***b***.

5. ***e*** will learn about the path to the netmark using ***c*** as a next hop. ***e*** will be using the path e-c-b-a.

6. Similarly, ***d*** will learn about a path through ***b***, using the path d-b-a, but also about the alternate path d-f-g-a with ***f*** as a next hop. ***d*** chooses the path through ***b*** as it is of smaller length.

7. Future Hellos from node ***e*** will now indicate that it is using netmark ***a*** with ***c*** as next hop.

8. Future Hellos from node ***c*** will now indicate that it is using netmark ***a*** with ***b*** as next hop, and node ***e*** in its downstream tree.

9. After a few more updates, ***b*** announces that is uses Netmark ***a*** with ***a*** as the next hop, and that its downstream tree includes ***c***, ***d***, ***e*** and ***f***. When Netmark ***a*** receives this update, it will know about and have a route to all these nodes.

4.3.3 Link Breaks and Path Maintenance to Netmarks

Mobility of nodes not lying along an active path between a common node and a netmark does not affect the netmark routing, i.e. the size of their downstream trees are zero. These are typically the downstream leaf nodes. However, when a link break is detected by an intermediate node with an active downstream tree, a repair procedure is started. A link is deemed broken if a node has not received any Hello messages within a predefined amount of time.

The repair procedure algorithm operates in the following way:

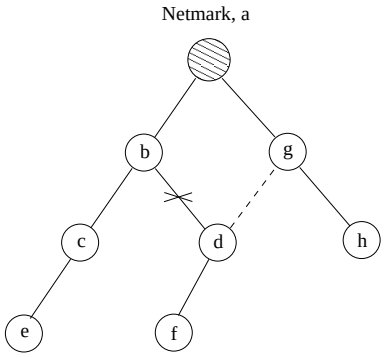


Fig. 4.2: Node d repairs path to netmark a

1. The repair procedure is initiated by letting the repairing node search its list of neighbors. Each neighbor entry in this list contains information about its next hop, the hop count and its netmark affiliation.

2. If a neighbor is found to be affiliated with the same netmark as the repairing node, this neighbor is marked as a candidate for being elected as the new next hop towards the netmark.

3. If this candidate is not using the reparing node as a next hop towards the netmark, the candidate is marked as valid. However, if this candidate is using the reparing node as its next hop, choosing that neighbor would create a loop. In that case, the candidate should be marked as invalid.

4. Once all neighbors have been searched and evaluated, the valid candidate with the smallest hop count is elected as the new next hop, see Fig. 4.2.

Fig. 4.2 illustrates the process of repairing a path between a node and a netmark:

1. Node *d* starts the repair procedure upon detecting a link break between *d* and *b*.

2. After searching its list of neighbors, *d* finds two candidates, *f* and *g*.

3. Because node *f* is using *d* as a next hop, *f* is marked as invalid.

4. Node *d* chooses *g* as the next hop towards the netmark.

In the case when a node can't find any valid next hop neighbor, the node declares the netmark unreachable and revert to the topological table, *TT* for finding a new netmark. The *TT* search procedure works in much the same way as for the neighbor table. If an entry is found during the search phase that is not pointing towards the old netmark, nor the broken next hop towards the old netmark, it is

marked as a candidate. When the whole table has been searched, the candidate with the smallest hop count is elected. The major difference between this search and the initial one is that we are searching the topological table instead of our list of neighbors.

If a new netmark has been elected, a Hello packet with the new information is broadcasted. Even though a new netmark has been elected, downstream neighbors are still using the old netmark. These downstream nodes need to be updated with the new information, otherwise they will continue sending packets upstream towards the old netmark where the link break occurred.

A neighbor upon receiving an update indicating it as next hop from a node that it self is currently using as its next hop, mark the old netmark as invalid and puts the initiating neighbor in its downstream tree. The neighbor also mark the new netmark as the current one and looks in the *TT* to find the new next hop. After this stage, the new information is immediately rebroadcasted in a new Hello packet. Eventually the new information reaches the new netmark, creating a forward and a reverse path between the netmark and the repairing node.

4.3.4 Mobility and handover between netmarks

As nodes are mobile and move around in the network, they will learn of netmarks closer in location than the one they are currently affiliated with. The next hop link towards a nodes' netmark might also break, and a new route to the current netmark can not be found. In both cases it is necessary for the node to change its affiliation to a new netmark.

A node only changes its netmark affiliation if the newer netmark is 2 or 3 hops closer (a configurable parameter) than its current one, or the next hop link breaks and a new route can not be found. If a node changes its affiliation as soon as it learns of a closer netmark, the system can become unstable. This is due to both the mobility of the nodes and the unstable nature of the wireless channel, i.e. fading, collisions etc. Under these conditions links may temporarily go down or become unavailable. If a node changes it affiliation too soon, an oscillation between the two netmarks may occur.

When a mobile node make the decision to change its affiliation, it does so by sending an update message to both the previous and the new netmark. It also immediately broadcasts a Hello message containing this new information. When the previous netmark receives the update message it creates a soft binding for this node with information about the new location of the node. Any subsequent LREQs arriving at the previous netmark is processed as normally but the transmitted LREPs will indicate the new netmark. After a few seconds the soft binding is timed out and is removed.

The update messages sent to the two netmarks also include the mobile nodes downstream tree. This is done because all the downstream neighbors will be affected by the handover, and they will automatically be affiliated with the new netmark. That is the reason why the previous Hello packet was broadcasted. When a

downstream neighbor receives the Hello message, it immediately changes its affiliation and rebroadcasts the message. All downstream nodes will in this way change their affiliation in an efficient way.

4.4 Performance Evaluation

GloMoSim [9] is the simulation platform used for evaluating the proposed approach. GloMoSim is a discrete-event, detailed simulator for wireless network systems. In our experiments, the MAC layer is implemented using the default characteristics of the distributed coordination function (DCF) of IEEE 802.11 [10].

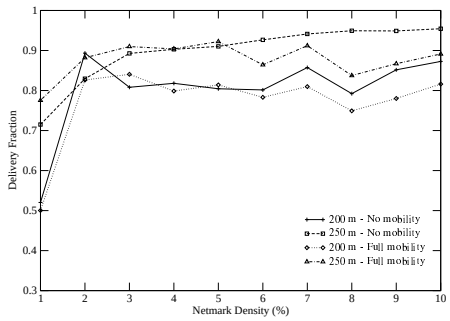
The mobility model used for the simulations is the Modified Random Direction model [11] and simulations are run for 600 simulated seconds. In this model, each node randomly selects a direction in which to travel, where a direction is measured in degrees. The node then randomly selects a speed and destination along the direction and then travels there. Once it reaches the destination, it remains stationary for some pre-defined pause time. At the end of the pause time, a new direction and speed is selected, and movement is resumed.

The following metrics are used to evaluate the performance: (i) *Packet delivery fraction* - the ratio between the number of data packets delivered to the destination and those originated by the sources. (ii) *Control overhead* - the total number of control packets transmitted by each node. Each hop-wise transmission is counted as one packet. (iii) *Normalized routing load* - the number of routing packets "transmitted" per data packet "delivered" at the destination.

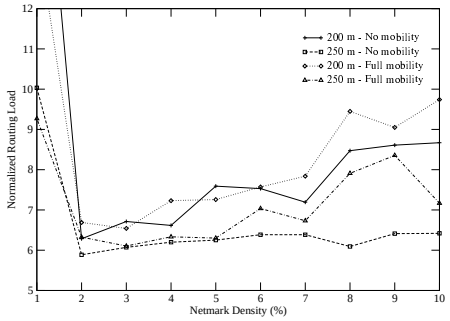
4.4.1 Netmark Traffic model

We introduce the NetMark, NM, traffic model for performance evaluation. This traffic model, which simulates the client server behaviour between the common node and a netmark, is more realistic than plain CBR traffic. It consists of sequences of FLOW_OFF and FLOW_ON periods where the OFF periods correspond to the user's think time, while the ON period represent user activity. Both the FLOW_ON and FLOW_OFF period is exponentially distributed with a mean value of 5 and 20 seconds respectively. During the FLOW_ON period, *requests* are generated with an exponentially distributed mean of 1.5 seconds. When the server, (in this case the netmark), receives a *request* it generates a *reply* with a pareto distributed mean of 3000 bytes. The pareto distribution was chosen because of its heavy tailed properties. This *reply* is then fragmented and sent back to the client.

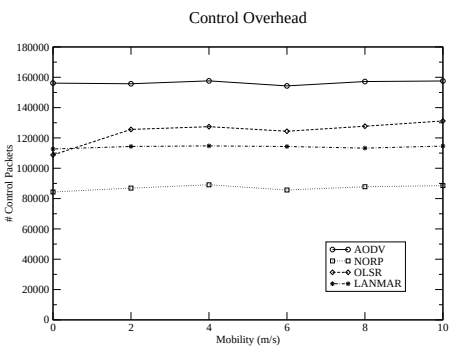
In our experiments, common nodes have continuous flows of NM traffic with the netmarks as specified above. In addition to that, 10 local short-lived CBR sources transmitting 4 packets per second are spread randomly over the network. When one session ends, a new source-destination pair is randomly selected. Thus the input traffic load is constantly maintained.



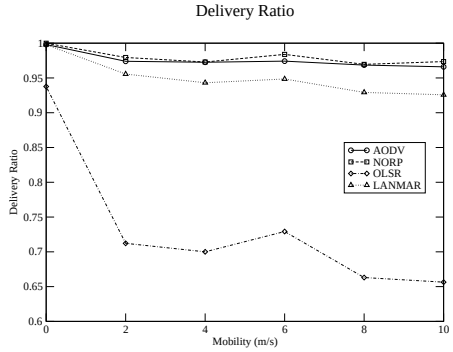
(a) Delivery Fraction with Increasing Netmark Density for two transmission ranges



(b) Normalized Routing Load with Increasing Netmark Density for two transmission ranges



(c) Number of Control Packets transmitted



(d) Delivery Ratio

4.4.2 Simulations

In these experiments, we evaluate the performance of our approach. The terrain size is set to 1000m x 1000m and the network is composed of 100 nodes. The rate of mobility is varied and the pause time is set to 10s. The transmission range is 250 meters unless stated otherwise.

Although the number of netmarks affects the performance, it is also important to comment on the deployment of the netmarks. Since the netmarks are special service providers, it is not just any common node that can be elected as a netmark. In this paper we assume that netmarks are predefined and that a netmark remains being a netmark for the whole duration of the simulation. We leave the case where a netmark can be elected among a set of nodes for future work. Therefore, the simplest method would be to preassign netmark nodes and scatter them uniformly within the terrain area at initialization time. However, this means that if the netmarks are fully mobile, some netmarks might group together in one part of the network and thus lower the performance. We have simulated two cases where netmarks are either immobile or have full mobility.

Figure 4.3(a) illustrates how the delivery fraction varies with the number of netmarks. Except for the netmark density of 1 %, the delivery is more or less the same for the specified setup. We can see that the transmission range is more

important than the mobility of the netmarks.

Figure 4.3(b) shows how the normalized routing load varies with the netmark density. The reason for an increase in overhead as the number of netmarks grows is that more control packets are being generated in the netmark overlay. When a destination needs to be found, the LREQs are sent to different netmarks. More netmarks means more LREQs.

In Figure 4.3(c) we can see that NORP can deliver packets using much less control overhead than the other protocols. Here we also see that reactive AODV generate many control packets because of the large number of nodes, but also because of the bursty nature of our traffic model.

Figure 4.3(d) illustrates how the delivery ratio varies for different protocols as the rate of mobility increases. We can see that NORP sustains a very high delivery rate for all mobility speeds. Because of the large number of nodes, the high traffic load, proactive OLSR have trouble delivering packets during mobility.

4.5 Conclusions

In this chapter, a new routing scheme is proposed, Netmark Overlay Routing Protocol (NORP). NORP proactively maintains routes to *special service providing* nodes in the network. These nodes are called netmarks. This is achieved through an extensive neighbor protocol that creates a bidirectional routing tree with the root attached to the netmark. In addition, NORP reactively searches for nodes by querying the different netmarks about the locatation of a destination node. Data packets are then routed using landmark routing towards the netmark closest to the destination node. As the data packet comes closer to the destination netmark, it will eventually arrive at node within the routing tree of destinations netmark, where it will be routed to the destination.

When netmarks provide internet connectivity the protocol also provide micro mobility functionalities. Simulations show that NORP achieves very high delivery rates in dense networks and under high traffic loads. In addition, it has been shown that NORP performs excellent under mobile conditions and has scalability properties. We have also evaluated how the performance is affected when the number of netmarks in the network is increased. To conclude, NORP is a service providing routing protocol that scales well with the size of the network.

BIBLIOGRAPHY

[1] Soumya Roy and J.J Garcia-Luna-Aceves. Node-centric hybrid routing for ad hoc wireless extensions of the internet. In *Proc. IEEE Global Telecommunications Conference (GLOBECOM), Tapei*, November 2002.

[2] P. Tsuchiya. The landmark hierarchy : A new hierarchy for routing in very large networks. In *ACM Sigcomm*, 1988.

[3] G. Pei, M. Gerla, and X. Hong. Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility. In *Proceedings of IEEE/ACM MobiHOC 2000, Boston, MA*, 2000.

[4] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *ICC (1)*, pages 70–74, 2000.

[5] Kaixin Xu, Xiaoyan Hong, and Mario Gerla. An ad hoc network with mobile backbones. In *IEEE ICC 2002, New York, NY*, 2002.

[6] Xiaoyan Hong, Nam Nguyen, Shaorong Liu, and Ying Teng. Dynamic group support in lanmar routing ad hoc networks. In *Proceedings of the Fourth IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002), Stockholm, Sweden*, 2002.

[7] Xiaoyan Hong, Mario Gerla, and Li Ma. Multiple-landmark routing for large groups in ad hoc networks. In *Proceedings of MILCOM 2002 Military Communications Conferences, Anaheim, CA*, 2002.

[8] Z. Haas. A new routing protocol for the reconfigurable wireless network. In *Proc. of the IEEE Int. Conf. on Universal Personal Communications*, october 1997.

[9] Lokesh Bajaj, Mineo Takai, Rajat Ahuja, Rajive Bagrodia, and Mario Gerla. Glomosim: A scalable network simulation environment. Technical Report 990027, 12, 1999.

[10] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification, IEEE Std 802.11-1997*. The Institute of Electrical and Electronics Engineers, New York, 1997.

[11] E. Royer, P. Melliar-Smith, and L. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *Proceedings of the IEEE International Conference on Communications, Helsinki, Finland,* 2001.

5. CHAPTER V

Micro Mobility and Internet Access Performance in Ad hoc Networks

5.1 Introduction

Today, many laptops, PDAs, handhelds, and other portable computing devices include wireless connectivity as a standard feature, and more people are carrying computers when they travel to access the Internet anytime, anywhere. More and more of these devices now use the Internet Protocol (IP) together with IEEE 802.11 [1]. In addition, broadband wireless access networks based on IEEE 802.11 are emerging, and the existing wireless technologies are moving towards an all IP infrastructure.

However, a big problem with IP is that it was never designed to support mobility management. One of the most widely known Mobility solutions for IP networks is the IP Mobility Support protocol, commonly referred to as Mobile IP [2]. With Mobile-IP, nodes are able to communicate independently of their current point of attachment to the Internet. Mobile-IP can handle both local-area and wide-area movement in both wired and wireless networks. However, mobile nodes must report their change of access point to their home networks. These location updates incur a long latency of registration processes and cause a large amount of control overhead over the Internet. The concept of IP Mobility has therefore been divided into two main categories, Macro Mobility and Micro Mobility. Macro Mobility is the management of IP nodes at a larger global scale. Once a node enters a cellular or wireless network domain the Mobility management is local to that network; the node is allowed to move within the network and be controlled locally by the micro mobility management protocol while the mobility management from a global scale remains unchanged.

In ad hoc networks, an infrastructure is not needed for the network to successfully operate, but an ad hoc network can enable the coverage area of access networks to be extended and deal with situations where it is either not possible or too expensive to deploy cell-based mobile network infrastructures. Combining Mobile-IP with ad hoc networking enables roaming between different ad-hoc networks while still being able to access the Internet. In this chapter, a solution is presented, and evaluated for TCP connections, that enable mobile nodes in an ad hoc network to have internet connectivity. Here, ad hoc networks are regarded as

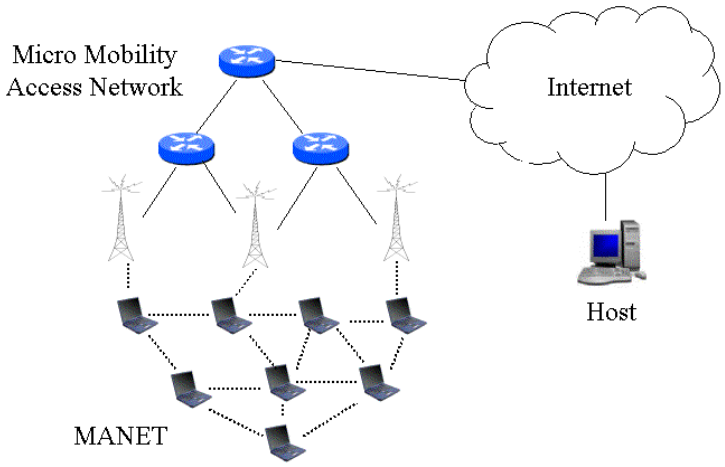


Fig. 5.1: The simulated scenario. A mobile multihop ad hoc network is connected to an access network that supports Mobile IP and micro mobility. Nodes in the wireless network are communicating with correspondent hosts on the Internet.

subnets of the Internet, that creates an integrated environment that supports both macro and micro IP mobility, see Figure 5.1.

The solution is based on Mobile IP, which enables the macro mobility capabilities. From the mobile IP perspective, foreign agents service ranges are no longer limited to hosts within a single wireless hop; the use of Manets lets mobile hosts immediately utilize available Internet services without concern about disconnection. This provide mobile nodes with the ability to form and enter ad hoc networks, while still being able to access the Internet. When several base stations or access points are available, the node is able to roam around in the ad hoc network, switching to new access points when needed. Micro mobility within the access network domain is supported by HAWAII [3], as explained below.

5.2 Related Work

In [4] a solution is presented that interconnects ad hoc networks with infrastructure networks. For micro mobility, the Cellular IPv6 [5] protocol is utilized on the edge of the Internet. AODV is used as the routing protocol within the ad hoc network. Performance is measured mainly with regards to control overhead and delivery ratio, when the mobility speed is varied.

In MIPMANET [6], the authors integrate AODV with Mobile IP. Their solution utilizes IP tunneling for separating the ad hoc network from Mobile IP. Nodes in

the ad hoc network send their packets to a correspondent node in the Internet by encapsulating the packet into another IP packet, which is destined to the Mobile IP foreign agent. A Mobile IP care-of-address is used to provide appropriate routing from the Internet to the mobile node.

In [7] a Mobile IP micro mobility architecture is presented with OLSR as the routing protocol both in the ad hoc domain as well as the access network. The work focuses on integrating Mobile IP with OLSR in order to support both micro and macro mobility. This work was later implemented and experimentally evaluated with UDP traffic in [8].

As far as we know, this is the first work that considers micro mobility ad hoc networks with respect to TCP traffic for different routing protocols.

5.3 Protocol Descriptions

5.3.1 Mobile IP

Mobile IP is a proposed standard for location independent routing. It makes mobility transparent to applications and higher level protocols like TCP and UDP. Mobile IP allows mobile nodes to have seamless access to the Internet while roaming between different networks. In order to maintain existing transport layer connections while roaming, every mobile node is assigned a home address. The home address enables the mobile node to always be able to receive data as if it was on its home network, i.e. the network to which its home address belongs.

When the mobile node is attached to a network other than its home network, it uses a care of address. The care of address is an IP address valid on the foreign network that the mobile node is visiting.

In Mobile IP, the basic mobility management procedure is composed of two parts : the movement detection performed by the mobile node and the registration to the Home Agent (HA). The home agent is a dedicated router on the mobile node's home network that forward packets through tunneling to the foreign network. This enable the mobile node to receive packets through the care of address.

- Movement detection latency : this is the time required by the mobile node to detect that it has changed its IPPOA.

- Registration latency : as the home agent can be located anywhere on the Internet, this process can take a long time and sometimes be impossible to complete. This is obviously, by far, the main expected part of the total handover latency.

In the case of a quickly moving mobile node which changes its IPPOA rapidly, the registration process will become totally inefficient. Moreover, this mechanism produces a lot of control traffic inside the local domain and across the Internet.

5.3.2 Micro mobility and HAWAII

In order to minimize the movement latencies that occur when Mobile IP is used, where the home agent is located far from a mobile nodes current location, and the mobile is moving at a high speed frequently changing its point of attachment, a micro mobility protocol is used. Micro mobility protocols uses a concept of domains, which is an area consisting of several base stations (access points) in which they cooperate. When a mobile node first connects to a domain, it obtains a care-of-address as in normal Mobile IP operation. This care of address however, remains valid for the whole duration of mobile node's stay in the same domain. The mobile node will thus make only one home registration (registration with the home agent) at the time it connects to the domain. The users movements inside the domain are then managed by the micro mobility protocol.

HAWAII [3], Handoff-Aware Wireless Access Internet Infrastructure, is a natural extension to Mobile IP to efficiently support micro-mobility in wireless networks. After the first connection of a mobile node to a domain and its home registration, the mobile node will perform local registrations only. A common approach for allowing mobility to be transparent to correspondent hosts is to divide the network into hierarchies. HAWAII uses a similar strategy, segregating the network into a hierarchy of domains, loosely modeled on the autonomous system hierarchy used in the Internet. The gateway to each domain is called the domain root router. Each mobile node is assumed to have an IP address and to have a home domain to which it belongs. While moving in its home domain, the mobile nodes retains its IP address. Packets destined to the mobile node reach the domain root router based on the subnet address of the domain and are then forwarded over special dynamically established paths to the mobile node.

When the mobile node moves into a foreign domain, HAWAII revert to traditional Mobile IP mechanisms. If the foreign domain is also based on HAWAII, the mobile node is provided with a care-of address from the foreign domain. While moving within the foreign domain, the mobile host retains its care-of address unchanged, and connectivity is maintained using dynamically established paths.

A mobile host that first powers up and attaches to a domain sends a Mobile IP registration request to the nearest base station (BS). The base station is sometimes also called the access router, as it also has routing capabilities in addition to providing fixed network access. The BS is responsible for exchanging Mobile IP messages with the mobile host's home agent, in order to register the current location of the mobile host. The base station also sends a path setup message to the domain root router, which is the gateway between the micro mobility access network and the Internet. This has the effect of establishing a host specific route for the mobile host in the domain root router. Each intermediate router on the path between the base station and the domain root router also adds a forwarding entry for the mobile node, when forwarding the path setup message. Thus, the connectivity from the domain root router to the mobile hosts connected through it forms a virtual tree overlay.

The mobile node infrequently sends periodic registration renewal messages to the base station to which it is currently attached in order to maintain the registration and the host based entries, failing which they will be removed by the base station. The base station and the intermediate routers, in turn, sends periodic aggregate hop-by-hop refresh messages towards the domain root router.

5.3.3 Other micro mobility protocols

Hierarchical Mobile IP is a natural extension to Mobile IP to efficiently support micro-mobility. After the first connection of a MN to a domain and its home registration with the address of the Gateway Foreign Agent (GFA) as Care of Address, CoA, the MN will perform Regional Registrations only. This type of registration is sent by the mobile node to the GFA each time it changes FA (i.e. of IPPOA). The registration contains the new ”local” CoA of the MN: the address that can be used by the GFA to reach the MN while it remains connected to the same FA. The routing with Hierarchical Mobile IP is then very simple. A packet destined to the MN is first intercepted by the HA and tunneled to the GFA. Then, the GFA de-capsulate and re-tunnels it towards the current local CoA of the MN.

Cellular IP aims to replace IP inside the wireless access network. A Cellular IP domain is composed of Mobile Agents (MA) and one of them acts as a gateway towards the Internet and as a Mobile IP FA for macro-mobility. Each MA maintains a routing cache that contains the next hop to join a MN (one entry per mobile) and the next hop to join the gateway. This allows the MA to forward packets from the gateway to the MN or from the MN to the gateway. The routes are established and basically maintained by the hop-by-hop transmission of two special control packets, beacon and route update. Upon receiving one of these packets the stations are triggered to update their routing cache.

The solution presented in this chapter is based upon HAWAII. It should be pointed out that the focus of this chapter is on the multi hop access portion of the ad hoc network. Basically, the results presented here should also be valid if Hierarchical Mobile IP or Cellular IP are used in the access network, as the specific operation within the access network is not the most important performance factor. If, in the future, it is determined that it would be of interest to evaluate and compare the performance of ad hoc access networks with other micro mobility protocols, it should be fairly easy to extend this study.

5.4 Mobile Ad hoc Internet Access Solution

In this solution base stations, which are also acting as Home and Foreign agents advertise their services by periodically sending *Agent Advertisement* messages.

These messages are broadcasted to the wireless ad hoc network, and their dissemination are limited by the *Time To Live* (TTL), set to an appropriate value that depends on the size of the network. Alternatively, the mobile node may broadcast an *Agent Solicitation* message, requesting mobile IP services. This is typically done by mobile nodes that are located outside of the base stations broadcast radius, the radius created by the specfied TTL value. When a base station receive an *Agent Solicitation*, it update its *network size* variable to enable the *Agent Advertisements* to propagate further and to also cover the new mobile node.

When a currently unregistered mobile node receives an advertisement, it unicasts a *Registration Request* to the advertising agent, the Base Station (BS). The BS answer this message by sending a *Registration Reply*. If this is the first registration sent by the mobile node inside this domain, HAWAII, the micro mobility protocol, sends path setup power-up messages in order to establish a routing path within the domain hierarchy toward the mobile node. The mobile node now also attains its care of address which is then registered with the home agent of the mobile node. Note that the mobile node will retain this care of address throughout its stay in the current domain.

Packets between the home agent and the mobile node are routed toward the wireless network based on the network id part of the care of address. The *domain root router* of the HAWAII domain is the root of the access network. It is also the gateway router between the local domain and the Internet, and to which the network id belongs. As the mobile node moves within the ad hoc network, from base station to base station, it will continue to be accessible from the Internet; only the local path within the lower hierarchy of the domain will be updated.

5.4.1 Internet host determination

When an on-demand routing protocol, such as AODV is used within an ad hoc network, a node cannot expect to have routes to all hosts reachable within the network. This is because routes are only set up when they are needed. The fact that we do not have a host route to a host does not necessarily mean that it is not reachable within the ad hoc network. Thus, the route discovery mechanism of the routing protocol has to search for the destination within the ad hoc network, *before* it can decide whether the destination node is located in the network or not. Because the route discovery process of AODV repeatedly searches for the destination within an increasing radius, the time it takes for AODV to determine that the destination is unreachable is quite significant. This problem has been solved in our solution by letting the base stations send proxy route replies.

When a base station receives a route request from one of its registered nodes, it searches its registration list, (also called visitor list within the Mobile IP terminology), for a match with the requested destination. If a match is found, a normal route reply is generated. If a match is not found, a special proxy route reply indicated by an 'I' flag is generated. This proxy route reply will also establish a route path between the requesting node and the requested destination. It is therefore important

that the base station receiving the request also check that the requesting node is registered. If this is not done, and another base station than the one the requesting mobile node is currently registered with answer the request, an asymmetric route will appear. An asymmetric route will, among other things, render transport layer enhancements such as Snoop [9] useless.

Another thing that needs consideration when using proxy route replies is what sequence number to use in the reply. In a normal route reply, a destination indicates its own sequence number in the reply. If the node processing the request is not the destination, it specifies its last known sequence number to the destination. In our case however, the base station processing the request do not know the destination sequence number, because the destination is located in the Internet. If an unknown sequence number is used as is normally done in route requests, it will be hard to keep the routes fresh. In this solution, proxy route replies use the sequence number indicated in the request, plus one. The replying base station then remembers the sequence number used, and any subsequent replies for this destination will now indicate this number plus one, or if the sequence number in the request is higher, this number plus one. The 'I' flag still indicates that it is an Internet route, and that a normal direct route should be preferred.

5.4.1.1 OLSR operation

If a proactive protocol such as OLSR is used for routing in the ad hoc network, things are a lot easier. If a node does not have a routing table entry for a specific destination, the destination is normally not located inside the network. When a mobile node wishing to transmit a packet fails looking up a destination in the routing table, it tunnels the packet to the base station to which it is currently registered.

A base station receiving a tunneled, IP within IP encapsulated packet, untunnels the packet and forward the packet using normal IP routing mechanisms.

A note to consider when configuring an OLSR base station is to ensure that no routes from any wired interfaces are announced in the OLSR update messages being transmitted on the wireless interface.

5.4.2 Handover

As a mobile node moves inside the ad hoc network it will eventually come into communication range of new and closer base stations. The mobile node may also move out of the communication range of its current base station. The question then arises when it is time to switch to a new base station and perform an handover.

Since the mobile node is moving inside a multi hop network, a natural criteria for performing a handover would be to perform a handover as soon as the mobile node learns of a closer base station. This might be a good criteria but it do have a few drawbacks. When the mobile node learns of an other base station with the same distance as the one it is currently registered with, it will continue to use its current one. It will continue to use its current base station until the registration

times out, or the route toward the base station breaks. Both of these cases are bad from a performance perspective, and will lead to throughput degradation and in the absolutely worst case, loss of its active connections. A mobile node will therefore perform handover as soon as it learns of a new base station that is closer or as close as the one it is currently registered with. In order to avoid registration oscillation, the node remembers its previous base station and only perform a new handover to the previous base station if: 1) the distance becomes lower than the distance to the current base station 2) the registration with the current base station times out 3) the route towards the current base station times out.

When a mobile node determine that a handover needs to be performed, the handover procedure is initiated. This is done by sending a *Registration Request* to the new base BS that includes information about the previous BS. When the new BS receives this message it replies by sending a *Registration Reply* as normal. The HAWAII micro mobility protocol at the new BS now also sends path update messages to the local micro mobility domain and a handover notification is sent to the old BS. The old BS thus removes the mobile node from its registration list and updates its routing table accordingly.

The decision to perform handover is always made upon information received in *Agent Advertisements*. When the handover is initiated, and the *Registration Request* is sent, the mobile node updates its *Pending Registration* flag. When this flag is set, the mobile node can not send any *Registration Requests*. This is because during the time interval when the registration is pending, it is possible for the requesting node to receive new *Agent Advertisements* from other base stations or other neighboring nodes forwarding the same advertisement. The *Pending Registration* therefore have the dual purpose of preventing unnecessary registrations, but also to prevent the mobile node from registering to two different base stations within the request reply time interval. When a *Registration Reply* is received, the handover procedure is considered completed and the mobile node now updates its registration information about its current base station, previous base station, registration lifetime, distance to the new base station and reset the *Pending Registration* flag to false.

When the *Registration Request* is first sent, a timer is also started that will check whether a reply was successfully received. If a reply has not been received when the timer expires, the mobile node may either send a new *Registration Request* or decide that the base station is unreachable and wait for a new advertisement.

If a proactive routing protocol such as OLSR is used, a handover can only be performed if the mobile node have a valid routing table entry towards the new base station. Because *Agent Advertisements* are broadcasted, not unicasted, it is possible for a mobile node to receive information about a base station before a route has been completely setup. In order to avoid setting unnecessary timers and *Pending Registration* flags, the mobile node checks whether a valid route to the base station exists, *before* sending a *Registration Request*.

Once the handover and registration procedure has been successfully completed,

the mobile node is reachable from the internet through the registration in the new base station. If a reactive protocol such as AODV is used, any active routes within the ad hoc network toward Internet hosts will still point towards the old base station. AODV therefore generate new route requests as described above. In order to prevent intermediate nodes between the mobile node and the base station from replying to the request, the 'D' flag of the AODV route request message is specified. The 'D' flag specifies that only the destination node may reply to the request, assuring that the message will propagate all the way to the new base station so that a new route reply and route can be established.

5.5 Distance Update Procedures

An important metric in the handover determination procedure is the distance in hops between the mobile node and its current base station. Some considerations have to be taken as to when and how this distance is determined. A mobile node have the capability of determining the number of hops an *Agent Advertisement* has traversed when it is received by looking at a new *distance* field. This field is currently taken from the reserved bits of the Mobile IP *Mobility Agent Advertisement Extension* header.

A naive method of determining the distance to a base station would be to simply look at the distance field and update the mobile nodes registration information with this value. But as a mobile node can receive *Agent Advertisements* from many different neighbors reporting different distances, this would cause the distance value to oscillate, and would not reflect the shortest path available. This could also lead to handover oscillations, because the mobile node may determine that some other base station is closer and perform handover to it, and later receive another advertisement causing it to switch back to the original base station because that is now again the closest.

When a mobile node receive an advertisement about is current base station, it updates the its distance information only if any of the following conditions are valid:

- the advertisement was received from the next hop neighbor on the route toward the mobile nodes current base station *or*
- the advertisement was received from the current base station *or*
- the advertised distance is closer *or*
- the route towards the current base station is broken *or*
- the route towards the current base station broke down within the last advertisment period

If the advertised distance is closer and AODV is used as the routing protocol, the mobile node checks if the advertisement message received was from its next

hop neighbor toward the base station. If the check fails, a new route request is issued for the base station, and for any active destinations the mobile node might be communicating with on the Internet. This is because although the received advertisement indicates a better route toward the base station, the new route toward the base station might not yet be updated. If this procedure is not followed the mobile node might continue using the older and longer route, or the bidirectional route might be become asymmetric. If a proactive protocol such as OLSR is used, routes are updated periodically and the new route should be found automatically.

5.5.1 Link Breaks

When a link break is detected by the routing protocol, a check is performed that determine whether this was the link the mobile node was currently using to reach the base station. If it was, the mobile node will take this into account when the next *Agent Advertisement* is received. If the routing protocol hasn't reestablished the route, and the advertisement received indicates a new base station, a handover will be performed. If the route has successfully be reestablished, the distance information is updated.

5.6 Performance Simulations

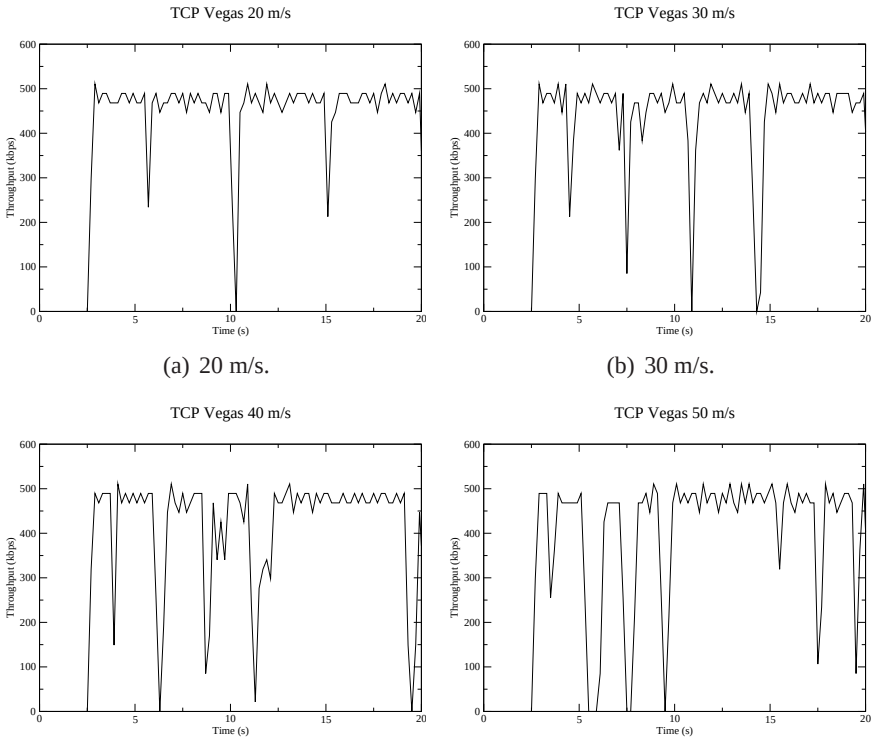
This paper aims to investigate the performance of micro mobility movement in a hybrid ad hoc network as described above.

The presented solution have been evaluated in two popular network simulators, NS-2 [10], and GloMoSim [11].

Glomosim is a discrete-event, detailed simulator for wireless network systems. It is based on the C-based parallel simulation language PARSEC [12]. In our glomosim experiments, the MAC layer is implemented using the default characteristics of the distributed coordination function (DCF) of IEEE 802.11b [1] with a data rate of 11Mbps. The data transmission range is approximately 250m.

The NS-2 simulator is a discrete event simulator widely used in the networking research community. It was developed at the University of California at Berkeley and extended at Carnegie Mellon University to simulate wireless networks. These extensions provide a detailed model of the physical and link layer behavior of a wireless network and allow arbitrary movement of nodes within the network. In our NS-2 experiments, the MAC layer is implemented using the default characteristics of the distributed coordination function (DCF) of IEEE 802.11 [1]. The data rate for the simulations is 2 Mbits/sec.

The simulations conducted aim to analyze the performance of TCP flows during internet connectivity and during handover. The current Internet host implementations contain a variety of TCP flavours. In order to investigate the differences and effect on micro mobility between these, various TCP versions have been selected



(a) 20 m/s. (b) 30 m/s.

(c) 40 m/s. (d) 50 m/s.

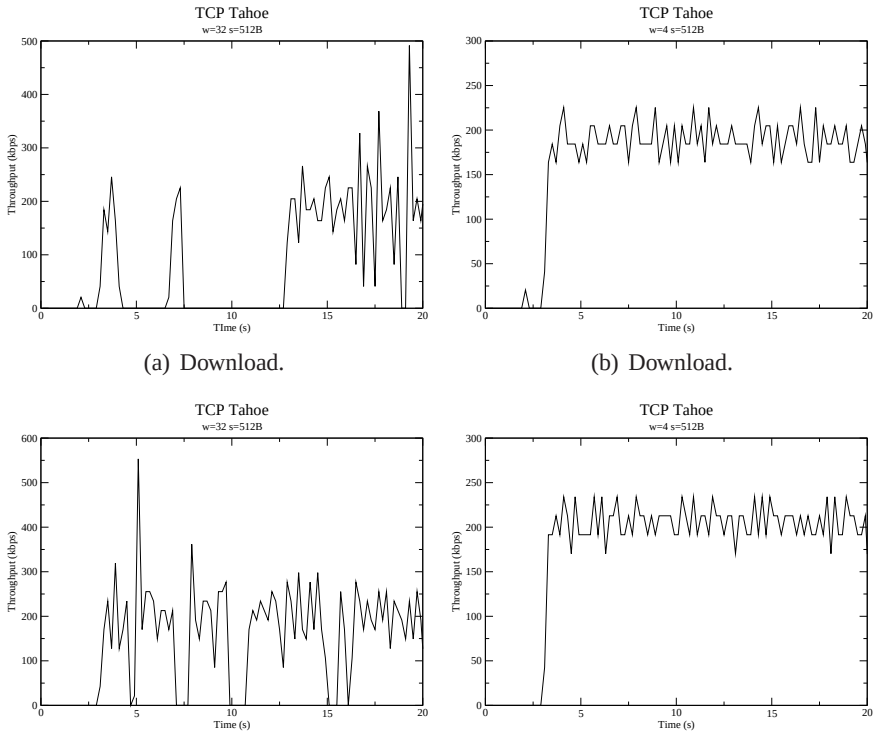
Fig. 5.2: TCP throughput in kbps for different mobility speeds

and analyzed. These are TAHOE, NEWRENO, VEGAS and ATCP [13]. The impact of mobility and ad hoc routing protocols and the relation between the two during handover have a big impact on the performance. It is also so that the different versions of TCP behave differently in this mobile environment.

5.6.1 NS-2 simulations

The NS-2 simulated scenario mainly consist of two parts, an access network consisting of base stations connected by wired links, and a wireless ad hoc network, see Figure 5.1. The access network is also the micro mobility domain and consists of 4 base stations connected in a three-level network hierarchy of in total 6 routers, excluding the base stations. Connected to the top domain router is a correspondent wired host that will be communicating with nodes in the wireless ad hoc network.

Figure 5.2 shows the throughput of a TCP Vegas connection between the correspondent host in the wired domain and a mobile node in the ad hoc network. The mobile node moves in parallel with the base stations at different mobility speeds, and as it learns of new and closer base stations, performs handovers. The hop distance between the mobile node and its associated base station varies between two and three, depending on the connectivity of the network. The hop distance is never



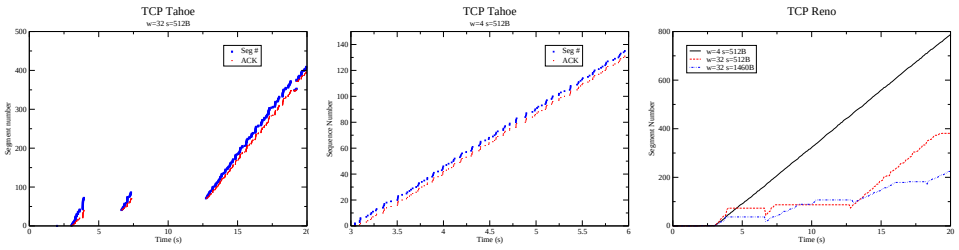
(a) Download. (b) Download.

(c) Upload. (d) Upload.

Fig. 5.3: TCP throughput in kbps for a static upload and download scenario

shorter than two, because the distance between mobile node and the base station is such that, at least one intermediate node is needed for connectivity. The periodicity of base station advertisements is 1 second.

In Figure 5.2(a) we see the throughput when the node is moving at 20 m/s. The node is able to sustain a fairly high throughput, but it drops for a short duration during handovers. One reason for this can be found in the way mobile nodes decides when it is time to perform a handover. When a node learns of a new and closer BS, it switches to it. The mobile node also switches to a new BS if the registration of the old one timed out, and there is a certain latency before a new connection can be established. Another reason is that the next hop link towards the BS in the ad hoc network breaks, and the route repair mechanism of AODV is invoked. This is typically detected through a packet drop. If it was the next hop towards the base station that was broken, a check is performed to see if a handover is needed. The link break and packet drop in combination with TCP's window behaviour may cause the throughput to momentarily drop to a lower level. Figure 5.2(b), 5.2(c) and 5.2(d) shows the same scenario for speeds at 30 ,40 and 50 m/s respectively. We can see here that as the mobility speed increases, the dips frequently becomes wider and deeper. At 50 m/s it takes about 1 second for the connection to regain its throughput, but only for a short time before a new handover takes place. It should



(a) Tahoe window size w=32 (b) Tahoe window size w=4 (c) Reno segment numbers

Fig. 5.4: TCP segment number and acks for two window sizes

be noted that 50 m/s is a very high speed, it corresponds to 180 km/h or 112 mph.

The fact that the wireless environment itself is unreliable has a big impact on the performance. This can be observed in Figure 5.3, that illustrates a static scenario between the mobile node and the correspondent host. The distance between the mobile node and its base station is 5 wireless hops. As can be seen from the figures the throughput is fairly poor, and is significantly lower than the ones seen in Figure 5.2, even though those figures refer to a mobile scenario. One of the reasons for this is the exposed node problem [14], which 802.11 doesn't address. This problem basically means that a node is prevented from transmitting when it is either: within the range of a sender but not the receiver, or within the range of the receiver but not the sender. Another factor is that these nodes operate under half duplex, where a node has to wait for it own packet to be transmitted by the next hop, before it can transmit the next packet in sequence. The result here is that the throughput is lowered for each additional hop, see below. Another problem is the window behaviour of TCP. The different figures in Figure 5.3 all show the behaviour of TCP Tahoe, a fairly aggressive flavour of TCP. When Tahoe is used with a large maximum window size, TCP will start transmitting packets with an exponential increase in the window size for each received acknowledgement. This means that the sender transmitts packets in fast order, causing collisions and interference to intermediate wireless nodes, with the result of packets being dropped. TCP will therefore timeout, the window lowered and the packets retransmitted, again in fast order. The same thing will happen again, causing the throughput to oscillate, as seen in Figure 5.3(c).

We can also see a distinct difference in performance between the download and upload scenario in Figure 5.3(a) vs 5.3(c). The is because our network is heterogeneous, and the wired sender has a different sending behaviour than the wireless one. The wired sender will rapidly start sending packets, the wired link have a higher bandwidth, and when they reach the base station buffering will take place. Because of the lower bandwidth and the unreliable nature of the wireless channel, packets will be dropped. This can observed in Figure 5.4(a) that show TCP segment numbers and acks. The throughput of this scenario is the one seen in Figure 5.3(a). Because of the lower bandwidth at the wireless sender in the upload sce-

nario, the same amount of packet drops doesn't take place, see the corresponding throughput in Figure 5.3(c).

A way to cope with this problem, and to make the transmission process less aggressive, is to lower the maximum congestion window size. When the windows size is lowered from 32 to 4 segments, the difference between the upload and download throughput disappears, see fig 5.3(b) and 5.3(d).

Yet another factor that impact the performance is the size of the packets. Figure 5.4(c) show the increase of the segments number for a downlink TCP Reno connection. The fastest segments number increase in this figure is those with a packet size of 512 bytes. The slowest increase is achieved when the packet size is 1460 bytes. The reason for this is quite simple; the longer the transmission of packet takes on the wireless channel, the higher the probability for interference to cause an unsuccessful reception.

Throughput (kbps)	*Upload*	*Download*
Vegas 20m/s	429.3	391.4
Vegas 30m/s	424.0	386.0
Tahoe 20m/s	442.0	382.8
Tahoe 30m/s	439.8	374.8
Reno 20m/s	423.5	370.1
Reno 30m/s	421.9	346.6

Tab. 5.1: Mean Throughput for various TCP flavors during upload and download

Table 5.1 shows the corresponding throughput for various flavors during upload and download. We can see here that Tahoe achieves the highest throughput during upload for both 20 and 30 m/s mobility. This is because Tahoe accesses the wireless channel more aggressively than Vegas, as was explained above. However, this aggressiveness is less advantageous in the download case where Vegas achieves the highest throughput. It should be noted that no congestion in the normal sense occur in this scenario, which is the reason why Reno perform worse than Tahoe.

Another issue with TCP flows in ad hoc networks is unfairness. This can observed in Figure 5.5. Here we can see that the ongoing TCP download connection is completely shut down by a short lived local connection. When the local flow terminates, the previous connection can be resumed, but only until another local flow starts. The reason for this is a complicated interaction between the 802.11 MAC layer and TCP that forces the MAC layer into exponential backoff. This is a problem that has been discussed before [15], but for TCP Reno, so this situation is still somewhat different. In our case, the old flow interprets the increased contention as congestion and lowers the congestion window, which enables the new flow to more easily grab the channel. When a 802.11 flow fails to grab the channel, it is left in exponential backoff. The Internet TCP flow in this example has a higher delay, which translates into a higher bandwidth delay product, which basically will

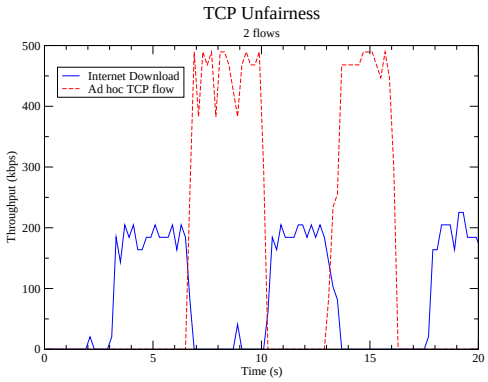
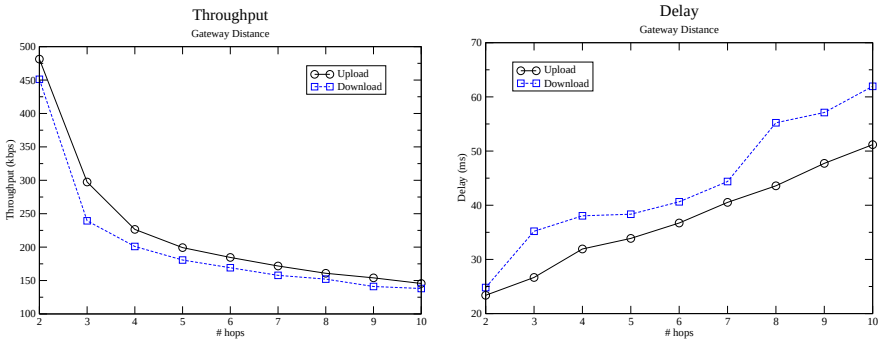


Fig. 5.5: TCP Unfairness during a static download scenario with a competing local flow inside the ad hoc network. TCP Vegas.

translate into a larger congestion window. While Vegas is more sensitive and can proact to congestion, it still operates with a slow start procedure similar to that of Tahoe/Reno. This enables our local flow to initially push harder and grab the channel, more often leaving the other flow in exponential backoff. A similar thing also happens with Reno and Tahoe flows, except that the congestion window will always increase for every acknowledgement. This problem has been addressed in [15], where an extra delay is scheduled before a packet is given to the MAC layer. The solution operates on principles similar to those used in the Vegas congestion detection algorithm. The duration of extra the delay is calculated by observing the difference between the queue output rate, and the ideal output rate based upon the physical layer transmission rate, if no contention is experienced. The higher this difference is, the higher the delay. This enables lower throughput flows to more easily grab the channel, and be less likely to become stuck in exponential backoff. This proves to be very effective and fairness is greatly improved. As that work was conducted upon the observation of TCP NewReno flows, the question is how effective this solution is for TCP Vegas. Will the Vegas congestion detection mechanism and the MAC delay mechanism work in combination, with the exponential backoff? Another question is how the delay should be calculated when the data transmission rate is dynamically adjusted. This is left as future but interesting work.

During the course of our investigation, we also observed that the throughput and delay clearly depends on the distance between a mobile node and its corresponding base station. As the number of hops increases, the throughput decreases while the delay increases, see Figure 5.6. We can see here that the throughput during upload is around 475 kbps when the distance is 2 hops, but only around 150 kbps for 10 hops. The decrease is faster in the beginning and seems to be exponentially declining. The main reason for this is probably the exposed node problem, and that nodes are prevented from transmitting because the next to next hop node is transmitting. The upload throughput is also always higher than during download,



(a) Throughput for TCP Vegas (b) Delay for TCP Vegas

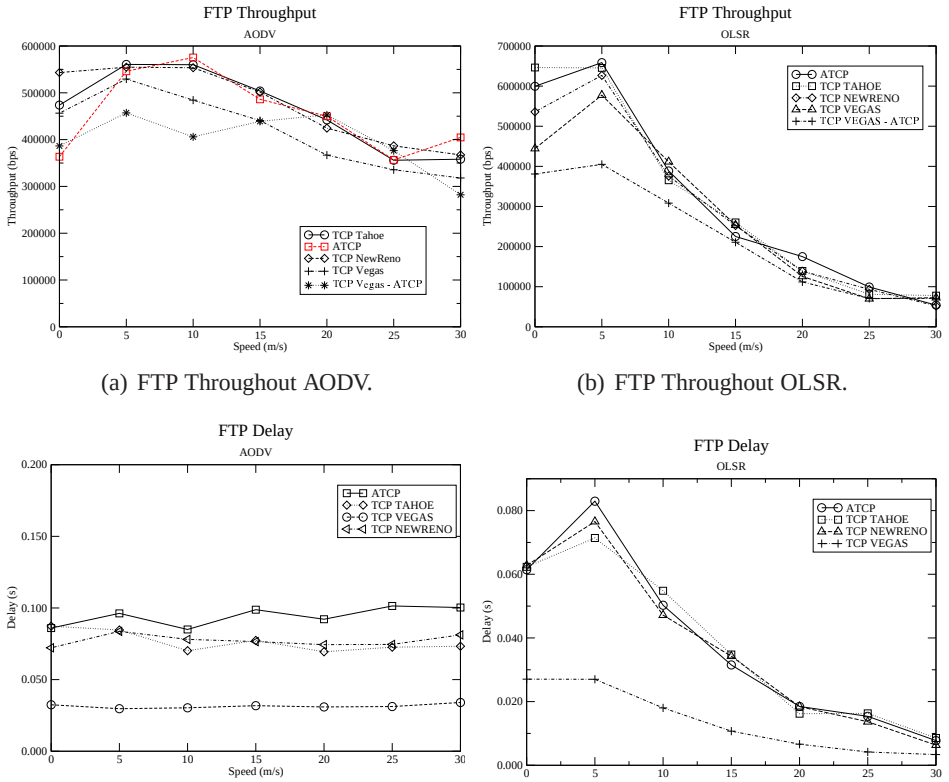
Fig. 5.6: TCP throughput and delay for different gateway distances, for the static scenario

as has been discussed above. The increase in delay seems to be almost linear with the number of hops, at least during upload. For 2 hops, the delay is around 25ms, but for 10 hops it has been doubled to around 50-60ms. As each additional hop introduces additional processing time, this makes perfect sense.

5.6.2 Glomosim simulations

The glomosim simulated scenario mainly consist of two parts, an access network consisting of base stations connected by wired links, and a wireless ad hoc network. The access network is also the micro mobility domain and consists of 6 base stations, 300 meters apart connected in a three-level network hierarchy with in total 10 wired routers. Connected to the top domain router is a correspondent wired host with an emulated long delay WAN link, that will be communicating with nodes in the wireless ad hoc network. The number of wireless nodes in the ad hoc network is 30.

The first set of simulations show the performance of FTP downloads at different mobility speeds and for different TCP flavours, see Figure 5.7. In this scenario a mobile node is moving along a straight line, such as freeway or some other open road, in parallel with base stations. The mobile node frequently performs handover while the TCP connection is maintained active. The distance between the mobile node and the base station is minimum 2 hops, but depend on the present network topology and radio conditions. AODV and OLSR is used for routing in the ad hoc network. Figure 5.7(a) show the FTP throughput for AODV and Figure 5.7(b) show the throughput for OLSR. As can be expected, the throughput generally becomes lower with increasing mobility. The highest throughput is achieved by ATCP for both AODV and OLSR at speeds around 5-10 m/s. The difference between the various TCP flavours is small but TCP Vegas always achives the lowest throughput. The reason for this is that Vegas is very good at performing congestion control, but the main factor affecting TCPs performance is actually link breaks. That link breaks is the main TCP performance bottleneck can be observed by studying



(a) FTP Throughout AODV. (b) FTP Throughout OLSR.

(c) FTP Delay AODV. (d) FTP Delay OLSR.

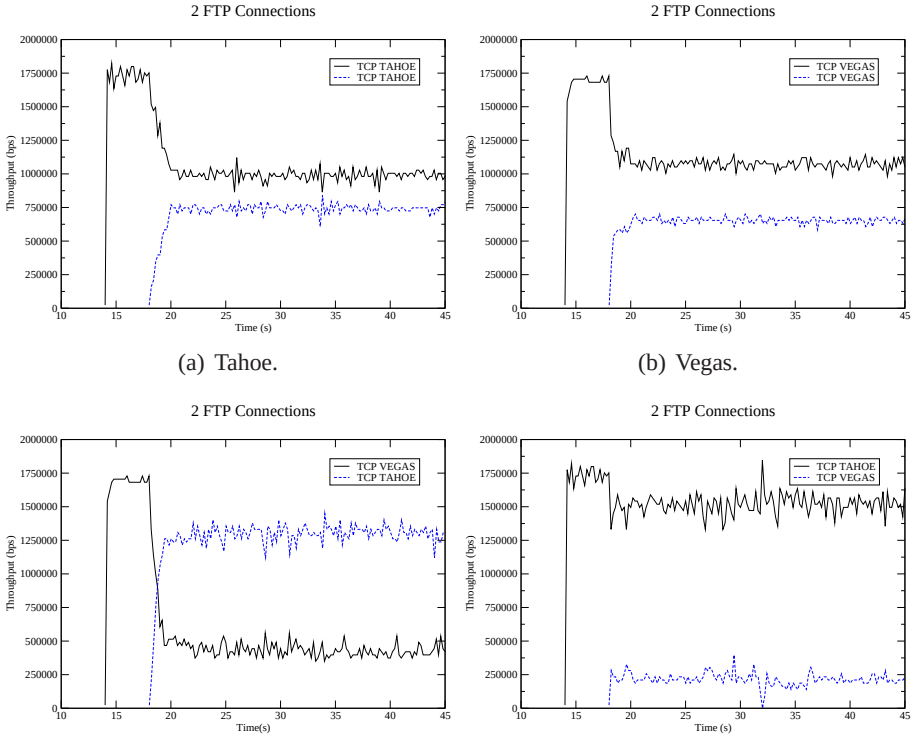
Fig. 5.7: FTP Download at different speeds

the difference between AODV and OLSR in Figure 5.7(a) and 5.7(b). The TCP throughput is declining faster for OLSR than for AODV. The reason for this is that, when a route breaks, AODV will try to repair it, while OLSR will have to wait for a new route to be established. It is also illustrated by the fact that ATCPs ability to freeze the transmission window and go into persist mode doesn't have much of an effect, because link breaks have a greater impact on performance than actual packet loss. Although the throughput is declining faster for OLSR with increased mobility, OLSR produces a higher throughput than AODV for lower mobility speeds. For very high mobility speeds AODV performs better.

Figure 5.7(c) and 5.7(d) show the delay for the same set of simulations. We can see here that ATCP always have the highest delay, while TCP Vegas have the lowest. An interesting observation here is that the declining trend of the delay figures of OLSR look very similar to those of the throughput for OLSR. OLSR therefore seems to deliver more packets on slower routes for the higher speeds, and less packets on faster router for slower speeds, but the throughput is still becoming lower as more link breaks occur. Less packets being delivered translates into packets being delivered faster. The delay for AODV on the other hand, seems to be more or less independent upon the mobility. The route repair feature of AODV is

effective enough and manages to quickly repair a route that breaks. However, for ATCP, Tahoe and NewReno, the delay is always higher for AODV than for OLSR.

We can therefore make the following conclusion as to the choice of the routing protocol. If the mobility is low, OLSR is to be preferred as it achieves a higher throughput and lower delay for most of the TCP flavours. For higher mobility speeds, AODV would be the better choice. There is an extension to OLSR called Fast-OLSR [16] that dynamically tunes the *Hello* frequency to the perceived mobility rate. With this extension in place, OLSR might a choice also for the higher mobility rates. This could be evaluated in future studies. Similarily AODV may use proactive link management strategies by monitoring the signal strength to active next hop neighbors, in order to repair the route before the link breaks.



(a) Tahoe. (b) Vegas.

(c) Vegas-Tahoe. (d) Tahoe-Vegas.

Fig. 5.8: TCP throughput of two competing FTP connections

TCP Vegas capability of producing connections with lower delay needs a few comments. The first is that although TCP Vegas produces connections with lower throughput, it is better suited for delay sensitive applications. The second is that the smoother throughput and lower delay of Vegas suffer a new kind of unfairness, see Figure 5.8. This unfairness is different than the unfairness described above in section 5.6.1. Here, the unfairness is caused by differences in the flavour of TCP. The figure show the throughput of two competing FTP connections for different Tahoe Vegas combinations. It simulates what happens when a new concurrent

FTP download connection starts at a neighboring node 5 seconds after the first one. The ideal bandwidth delay product of the two flows are the same. When Tahoe is used together with Tahoe, or Vegas together with Vegas, they are able to share the available capacity fairly well, with the original connection receiving the higher throughput. However, when a new Tahoe connection is started after a Vegas connection, see Figure 5.8(c), Tahoe lowers the throughput of the Vegas connection down to a much lower value. The reason for this is the lower delay and smoother congestion window behaviour achieved by Vegas. The result is that Vegas will have a lower transmission window with lowered throughput. When two connections such as FTP are competing and trying to transmit packets as fast as possible a few issues arise at the MAC layer. For example, contention for a busy channel causes the contending node to exponentially backoff, leading to well known unfair behaviours. The increased delay caused by the contention is interpreted by Vegas as congestion. What is interesting in this special scenario, is that (data) packet drops didn't actually occur at either the wireless medium nor in any of the buffers. The unfair behaviour is a result purely from the different delay characteristics causing TCP Vegas to lower its transmission window. Further simulations conducted show that by using smaller buffers, and thus force congestion to occur, actually lowers the unfairness and increases the throughput of TCP Vegas, due to Tahoes reaction to the congestion. This is something that should be studied more in future work.

5.7 Conclusion

This chapter have presented a solution that provide Internet connectivity and micro mobility to ad hoc networks. The solution relies on the AODV or OLSR routing protocols for establishing multihop paths between a mobile node and a base station. For micro mobility, the solution is based on HAWAII, a domain based micro mobility scheme.

The transport layer performance of the proposed solution has been evaluated using simulations. The simulations indicate that a fairly high throughput can be achieved, even during very high mobility speeds. However, the characteristics of the wireless environment itself, as well as inefficiencies of the 802.11 MAC layer protocol, lowers the performance when the number of hops increases. By using a less aggressive version of TCP such as Vegas, or lowering the maximum window size, the throughput can be somewhat increased.

TCP Vegas produces connections with lower delays due both to its ability to avoid congestion and overflow as well as it being more resilient to random packet loss.

Simulations also show that the main factor of concern to the throughput of TCP connections are link breaks, rather than flavour and window behaviour.

If the mobility is low, OLSR is to be the preferred routing protocol as it achieves a higher throughput and lower delay for most of the TCP flavours. For higher mobility speeds, AODV would be the better choice.

The unfairness problem needs to be considered when multiple TCP flows are to be supported.

Different aspects and considerations have been discussed that have importance when trying to implement a micro mobility ad hoc network. This include criterias for determining when handover to a new base station needs to be performed, and how these criteria should be maintained and updated.

BIBLIOGRAPHY

[1] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification, IEEE Std 802.11-1997*. The Institute of Electrical and Electronics Engineers, New York, 1997.

[2] C. Perkins. Ip mobility support. IETF RFC 3344, August 2002.

[3] R. Ramjee, T. La Porta, S. Thuei, K. Varadhan, and S.Y. Wang. Hawaii: A domain-based approach for supporting mobility in wide-area wireless networks. In *Proceedings IEEE Intl Conference on Network Protocols, Toronto, Canada*, 1999.

[4] V. Typpo. Micro-mobility within wireless ad hoc networks: Towards hybrid wireless multihop networks. 2001, http://citeseer.nj.nec.com/488851.html.

[5] Z. Shelby, D. Gatzounas, A. Campbell, and C-Y. Wan. Cellular ipv6. In *IETF Internet Draft (expired), draft-shelby-cellularipv6-01*, July 2001.

[6] U. Jonsson, F. Alriksson, T. Larsson, P. Johansson, and G. Maguire Jr. Mipmanet - mobile ip for mobile ad hoc networks. In *Proceedings IEEE/ACM Workshop on Mobile and Ad Hoc Networking and Computing, Boston, MA, USA*, August 1999.

[7] M. Benzaid, P. Minet, and K. Al Agha. A framework for integrating mobile-ip and olsr ad-hoc networking for future wireless mobile systems. In *1st Mediterranean Ad-Hoc Networks workshop (MedHoc 2002, Sardegna Italy*, 2002.

[8] M. Benzaid, P. Minet, and K.A. Agha. Performance evaluation of the implementation integrating mobile-ip and olsr in full-ip networks. In *Proceedings of IEEE Wireless Communications and Networking Conference WCNC, Atlanta, Georgia, USA*, 2004.

[9] H. Balakrishnan, S. Seshan, and R. H. Katz. Improving reliable transport and handoff performance in cellular wireless networks. *Wireless Networks*, 1(4):469–481, February 1995.

[10] UCB/LBNL/VINT. Network simulator - (version 2). 1999, http://www.isi.edu/nsnam/ns.

[11] Lokesh Bajaj, Mineo Takai, Rajat Ahuja, Rajive Bagrodia, and Mario Gerla. Glomosim: A scalable network simulation environment. Technical Report 990027, 12, 1999.

[12] R. Bagrodia and R. Meyer. Parsec: A parallel simulation environment for complex system. Technical report, 1998.

[13] J. Liu and S. Singh. Atcp: Tcp for mobile ad hoc networks. *IEEE JSAC*, 19(7):1300–1315, July 2001.

[14] A. Velayutham and H. Wang. Solution to the exposed node problem of ieee 802.11 wireless ad-hoc networks. 2003, http://www.cs.iastate.edu/ vel/research/E-MAC.pdf.

[15] L. Yang, W. Seah, and Q. Yin. Improving fairness among tcp flows crossing wireless ad hoc and wired networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing, Annapolis, MD, USA*, 2003.

[16] M. Benzaid, P. Minet, and K. Al Agha. Integrating fast mobility in the olsr routing protocol. In *Proceedings of the Fourth IEEE Conference on Mobile and Wireless Communications Networks (MWCN), Stockholm Sweden*, September 2002.

6. CHAPTER VI

Diversity forwarding in Ad hoc and Mesh Networks

6.1 Introduction

In wireless networks, Medium access control (MAC) protocols decide what device should be allowed access to the physical medium, at a given time. In a wireless environment, if two nodes transmit at the same time, they will cause interference which may result in loss of data. A common solution to this problem is to only allow a single node to transmit at the same time, thus enabling successful transmissions and preventing collisions from occurring. The most popular wireless MAC layer protocol used today is the IEEE 802.11 DCF [1] [2] that is being used in almost in every laptop computer as a wireless LAN technology. IEEE 802.11 can be used in both infrastructure mode to gain access to the Internet, or in ad hoc mode for easy communication between peer nodes without the need for an access point.

Recently there has been work that considers the current interference situation when setting up access to the wireless medium [3].If we know the current interference situation at our intended destination, we can make more intelligent decisions on whether we should transmit or not. If we also know the gain or path loss between the transmitter and the receiver it will be possible to perform power control; we could set the power level to a value such that a certain SINR (Signal to Interference and Noise Ratio) threshold is achieved. What would be even better is if we could coordinate the different transmitters and their power levels. If they transmit at the same time and at the appropriate power level, throughput could be improved and interference can be controlled.

Another popular research topic is that of multi path routing. If we setup multiple paths between a source and a destination, it is easy to switch to a new path if the old path breaks. This will also enable the possibility for load balancing between different routes, and to distribute the load in the network. A special type of multi path routing is non-disjoint multi path routing. In this type of routing, every source and intermediate node on the path towards the destination has one or more next hop candidate nodes. This is in contrast to node-disjoint routes where a source has multiple paths to a destination, but no paths share any nodes. In the same way, link-disjoint routes don't share any links. By having a non-disjoint routing scheme we can let each forwarding node make a forwarding decision based on the best current channel conditions. If the signal strength on a link to one next hop neighbor is in a current bad state due to fading, it may be possible to choose another next hop,

that is currently in a better fading situation. The different next hops are therefore queried before a decision is made about which next hop to choose. This is called diversity forwarding.

6.2 related work

To my knowledge, the first work about diversity forwarding is the Selection Diversity Forwarding (SDF) scheme, presented in [4]. Here a node first multicasts a data packet to a set of candidate nodes, and then the forwarding decision is made based upon responses from the candidates. A similar and sort of reverse idea was later developed in [5] [6], where a small probe, or RTS packet is multicasted to a set of receivers, and the candidate that responds first is chosen as the next hop. Similary, [7] [8] first transmits a probe, but they wait for *all* receivers to respond, before choosing the candidate with the best current radio conditions.

When a diversity protocol queries the set of candidates, it may learn from the probe messages not only which of the candidates that are available, but also the *Channel State Information* (CSI). This information can enable the transmitter to determine which of the available date rates that will be best suited for the current channel radio conditions. In [9] a rate adaptive MAC protocol called Reveiver-Based AutoRate (RBAR) is presented that changes the modulation scheme and thus the data rate based upon the current radio conditions. In an other aspect, [2] presents a MAC protocol that performs power control that takes into both the current radio conditions, and the location of neighboring nodes in order to allow parallel transmissions.

6.3 On Demand Multpath Link State routing

In this chapter, we present a cross layer solution where the MAC protocol can perform power and interference control by querying a number of possible candidate terminals. Each candidate is a possible next hop forwarder toward the destination, as determined by the upper layer routing protocol.

For this scheme to work successfully we need a routing protocol that can setup multiple non-disjoint paths to destinations. The routing protocol should also be able to provide the MAC layer with information about possible candidates.

We are using a hybrid on demand scheme that consists of two parts: the route *setup* part and the route *calculation* part. The route calculation part consists of calculating the cost toward different destinations using a link state database. The link state database can be created either by listening on other nearby data transmissions and overhearing or forwarding routing messages. It could also be created in a more proactive way as is in OLSR [10] or Fisheye State Routing [11].

While multiple non-disjoint routes can be calculated using the link state database, this calculation can not ensure that loops will not exist when packets are transmitted and forwarded by intermediate nodes. When a packet arrives at an intermediate

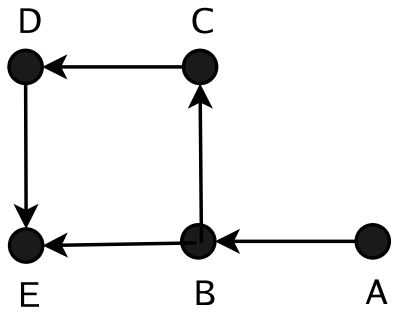
node, where multiple next hop candidates exist, the forwarding node can not be sure that this packet has not already passed through one of the candidate nodes.

One way to avoid this is by using greedy forwarding, where a packet is only forwarded to a candidate whose distance to the destination is less than the minimum distance from the current node. A problem with this approach when diversity forwarding is applied, is that the minimum distance on the short time scale is not the same as the distance on the long time scale. The long time scale is the distance information available from the link state database. The short time scale is the information learned through the querying of the different candidates. When both time scales are taken into consideration, the result might be that the candidate with the shortest distance, has a long term distance higher than the current nodes minimum long term distance. This is especially true if buffer queue size and contention information are taken into consideration when calculating the short term distance. The result is that the shortest path for this packet might be very different than the average shortest path. This means that loops can be created as a forwarding node has no knowledge of the previous path. This problem can of course be solved by only querying candidates with a lower long term distance.

Another way of solving this problem is to use source routing. However, since source routing is performed at the source node based on information included in the link state database (or in the routing cache), which is not updated frequently enough to include the channel state information, source routing can never be channel dependent. Yet another solution to the loop problem would be to include a record route option, where each hop is recorded. Each intermediate node can then make sure the packet is not forwarded to the same node twice.

In our solution, we use an on demand route setup phase that create loop free routing table entries. Loop freedom is ensured during the setup phase while still enabling diversity forwarding. When a node is about the send a packet to some destination, it first searches its routing table. If an entry can not be found, the node searches the link state database to see whether it can calculate one or more paths to the destination. If it can't do this either, it issues a *Route Request*, RREQ, message. This enables the protocol to fall back and behave as a normal on demand routing protocol such as AODV [12], but where each rebroadcasting node also adds the previous link to the RREQ packet. Each forwarding node adds this previous link information, plus any link information it finds included in the RREQ packet. This information will thus update the link state database in each of the forwarding nodes.

If the node do find a path in the link state database, it unicasts a *Route Enforce*, RENF message to each of its neighbors that it determines can be used as a next hop forwarding node towards the final destination. Each of these messages include the link state information of each forward hop as perceived by the sending node. Each node that receives this RENF message creates a forward and backward routing table entry, between the source and the destination. The node then forwards the message along the path toward the destination as specified in the RENF. Similar to the source node, each forwarding node also consult its own link state database



(a)

Fig. 6.1: Reuse of routes. A has a loop-free redundant route to destination E. If C were to reuse this route, it can not use B as next hop.

,creates and then sends a new RENF message to each of its own neighbors that it determines can be used for reaching the destination. Note that each RENF message associated between a source and destination setup can only be forwarded once. Any subsequent RENF messages from the same source destination setup phase are dropped. This prevents loops and enables each intermediate node to have unique but multiple routing table entries toward a destination.

Each entry in the routing table maps the source of the route to the destination, and possibly a multiple number of next hops. This differs from a normal routing table, which only routes according to the destination and through a specified next hop. The source address of the route is needed in order to simplify the loop freedom criteria, as the originating source node has reserved multiple routes towards the destination. While it may be possible to create routing table entries that are loop free without using the source address, which can later be reused by other nodes, this will create less multiple paths. This is because when the route is being setup, the path doesn't allow a packet to visit a previous node again. This means that any intermediate node that wishes to reuse the route also will not be able to visit these nodes, thereby limiting the number of multiple paths. An example of this is shown in figure 6.1(a). Here, A has a redundant route to E, where node B can use diversity forwarding to either C or E. If C were to reuse to this route to E, it can only use D as a next hop. By setting up its own route to E, C can forward packets to either B or D.

This is also in contrast to normal shortest single path routing schemes where the *shortest* path is the most important objective. Here, we wish to create redundant routes that can be used in case the link of the shortest next hop goes into a bad fading phase. Another important selection criteria besides the shortest path, is the state of the link *after* the next hop. If for example the buffer of one next hop candidate is queued up by three packets, choosing another next hop that is 1 hop further away can still be a good choice if its buffer is empty.

Since this is an on demand routing protocol, routing table entries will time

out if they are not being used. Because multiple redundant routes are used, this protocol is not so sensitive to ”link breaks” as normal single path protocols. Also, since the MAC protocol performs power control, link breaks will only occur when the link destination is no longer reachable. Also, this protocol has no route repair or local repair feature as for example AODV does. Even though routes are redundant, they need to be refreshed every now and then. This is done by issuing a new RENF, controlled by a refresh timer. The timeout value of this timer should depend on the mobility of the node, and the network in general, but how to determine this value is out of scope of this work.

6.4 Multipath Power and Interference Control

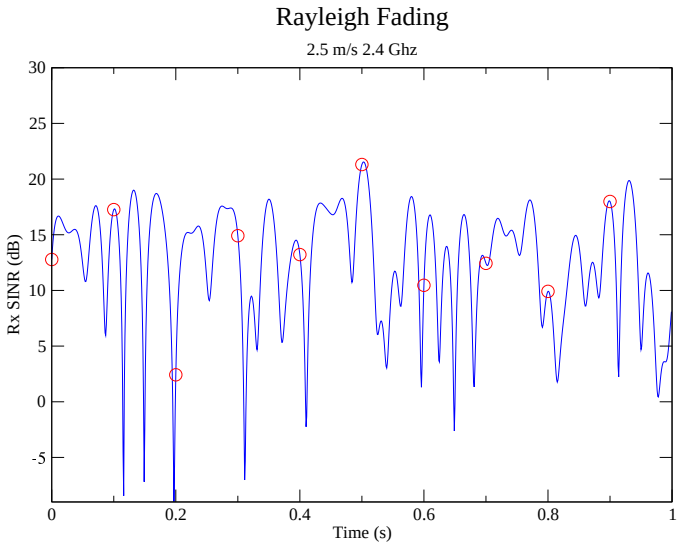
So far we have only described how a routing protocol can setup multiple paths to a destination that enables each intermediate node to make an independent relaying choice. However, in order for this scheme to allow us to make channel dependent forwarding decisions, we also need a MAC protocol that can evaluate the state of the channel towards the different next hop candidates. To make this possible we need to exchange information across layers. The MAC protocol needs information about different candidate next hops. The routing protocol also uses information, provided by the MAC protocol during receptions and transmissions, about the status and capacity of the different links. This information is included in the link state database.

6.4.1 Diversity and Relaying Node Selection

In our scheme, the MAC layer performs both Medium Access Control procedures and take the relaying decision. By evaluating different candidate nodes, as described below, in a query-reply evaluation procedure, the protocol is able to determine the best possible next hop. Not only do this allow us to determine the candidate with the best channel conditions, but it also allows us to perform power control, and implicitly interference control.

Please consider Figure 6.2. Here we see a simulated example of the signal variations in a Rayleigh fading channel when the terminal is moving at 2.5 m/s. The small circlets indicate packet reception instants and we can see how large the signal variations are for this simple scenario. When we are in a deep fade it will not be possible to receive even at the lowest rate, while at good instants it will be possible to receive at the maximum rate. If one candidate is currently in a deep fade, one can argue that some other candidate might be in a more advantageous situation due to the channels being independent. If the candidates are placed about a half a wavelength apart, it is generally enough to create independent radio channels (less than 0.1 meter for 2.4GHz).

Consider this fading situation in a single path scheme. If the channel towards our next hop is in a bad fading state, it may not be possible to transmit to that node at all, until the channel comes into a more favorable phase. If we instead use a



(a)

Fig. 6.2: Packet reception for a Rayleigh fading channel at 2.5 m/s mobility at 2.4 GHz

multi path scheme as we are proposing, we might be able to choose a better next hop, and perhaps even transmit at a higher rate, and lower power.

6.4.2 Extended Multi Path Power Control Mac Protocol

In standard 802.11 DCF, a terminal may use a simple RTS-CTS cycle to inform its neighbors about its intended transmission. RTS stands for Request To Send and CTS stands for Clear To Send. This makes all neighbors defer their transmissions for the duration of the scheduled transmission. In our protocol, MPPOW, we extend this cycle by including two new messages, DTS and ATS. These abbreviations stand for Determine To Send (DTS) and Acknowledge To Send (ATS).

In MPPOW, whenever a node wants to send a packet, it multicasts a RTS message by indicating which two or more destinations that it wishes to transmit to. It also includes information about the power level used to transmit the RTS, and how many more users that are used to transmit in parallel, *Nusers*.

When a neighbor indicated in the RTS receives this message, it calculates the current path gain to the transmitter. This is done by comparing the transmit power level as indicated in the RTS to the power level at which the RTS was received. The ratio between these levels is the gain, G:

$$G = \frac{P_{rx}}{P_{RTS}} \tag{6.1}$$

The node then uses this gain to calculate the data power level, *Pdata*, which the actual data packet will use. When calculating this power level, the node takes into account the needed SINR target ratio, μ^*, which depends upon the current data

rate. It also takes into account the current noise level, and the estimated interference level as described below. By using these values, the node is able to calculate the minimum power level, *Pmin* needed to achieve the SINR target ratio. This calculation is performed in the following way:

$$P_{min} = \frac{\mu^* P_{noise}}{G} \tag{6.2}$$

While *Pmin* is the minimum transmit power needed by the transmitter for the receiver to successfully receive the packet at this time instant, the data power level, *Pdata*, that the node will propose will include an additional interference budget ξ:

$$P_{data} = \frac{\mu^* \xi P_{noise}}{G} \tag{6.3}$$

This is done both to compensate for other interferences and noise that may start during the transmission, but also to allow for an extra budget that enables other nodes to transmit in parallel. By having this budget we increase the level of interference the receiver can tolerate during the transmission. In fact, it can tolerate an additional interference level of *Pai*:

$$P_{AI} = \frac{G}{\mu^*}(P_{data} - P_{min}) \tag{6.4}$$

So, if we allow *N* transmissions to run in parallel in addition to our own, the maximum tolerable interference, *Pmti* we can accept is:

$$P_{MTI} = \frac{P_{AI}}{N} \tag{6.5}$$

This value, *Pmti* is a constraint put on each possible parallel transmitter. Whenever they are about to schedule a new transmission they have to make sure that the amount of received interference at the already scheduled receiver does not exceed *Pmti*. It should be noted that this type of power control scheme is very close to the one used in [2].

Initially, each of these power calculations are performed with regard to the SINR target ratio, μ^*, of the highest physical layer rate. If, during these calculations it is found that either *Pdata* or *Pmin* is higher than the maximum transmit power, the target rate will be lowered to the next highest rate, and μ^* is updated accordingly. This means that the power control procedure tries to maximize the link rate under a given maximum power constraint, and only updates the power levels accordingly. It is possible to design other schemes that for example considers a certain power level that a node wishes to use, and instead modify the data rate accordingly. This would also make sense, because a higher rate typically translates into higher power, because the μ^* is higher for the higher rates. It is also possible to design and define other non linear cost functions that takes more parameters and aspects into account. This could for example be maximum forward progress, or remaining battery lifetime.

Each destination that receives an RTS replies by sending a CTS in the order they were listed in the RTS. For example if node 1 transmits an RTS 1→2,3 to destination nodes 2 and 3, node 2 will first send a CTS, and then node 3 will send a CTS. Just as with the RTS, the CTS include the power level used for transmitting the CTS. In addition to this, the CTS include the power level *Pdata* that was calculated as described above.

$$RTS(i \rightarrow j, h) = \{i, j, h, Prts, Pmap, Nusers, PayloadSize, DataRate\} \tag{6.6}$$

$$CTS(j \rightarrow i) = \{j, i, rr, Nusers, Pmap, Pdata, Pmti, Pcts, duration, rate\} \tag{6.7}$$

$$DTS(i \rightarrow j) = \{i, j, Pdata, Pdts, duration, dataOffset, rate, Nusers\} \tag{6.8}$$

$$ATS(j \rightarrow i) = \{j, i, Nusers, Pmti, Pats, duration, dataOffset, rate\} \tag{6.9}$$

In one version of the protocol, the CTS will also include the current queue size of the receiving node. In the next hop selection procedure described below, a buffered packet will be regarded as an additional hop. The reasoning behind this is that if the packet is transmitted to a node, and then have to wait in the buffer while another packet is transmitted, this has the same effect on the delay of the packet as if it were transmitted over two hops. If an other metric besides hop count is used, the buffer length should be converted into that metric in a similar way. A note of caution should be made here though. If a random access contention scheme similar to that of 802.11 DCF is used, consideration should be taken about the creation of contention, or possibly interference between loaded nodes that try to route packets based on buffer lengths. This means that although a packet might be routed to a node with an empty queue, it might still have to wait for a buffered packet it was trying to avoid. A solution to this is to apply a *buffer margin* between the considered candidates. For example, suppose a node A has two candidates B and C it can use for diversity forwarding, where B has the best channel conditions. If a *buffer margin* of 2 is used, and B has 3 buffered packets and C has none, the cost through B would be increased by 1. The total cost of routing through B (or C), would then determine which candidate that will be chosen. On the other hand, if B would have had 2 buffered packets, the cost would not have been increased and B would be chosen.

When the initiating node has received all the CTS messages it expects, or they have timed out, it chooses an appropriate destination, sets its corresponding power level *Pdata*, pick an appropriate transmission *rate* and calculates the *duration*. In addition to this, it also calculates the *dataOffset*, the time in μs until the transmission is scheduled to start. The node transmits a DTS that includes these values; *Plevel*, *rate*, *duration*, *dataOffset* as well as the power level used to transmit the DTS. This informs neighboring nodes about the scheduling of the transmission, and allows them to determine the start and end time as well as how much interfer-

ence the transmission will cause them. When the receiving node receives the DTS, it replies by sending an ATS. This ATS includes in addition to the information contained in the DTS, the value *Pmti* that states the Maximum Tolerable Inference that it can accept before it will be unable to successfully receive the packet. Other neighboring nodes that wish to transmit in parallel may do so, as long as they don't exceed this value at the scheduled receiver. The ATS is needed because there might be possible interferers close to the receiver that did not receive the DTS, and therefore needs to receive the ATS to learn about the scheduled transmission.

6.4.3 Next hop selection procedure

A very important step of this cross layer solution is the selection of the next hop forwarding node. This decision is based both upon the information gained by the MAC protocol during the RTS-CTS-DTS-ATS signaling phase, and information provided by the routing protocol. This means that the selection of the next hop is based on information gained from two different time scales, a short MAC time scale and a longer average routing time scale.

After the MAC signaling phase, we now have enough information to choose the next hop depending on the quality of the links. But even if we choose the perceived *best link* and forward the packet to that next hop, it does not necessarily mean that it is the *best path* to the destination. The link to the next hop candidate might be very good, but if conditions after that hop seems to be unfavorable according to the link state of the routing protocol, it would still be a bad choice. This is how the protocol operates on the two timescales as seen by the two layers. The decision flow can be seen as first coming from the network layer with a set of candidates, then going to the MAC layer for candidate evaluation, then back to the network layer for the final next hop decision, then again down to the MAC layer for transmission of the packet. This is not to be seen as if the packet is being passed back and forth between the layers. The packet is only "passed" once, but the two layers have callback functions into each other in a very cross layer manner.

We determine the best next hop relaying node in the following way:

1. Determine what candidates to include in the MAC signaling phase by evaluating the link state database. This database has been updated by the routing protocol. The candidates with the least cost will be used in the evaluation.

2. Perform the MAC signaling evaluation.

3. Determine the short term cost, C_{STi} to each next hop *i* based upon the MAC evaluation.

4. Determine the average long term cost, C_{LTi} to each next hop *i* based upon the link state database.

5. Determine the routing cost, C_{RCi} to the final destination through each next hop *i* based upon information in the link state database.

6. Determine the current path cost, C to the final destination through each next hop by subtracting the long term cost from the short term cost and adding this difference to the routing cost: $C = ((C_{STi} - C_{LTi}) + C_{RCi})$.

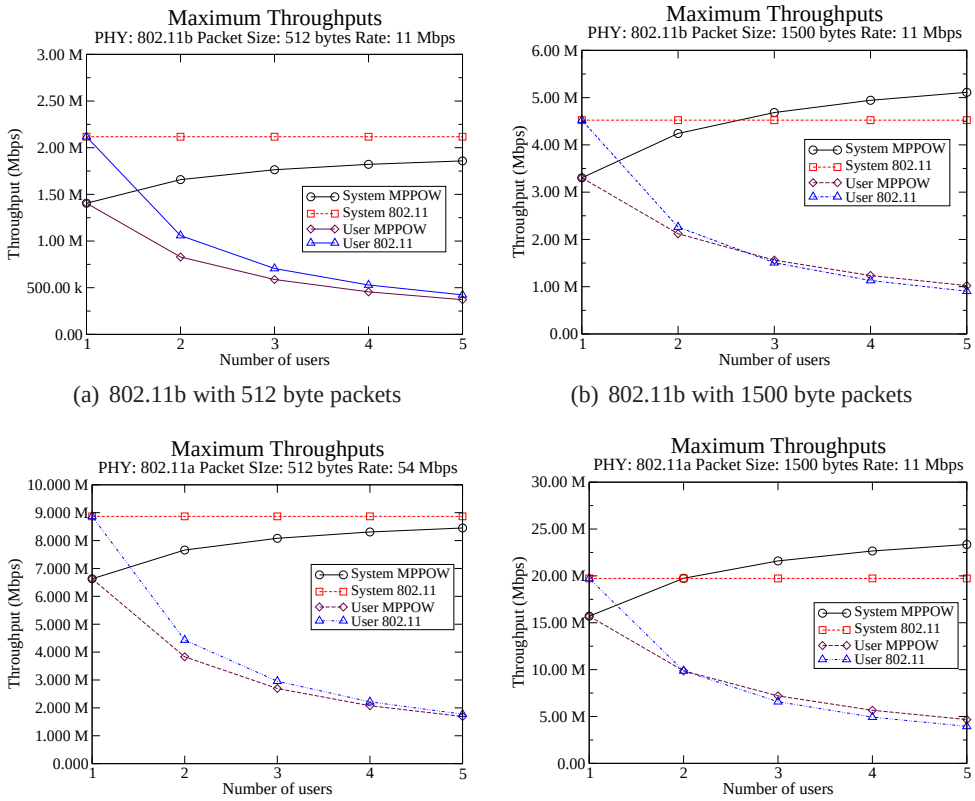
7. Choose the next hop relaying node with the least current cost, C.

Another important question here is how we determine the actual cost of each link, ie. the metric. In most routing protocols for ad hoc networks used today, a simple hop count metric is used. This is a fairly simple and robust metric that considers the fact that more hops means that more resources and capacity have to be used for transferring a packet to its destination. This is especially true in a wireless ad hoc network, where one transmission not only affect other transmissions on the same link, but all other possible transmissions operating on the same channel in the area around the node, and around the receiver.

Other metrics and more sophisticated cost functions are also possible. In the simulations performed for this chapter a metric was used that defined the cost of a link as the inverse bit datarate. This makes sense if we consider the following case: consider two links where one link has a bitrate of 1Mbps and the other has 2Mbps. Since the data transmission duration on the 1Mbps link is twice as long as the duration for the 2Mbps link, it makes sense that the cost of that link is also twice as high.

Figure 6.3 shows the theoretical maximum throughput that the extended version of the protocol is able to achieve for different 802.11 physical layers, packet sizes and number of users. Here we can clearly see that the packet size used for each transmitted packet is a very important parameter. When the packet size is small, the signaling overhead induced is simply more than the protocol can handle, and standard 802.11 will always be more efficient. However, the reason that the difference between MPPOW and 802.11 is larger for 802.11b, is the long preamble size. 802.11b uses a preamble that takes $144\mu s$ to transmit, while 802.11a uses a $16\mu s$ preamble. Since this time is added to every frame transmitted, i.e. RTS, CTS, DTS etc, this translates into a lot of overhead, reducing the performance of the protocol.

When the packet size is larger, MPPOW becomes more efficient than 802.11, both from a system wide view as well from an individual user's perspective, when the number users transmitting is either two or three or more. If only one node is transmitting the overhead is simply too much, and we can't reach any improvement in throughput. It should be pointed out that these figures only show the maximum throughput on a single link, and doesn't consider the multi-path and power control features provided by the protocol. The conclusion that can be drawn from these figures, is that if the packet size isn't too small, throughput especially from a system perspective can be increased, if users are transmitting in parallel. This means that channel contention can be used to introduce parallel transmissions, and the throughput can increase for flows over several hops.



(a) 802.11b with 512 byte packets

(b) 802.11b with 1500 byte packets

(c) 802.11a with 512 byte packets

(d) 802.11a with 1500 byte packets

Fig. 6.3: Maximum Throughput for MPPOW and 802.11 DCF for different 802.11 PHYs and packet sizes. The number of users trying accessing the channel either in parallel or in alteration is also shown.

6.4.4 Lite Multi Path Power Control Mac Protocol

As we saw in the previous section, the MAC protocol described in section 6.4.2 relies on a quite heavy signaling phase that takes place before each packet is transmitted. If two nodes wish to transmit in parallel, the RTS-CTS-DTS-ATS phase will be performed twice. This means a lot of overhead. Still, we may gain from this if two packets are transmitted in parallel, so that one transmitter does not have to wait for the other to finish. But, the overhead is also heavily dependent upon the type of physical layer used. In 802.11b for example, which is used for simulations in this study, the preamble and physical layer headers is always transmitted at 1Mbps, even though the data payload can be transmitted at a higher data rate such as 11Mbps. This means that the 192 overhead bits in 802.11b, can be regarded as 2112 overhead bits with the 11Mbps datarate. If we also consider that these bits will be used for each of the RTS-CTS-DTS-ATS signaling packets, the overhead can be quite significant. In fact, as we discussed above, if we consider two parallel transmitters with 1500 bytes payload operating at 11Mpbs, we will not gain

anything and only slightly for 5.5Mbps, but more for 2Mbps and 1Mbps. If an other type of physical layer is used, such 802.11a or 802.11g in ”g-only mode”, the situation is different.

For this reason we also have a light weight version of our protocol. In this version, only RTS and CTS messages are exchanged, and transmissions are not scheduled in parallel. We still perform the candidate evaluation procedure as described above, because the RTS is still multicasted and multiple CTS messages are received. We still also perform the power and rate control, and other nodes are still actually allowed to transmit in parallel. This is done by using information from the CTS, and as long as they do not interfere with an ongoing transmission, the parallel transmission is allowed. They are however, not scheduled in parallel through the full multiple RTS-CTS-DTS-ATS phases. The light version of the protocol is used for some of the simulations that will be presented in the next chapter.

6.5 Simulations

The protocols described in this paper were implemented and evaluated in the Qualnet simulator [13]. Qualnet is a discrete event network simulator that includes a rich set of very detailed models.

6.5.1 Packet Capture and Interference calculation

The packet capture and interference calculation method presented in this section has been used for all simulations in this and the following two chapters.

Computation of interference and noise at each receiver is a critical factor in wireless communication modeling, as this computation becomes the basis for the SINR or SNR (Signal to Noise Ratio) that has a strong correlation with PER (Packet Error Rate). The power of interference and noise is calculated as the sum of all signals on the channel other than the one being received by the radio plus the thermal noise. The resulting power is used as the base of SNR, which determines the probability of a successful signal reception for a given frame. For a given SNR, Qualnet uses the BER (Bit Error Rate) signal reception model. The BER based model probabilistically decides whether or not each frame is received successfully is based on the frame length and the BER deduced by the current SNR and modulation scheme used at the receiver.

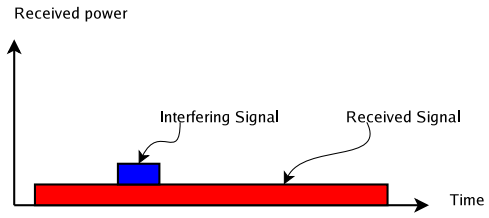


Fig. 6.4: A signal being received is affected by an interfering signal

An important question in network packet simulators are how interfering signals are treated, see figure 6.4. In Qualnet, when a new signal arrives that interferes with the current reception, a new SINR is calculated and the resulting BER is determined. A new PER is then calculated and a “coin is tossed” to determine if the packet can be received or not. If it is determined that the packet can not be received, the receiver cancels the current reception and goes into sensing mode. In standard Qualnet, this procedure is performed regardless of the duration of the interfering signal. For example, if the duration of the interfering signal is very short compared to the signal being received, simply looking at the SINR may not be ideal as bits lost due to interference may be recovered using the system applied error correcting codes.

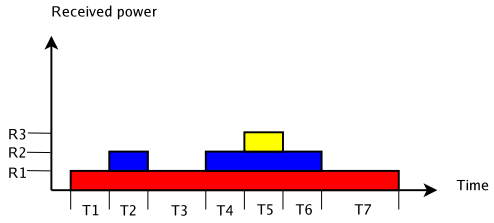


Fig. 6.5: A signal being received (red) is affected by several interfering signals (blue, yellow). The total interference energy is: T2*(R2-R1) + T4*(R2-R1) + T5*(R3-R1) + T6*(R2-R1).

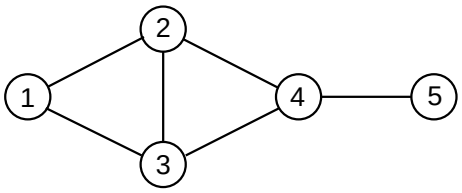
So, when we have one or more interfering signals, it instead makes sense to study the interference energy rather than the interference power, see figure 6.5. This is achieved by both looking at the cumulative interference power, as well as for how long each individual interferer makes an interference contribution. Now, the SINR is calculated by looking at the received signal energy compared to the received interference and noise energy. A simple method for calculating these energies is to divide the reception time into different periods. Whenever a new interference period starts or ends, the energy of the previous period is calculated and added to the cumulative interference energy. Whenever a signal event occurs within the simulator, the energy level is updated where the signal event can be one of the following: a new signal is received, a signal being received ends, an interfering signal start, or an interfering signal ends. The time of these events are recorded in order to determine the duration of each period, see figure 6.5. I have implemented the energy and SINR calculation within Qualnet as described in the Calculate_Interference method below.

After this study had been conducted, a similar interference calculation method was described in [14]. Here each parallel signal that occurs during a signal reception is recorded in an interference list associated to the received signal. After the signal being received ends, this list is evaluated to calculate the resulting interference. While the end result in this approach is the same as have been described, it requires more memory resources.

```
Calculate_Interference(e)
{
  if (e.EventType == SignalReceive) {
    SignalReceiveStartTime = NOW
    SignalReceivePowerLevel = e.SignalPower
  }
  else if (e.EventType == InterferenceStart) {
    if (InterferencePowerLevel == 0)
      InterferencePeriodStartTime = NOW
      InterferenceEnergy = 0
      InterferencePowerLevel += e.SignalPower
    }
    else {
      InterferencePeriod = (NOW - InterferencePeriodStartTime)
      InterferenceEnergy += (InterferencePeriod * InterferencePowerLevel)
      InterferencePowerLevel += e.SignalPower
      InterferencePeriodStartTime = NOW
    }
  }
  else if (e.EventType == InterferenceEnd) {
      InterferencePeriod = (NOW - InterferencePeriodStartTime)
      InterferenceEnergy += (InterferencePeriod * InterferencePowerLevel)
      InterferencePowerLevel -= e.SignalPower
      InterferencePeriodStartTime = NOW
  }
  else if (e.EventType == SignalEnd) {
    SignalReceiveEndTime = NOW
    InterferencePeriod = (NOW - InterferencePeriodStartTime)
    InterferenceEnergy += (InterferencePeriod * InterferencePowerLevel)
    SignalReceivePeriod = (NOW - SignalReceiveStartTime)
    SignalReceiveEnergy = (SignalReceivePeriod * SignalReceivePowerLevel)
    ThermalNoiseEnergy = (SignalReceivePeriod * ThermalNoisePowerLevel)
    SINR = SignalReceiveEnergy/(InterferenceEnergy + ThermalNoiseEnergy)
  }
```

Another important factor in network packet simulators is how physical layer capture is modeled. This refers to what happens when two signals of different signals collide at a receiver. Which signal should be received? In Qualnet this depends on which signal arrived at the receiver first. If the first signal is stronger than the second, the second is treated as noise for the reception of the first. If the second is stronger, the current packet being received is marked as received in error. This is done irrespective of how strong the second signal is, or how weak the first signal is. What we need is a capture and reception model that considers the behaviour of a correlation detection circuit when a new and stronger signal arrives that cause interference with the ongoing reception process. In cases where the energy of the new signal is sufficiently higher than the initial signal, then there is a possibility that the correlation detector will be "reset" by the increase in energy. This capture model has been described in [15] where they point to results that indicate that if the new signal is 3 to 5dB stronger, the initial signal is dropped. This model have been used in the current simulations with a threshold value of 5dB.

6.5.2 Network Setup and Results



(a)

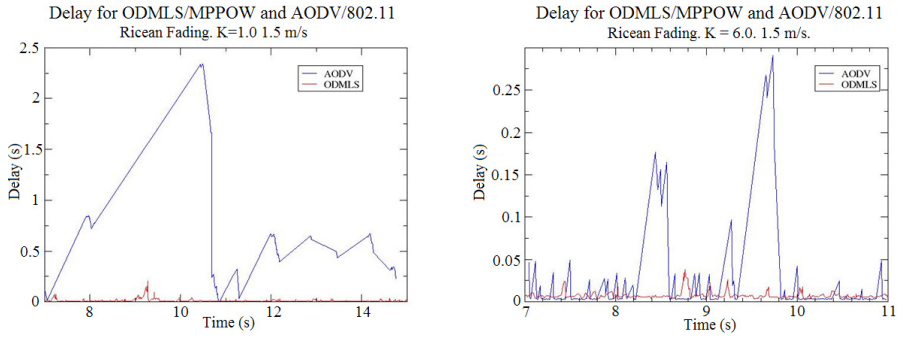
Fig. 6.6: Network setup used during the simulations.

Figure 6.6 describes the network setup used for the simulations in this study. The traffic used in the simulations are UDP Constant Bit Rate (CBR) 275kbps data flows between node 1 and node 5. This is a fairly simple network setup, but it still provide intermediate node with multiple next hop candidates at several nodes, i.e. node 1,2 and 3. It allows the possibility for parallel transmissions, for example node 1 and node 3 can transmit in parallel to node 2 and node 4 respectively. Each of the available links fades according to the Ricean distribution, see figure 6.2. This means that although the independent distances between each of the nodes never changes, the signal strength still varies a lot. This can be compared to a situation where the whole network is moving with a certain speed through the environment, although the network topology never changes. This technique allows us to completely specify the network topology as we wish to, while still having fading links as if the nodes were moving through a variable environment.

Our ODMLS routing protocol combined with the MAC used the extended

MAC version of MPPOW. RTS and CTS messages were transmitted at 15dBm at 1Mbps. The physical layer used in all the simulations were 802.11b, and the routing protocols used for comparison were AODV [12] and OLSR [10]. AODV and OLSR used the 802.11 MAC protocol.

Figure 6.7 shows the instantaneous delay during 8 seconds for both ODMLS/MPPOW (full MPPOW) and AODV/802.11. In figure 6.7(a) Ricean fading with a K-factor of 1 is used and in figure 6.7(b) a K-factor of 6. In 6.7(b) where fading is less severe we only see a significant difference in delay between AODV and ODMLS during two half a second long periods, at 8.5s and 9.5s. During these periods, the link AODV uses goes into a bad fading phase. For the duration of such a phase, AODV will experience several unsuccessful packet transmission attempts over the link, which increases the packet end-to-end delay. During periods like these, ODMLS/MPPOW is able to avoid the bad links through the next hop diversity selection. When fading becomes severe, i.e. a lower K factor, the bad fading phases comes more rapidly and longer. Longer in this case for AODV, in fact longer than the interarrival time of packets, causes a build up of the queue length resulting in longer queuing delays. Although AODV might be triggered to believe the link is broken, and to start a route repair procedure, it still takes a long to repair the route. Even if the route repair is successful, the new link will soon become bad again. Here we clearly see the benefits of diversity forwarding, where bad links can be avoided on a per packet basis.



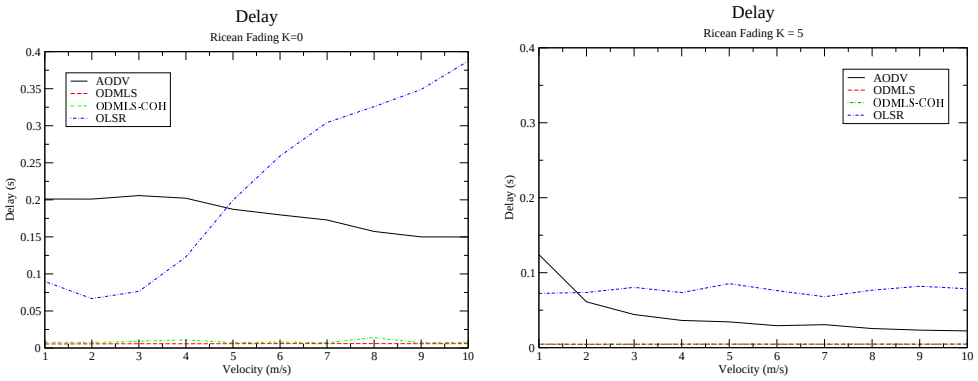
(a) Ricean Fading K=1

(b) Ricean Fading K=6

Fig. 6.7: Realtime delay during a 8 second interval for various protocols

In figure 6.8 the light version of the MAC protocols is used, and we see the average end-to-end delay instead of the instantaneous delay for different protocols, but here also for OLSR and different K-factor velocity speeds. The main observation here is that during the faster fading case of K=0 (Rayleigh Fading, figure 6.8(a)), delay increases for OLSR at higher mobility. This stands in direct relation to the rate of the topology updates sent by OLSR to update its link state. As the mobility increases, the OLSR link state database will become more and more inaccurate. For AODV it is actually the direct opposite; as the mobility increases

AODV will more and more often believe the link to be broken, causing it to repair the route. The reason that the AODV performance increases with higher velocity, even though it is not mobility in terms of topology movement, is that when the route is first setup, the first RREQ that is received at the destination will determine the route. This route may not be the most optimal route. As the fading velocity increases, AODV will more often believe the route to be broken, and therefore repair the route. The chances that a good path is eventually chosen therefore increases. ODMLS/MPPOW maintains a low delay as long it manages to find new valid and good next hop relays, which it does.

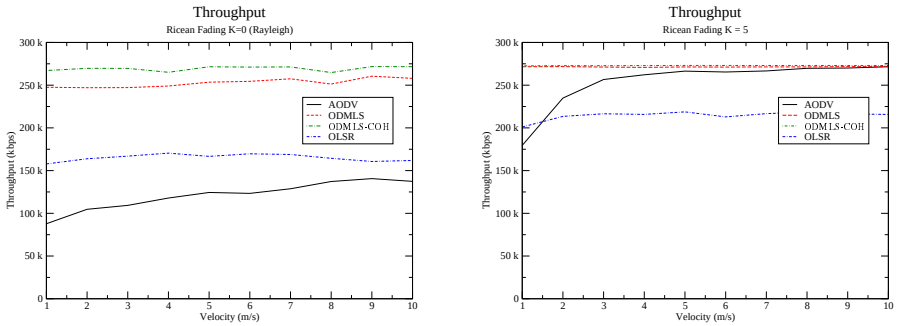


(a) Ricean Fading K=0 (b) Ricean Fading K=5

Fig. 6.8: End-to-end delay during different fading factors and mobilities for various protocols

In figure 6.9 we can see the same improving trend for AODV as we did for the delay. The reason for the low throughput for AODV during the faster fading (K=0) is packet drops. For the slower fading situation (K=5, fig 6.9(b)), there is no significant difference between AODV and ODMLS for the higher speeds. Here, AODVs repair procedure is effective enough, giving it a high throughput, but at the price of a higher delay as we saw in figure 6.8(b). ODMLS/MPPOW in this case maintains a high throughput and low delay regardless of the fading velocity. We also see two different curves for two versions of ODMLS; ODMLS-COH and ODMLS. The difference between these is that ODMLS-COH includes an extra feature where a failure to receive an ACK after data transmission is first regarded as a transmission failure due to fading. This means that the transmitter will wait for a duration of one and half times the coherence time of the channel, before retransmitting. In the simulations this value is preset and fixed based on a known channel coherence time and gives a slight throughput improvement, mainly for K=0.

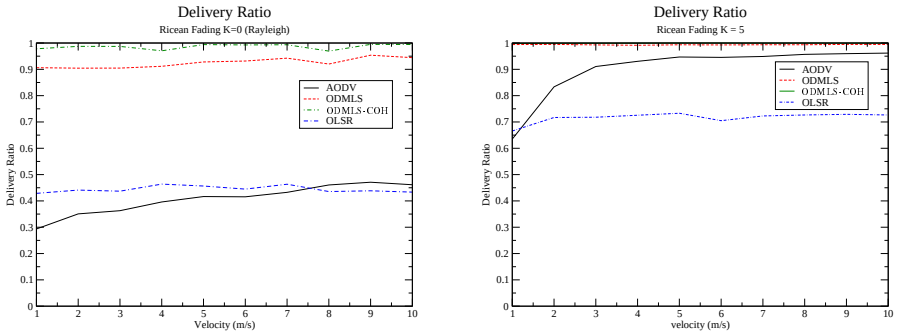
However, this comes at the price of a slight increase in delay, which can be seen in the lower part of figure 6.8(a) at close examination. While the gain of doing this isn't very high, the gain in our simulations is still roughly a 7% increase in throughput for K=0, but less than 1% for K=5.



(a) Ricean Fading K=0 (b) Ricean Fading K=5

Fig. 6.9: Throughput in kbps during different fading mobilities and Ricean K factors for various protocols

The channel coherence time is a parameter that could be provided by the physical layer. It could for example be determined by looking at the time difference between the events where the signal crosses a certain reference level. Other methods that depend on the type of physical layer used is also possible.



(a) Ricean Fading K=0 (b) Ricean Fading K=5

Fig. 6.10: Delivery Ratio for various protocols and different Ricean K factors

Figure 6.10 confirms that packet drops cause the lower throughput for AODV and OLSR. If we compare Figure 6.10 and figure 6.9 we can see the close relation between delivery ratio and throughput; when the packet delivery increases, so does the throughput, as expected. ODMLS manages to deliver a high number of packets, but without the coherence time feature (retransmission hold-off), the packet drop ratio is around 10% for K=0. For K=5 almost all packets are delivered.

6.6 Discussion and conclusion

We have presented a cross layer solution that defines and specifies a MAC and a routing protocol that interact in order to create efficient diversity forwarding. The routing protocol (ODMLS) is semi reactive and operates by setting up routes on

demand, but maintains a link state database that is continuously updated by using a promiscuous mode operation, as the one specified in 802.11, and listening to other data and control traffic. The routing protocol setup multiple non-disjoint paths between a source and destination and presents the MAC layer with a set of candidate next hop forwarding nodes. The MAC protocol evaluates the candidates presented by the routing protocol, and performs power, rate and interference control in addition to implementing the diversity forwarding capabilities. The MAC protocol also has the ability to dyanmically schedule neighboring parallel transmissions, as long as they don't interfere with each other. Both protocols are involved in the process of routing a packet, but they operate on different timescales and on different horizons. The routing protocol operates on information that is provided by the link state database, which is averaged and filtered over time. The MAC protocol operates on a shorter timescale and tries to determine the status and condition of a link with a ms resolution. The routing process is truly cross layer, and the final routing decision is actually made the MAC protocol, by using the routing table created by the routing protocol in combination with its own fast link evaluation. This faster link evaluation is what enables it to adapt to bad fading situations. Even though power control is performed which improves performance, it is the link diversity and the fading awareness that improves performance the most. This has been confirmed by a control experiment, where the power control feature was turned off. The gain was still very high, although slightly less than with power control, and it therefore confirmed that the highest gain is accomplished through link diversity.

Simulations show that the end to end delay can be significantly reduced.

The presented solution indicates that significant performance gains may be achieved, as has been indicated through a set of simulations in a 5 node network topology.

BIBLIOGRAPHY

[1] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification, IEEE Std 802.11-1997*. The Institute of Electrical and Electronics Engineers, New York, 1997.

[2] A. Muqattash and M. Krunz2006. A single-channel solution for transmission power control in wireless ad hoc networks. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, Tokyo, Japan*, May 2004.

[3] M. Cesana, D. Maniezzo, P. Bergamo, and M. Gerla. Interference aware (ia) mac: an enhancement to ieee802.11b dcf. In *Proceedings of the IEEE Vehicular Technology Conference 2003, VTC fall 2003, Orlando, Fl, USA*, October 2003.

[4] P. Larsson. Selection diversity forwarding in a multihop packet radio network with fading channel and capture. *ACM SIGMOBILE Mobile Comnputing and Communication Review*, 5(4):47–54, 2001.

[5] Shweta Jain and Samir R. Das. Exploiting path diversity in the link layer in wireless ad hoc networks. In *Proceedings of the 6th IEEE WoWMoM Symposium, Taormina, Italy*, June 2005.

[6] J. Wang, H. Zhai, Y. Fang, and M. C. Yuang. Opportunistic media access control and rate adaptation for wireless ad hoc networks. In *Proceedings of the IEEE Communications Conference (ICC'04), Paris, France*, June 2004.

[7] P. Larsson and N. Johansson. Multiuser diversity forwarding in multihop packet radio networks. In *Proceedings of the 2005 IEEE Wireless Communications and Networking Conference*, volume 4, pages 2188– 2194, March 2005.

[8] M. Souryal and N. Moayeri. Channel-adaptive relaying in mobile ad hoc networks with fading. In *Proceedings of the IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, California*, pages 142–152, September 2005.

[9] G. Holland, N. Vaidya, and P. Bahl. A rate adaptive mac protocol for multi hop wireless networks. In *Proceedings of the Seventh Annual ACM/IEEE*

Internation Conference on Mobile Networking (MobiCom'01), Rome, Italy, October 2001.

[10] P. Jacquet, P. Muhlethaler, T Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *IEEE International Multi Topic Conference*, 2001.

[11] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *Proceedings of IEEE International Conference on Communications, ICC*, pages 70–74, 2000.

[12] C. Perkins. Ad-hoc on-demand distance vector routing. In *Second IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[13] Scalable Networks. Qualnet network simulator, version 3.9. 2005.

[14] Ruben Merz, Jean-Yves Le Boudec, and Jorg Widmer. An architecture for wireless simulation in ns-2 applied to impulse-radio ultra-wide band networks. Technical Report LCA-REPORT-2006-126, 206.

[15] C. Ware, J. Chicharo, and T. Wysocki. Simulation of capture behaviour in ieee 802.11 radio modems. In *Proceedings of the IEEE Vehicular Technology Conference 2001, VTC fall 2001, Atlantic City, NJ, USA*, October 2001.

7. CHAPTER VII

Urban Mesh and Ad hoc Mesh Access Networks

7.1 Introduction

Wireless mesh networks is an area that has been receiving a lot of attention within the networking community the last few years. They can be regarded as wireless networks where each node can function both as an access point, and as a router responsible for forwarding traffic from other parts of the network. Wireless ad hoc networks are another type of wireless networks that is closely related to mesh networks. The main difference between mesh and ad hoc networks is that ad hoc networks are constructed by user terminal nodes, and these user nodes are expected to be highly mobile, while mesh nodes are expected to more static and have a more permanent power supply.

The mobility aspect has led to extended amount of research performed on the topic of mobile routing, because as the user nodes move around in the network, the network topology will constantly be changing. Within this topic, IETF has had a very important role, resulting in standardization through experimental RFCs, of a couple of routing protocols.

While ad hoc networks are still waiting for a killer application into the commercial market, mesh networks devices are already available through different manufactures. Mesh networks are built as cost effective wireless access networks, and have been deployed in some cities as wireless MAN networks.

Mesh networks are also considered a promising technology for emergency, search and rescue operations. This is because mesh networks can rapidly be deployed and be made to need little or no configuration. With well configured mesh devices, the only thing an emergency team needs to do to get a fully operational access networks, is to bring the devices to a location, turn them on, and distribute them in the area. This is especially useful in areas where very little communication infrastructure is currently available, or where the communication infrastructure has been destroyed.

This chapter focuses on both ad hoc and mesh networks, and on a combination of the two. We investigate how well these types of networks can expected to be operational in a typical city environment, and a type of suburban environment. As far as we know, this is the first simulation study of ad hoc/mesh networks in an urban setting, that takes into account fading and the propagation effect of walls of different buildings, in combination with different well known routing, MAC

and physical layer protocols. We also present and analyze a new MAC and a new routing protocol, are more sensitive to the current radio conditions. A well known problem in urban city environments is the effect multi path propagation has on the received signal strength. While the effects this fast fading has on the physical layer is well known, and is taken into account by most physical layer protocols, it has not yet been properly addressed on the network layer. The traditional view of a network link is that either it is up, or it is down, and this must be handled properly by the routing protocol. The wireless routing protocols proposed by the IETF take into account that links can rapidly appear or disappear as nodes move around in the network. What they do not address is that the quality of links over a wireless fading channel, fluctuates heavily. This means that while a link may be unusable due to heavy fading, it may be so for only a few milliseconds. The failure of a data packet over a wireless link may therefore be the result of the channel being in a deep fade, not the link being unavailable.

This chapter therefore also present a type of forwarding called diversity forwarding. It is based on the idea that if one link is currently unavailable due to fading, it may be possible to use another available link with a currently better communication channel.

We therefore also analyze mesh networks that use standard routing and MAC protocols, in an urban setting with fading channels and compare them with our newly proposed protocols. We also analyze a combination of mesh and ad hoc networks, where user nodes connect to the mesh network over several hops. This means that user nodes need to use ad hoc routing in order to reach the mesh access network.

7.1.1 Related Work

The performance of mesh networks has been studied previously, although not in ad hoc and urban scenarios. In [1], the authors investigate the performance of VOIP traffic in mesh networks and study the number of supported calls. By using packet aggregation and header compression, they increase the number of supported VoIP calls to about 30 for 1 hop and 10 for 3 hops. [2] analyzes interference and the performance throughput of CBR connections in mesh networks.

Another very active research area is that of multi channel protocols for wireless and ad hoc networks. This topic has been presented in many papers, and these differ mainly on how they control access to the different channels, comparisons on some of the schemes can be found in [3] and [4]. These are most often designed as scheduled access protocols, where the MAC scheme controls which channels can be used and when. Some schemes are more opportunistic or use physical or virtual carrier sensing [5], [6] and [7] to determine the channel, while some use busy tones [8], or channel hopping [9].

Several papers have recently discussed the topic of diversity forwarding. One of the first protocols written on this subject was the 2001 paper “Selection Diversity Forwarding” (SDF), by Larsson et al [10]. This scheme works by first

letting a node broadcast the data packet to a subset of potential relay nodes, listed in priority order. These nodes send back acknowledgement messages, and based on this information, the node takes the forwarding decision. This decision consists in electing one of the potential relays as the actual relay node. This scheme was later modified and updated in their "Multiuser Diversity Forwarding" (MDF) paper [11]. In MDF, a small probe pilot is first used that indicates potential relays, which after reception of the pilot reports their perceived channel quality back to the sender. The sender then determines the next hop based not only on the channel state, but also picks the appropriate flow and rate. A main difference between SDF and MDF is that in MDF the evaluation is conducted before the data packet is actually transmitted. Several other papers have presented similar protocols that specify various MAC and network layer schemes for diversity forwarding [12], [13], [14], [15], [16] [17], [18], [19]. This chapter presents a protocol that operates on principles similar to those used in MDF. It also uses a simple channel hopping scheme, as will be explained. The main difference between these earlier protocols and the protocols used here, is that here diversity forwarding is performed in combination with power control, rate control, channel selection and in consulation with the topology of network *after* the candidate evaluation has been performed. It is also the first study to evaluate these concepts in a city and suburban based environment, with voice related traffic.

7.2 Urban Mesh Ad Hoc Network Types

Depending on the type of network and node architecture used, we can define a number of mesh ad hoc network architecture types. The type of map we are considering is a Manhattan city grid.

7.2.1 Pure Ad Hoc Networks

The pure ad hoc network is a network that only consists of user nodes equipped with a single wireless interface running an ad hoc routing protocol. This type of network is typically connected through a single MAC and PHY technology operating on a single channel. If the routing protocol supports routing over multiple channels, a multi channel solution is also possible. The advantage of this solution is that it is very simple. All that is needed is a device with wireless capabilities, running an ad hoc routing protocol. The disadvantage is that in an urban environment, the network must cover a large geographical area and therefore many links. This will put a hard strain on the available resources, but the performance may still be acceptable for some types of applications, such as voice or text messaging. In the future, the available bandwidth and bitrate might be sufficiently high making this type of solution acceptable also for other type of applications.

7.2.2 Supported Ad Hoc Networks

This architecture is very much like the pure ad hoc network, but with the addition of some fixed and static support nodes. The support nodes are placed at strategic locations in the network, and aid the network by forwarding routing and data traffic. Since these nodes will be fixed, their placement can be planned in such a way as to help maintain good connectivity throughout the network. In the pure ad hoc network, the user nodes are all assumed to be mobile, and the connectivity of the network can therefore not be guaranteed. The support nodes should be placed at locations that maximize the general connectivity, or in locations that normally have a low user node density.

7.2.3 Single Hop Mesh Networks

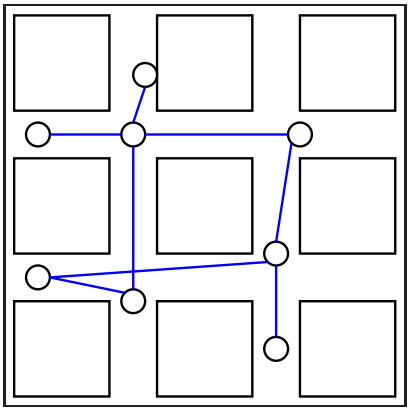
This is currently the most common type of mesh network. The network consists of Mesh Points (MP) equipped with at least two wireless interfaces. The MP is connected in a dedicated routed network used for the transport of user data. The MPs maintain the mesh network using one wireless interface, while communicating with the user nodes on another interface. User nodes in the network are equipped with only one wireless interface, which it uses to communicate directly with a specific MP.

If the wireless interface of the user nodes is of 802.11 type, the interface can be made to operate in either infrastructure mode, or ad hoc mode. If the interface of the user node and MP operates in infrastructure (association) mode, no special software or configuration is needed. The user can then use the 802.11 configuration applications provided by all modern OS, and connect to the urban network as if it was a normal 802.11 access point.

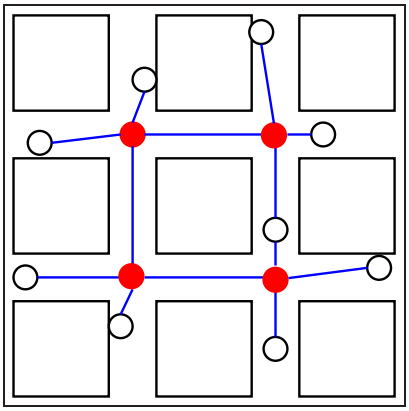
There is nothing that prevents the 802.11 user nodes and MPs from operating in ad hoc mode. However, in this case some other applications are needed that determine the existence and address of an MP, and accordingly configure the node and the network interface information. In this case, the user node can also be made to communicate with other user nodes directly. If an ad hoc routing protocol is running, these other user nodes could also be reached over multi hop routes.

7.2.4 Urban Mesh Ad Hoc Networks

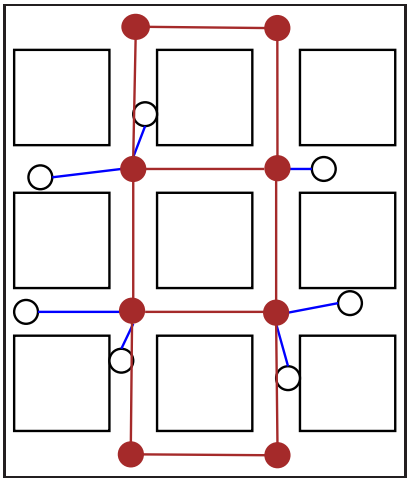
In this type of network every user node runs an ad hoc routing protocol. Every MP also runs this protocol on at least one of its interfaces. The MPs also run an other routing protocol that configures the mesh part of the access network. A typical feature here is that user nodes can connect to the mesh network through MPs over multi hop routes, using other user nodes as relays.



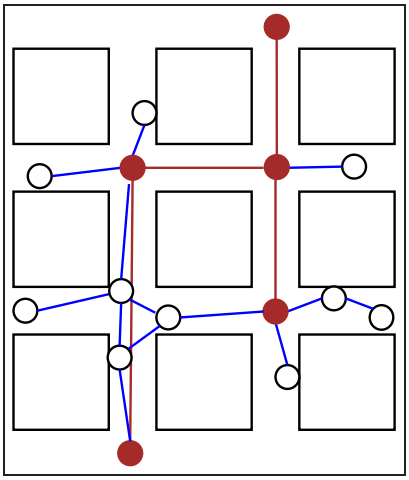
(a) Pure Urban Ad Hoc Network.



(b) Supported Urban Ad Hoc Network.



(c) Urban mesh network. Nodes access the network in a single hop.



(d) Urban Mesh Ad Hoc Networks. Nodes can access the network through multihop routes.

Fig. 7.1: The four network types

7.3 Mesh Network Registration Application

Every user node runs an application that registers the user node's address to a Mesh Point (MP). This registration is needed by the mesh network to keep track of the current location of user nodes in the network. Every MP maintains a list of all its currently registered nodes, and periodically exchanges this information with the other MPs. This allows every MP to know the location of every user node in the network.

7.3.1 User node data transmission

When a user node has data that it needs to transmit, it tunnels its data packets to its currently registered MP. The packet is routed towards the MP using the ad hoc network routing protocol used by all user nodes. When the packet is received at the MP, the MP determines which MP the destination address belongs to, and tunnels the packet to this destination MP. If the destination IP address is not found in the MP registration list, and the IP address doesn't belong to the user node subnetwork, the packet is tunneled to the Internet Mesh Point Gateway, IMPG, if such an MP is available. When the data packet is received at the destination MP, it is detunneled and routed to the final destination using normal ad hoc routing.

7.3.2 Mesh Point Registration procedures

When a user node joins a mesh network, it first needs to determine a MP that it can register with. If no local MP is known, the node *may* broadcast a *Mesh Point Search* message. If a MP receives the search message, it will respond by sending a *Mesh Point Search Reply*. However, MPs will periodically broadcast a *Mesh Point Advertisement* message with the IP TTL or Hop Limit field set to 1. Each User Node (UN), that receives this message will update its *Mesh Point List* with the corresponding information about the MP: the *Mesh Point Address*, the *distance* in hops, the *lifetime* of the advertisements, the *sequence number*, and the *announce time* (i.e. the current time). User nodes will use this list to determine the best point of attachment (i.e. the best MP) to the network, and to determine when a handover is needed to a new MP.

Type	Mesh Point Address	User Node Address
Current Distance	Sequence Number	

Fig. 7.2: The fields of the *Mesh Point Search Reply* and *Mesh Point Advertisement* message

.

The purpose of the advertisements, and the search replies, are for user nodes to learn about the existence and distance to different MPs. An important question, when we have an ad hoc network structure, is how the advertised MP information should be disseminated across multiple hops. One possibility is to increase the TTL or Hop Limit to an appropriate value, and then let each user rebroadcast the

message upon receiving it. The main problem with this approach is that every UN that receives an advertisement needs to rebroadcast it. For example, if 5 nodes are located within the one hop range of a MP, they will hear the same advertisement 5 times, possibly within a short time frame. If one of these nodes are within range of 5 additional two hop nodes, it will hear the same advertisement 5 more times, and so on. As the network spreads over many MPs, and every advertisement with an active TTL will be rebroadcasted, a lot of advertisements will be spread over the network.

A simpler solution is to let each user node send their own advertisements with the TTL or Hop Limit value set to 1, instead of rebroadcasting advertisements from MPs. The number of advertisments a node overhears within a time frame will then be bounded by the node density, the number of nodes located within communication range of the node. This process can ideally be coordinated with different neighbor discovery processes, used by many different protocols, as it provides information about the state and existence of different neighbors.

When a user node determines that it needs to register with the network, it searches its *Mesh Point List* for the MP that it determines to be the most appropriate, and issues an *Mesh Point Registration Request*. How to determine the best MP is not covered here, but any type of method can be used, such as the MP with the shortest hop count, as is used in this chapter. The purpose of the registration request is to let the mesh network know about the existence of the user node, and that this user node intends to use the indicated MP as an access point to the network. The *Mesh Point Registration Request* message includes the following fields:

Type	Mesh Point Address	Current Distance
User Node Address	Sequence Number	Registration Lifetime
Handover Flag		

Fig. 7.3: The fields of the *Mesh Point Registration Request* message

.

When the MP receives the request, it registers the node by adding it to its *Registration List*. The Registration List is a list that contains all the user nodes and their most recent sequence number, currently registered with the mesh network. These lists are periodically exchanged among MPs in the network in order for them to know the location (i.e. their MP) of every user node in the network. The lists are exchanged in an approach similar to that of how the Distance Vector protocol exchange routing information, meaning that a message containing the list, or a part of the list, is broadcasted to all one hop MP neighbors. The MP that receives the registration request, then transmits a *Mesh Point Registration Reply* message (fig 7.3b).

The registration lifetime of the request is the period of time in seconds that the user node wishes to be registered. If the MP accepts the *Lifetime* value proposed by the user node in the Request message, it copies this value into the Registration Lifetime field of the Reply message. Otherwise the MP will use some other default or determined Lifetime value. This message will update the registration at the MP

Type	Mesh Point Address	Old Mesh Point Address
Current Distance	User node Address	Sequence Number
Registration Lifetime		

Fig. 7.4: The fields of the *Mesh Point Registration Reply* message

.

for the specified amount of seconds. The *Handover Flag* is a special flag that a user node can use to make a fast handover to another MP. For example, when a user node that is already registered with an MP, determines that it needs to switch registration to another MP, it sets the *Handover Flag*. This flag informs the receiving MP that this is a new registration but that the sending user node was previously registered with another MP. This prompts the MP to lookup the sending user node, which it knows from the *User node Address* field, in its *Registration List*. If the address is found, the originating node is indeed registered with some other MP, and the new MP needs to notify the previous MP about the handover. The new MP therefore first sends a *User Node Handover Notification* to the old MP, and then a *Mesh Point Registration Reply* to the user node.

Type	Old Mesh Point Address	New Mesh Point Address
User node Address	Sequence Number	

Fig. 7.5: The fields of the *User Node Handover Notification* message

.

When the old MP receives the Handover Notification, it updates its *Registration List* entry with the corresponding New Mesh Point Address and Sequence Number. Any subsequent data packets received at the old MP with a destination IP address pointing to the User Node Address, will now be tunneled to the New Mesh Point Address. This will in effect enable the user node to perform a soft handover between the two MPs. Since the Handover Notification is only sent to the old MP, and not every MP in the mesh network, it may take some time before other and intermediate MPs learn about the registration. The tunneling of data packets, however, will prevent this from becoming a problem since the tunneled packets are routed directly to the new MP. After a while, the periodic *Registration List* updates will see to that the new user node location is known by every MP in the mesh network.

7.3.3 User Node Handover Determination

An important decision that a user node needs to make, is to determine when it is time to switch to a new MP. When a user node has an active communication through an MP, it will periodically send registration updates that triggers registration replies. From the *Current Distance* field of the registration reply, it can keep

track of the distance (in number of hops) to its current MP. When this distance increases by one hop, it might be a good idea to initiate a handover to a new MP. If we consider the network topology, and what event it is that actually causes the distance to the MP to increase by one; We then see that a link break has usually occurred. Some of the links on the route to the MP have broken, inserting an extra intermediate node into the route. If we consider a Manhattan topology, a likely scenario might be that the user node has rounded a corner, thereby causing the signal strength to rapidly drop, breaking the link to the next hop towards the MP. In this case it makes sense to start searching for a new MP on the new street. This is also our recommended action upon noticing the increased distance. When the distance has increased by one, the node issues a new mesh point search procedure, as described above. If a closer MP is discovered, a fast handover registration update will be issued, also as described above.

Since we are considering the case where a user node has an ongoing communication with a MP, a link break and the following increased distance will have an impact on the performance of the communication session. If we have an option available from the OS that enables us to determine the signal strength (RSSI), of the last reception from our next hop towards the MP, we may have the option to proactively start searching for a new MP. Most OSs used today provide some form of RSSI support, and considering the interest the research community has shown to these features, it may very well soon become a standard feature.

7.4 Fading and Forwarding in the Mesh Access Network

Fading is something that has a significant impact on the performance of wireless networks. This is something that is especially true when the network is located in an urban environment.

7.4.1 Fading

Fading results from the superposition of transmitted signals that have experienced differences in attenuation, delay and phase shift while traveling from the source to the receiver. It may also be caused by attenuation of a single signal. A significant result of fading is heavy fluctuations of the received signal strength.

In mobile communication we often talk about two different type of fading: **Slow Fading** and **Fast Fading**. Slow Fading is caused by the larger movements of a node and obstructions within the propagation path. For example, when a node is moving in a suburban or urban environment, buildings and trees will sometimes block the direct path between the sender and the receiving node. While the direct path is obstructed the signal strength will drop, and when the mobile node has moved past the obstruction, the signal strength will again increase. Fast fading has a more complex explanation. The reason for the fast fading signal loss is the destructive interference that multiple reflected copies of the signal make with itself.

To understand how a signal can destructively interfere with itself, consider how the sum of two complex sinus waveforms with different phases interleave.

Please consider Figure 7.6. Here we see an example of the signal variations in a rayleigh fading channel, when the mobile node is moving at 2.5 m/s. The small circlets indicate packet reception instants and we can see how large the signal variations are for this simple scenario. When we are in a deep fade it will not be possible to receive even at the lowest rate, while at good instants it will be possible to receive at the maximum rate.

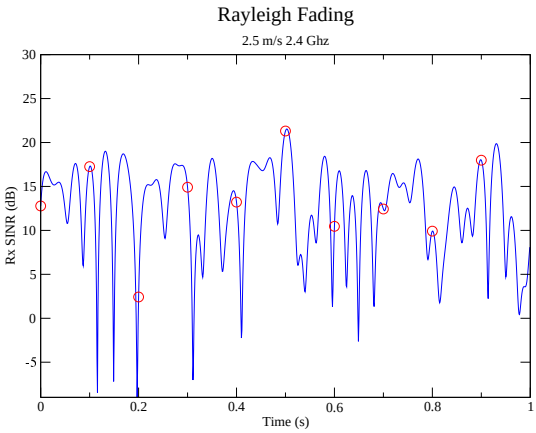


Fig. 7.6: Packet reception for a rayleigh fading channel at 2.5 m/s mobility at 2.4 GHz

7.4.2 Diversity Forwarding and Diversity MAC

In an urban wireless setting where both the environment itself as well as the mobile nodes themselves are moving, fading will affect the signal strength of every wireless link. This means that sometimes the link will be very good and sometimes very bad. In the wireless access mesh, it will often be possible for one MP to have several possible routes to another MP. This means that the MP may have several possible MPs as the next hop. If the link towards one next hop MP is currently in a bad fading state, one of the other possible next hops might be in a better fading situation. So, if we have a mechanism that allows us to evaluate different next hop candidates before the packet is actually transmitted, we could gain a lot both in terms of performance and power consumption.

7.4.2.1 Diversity MAC

We present a MAC protocol that enables diversity forwarding and allows power and interference control by querying a number of possible candidate nodes. Each candidate is a possible next hop towards the final destination as determined by the upper layer routing protocol.

In normal IEEE 802.11 DCF, a terminal may use a simple RTS-CTS cycle to inform its neighbors about an intended transmission. Nodes overhearing a RTS or CTS defer their transmissions for the duration of the scheduled transmission. In our protocol, the RTS and CTS functionality is extended to also include diversity, power and channel information.

IEEE 802.11 specifies that a number of different channels can be used. 802.11b/g specifies 3 concurrent channels while 802.11a specifies 8 concurrent channels. In order to increase the efficiency of the protocol, we also provide a simple mechanism to dynamically pick the best current channel. This means that control information is always transmitted on a predefined control channel (let us assume channel 1), while data traffic can be transmitted on any channel.

Whenever a node wants to transmit a packet, it multicasts an RTS message that includes two or more destinations. The node also includes information about the power level used when transmitting the RTS. Each destination receiving an RTS reply by sending a CTS in the order they are listed within the RTS. Just as with the RTS, the CTS also includes the power level used for transmitting the CTS. In addition to this, the CTS includes the power level *Plevel* needed for this node to successfully receive the scheduled packet. This power level is calculated based upon the *gain* as perceived by the target node, and the current noise level. It is now also possible for the target candidate to calculate the expected SINR of the requested transmission. Based on this the target picks an appropriate datarate. The datarate picked depends on the type of modulation and coding used by the different available rates. Different rates use different modulation and coding schemes with different minimum target SINR levels.

The target node also performs a multi channel carrier sense. This means that it first senses every available data channel to determine which channels that are idle. It can then randomly pick a channel among the idle ones, and specify this channel in the *channel* field of the CTS. If no data channel is idle, the control channel can be used. However, in our simulations, we use a simpler approach that is easier to implement, i.e. the data channel is mapped to the destination address, as explained below.

As each of the different channels can be regarded as independent, the estimations and calculations performed on one channel can not simply be used on an other channel. A solution to this could be to transmit a small and short wideband probe signal just prior to the first RTS message. This probe would be spreaded differently than RTS,CTS or data packets, and the receiver could have a separate receiver structure for these probes. When a RTS has been received, it could use that receiver structure to calculate an estimate of the status of wider frequency band, when can then be used for picking the appropriate channel. An other alternative is to have dedicated field within the RTS, during which the probe is transmitted. This will be discussed more in the next chapter.

When the initiating node has received all the CTS messages it is expecting, or they have timed out, it chooses an appropriate destination, tunes the radio to the indicated channel, sets the corresponding and indicated power level and sets the

data transmission rate.

An issue with this type of scheme is how to set the Network Allocation Vector (NAV). In 802.11, the NAV is set by all nodes receiving either the RTS or the CTS, and prevents these nodes from transmitting for the duration of the scheduled transmission. In our case, it is very hard to set the NAV for three reasons. The first reason is, when the source node issues the RTS, it doesn't know the duration of the upcoming transmission. The duration depends on the data rate used, which is determined by the receivers, as explained above. This means that any neighboring node overhearing the RTS, can not set the NAV as it doesn't know the duration. Secondly, as the RTS is sent to multiple candidates, several CTS responses will also be generated. Neighboring nodes that overhear an CTS, can not determine if this candidate will be chosen or not. This decision is made by the initiating transmitter, and therefore prevents neighboring nodes from correctly setting the NAV. The third implication is that, future transmitters might try to contact a node that is already transmitting on an other data channel. This node is then not able to transmit to that particular target node. The result of all this is in that in our protocol, RTS and CTS messages do not set any NAV.

While it isn't possible to set a NAV, it doesn't have to be too bad. Since the RTS may be multicasted to several candidates, maybe one of the other candidates are available. So, if one candidate is busy transmitting, some of the other candidates may still be free to accept the upcoming transmission. The requesting node has to consider the case that a failure to receive a CTS may be the result of the terminal being busy in some other transmission, maybe on a different channel. This problem of a candidate target node being busy on a different channel is commonly called the Hidden Terminal Channel Problem. If only one candidate is used, the fact that the candidate may be busy on a different channel, must also be taken into account when choosing the RTS timeout value.

Another issue here is that in normal CSMA/CA operation, as defined by 802.11, the time between consecutive RTS retries increases exponentially. A consequence of this is that if some other neighboring node wishes to transmit to the same target node, that node will initially start with a small random backoff value that depends on the default minimum Contention Window, CW. Since the first node started with a small value that is increased for every failed attempt, and the second node comes into the game at a later point in time, that node will have a smaller backoff interval. The chances for the second competing neighboring node to gain access to the target node therefore increases. In order to ease this situation a bit, consider a node that is currently in a backoff after failed a RTS. When it overhears an RTS from a neighboring node destined to the same target destination, it will reset its CW (reset its backoff timeout value to the default value). However, while this helps, it is still not enough. Some neighboring nodes close to the target, but out of transmission range of the initiator, might still be able to unfairly win in a situation like this.

To thus make the situation fairer, we therefore propose the following two approaches: (1) increase the CW only after every 3rd failed attempt. (2) freeze the CW for the first 3 attempts as in (1), but compensate for this by increasing the CW

by a power of 3 after the 3rd failed attempt.

This will make the situation fairer among terminals competing for access to a node that is currently busy transmitting on one of the data channels. While this is a very simple approach, the drawback is that the probability for real collisions will be higher. If a node experiences severe problems, or if it determines that the node and traffic density so demands, it might dynamically adjust the CW. How to dynamically adjust the backoff procedure and set the CW has been studied in several papers [20] [21] [22] [23]. However, this is not something that we currently have implemented.

7.4.3 Multiple channels and planning

Most wireless technologies available on the market today provide the ability for radios to communicate on a multiple number of different channels. The MAC protocol, we presented above, is designed to optionally use these channels as dynamically as possible. However, if a standard MAC protocol like 802.11 DCF is used, this flexibility will not be available. This doesn't mean that nodes in the network have to operate on a single channel, at least not the MPs. Since the topology in a mesh network is typically static, smart network planning can enable the use of several channels.

A simple approach, which was presented in [24], use a simple hash function to map a destination node to a specific channel. While this is not very dynamic, it is very simple and can achieve significant performance improvements. This approach works in the following way: when a node is about to transmit a packet to a destination node, it uses the hash function with the destination IP address and tunes its radio to the corresponding channel. After the transmission, it tunes the radio back to its own dedicated channel, as determined by its own IP address. We will evaluate this approach in our solution, in combination with standard routing protocols.

7.4.3.1 Diversity Routing

A very important part in order to achieve diversity forward is how we perform routing. This is important not only in terms of finding routes in the classical way, but as a very important input to the MAC layer. The routing algorithm needs to provide the MAC protocol with different candidate next hop destinations. This type of routing also puts a new and different constraint on the routing protocol. It needs to provide multiple non node-disjoint paths to a destination. In a normal link state routing protocol, single paths are setup through independent calculations of every node in the networks. Loop freedom is assured by the fact that every node uses the same algorithm with the same input information. This will in effect create a single source destination path through the network. In our case, every intermediate node in the network will make an independent choice of which next node to send the packet. This means that preventing loops will become much more

difficult and complex. One might think that source routing is a possible solution to this, but this means that only the source can take any path decisions.

One solution is to use greedy forwarding as explained in the previous chapter, where only candidates with a lower cost are evaluated.

Another possible solution though, is to setup the route, or rather the multiple non-disjoint paths, when needed, and in the process ensure that loops will not be created. One solution like this is presented in [12]. That paper presents a working protocol, but some diversity is lost in the process of ensuring loop freedom.

Another option that enable maxmimum diversity is to use some form of route recording, either at the MAC level, or at the IP level. This would enable each forwarding node to take the previous path into account when taking the forwarding decision. It would also effectively prevent loops. The drawback is of course the extra overhead, but with some intelligent address compression and address planning, the overhead can be minimized. This is the approach we currently have implemented.

7.4.3.2 Diversity forwarding

The network routing protocol has to make the decision of what candidates to propose to the MAC layer. We do this by using standard shortest path calculations (Dijkstra), but with the constraint that the forward path doesn't cross the previous path, that is, we will not allow any loops. This decision is then forwarded to the MAC protocol, which evaluates the proposed candidates in the way we explained above. The MAC layer will now decide the next hop based both on the information about the candidates it received during the RTS/CTS cycle, and path information it received from the routing protocol. Basically, what happens is that the RTS/CTS based information temporarily updates the network path costs. The reasoning is that, although if one candidate may have perfect link conditions, that candidate node may have a bad forward path. Some other candidate nodes with only average link conditions may have a perfect forward path. We could say that the two protocols, the MAC protocol and the routing protocol, operate on different time scales. The MAC protocol operates on a short time scale, while the routing protocol operates on a longer time scale. It is the combination of both these time scales that will form the basis of the forwarding decision.

We believe this sharing of information across layers is very promising. The fast query reply probing on the MAC layer will enable us to perform diversity forwarding that will be efficient in fading channels. As the environment normally is constantly changing, especially in an urban environment, fading sensitive forwarding can be really helpful. If one of the candidates is currently in a deep fade, it will either not respond, or it will respond and increase the path cost significantly. Choosing another next hop in this case would make a lot of sense.

7.4.4 User node vs Mesh node forwarding

User nodes use a slightly different type of forwarding. Firstly, they only use diversity forwarding for packets routed towards an MP. Secondly, they only query two candidates, and these two candidates must be closer in number of hops to the MP. These candidates can be determined by listening and analyzing the routing protocol control traffic. When it overhears a routing control message from a neighboring node, and determines that it is using a route to the MP the node is currently using, and that node is closer to MP, it will mark that node's address as a possible next hop candidate.

Another solution for a user node to determine possible next hop candidates, is to use information that it learns from the registration application that we described earlier. When the node forwards, or overhears a message from a closer neighboring node, and that node is also registered or is registering with the same MP, it will mark that node's address as a possible next hop candidate.

The user node can then use the diversity MAC described above. The main difference here is that only the hop count metric is used. If the hop count to the MP through the candidates is the same, it will randomly select among the ones with the highest datarate. If the datarate is the same it will randomly pick one of the available candidates.

7.5 Simulations

We have used the Qualnet simulator [25] to evaluate our proposed solutions and architectures. Qualnet is a popular commercial event driven and scalable network simulator.

In addition to the Qualnet simulator, we have developed a small IRT (Intelligent Ray Tracing) [26] tool, that takes as input maps defined in Qualnet and outputs a tile based propagation matrix. The raytracing tool considers diffraction around corners and we use up to four reflections to calculate the multi path propagation, although more reflections can be specified if needed. The IRT technique divids the simulation area and all defined walls into a number of tiles. The propagation path loss from between every source destination tile pair is then calculated and put into a matrix. For points within each tile we then use bilinear interpolation in order to get a better approximation within each tile and a higher resolution. The type of map we are considering is a Manhattan city grid, as described in the next subsection.

7.5.1 Simulation Setup and Scenarios

We will simulate the four network architectures we described earlier in section 7.2. Each network type will be simulated in a 525 x 525m urban city area, consisting of 100 nodes. The city is modeled according to the well known Manhattan city topology model [27]. For the first two network types, the node will consist of 100 user nodes. For the second network type, only 75 of the user nodes will be mobile,

while the other 25 are static and placed at different intersections. The third and fourth network types consist of 75 user nodes and 25 mesh points. The fourth ad hoc mesh network consists of 9 mesh access points, with 16 mesh relay points. A mesh relay point only forwards mesh traffic, and operates on a different channel than is used by the user nodes.

Every node in the network will use 802.11g at the physical layer. The MAC protocol and routing protocol, will differ between the different scenarios as explained below.

While this simulation area might be smaller than what is normally used in ad hoc network simulation studies, it should be noted that the urban simulation area consists of 16 city blocks, where each block is a square of 100m, intersected by 25m wide streets. This area will produce a topology with an average hop-count of about 6-7 hops, which is significantly longer than what is normally used in ad hoc simulation studies. The suburban environment consist of the same city topology, but here a block consists of many smaller and lower houses. Mesh Points as used in the mesh topologies are placed above these houses, while user nodes are placed below. This means that for the suburban environment it might be possible for two user nodes to find a connecting signal path through a city block. This is not possible in the urban environment, as the walls around the city block completely blocks the signal path, although the signal might be reflected or refracted around corners and thus eventually reach an other user node.

The power consumption of each individual node is measured during each simulation. Different power values are used depending on whether the node is transmitting, receiving, sensing or if it is idle. The power values of these different modes are modeled after the Cisco Aironet 802.11 a/b/g chip [28].

In the first network traffic type we are simulating are a varying number of bidirectional 56 kbps CBR traffic sources, modeled after the G.711 [29] codec.

A technique that can be used to predict user satisfaction of a conversational speech quality is the ITU-T E-model. The E-Model is standardized by the ITU as G.107 [30], and is a tool that can estimate the end-to-end voice quality, taking the IP telephony parameters and impairments into account. This method combines individual impairments (loss, delay, echo, codec type, noise, etc.) due to both the signal's properties and the network characteristics into a single R-rating. This method can be used as a good quality of service measure for VoIP calls that consider a user's opinion about the service. The Mean Opinion Score (MOS) is a method recommended by the ITU and the IEEE 802.20 group [31] to measure speech quality. In this method, the users rate the call quality in a range varying from 1 (bad) to 5 (excellent). We will apply this rating to both the G.711, and the enhanced G.711 [32], codecs.

In these cases we are simulating a constant random load of 5, 10, 15, 20 and 25 source destination traffic pairs. By constantly random we mean, that the traffic load on the network is constant, but that the source destination pairs are constantly changing. Each CBR flow is one minute long, and when one flow ends, a new flow will instantly start between a new source destination pair somewhere else in

the network. These simulations are running for a total of 60 minutes. These traffic scenarios are designed to roughly model voice communications between two voice capable devices, when the network is loaded by various amount of traffic.

For each simulation scenario and network type, user nodes will move according to the pedestrian mobility model defined in Qualnet 4.0. Here, user nodes move along the streets to a randomly chosen street corner somewhere on the city map. When the nodes arrive at their destination, they will randomly chose a new destination somewhere on the map. Every time they arrive at a street corner, they run a 50 % chance of having to wait for a red light before proceeding across the street. Each pedestrian user node will move at a constant speed uniformly choosen between 1.5 and 2.5 m/s.

Radio signals will be affected by multi path fading according to the Ricean fading model, with a k-factor of either 0 or 1, depending on whether a direct line of sight between each pair of hops are available or not. The multi-paths and attenuations are calculated by the IRT tool.

7.5.2 Pure Ad Hoc Network Simulations

Here all user nodes are mobile as described above. The first simulation setup consists of user nodes running standard 802.11g DCF, with AODV as the routing protocol, on a single channel. The second setup we are considering is the same as the first, but DIVR is used as the routing protocol, and our DIVM MAC protocol.

7.5.3 Supported Ad Hoc Networks Simulations

These simulations are exactly the same as those described in section 7.5.2. The only difference here is that 25 nodes will be static, and strategically placed in intersections.

7.5.4 Single Hop Mesh Network Simulations

In these setups, all user nodes will run the registration application defined in section 7.3.2. Each user node will have access to an MP within a single hop.

Within the mesh, we will use AODV or our DIVR routing, with all mesh nodes having two different interfaces running either IEEE 802.11g DCF and/or our DIVM MAC protocol. Every mesh node has at least one interface using 802.11g DCF, which is used for communication with user nodes.

Mesh nodes, use the simple address mapped multi channel scheme [24]. No coordination whatsoever is done by a 802.11 source before it chooses a channel, it depends purely on the address of target node. The channel is chosen as channel=(address) mod (number of channels).

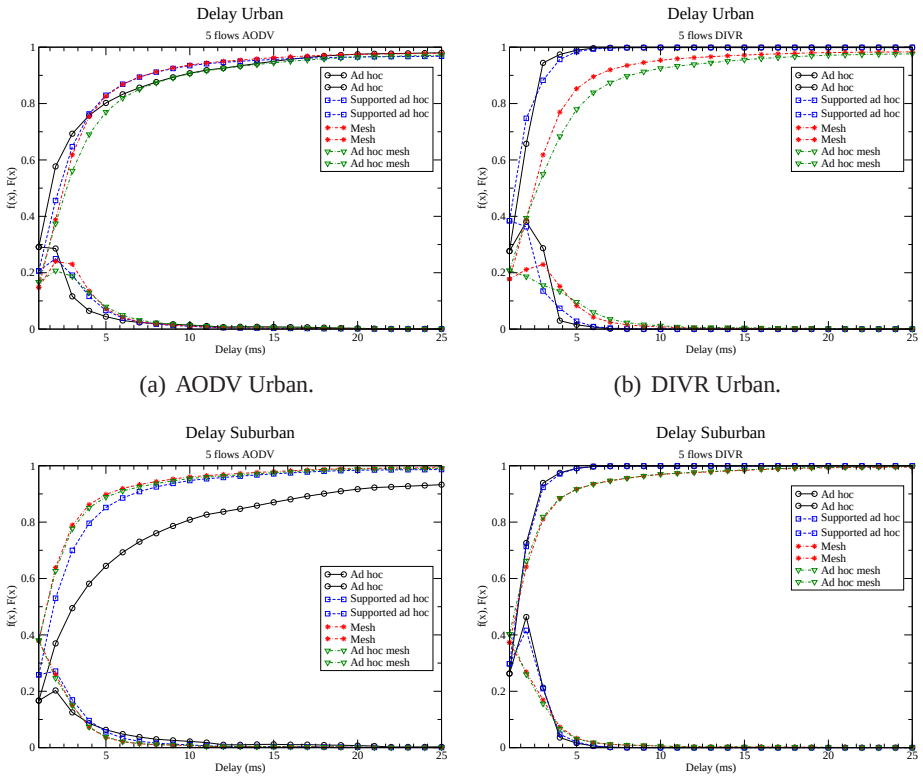
Contrary to the two ad hoc scenarios, all user nodes are equipped with a single 802.11g interface. Since the user nodes connect directly to the MPs, no dynamic routing protocol needs to be running on these interfaces.

7.5.5 Mesh Ad Hoc Network Simulations

This setup considers a more sparsely deployed mesh network where user nodes may need to connect to the MP over several hops. Mesh access points are configured in the same way, and with the same protocol types, as in the single hop case. Mesh relay points lacks the 802.11g DCF interface used to communicate with user nodes.

User nodes on the other hand, still have a single wireless interface, but are now also running the AODV routing protocol. Otherwise, they are configured as in the previous case.

7.5.6 Simulation Results



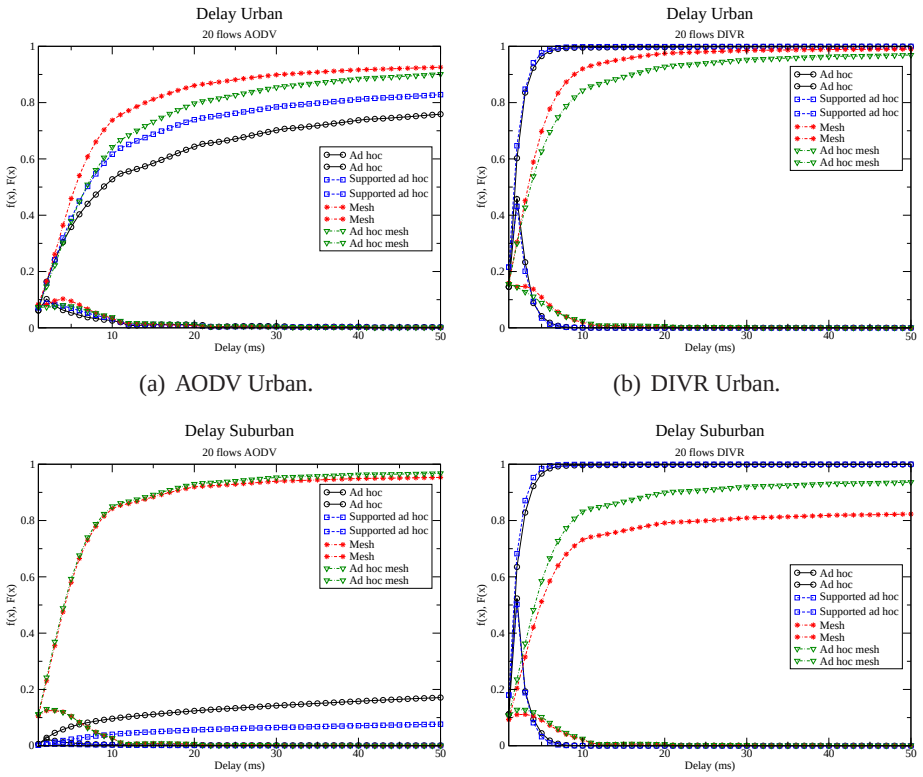
(a) AODV Urban. (b) DIVR Urban.

(c) AODV Suburban. (d) DIVR Suburban.

Fig. 7.7: Delay distributions for 5 flows

7.5.7 Discussion

First let us consider figure 7.7 that illustrates the delay distributions when we have 5 bidirectional 64 kbps UDP flows. The most obvious difference we can observe



(a) AODV Urban. (b) DIVR Urban.

(c) AODV Suburban. (d) DIVR Suburban.

Fig. 7.8: Delay distributions for 20 flows

here is the significantly lower delays for the DIVR ad hoc scenarios. In fact, here the cumulative probability starts approaching one for packet delays at around 6-7 ms. If we now also look at the 2nd and 4th tables in table 7.1 and table 7.2, we can see that the average delay for DIVR is around 2 ms independent of the amount of traffic, and whether the environment is urban or suburban. So, the conclusion here is that for the ad hoc scenarios, the delay DIVR is fairly independent of the type of environment and the amount of traffic.

If we consider figure 7.7(c) we can see that the delay for AODV (suburban) especially for the pure ad hoc case is significantly higher than for any of the scenarios and environments for 5 bidirectional flows. A look at table 7.2 reveals that the average delay for this case is 16.7 ms. For the AODV urban ad hoc case, the delay is (see table 7.1) 6.2 ms. With a higher traffic load, see figure 7.8 , table 7.2 and 7.1, the effect is much more severe with a delay of 195ms and 640ms for 25 AODV urban and suburban flows.

What we see here is the effect of the environment itself, how the height of the buildings affects the signal path and the performance. In a city environment, building walls completely block the signal from one parallel street to another, while in

AODV URBAN Ad hoc

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.809 (0.772, 0.846)	6.18	35.6	0.987	2.56	3.97	107
10	0.784 (0.780, 0.788)	14.2	44.1	0.965	2.41	3.91	85.5
15	0.705 (0.671, 0.738)	40.1	70.7	0.891	2.04	3.72	66.8
20	0.564 (0.493, 0.634)	89.3	126	0.758	1.59	3.35	39.2
25	0.378 (0.314, 0.442)	195	216	0.545	1.00	2.13	35.9

DIVR URBAN Ad hoc

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.676 (0.634, 0.717)	1.98	48.6	0.999	1.93	3.64	46.0
10	0.658 (0.625, 0.692)	2.11	52.2	0.998	1.87	3.60	35.0
15	0.631 (0.589, 0.674)	2.25	59.2	0.998	1.78	3.53	27.8
20	0.611 (0.580, 0.642)	2.13	65.1	0.998	1.72	3.48	24.6
25	0.586 (0.520, 0.653)	2.70	68.5	0.998	1.65	3.41	21.9

AODV URBAN Supported ad hoc

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.832 (0.758, 0.907)	12.3	38.4	0.969	2.71	4.02	58.2
10	0.791 (0.760, 0.821)	20.1	51.0	0.947	2.45	3.93	26.3
15	0.750 (0.679, 0.821)	27.2	64.9	0.928	2.23	3.83	18.6
20	0.609 (0.579, 0.639)	63.7	115	0.828	1.71	3.47	12.5
25	0.543 (0.524, 0.562)	100	146	0.719	1.54	3.30	10.3

DIVR URBAN Supported ad hoc

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.789 (0.743, 0.835)	1.6	25.9	1.000	2.44	3.92	47.0
10	0.712 (0.643, 0.780)	2.1	34.6	0.999	2.07	3.73	33.6
15	0.716 (0.629, 0.802)	2.0	42.4	0.999	2.08	3.74	27.4
20	0.695 (0.662, 0.729)	2.0	39.2	1.000	2.00	3.69	22.7
25	0.615 (0.512, 0.719)	2.1	51.9	1.000	1.73	3.49	21.0

Tab. 7.1: URBAN ad hoc scenarios. Delivery ratio with standard deviation. Delay in ms (d), Jitter in ms (J), Probability of delay less than 50ms (P), Mean Opinion Score for G.711 (MOS), MOS for enhanced G.711 (MOS2), Battery lifetime in hours for 1200mAh (B)

the simulated suburban environment the signal is not completely blocked but is still affected by multi-path fading. The main difference this has on the MAC layer is how carrier sensing are affected and hidden terminals are created. In the suburban environment, carrier sensing is possible across a block, but not in the urban environment. In the suburban environment, RTS and CTS packets will protect a 802.11 transmission from parallel transmitters, which increases the time it takes for a packet to access the channel. But since some links now experience non line of sight multi-path propagation fewer, packets will also be delivered on average.

For the supported ad hoc scenarios, see table 7.1 and 7.2. When a relay mesh node is placed at each intersection, the delay increases for low AODV traffic in the urban environment, while it decreases for the suburban environment. An important factor here is the connectivity and medium contention on the routes. In the urban environment, the best path across a few blocks will always pass through a relay

AODV SUBURBAN Ad hoc

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.746 (0.730, 0.762)	16.7	52.6	0.959	2.22	3.82	18.3
10	0.532 (0.495, 0.570)	122	140	0.646	1.45	3.27	9.0
15	0.263 (0.244, 0.283)	433	289	0.248	1.00	2.57	7.8
20	0.235 (0.227, 0.242)	583	325	0.171	1.00	2.49	7.6
25	0.210 (0.206, 0.215)	640	346	0.133	1.00	2.44	7.5

DIVR SUBURBAN Ad hoc

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.543 (0.473, 0.613)	1.7	96.5	0.999	1.54	3.30	49.6
10	0.539 (0.484, 0.594)	1.9	91.4	0.999	1.53	3.29	35.1
15	0.501 (0.468, 0.534)	2.1	104	0.999	1.45	3.19	27.3
20	0.497 (0.448, 0.547)	2.2	108	0.999	1.45	3.18	22.8
25	0.484 (0.472, 0.495)	2.3	110	0.999	1.42	3.14	20.6

AODV SUBURBAN Supported ad hoc

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.908 (0.890, 0.926)	5.4	19.3	0.990	3.32	4.18	31.9
10	0.911 (0.897, 0.925)	7.7	21.4	0.989	3.34	4.19	15.5
15	0.709 (0.589, 0.830)	218	110	0.582	1.57	3.73	7.5
20	0.393 (0.373, 0.413)	764	234	0.077	1.00	2.90	6.4
25	0.340 (0.328, 0.351)	815	243	0.048	1.00	2.76	6.4

DIVR SUBURBAN Supported ad hoc

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.832 (0.802, 0.862)	1.6	21.7	1.000	2.71	4.02	57.5
10	0.798 (0.742, 0.853)	1.7	32.7	0.999	2.49	3.94	37.7
15	0.808 (0.762, 0.854)	1.7	30.4	1.000	2.55	3.97	29.2
20	0.799 (0.767, 0.831)	1.8	35.9	1.000	2.50	3.95	23.5
25	0.739 (0.715, 0.763)	2.0	43.9	1.000	2.18	3.80	20.7

Tab. 7.2: SUBURBAN Ad hoc scenarios. Delivery ratio with standard deviation. Delay in ms (d), Jitter in ms (J), Probability of delay less than 50ms (P), Mean Opinion Score for G.711 (MOS), MOS for enhanced G.711 (MOS2), Battery lifetime in hours for 1200mAh (B)

point, which increases the contention for those nodes and therefore the delay. In the suburban environment they increase the connectivity of the network, but all routes doesn't necessarily pass through them, resulting in a lower delay. With very high traffic, it is more complicated. Now links, or routes, may be reported as broken due to collisions, when in fact they are not. When the route is then repaired and resetup, RREQ packets are rebroadcasted by neighboring nodes. With a higher connectivity, more packets will be rebroadcasted, increasing the probability for collisions, and the delay. In the urban environment, the rebroadcasting collision effect is not high enough to decrease performance and overcome the positive effect of the higher connectivity. The delay is thus lower for the urban environment than for the suburban environment.

If we look at the results for mesh scenarios, table 7.4 and 7.3, the biggest difference between the urban and suburban environments are the much higher suburban

AODV URBAN Mesh

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.750 (0.713, 0.788)	8.9	48.8	0.983	2.24	3.83	25.4
10	0.720 (0.640, 0.800)	10.2	58.3	0.982	2.10	3.76	21.1
15	0.731 (0.719, 0.743)	15.0	57.0	0.969	2.15	3.78	17.9
20	0.699 (0.675, 0.723)	27.1	71.0	0.926	2.01	3.70	15.7
25	0.652 (0.622, 0.683)	58.1	89.4	0.824	1.84	3.58	13.8

DIVR URBAN Mesh

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.662 (0.623 0.701)	8.1	65.6	0.987	1.88	3.61	25.8
10	0.653 (0.598 0.708)	7.4	73.5	0.989	1.85	3.58	20.5
15	0.530 (0.502 0.558)	7.4	90.1	0.988	1.51	3.26	18.5
20	0.527 (0.469 0.584)	7.0	88.5	0.990	1.51	3.25	16.4
25	0.509 (0.459 0.560)	9.3	95.2	0.984	1.47	3.21	14.4

AODV URBAN Ad hoc mesh

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.702 (0.663, 0.742)	9.7	58.6	0.981	2.03	3.71	23.2
10	0.701 (0.665, 0.736)	12.9	63.9	0.974	2.02	3.71	19.1
15	0.688 (0.654, 0.722)	17.4	70.4	0.961	1.97	3.67	15.7
20	0.665 (0.622, 0.708)	33.3	80.1	0.901	1.89	3.62	13.8
25	0.630 (0.586, 0.674)	76.3	97.1	0.792	1.77	3.52	12.5

DIVR URBAN Ad hoc mesh

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.599 (0.545, 0.654)	8.8	83.9	0.984	1.68	3.44	22.5
10	0.605 (0.536, 0.674)	11.7	82.5	0.974	1.70	3.46	19.9
15	0.505 (0.477, 0.534)	13.1	109	0.971	1.46	3.20	16.4
20	0.484 (0.433, 0.534)	12.9	105	0.969	1.42	3.14	14.8
25	0.487 (0.454, 0.520)	14.6	112	0.960	1.43	3.15	13.3

Tab. 7.3: URBAN Mesh scenarios. Delivery ratio with standard deviation. Delay in ms (d), Jitter in ms (J), Probability of delay less than 50ms (P), Mean Opinion Score for G.711 (MOS), MOS for enhanced G.711 (MOS2), Battery lifetime in hours for 1200mAh (B)

delivery ratios. For low traffic rates, both AODV and DIVR manages to sustain high delivery ratios in the suburban environment, but AODV more or less maintains these ratios for a higher number of flows than DIVR.

For the urban environments, the delivery ratio isn't very high, the maximum delivery ratio for any urban mesh scenario is 75%. In fact, if we compare table 7.3 and 7.1 we see that for low traffic rates, the ad hoc scenarios actually perform better, which is not true for suburban (tables 7.4 and 7.2). We can now also take a look at the different Mean Opinion Scores. To at least have some form of acceptable VOIP experience in an urban environment, the enhanced G.711 codec should be used. We can also make the interesting conclusion, that for urban environments, it is actually better to use the ad hoc technologies, with the supported ad hoc network performing slightly better. If we also look at the battery lifetimes, we see that the lifetimes are significantly longer for the ad hoc scenarios. This is an interesting, and

AODV SUBURBAN Mesh

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.948 (0.927, 0.969)	3.2	11.9	0.996	3.73	4.26	19.9
10	0.947 (0.936, 0.958)	4.8	13.0	0.993	3.72	4.26	14.2
15	0.945 (0.921, 0.970)	18.7	17.3	0.968	3.70	4.25	10.7
20	0.917 (0.739, 1.000)	19.8	27.7	0.953	3.40	4.20	8.1
25	0.658 (0.359, 0.957)	232	107	0.452	1.86	3.60	6.5

DIVR SUBURBAN Mesh

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.943 (0.924, 0.962)	2.4	11.1	0.999	3.67	4.25	18.0
10	0.933 (0.919, 0.947)	3.6	13.3	0.997	3.57	4.23	13.3
15	0.874 (0.658, 1.000)	18.1	25.8	0.963	3.02	4.11	9.7
20	0.787 (0.618, 0.956)	62.5	73.5	0.823	2.43	3.92	7.1
25	0.460 (0.276, 0.645)	388	181	0.260	1.38	3.08	5.7

AODV SUBURBAN Ad hoc Mesh

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.953 (0.939, 0.966)	3.9	11.2	0.994	3.79	4.27	20.3
10	0.946 (0.936, 0.957)	5.4	13.2	0.993	3.71	4.25	14.5
15	0.948 (0.940, 0.956)	6.4	14.2	0.992	3.73	4.26	11.3
20	0.904 (0.750, 1.000)	16.8	28.2	0.968	3.28	4.17	8.1
25	0.417 (0.223, 0.611)	120	115	0.744	1.31	2.96	5.8

DIVR SUBURBAN Ad hoc Mesh

Flows	*Delivery*	*d*	*J*	*P*	*MOS*	*MOS2*	*B*
5	0.945 (0.928, 0.962)	2.3	10.4	1.000	3.70	4.25	19.0
10	0.929 (0.904, 0.954)	3.9	13.3	0.996	3.52	4.22	13.5
15	0.911 (0.755, 1.000)	9.9	20.1	0.980	3.34	4.19	9.9
20	0.813 (0.649, 0.977)	29.4	43.2	0.936	2.58	3.98	7.2
25	0.394 (0.200, 0.588)	134	120	0.713	1.28	2.90	5.8

Tab. 7.4: SUBURBAN Mesh scenarios. Delivery ratio with standard deviation. Delay in ms (d), Jitter in ms (J), Probability of delay less than 50ms (P), Mean Opinion Score for G.711 (MOS), MOS for enhanced G.711 (MOS2), Battery lifetime in hours for 1200mAh (B)

somewhat unexpected result. Even though the mesh network operate on separate channels than the user nodes, we don't really gain anything by using their extra interfaces in a harsh urban environment. With a different mesh configuration, and by using more interfaces in each mesh point, or a different physical layer with a higher capacity, this will probably change. But for this configuration, with the same physical layer on both user nodes and mesh nodes, we can't see any significant gain for urban environments. We leave it for future work to study the needed capacity, and the various dependent factors, for a mesh network to outperform an ad hoc network in an urban environment.

Continuing looking at battery lifetimes, for the ad hoc scenarios we see that DIVR achieve its longest for the suburban environment. For AODV however, the longest lifetimes are achieved for the urban environment.

So, when looking at all the tables, we can see that the best VOIP MOS perfor-

mances are for suburban mesh scenarios. If we limit the number of flows to 20, we can still achieve very high VOIP performance by using an ad hoc mesh topology instead of using a single hop mesh. We can do this by maintaining the same battery lifetime, and with a cheaper infrastructure.

The MOS VOIP performance in the urban scenarios are always a bit lower than for suburban. With the consideration of the higher lifetime, in combination with comparable MOS performance, it seems better to use a supported ad hoc network for urban environments.

In conclusion we can say that what type of protocol that is optimal for a certain situation, depends on the environment, the type of network and the amount of traffic.

7.6 Future work

As future work other routing protocols such as OLSR should be considered for routing within the mesh as well as the ad hoc networks. The main reason for OLSR not being part of this study is CPU processing power. The simulation time for OLSR in Qualnet is significantly longer than for the other protocols, whose simulation time is also very long. We would also like to study the effect other routing metrics besides hop count has on the performance. Finally, we are working on a new solution that significantly increases the number of delivered packets for the DIVM mac protocols.

7.7 Conclusion

We have described a few different type of wireless ad hoc and mesh networks, and how they can be designed to operate efficiently in urban and suburban city environments. We analyzed how well these networks perform with standard protocols, and with our newly proposed protocols. We saw that our new protocols deliver packets with a significantly lower delay, although at the price of a somewhat lower delivery ratio. A single hop mesh network can operate with mesh access points set in association mode, enabling user nodes to connect with standard IEEE 802.11 devices and software. By instead operating the mesh access points in ad hoc mode, we enable more flexibility and functionality to be defined in the software of both the user nodes and the mesh nodes. Our simulations show the urban environment has significantly lower performance than a suburban environment. We also see that for a suburban environment it is better to use a mesh type of network, while in urban environments, an ad hoc type of network is more beneficial. What type of network configuration to choose in the different environments, depends on what protocols we are using.

BIBLIOGRAPHY

[1] S. Ganguly, V. Navda, K. Kim, A. Kashyap, D. Niculescu, R. Izmailov, S. Hong, and S. R. Das. Performance optimizations for deploying voip services in mesh networks. *IEEE Journal on Selected Areas in Communications (JSAC)*, 24(11):2137–2158, November 2006.

[2] Krishna N. Ramachandran, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Milind M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *Proceedings of Infocom 2006, Barcelona, Spain*, April 2006.

[3] J. Mo, H-S.W. So, and J. Walrand. Comparison of multichannel mac protocols. In *Proceedings of the 8-th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Montreal, Qc, Canada*, April 2005.

[4] I. Wormsbecker and C. Williamson. On channel selection strategies for multi-channel mac protocols in wireless ad hoc networks. In *Proceedings of the 2nd IEEE International Conference on Wireless and Mobile Computing, Networking, and Communications (WiMob), Montreal, Qc, Canada*, June 2006.

[5] D. Zheng and J. Zhang. Protocol design and performance analysis of frequency-agile multi-channel medium access control. *IEEE Transactions on Wireless Communications*, 5(10):2887–2895, October 2006.

[6] T. Tang, K. Mandke, C-B. Chae, R.W. Heath Jr, and S. Nettles. Multichannel feedback in ofdm ad hoc networks. In *Proceedings of the International Workshop on Wireless Ad-hoc Networks, New York, New York, USA*, June 2006.

[7] N. Jain, S.R. Das, and A. Nasipuri. A multichannel mac protocol with receiver-based channel selection for multihop wireless networks. In *Proceedings of the 9th International Conference on Computer C ommunications and Networks (IC3N), Phoenix, Arizona, USA*, October 2001.

[8] Asis Nasipuri and Jai Mondhe. Multi-channel mac with dynamic channel selection for ad hoc networks. Technical report, January 2004.

[9] J.A. Patel, H. Luo, and I. Gupta. A cross-layer architecture to exploit multi-channel diversity with a single transceiver. In *Proceedings of Infocom 2007, Anchorage, Alaska*, May 2007.

[10] P. Larsson. Selection diversity forwarding in a multihop packet radio network with fading channel and capture. *ACM SIGMOBILE Mobile Comnputing and Communication Review*, 5(4):47–54, 2001.

[11] P. Larsson and N. Johansson. Multiuser diversity forwarding in multihop packet radio networks. In *Proceedings of the 2005 IEEE Wireless Communications and Networking Conference*, volume 4, pages 2188– 2194, March 2005.

[12] A. Nilsson, P. Johansson, and U. Körner. Cross layer routing and medium access control with channel dependant forwarding in wireless ad hoc networks. *Lecture Notes in Computer Science, Volume 4396, Wireless Systems and Mobility in Next Generation Internet, Springer Verlag*, 4396, January 2007.

[13] M. Tope, J.C. McEachen, and A.C. Kinney. Ad-hoc network routing using co-operative diversity. In *Proceedings of the 20th International Conference on Advanced Information Networking and Applications AINA'06*, volume 1, pages 55–60, March? 2006.

[14] Shweta Jain and Samir R. Das. Exploiting path diversity in the link layer in wireless ad hoc networks. In *Proceedings of the 6th IEEE WoWMoM Symposium, Taormina, Italy*, June 2005.

[15] J. Wang, H. Zhai, Y. Fang, and M. C. Yuang. Opportunistic media access control and rate adaptation for wireless ad hoc networks. In *Proceedings of the IEEE Communications Conference (ICC'04), Paris, France*, June 2004.

[16] M. Park, J.G. Andrews, and S. Nettles. Wireless channel-aware ad hoc cross-layer protocol with multi-route path selection diversity. In *Proceedings of the IEEE Vehicular Techology Conference (VTC-03-Fall), Orlando, Florida*, October 2003.

[17] M. Souryal and N. Moayeri. Channel-adaptive relaying in mobile ad hoc networks with fading. In *Proceedings of the IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, California*, pages 142–152, September 2005.

[18] Jia Liu and A. Annamalai. Channel-aware routing protocol for ad hoc networks: Generalized multiple-route path selection diversity. In *Proceedings of the IEEE Vehicular Techology Conference (VTC-05-Fall), Dallas, Texas*, October 2005.

[19] J. Ai, A.A. Abouzeid, and Z. Ye. Cross-layer optimal decision policies for spatial diversity forwarding in wireless ad hoc networks. In *Proceeding of Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), Vancouver, Canada*, October 2006.

[20] F. Cali and M. Conti. Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit. *ACM/IEEE Transactions on Networking*, 8(6):785–799, December 2000.

[21] G. Bianchi and I. Tinnirello. Kalman filter estimation of the number of competing terminals in an ieee 802.11 network. In *Proceedings of Infocom 2003, San Francisco, CA, USA*, March 2003.

[22] L. Bononi, M. Conti, and E. Gregori. Design and performance evaluation of an asymptotically optimal backoff algorithm for ieee 802.11 wireless lans. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HiCss-33), Maui, Hawaii, USA*, January 2000.

[23] J-S. Liu and C-H. R. Lin. Performance improvements with a p-persistent enhanced dcf for wlans. In *Proceedings of the IEEE Vehicular Technology Conference 2006, VTC Spring 2006, Melbourne, Australia*, May 2006.

[24] S.L. Wu, C.Y. Lin, Y.C. Tseng, and J.P. Sheu. A new multi-channel mac protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *Proceedings of the 2000 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '00), Dallas, Texas, USA*, December 2000.

[25] Scalable Networks. Qualnet network simulator, version 4.0. 2006.

[26] G. Wolfle, B. Gschwendtner, and F. Landstorfer. Intelligent ray tracing - a new approach for field strength prediction in microcells. In *Proceedings of the IEEE Vehicular Technology Conference 1997, VTC 1997, Phoenix, AZ, USA*, May 1997.

[27] J. Caffery and G. L. Stuber. Subscriber location in cdma cellular networks. In *Proceedings of the IEEE Vehicular Technology Conference 1998, VTC 1998, Ottawa, Canda*, May 1998.

[28] Cisco. *Cisco Aironet 802.11a/b/g Wireless CardBus Adapter*. Cisco, http://www.cisco.com/en/US/products/hw/wireless/ps4555/products_data_sheet09186a00801ebc29.html, 2007.

[29] ITU-T Rec. G.711. *Pulse code modulation (PCM) of voice frequencies*. International Telecommunications Union, ITU, November 1988.

[30] ITU-T Rec. G.107. *The E-Model, a computational model for use in transmission planning*. International Telecommunications Union, ITU, March 2003.

[31] 802.20. *MBWA Call for Contributions for Evaluation Criteria for VoIP application, MOS-VOIPC802.20-05-51*. IEEE 802.20 Working Group on Mobile Broadband Wireless Access, The Institute of Electrical and Electronics Engineers, New York, 2005.

[32] GIPS. *Enhanced G.711 with GIPS NetEQ*. Global IP Solutions, http://www.gipscorp.com/files/english/datasheets/EG711.pdf, 2007.

8. CHAPTER VIII

A cross layer protocol with hybrid multi channel CDMA/OFDMA and diversity forwarding

8.1 Introduction: CDMA and spread spectrum multiple access

Code division multiple access (CDMA) is a form of multiplexing method to enable multiple access to a physical medium such as a radio channel. In contrast to CSMA techniques as used in for example 802.11 [1], and Ethernet, several users can use the medium at the same time thanks to the use of different coding sequences.

CDMA uses a technique called spread spectrum, which basically means that the signal is spread so that it occupies a bandwidth much greater than that which is necessary to send the information. This results in the signal being much less sensitive to interference. The bandwidth is spread by means of a code which is independent of the data that is to transmitted. The independence of the code distinguishes this from standard modulation schemes in which the data modulation will always spread the spectrum somewhat. The receiver of the signal then synchronizes by using the code to recover the data of the spreaded signal. The use of an independent code and synchronous reception allows multiple users to access the same frequency band at the same time.

In order to protect the signal, the code used is pseudo-random. It appears to be random, but is actually deterministic, so that the receiver can reconstruct the code for synchronous detection. This pseudo-random code is commonly called pseudo-noise (PN) code, or CDMA code.

There are several types of spread spectrum systems, and CDMA systems uses a method called Direct Sequence Spread Spectrum (DSSS). Here, the digital data is directly coded at a much higher frequency than the signal and the bits themselves. The code is generated pseudo-randomly, the receiver knows how to generate the same code, and correlates the received signal with that code to extract the data.

Since it is not mathematically possible to create signature sequences that are orthogonal for arbitrarily random starting points, unique ”pseudo-random” or ”pseudo-noise” (PN) sequences are used in Asynchronous CDMA systems. These PN sequences are statistically uncorrelated, and the sum of a large number of PN sequences results in Multiple Access Interference (MAI) that is approximated by a Gaussian noise process (following the ”central limit theorem”). If the signals from all of the users are received with the same power level, then the variance (e.g., the noise power) of the MAI increases in direct proportion to the number of users.

In wireless systems, the *near-far* effect is a very important problem that needs to be taken into account. The near far effect means that a receiving mobile node very close to another transmitting mobile node, might be drowned by noise caused by a transmitter communicating with a mobile node much further away. This problem is in cellular networks solved by different optimized power control algorithms, and the decreasing of the actual geographical cell size if needed.

In ad hoc or mesh network, this near-far problem is much more difficult to solve, because there is no central entity that can perform power control. Several CMDA protocols for ad hoc networks have been proposed [2] [3] [4], that with appropriate code assignment and spreading code schemes are guaranteed to be free of primary collisions. Generally, these protocols are based on random channel access, whereby a mobile node with a packet to transmit can proceed immediately with its transmission, possibly after RTS/CTS exchange. While these protocols are free from primary collisions, they are still subject to secondary collisions caused by MAI from two or more transmissions that use different codes, that is, the *near-far* problem.

A few solutions to the *near-far* problem in ad hoc networks have recently been presented. In [5] a CDMA MAC protocol is presented that uses channel gain information overheard through RTS and CTS on a common control channel, to perform power control that prevent MAI at receiving nodes to cause secondary collisions. The authors in [6], solve the *near-far* problem by proposing a dynamic clustering algorithm, that creates a cell-type structure. An iterative power control scheme similar to that of cellular networks are then used.

8.2 OFDM and Dynamic Channel Adaptation

Orthogonal Frequency Division Multiplexing, OFDM is a widely used modulation scheme that for example is used in 802.11a/g. OFDM works by dividing a wideband channel into a larger number of sub-channels. By placing a subcarrier in each sub-channel, each subcarrier may be modulated separately depending on the SNR characteristics in that particular narrow portion of the channel. As the channel varies over time, further adaptations can be made on each subcarrier in order to continually optimize the data capacity of the channel.

Much research has lately gone into developing techniques to increases the achieved capacity of OFDM systems. This includes for example adaptive modulation, subcarrier power allocation and different coding techniques. The basic idea of these methods is to differentiate between good and bad subcarriers in such a way that the data capacity is maximized. This approach is often called Waterfilling, where more power and higher order modulation is put onto subcarriers with larger SNRs, while lower SNR subcarriers will receive less power and lower order modulation up to a certain threshold after which the subcarrier is not used at all.

In [7] it is suggested that the RTS and CTS handshake can be used to achieve fast link adaptation for OFDM systems. A receiving station measures channel

characteristics when receiving the RTS frame and calculates the appropriate bit and power-loading allocation. The results of this calculation is transferred back to the receiver in the CTS message. The transmitter will subsequently modulate subcarriers in accordance with the received modulation parameters. Their simulations indicate that for higher SNRs the increase in throughput is as much as 10 Mbps compared to when standard adaptive bit and power loading schemes are applied. This is further specified in [8] where a method is described to insert a wideband training probe into the RTS message in order to receive the needed channel state information. In [9] it is specified how the RTS and CTS can provide channel information used to achieve a dynamic fast adaptive Waterfilling approach.

This chapter describes a solution that achieves fast OFDM link adaptation through a RTS and CTS handshake, in combination with a simple CDMA code assignment scheme that in addition is able to achieve diversity forwarding.

8.3 CDMA-codes and address hashing

In [10], a dynamic and distributed code assignment protocol is presented. Each code is here defined as a unique channel, and in order to agree on a data transmission code, RTS and CTS messages are exchanged. The code to be used by a transmitter is randomly chosen from a list of available code, and included in the RTS. If the code is not available, an out of band *busy* tone is transmitted.

The system and method presented in this chapter, includes a pool of predefined and orthogonal CDMA-codes, as used in spread spectrum systems as described above. The number of codes should be as large as possible in order to allow many simultaneous transmitters, but a trade-off might exist between the number of codes, the orthogonality and the resulting noise level.

The system presented specifies a common and predefined hash function, that maps a specific node address, or node address pair, to a specific CDMA-code. This means that the address used to identify a specific node, or the addresses used to identify a link, directly maps to a specific code. When a node has data that it wishes to transmit to another nearby node, it uses the hash function to determine a code that will be used to encode and spread the data signal. The protocol does not rely upon the use of busy tones, but uses network topology information to determine if a primary collision is possible.

The address used by the system in the hash function can be any address that identifies a certain node, but using a network address has several advantages. The first advantage is that network addresses are generally not assigned randomly. Normally, some administration is involved when a node receives a network address, whether it be automated by a protocol, or manually assigned by a human administrator. As the hash-function is also assumed to be known by the authority that assigns addresses, it will allow the system to take into account already assigned addresses in order to minimize code collisions. Secondly, the network itself might be aware of where and when collisions might occur through the topology informa-

tion. For example, nodes in the network might know the address of other nodes in its local neighborhood, that may have been provided by a routing protocol or some other application.

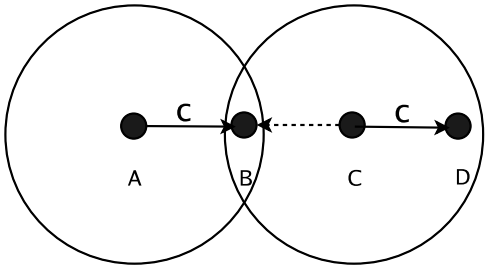


Fig. 8.1: 3 hop code collision area

The topology information needed to determine if a collision is possible, is the set of nodes and node identifiers within the 3-hop neighborhood. Please consider figure 8.1. Node B,C and D are within the 3-hop neighborhood of A. If A is transmitting to B using code c, and C is transmitting to D using the same code c, B will experience a collision as it will also receive the transmission from C.

The topology of the network can be represented by an undirected graph G = (V,E), where V is the set of network nodes, and E is the set of links between the nodes. If $(u, v) \in E$, then $(v, u) \in E$ and node *u* and *v* are within transmission range of each other and can exchange packets with each other using some code. The nodes *u* and *v* are then *one-hop neighbors* of each other, such as for example node A and B in figure 8.1. The set of one-hop neighbors of a particular node *i* is denoted by N_i^1.

Each node has available a pool of well chosen codes, for example quasi-orthogonal PN codes, $C_{pn} = \{c^k\}$. Each code is identified by the superscript $k = 0, 1, 2, ..., |C_{pn}| - 1$.

When a node wishes to calculate the code to be used when transmitting to a

HASH (k)
{
  h = `Hash(`k`);`
  code = h `mod` $|Cpn|$
}

HASH (i,k)
{
  h = `Hash(`$i \oplus k$`)`
  code = h `mod` $|Cpn|$
}

node *k*, it uses the method *HASH(k)*. One implementation of the function `Hash(x)` is to use an integer pseudo-random number generator that generates the message digest from a byte-stream input *x*. This integer is then *mod*ulod to the number of available codes. This implementation is a good choice if addresses are assigned randomly in the network, or if there is no administration to assign addresses. If there is a centralized addressing authority available, this authority can assign address incrementally in such a way that no code collision will occur while the number of assigned addresses are less than the number of available codes. This hash function could then be implemented `x mod y`, where x is the address and y is the number of codes. If the used address consists different parts, for example as in IP addresses, the function might need to be modified slightly to take this into account.

In these cases a receiver based code used, but it also possible to use a link based code, as defined in *HASH(i,k)*, where the link between *i* and *k* is assigned a code. This means that each link, $(i, j) \in E$, is assigned a code.

Set_Codes (i)
{
  for ($j \in N_i^1$)
    for ($\forall k \in N_j^1 \cup (\bigcup_{l \in N_j^1} N_l^1)$) {
      code = **HASH(k)**
      k.TxCode = c^{code}
      $R \longleftarrow k.TxCode$
    }
}

The *Set_Codes(i)* method calculates the set of codes, *R* to be used by a node *i* by calculating the code corresponding to each node in the 3-hop neighborhood.

Calc_Collisions (i)
{
  for ($j \in N_i^1$)
    for ($\forall k \in N_j^1 \cup (\bigcup_{l \in N_j^1} N_l^1)$) {
      if ($\exists k.TxCode \in R, k \neq i$) {
        $Q \longleftarrow k$
      }
    }
}

After this, node *i* uses the method *Calc_Collisions (i)* to calculate the set of nodes, *Q*, within the neighborhood that share the same code.

A node wishing to transmit data to a certain node, can therefore check if a code collision is possible due to two or more local nodes using the same code,

Check_Collision (j)
{
if ($Q == \emptyset$)
return false
else if ($\exists j \in Q$)
return true
else
return false
}

before the transmission takes place, by using *Check_Collision(j)*. If two nodes in the neighborhood is using the same code, any node that wishes to transmit to them, first uses a CSMA scheme in order to determine if some other node is currently transmitting on that code. See *Packet_To_TX(j)*.

Packet_To_TX (j)
{
$F \longleftarrow f_0$;
if (***Check_Collision(j)***)
`Perform Carrier Sense on: F, j.TxCode`
else
`Transmit RTS to j on: F, j.TxCode`
$F, \mathcal{P}, R \longleftarrow$*Receive CTS(j)*
`Transmit DATA on: F, j.TxCode, using` $\mathcal{P}$ `and R`
}

8.4 Pre data signaling

Before any data is transmitted, an RTS message is transmitted. This RTS has several purposes, such as to determine if the intended destination(s) can receive the packet, and to determine appropriate transmission parameters.

Upon receiving the RTS message, the receiver performs a number of computations in order to determine the following:

- The transmission power
- The data and coding rate
- The type of modulation to use
- The number of subtones to use in the OFDM system
- Waterfilling, in order to determine the power level of each OFDM subtone
- Bit/power loading to determine the amount of information to transmit over each OFDM subtone.

In order for the receiver to determine these parameters, the receiver needs to know the correct channel state information. This information can either be determined by transmitting a small probe signal just prior to the RTS, or by having a specific field within the RTS that contains the probe signal. This probe signal or field is then used to calculate the channel impulse response.

The probe signal used, can be made to spread over a very wide band, but for a short time. This will allow the receiver to determine what subtones, or what range of subtones, that is to be used among all the subtones available within the allocated spectrum. The predefined code (by hashing) might be specified so that it can only spread over a certain limited frequency band, and the receiver will then be able determine the location of this band by using the probe signal and the state of the current subtones located within it.

For example, 802.11a defines a set of up to 14 frequency channels. While a single and unique code can be used to transmit to a particular node, we can choose between 14 different frequency bands to spread the signal over, effectively creating 14 different channels. This also means that although two nodes located in the same area might be using the same code, a collision will not occur, because they will transmit on different frequency bands.

Rx_RTS (i,j)
{
$\mathcal{G} \longleftarrow$ *Subtone Gain(i,j)*
$F \longleftarrow$ *Pick Frequency Range*$(\mathcal{G}, \mathcal{N})$
$\mathcal{P} \longleftarrow$ *Subtone Powerset*$(F, \mathcal{G}, \mathcal{N})$
$R \longleftarrow$ *Rate*$(\mathcal{P}, F, \mathcal{G}, \mathcal{N})$
`Transmit F,`$\mathcal{P}$`,R in CTS to i`
}

Rx_RTS (i,j) summarises the computation of the transmission parameters performed upon the reception of a RTS. $\mathcal{N}$ is the noise power subset.

Other combinations of this scheme are possible. We can, for example, if a lower data rate is needed by the transmitter, only use a portion of the wider frequency band, and let the receiver notify the transmitter of the set of subtones to use. Similarly, more subtones, possibly from several frequency bands, could be allocated if a higher data rate is needed.

The response CTS message transmitted by the receiver, is using the hash-function, *HASH(i)*, to determine the code to use for spreading the CTS.

The receiver also includes the current size of its queue in the CTS, for the given priority, if multiple queues are being used.

8.5 Diversity forwarding

As has been described in the previous two chapters, we will also support diversity forwarding.

When a transmitter has determined two or more receiver that it wishes to evaluate as candidates for forwarding the packet, it transmits the RTS packet to them using a common, well known RTS code that all nodes uses to try to decode a packet, see *Packet_To_TX(i,j,k)*.

Packet_To_TX (i,j,k)
{
$F \longleftarrow f_0$;
if Check_Diversity_Code (i,j,k)
`Perform Carrier Sense on: F, j.TxCode(j,k)`
`Transmit RTS to j,k on: F, j.TxCode(j,k)`
else
`Perform Carrier Sense on: F, 0.TxCode`
`Transmit RTS to j,k on: F, 0.TxCode`
$F_1, \mathcal{P}_1, R_1 \longleftarrow$*Receive CTS*(*j*)
$F_2, \mathcal{P}_2, R_2 \longleftarrow$*Receive CTS*(*k*)
$l, F_l, P_l, R_l \longleftarrow$ *Pick Candidate*$(F_{1,2}, \mathcal{P}_{1,2}, R_{1,2})$
`Transmit DATA on:` F_l`, l.TxCode, using` $\mathcal{P}_l$ `and` R_l
}

When a node receives a RTS message where its address is specified in one of the destination fields, it will use another hash-function to determine a group code. This group code will take as input parameters the addresses specified in the RTS messages. All nodes targeted by the RTS will then listen to and use this code for a certain specified time period when decoding packets. Within this specified time, whenever a source node transmits a diversity RTS message, this group code will be used to spread the RTS signal.

After this specified time period, the group code will be inactivated, and the common code will again be used to transmit the initial RTS message.

In the same manner as when data packets are being transmitted, before a RTS packet is transmitted, a check is made to determine if there is a collision risk when using the defined code. This is performed in *Check_Diversity_Code(i,j,k)*. Here, the diversity spreading code is first calculated. The set of diversity spreading codes currently used by a node is denoted *T* and the set of codes used for the actual data transmissions is as earlier denoted *R*. Then a check is performed to determine whether that code is already used as either a diversity code or as a normal data code. If it is, a CSMA carrier sensing scheme using that code is used prior to transmitting the RTS. The first time this lookup is performed, neither of the receiving candidates are listening to the specified group, as they are not yet aware of this setup. The

Check_Diversity_Code (i,j,k)
{
  h = `Hash(`$i \oplus j \oplus k$`)`;
  code = h `mod` $|Cpn|$;
  j.TxCode(j,k) = c^{code};
  k.TxCode(j,k) = c^{code};
  if ($\exists j.TxCode(j,k) \in (T \cup R)$)
    return true;
  else {
    $T \longleftarrow j.TxCode(j,k)$
    return false;
  }
}

common code will therefore be used when transmitting the first RTS. After this, the group code will be included in the *T* set, and candidate receivers will be listening to the code. As nodes can not be aware of group codes used by other nodes, carrier sensing is always performed before transmitting a diversity RTS.

It is also possible for the receiver to, in its CTS message, indicate an other data code than the one that the hash function is pointing to. It could for example indicate a longer code in order to increase the processing gain and the chance for a successful delivery, but at the price of a lower achieved data rate.

8.6 Near-far effect and acknowledgments

In cellular systems, the near-far effect is a very important problem that needs to be taken into account. The near far effect means that a mobile terminal very close to a base station, might be drowned by noise caused by the base station while it is communicating with a mobile terminal much further away. This problem is solved by different optimized power control algorithms and decreasing the actual geographical cell size if needed.

In ad hoc or mesh network, this problem is much more difficult to solve because power control is not centralized. One solution that eases this problem are to use acknowledgments, where each data packet is acknowledged separately. If the packet fails to be delivered at the receiver , the packet is retransmitted by the transmitter. This increases the end to end delay but increases the packet delivery ratio.

In this protocol, if the receiver determines that an acknowledgment might be useful, it indicates this with an 'ack' flag in the CTS message.

The receiver monitors the noise level both when the RTS message is received, and after a data packet is received. It then uses these values to calculate a noise-jitter that is then used to either determine whether an ACK should be transmitted,

or if the power of the transmitter should be increased accordingly. A problem with increasing the power level is of course, that the average power level increases, as other nodes might need to increases its power level, in turn affecting other nodes to turn up its power level and so on. This is something that therefore should be used carefully. Using an ACK is a safer option unless some coordination is used.

In this protocol, the near-far effect is also mitigated through the dynamic allocation of frequency bands. This means that a receiver that experiences cross-code interference from a nearby transmitter, might be allocated a different frequency band with less interference and noise.

8.7 The number of codes

An important question is how many codes we need to use in order to minimize the probability for a code collisions. We know that when we check for a code collision, we search within our 3-hop neighborhood. Many papers that model the connectivity of ad hoc network, including [11], model the number of nodes in a specific area with a Poisson process:

$$P(n\ nodes\ in\ A) = \frac{(\rho A)^n}{n!} e^{-\rho A} \tag{8.1}$$

where *P* is probability of finding *n* nodes in area *A* with density of ρ nodes per m^2.

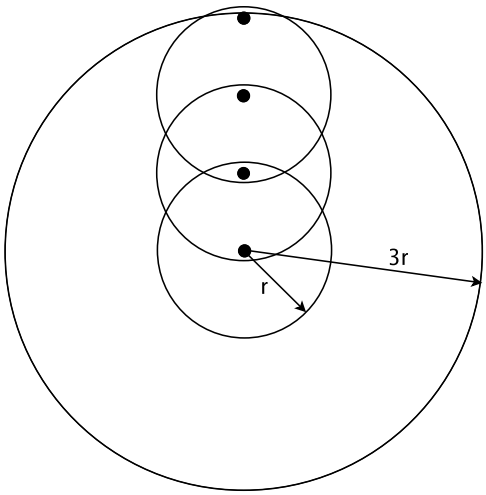


Fig. 8.2: Transmission areas for 1 and 3 hops

If we assume that a node has a transmission radius of *r* m, the expected number of neighbors a node has is simply the expected number of nodes located within its transmission radius:

$$E(\rho, r) = \rho \pi r^2 \tag{8.2}$$

The number of neighbors a node has is a nice metric that tell you about the density of the network. The ρ density, is defined as $\rho = \frac{n}{A}$, which depends on

the size of the network. But we can also express the density of the network as dependent upon the number of neighbors:

$$\rho = \frac{E}{\pi r^2} \tag{8.3}$$

This means we can now express the probability, P, of finding *n* nodes within a nodes 1-hop neighborhood, $A = \pi r^2$, dependent upon the number of neighbors, *E*.

$$P(n, E) = (\frac{\frac{E}{\pi r^2}\pi r^2}{n!})^n e^{-\frac{E}{\pi r^2}\pi r^2} = \frac{E^n}{n!}e^{-E} \tag{8.4}$$

We are now ready for expressing the probability of having a code collision. If we have *c* codes available, the probability of a code collision is the same having as at least *c* nodes in our 3-hop neighborhood:

$$P_{codes}(c, E) = 1 - \sum_{n=0}^{c-1}(\frac{(9E)^n}{n!}e^{-9E}) \tag{8.5}$$

For larger *c* and *E*, our used Poisson distribution becomes a normal distribution. So, for large *c* and *E* we get:

$$P_{codes}(c, E) = 1 - \sum_{n=0}^{c-1}(\frac{1}{\sqrt{2\pi 9E}}e^{(-\frac{(n-\sqrt{9E})^2}{2*9E})}) \tag{8.6}$$

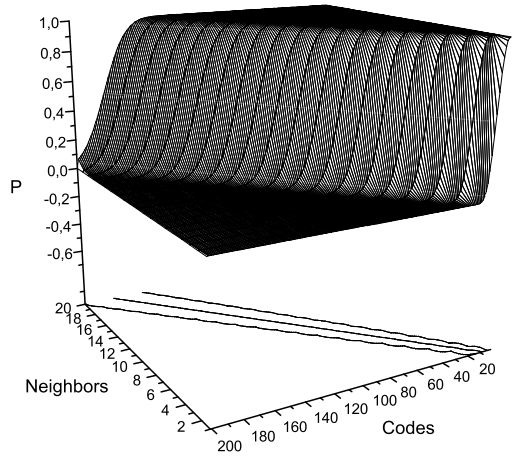


Fig. 8.3: Probability for code collision

As we can see fig 8.3, when the number of codes is less than 9 times the number of neighbors, the probability for a collision is one. This means that there will

always be more nodes in the 3-hop neighborhood, than there will be codes. As the number codes increases above 10 times the number of neighbors, we will always have enough codes.

8.8 Simulations

The Qualnet simulator [12] has been used to evaluate the proposed MAC protocol. Qualnet is a popular commercial event driven and scalable network simulator.

In the simulations, the diversity solution is not tested, and a simpler version of the protocol is used where RTS messages are broadcasted using the common code, after performing a carrier sense. The result of this is that RTS messages can collide, which is also the case for 802.11g DCF, which is used as a comparison. The protocol is also compared against 802.11g DCF without the use of RTS/CTS messages. Since we will have hidden terminals in these situations, this is something that is interesting to compare with.

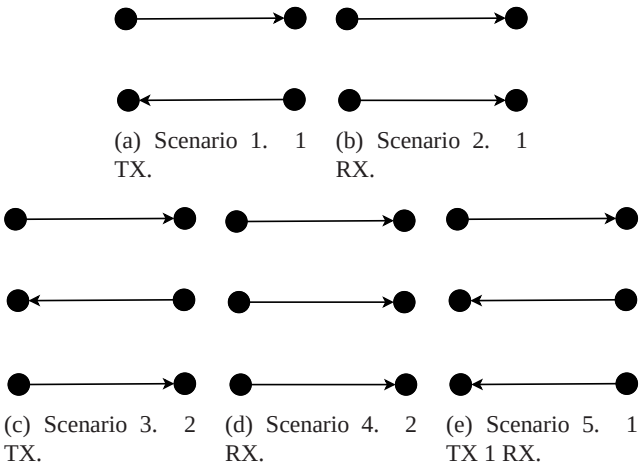
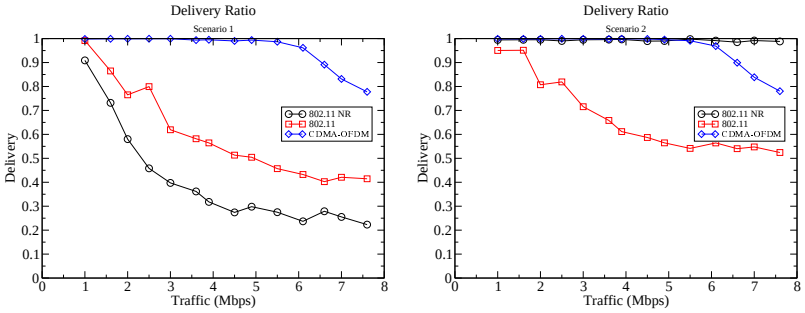


Fig. 8.4: The 5 simulated scenarios

5 different setups as illustrated in figure 8.4 have been simulated. They represent the cases where the middle flow (or lower flow) is competing with either 1 other transmitter, 1 other receiver, 2 other receivers, 2 other transmitter or 1 transmitter and 1 receiver. The distance between each transmitter is 250 meters, and the transmission channel experience Ricean Fading with k=1 with a simulated velocity of 1.5 m/s.

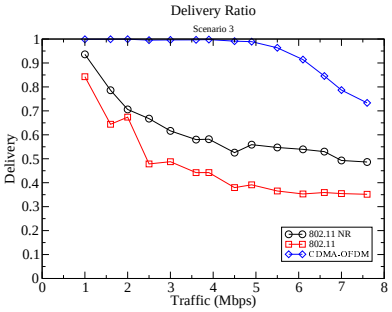
8.8.1 Simulation Results

The simulation results we will first have a look at is the delivery ratio for the 5 scenarios, shown in figure 8.5. We can see here that curves for CDMA-OFDM looks exactly the same for all the 5 simulated scenarios. This means that the expected performance doesn't depend on the presence of other parallel transmitters

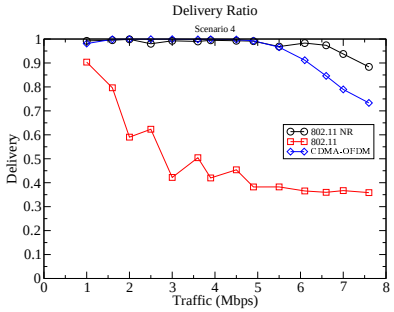


(a) Scenario 1.

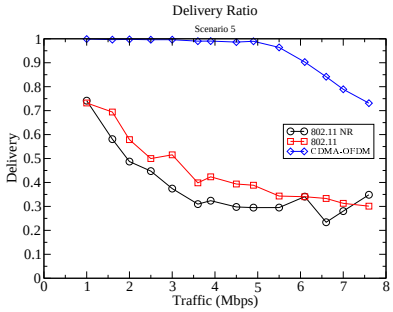
(b) Scenario 2.



(c) Scenario 3.



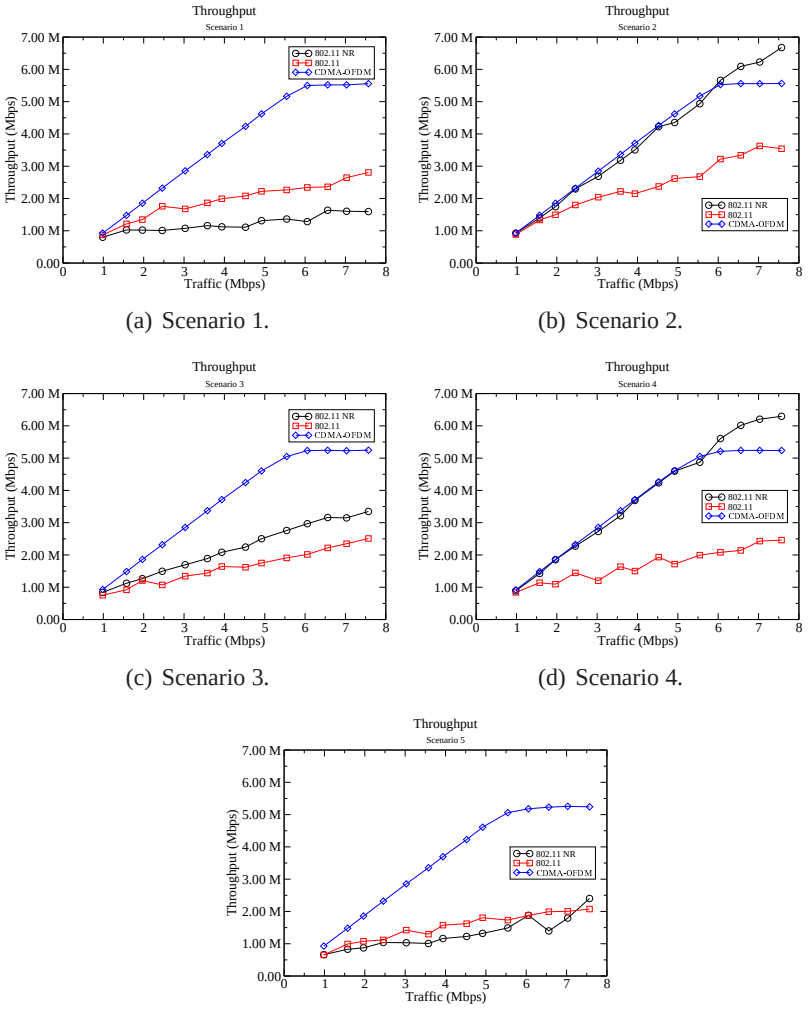
(d) Scenario 4.



(e) Scenario 5.

Fig. 8.5: Delivery Ratio for the 5 simulated scenarios

and receivers. We can see that the protocol delivers more or less all packets up to a certain point, where packets start to drop. This threshold is the capacity of the transmission channel for this particular setup. This can be seen in figure 8.6 that show the throughput. Here we see that the throughput levels off at around 5.5 Mbps for the 5 cases.



(a) Scenario 1. (b) Scenario 2.

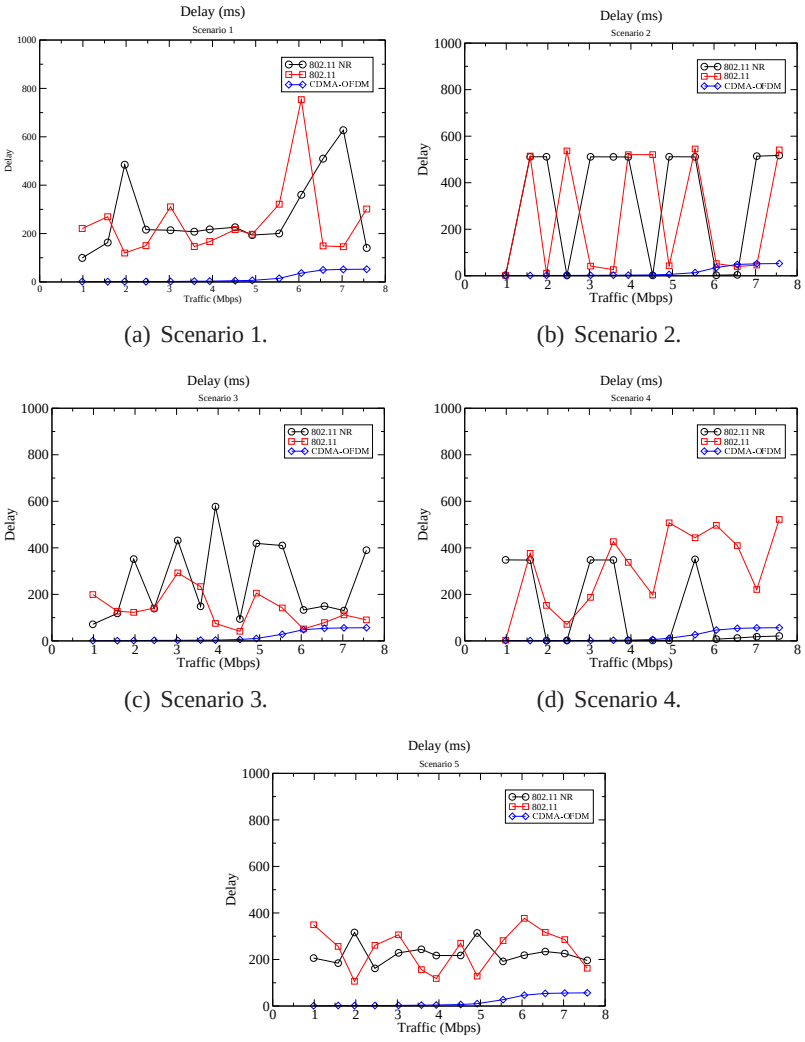
(c) Scenario 3. (d) Scenario 4.

(e) Scenario 5.

Fig. 8.6: Throughput for the 5 simulated scenarios

802.11 achieves the highest delivery ratio in scenario 2 and 4, where the ratio is around 100% up to, and through the CDMA-OFDM threshold. This is achieved when no RTS/CTS messages are sent prior to the data transmission. The reason for this is because in scenario 2 and scenario 4, the transmitter can sense other transmitters. In this case, the CSMA/CA carrier sensing functionality of 802.11 therefore works as it is supposed to. In scenario 1, for example, we have a hid-

den node and here we get higher delivery with RTS/CTS (fig 8.5(a)) and higher throughput (fig 8.6(a)). For scenario 3, 802.11 get a higher delivery and throughput without RTS/CTS because two of the transmitters can sense each other, even though they are outside each others tranmission zones.



(a) Scenario 1.

(b) Scenario 2.

(c) Scenario 3.

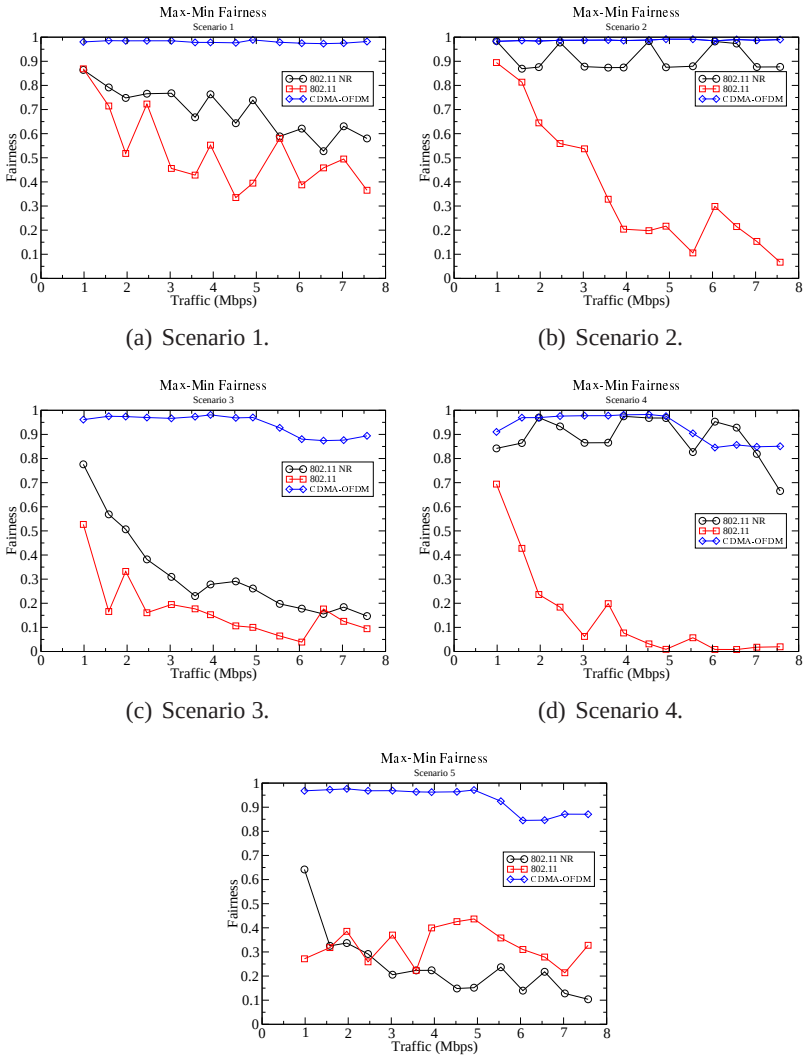
(d) Scenario 4.

(e) Scenario 5.

Fig. 8.7: Delay for the 5 simulated scenarios

When we look at the delay figures 8.7, we can see the curves for 802.11 jumping up and down a lot. What we actually see is the random access and the contention that occurs between nodes. CDMA-OFDM has the same low delay regardless of the traffic load, until the capacity thresholds is reached. Here we might have some contention among the RTS/CTS messages, but as data traffic is not transmitted on the same channel, this is not a problem. 802.11 devices on the other hand,

really fights among themselves to get access to the channel.



(a) Scenario 1. (b) Scenario 2.

(c) Scenario 3. (d) Scenario 4.

(e) Scenario 5.

Fig. 8.8: Max-Min Fairness for the 5 simulated scenarios

Figure 8.8 show the Max-Min fairness among the different flows. The Max-Min fairness is defined as the ratio between the lowest throughput and highest throughput among the flows. CDMA-OFDM illustrates very high fairness in the first two scenarios (fig 8.8(a) and 8.8(b)). When there are more flows, the fairness is still very high until traffic becomes very high, and above the capacity threshold. 802.11 with RTS/CTS has really bad fairness in scenario 2 and 4 (fig 8.8(b) and 8.8(d)). This is interesting, because these are cases when RTS/CTS are not really needed as carrier sensing works fine, as discussed above. On the other hand, fairness is not good in any of the cases. The conclusion have to be that with the

way 802.11 uses RTS/CTS messages, the flow allocation among flows is unfair and somewhat random.

8.9 Conclusion

A MAC protocol has been presented that uses CDMA and OFDMA to allocate channels to transmitting nodes. With this protocol, maximum flexibility in channel allocation can be achieved in both the frequency domain, and the code domain. A simple algorithm maps an address used by a node to a specific code it then uses for receiving packets. The code can be used in a specific frequency range that can be determined dynamically on a packet per packet level. The protocol enables channel estimation through the exchange of RTS and CTS messages that enables an OFDM transmitter to do power allocation on a per packet basis. The algorithm can detect the possibility of a code collision among neighboring nodes, and react to this by either using carrier sensing, dynamic frequency assignment, acknowledgements or a combination of these. The protocol also enables diversity forwarding where multiple nodes can be addressed with a group code that is created from a simple hash function. Simulations of the protocol show that it achieves good reliability, high throughput and fairness.

BIBLIOGRAPHY

[1] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification, IEEE Std 802.11-1997*. The Institute of Electrical and Electronics Engineers, New York, 1997.

[2] J. Garcia-Luna-Aceves and J. Raju. Distributed assignment of codes for multihop packet-radio networks. In *Proceedings of IEEE MILCOM 1997, Monterey, California, USA*, April 1997.

[3] S.W. Lee and D.H. Cho. Distributed reservation cdma for wireless lan. In *Proceedings of IEEE Globecom Conference 1995*, November 1995.

[4] M. Joa-Ng and I.-T. Lu. Spread spectrum medium access protocol with collision avoidance in mobile ad-hoc wireless network. In *Proceedings of IEEE Globecom Conference 1999*, November 1999.

[5] A. Muqattash and M. Krunz. Cdma-based mac protocol for wireless ad hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing, Annapolis, MD, USA*, June 2003.

[6] A. Yener and S. Kishore. Distributed power control and routing for clustered cdma wireless ad hoc networks. In *Proceedings of the IEEE Vehicular Technology Conference 2004, VTC fall 2004, Los Angeles, CA, USA*, September 2003.

[7] B. Bangerter, E. Jacobsen, M. Ho, A. Stephens, A. Maltsev, A. Rubtsov, and A. Sadri. High-throughput wireless lan air interface. *Intel Technology Journal*, 7(3):47–57, 2003.

[8] A.A. Maltsev, V.S. Sergeyev, and A.P Stephens. *System and Method for High-Throughput Wideband Wireless Local Area Network Communications, WO 2005/034435 A2*. World Intellectual Property Organization, 2005.

[9] A.A. Maltsev, A.S. Sadri, S. Tiraspolsky, and V.S. Sergeyev. *Method and Apparatus to Exchange Channel Information, WO 2005/067245 A1*. World Intellectual Property Organization, 2005.

[10] A. Butala and L. Tong. Cross-layer designs for medium access control in cdma ad hoc networks. *EURASIP Journal on Applied Signal Processing*, 2005(2):129–143, 2005.

[11] C. Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 80–91, June 2002. Lausanne, Switzerland.

[12] Scalable Networks. Qualnet network simulator, version 4.0. 2006.

9. CHAPTER IX

Summary and Conclusions

9.1 Performance Analysis of Traffic Load and Node Density in Ad hoc Networks - Chapter II

With the increasing popularity of mobile and wireless networking, it is important to understand how networks such as ad hoc networks behave in different situations so that they can be tuned to achieve optimum performance. A key component for achieving this is the connectivity of the network that can be estimated through the transmission power. For wireless transmission, a tradeoff exists between increasing the number of neighbors, and thus the connectivity, and decreasing the effective bandwidth available to individual network nodes.

It is desirable to increase the node density and transmission power in order to achieve high delivery of data packets to their destinations. However, while the optimum connectivity level of a network depend upon the mobility of the nodes, it also depends upon the traffic load on the network. In sparser networks it is possible to achieve high delivery rates up to a certain point, after where it starts to decline. When the transmission power of the individual nodes is increased, the delivery rate will also increase in a rate that is dependent upon the traffic load in the network. For lower traffic loads the increase in delivery is quite fast. As the traffic gets higher, the rate of this increase becomes slower. Although denser networks can generally achieve a higher delivery ratio, the cost will also be higher as more collisions occur which consume more power and channel bandwidth.

The conclusion we can draw from this work is that when the behavior, capacity and performance of a wireless ad hoc network is to be determined, the amount of traffic expected in the network, the expected mobility of nodes, the routing protocol as well as the node density needs to be taken into account. These results can be used as an aid when planning future simulations or deployments, and to get a rough overview of what capacity region the system is expected to operate within.

9.2 Internet Connectivity for Mobile Ad hoc Networks - - Chapter III

With the continued growth of interest in ad hoc networks, it is inevitable that some of them will at least occasionally encounter nearby potential points of attachment to different type of networks, including the global Internet. With today's wireless hot spot and mobile internet technologies, wireless access will be become very

familiar in our everyday life and enable Internet access from many locations within urban areas. Most hot spots support IP addressable devices and should be enhanced to enable the construction of a wireless ad hoc network. The point at which and attachment to the global Internet is to be made is called the *Internet Gateway*.

Some of the problems encountered while attempting to connect nodes in an ad hoc network to the Internet with mobility support in IPv6 networks are:

- site-local address acquisition and Duplicate Address Detection;
- acquiring a routing prefix from an Internet Gateway;
- establishing a default route and a host route toward the gateway;
- formulating a globally unique and topologically correct IPv6 address using the acquired routing prefix;
- soliciting gateway information whenever needed;
- when it is unknown whether a destination is present in the ad hoc network, determining whether to acquire a host route or using the default router;
- using the globally unique IPv6 address with Mobile IPv6;
- modifying the IPv6 ICMPv6 Router Solicitation and Advertisement messages to work across multihop networks;
- extending the route discovery mechanisms for on demand routing protocols to enable gateway discovery.

It is proposed that a manet node with a need for global communication contacts an Internet Gateway by either sending a modified Router Solicitation, called Gateway Solicitation, or relying on routing protocol route discovery functions. When the gateway receives one of these messages, it unicasts a response back to the requesting node, specifying its globally routable prefix and IPv6 address. The node then uses this information to configure an address that is globally reachable throughout the Internet. With Mobile IPv6, the mobile node can use this address as its care-of address and make a Binding Update to its Home Agent.

When sending packets to the Internet, the node can either use a routing header specifying the Internet Gateway as the first destination and rely on ordinary ad hoc routing to route the packet to the gateway, or send the packets through the default route, relying on intermediate nodes to forward the packet toward the destination.

This chapter may help future deployers of multi hop access technologies to better understand the constraints on the network layer, especially when IPv6 is being used.

9.3 Routing in Hybrid Ad hoc Networks using Service Points - Chapter IV

Table-driven or proactive protocols can become expensive in terms of control overhead, because each node in the network must maintain routing information for every other node, although the node only occasionally handles traffic destined for some of the nodes. To address the scaling problem of table-driven routing, on-demand routing protocols have been proposed for ad hoc networks. Nodes running such protocols set up and maintain routes to destinations only if they are active recipients of data packets. However, when routing information between only a few sources and destinations is constantly being maintained on-demand, possibly because the destination is a service point, it might be more attractive to use the proactive approach for these nodes, while on-demand routing is used between less accessed nodes.

In many practical scenarios, certain nodes provide special services that are being requested throughout the network. For example, when ad hoc networks are wireless extensions of the Internet, these nodes may act as DNS servers, Internet Access points, web proxies or AAA servers. Services can also be local, for example locally stored data or database information. These nodes that host special services have a higher likelihood of communicating with the rest of the network, and are called *netmarks*.

A new routing scheme is proposed, Netmark Overlay Routing Protocol (NORP). NORP proactively maintains routes to *special service providing nodes* in the network. These nodes are called netmarks. This is achieved through an extensive neighbor protocol that creates a bidirectional routing tree with the root attached to the netmark. In addition, NORP reactively searches for nodes by querying the different netmarks about the location of a destination node. Data packets are then routed using landmark routing towards the netmark closest to the destination node. As the data packet comes closer to the destination netmark, it will eventually arrive at a node within the routing tree of destination's netmark, where it will be routed to the final destination.

Simulations show that NORP achieves very high delivery rates in dense networks and under high traffic loads. They also show that NORP performs excellent under mobile conditions and has good scalability properties. In conclusion, NORP is a service providing routing protocol that scales well with the size of the network.

9.4 Micro Mobility and Internet Access Performance in Ad hoc Networks - Chapter V

In ad hoc networks, an infrastructure is not needed for the network to successfully operate, but an ad hoc network can enable the coverage area of access networks to be extended and deal with situations where it is either not possible or too expensive to deploy cell-based mobile network infrastructures.

A problem with IP is that it was never designed to support mobility management. One of the most widely known Mobility solutions for IP networks is the IP Mobility Support protocol, Mobile IP. With Mobile-IP, nodes are able to communicate independently of their current point of attachment to the Internet.

A solution has been presented, and evaluated for TCP connections, that enable mobile nodes in an ad hoc network to have internet connectivity. Here, the ad hoc networks are regarded as subnets of the Internet, that creates an integrated environment that supports both macro and micro IP mobility. This solution relies on the AODV or OLSR routing protocols for establishing multihop paths between a mobile node and a base station. For micro mobility, the solution is based on HAWAII, a domain based micro mobility scheme.

Evaluations of the TCP transport layer performance of the solution indicate that a fairly high throughput can be achieved, even during very high mobility speeds. However, the characteristics of the wireless environment itself, as well as inefficiencies of the 802.11 MAC layer protocol, lowers the performance when the number of hops increases. By using a less aggressive version of TCP such as Vegas, or lowering the maximum window size, the throughput can be somewhat increased as well stabilized.

TCP Vegas produces connections with lower delays due both to its ability to avoid congestion and overflow as well as it being more resilient to random packet loss.

Simulations also show that the main factor of concern to the throughput of TCP connections are link breaks, rather than flavour and window behaviour.

If the mobility rate is low, OLSR is to be the preferred routing protocol as it achieves a higher throughput and lower delay for most of the TCP flavours. For higher mobility speeds, AODV would be the better choice.

The problem with unfairness needs to be considered when multiple TCP flows are to be supported.

For future deployments of micro mobility ad hoc networks, I would recommend the use of a slightly modified version of TCP Vegas on the transport layer. TCP Vegas is much more resilient to random packet loss, which is a common and well known problem for wireless networks. TCP Vegas also has more efficient congestion control than TCP Reno and Tahoe. A problem with TCP Vegas is a that a connection cannot cope with path changes that changes the round trip time. A minor but important modification of TCP Vegas would therefore be to dynamically and constantly adjust the lowest experienced round trip time variable. Another recommended modication is the addition of a more efficent bandwidth estimation scheme.

When choosing the routing protocol, the mobility rate, the type of mobile devices, the amount of traffic and other scenario dependent aspects should be taken into account. For battery operated devices with only sporadic traffic, a reactive protocol might choosen. For nodes with a more permanent supply in not so mobile networks, a proactive protocol would be preferred. For other situations a hybrid protocol could be used.

The MAC protocol should be able to handle the medium access in such a way that different TCP flows are not affected by uneccessary unfairness.

9.5 Diversity forwarding in Ad hoc and Mesh Networks - Chapter VI

In multi path routing, multiple paths between a source and a destination is setup in order to easily switch to a new path if the old path breaks. This will also enable the possibility for load balancing between different routes, and to distribute the load in the network. A special type of multi path routing is non-disjoint multi path routing. In this type of routing, every source and intermediate node on the path towards the destination has one or more next hop candidate nodes.

By having a non-disjoint routing scheme we can let each forwarding node make a forwarding decision based on the best current channel conditions. If the signal strength on a link to one next hop neighbor is in a current bad state due to fading, it may be possible to choose another next hop, that is currently in a better fading situation. This is commonly called diversity forwarding.

A cross layer solution is presented that defines and specifies a MAC and a routing protocol that interact in order to create efficient diversity forwarding.

The routing protocol (ODMLS) is semi reactive and operates by setting up routes on demand, but maintains a link state database that is continuously updated by using a promiscuous mode operation, like the promiscuous mode specified in 802.11, and listening to other data and control traffic.

The routing protocol setup multiple non-disjoint paths between a source and destination and presents the MAC layer with a set of candidate next hop forwarding nodes. The MAC protocol evaluates the candidates presented by the routing protocol, and performs power, rate and interference control in addition to implementing the diversity forwarding capabilities. The MAC protocol also has the ability to dyanmically schedule neighboring parallel transmissions, as long as they don't interfere with each other.

Both protocols are involved in the process of routing a packet, but they operate on different timescales and on different horizons. The routing protocol operates on information that is provided by the link state database, which is averaged and filtered over time. The MAC protocol operates on a shorter timescale and tries to determine the status and condition of a link with a ms resolution. The routing process is truly cross layer, and the final routing decision is made by using the routing table in combination with fast link evaluation. This faster link evaluation is what enables it to adapt to bad fading situations.

Simulations show that the end to end delay can be significantly reduced, and indicates that significant performance gains may be achieved.

9.6 Urban Mesh and Ad hoc Mesh Access Networks - Chapter VII

This chapter focuses on both ad hoc and mesh networks, and on a combination of the two. It is investigated how well these types of networks can be expected to operatel in a typical city environment, and a type of suburban environment. The simulations is performed on ad hoc/mesh networks in an urban setting, that takes into account fading and the propagation effect of the walls of different buildings, in combination with different well known routing, MAC and physical layer protocols.

These results are compared against a diversity forwarding solution, that includes a diversity MAC and routing protocols.

The MAC protocol is similar to the lite MAC protocol in Chapter VI, with the additional capability of dynamically selecting a channel.

The routing protocol is a form of proactive link state protocol. The link information is broadcasted to one hop neighbors, where a fisheye scope is used to determine which links that are to included in the update.

User nodes use a registration application to register with Mesh access Points, MPs in order to gain access to the mesh network. This allow user nodes to use the mesh network as an access network to external services, as well as a transportation network when they are communicating with nodes inside the network. The registration application allow user nodes that is not running a diversity routing protocol to use diversity forwarding on the MAC layer towards its current MP.

The methodology has been simulated in an urban and a suburban city environment for voice related traffic.

Four different type of networks have been considered. The first is a pure ad hoc network where user nodes are moving along the streets of the city, and communicating with other user nodes. The second is the same as the first, but here some of the nodes are not user nodes, but fixed nodes placed strategically in intersections. In the third type the fixed nodes are equipped with dual radio interfaces and is called MPs as they establish the mesh infrastructure. User nodes run the registration application and registers with and send their data traffic to the MPs. The fourth type is the same as the third, but the number of MPs are much fewer and user nodes need to use multi hop routing to reach the MPs.

Simulations show that the new protocols deliver packets with a significantly lower delay, although at the price of a somewhat lower delivery ratio.

By operating the MPs in ad hoc mode, we enable more flexibility and functionality to be defined in the software of both the user nodes and the mesh nodes. The simulations show that the urban environment has significantly lower performance than a suburban environment. We also see that for a suburban environment it is better to use a mesh type of network, while in urban environments, an ad hoc type of network is beneficial. What type of network to choose in the different environments, depends on what protocols we are using.

9.7 Hybrid multi channel CDMA/OFDMA and diversity forwarding - Chapter VIII

This chapter presents a MAC protocol that uses CDMA and OFDMA to allocate channels to transmitting nodes.

The presented method uses a pool of predefined and orthogonal CDMA-codes. The system also uses a predefined hash function, that maps a specific node address, or node address pair, to a specific CDMA-code. This means that the address used to identify a specific node, or the addresses used to identify a link, directly maps to a specific code. When a node has data that it wishes to transmit to another nearby node, it uses the hash function to determine a code that will be used to encode and spread the data signal.

The algorithm can detect the possibility of a code collision among neighboring nodes by using topology information, and react to this by either using carrier sensing, dynamic frequency or time slot assignment, acknowledgements, or a combination of these.

With this protocol, maximum flexibility in channel allocation can be achieved in both the frequency domain, and the code domain. The code can be used in a specific frequency range that can be determined dynamically on a packet per packet level. The protocol enables channel estimation through the exchange of RTS and CTS messages that enables an OFDM transmitter to do power allocation on a per packet basis.

Before any data is transmitted, an RTS message is transmitted. The RTS has several purposes, such as to determine if the intended destination(s) can receive the packet, and to determine appropriate transmission parameters so that fast OFDM link adaptation can be performed.

Upon receiving the RTS message, the receiver performs a number of computations in order to determine the following:

- The transmission power
- The data and coding rate
- The type of modulation to use
- The number of subtones to use in the OFDM system
- Waterfilling, in order to determine the power level of each OFDM subtone
- Bit/power loading to determine the amount of information to transmit over each OFDM subtone.

The protocol also enables diversity forwarding where multiple nodes can be addressed with a group code that is created from a simple hash function.

Simulations of the protocol show that it achieves good reliability, high throughput and fairness.

Reports on Communication Systems

101. **On Overload Control of SPC-systems**
Ulf Kőrner, Bengt Wallstrőm, and Christian Nyberg, 1989.
CODEN: LUTEDX/TETS- -7133- -SE+80P

102. **Two Short Papers on Overload Control of Switching Nodes**
Christian Nyberg, Ulf Kőrner, and Bengt Wallstrőm, 1990.
ISRN LUTEDX/TETS- -1010- -SE+32P

103. **Priorities in Circuit Switched Networks**
Åke Arvidsson, Ph.D. thesis, 1990.
ISRN LUTEDX/TETS- -1011- -SE+282P

104. **Estimations of Software Fault Content for Telecommunication Systems**
Bo Lennselius, Lic. thesis, 1990.
ISRN LUTEDX/TETS- -1012- -SE+76P

105. **Reusability of Software in Telecommunication Systems**
Anders Sixtensson, Lic. thesis, 1990.
ISRN LUTEDX/TETS- -1013- -SE+90P

106. **Software Reliability and Performance Modelling for Telecommunication Systems**
Claes Wohlin, Ph.D. thesis, 1991.
ISRN LUTEDX/TETS- -1014- -SE+288P

107. **Service Protection and Overflow in Circuit Switched Networks**
Lars Reneby, Ph.D. thesis, 1991.
ISRN LUTEDX/TETS- -1015- -SE+200P

108. **Queueing Models of the Window Flow Control Mechanism**
Lars Falk, Lic. thesis, 1991.
ISRN LUTEDX/TETS- -1016- -SE+78P

109. **On Efficiency and Optimality in Overload Control of SPC Systems**
Tobias Rydén, Lic. thesis, 1991.
ISRN LUTEDX/TETS- -1017- -SE+48P

110. **Enhancements of Communication Resources**
Johan M Karlsson, Ph.D. thesis, 1992.
ISRN LUTEDX/TETS- -1018- -SE+132P

111. **On Overload Control in Telecommunication Systems**
Christian Nyberg, Ph.D. thesis, 1992.
ISRN LUTEDX/TETS- -1019- -SE+140P

112. **Black Box Specification Language for Software Systems**
Henrik Cosmo, Lic. thesis, 1994.
ISRN LUTEDX/TETS- -1020- -SE+104P

113. **Queueing Models of Window Flow Control and DQDB Analysis**
Lars Falk, Ph.D. thesis, 1995.
ISRN LUTEDX/TETS- -1021- -SE+145P

114. **End to End Transport Protocols over ATM**
Thomas Holmstrom, Lic. thesis, 1995.
ISRN LUTEDX/TETS- -1022- -SE+76P

115. **An Efficient Analysis of Service Interactions in Telecommunications**
Kristoffer Kimbler, Lic. thesis, 1995.
ISRN LUTEDX/TETS- -1023- -SE+90P

116. **Usage Specifications for Certification of Software Reliability**
Per Runeson, Lic. thesis, May 1996.
ISRN LUTEDX/TETS- -1024- -SE+136P

117. **Achieving an Early Software Reliability Estimate**
Anders Wesslén, Lic. thesis, May 1996.
ISRN LUTEDX/TETS- -1025- -SE+142P

118. **On Overload Control in Intelligent Networks**
Maria Kihl, Lic. thesis, June 1996.
ISRN LUTEDX/TETS- -1026- -SE+80P

119. **Overload Control in Distributed-Memory Systems**
Ulf Ahlfors, Lic. thesis, June 1996.
ISRN LUTEDX/TETS- -1027- -SE+120P

120. **Hierarchical Use Case Modelling for Requirements Engineering**
Bjorn Regnell, Lic. thesis, September 1996.
ISRN LUTEDX/TETS- -1028- -SE+178P

121. **Performance Analysis and Optimization via Simulation**
Anders Svensson, Ph.D. thesis, September 1996.
ISRN LUTEDX/TETS- -1029- -SE+96P

122. **On Network Oriented Overload Control in Intelligent Networks**
Lars Angelin, Lic. thesis, October 1996.
ISRN LUTEDX/TETS- -1030- -SE+130P

123. **Network Oriented Load Control in Intelligent Networks Based on Optimal Decisions**
Stefan Pettersson, Lic. thesis, October 1996.
ISRN LUTEDX/TETS- -1031- -SE+128P

124. **Impact Analysis in Software Process Improvement**
Martin Host, Lic. thesis, December 1996.
ISRN LUTEDX/TETS- -1032- -SE+140P

125. **Towards Local Certifiability in Software Design**
Peter Molin, Lic. thesis, February 1997.
ISRN LUTEDX/TETS- -1033- -SE+132P

126. **Models for Estimation of Software Faults and Failures in Inspection and Test**
Per Runeson, Ph.D. thesis, January 1998.
ISRN LUTEDX/TETS- -1034- -SE+222P

127. **Reactive Congestion Control in ATM Networks**
Per Johansson, Lic. thesis, January 1998.
ISRN LUTEDX/TETS- -1035- -SE+138P

128. **Switch Performance and Mobility Aspects in ATM Networks**
Daniel Søbirk, Lic. thesis, June 1998.
ISRN LUTEDX/TETS- -1036- -SE+91P

129. **VPC Management in ATM Networks**
Sven-Olof Larsson, Lic. thesis, June 1998.
ISRN LUTEDX/TETS- -1037- -SE+65P

130. **On TCP/IP Traffic Modeling**
Pär Karlsson, Lic. thesis, February 1999.
ISRN LUTEDX/TETS- -1038- -SE+94P

131. **Overload Control Strategies for Distributed Communication Networks**
Maria Kihl, Ph.D. thesis, March 1999.
ISRN LUTEDX/TETS- -1039- -SE+158P

132. **Requirements Engineering with Use Cases - a Basis for Software Development**
Bjorn Regnell, Ph.D. thesis, April 1999.
ISRN LUTEDX/TETS- -1040- -SE+225P

133. **Utilisation of Historical Data for Controlling and Improving Software Development**
Magnus C. Ohlsson, Lic. thesis, May 1999.
ISRN LUTEDX/TETS- -1041- -SE+146P

134. **Early Evaluation of Software Process Change Proposals**
Martin Host, Ph.D. thesis, June 1999.
ISRN LUTEDX/TETS- -1042- -SE+193P

135. **Improving Software Quality through Understanding and Early Estimations**
Anders Wesslén, Ph.D. thesis, June 1999.
ISRN LUTEDX/TETS- -1043- -SE+242P

136. **Performance Analysis of Bluetooth**
Niklas Johansson, Lic. thesis, March 2000.
ISRN LUTEDX/TETS- -1044- -SE+76P

137. **Controlling Software Quality through Inspections and Fault Content Estimations**
Thomas Thelin, Lic. thesis, May 2000
ISRN LUTEDX/TETS- -1045- -SE+146P

138. **On Fault Content Estimations Applied to Software Inspections and Testing**
Håkan Petersson, Lic. thesis, May 2000.
ISRN LUTEDX/TETS- -1046- -SE+144P

139. **Modeling and Evaluation of Internet Applications**
Ajit K. Jena, Lic. thesis, June 2000.
ISRN LUTEDX/TETS- -1047- -SE+121P

140. **Dynamic traffic Control in Multiservice Networks - Applications of Decision Models**
Ulf Ahlfors, Ph.D. thesis, October 2000.
ISRN LUTEDX/TETS- -1048- -SE+183P

141. **ATM Networks Performance - Charging and Wireless Protocols**
Torgny Holmberg, Lic. thesis, October 2000.
ISRN LUTEDX/TETS- -1049- -SE+104P

142. **Improving Product Quality through Effective Validation Methods**
Tomas Berling, Lic. thesis, December 2000.
ISRN LUTEDX/TETS- -1050- -SE+136P

143. **Controlling Fault-Prone Components for Software Evolution**
Magnus C. Ohlsson, Ph.D. thesis, June 2001.
ISRN LUTEDX/TETS- -1051- -SE+218P

144. **Performance of Distributed Information Systems**
Niklas Widell, Lic. thesis, February 2002.
ISRN LUTEDX/TETS- -1052- -SE+78P

145. **Quality Improvement in Software Platform Development**
Enrico Johansson, Lic. thesis, April 2002.
ISRN LUTEDX/TETS- -1053- -SE+112P

146. **Elicitation and Management of User Requirements in Market-Driven Software Development**
Johan Natt och Dag, Lic. thesis, June 2002.
ISRN LUTEDX/TETS- -1054- -SE+158P

147. **Supporting Software Inspections through Fault Content Estimation and Effectiveness Analysis**
Håkan Petersson, Ph.D. thesis, September 2002.
ISRN LUTEDX/TETS- -1055- -SE+237P

148. **Empirical Evaluations of Usage-Based Reading and Fault Content Estimation for Software Inspections**
Thomas Thelin, Ph.D. thesis, September 2002.
ISRN LUTEDX/TETS- -1056- -SE+210P

149. **Software Information Management in Requirements and Test Documentation**
Thomas Olsson, Lic. thesis, October 2002.
ISRN LUTEDX/TETS- -1057- -SE+122P

150. **Increasing Involvement and Acceptance in Software Process Improvement**
Daniel Karlstrom, Lic. thesis, November 2002.
ISRN LUTEDX/TETS- -1058- -SE+125P

151. **Changes to Processes and Architectures; Suggested, Implemented and Analyzed from a Project viewpoint**
Josef Nedstam, Lic. thesis, November 2002.
ISRN LUTEDX/TETS- -1059- -SE+124P

152. **Resource Management in Cellular Networks -Handover Prioritization and Load Balancing Procedures**
Roland Zander, Lic. thesis, March 2003.
ISRN LUTEDX/TETS- -1060- -SE+120P

153. **On Optimisation of Fair and Robust Backbone Networks**
Pål Nilsson, Lic. thesis, October 2003.
ISRN LUTEDX/TETS- -1061- -SE+116P

154. **Exploring the Software Verification and Validation Process with Focus on Efficient Fault Detection**
Carina Andersson, Lic. thesis, November 2003.
ISRN LUTEDX/TETS- -1062- -SE+134P

155. **Improving Requirements Selection Quality in Market-Driven Software Development**
Lena Karlsson, Lic. thesis, November 2003.
ISRN LUTEDX/TETS- -1063- -SE+132P

156. **Fair Scheduling and Resource Allocation in Packet Based Radio Access Networks**
Torgny Holmberg, Ph.D. thesis, November 2003.
ISRN LUTEDX/TETS- -1064- -SE+187P

157. **Increasing Product Quality by Verification and Validation Improvements in an Industrial Setting**
Tomas Berling, Ph.D. thesis, December 2003.
ISRN LUTEDX/TETS- -1065- -SE+208P

158. **Some Topics in Web Performance Analysis**
Jianhua Cao, Lic. thesis, June 2004.
ISRN LUTEDX/TETS- -1066- -SE+99P

159. **Overload Control and Performance Evaluation in a Parlay/OSA Environment**
Jens K. Andersson, Lic. thesis, August 2004.
ISRN LUTEDX/TETS- -1067- -SE+100P

160. **Performance Modeling and Control of Web Servers**
Mikael Andersson, Lic. thesis, September 2004.
ISRN LUTEDX/TETS- -1068- -SE+105P

161. **Integrating Management and Engineering Processes in Software Product Development**
Daniel Karlstrőm, Ph.D. thesis, December 2004.
ISRN LUTEDX/TETS- -1069- -SE+230P

162. **Managing Natural Language Requirements in Large-Scale Software Development**
Johan Natt pch Dag, Ph.D. thesis, Febuary 2005.
ISRN LUTEDX/TETS- -1070- -SE+222P

163. **Designing Resilient and Fair Multi-layer Telecommunication Networks**
Eligijus Kubilinskas, Lic. thesis, Febuary 2005.
ISRN LUTEDX/TETS- -1071- -SE+136P

164. **Internet Access and Performance in Ad hoc Networks**
Anders Nilsson, Lic. thesis, April 2005.
ISRN LUTEDX/TETS- -1072- -SE+119P

165. **Active Resource Management in Middleware and Serviceoriented Architectures**
Niklas Widell, Ph.D. thesis, May 2005.
ISRN LUTEDX/TETS- -1073- -SE+162P

166. **Quality Improvement with Focus on Performance in Software Platform Development**
Enrico Johansson, Ph.D. thesis, June 2005.
ISRN LUTEDX/TETS- -1074- -SE+139P

167. **On InterSystem Handover in a Wireless Hierarchical Structure**
Henrik Persson, Lic. thesis, September 2005.
ISRN LUTEDX/TETS- -1075- -SE+90P

168. **Prioritation Procedures for Resource Management in Cellular Networks**
Roland Zander, Ph.D. thesis, December 2005.
ISRN LUTEDX/TETS- -1076- -SE+181P

169. **Strategies for Management of Architectural Change and Evolution**
Josef Nedstam, Ph.D. thesis, December 2005.
ISRN LUTEDX/TETS- -1077- -SE+192P

170. **Internet Access and QoS in Ad Hoc Networks**
Ali Hamidian, Lic. thesis, April 2006.
ISRN LUTEDX/TETS- -1078- -SE+118P

171. **Managing Software Quality through Empirical Analysis of Fault Detection**
Carina Andersson, Ph.D. thesis, May 2006.
ISRN LUTEDX/TETS- -1079- -SE+216P

172. **Fairness in Communication and Computer Network Design**
Pål Nilsson, Ph.D. thesis, September 2006.
ISRN LUTEDX/TETS- -1080- -SE+138P

173. **Requirements Prioritation and Retrospective Analysis for Release Planning Process Improvement**
Lena Karlsson, Ph.D. thesis, October 2006.
ISRN LUTEDX/TETS- -1081- -SE+192P

174. **Overload Control and Performance Evaluation of Web Servers**
Mikael Andersson, Ph.D. thesis, May 2007.
ISRN LUTEDX/TETS- -1082- -SE+137P

175. **On Overload Control and Performance Agreements in a Parlay/OSA Environment**
Jens K. Andersson, Ph.D. thesis, May 2007.
ISRN LUTEDX/TETS- -1083- -SE+154P

176. **Wireless Multi Hop Access Networks and Protocols**
Anders Nilsson Plymoth, Ph.D. thesis, December 2007.
ISRN LUTEDX/TETS- -1084- -SE+194P