Internet Access and QoS in Ad Hoc Networks

Hamidian, Ali

2006

Internet Access and QoS in Ad Hoc Networks

Ali Hamidian

LUND UNIVERSITY

Department of Communication Systems
Faculty of Engineering

ISSN 1101-3931
ISRN LUTEDX/TETS–1077–SE+118P

Printed in Sweden
E-kop
Lund 2006

To Davoud, Susanne, Reza, Amir, and Arash

This thesis is submitted to Research Board FIME - Physics, Informatics, Mathematics and Electrical Engineering at Faculty of Engineering (LTH), Lund University in partial fulfillment of the requirements for the degree of Licentiate in Engineering.

Contact information:
Ali Hamidian
Department of Communication Systems
Lund University
P.O. Box 118
SE-221 00 Lund
Sweden

Tel: +46 46 222 04 21
Fax: +46 46 14 58 23
e-mail: alex.hamidian@telecom.lth.se

Abstract

It is likely that the increased popularity of *wireless local area networks* (WLANs) together with the continuous technological advances in wireless communication, also increase the interest for ad hoc networks. An ad hoc network is a wireless, autonomous, infrastructure-less network composed of stations that communicate with each other directly in a peer-to-peer fashion. When discussing *mobile ad hoc networks* (MANETs), we often refer to an ad hoc network where the stations cooperate in forwarding packets on behalf of each other to allow communication beyond their transmission range over multi-hop paths.

In order to realize the practical benefits of ad hoc networks, two challenges (among others) need to be considered: distributed *quality of service* (QoS) guarantees and multi-hop Internet access. This thesis presents conceivable solutions to both of these problems. The first two papers focus on the network layer and consider the provisioning of Internet access to ad hoc networks whereas the last two papers focus on the data link layer and investigate the provisioning of QoS to ad hoc networks. The first paper studies the interconnection between a MANET and the Internet. In addition, it evaluates three approaches for gateway discovery, which can be initiated by the gateway (proactive method), by the mobile station (reactive method) or by mixing these two approaches (hybrid method). The second paper also studies Internet access for MANETs, but with focus on micro mobility, i.e. mobile stations moving from one gateway to another. In particular, it evaluates a solution that allows mobile stations to access the Internet and roam from gateway to gateway. The third paper, gives an overview of the medium access mechanisms in IEEE 802.11 and their QoS limitations. Moreover, it proposes an enhancement to the contention-free medium access mechanism of IEEE 802.11e to provide QoS guarantees in WLANs operating in ad hoc network configuration. The fourth paper continues the work from the third paper by enhancing the scheme and dealing with the problems that occur due to hidden stations. Furthermore, it discusses how to deal with the problems that occur when moving from single-hop ad hoc networks (i.e. WLANs in ad hoc network configuration) to multi-hop ad hoc networks.

Acknowledgments

I would like to begin thanking my supervisor and the head of the department Professor Ulf Körner for supporting and guiding me through my research. It was he who presented an interesting topic for my Master's thesis and gave me the opportunity to pursue my Ph.D. program here at the Department of Communication Systems at Lund University. I would also like to express my gratitude to Dr. Christian Nyberg for his pedagogical explanations to the most difficult and abstract problems. Furthermore, I owe my colleague Anders Nilsson special thanks for sharing his experiences and for all interesting discussions we have had so far. Of course I will not forget to thank all the staff at the department for making the workplace such a pleasant environment - it really makes a great difference! Finally, I would like to thank the most important persons in my life, namely my family consisting of my parents Susanne and Davoud plus my brothers Reza, Amir and Arash.

Ali Hamidian
Lund, Sweden
April 2006

Contents

1 Introduction

The interest for *wireless local area networks* (WLANs) based on IEEE 802.11 [1] has been growing quickly during recent years. Today, 802.11 has become a de facto standard for WLANs. As a consequence of the increased popularity of WLANs, the interest for ad hoc networks has also increased. An ad hoc network is a wireless network composed of stations that communicate with each other directly in a peer-to-peer fashion. Thus, an ad hoc network is independent of any existing network infrastructure such as base stations or access points. Examples of simple ad hoc networks are two mobile phones connected through Bluetooth or two laptops connected through 802.11 (operating in ad hoc mode). When discussing *mobile ad hoc networks* (MANETs), we often refer to an ad hoc network where the stations cooperate in forwarding packets on behalf of each other to allow communication beyond their transmission range over multi-hop paths.

Two challenges, among many others, that need to be paid attention to in order to realize the practical benefits of ad hoc networks, are providing distributed *quality of service* (QoS) guarantees and multi-hop Internet access. In this thesis, both of these two topics are investigated.

2 The Data Link Layer: QoS in Ad Hoc Networks

From a layered point of view, the QoS issue can be treated in different layers of the protocol stack. However, QoS provisioning at the *medium access control* (MAC) sublayer[1] is a necessary (but sometimes not sufficient) condition. In other words, QoS provisioning in ad hoc networks is not possible unless supported by the MAC protocol. For example, in an ordinary ad hoc network based on 802.11a/b/g - no matter what QoS approach is used at higher layers - one cannot guarantee e.g. a maximum delay because the MAC protocol gives an unpredictable random waiting time before accessing the medium.

When talking about QoS guarantees, we must keep in mind that since a wireless medium is much more unpredictable and error-prone than a wired medium, QoS cannot be guaranteed as in a wired system, especially in un-

[1] The data link layer is composed of two sublayers: *logical link control* (LLC) and MAC.

licensed spectra. However, it is possible to provide techniques that increase the probability that certain traffic classes get adequate QoS and that can provide QoS guarantees in controlled environments. This is also formulated in the 802.11e standard (Section 5.1.1.2 - Media impact on design and performance) [2]:

When providing QoS services it should be understood that the MAC endeavors to provide QoS "service guarantees" within the limitations of the medium properties identified above. That is, particularly in unlicensed spectrum, true guarantees are often not possible. However gradations of service are always possible, and in sufficiently controlled environments, QoS guarantees can truly be made.

This section gives an overview of the 802.11 standard. In particular, it describes the extensions and enhancements to the 802.11 standard with focus on 802.11e, which aims at providing QoS.

2.1 IEEE 802.11

The 802.11 standard, which covers both the *physical* (PHY) layer and the MAC sublayer, provides two network configuration modes: `infrastructure` and `ad hoc`. Using the infrastructure network configuration, all unicast transmissions must pass through an *access point* (AP) that relays them to the destination. The AP can also be used by the stations to access the Internet. Using the ad hoc network configuration, any station can communicate to another station directly without the need of any AP. In this thesis, we study stations that use the ad hoc network configuration.

2.1.1 IEEE 802.11 PHY

The first version of the 802.11 standard was released in 1997. It specified three PHY layer options: *infrared* (IR), *frequency-hopping spread spectrum* (FHSS) and *direct sequence spread spectrum* (DSSS). While FHSS and DSSS operate at the *industrial, scientific and medical* (ISM) band at 2.4 GHz, IR uses near-visible light in the 850 nm to 950 nm range for signaling. Since all three PHY options offered low data rates of up to 2 Mbit/s, none of them became widely used. The breakthrough came in 1999 when 802.11b (based on DSSS), with a maximum data rate of 11 Mbit/s, was ready. The same year,

IEEE ratified 802.11a, which is based on *orthogonal frequency division multiplexing* (OFDM) and allows for data rates theoretically up to 54 Mbit/s. However, despite the higher throughput, 802.11a didn't become widely accepted as opposed to 802.11b. One important reason to this is that 802.11a operates in the 5 GHz band, implying that it is not backward compatible with the original standard. Another reason is that 802.11a has a shorter transmission range (also a consequence of the operation in the 5 GHz band). On the contrary, 802.11b operates in the ISM band at 2.4 GHz implying longer transmission range compared to 802.11a and backward compatibility with the original standard. Later in 2003, 802.11g was ratified. This version, which rapidly became the most popular standard, uses both DSSS and OFDM, operates at the 2.4 GHz ISM band, is backward compatible with 802.11b and allows for data rates theoretically up to 54 Mbit/s.

Currently, a task group is working on 802.11n that is expected to become the next-generation standard for WLANs. The task group is studying various enhancements to the PHY layer and the MAC sublayer to improve throughput. The goal is to support data rates of at least 100 Mbit/s, measured at the interface between the MAC sublayer and higher layers. The motivation to measure at a higher level than at the physical interface to the wireless medium (where 802.11/a/b/g measure the data rate), is to better match the data rates that a user experiences. In addition to improved throughput, 802.11n addresses improved range at existing throughputs and increased resistance to interference. One way to achieve these improvements is to use the *multiple-input multiple-output* (MIMO) technology, which uses multiple transmitter and receiver antennas to allow for higher data rates through spatial multiplexing and longer range by exploiting the spatial diversity. In order to improve the transfer efficiency, the MAC sublayer can be enhanced by aggregating multiple frames into a single PHY layer packet instead of initiating a new transfer for every frame.

2.1.2 IEEE 802.11 MAC

Although there has been several enhancements to the PHY layer since the first version of the standard was released (more specifically, these enhancements are 802.11a/b/g focusing on higher data rates), the medium access mechanism in the MAC sublayer has not been changed until recently

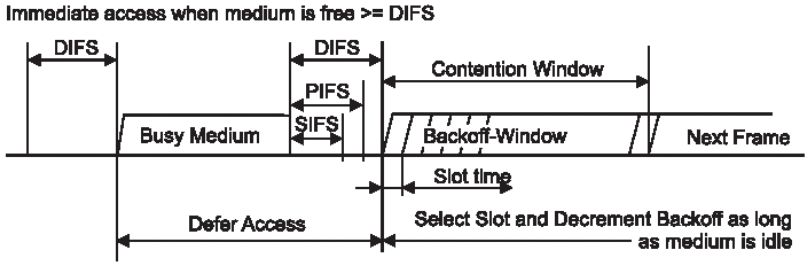


Figure 1: The interframe space relationships (copied from [1])

(with 802.11e that specifies a new coordination function). In other words, 802.11/a/b/g have all so far used the same protocol in the MAC sublayer. The original 802.11 standard has defined two medium access mechanisms: the mandatory *distributed coordination function* (DCF) and the optional *point coordination function* (PCF). The DCF is the basis for the PCF and is used for best effort contention services. Since the DCF is a distributed access method, it can be used not only in infrastructure network configurations, but also in ad hoc network configurations. The PCF on the other hand, depends on the DCF and is required for contention-free services. Furthermore, it is only usable in infrastructure network configurations.

To control the waiting time before medium access, the DCF and the PCF use four parameters called interframe spaces (illustrated in Figure 1): *Short interframe space* (SIFS), *PCF interframe space* (PIFS), *DCF interframe space* (DIFS) and *Extended interframe space* (EIFS). SIFS is the shortest waiting time, and is thus used to give the highest priority for medium access. It is used by short control messages, such as *clear to send* (CTS) frames, *acknowledgement* (ACK) frames and polling responses. PIFS is a waiting time longer than SIFS but shorter than DIFS (resulting in medium priority). It is used only by stations operating under the PCF, e.g. by the AP polling other stations. DIFS is a waiting time longer than both SIFS and PIFS and gives therefore the lowest priority for medium access. It is used only by stations operating under the DCF transmitting data or management frames. EIFS is the longest waiting time used by stations operating under the DCF, but only when a transmission failure occurs. A station that receives an incorrect frame must wait for EIFS before starting its transmission in order to give other stations enough time to acknowledge the frame that the station

received incorrectly.

The DCF uses *carrier sense multiple access with collision avoidance* (CSMA/CA) to regulate access to the shared medium. Whenever a station desires to transmit a frame, it must invoke the carrier-sense mechanism to determine whether the medium is busy or idle. There are two carrier-sense mechanisms: one physical provided by the PHY layer and one virtual, referred to as the *network allocation vector* (NAV), provided by the MAC sublayer. If both of these functions indicate an idle medium, the medium is considered idle; otherwise, the medium is considered busy. If the medium has been sensed to be idle for the duration of at least DIFS, the station can initiate a transmission immediately. Otherwise, if the medium is sensed to be busy or if it is sensed to be idle but becomes busy before the duration of DIFS, the station must defer until the end of the ongoing transmission. Then, the station must wait until the medium is determined to be idle without interruption for the duration of DIFS. Finally, it must invoke the random backoff process, which is an additional random waiting time necessary to reduce the probability of collision. This is because at this moment (after the medium becomes idle following a busy medium) there is a high probability of collision since several stations may be waiting for the medium to become idle. The random backoff time is calculated as follows:

$$\text{backoff time} = \text{random}() \times \text{slot\_time},$$

where random() is a uniformly distributed pseudo-random integer between zero and *contention window* (CW) while slot_time is a PHY-dependent value. The value of CW varies between CWmin and CWmax, which are PHY-dependent, but initially the CW is set equal to CWmin. The backoff time can be seen as a waiting time before accessing the medium equal to a random number of time slots. If the medium is sensed to be idle for the duration of one complete time slot, the backoff time is decremented with one slot_time. If the medium becomes busy in the middle of a time slot, the backoff time becomes suspended until the medium is sensed to be idle for the duration of DIFS. Then the backoff procedure resumes and starts decrementing the backoff time again. When the backoff timer reaches zero the station can start transmitting. When the destination receives the data

frame, it waits for the duration of SIFS and responds with an ACK frame to notify the sender of a successful reception. If two or more stations choose the same random number, their backoff timers will reach zero simultaneously resulting in a collision since both begin transmitting. To reduce the probability of choosing the same random number (causing collision), the CW is doubled every time a collision occurs, from its initial value of CWmin until CWmax is reached.

The DCF suffers from the well-known hidden station problem that exists among contention-based protocols. Two stations are hidden from each other if they are out of signal range and thus cannot hear each other. In such a case, the carrier sense mechanisms will not work properly and the hidden stations may both sense the medium free and start transmitting at the same time to a common receiver causing a collision. To deal with the hidden station problem, an optional handshake mechanism with *request to send* (RTS) and *clear to send* (CTS) control frames is used to announce the neighborhood of the impending use of the medium. Before sending a data or a management frame, a station can transmit an RTS frame and await a CTS frame. The control frames contain a duration field that is used to tell the neighbors about the duration of the impending data transmission. When neighbors receive the RTS or CTS frames, they update their NAVs so that they consider the medium busy until the end of the transmission. Thus, collisions occur only on RTS frames and are detected by the absence of a CTS frame. Because of the additional overhead imposed by the handshake mechanism, it is not recommended to be used for short data frames. In other words, the RTS/CTS mechanism is used only if the length of the data or management frames is greater than the threshold dot11RTSThreshold and only for directed frames. A typical frame exchange sequence (if the RTS/CTS mechanism is used) in the DCF is RTS-{SIFS}-CTS-{SIFS}-data-{SIFS}-ACK.

The PCF uses polling to regulate access to the shared medium. Instead of contending for access to the medium, the stations can request to be polled by a *point coordinator* (PC). The PC performs the role of the polling master and operates at an AP (which explains why the operation of the PCF is restricted to infrastructure networks). When PCF is used, the time is divided into a *contention period* (CP), when the DCF manages access to the medium, and

a *contention-free period* (CFP), when the PCF controls the medium access. During a CFP, the PC maintains a polling list of registered stations and polls them according to the list. A station is allowed to start a transmission only after it has been polled. At the nominal start of a CFP, the PC starts sensing the medium. Once the medium is determined to be idle for the duration of PIFS, the PC transmits a beacon frame. Thus, a CFP begins with a beacon frame and is generated by the PC at a defined rate, called the *contention-free repetition rate* (CFPRate). The length of a CFP is controlled by the PC. At the nominal start time of each CFP, stations set their NAV to CFPMaxDuration, which is a parameter that indicates the maximum duration of the CFP. This action prevents stations to contend for access to the medium and thus, they will not initiate a transmission unless they are polled by the PC. Note that the actual duration of a CFP is controlled by the PC, which can terminate a CFP at or before the CFPMaxDuration, based on available traffic and size of the polling list. During the CFP, the stations update their NAV using the CFPDurRemaining value.

To give the PC a higher priority to access the medium, it waits for the duration of PIFS before accessing the medium while the stations must wait for the duration of DIFS, which is longer. At the end of a CFP, the PC transmits a CF-End or CF-End+ACK frame, which will cause the stations receiving the frame resetting their NAV and start contending for access to the medium.

Regarding QoS provisioning, unfortunately both access methods have their limitations. The DCF only provides a best-effort service and all stations contend for access to the medium with the same priority. Thus, the DCF does not provide any differentiation mechanism to give application with QoS requirements better service than other applications. Although the PCF was designed to support time-sensitive applications, it has a few problems that lead to poor QoS performance. Due to this fact, together with the fact that it is an optional access mechanism, the PCF never became commonly implemented. The main problems with the PCF are the following:

- A beacon frame transmission, which indicates the start of a CFP, may be delayed because of a possible transmission in progress from the CP. This will result in a shortened CFP and less time for stations with QoS applications requiring contention-free access to the medium.

- The PCF cannot handle the various QoS requirements of different types of applications because there is no way for the stations to send their requirements to the PC. Furthermore, the scheduling algorithm used by the PC is rather simple, polling stations one after another.
- When a station is polled, it may send a frame between 0-2304 bytes. Therefore, the PC is not able to predict the transmission time of the polled stations and thus, it cannot provide any delay guarantees.

Another QoS problem, which is common for both DCF and PCF, is that there is no admission control mechanism to regulate the usage of the medium. An admission control mechanism is necessary to prevent performance degradation of existing traffic streams when the network becomes overloaded.

2.2 IEEE 802.11e

Due to the QoS limitations of DCF and PCF, there has been a lot of research focusing on the enhancement of the MAC sublayer of 802.11 to provide QoS. To support multimedia applications with QoS requirements, the 802.11 working group started to work on 802.11e to address the QoS issues in the MAC sublayer. Recently, the 802.11e standard was finally ready. The new standard specifies a new coordination function, the *hybrid coordination function* (HCF), which has both a contention-based and a contention-free medium access method. The *enhanced distributed channel access* (EDCA) is contention-based and provides prioritized QoS support while the *HCF controlled channel access* (HCCA) is contention-free and provides support for parameterized QoS. Whereas the EDCA is distributed (like the DCF) and can be used in ad hoc networks, the HCCA is centralized (like the PCF) and thus useful only in infrastructure networks.

One important new feature of the HCF is the introduction of *transmission opportunity* (TXOP), which is a time interval defined by a starting time and a maximum duration. During a TXOP, a station may send several frames as long as the duration of the transmissions does not extend beyond the maximum duration. Since no transmission can violate the TXOP limit, frames that are too large to be transmitted in a single TXOP, must be fragmented into smaller frames. Using ad hoc network configuration, broadcast

and multicast frames are not allowed to be sent more than one at a time.

Another important new feature introduced in the HCF is the concept of *traffic specification* (TSPEC), which describes the QoS requirements of a traffic stream by specifying a set of parameters such as nominal/maximum frame size, minimum/maximum service interval, service start time, minimum/mean/peak data rate, burst size, delay bound, minimum PHY rate and medium time. Most of the above-mentioned parameters are typically set according to the requirements from the application while some are generated locally within the MAC. The parameter minimum/maximum service interval specifies the minimum/maximum time interval between the start of two consecutive TXOPs and service start time specifies the time when the service period starts, i.e. when the station expects to be ready to send frames. Burst size specifies the maximum size of the data burst that can be transmitted at the peak data rate. Medium time is the amount of time admitted to access the medium.

In addition to the QoS enhancements, the 802.11e specification has defined the following optional features to improve the MAC performance: *automatic power-save delivery* (APSD), *block acknowledgement* (BA) and *direct link setup* (DLS). The APSD is an enhancement to the existing power save mechanism in 802.11 and used for delivery of downlink unicast frames to power-saving stations.

The BA mechanism allows a station to aggregate several (up to 64) ACK frames into one instead of sending one ACK after each successfully received data frame. Once the BA mechanism is initialized by an exchange of *add block acknowledgement* (ADDBA) request/response frames, blocks of data frames can be transmitted. When the sender needs an ACK, it sends a *block acknowledgement request* (BlockAckReq) control frame to the receiver which replies with a *block acknowledgement* (BlockAck) control frame acknowledging the successfully received data frames. There are two types of BA mechanisms: immediate BA and delayed BA. If the immediate BA mechanism is used, the BlockAck frame must be sent immediately after a BlockAckReq frame is received. However, if the delayed BA mechanism is used, an ACK frame is used to respond to a BlockAckReq frame and the BlockAck frame can be delayed and sent somewhat later. The delayed BA option is intended to be used by stations with low processing power, i.e. to

give these stations enough time to calculate and prepare the content of the BlockAck frame. A BA setup can be torn down, e.g. when there are no more data frames to be sent, by sending a *delete block ack* (DELBA) frame.

The DLS mechanism allows stations operating in infrastructure mode to transmit frames directly to each other (the same way that stations operating in ad hoc mode communicate) without relying on the AP to forward the frames. DLS requires a handshake process where the station intending to initiate a direct link to another station sends a DLS request action frame to the *QoS access point* (QAP). The QAP relays the request to the other station, which responds with a DLS response. The QAP relays the response back to the station that requested the DLS and finally, in case of successful negotiation, the two stations can communicate with each other directly.

2.2.1 Enhanced Distributed Channel Access (EDCA)

The EDCA is a distributed, contention-based medium access mechanism and an enhanced variant of the DCF. The main problem with the DCF, regarding QoS provisioning, is that it cannot provide any service differentiation since all stations have the same priority, i.e., the same CWmin, CWmax and waiting time before backoff or transmission (equal to DIFS). In addition, the DCF uses one single transmit queue and one channel access function. To overcome this problem, each station using the EDCA mechanism has four *access categories* (ACs) and for each of these there is one transmit queue with an *enhanced distributed channel access function* (EDCAF) that contends for TXOPs independently of the EDCAFs of the other ACs. Thus, each AC behaves like an enhanced and independent DCF contending for medium access. Before entering the MAC sublayer, frames are assigned a *user priority* (UP) and based on these UPs each frame is mapped to an AC according to Table 1. Besides using the UPs, the frames can be mapped to ACs based on frame types. The management type frames, for example, shall be sent from AC_VO (without being restricted by any admission control though) and RTS frames shall use the same AC as the corresponding data or management frame(s). The four different kind of ACs can be used for different kind of traffic: AC_BK for background traffic, AC_BE for best effort traffic, AC_VI for video traffic and AC_VO for voice traffic. Differentiated medium access is realized by varying the contention parameters for each AC:

Table 1: Mapping from UPs to ACs

Priority	User Priority (same as in 802.1D)	Access Category	Designation
lowest	1	AC_BK	Background
	2	AC_BK	Background
	0	AC_BE	Best Effort
	3	AC_BE	Best Effort
	4	AC_VI	Video
	5	AC_VI	Video
	6	AC_VO	Voice
highest	7	AC_VO	Voice

- CWmin[AC] and CWmax[AC] - the minimum and maximum value of the CW used for calculation of the backoff time. These values are variable and no longer fixed per PHY as with the DCF. By assigning low values to CWmin[AC] and CWmax[AC], an AC is given a higher priority.

- *arbitration interframe space number* (AIFSN[AC]) - the number of time slots after a SIFS duration that a station has to defer before either invoking a backoff or starting a transmission. AIFSN[AC] affects the *arbitration interframe space* (AIFS[AC]), which specifies the duration (in time instead of number of time slots) a station must defer before backoff or transmission:
  AIFS[AC] = SIFS + AIFSN[AC] $\times$ slot_time. Thus, by assigning a low value to AIFSN[AC], an AC is given a high priority.

- $TXOP_{limit}[AC]$ - the maximum duration of a TXOP. A value higher than zero means that an AC may transmit multiple frames (if all belong to the same AC since a TXOP is given to an EDCAF in a specific AC and not to a station) as long as the duration of the transmissions does not extend beyond the $TXOP_{limit}[AC]$. A $TXOP_{limit}[AC]$ value equal to zero indicates that only one data or management frame (plus any corresponding RTS/CTS frames) may be sent. Thus, by assigning a high value to the $TXOP_{limit}[AC]$, an AC is given a high priority.

Table 2: Default EDCA parameter set

AC	CWmin	CWmax	AIFSN	TXOP Limit (ms)	
				802.11/b	802.11a/g
AC_BK	CWmin	CWmax	7	0	0
AC_BE	CWmin	CWmax	3	0	0
AC_VI	(CWmin+1)/2-1	CWmin	2	6.016	3.008
AC_VO	(CWmin+1)/4-1	(CWmin+1)/2-1	2	3.264	1.504

Table 2 shows the default EDCA parameters used by the ACs. The values of these parameters can be changed by the QAP announcing the new values in selected beacon frames, any probe response or (re)association response frames. In order to prevent stations to interfere with the operation of APs, it is important to have AIFSN[AC] $\geq$ 2 for stations, resulting in AIFS[AC] $\geq$ DIFS. If the backoff time of more than one EDCAF counts down to zero at the same time, an internal collision occurs within a station. These collisions are resolved such that the frames in the high-priority AC receive the TXOP whereas the frames in the low-priority AC(s) act as if there was an external collision on the wireless medium.

2.2.2 Hybrid Controlled Channel Access (HCCA)

The HCCA is the centralized, contention-free medium access mechanism of the HCF and uses a *hybrid coordinator* (HC), collocated with the QAP, to manage access to the medium. Whenever the medium is sensed to be idle for at least PIFS, the QAP may take control over the medium and start a *controlled access period* (CAP). A CAP is a time period during which a QAP maintains control of the medium to allocate TXOPs to itself or other stations for contention-free medium access. It may span multiple consecutive TXOPs and its duration is limited by the dot11CAPLimit parameter. The QAP has a higher medium access priority than other stations since it needs to wait for only PIFS, which is shorter than DIFS and AIFS that other stations must wait, before accessing the medium. In the HCCA, stations are allowed to reserve TXOPs for their traffic streams by sending an *add traffic stream* (ADDTS) request frame to a QAP. The ADDTS request is a management action frame and contains a TSPEC. The HCCA provides

parameterized QoS, which refers to the capability of allowing applications to specify required QoS parameters in the TSPEC. Upon receiving an ADDTS request, the scheduler of the QAP uses the mandatory set of TSPEC parameters to generate a schedule: mean data rate (ρ), nominal frame size (L) and maximum service interval or delay bound. If both maximum service interval and delay bound are specified in the TSPEC, the maximum service interval is used. Using these mandatory parameters, the scheduler calculates two parameters, the *scheduled service interval* (SI) and the TXOP duration, and use them in the admission control decision:

1. The SI specifies the time interval between the start of two consecutive TXOPs and is the same for all stations. To calculate the SI, the scheduler calculates the minimum `m` of all maximum service intervals for all admitted traffic streams. Then SI equals a value lower than `m` that is a submultiple of the beacon interval. SI must be recalculated when a new traffic stream is admitted that has a maximum service interval smaller than the current SI.

2. The TXOP duration for a given SI specifies the length of the TXOPs given to a traffic stream. To calculate the TXOP duration for an admitted traffic stream, the scheduler uses the mean data rate (ρ) and nominal frame size (L) from the TSPEC, the SI as calculated above, the physical transmission rate (R), the maximum allowable frame size (M), i.e. 2304 bytes, and the overhead in time due to MAC and PHY headers (O). First, the scheduler calculates the number of data frames that arrived at ρ during the SI:

   $$N_i = \left\lceil \frac{SI \times \rho_i}{L_i} \right\rceil$$

   Then the scheduler calculates the TXOP duration as the maximum of the time to transmit N_i frames at R_i plus overhead and the time to transmit one maximum size data frame at R_i plus overhead:

   $$TXOP_i = \max\left(\frac{N_i \times L_i}{R_i} + O, \frac{M}{R_i} + O \right)$$

3. Once the SI and the TXOP duration are calculated, the admission control decision is easy. If there are k admitted traffic streams, a new stream (k+1) can be admitted if it satisfies the following inequality:

$$\frac{TXOP_{k+1}}{SI} + \sum_{i=1}^{k} \frac{TXOP_i}{SI} \leq \frac{T - T_{CP}}{T},$$

where T is the beacon interval and T_{CP} is the duration of the contention period.

To improve the performance of the scheduler, it can for example be modified to generate different SIs for different traffic streams or consider retransmissions while calculating TXOP durations. To improve the performance of the admission control algorithm, it might include UPs in the decision of admitting, retaining or dropping a traffic stream. Thus, the scheduler and the admission control algorithm presented in the 802.11e specification are just examples and any modification can be made in order to improve their performance [3, 4, 5, 6, 7].

2.3 IEEE 802.11 Standards and Recommendations

The standards, recommendations, and task groups in the 802.11 working group have been expanding for a long time. To bring some order into the alphabet soup, the work of all task groups is summarized here [8, 9]. Note that there is no standard or task group called "802.11x", which sometimes is used to denote any current or future 802.11 standard. In addition, 802.11l, 802.11o, and 802.11q are not used.

- IEEE 802.11 - a WLAN standard specifying the PHY layer and the MAC sublayer. It specifies three PHY layer options with data rates of 1-2 Mbit/s: IR at 850-950 nm, FHSS and DSSS at 2.4 GHz. (1997)
- IEEE 802.11a - a PHY layer standard at the 5 GHz band providing data rates up to 54 Mbit/s. (1999)
- IEEE 802.11b - a PHY layer standard extending 802.11 PHY (DSSS) at the 2.4 GHz band to support 5.5 and 11 Mbit/s. (1999)

- IEEE 802.11c - a network interoperability standard that deals with bridge operation procedures. A bridge is a device that connects local area networks with a similar or identical MAC protocol. This standard is included in the IEEE 802.1D standard. (2003)
- IEEE 802.11d - a "global harmonization" standard that defines PHY layer requirements to satisfy regulatory domains since the allowed frequencies, power levels and signal bandwidth may differ between different countries. Thus, the specification eliminates the need for manufacturing country-specific products. (2001)
- IEEE 802.11e - a QoS standard that defines enhancements to the 802.11 MAC sublayer. In addition to providing QoS, the standard improves the MAC performance by specifying functions such as *block acknowledgement* (BA), *direct link setup* (DLS) and *automatic power-save delivery* (APSD). (2005)
- IEEE 802.11F - an AP interoperability `recommendation` that defines an extension (Inter-Access Point Protocol) to 802.11 to simplify wireless communications among APs from different vendors. In other words, 802.11F facilitates roaming of a stations from one AP to another. (2003)
- IEEE 802.11g - a PHY layer standard extending 802.11 PHY to support data rates up to 54 Mbit/s at the 2.4 GHz band. This standard is backward compatible with 802.11b. (2003)
- IEEE 802.11h - a spectrum and transmit power management standard that allows 802.11a devices to co-exist with devices using other standards operating at the same 5 GHz frequency band. In the European Union, the 5 GHz band is used for e.g. satellite communication, so the standard uses a dynamic frequency selection mechanism to prevent selection of congested channels. The transmit power control function of the standard adjusts the power to the EU requirements. (2004)
- IEEE 802.11i - a security standard that defines enhancements to the 802.11 MAC sublayer. The standard supersedes the *wired equivalent privacy* (WEP) algorithm, which was specified in the original standard and had severe security weaknesses. (2004)

- IEEE 802.11j - a standard that specifies the 4.9 - 5 GHz operation in Japan. It defines methods that let APs move to new frequencies or change the channel width for better performance and capacity and not to interfere with other wireless devices using the same frequency band in Japan. (2004)

- IEEE 802.11k - an upcoming standard for radio resource measurement that is intended to improve the way the traffic is distributed within a WLAN. Instead of connecting to the AP with the strongest signal, a station will also consider the load of the existing APs. In other words, 802.11k provides information to discover the best available AP. Thus, a station may connect to an AP with a weaker signal but that is underutilized; thereby increasing the overall performance in the WLAN.

- IEEE 802.11l - not used

- IEEE 802.11m - ongoing work to correct the editorial and technical issues in the 802.11 family specifications.

- IEEE 802.11n - an upcoming standard that is expected to be the successor of 802.11g. The goal is to improve the PHY layer and the MAC sublayer to enable higher throughput (several hundreds of Mbit/s), partly by adding *multiple-input multiple-output* (MIMO) technology, i.e. by using multiple transmitter and receiver antennas.

- IEEE 802.11o - not used

- IEEE 802.11p - an upcoming standard that defines enhancements required to support *Intelligent Transportation Systems* (ITS) applications such as toll collection, vehicle safety services, and commerce transactions via cars. The goal is to enable communication between vehicles and roadside APs or other vehicles. This includes data exchange between high-speed vehicles and between these vehicles and the roadside infrastructure in the licensed ITS band of 5.9 GHz. The standard is sometimes referred to as *Wireless Access for the Vehicular Environment* (WAVE).

- IEEE 802.11q - not used

- IEEE 802.11r - an upcoming standard for fast roaming of stations. This standard will enable connectivity aboard vehicles in motion, with fast roaming from one AP to another. Furthermore, it will facilitate the deployment of IP-based telephony over 802.11-enabled phones.

- IEEE 802.11s - an upcoming standard for mesh networking that defines extensions to the 802.11 MAC sublayer. The purpose of the project is to provide a protocol for auto-configuring paths between APs over multi-hop topologies.

- IEEE 802.11T - an upcoming `recommendation` that specifies test methods and metrics to measure and evaluate the performance of 802.11-based devices and networks.

- IEEE 802.11u - an upcoming standard extending both the PHY layer and the MAC sublayer to enable interworking with external networks (e.g. the Internet or cellular networks). Since 802.11-based WLANs has become more widespread, this work started to solve the problems related to the connection of a WLAN to an external network in a standardized manner.

- IEEE 802.11v - an upcoming standard extending both the PHY layer and the MAC sublayer to provide wireless network management for stations.

- IEEE 802.11w - an upcoming standard with focus on security of management frames by enhancing the MAC sublayer. The 802.11i standard addresses the security of only data frames so the WLANs are still vulnerable to malicious attacks because of the unprotected management frames.

- IEEE 802.11x - not used

- IEEE 802.11y - an upcoming standard extending the PHY layer and using the 3.65-3.7 GHz band, which previously was reserved for fixed satellite service networks. The amendment will provide a standardized interference avoidance mechanism and streamline the adoption of new frequencies in the future.

3 The Network Layer: Routing in Ad Hoc Networks

The mobile stations in a MANET need a routing protocol to route a packet from the source, through possible intermediate stations, to the destination. These protocols can be classified into two main classes[2]: proactive and reactive routing protocols. In proactive routing, the routing table of every station is updated periodically. Thus, the delay before sending a packet is minimal but at the cost of increased routing overhead. On the contrary, reactive routing is performed on-demand, i.e. the sending station searches for a route to the destination station only when it needs to communicate with it. Hence, the routing overhead is minimized but the route discovery process may result in considerable delay.

This section gives an overview of some common ad hoc routing protocols. In addition, it discusses the issue of Internet access in ad hoc networks.

3.1 Routing Protocols

The issue of routing in a MANET has been a challenging task for a long time since the stations are mobile and free to move randomly. Consequently, many routing protocols have been proposed and there has been a lot of research focusing on routing protocols for MANETs. Among these, the MANET working group [14] in the *Internet Engineering Task Force* (IETF) chose four protocols to go on with. These protocols are *Ad hoc On-Demand Distance Vector* (AODV) [15], *Dynamic Source Routing* (DSR) [16], *Optimized Link State Routing* (OLSR) [17] and *Topology Dissemination Based on Reverse-Path Forwarding* (TBRPF) [18]. Two of these protocols, AODV and DSR, are reactive while the other two, OLSR and TBRPF, are proactive. DSR is currently an Internet draft submitted for publication as an experimental *request for comment* (RFC), while AODV, OLSR and TBRPF are already published as experimental RFCs. Based on the work and experience on these protocols, the working group aims at developing two standard routing protocol specifications: one reactive MANET protocol and one proactive MANET protocol. If the reactive and the proactive protocol turn out to have many parts in common, the working group may decide to continue

[2]There are others ways of categorizing ad hoc routing protocols [10, 11, 12, 13].

with a converged approach. The goal is that the final protocol(s) supports both IPv4 and IPv6 and that it will address routing security requirements. The work with the reactive MANET protocol has resulted in the *Dynamic MANET On-demand* (DYMO) [19] routing protocol while *Optimized Link State Routing version 2* (OLSRv2) [20] is the candidate for the proactive MANET protocol.

3.1.1 Ad hoc On-Demand Distance Vector (AODV)

AODV is a popular, reactive routing protocol, which guarantees loop-free routes by using sequence numbers that indicate how new a route is. AODV requires each station to maintain a routing table containing one route entry for each destination that the station is communicating with. Each route entry keeps track of certain fields such as: `Destination IP Address`, `Destination Sequence Number`, `Next Hop` (a neighbor station chosen to forward packets to the destination), `Hop Count` (the number of hops needed to reach the destination) and `Lifetime` (the expiration or deletion time of the route).

Whenever a station needs a route to a station for which it does not have a route, it starts the route discovery process by broadcasting a *route request* (RREQ) to all its neighbours. A neighbour receiving a RREQ unicasts a *route reply* (RREP) back to the source if it is either the destination or if it has an unexpired route to the destination. If none of these two cases is satisfied, the neighbour rebroadcasts (forwards) the RREQ. To prevent dissemination of duplicated RREQs, stations keep a cache where they store the source IP address and ID of the received RREQs during a short period of time. If the stations receive another RREQ with the same source IP address and RREQ ID during this period, it is discarded.

When searching for a route to the destination, the source may use the `expanding ring search` technique to prevent unnecessary network-wide dissemination of RREQs. This is done by controlling the value of the *time to live* (TTL) field in the IP header, which defines the maximal number of hops a RREQ can move through the network.

When a link break occurs, the station upstream of the break invalidates all its routes that use the broken link. Then, the station broadcasts a *route error* (RERR) to its neighbors. The RERR contains a list of each destination

that has become unreachable due to the link break. Upon reception of a RERR, a station invalidates possible routes to the unreachable destinations and broadcasts a new RERR to its neighbors. This process continues until the source receives a RERR. The source invalidates the listed routes and reinitiates a route discovery process if needed.

3.1.2 Dynamic Source Routing (DSR)

DSR is a purely reactive routing protocol that uses `source routing` to send packets. Source routing means that the header of each data packet carries the complete list of intermediate stations to the destination. Consequently, the overhead caused by DSR increases but on the other hand, its entirely reactive behavior means that DSR requires no periodic packets of any kind at any layer within the network; thereby decreasing the overhead. An advantage of source routing is that stations forwarding or overhearing data packets, can cache the routing information included in the header of the data packets for future use. Other advantages of using source routing is that it guarantees loop-free routes and supports the use of multiple routes to any destination. The support for multiple routes results in fast reaction to routing failures since a station can try another cached route.

The route discovery process is initiated only if a station needs a route that cannot be found in the route cache. The station broadcasts a `route request`, which contains the address of the source and the destination, and a unique identification number. An intermediate station that receives a route request searches its route cache for a route to the destination. If no route is found, it appends its address to the route record of the route request and forwards the message to its neighbors. The message propagates through the network until it reaches either the destination or an intermediate station with a route to the destination. Then a `route reply`, containing the proper hop sequence for reaching the destination, is generated and unicast back to the source station. To limit the number of route requests propagated, a station discards route requests that it has received recently with the same identification number and destination address or if its address is already present in the route record of the route request.

Route maintenance is used to handle route breaks. When a station encounters a transmission problem at its data link layer, it removes the

route from its route cache and generates a `route error`. The route error is sent to each station that has sent a packet routed over the broken link. When a station receives a route error, it removes the hop in error from its route cache.

3.1.3 Optimized Link State Routing Protocol (OLSR)

OLSR is a proactive routing protocol developed for MANETs. The routing tables are kept updated by regularly exchanging topology information; thus, routes are maintained for all known destinations at all times. OLSR substantially reduces the large message overhead, which usually is associated with classical flooding mechanisms, by reducing redundant retransmissions. This is done by allowing only some selected stations, called *multipoint relays* (MPRs), to forward the broadcast messages during the flooding process. Each station in the network selects a subset of its neighbors as MPRs. To avoid problems associated with uni-directional links, the candidates for MPRs must have a bi-directional link to the selecting station. Another optimization is achieved by minimizing the set of links flooded in the network. As opposed to the classic link state algorithm, a station declares only the MPR links to its neighbors, rather than all links to all neighbors.

The concept of relaying in OLSR has been inherited from the MAC protocol *High Performance Radio Local Area Network* (HIPERLAN)[3], which is standardized by the *European Telecommunications Standards Institute* (ETSI).

3.1.4 Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)

TBRPF is a proactive, link-state routing protocol designed for MANETs. It provides hop-by-hop routing along shortest paths to each destination. Each station running TBRPF computes a source tree (providing paths to all reachable stations) based on partial topology information stored in its topology table, using a modification of Dijkstra's algorithm. To minimize overhead,

[3]Before the IEEE 802.11 standards became a de facto standard for WLANs, HIPERLAN was discussed and considered as a competitor. However, today we know that HIPERLAN did not become widely used, although it had many benefits compared to 802.11, which had many unsolved problems that were dealt with in later amendments.

each station reports only part of its source tree to neighbors. TBRPF uses a combination of periodic and differential updates to keep all neighbors informed of the reported part of its source tree. Each station also has the option to report additional topology information (up to the full topology), to provide improved robustness in highly mobile networks. TBRPF performs neighbor discovery using "differential" HELLO messages that report only changes in the status of neighbors. This results in HELLO messages that are much smaller than those of other link-state routing protocols such as OSPF.

3.1.5 Dynamic MANET On-demand (DYMO)

DYMO is a reactive routing protocol and the main candidate for the upcoming reactive MANET protocol. It is based on the work and experience on previous reactive routing protocols, especially AODV and DSR. To ensure loop-free routes, DYMO uses the same technique as in AODV, namely sequence numbers. The route discovery process is done using RREQs and RREPs, while RERRs are used to maintain routes. The stations monitor links on active routes through e.g. link-layer feedback, hello messages, neighbor discovery and route timeouts. The DYMO draft specifies the base specification but by using the generalized MANET packet and message format [21], it is prepared for extensions. A new feature of DYMO is that it is being prepared for allowing the mobile stations in the MANET to have Internet access.

3.1.6 Optimized Link State Routing Protocol version 2 (OLSRv2)

As the name implies, the OLSRv2 is very similar to the OLSR protocol described earlier. The protocol has the same key optimization techniques as in the OLSR protocol, i.e., using MPRs responsible for forwarding control traffic that must be flooded in the entire network, and maintaining partial link-state information reducing the number and size of the network-wide broadcasts.

The main differences compared to the first version are more flexible signaling framework and some simplifications on the messages. Moreover, the OLSRv2 has a uniform treatment of both IPv4 and IPv6.

3.2 Internet Access

Although an autonomous, stand-alone MANET is useful in many cases, a MANET connected to the Internet is much more desirable. This interconnection is achieved by using gateways, which act as bridges between a wireless MANET and the wired Internet. Before a mobile station can communicate with an Internet host, it needs to find a route to a gateway. Thus, a gateway discovery mechanism is required.

The ad hoc routing protocols were designed for communication within an autonomous MANET. Therefore, a routing protocol needs to be modified in order to achieve routing between a mobile device in a MANET and a fixed device in a wired network (e.g. the Internet). To achieve this network interconnection, gateways that understand the protocols of both the MANET protocol stack and the TCP/IP suite are needed. Thus, a gateway acts as a bridge between a MANET and the Internet and all communication between the two networks must pass through the gateway.

3.2.1 Gateway Discovery

The question of whether the registration process with the gateway should be initiated by the gateway (proactive method), by the mobile station (reactive method) or by mixing these two approaches (hybrid proactive/reactive method) has been studied in this thesis.

The proactive gateway discovery is initiated by the gateway itself. The gateway periodically broadcasts a gateway advertisement message to inform all mobile stations residing in its transmission range about its presence. The time between two consecutive advertisements must be chosen with care so that the network is not flooded too frequently. Upon receipt of the advertisement, the mobile stations update their routing tables and forward the advertisement to other mobile stations. Redundant retransmissions can be avoided by using identification numbers for the advertisements. Although the problem of duplicated broadcast messages can be solved, one disadvantage remains. This disadvantage, which is general for all proactive approaches, is the fact that the message is flooded through the whole MANET periodically. This a very costly operation, especially in MANETs with limited resources, such as power and bandwidth. However, the advantage is

that the need for a time-consuming route discovery process is eliminated since the routes to the gateways are updated periodically.

The reactive gateway discovery is initiated by a mobile station that determines that it needs to access the Internet. The mobile station broadcasts a RREQ with an 'I'-flag set, i.e. a RREQ_I, which is processed only by the gateways in the MANET. Intermediate mobile stations that receive the message just forward it. Upon receipt of a RREQ_I, a gateway unicasts back a RREP with an 'I'-flag set, i.e. a RREP_I which, among other things, contains the IP address of the gateway.

The advantage of this approach is that RREQ_Is are sent `only` when a mobile station needs the information about reachable gateways. Hence, periodic flooding of the complete MANET, which has obvious disadvantages is prevented. The disadvantage of reactive gateway discovery is the delay caused by the route discovery process.

To minimize the disadvantages of the proactive and reactive gateway discovery methods, the two approaches can be combined. This results in a hybrid proactive/reactive method for gateway discovery. For mobile stations in a certain area around a gateway, proactive gateway discovery is used. Mobile stations residing outside this area use reactive gateway discovery to obtain information about the gateway. The size of the area is defined by a maximum number of hops the advertisement can move through the MANET and it can be adjusted.

4 List of Papers

The following papers are included in the thesis:

Paper I

Ali Hamidian, Ulf Körner and Anders Nilsson:
Performance of Internet Access Solutions in Mobile Ad Hoc Networks, Dagstuhl-Workshop "Mobility and Wireless in Euro-NGI", G. Kotsis and O. Spaniol (Eds.): Mobile and Wireless Systems, LNCS 3427, pp. 189-201, 2005.

Paper II

Micro Mobility and Internet Access Performance for TCP Connections in Ad Hoc Networks

Anders Nilsson, Ali Hamidian and Ulf Körner, Nordic Teletraffic Seminar 17, Oslo, Norway, 2004.

Paper III

An Enhancement to the IEEE 802.11e EDCA Providing QoS Guarantees

Ali Hamidian and Ulf Körner, Telecommunication Systems journal, Vol. 31, Issue 2-3, 2006.

Paper IV

Providing QoS Guarantees in Ad Hoc Networks through EDCA with Resource Reservation

Ali Hamidian, Internal report, Department of Communication Systems, Lund University, CODEN: LUTEDX (TETS-7217)/1-8/(2006) & local 6, 2006.

The following papers do not appear in the thesis:

Paper V

Evaluation of Solutions for Internet Access in Mobile Ad Hoc Networks

Ali Hamidian, Ulf Körner and Anders Nilsson, Nordic Teletraffic Seminar 17, Oslo, Norway, August 2004.

Paper VI

Towards a Solution Providing QoS in Ad Hoc Networks

Ali Hamidian and Ulf Körner, The 19th International Teletraffic Congress (ITC19), Beijing, China, September 2005.

Paper VII

QoS in Ad Hoc Networks

Ulf Körner and Ali Hamidian, Aalborg University - Keio University Joint Workshop for Broadband Wireless Communications, Aalborg, September 2005.

References

[1] ANSI/IEEE Std 802.11. *Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[2] IEEE P802.11e/D13.0. *Part11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 7: Medium Access Control (MAC) Quality of Service (QoS) Enhancements*, January 2005.

[3] A. Grilo, M. Macedo and M. Nunes. A Scheduling Algorithm for QoS Support in IEEE 802.11e Networks. *IEEE Wireless Communications Magazine*, 2003.

[4] L. Romdhani, Q. Ni, T. Turletti. Adaptive EDCD: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad Hoc Networks. Proceedings of IEEE WCNC, March 2003.

[5] P. Ansel, Q. Ni, T. Turletti. FHCF: A Fair Scheduling Scheme for 802.11e WLAN. Research report number 4883, INRIA Sophia Antipolis, July 2003.

[6] P. Ansel, Q. Ni, T. Turletti. An Efficient Scheduling Scheme for IEEE 802.11e. Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, March 2004.

[7] D. Skyrianoglou, N. Passas and A. Salkintzis. Traffic Scheduling in IEEE 802.11e Networks Based on Actual Requirements. Mobile Venue '04 Mobile Location Workshop Athens, May 2004.

[8] W. Stallings. IEEE 802.11: Wireless LANs from a to n. IEEE Computer Society, September - October 2004.

[9] Wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/802.11`, Page accessed November 2005.

[10] D. Lang. A comprehensive overview about selected Ad Hoc Networking Routing Protocols. Department of Computer Science, Technische Universitat Munchen, Germany, March 2003.

[11] X. Zou, B. Ramamurthy, S. Magliveras. Routing Techniques in Wireless Ad Hoc Networks - Classification and Comparison, 2002.

[12] X. Hong, K. Xu and M. Gerla. Scalable Routing Protocols for Mobile Ad Hoc Networks. *IEEE Network*, July/August 2002.

[13] L. M. Feeney. A Taxonomy for Routing Protocols in Mobile Ad Hoc Networks. Technical report, Swedish Institute of Computer Science, October 1999. Technical Report T99/07.

[14] The homepage of the MANET working group. `www.ietf.org/html.charters/manet-charter.html`, Page accessed November 2005.

[15] C. Perkins, E. M. Belding-Royer and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. Experimental RFC 3561.

[16] D. B. Johnson, D. A. Maltz, Y. Hu and J. G. Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). IETF Internet Draft.

[17] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum and L. Viennot. Optimized Link State Routing Protocol. Experimental RFC 3626.

[18] R. Ogier, M. Lewis and F. Templin. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). Experimental RFC 3684.

[19] I. Chakeres and C. Perkins. Dynamic MANET On-demand (DYMO) Routing. IETF Internet draft.

[20] T. Clausen. The Optimized Link-State Routing Protocol version 2. IETF Internet draft.

[21] C. Dearlove T. Clausen and J. Dean. Generalized MANET Packet/Message Format. IETF Internet draft.

Performance of Internet Access Solutions in Mobile Ad Hoc Networks

Ali Hamidian, Ulf Körner and Anders Nilsson

Abstract

Although an autonomous mobile ad hoc network (MANET) is useful in many scenarios, a MANET connected to the Internet is more desirable. This interconnection is achieved by using gateways, which act as bridges between a MANET and the Internet. Before a mobile node can communicate with an Internet host it needs to find a route to a gateway. Thus, a gateway discovery mechanism is required. In this paper the MANET routing protocol Ad hoc On-Demand Distance Vector (AODV) is extended to achieve the interconnection between a MANET and the Internet. Moreover, the paper investigates and compares three approaches for gateway discovery. The question of whether the configuration phase with the gateway should be initiated by the gateway, by the mobile node or by mixing these two approaches is being discussed. We have implemented and simulated these three methods and we discuss the advantages and disadvantages of the three alternatives.

1 Introduction

A mobile ad hoc network (MANET) is an autonomous network that can be formed without need of any established infrastructure or centralized administration. It normally consists of mobile nodes, equipped with a wireless interface, that communicate with each other. Because these kinds of networks are very spontaneous and self-organizing, they are expected to be very useful. It is also highly likely that a user of the network will have the need to connect to the Internet.

The Internet Engineering Task Force (IETF) has proposed several routing protocols for MANETs, such as Ad hoc On-Demand Distance Vector (AODV) [1], Dynamic Source Routing (DSR) [2], Optimized Link State Routing Protocol (OLSR) [3] and Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [4]. However, these protocols were designed for communication within an autonomous MANET, so a routing protocol needs to be modified in order to achieve routing between a mobile device in a MANET and a host device in a wired network (e.g. the Internet). To achieve this network interconnection, gateways that understand not only the IP suite, but also the MANET protocol stack, are needed. Thus, a gateway acts as a bridge between a MANET and the Internet and all communication between the two networks must pass through any of the gateways.

The AODV routing protocol is one of the most developed and implemented routing protocols investigated by the IETF MANET working group. In this work AODV has been modified to achieve routing of packets towards a wired network [5]. Although AODV was used in this study, our approach can be applied to any reactive MANET routing protocol and with some modifications to proactive MANET routing protocols as well.

This paper evaluates three approaches for gateway discovery. An interesting question is whether the configuration phase with the gateway should be initiated by the gateway (proactive method), by the mobile node (reactive method) or by mixing these two approaches. We have implemented these three methods in Network Simulator 2 (ns-2) [6] and compare them by means of simulation. We also discuss the advantages and disadvantages of the three alternatives.

The remainder of this paper is organized as follows: Section 2 gives an

overview of AODV and presents an Internet access solution for MANETs. Section 3 investigates three gateway discovery strategies. The simulation results are presented and discussed in Sect. 4. Finally, Sect. 5 concludes this paper and gives some directions for future work.

2 Protocol Description

As mentioned above, AODV was originally designed for routing packets within a MANET and not between a MANET and a wired network. In order to achieve routing across the network interconnection, the routing protocol needs to be modified. After giving an overview of AODV, we present a solution, which is referred to as AODV+ [7], where AODV is extended to provide Internet access for mobiles node in a MANET.

2.1 Ad hoc On-Demand Distance Vector (AODV)

Ad hoc On-Demand Distance Vector (AODV) is a reactive MANET routing protocol [1], where the reactive property implies that a mobile node requests a route only when it needs one. Consequently, the node maintains a routing table containing route entries only to destinations it is currently communicating with. Each route entry contains a number of fields such as *Destination IP Address*, *Next Hop* (a neighbor node chosen to forward packets to the destination), *Hop Count* (the number of hops needed to reach the destination) and *Lifetime* (the expiration or deletion time of the route). AODV guarantees loop-free routes by using sequence numbers that indicate how fresh a route is.

2.1.1 Route Discovery

Whenever a node (source) determines that it needs a route to another node (destination) it broadcasts a *route request* (RREQ) message and sets a timer to wait for the reception of a *route reply* (RREP). A node that receives a RREQ creates a *reverse route entry* for the source in its routing table. Then it checks to determine whether it has received a RREQ with the same Originator IP Address and RREQ ID within the last PATH_DISCOVERY_TIME. If such a RREQ has been received, the node discards the newly received

RREQ in order to prevent duplicated RREQs from being forwarded. If the RREQ is not discarded the node continues to process it as follows: If the node is either the destination or if it has an unexpired route to the destination, it unicasts a RREP back to the source; otherwise it rebroadcasts the RREQ. If a RREP is generated, any intermediate node along the path back to the source creates a *forward route entry* for the destination in its routing table and forwards the RREP towards the source.

If the source does not receive any RREP before the RREQ timer expires, it broadcasts a new RREQ with an increased time to live (TTL) value. This technique is called *expanding ring search* and continues until either a RREP is received or a RREQ with the maximum TTL value is broadcasted. Broadcasting a RREQ with the maximum TTL value is referred to as a *network-wide search* since the RREQ is disseminated throughout the MANET. If a source performs a network-wide search without receiving any corresponding RREP, it may try again to find a route to the destination, up to a maximum of RREQ_RETRIES times after which the session is aborted.

2.1.2 Route Maintenance

When an active link breaks, the node upstream of the break invalidates all its routes that use the broken link. Then, the node broadcasts a *route error* (RERR) message that contains the IP address of each destination that has become unreachable due to the link break. Upon reception of such a RERR message, a node searches its routing table to see if it has any route(s) to the unreachable destination(s) (listed in the RERR message) that uses the originator of the RERR as the next hop. If such routes exist, they are invalidated and the node broadcasts a new RERR message. This process continues until the source receives a RERR message. The source then invalidates the listed routes as previously described and initiates a route discovery process if needed.

2.2 Internet Access for Mobile Ad Hoc Networks

Whenever a mobile node is about to communicate with a fixed wired node, it searches its routing table for a route towards the destination. If a route is found, the communication can be established. Otherwise, the mobile

node starts a route discovery process by broadcasting a RREQ message as described previously.

When an intermediate mobile node receives a RREQ message, it searches its routing table for a route towards the wired destination. If a route is found, the intermediate node would normally send a RREP back to the originator of the RREQ. But in that case, the source would think that the destination is a mobile node that can be reached via the intermediate node. It is important that the source knows that the destination is a fixed node and not a mobile node, because these are sometimes processed differently. In our solution, this problem has been solved by preventing the intermediate node to send a RREP back to the originator of the RREQ if the destination is a wired node. Instead, the intermediate node updates its routing table and rebroadcasts the received RREQ message. To determine whether the destination is a wired node or not, an intermediate node consults its routing table. If the next hop address of the destination is a default route (see Table 1), the destination is a wired node. Otherwise, the destination is a mobile node or a gateway.

Since neither the fixed node nor the mobile nodes in the MANET can reply to the RREQ, it is rebroadcasted until its TTL value reaches zero. When the timer of the RREQ expires, a new RREQ message is broadcasted with a larger TTL value. However, since the fixed node cannot receive the RREQ message (no matter how large the TTL value is) the source will never receive the RREP message it is waiting for. This problem has been solved by letting the source assume the destination is a fixed node if a network-wide search has been done without receiving any corresponding RREP. In that case, the source must find a route to a gateway (if it does not have one already, see Sect. 3) and send its data packets towards the gateway, which will forward them towards the fixed node.

It should be mentioned that when using the expanding ring search, a considerable route discovery delay will occur if the destination is a fixed node. Modifying the parameters involved in the expanding ring search technique (such as TTL_START and TTL_THRESHOLD) can decrease the route discovery delay if the destination is a fixed node. However, the modification can also result in increased routing overhead if the destination is a mobile node. The modification could for example be to increase TTL_START.

Assuming the destination is a fixed node, increasing TTL_START would result in less number of broadcasted RREQs (and consequently less delay) before the source assumes that the destination is a fixed node. Thus, different approaches are preferable depending on whether a mobile node is to communicate mostly with the MANET or the Internet.

2.2.1 Handover

Due to the multihop nature of a MANET, there might be several reachable gateways for a mobile node at some point of time. If a mobile node receives gateway advertisements from more than one gateway, it has to decide which gateway to use for its connection to the Internet. In this solution a mobile node initiates a handover when it receives an advertisement from a gateway that is closer (in terms of number of hops) than the one it is currently registered with. Apart from the hop count, there are other potential criteria that could be used to determine whether a handover is needed or not; e.g. geographical distance, radio signal level, signal delay and direction of node movement [8]. However, the question of a suitable metric for route selection is a general routing issue in MANETs.

2.2.2 Gateway Operation

When a gateway receives a RREQ, it consults its routing table for the destination IP address specified in the RREQ message. If the address is not found, the gateway sends a RREP with an 'I' flag (RREP_I) back to the originator of the RREQ. On the other hand, if the gateway finds the destination in its routing table, it unicasts a RREP as normal, but may also optionally send a RREP_I back to the originator of the RREQ. This will provide the mobile node a default route although it has not requested it. If the mobile node is to communicate with the Internet later, the default route is already established, and another time consuming gateway discovery process can be avoided.

2.2.3 Routing Table Management

Another issue that must be taken into consideration is how the routing table should be updated after a network-wide search without receiving any

Table 1: The routing table of a mobile node after creating a route entry for a fixed node

Destination Address	Next Hop Address
Fixed node	Default
Default	Gateway
Gateway	IMN

corresponding RREP. Once the source has determined that the destination is a fixed node located on the Internet, it has to create a route entry for the fixed node in its routing table. If the route entry for the fixed destination would not be created in the routing table, the source would not find the address to the fixed node in its routing table when the next data packet would be generated and hence, the source would have to do another time consuming network-wide search.

Table 1 shows how the routing table of a mobile node should look like after creation of a route entry for a fixed node. The first entry tells the node that the destination is a fixed node since the next hop is specified by the default route. The second entry specifies which gateway the node has chosen for its Internet connection. The last entry gives information about the next hop towards the gateway.

Another challenge is how to setup the routing table of an intermediate mobile node (IMN) chosen to forward data packets towards the gateway. Since the forward route entries are created for the gateway (the source of the RREP_I) and not for the fixed node, which is the final destination of the data packets, IMN will not find any valid route for the fixed node when it receives data packets from the source. Therefore, it would normally drop the data packets because it does not know how to forward them. In our solution, if IMN does not find a valid route to the destination and if the destination is a fixed node, it creates a (or updates the) route entry for the fixed node in its routing table and forwards the data packets towards the gateway.

3 Gateway Discovery

An interesting question to investigate is whether the configuration phase with the gateway should be initiated by the gateway (proactive method), by the mobile node (reactive method) or by mixing these two approaches (hybrid proactive/reactive method). In the following, the mechanisms of these three approaches are discussed.

3.1 Proactive Gateway Discovery

The proactive gateway discovery is initiated by the gateway itself. The gateway periodically broadcasts a *gateway advertisement* (GWADV) message with the period determined by ADVERTISEMENT_INTERVAL [7, 9]. The advertisement period must be chosen with care so that the network is not flooded unnecessarily.

The mobile nodes that receive the advertisement, create a (or update the) route entry for the gateway and then rebroadcast the message. To assure that all mobile nodes within the MANET receive the advertisement, the number of retransmissions is determined by NET_DIAMETER defined by AODV. However, this will lead to enormously many unnecessary duplicated advertisements. A conceivable solution that prevents duplicated advertisements, is to introduce a "GWADV ID" field in the advertisement message format similar to the "RREQ ID" field in the RREQ message format (see Sect. 2.1.1).

It is worth mentioning that the mobile nodes randomize their rebroadcasting of the GWADV message in order to avoid synchronization and subsequent collisions with other nodes' rebroadcasts.

The advantage of this approach is that there is a chance for the mobile node to initiate a handover before it looses its Internet connection. The disadvantage is that since a control message is flooded through the whole MANET periodically, limited resources in a MANET, such as power and bandwidth, will be used a lot.

3.2 Reactive Gateway Discovery

The reactive gateway discovery is initiated by a mobile node that is to create or update a route to a gateway. The mobile node broadcasts a RREQ with

an 'I' flag (RREQ_I) to the ALL_MANET_GW_MULTICAST [5] address, i.e. the IP address for the group of all gateways in a MANET. Thus, only the gateways are addressed by this message and only they process it. Intermediate mobile nodes that receive a RREQ_I are not allowed to answer it, so they just rebroadcast it. When a gateway receives a RREQ_I, it unicasts back a RREP_I which, among other things, contains the IP address of the gateway.

The advantage of this approach is that control messages are generated only when a mobile node needs information about reachable gateways. Hence, periodic flooding of the whole MANET, which has obvious disadvantages as discussed in Sect. 3.1, is prevented. The disadvantage of reactive gateway discovery is that a handover cannot be initiated before a mobile node looses its Internet connection. As a consequence, a situation can occur where a mobile node uses a gateway for its Internet connection although there are other gateways that are closer.

3.3 Hybrid Gateway Discovery

To minimize the disadvantages of the proactive and reactive strategies, they can be combined into a hybrid proactive/reactive method for gateway discovery. For mobile nodes in a certain range around a gateway, proactive gateway discovery is used while mobile nodes residing outside this range use reactive gateway discovery to obtain information about the gateway.

The gateway periodically broadcasts a GWADV message. Upon receipt of the message, the mobile nodes update their routing table and then rebroadcast the message. The maximum number of hops a GWADV can move through the MANET is determined by ADVERTISEMENT_ZONE. This value defines the range within which proactive gateway discovery is used. When a mobile node residing outside this range needs gateway information, it broadcasts a RREQ_I to the ALL_MANET_GW_MULTICAST address. Mobile nodes receiving the RREQ_I just rebroadcast it. When a gateway receives a RREQ_I, it sends a RREP_I towards the source.

Thus, the proactive gateway discovery method is used to handle the mobile nodes less or equal than ADVERTISEMENT_ZONE hops away from the gateway and the reactive gateway discovery method is used to handle the mobile nodes more than ADVERTISEMENT_ZONE hops away from the gateway.

4 Performance Evaluation

In order to evaluate the performance of the three gateway discovery methods, the network simulator ns-2 has been used. First, the source code of AODV in ns-2 was extended to provide Internet access to mobile nodes. Then the three gateway discovery methods were implemented. This code, which is referred to as AODV+, has been contributed [7] to ns-2 and is free to be downloaded and used by everyone. The latest version of ns-2 (ns-2.27) has been used in this study.

4.1 Simulation Scenario

The studied scenario consists of 60 mobile nodes, two gateways, two routers and two hosts. The topology is a rectangular area with 1300 m length and 800 m width. A rectangular area was chosen in order to force the use of longer routes between nodes, compared to a square area with the same node density. The two gateways were placed on each side of the area; their x- and y-coordinates in metres are (200,500) and (1100,500). All simulations were run for 1000 seconds of simulation time. Since we were interested in studying the behaviour of the network in steady state, the first 100 seconds of the simulation were ignored.

Ten of the 60 mobile nodes are constant bit rate (CBR) traffic sources sending data packets with a size of 512 bytes, to one of the two hosts, chosen randomly. The sources are distributed randomly within the MANET. The transmission range of the mobile nodes is 250 metres.

A screenshot of the simulation scenario is shown in Fig. 1. The 60 small circles represent the mobile nodes. The two hexagonal nodes at each side of the figure are the gateways and the four square nodes are the two hosts and the two routers.

4.2 The Mobility Model

The mobile nodes move according to an improved version of the commonly used random waypoint model. It has been shown that the original random waypoint model can generate misleading results [10]. With the improved random waypoint model the mobile node speed reaches steady state after a quick warm-up period.

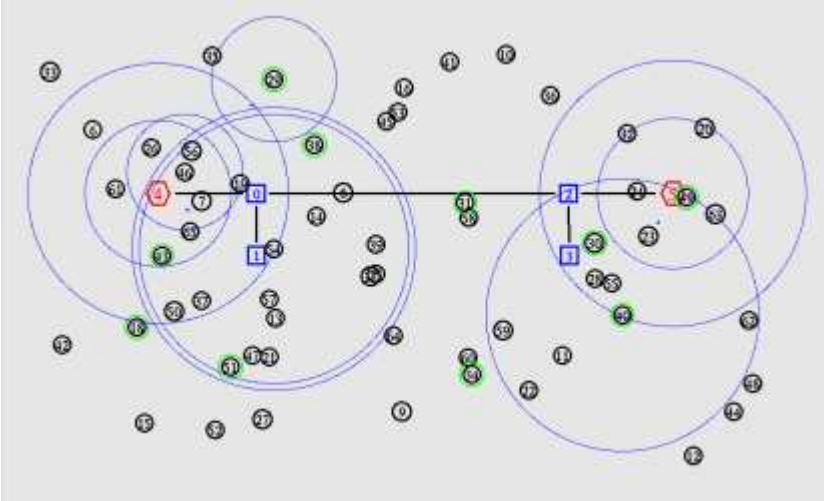


Figure 1: Screenshot of the simulation scenario.

Each mobile node begins the simulation by selecting a random destination in the defined area and moves to that destination at a random speed. The random speed is distributed uniformly in the interval [1,19] m/s. Upon reaching the destination, the mobile node pauses for 10 seconds, selects another destination, and proceeds as described. This movement pattern is repeated for the duration of the simulation.

The gateways broadcast GWADVs every ADVERTISEMENT_INTERVAL (equal to five seconds) when the proactive or hybrid discovery method is used (see Sect. 3.1 and 3.3). ADVERTISEMENT_ZONE, which is set to three, is used for the hybrid gateway discovery method and defines the range within which proactive gateway discovery is used. Outside this range the reactive gateway discovery is used.

4.3 Performance Metrics

In comparing the gateway discovery approaches, the evaluation has been done according to the following three metrics:

- The packet delivery ratio is defined as the number of received data packets divided by the number of generated data packets.
- The end-to-end delay is defined as the time a data packet is received by the destination minus the time the data packet is generated by the source.

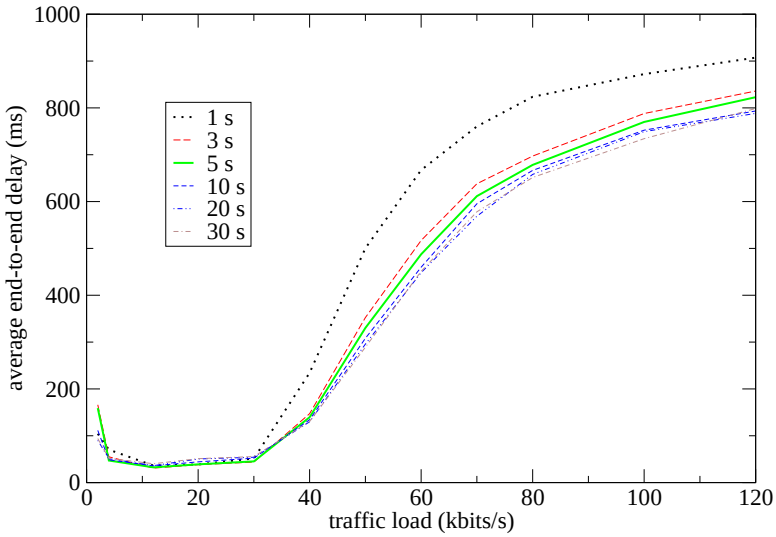


Figure 2: The impact of advertisement interval.

- The overhead is defined as the amount of AODV messages in bytes divided by the sum of the AODV messages plus the data packets in bytes.

Each data point is an average value of ten runs with different randomly generated movement patterns.

4.4 Simulation Results

In all figures discussed in this section it should be noted that the term "traffic load" denotes only the data traffic that each source generates, which is ten times less than the total data traffic in the whole network. To that come also control packets sent by the data link and network layers.

Figure 2 shows the impact of the advertisement interval on the average end-to-end delay when the traffic load changes for the proactive gateway discovery method. It can be observed that the curve representing the advertisement interval of one second differs greatly from other curves representing higher advertisement intervals. The reason is that a very short interval leads to a lot of advertisements and thus a lot of overhead, which in turn means many collisions, retransmissions and route discoveries that increase the end-to-end delay.

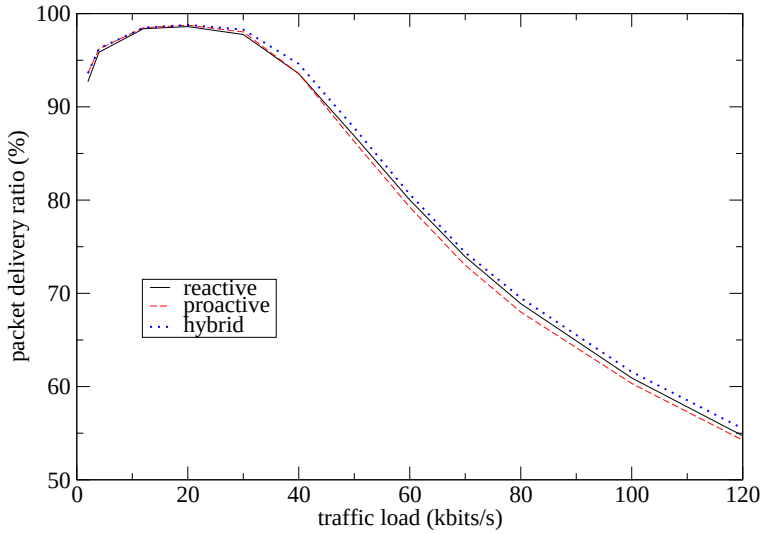


Figure 3: Packet delivery ratio vs. traffic load.

Figures 3, 4 and 5 show the packet delivery ratio, the average end-to-end delay and the AODV overhead respectively for the three gateway discovery methods when the traffic load changes.

Packet losses occur frequently due to many reasons, e.g. when a source sends packets along a path that recently has broken but the source has not been informed about that yet; or when a source has no other nodes within its transmission range (i.e. the node is isolated) for some time and its outgoing buffer is full. Since we have omitted the TCP protocol and its retransmission function from our model high packet losses may occur.

As Fig. 3 shows, the packet delivery ratio is high when the traffic load is light but decreases when the traffic increases. This result is expected but it can also be seen that increasing the traffic affects all three approaches pretty much the same way. One can also see that the delivery ratio is somewhat lower for very light loads (5 kbits/s/source) compared to light loads (20 kbits/s/source). The reason for this is that once a connection has been established, it is not fully used when the traffic is very light. Therefore, only a few number of packets are sent before the connection breaks and a new route must be discovered.

Figure 4 shows that the average end-to-end delay increases as expected when the load increases, since increased load means more collisions, retrans-

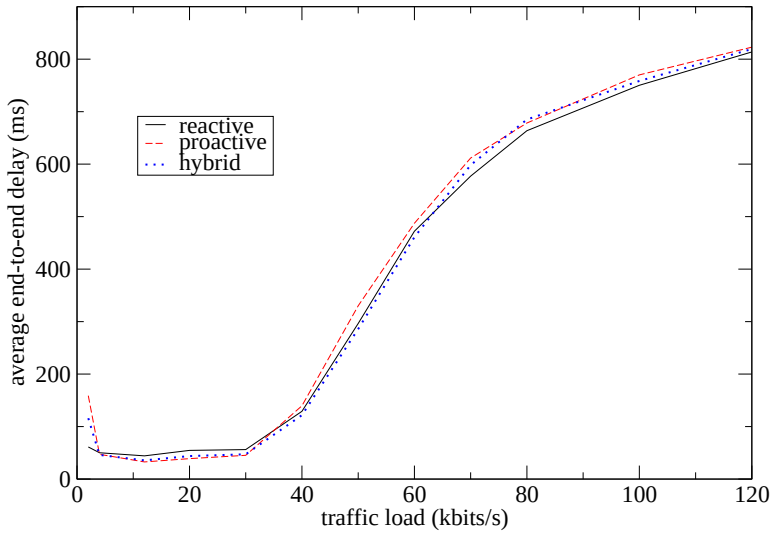


Figure 4: Average end-to-end delay vs. traffic load.

missions and route discoveries. We can also see that the difference between the different strategies is negligible.

One might have expected that the delivery ratio and the average end-to-end delay would have been different for the reactive method compared to e.g. the proactive. From one point of view, the reactive method should perform better since it generates less overhead, which should cause less number of collisions. On the other hand, the reactive method should perform worse because it does not send periodic advertisements, which would give shorter routes (in terms of number of hops) in the long term. Since a number of other aspects need to be taken into account, it is our belief that the given scenario and the assumptions made for the simulation have a significant impact on the results.

There are some problems with the ARP[4] implementation in ns-2, which is based on the BSD[5] implementation of ARP [11], that have negative impact on our results. Each node has an ARP queue that can hold only one packet for each neighbour while requesting the MAC address of the next hop. If other packets arrive to the queue before the MAC address is resolved, all but the last one will be dropped [12]. This can lead to loss of important messages

[4]Address Resolution Protocol
[5]Berkeley Software Distribution

from upper layers, such as the RREP or the RREP_I messages from AODV. Consequently, if the source does not receive any RREP or RREP_I before its timer expires, it has to reattempt its gateway discovery process where the reply could be lost again. Remember that this important message can be dropped by ARP on each hop between the gateway and the source where an address resolution process is started. In the worst case, the source will give up after some attempts and the session is aborted. Increasing the buffer size of ARP can prevent situations like this to occur.

There is another problem, where ARP is involved, which cannot be solved by increasing the buffer size. Since there is no timer involved in the address resolution process, a retransmission will not occur until it is triggered by a new incoming packet. This can have a significant impact on the end-to-end delay. Suppose that a data packet is sent to ARP from the routing protocol. Because of some reason (e.g. collision) the address resolution fails. Before a new data packet is sent to ARP to trigger an ARP request retransmission, the routing protocol changes its route towards the destination (with a new next hop) and, hence, no MAC address resolution is needed for the old next hop anymore. So far there is no problem except that the old data packet remains in the ARP queue. If the node much later needs to resolve the MAC address of the old next hop and the ARP resolution succeeds, the data packet waiting in the queue will be sent to the next hop resulting in a very long end-to-end delay. Increasing the buffer size will in fact only make the problem even worse since then there are more than a single data packet that will be delivered to the next hop with a very long end-to-end delay.

Furthermore, the lack of retransmissions means that one single loss of an ARP request or an ARP reply means that the data (e.g. RREP_I) cannot be sent to the source, which will be forced to reattempt its gateway discovery process.

The first problem caused by ARP has been investigated in [13], which shows that increasing the ARP buffer size makes the situation much better (although another solution is preferred). The second problem is discussed in [14], which suggests a cross-layer feedback mechanism from MAC to ARP.

Another thing that affects the simulation results in a negative way is when sources become isolated from the MANET such that they cannot reach any gateway. Isolated sources result in decreased packet delivery ratio and

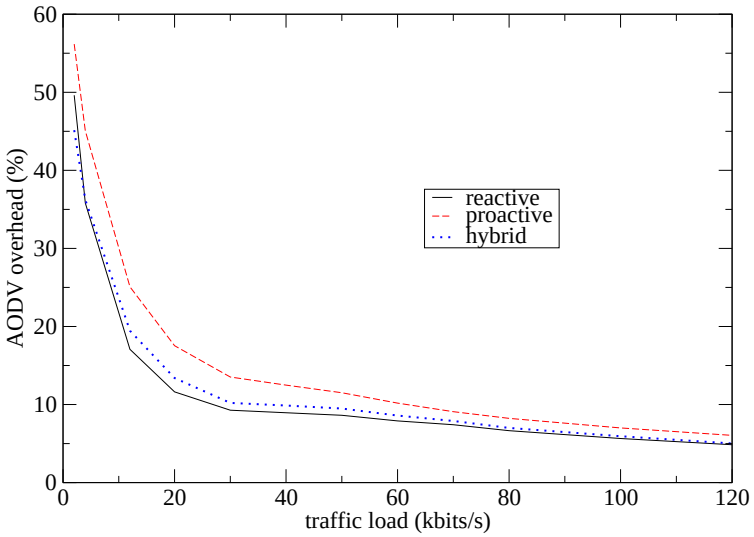


Figure 5: AODV overhead vs. traffic load.

increased end-to-end delay.

In Fig. 5 the AODV overhead is dominated by the periodically broadcasted GWADV messages. As the figure shows, the AODV overhead is significantly larger for the proactive approach than for the reactive approach, especially for light traffic loads. This is an expected result since the proactive approach periodically broadcasts gateway information no matter if the mobile nodes need them or not, while the reactive approach broadcasts gateway information only when a mobile node sends a request for it. Moreover, the figure shows that the overhead of the hybrid approach, which is a mixture of both the proactive and the reactive approach, is between the two other methods.

5 Conclusion

We have presented a solution for Internet access for mobile nodes in a MANET. The MANET routing protocol AODV has been extended to route packets, between a wireless MANET and the wired Internet. To achieve this, some nodes must be able to communicate with the MANET and with the fixed Internet. As all communication between the wireless and the wired network must pass through these nodes, they are referred to as gateways.

In this paper, three methods for detection of these gateways have been presented, implemented and compared. The three methods for gateway detection are referred to as reactive, proactive and hybrid gateway discovery. When it comes to end-to-end delay and packet delivery ratio, the three methods show surprisingly similar behaviour. The fact that the proactive method shows much higher overhead in terms of control packets than the other methods is more obvious.

In order to fully understand the reasons behind the large delays and the rather low packet delivery ratio that were found, the authors plan to do a more detailed study. This would provide a better understanding of which parts of the end-to-end path that contribute most to the discovered delays and packet losses.

References

[1] C. Perkins, E. M. Belding-Royer and S. Das. *"Ad hoc On-Demand Distance Vector (AODV) Routing"*. Experimental RFC 3561.

[2] D. B. Johnson, D. A. Maltz, Y. Hu and J. G. Jetcheva. *"The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)"*. IETF Internet Draft, April 2003. Work in progress.

[3] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum and L. Viennot. *"Optimized Link State Routing Protocol"*. Experimental RFC 3626.

[4] R. Ogier, M. Lewis and F. Templin. *"Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)"*. Experimental RFC 3684.

[5] R. Wakikawa, J. Malinen, C. Perkins, A. Nilsson and A. J. Tuominen. *"Global Connectivity for IPv6 Mobile Ad Hoc Networks"*, IETF Internet Draft, February 2003. Work in progress.

[6] S. McCanne and S. Floyd. *"The Network Simulator - ns-2"*. K. Fall, K. Varadhan. *"The ns Manual"*. `www.isi.edu/nsnam/ns/`.

[7] *"The Network Simulator: Contributed Code"*. `www.isi.edu/nsnam/ns/ns-contributed.html`.

[8] M. Bernard. *"Gateway Detection and Selection for Wireless Multihop Internet Access"*. Master's thesis. Olching, Germany, May 2002.

[9] A. Hamidian. *"A Study of Internet Connectivity for Mobile Ad Hoc Networks in NS 2"*. Master's thesis. Department of Communication Systems, Lund Institute of Technology, Lund University. January 2003.

[10] J. Jungkeun, M. Liu and B. Noble. *"Random Waypoint Considered Harmful"*. IEEE INFOCOM 2003, San Francisco, April 2003.

[11] W. R. Stevens. *"TCP/IP Illustrated, Volume 1"*. Addison Wesely, 1994.

[12] J. Broch, D. Maltz, D. B. Johnson, Y. Hu and J. Jetcheva. *"A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols"*. In proceedings of the 4th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98), pages 85-97, October 1998.

[13] C. Carter, S. Yi and R. Kravets. *"ARP Considered Harmful: Manycast Transactions in Ad Hoc Networks"*. Proceedings of the IEEE Wireless Communications and Networking Conference, 2003.

[14] S. Perur, L. Wadia and S. Iyer. *"Improving the Performance of MANET Routing Protocols using Cross-Layer Feedback"*. `www.it.iitb.ac.in/~srinath/pubs/cit03.pdf`.

Micro Mobility and Internet Access Performance for TCP Connections in Ad Hoc Networks

Anders Nilsson, Ali Hamidian, Ulf Körner

Abstract

In ad hoc mobile networks nodes typically communicate over wireless channels and are capable of movement. These are networks that support multihop communication and can be formed on a temporary basis. This paper evaluates a solution that allows mobile nodes to access the wired Internet and roam from base station to base station. The solution is based on the extension of Mobile IP capabilities to the ad hoc network while a micro-mobility protocol is adapted to support local migration. We evaluate the performance of this solution with regard to reliable transport layer connections. It is shown that a high throughput is possible to achieve for high mobility speeds. It is also observed that, as the number of hops between a mobile node and the base station increases, the throughput is decreased because of the characteristics of the wireless environment and the medium access layer protocol.

1 Introduction

Many portable computing devices such as laptops and PDAs now include wireless connectivity as a standard feature. More people are also carrying computers when they travel, and want access to the Internet anytime and anywhere.

The Internet Protocol (IP) as defined by The Internet Engineering Task Force (IETF) has become the most widely accepted standard for internetwork communication. Today, broadband wireless access networks based on IEEE 802.11 [1] are rapidly being deployed. In addition, other 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. Along with the mobility management issues, the new protocols that are currently under development, must also be radio independent. One of the most widely known mobility solutions for IP networks is the IP Mobility Support [2], commonly referred to as Mobile IP. Mobile IP do have some drawbacks and the concept of IP mobility has now 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.

Another emerging wireless architecture, namely *mobile ad hoc networks* (MANETs) [3], are networks that can be flexibly deployed in most environments without the need for infrastructure, such as base stations. A MANET is a network consisting of a set of mobile nodes, which may communicate with one another and roam around at will. The routing path may consist of a sequence of wireless links without the need to pass base stations (i.e., in a multi-hop manner). This requires each mobile node to serve as both a host and a router. In most cases today, MANETs use IEEE 802.11 network interface cards. MANET applications and scenarios include situations in which a network infrastructure is not available but immediate deployment of a network is required, such as outdoor assembly or emergency rescue.

Integrating and combining these wireless and mobile architectures will

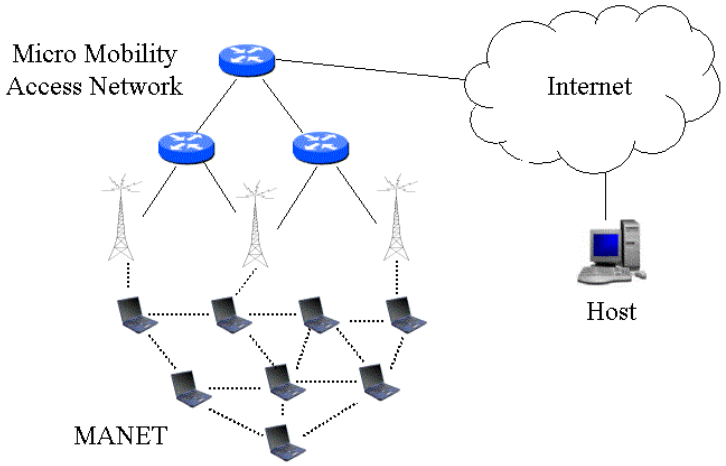


Figure 1: The simulated scenario. A mobile multihop ad hoc network is connected to an access network that supports Mobile IP and micro mobility. Nodes in the wireless network are communicating with correspondent hosts on the Internet.

facilitate the current trend of moving towards an all-IP wireless environment. This paper evaluates a solution that extends the typical wireless access points to multiple MANETs, each as a subnet of the Internet, to create an integrated environment that supports both macro and micro IP mobility, see Figure 1. 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.

The rest of this paper is organized as follows: Section 2 briefly describes the involved protocols. Section 3 presents the evaluated solution and Section 4 presents performance results obtained through simulation. Section 5 discusses related work and Section 6 concludes the paper.

2 Protocol Descriptions

2.1 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 forwards packets through tunneling to the foreign network. Every time the mobile changes its IP Point Of Attachment (IPPOA), these two steps must be accomplished in order to allow the mobile node to receive packets. However, it is the mobile node that initiates the process by sending a registration request once it has detected that it has moved from one network to another and has obtained a new care-of address. This introduces two causes of latency:

- 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 that 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.

2.2 Micro Mobility and HAWAII

To minimize the movement latencies discussed above, the concept of micro-mobility protocols have been introduced. A micro-mobility protocol operates as follows. The mobile node obtains a local care-of address when it first connects to a domain. This care-of address remains valid while it stays in the same domain and the mobile 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 managed by a micro-mobility protocol. This mobility is transparent to the home agent and the rest of the Internet. Latency and control traffic across the whole network are thus extremely reduced.

HAWAII [4], 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 where it belongs. While moving in its home domain, the mobile node 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 reverts 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. The base station is sometimes also called the access router, as it also has routing capabilities in addition to providing fixed network access. The base station 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, send periodic aggregate hop-by-hop refresh messages towards the domain root router.

2.3 AODV

The Ad hoc On-Demand Distance Vector (AODV) routing protocol [5] 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 connectivity 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 forward 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 that 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 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.

3 Mobile Ad hoc Internet Access Solution

In this solution, base stations that are also acting as Home and Foreign agents advertise their services by periodically sending out *Agent Advertisement* messages. These messages are broadcasted to the wireless ad hoc network, and its dissemination is limited by specifying the TTL to an appropriate value that depends on the size of the network. When a currently unregistered mobile node receives an advertisement, the mobile node uni-

casts a *Registration Request* to the sending base station. The base station will reply to this message by sending a *Registration Reply* back to the mobile node. If this is the first registration sent by the mobile node inside this domain, HAWAII, the micro-mobility protocol, will send path setup power-up messages in order to establish a routing path within the domain hierarchy towards the mobile node. The mobile node now also attains its care-of address, which the base station registers with the home agent. 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 towards 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, owning the network id. 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.

3.1 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, will be generated. This proxy route reply will also establish a route path between the requesting node and the requested destination. Note, however, that the ad hoc network may span several base stations, and therefore include nodes registered with other base stations. In order to let a direct route prevail, the proxy route reply will indicate a high hop count.

3.2 Handover

When a mobile node receives an agent advertisement from a base station that is closer than the one it is currently registered with, or if the old registration timed out, it initiates a handover. This is done sending a registration request to the new base station, that also includes information about the previous base station. When the new base station receives this message it replies by sending a registration reply as normal. The HAWAII part of the new base station also sends path update messages to the local micro-mobility domain and a handover notification is sent to the old base station. The old base station thus removes the mobile node from its registration list and updates its routing table accordingly.

The mobile node will now be reachable from the Internet through the registration in the new base station. The route within the ad hoc network, will however, point towards the old base station. A possible solution to this problem could be to let AODV send a route error message that deletes the route to the old base station, and then send a new route request as described above. The route error is needed because an intermediate node on the old route might otherwise reply to the request. This could, however, disrupt an active transport layer connection, something that we would like to avoid. In this solution, each mobile node instead has a list of destinations located on the Internet that it is currently communicating with, i.e., destinations learned of through the reception of route replies with an 'I'-flag. The mobile node parses this list of destinations, and sends a route request with the 'D'-flag specified, for each of these destinations. 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 without any intermediate nodes replying to the request.

4 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 has been evaluated in the popular network simulator, ns-2 [6]. 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 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 Mbps.

The simulations conducted aim to analyze the performance of TCP flows during an Internet connection and during handover. The current Internet host implementations contain a variety of TCP flavors. In order to investigate the differences between these, various TCP versions have been selected and analyzed. These are TAHOE, RENO, and VEGAS. 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.

The simulated scenario consists mainly of two parts, an access network consisting of base stations connected by wired links, and a wireless ad hoc network, see Figure 1. The access network is also the micro-mobility domain and consists of four base stations connected in a three-level network hierarchy of in total six 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 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, it performs handovers. The hop distance between the mobile node and its associated base station varies between two and three, depending on the

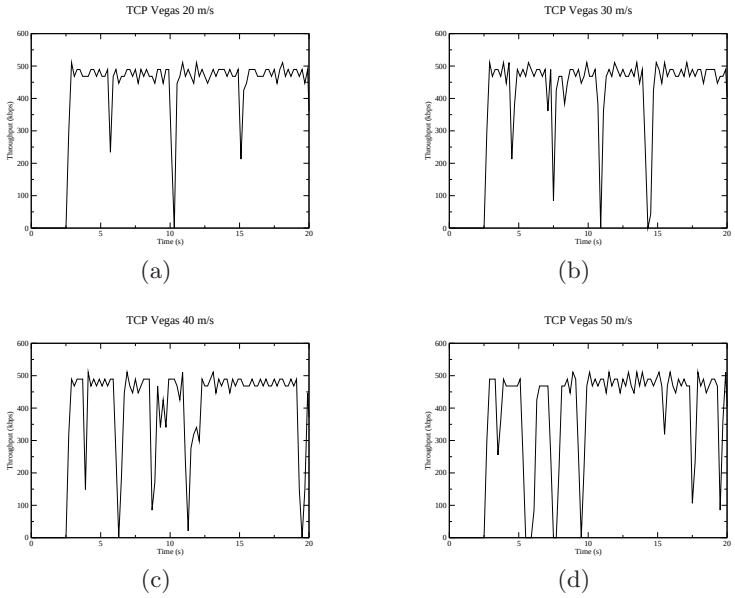


Figure 2: TCP throughput (kbps) for different mobility speeds (a) 20 m/s, (b) 30 m/s, (c) 40 m/s, and (d) 50 m/s.

connectivity of the network. The hop distance is never 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 one second.

In Figure 2a 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 decide when it is time to perform a handover. When a node learns of a new and closer base station, it switches to it. The mobile node also switches to a new base station 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 base station 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

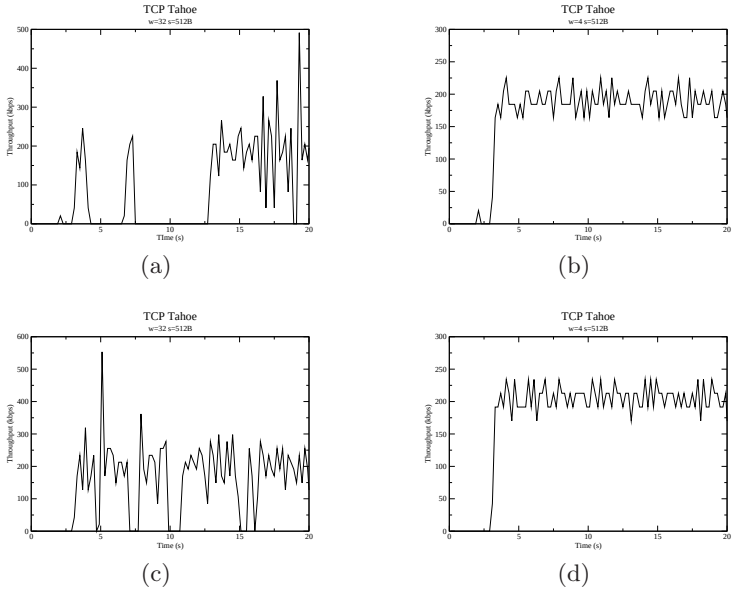
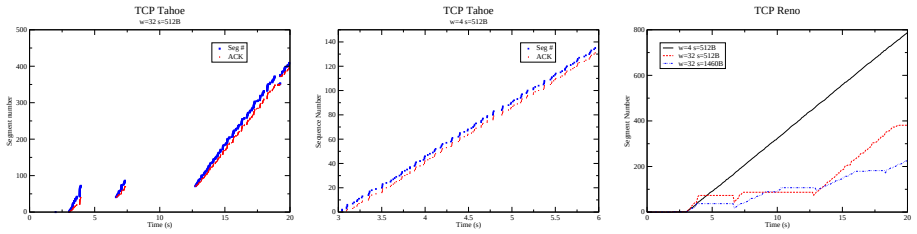


Figure 3: TCP throughput (kbps) for a static upload and download scenario (a) download, (b) download, (c) upload, and (d) upload.

to a lower level. Figure 2b, 2c and 2d show 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 become wider and deeper. At 50 m/s it takes about one second for the connection to regain its throughput, but only for a short time before a new handover takes place. It should be noted that 50 m/s is a very high speed; it corresponds to 180 km/h.

The fact that the wireless environment itself is unreliable has a big impact on the performance. This can be observed in Figure 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 five wireless hops. As can be seen from the figures, the throughput is fairly poor, and is significantly lower than the ones seen in Figure 2, even though those figures refer to a mobile scenario. One of the reasons for this is the exposed node problem [7], which 802.11 does not 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. The result here is that the throughput is lowered for each additional hop. Another problem is the window behaviour of TCP. The different



(a) (b) (c)

Figure 4: TCP segment number and ACKs for two window sizes (a) Tahoe window size w=32, (b) Tahoe window size w=4, and (c) Reno segment numbers.

figures in Figure 3 all show the behaviour of TCP Tahoe, a fairly aggressive flavor 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 transmits 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 3c.

We can also see a distinct difference in performance between the download and upload scenario in Figure 3a vs. 3c. 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 4a that show TCP segment numbers and ACKs. The throughput of this scenario is the one seen in Figure 3a. Because of the lower bandwidth at the wireless sender in the upload scenario, the same amount of packet drops does not take place, see the corresponding throughput in Figure 3c.

One 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 Figure 3b and 3d.

Table 1: Mean throughput for various TCP flavors during upload and download.

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

Yet another factor that impact the performance is the size of the packets. Figure 4c 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.

Table 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 m/s 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 occurs in this scenario, which is the reason why Reno performs worse than Tahoe.

Another issue with TCP flows in ad hoc networks is unfairness. This can observed in Figure 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 described before [8], and a few different solutions have been proposed. This problem needs to be solved before 802.11-based ad hoc networks can have any real success.

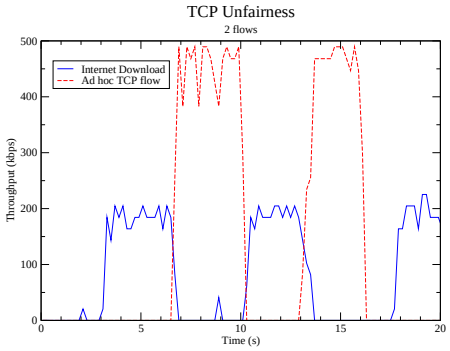


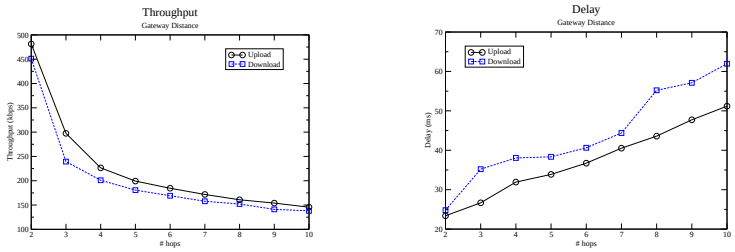
Figure 5: TCP (Vegas) unfairness during a static download scenario with a competing local flow inside the ad hoc network.

During the course of our investigation, we also observed that the throughput and delay clearly depend 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 6. We can see here that the throughput during upload is around 475 kbps when the distance is two hops, but only around 150 kbps for ten 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, as nodes are prevented from transmitting because the next hop node is transmitting. The upload throughput is also always higher than during download, as discussed above. The increase in delay seems to be almost linear with the number of hops, at least during upload. For two hops, the delay is around 25 ms, but for ten hops it has been doubled to around 50-60 ms. As each additional hop introduces additional processing time, this makes perfect sense.

5 Related Work

In [9], the authors present a solution that interconnects ad hoc networks with infrastructured networks. For micro mobility, the Cellular IPv6 [10] 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 [11], the authors integrate AODV with Mobile IP. Their solution utilizes IP tunneling for separating the ad hoc network from Mobile



(a) (b)

Figure 6: (a) TCP throughput and (b) delay for different gateway distances, for the static scenario.

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.

6 Conclusion

We have in this paper presented a solution for Internet access and micro mobility for ad hoc networks. The solution relies on the AODV routing protocol 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. The problem with unfairness needs to be solved before multiple TCP flows can be supported.

References

[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] S. Corson and J. Macker. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. IETF RFC 2501, January 1999.

[4] 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.

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

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

[7] 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.

[8] 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.

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

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

[11] 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.

An Enhancement to the IEEE 802.11e EDCA Providing QoS Guarantees

Ali Hamidian and Ulf Körner

Abstract

One of the challenges that must be overcome to realize the practical benefits of ad hoc networks is quality of service (QoS). However, the IEEE 802.11 standard, which undeniably is the most widespread wireless technology of choice for WLANs and ad hoc networks, does not address this issue. In order to support applications with QoS requirements, the upcoming IEEE 802.11e standard enhances the original IEEE 802.11 MAC protocol by introducing a new coordination function which has both contention-based and contention-free medium access methods. In this paper, we consider the contention-based medium access method, the EDCA, and propose an extension to it such that it can be used to provide QoS guarantees in WLANs operating in ad hoc mode. Our solution is fully distributed, uses admission control to regulate the usage of resources and gives stations with high-priority traffic streams an opportunity to reserve time for collision-free access to the medium.

1 Introduction

Today, *wireless local area networks* (WLANs) are deployed in many universities, homes, cafs, train stations, airports and even in airplanes. The key for these networks to become even more useful and popular is, among other things, to support applications with *quality of service* (QoS) requirements, such as video, audio, voice over IP and other multimedia applications. However, the original 802.11 standard [1] has not addressed the QoS issues sufficiently. The later 802.11a/b/g standards offer only higher maximum data rates by enhancing the *physical layer* (PHY) of the original standard - they all use the same medium access method that does not support QoS. Therefore, the 802.11 working group formed task group E (802.11e) to address the QoS issues in the *medium access control* (MAC) layer.

The upcoming 802.11e standard [2] defines a new coordination function, called *hybrid coordination function* (HCF), that includes two medium access methods: *enhanced distributed channel access* (EDCA) and *HCF controlled channel access* (HCCA). The EDCA is distributed and contention-based, making it suitable for ad hoc networks while the HCCA is centralized[6] and contention-free - thus, it cannot be used in infrastructure-less networks. One drawback with the EDCA compared to the HCCA is that a station cannot reserve the medium and access it without needing to contend for it; instead the station must contend for access to the medium by possibly starting a backoff timer of random length. Moreover, there is no distributed admission control algorithm. Therefore, it is not possible to provide QoS guarantees using the EDCA [3].

There has been a lot of research on QoS for WLANs and ad hoc networks. Among these studies, many have focused on a solution at the MAC sublayer. This is explained by the fact that QoS provisioning is not possible unless supported by the MAC protocol. In other words, a QoS-aware MAC protocol is necessary but perhaps not sufficient (at least for multi-hop ad hoc networks where e.g. a QoS-aware routing protocol is required to find a route, satisfying the QoS requirements, between a source and a destination). Most of these works have been made for infrastructure-based networks. Among the studies focusing on the MAC sublayer and infrastructure-less networks,

[6]The HCCA manages access to the medium using a QoS access point (QAP).

a few propose a solution that is compatible with 802.11. It is our belief that, since the 802.11 standard is so widely spread, any realistic proposal must be compatible with this technology.

The *fuzzy CW allocation control* (FCWAC) scheme is proposed in [4]. It aims to solve the problems occurring due to a) the doubling of the *contention window* (CW) after each unsuccessful transmission and b) the resetting of the CW to *CWmin* after each successful transmission. The first event results in very large delays after just a few unsuccessful retransmissions while the second event may result in a so called *bursty collision* since the current network conditions are not taken into consideration. Instead, after each collision or successful data transmission, FCWAC dynamically adjusts the CW based on the current queue length, loss probability and packet waiting time. Although FCWAC tries to improve the EDCA, it is still based on a random waiting time before accessing the medium and it does not have any distributed admission control algorithm. In other words, it is not possible to guarantee QoS because the scheme is based on service differentiation just as the EDCA.

In [5], the *Simple Scheduler*, the *Grilo Scheduler* [6] and an extension to the the Grilo Scheduler are presented. The Simple Scheduler is an example of a scheduler mentioned in [2] where each station can schedule medium time of fixed length at constant intervals. The Grilo Scheduler has extended this functionality by allowing the stations to schedule medium time of variable length at different intervals for each station. One drawback in the Simple Scheduler that has not been considered by the Grilo Scheduler is that the duration of the medium time to be reserved is calculated based on average traffic rates or estimations. In [5] the calculation is instead based on actual requirements; this is achieved by using two fields with information about the queue size and the requested duration of the medium time to be reserved. The proposed enhancement can be used in our scheme instead of the Simple Scheduler given as an example in [2].

Distributed MAC schemes based on 802.11 and designed for providing QoS are studied in [7]. The schemes are classified into priority-based and fair-scheduling-based approaches. The priority-based schemes, like the EDCA, provide service differentiation by allowing faster access to the channel to traffic classes with higher priority. The authors do not consider these

schemes in their simulations since they are unfair: as the number of high-priority streams increase, they tend to grab the channel, preventing fair access for low-priority streams. Thus, the authors make a simple comparison between the three approaches using fair scheduling: *distributed weighted fair queuing* (DWFQ) [8, 9], *distributed fair scheduling* (DFS) [10] and their own proposal *distributed deficit round robin* (DDRR) [11], which is based on the concept of *deficit round robin* (DRR) (ref. 18 in [7]). In DDRR, each traffic class determines its allotted *service quantum rate* based on its throughput requirements and maintains a deficit counter of accumulated quanta. The deficit counter is decreased by the size of the transmitted frame and a traffic class can transmit only when the counter is positive. As the authors themselves note, the proposed mechanisms only provide service differentiation; none of them can guarantee QoS since they do not have any mechanism for admission control or resource allocation.

An Admission Control and Dynamic Bandwidth Management scheme is proposed in [12] to provide fairness and soft guarantees in single-hop ad hoc networks. The main piece of the scheme is a centralized (but wireless) *Bandwidth Manager* (BM) used to dynamically allot each stream a share of the channel, depending on the requirements of the stream relative to the requirements of other streams in the network. The BM is also in charge for the admission control management. The authors consider a single-hop ad hoc network as we do, but their proposal can only give soft QoS guarantees and, in addition, it relies on a central station (the BM) and operates at the application layer.

In this paper we consider single-hop ad hoc networks, i.e. WLANs operating in ad hoc mode. One advantage of such networks, instead of the ones operating in the traditional infrastructure mode, is that all frames do not need to pass through an *access point* (AP) waisting bandwidth and making the communication inefficient. If an AP is used as an intermediary, direct one-hop transmissions needlessly become two-hop transmissions [12]. Moreover, the AP is a single point of failure and can thus cause the whole network to fail. Instead of relaying peer-to-peer transmissions between stations within the wireless network, the AP should be used only as a gateway toward the wired Internet. Hence, we propose an extension to 802.11e, which provides QoS guarantees in single-hop ad hoc networks by making

use of the advantages of the HCCA and integrate them into the EDCA. Our solution is fully distributed and gives the stations with high-priority traffic an opportunity to reserve medium time.

It is worth mentioning that, when talking about QoS guarantees, we must keep in mind that since a wireless medium is much more unpredictable and error-prone than a wired medium, QoS cannot be guaranteed as in a wired system, especially in unlicensed spectra. However, it is possible to provide techniques that increase the probability that certain traffic classes get adequate QoS and that can provide QoS guarantees in controlled environments [2].

The remainder of this paper is organized as follows: Section 2 gives an overview of the original 802.11 and its QoS limitations. Moreover, it describes the 802.11e draft with focus on the EDCA. Our proposed solution is presented in Section 3. The simulation results are presented and discussed in Section 4. Finally, Section 5 concludes this paper and gives some directions for future work.

2 IEEE 802.11 and IEEE 802.11e

Since the 802.11 standard did not address the QoS issues sufficiently, the 802.11 working group formed task group E. However, although 802.11e is an important enhancement of 802.11, its contention-based medium access method only provides service differentiation and hence, there is no way to provide QoS guarantees in networks independent of any centralized devices.

2.1 IEEE 802.11 MAC and its QoS Limitations

The 802.11 standard [1] has defined two medium access methods: the *distributed coordination function* (DCF) and the *point coordination function* (PCF). DCF provides a best effort data service and is mandatory while PCF is optional and provides a time-bounded service. For these access methods, four different parameters are used for controlling the waiting time before medium access.

Short interframe space (SIFS): The shortest waiting time, and thus the highest priority for medium access. The SIFS is used by short control messages, such as *clear to send* (CTS) frames, *acknowledgment* (ACK) frames,

or polling responses.

PCF interframe space (PIFS): A waiting time longer than SIFS but shorter than DIFS (and thus a medium priority). The PIFS is used only by stations operating under PCF, e.g. by the AP polling other stations.

DCF interframe space (DIFS): A waiting time longer than both SIFS and PIFS and thus the lowest priority for medium access. The DIFS is used only by stations operating under DCF transmitting data frames or management frames.

Extended interframe space (EIFS): The longest waiting time used by stations operating under DCF, but only when a transmission failure occurs. A station that receives an incorrect frame must wait for EIFS before starting its transmission in order to give other stations enough time to acknowledge the frame that the station received incorrectly.

The parameters *SIFS* and *SlotTime* are fixed per physical layer whereas DIFS and PIFS are derived from these two parameters. Table 1 shows these and some other MAC parameters (explained below) specified for 802.11b PHY [13]; *direct sequence spread spectrum* (DSSS).

2.1.1 The Coordination Functions of 802.11 - DCF and PCF

DCF is based on *carrier sense multiple access with collision avoidance* (CSMA/CA) which works as follows. If the medium is determined to be idle for at least the duration of DIFS, a station can start transmitting. If the medium is determined to be busy, a station defers its transmission until the end of the ongoing transmission. After deferral, the station selects a random backoff time as follows:

$$\text{backoff time} = \text{random}() \times \text{SlotTime},$$

where random() is a uniformly distributed integer in the interval [0,CW] and CW is an integer between *CWmin* and *CWmax*, i.e., CWmin $\leq$ CW $\leq$ CWmax (see Table 1).

A station performing the backoff procedure uses the carrier-sense mechanism to determine whether the medium is busy each time slot. As long as the medium is sensed to be idle for the duration of a time slot, the backoff procedure decrements its backoff time by a slot time. Whenever the medium

Table 1: MAC parameters for 802.11b PHY (DSSS)

SIFS	PIFS	DIFS	SlotTime	CWmin	CWmax
10 μs	30 μs	50 μs	20 μs	31	1023

is determined to be busy, the backoff procedure is suspended; that is, the backoff timer does not decrement for that slot. The medium shall be sensed to be idle for the duration of DIFS before the backoff procedure is allowed to resume. Once the backoff timer expires the station begins transmitting.

If two or more stations start transmitting at the same time a collision will occur. In this case, the CW is doubled and a new backoff procedure is started. The CW starts with CWmin and doubles up to a maximum of CWmax. Once it reaches CWmax, the CW maintains that value until it is reset. The CW shall be reset to CWmin after each successful attempt to transmit a frame. This process will continue until the transmission is successful or discarded.

DCF cannot guarantee a maximum access delay or minimum transmission bandwidth. To provide time-bounded service such as voice, audio or video, PCF has been specified. PCF is dependent on DCF and can thus not be used alone. Moreover, it requires an access point that controls the medium access and polls the stations; therefore it is only usable on infrastructure network configurations, i.e., infrastructure-less networks cannot use this function.

This access method uses a *point coordinator* (PC), which operates at the access point, to determine which station has the right to transmit. PCF is actually a polling medium access method with the PC performing the role of the polling master. The PC maintains a polling list of registered stations and polls each station one by one according to the list. No station is allowed to transmit unless it is polled, and stations receive data from the access point only when they are polled.

2.1.2 QoS Limitations of DCF and PCF

Some applications, such as data, audio and video, have different requirements in data rate, delay and jitter. However, in DCF all stations and data flows have the same priority to access the medium, i.e. in a first come first serve, best effort manner. Thus, there is no way to guarantee QoS, that is, there is no differentiation mechanism to guarantee data rate, delay or jitter for applications which are sensitive to these parameters.

Although PCF was specified to provide a time-bounded service, this access method has a few problems which leads to poor QoS performance: a) the lack of possibility for stations to communicate QoS requirements to the access point makes it hard to optimize the polling algorithm performance in the PC; b) the unpredictable beacon delays result in shortened *contention-free period* (CFP) and c) the transmission time of the polled stations is unknown, which makes it hard for the PC to predict and control the polling schedule for the remainder of the CFP. Therefore, PCF does not fulfill its task despite the fact that it uses an access point controlling access to the medium.

2.2 IEEE 802.11e

In order to solve the above-mentioned problems with PCF and DCF, the upcoming standard 802.11e [2] defines a new coordination function: *hybrid coordination function* (HCF). HCF has both contention-based and contention-free (controlled) medium access methods in a single medium access protocol, which explains why it is called *hybrid* coordination function.

The contention-based medium access method is called *enhanced distributed channel access* (EDCA) and provides prioritized QoS support by delivering traffic based on differentiating user priorities.

The controlled medium access method is called *HCF controlled channel access* (HCCA) and provides support for parameterized QoS by allowing for the reservation of transmission time. The HCCA manages access to the medium using a hybrid coordinator operating at a *QoS access point* (QAP).

In HCF, the concept of *transmission opportunity* (TXOP) is introduced. A TXOP is a bounded time interval, defined by a starting time and a maximum duration, that specifies when a station has the right to initiate trans-

Table 2: Default EDCA parameter set for 802.11b PHY (DSSS)

AC	CWmin	CWmax	AIFSN	TXOP Limit
AC_BK	31	1023	7	0
AC_BE	31	1023	3	0
AC_VI	15	31	2	6.016 ms
AC_VO	7	15	2	3.264 ms

missions to the wireless medium. During this time interval, a station can transmit multiple frames if the duration of the transmissions does not extend beyond the maximum duration. If a frame is too large to be transmitted in a single TXOP, it should be fragmented into smaller frames.

2.2.1 Enhanced Distributed Channel Access (EDCA)

In the EDCA every station has four transmission queues, or *access categories* (ACs), where each behaves like a virtual station. The four ACs are AC_BK (for background traffic), AC_BE (for best effort traffic), AC_VI (for video traffic) and AC_VO (for voice traffic). Thus, as opposed to DCF where all traffic shared a common queue, in the EDCA each traffic type is queued in the appropriate AC. By varying the following parameters for a specific AC, a differentiated medium access is realized:

- the length of the contention window to be used for the backoff
- the amount of time a station has to defer before backoff or transmission
- the duration a station may transmit after medium is accessed

Table 2 shows how this medium access differentiation is achieved by assigning certain parameters (explained below) different values.

The parameters *CWmin* and *CWmax* are the minimum and maximum value of the contention window. The contention window is used to calculate the number of time slots to backoff before accessing the medium. By assigning low values to CWmin and CWmax, we can give the AC a higher priority.

The *arbitration interframe space number* (AIFSN) is the number of time slots after a SIFS duration a station has to defer before either invoking a backoff or starting a transmission. AIFSN affects the *arbitration interframe space* (AIFS), which specifies the duration (in time instead of number of time slots) a station must defer before backoff or transmission. Thus, by assigning a low value to AIFSN, we give the AC a high priority. AIFS can be derived from the relation

$$AIFS[AC] = SIFS + AIFSN[AC] \times \text{SlotTime}$$

The parameter *TXOP limit* specifies the length (or maximum duration) of the TXOP. A TXOP limit higher than zero means that a station[7] can transmit multiple frames as long as the duration of the transmissions does not extend beyond the TXOP limit. A TXOP limit value of zero indicates that only one data or management frame (plus a possible RTS/CTS exchange) can be sent. Thus, by assigning a high value to the TXOP limit, we give the AC a high priority.

Each AC contends independently for TXOPs based on the parameters described above. Once the AC has sensed the medium idle for at least the duration of AIFS[AC], it starts its backoff timer. If two or more ACs within a single station get ready for transmitting at the same time slot, an internal collision occurs. The collision is resolved within the station such that the data frames from the higher-priority AC receive the TXOP and the data frames from the lower-priority colliding AC(s) behave as if there were an external collision on the wireless medium.

3 Proposed Approach

Although the distributed EDCA is an important enhancement of DCF, it is not enough to provide QoS guarantees due to its non-deterministic nature where stations use a backoff timer of random length to contend for access to the medium. Moreover, the EDCA does not have any distributed admission control, but the administration of the admission control is done at the QAP.

[7] In fact it is better to say an AC because during a TXOP, a station is not permitted to send frames from other ACs than the one that won the TXOP, even though there is time left in the TXOP.

On the other hand, the centralized HCCA, where e.g. a QAP controls the medium access and allows for TXOP reservations, cannot be used in networks independent of any centralized infrastructure.

Therefore, the purpose of our solution is to provide QoS guarantees in a infrastructure-less WLAN by transferring the best techniques from the HCCA and integrate them with the EDCA. In other words, our goal is to distribute the admission control and the scheduling algorithms and enhance every station in the distributed network with the QoS capabilities of a QAP.

3.1 Traffic Specification

The *traffic specification* (TSPEC) element contains information about the characteristics and QoS expectation of a traffic stream by specifying a set of parameters such as mean data rate, nominal frame size, service start time, maximum service interval, burst size, delay bound and medium time. The parameter *service start time* specifies the time when the service period starts, i.e. when the station expects to be ready to send frames and *maximum service interval* specifies the maximum time interval between the start of two consecutive service periods.

The information contained in the TSPEC helps other stations to schedule the TXOPs effectively. Most of the above-mentioned parameters are typically set according to the requirements from the application while other parameters are generated locally within the MAC.

The TSPEC element is sent within an *add traffic stream* (ADDTS) request frame, which is a management frame with subtype *action* [2].

3.2 Scheduling and Admission Control

Contention-based medium access is susceptible to significant performance degradation when overloaded. As the network becomes overloaded, the contention windows become large, leading to more time spent in backoff delays rather than sending data. This necessitates some admission-control mechanism to regulate the amount of traffic streams contending for access to the medium.

In 802.11e, the administration of the admission control is done at the hybrid coordinator, which is located in the QAP. The 802.11e draft gives

an example of a simple scheduler and an admission control unit but it is possible to modify these to improve the performance [5]. In our solution the scheduler and admission control algorithm are modified and moved from the central QAP to the stations. For scheduling TXOPs for an admitted traffic stream, the scheduler calculates two parameters:

scheduled service interval (SI): the interval between TXOPs, which is the same for all stations. To calculate the SI, the scheduler calculates the minimum m of all maximum service intervals for all admitted streams. Then SI equals a value lower than m and a submultiple of the beacon interval. SI must be recalculated when a new traffic stream is admitted that has a maximum service interval smaller than the current SI. To improve the performance of the scheduler, it can for example be modified to generate different SIs for different stations [6].

TXOP duration: to calculate the TXOP duration for an admitted stream, the scheduler uses the following parameters: mean data rate (ρ) and nominal frame size (L) from the TSPEC, the SI as calculated above, physical transmission rate (R), maximum allowable frame size, i.e. 2304 bytes (M) and overhead due to MAC and PHY headers (O). First, the scheduler calculates the number of data frames that arrived at the mean data rate during the SI:

$$N_i = \left\lceil \frac{SI \times \rho_i}{L_i} \right\rceil$$

Then the scheduler calculates the TXOP duration as the maximum of the time to transmit N_i frames at R_i plus overhead and the time to transmit one maximum size data frame at R_i plus overhead:

$$TXOP_i = \max\left(\frac{N_i \times L_i}{R_i} + O, \frac{M}{R_i} + O \right)$$

To improve the performance of the scheduler, it can for example be modified to consider retransmissions while allocating TXOP durations.

Once SI and TXOP duration are calculated, the admission control decision is easy. If there are k admitted streams, a new stream (k+1) can be

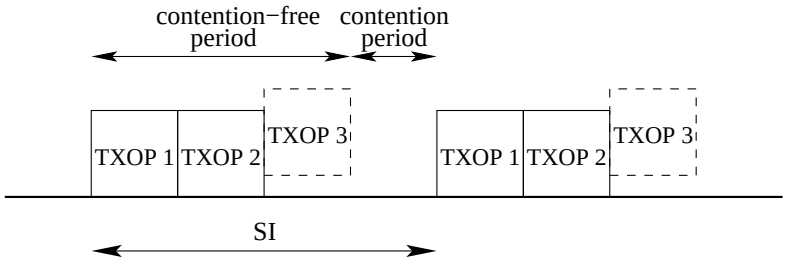


Figure 1: The scheduling of the reserved TXOPs

admitted if it satisfies the following inequality:

$$TXOP_{k+1} + \sum_{i=1}^{k} TXOP_i \leq SI - T_{CP},$$

where T_{CP} is the duration of the contention period. The last term ensures that some amount of time is saved for contending low-priority streams.

3.3 Resource Reservation

As long as there is no station that needs to reserve TXOPs for its high-priority traffic stream, our solution works like the EDCA. Once a station (sender) wishes to send traffic with strict QoS requirements, i.e. a high-priority traffic stream in either AC_VI or AC_VO, it requests admission for its traffic stream. The admission control request is not sent to any central station such as a QAP, but is handled internally within the sender. The sender either admits or rejects the requested traffic stream according to the admission control algorithm described in Section 3.2.

What happens if the traffic stream is rejected is described later, but if the traffic stream is accepted, the sender schedules its traffic by setting the SI and the service start time parameters. The SI is calculated as described previously and the service start time is set to the end time of the last TXOP in a service interval. Figure 1 shows an example where two stations have scheduled their TXOPs and a third station is about to schedule its TXOPs. If there are no TXOPs previously reserved, the service start time can instead be set to any appropriate value, which defines the start of the newly established service interval. Hence, during a service interval, the first part is used as a contention-free period by traffic streams that have reserved TXOPs and

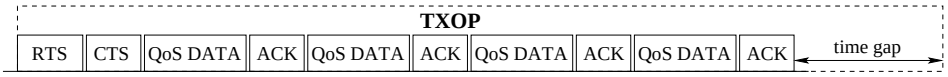


Figure 2: An example of a frame sequence during a TXOP

the second part is used as a contention period for low-priority streams.

Figure 2 shows an example of a frame exchange sequence in which four data frames are transmitted during a TXOP. It is worth mentioning that the sequence starts with an RTS/CTS exchange in order to prevent hidden stations from trying to access the medium while the sender transmits its frames. Thus, the hidden station problem is handled the same way (through RTS/CTS exchange) as in 802.11(e).

In addition, Figure 2 shows a time gap at the end of the TXOP. These time gaps between two TXOPs are caused by the scheduler's calculation of the duration of the TXOP, which is based on average values rather than exact values. The time gaps are used as an advantage by allowing the stations to transmit important control messages (such as ADDTS request, ADDTS response, messages from the routing protocol, etc.) or low-priority data frames instead of having to wait until the end of the last TXOP in the service interval.

To further increase the performance of the protocol and decrease the time it takes for the sender to reserve TXOPs, an AC (AC_MA) has been added that is used only by management frames, such as ADDTS request and ADDTS response, and routing messages. AC_MA has the same parameters as AC_VO except for TXOP limit, which is set to zero since a station does not need to send multiple management frames in a short time period. The reason behind the choice of AC_MA's parameters is to give the management frames a short waiting time before access to the medium.

Next, the sender broadcasts an ADDTS request containing a TSPEC element with information such as mean data rate, nominal frame size, service start time and SI. All stations that receive the ADDTS request store the information of the sender's service start time and SI and schedule the new traffic stream exactly as the sender. This ensures that no station starts a transmission that cannot finish before a reserved TXOP starts and thus collision-free access to the medium is guaranteed for the streams with reserved TXOPs. All neighbours have to unicast an ADDTS response back to

the sender. This is to make sure that the neighbours agree on the schedule and to keep the schedules synchronized.

Every time the sender receives an ADDTS response from a neighbour, it stores the address of the neighbour. After receiving a response from all neighbours, the sender waits until the service start time specified in the TSPEC element and initiates a transmission. If the time instant when all responses are received occurs later than the advertised service start time, the transmission is initiated at the next TXOP instead. During a TXOP, the sender can transmit multiple frames but it must stop sending when the remaining time of the TXOP is less than the transmission time of another data frame plus its corresponding ACK. Once the TXOP is finished, the station waits until the next TXOP, which occurs after an SI. A station that has reserved TXOPs for a traffic stream with strict QoS requirements, is not allowed to transmit frames belonging to that stream at time instants other than during the reserved TXOPs. In other words, the station can transmit frames only at

$$t = \text{service start time} + n \times SI, \quad n = 0, 1, 2...$$

Of course the station is allowed to transmit frames from other traffic streams, in other ACs, by contending for access to the medium. However, these streams and other low-priority streams from other stations must make sure to finish their transmission before a TXOP starts; otherwise the contending station must backoff and the frames are not allowed to be sent until after the reserved TXOP(s).

When a transmission failure occurs during a TXOP, the station does not start a backoff procedure. Instead, it retransmits the failed frame after SIFS if there is enough time left in the TXOP to complete the transmission.

When a traffic stream finishes and has no more frames to send, it broadcasts a *delete traffic stream* (DELTS) frame notifying other stations to delete the traffic stream and to reschedule the TXOPs of any remaining traffic stream.

If a traffic stream is rejected by the admission control algorithm, the sender can try to lower its QoS demands and retry. The demands should be lowered such that a lower TXOP duration is required. If this compromise is not enough, meaning that no TXOPs can be reserved, the sender has two

options left: a) the priority of the traffic stream is lowered such that the stream sends from another AC (with lower priority) that does not require admission control or b) the priority and thus the AC is kept, but the TXOP limit is set to zero. The second option means that, a traffic stream that cannot reserve TXOPs, does not necessarily have to move to another AC with lower priority and longer waiting time before medium access; instead the rejected traffic stream remains in the high-priority AC and contends for access to the medium using the parameters assigned to the high-priority AC, but once it gains access to the medium, it is not allowed to transmit more than one single data frame.

The advantages of this solution are that it is fully distributed, protects against network overload using an admission control algorithm and offers the possibility for stations with high-priority traffic to schedule their traffic in advance such that the QoS requirements of the traffic streams are satisfied. Moreover, since the solution is compatible with the widely used 802.11 standard and based on the upcoming 802.11e standard, it will be possible to integrate it into 802.11e without much difficulties. The proposed mechanism requires modifications only to the software of 802.11e, i.e. additional hardware is not needed.

4 Evaluation

In order to evaluate the performance of our solution, we have been working on a detailed implementation in the network simulator ns-2 [14]. Since the standard 802.11 implementation in ns-2 is rather simple, we used another more advanced 802.11 implementation [15] for ns-2, which also implements 802.11a/b/g and some features of 802.11e. This code was then modified and extended according to our solution described above. The implementation has been used to, by means of simulation, compare our solution against 802.11e concerning QoS guarantees in WLANs independent of centralized devices.

4.1 Simulation Scenario

The simulation scenario consists of a number of stations, all within transmission range of each other. The transmission range is 250 meters. The stations

use 802.11b DSSS in the physical layer and 802.11e EDCA or our modified version of 802.11e in the MAC sublayer. An error model is used causing 1% of the packets to be damaged. The high-priority streams are assumed to have a maximum delay bound of 10 ms so the parameter SI is 10 ms.

The high-priority streams are sent from AC_VO and use a constant bit rate traffic generator to generate UDP packets with a size of 210 bytes each third ms. The low-priority streams are sent from AC_BE and generate TCP segments according to an FTP application which always has data to transmit.

We have studied both the transient and the stationary behaviour of our solution and compared it toward 802.11e's medium access method EDCA. Regarding the transient behaviour, we study the impact of additional traffic streams on existing traffic by starting the applications at different time instants ten seconds apart. More specifically, we calculated the throughput and end-to-end delay (both at the transport layer) of all traffic streams when additional applications were started. In this scenario, there is one low-priority stream and four high-priority streams. Each traffic stream is sent from a unique source to a unique destination so the total number of stations is ten, i.e. five sources and five destinations. During these simulations, the size of the TCP segments is 210 bytes, i.e. equal to the size of the UDP packets.

Regarding the stationary behaviour, we study the impact of an increasing number of low-priority streams on one high-priority stream. More specifically, we calculated the average end-to-end delay[8], jitter[9] and squared coefficient of variance of the end-to-end delay ($C^2[d]$) for the high-priority stream when the number of low-priority streams was varied between zero and five. Each traffic stream is sent from a unique source to a unique destination so the number of stations is varied between two (one high-priority source-destination pair) and twelve (one high-priority and five low-priority source-destination pairs). The size of the TCP segments was increased to 1000 bytes during the stationary simulations. The reason for this is that we wanted to stress the compared MAC schemes regarding the QoS provi-

[8]The end-to-end delay is calculated as the time when a frame is received by the destination's application layer minus the time when the frame was generated at the application layer at the source.

[9]The jitter is calculated as the variance of the end-to-end delay.

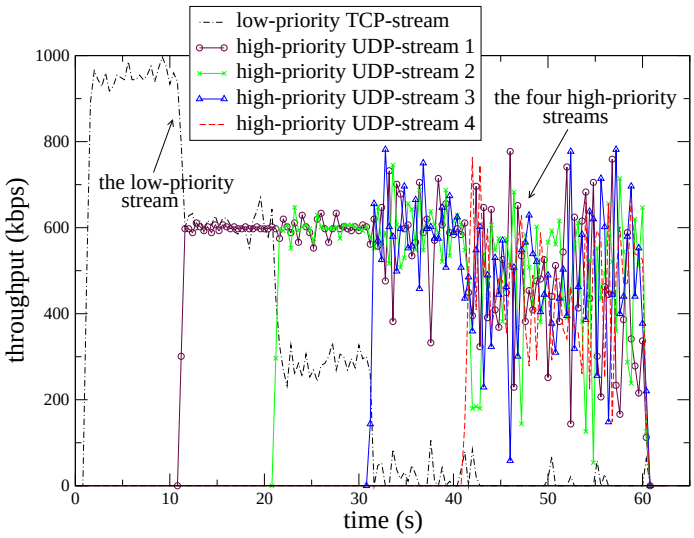


Figure 3: Throughput - EDCA

sioning to high-priority streams, by increasing the proportion of low-priority traffic load in the network.

For each of the six data points (the number of low-priority streams from zero to five) we ran 150 simulations each during 200 simulated seconds. Then, we calculated the average of the 150 averaged values for each data point and plotted them. Because of the extensive simulations, we could calculate the 99% confidence interval of the average end-to-end delay without getting too large intervals.

4.2 Simulation Results

We start by studying the transient behaviour of our scheme compared to the EDCA. There is a low-priority TCP-stream that is started at the 1st second. Then four high-priority UDP-streams are started at the 11th, 21st, 31st and 41st second. All streams continue sending until the 60th second.

Figure 3 and 4 show the throughput of the two schemes at the transport layer. Consequently we must keep in mind that the bit rate on the wireless medium is much higher due to the overhead at the network, data link and physical layer.

Figure 3 shows the case with the EDCA. In the beginning there is a single low-priority TCP-stream, which transmits around 950 kbps. When the first

high-priority UDP-stream starts after 11 seconds, the throughput of the low-priority stream drops down to 600 kbps and it continues to fall for each newly started high-priority stream. Once the fourth high-priority stream is started, the throughput of the low-priority stream falls to extremely low levels. Regarding the high-priority streams, it can be seen that the network behaves pretty well until the 31th second when the third high-priority stream is started. From that time on, the throughput of all high-priority streams starts to fall and the variance of the throughput increases drastically.

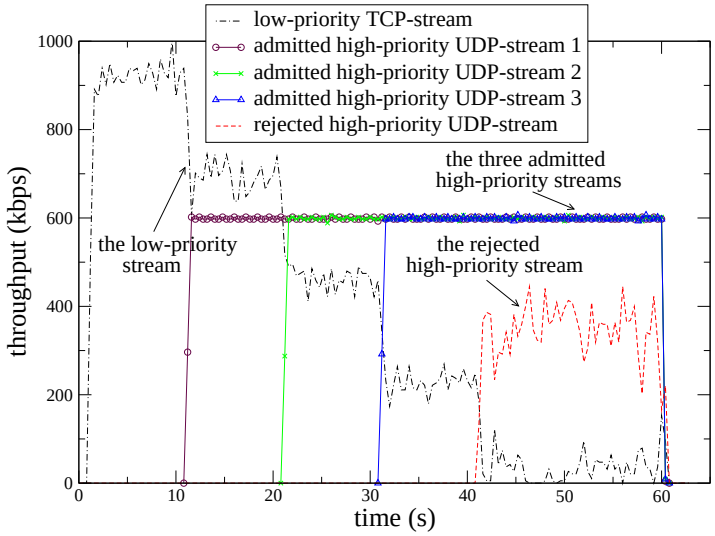


Figure 4: Throughput - our scheme

Figure 4 shows the case with our scheme. Again the simulation starts with a single low-priority TCP-stream, which transmits around 950 kbps. However, in this case the high-priority streams will reserve TXOPs, so if their traffic requests are admitted by the admission control algorithm, they get the amount of bandwidth they require. Thus, we see that the first three high-priority streams have been admitted while the fourth has been rejected. In the case with our scheme, three advantages can be noted compared to the case with the EDCA. First, the low-priority stream has higher throughput due to the fact that the high-priority streams do not collide. Second, the throughput of the admitted high-priority traffic streams is not decreased when new streams are started. Third, the variance of the throughput of the admitted high-priority streams is very low; i.e. the throughput is almost

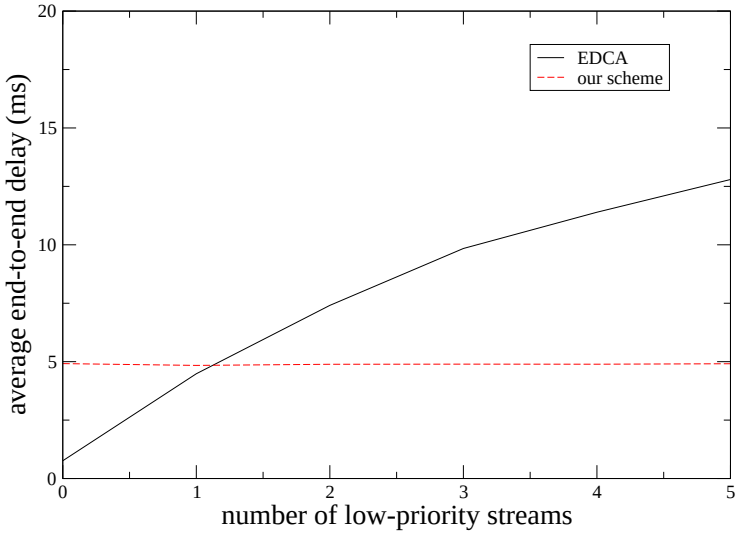


Figure 5: Impact of low-priority streams on average end-to-end delay for a high-priority stream

constant around 600 kbps.

For this transient scenario, our simulations also show that the end-to-end delay and the jitter for the three accepted high-priority streams are kept very low with our scheme, independent of the number of existing streams. It is only the rejected high-priority stream that envisions a notable delay and jitter. This is quite in contrast to the EDCA, which shows much larger delay as well as jitter for all high-priority streams.

Next, the stationary behaviour is studied. Figure 5 shows the average end-to-end delay for a high-priority stream when the number of low-priority streams is increased. As expected, the figure shows that using the EDCA, the end-to-end delay increases when the number of low-priority streams is increased. It is worth mentioning that what we see is a typical behaviour of contention-based medium access schemes and it is this kind of behaviour that we want to avoid for streams with strict QoS requirements, where e.g. the delay must be bounded. In addition, we see that the average end-to-end delay is smaller for the EDCA compared to our solution when the number of low-priority streams is very small. This is also expected, since random-access schemes are known to work well in very lightly loaded networks [16] since the medium access time is very low. When the high-priority stream is the only

Table 3: 99% confidence interval of the average end-to-end delay

nbr of LP-streams	confidence interval (ms)	
	EDCA	our scheme
0	(0.7609,0.7612)	(4.8604,4.9723)
1	(4.4188,4.5478)	(4.7827,4.8968)
2	(7.2580,7.5599)	(4.8319,4.9426)
3	(9.6543,10.0295)	(4.8345,4.9514)
4	(11.2196,11.5705)	(4.8309,4.9543)
5	(12.4882,13.0977)	(4.8502,4.9727)

active stream in the network, there is no waiting time at all for the frames. Consequently, the average end-to-end delay of a frame is almost equal to its transmission time plus the time it takes for the stations to process the frame (i.e. the time it takes for the sender/receiver to send the packet down/up from/to the transport layer). In fact, if it would not be for the lightly error-prone medium causing retransmission from time to time, the average end-to-end delay of a frame would be exactly equal to its transmission time plus the processing time. Hence, the EDCA is working under ideal conditions when the number of low-priority streams is zero. Using our scheme, on the other hand, the high-priority stream reserves TXOPs and can transmit during the reserved TXOPs only. It can be argued that the high-priority stream reserves TXOPs according to its needs (traffic specification), also under very light loads, and as long as the specification is not violated the goal of the traffic stream is fulfilled. The reservation of TXOPs by the high-priority stream results in guaranteed periodical access to the medium. This explains why the end-to-end delay (around 4.9 ms) is unaffected by the number of low-priority streams, i.e. the end-to-end delay is constant, no matter how much background traffic there is in the network.

In Table 3 we can see the 99% confidence interval for the data points in Figure 5. The table shows that the confidence intervals are pretty small in general. It is worth mentioning that the confidence intervals for the EDCA are bigger than the corresponding intervals for our scheme, except when there are no low-priority streams in the network. In addition, we can see that the confidence intervals for the EDCA increase as the number

Table 4: Jitter - our scheme vs. EDCA

nbr of LP-streams	jitter ($10^{-6}s^2$)		$C^2[d]$	
	EDCA	our scheme	EDCA	our scheme
0	0.074	6.6	0.13	0.27
1	37	6.8	1.84	0.29
2	125	7.0	2.28	0.29
3	223	6.9	2.30	0.29
4	275	6.9	2.12	0.29
5	351	6.9	2.14	0.29

of low-priority traffic streams increase while this is not the case for our scheme. These observations are explained by the random nature of the EDCA where high-priority streams must contend for access to the medium using the random backoff time resulting in large variances. In our scheme, this randomness is eliminated for the high-priority streams.

Continuing the study of the stationary behaviour, Table 4 shows the jitter and the $C^2[d]$ for the high-priority stream when the number of low-priority streams is increased. We can see that the jitter is constant low for our scheme independent of the number of low-priority streams. This is exactly what we want to achieve with our scheme. For the EDCA, on the other hand, the jitter starts from very low values and increases to high values as the number of low-priority streams increase. This behaviour is not acceptable for multimedia applications with QoS requirements on constant jitter. The reason for why the jitter and the $C^2[d]$ is very low for the EDCA when the high-priority stream is the only active stream in the network, is the same as why the average end-to-end delay is very low for the EDCA; i.e. there is no waiting time for the frames, which leads to the average end-to-end delay of a frame becoming almost equal to its transmission time plus its processing time. As for the $C^2[d]$, the table shows that the $C^2[d]$ is about 6-7 times larger for the EDCA compared to our scheme (except for the case when the number of low-priority streams is zero, i.e., when the EDCA works under ideal conditions).

5 Conclusion and Future Work

In this paper, we have presented a distributed MAC scheme based on 802.11e for providing QoS guarantees in WLANs operating in ad hoc mode. One advantage with this solution is that it regulates the medium access with a distributed admission control algorithm. Moreover, there is a resource reservation mechanism allowing the stations wishing to send traffic with strict QoS requirements to reserve TXOPs. These TXOPs are scheduled by a distributed scheduler, ensuring that all neighbours have the same schedule. Once a traffic stream is admitted and the TXOPs are reserved and scheduled, the station does not need to contend for medium access for that traffic stream anymore. The distributed scheduler ensures that no station starts a transmission that cannot finish before a reserved TXOP starts; in other words, a station with reserved TXOPs has collision-free deterministic access to the medium. Since our solution is based on existing commonly used protocols and easy to implement, it is a credible candidate for solving the QoS issues in single-hop ad hoc networks.

Our scheme has been compared to 802.11e's contention-free medium access method, the EDCA, which cannot provide any strict QoS guarantees. Through simulations we have shown that our scheme performs better than the EDCA except when the traffic load is very light. The simulations show that our scheme is able to guarantee constant throughput, delay and jitter to multimedia applications with QoS requirements.

The aim in future work will be to further evaluate and enhance our solution. Regarding the enhancement, we plan to add support for e.g. dynamic resource reservation, retransmitting lost ADDTS requests, removing and rescheduling reserved TXOPs for traffic streams that have completed their transmission and handling stations that move in to and out from the network. Finally, it is our aim to develop the implementation further such that it can be used in a multi-hop ad hoc network and reserve resources along a multi-hop route, perhaps with the aid of a QoS-aware ad hoc routing protocol.

References

[1] ANSI/IEEE Std 802.11, "Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1999.

[2] IEEE P802.11e/D10.0, "Part11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 7: Medium Access Control (MAC) Quality of Service (QoS) Enhancements", September 2004.

[3] A. Shepard: "Hybrid Change Makes WLAN QoS Come to Life", April 2004. `www.us.design-reuse.com/articles/article7742.html`.

[4] I. Hwang and C. Wang, "Improving the QoS Performance of EDCA in IEEE 802.11e WLANs Using Fuzzy Set Theory", Active Networking Workshop. 2004.

[5] D. Skyrianoglou, N. Passas and A. Salkintzis, "Traffic Scheduling in IEEE 802.11e Networks Based on Actual Requirements", Mobile Venue '04 Mobile Location Workshop Athens, May 2004.

[6] A. Grilo, M. Macedo and M. Nunes, "A Scheduling Algorithm for QoS Support in IEEE 802.11e Networks". IEEE Wireless Communications Magazine, June 2003, pp. 36-43.

[7] W. Pattara-Atikom and P. Krishnamurthy, "Distributed Mechanisms for Quality of Service in Wireless LANs". IEEE Wireless Communications Magazine, June 2003.

[8] A. Branchs, A. and X. Perez, "Providing Throughput Guarantees in IEEE 802.11 Wireless LAN", Proc. WCNC, 2002.

[9] A. Branchs, A. and X. Perez, "Distributed Weighted Fair Queuing in IEEE 802.11 Wireless LAN", Proc. IEEE ICC, 2002.

[10] N. H. Vaidya, P. Bahl and S. Gupta, "Distributed Fair Scheduling in a Wireless LAN", Proc. ACM MOBICOM, 2000.

[11] W. Pattara-Atikom, S. Banerjee and P. Krishnamurthy, "Starvation Prevention and Quality of Service in Wireless LANs", Proc. IEEE WPMC, 2002.

[12] S. H. Shah, K. Chen and K. Nahrstedt, "Dynamic Bandwidth Management in Single-Hop Ad Hoc Wireless Networks", Proc. IEEE Int'l. Conf. Pervasive Comp. and Commun., 2003.

[13] IEEE Std 802.11b-1999, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band", September 2004.

[14] S. McCanne and S. Floyd, "The Network Simulator - ns-2". K. Fall and K. Varadhan, "The ns Manual". `www.isi.edu/nsnam/ns/`.

[15] M. Moreton: 802.11e patch for ns-2.
`http://cvs.sourceforge.net/viewcvs.py/ns2-wlan-patch/patch\_802\_11/`.

[16] A. Muir and J.J. Garcia-Luna-Aceves, "Group Allocation Multiple Access in Single-Channel Wireless LANs", Proc. Communication Networks and Distributed Systems Modeling and Simulation Conference, 1997.

Appendix - A Lower Bound for the Average End-to-End Delay

The curves in Figure 5, i.e. the average packet delay for the EDCA and our MAC scheme, are results of very detailed and comprehensive simulations. Note that the 99% confidence intervals are very small as seen in Table 3. Our simulations were run on a standard 1.7 GHz PC and the values for each point in the curves, i.e. for each value of the low-priority streams, required about 10 hours of simulation. As mentioned, the simulations are built on ns-2 with a detailed 802.11e implementation for the EDCA [15]. A rough numerical calculation of the presented delays in our proposal validates, at least to some extent, our simulation results for the stationary case.

Regarding our scheme, it is obvious that the end-to-end delay is not affected at all by the number of low-priority streams. As seen from the diagram, the curve is flat slightly below 5 ms. Remember that packets are generated every third ms. The scheduled service interval, SI, is 10 ms and the TXOPs are 2.536 ms long according to the QoS requirements set up by the high-priority source. During each SI, 10/3 packets are generated and thus, on average during each SI 10/3 packets arrive to the MAC sublayer and must be transmitted. Of those packets that are generated during an SI, not more than one may be transmitted during that SI since packets are generated every third ms and the TXOP is just 2.536 ms long.

To calculate the probability that the first packet arriving in an SI (henceforth referred to as $packet_1$), will be transmitted during that SI, we note that it must arrive not only before the TXOP ends but also in time to be transmitted within that TXOP (see Figure 6). The time it takes to transmit $packet_1$ is composed of SIFS (10 μs) + QoS DATA (284 μs) + SIFS (10 μs) + ACK (152 μs) $\Rightarrow$ 456 μs. Hence, the packet must arrive within the first 2536 μs - 456 μs = 2080 μs of the SI. Consequently, the probability that the first packet, generated during an SI, will be transmitted during that SI is 2080/3000. Thus the mean number of packets generated in an SI that also will be sent out in the TXOP in that SI is 2080/3000, i.e. 0.69 packets. This also means that the mean number of packets generated in an SI but sent out in the next SI is 10/3-2080/3000 $\approx$ 2.64 packets. In other words, the average queue length when a new TXOP starts is 2.64 packets.

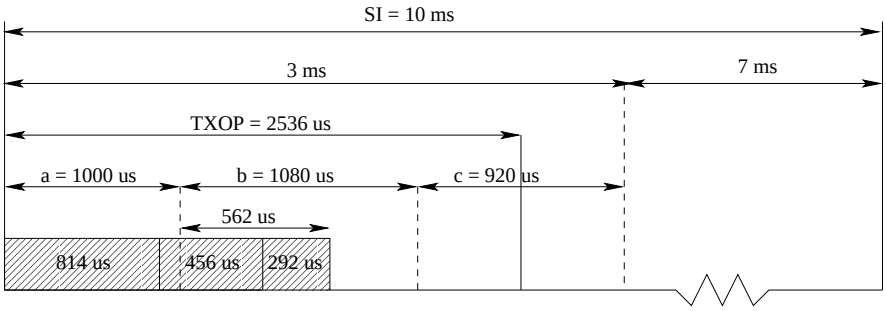


Figure 6: Time relationships

If a packet arrives during the first 2080 μs of an ongoing TXOP, its delay depends on where in the TXOP it arrives. The probabilities for where it arrives during those first 2080 μs and the corresponding delays it will experience can be calculated very straightforward. Given this, then it is easy to calculate the delays for the rest of the packets that arrive during that SI. Note that these remaining packets (on average 2.64 packets) will all be transmitted in the TXOP found in the next SI.

Figure 6 shows the usual situation that the first generated packet in an SI is faced with. There are on average 2.64 packets that must be transmitted before $packet_1$ can be transmitted. In the TXOP, the first of these 2.64 packets will be transmitted during the first 814 μs of the TXOP, the second packet during the next 456 μs and the remaining 0.64 packets during the next 0.64*456=292 μs. In the figure, the three non-overlapping intervals a, b and c are depicted. If $packet_1$ arrives during interval a, it will be transmitted during the running TXOP and there will be another three arrivals in that SI. If $packet_1$ arrives during interval b, then it will be transmitted during the current TXOP, but there will be just another two arrivals. Finally, if $packet_1$ arrives during interval c, it will be transmitted first in the TXOP of the next SI and furthermore there are another two arrivals in the current SI.

The probability that $packet_1$ arrives in interval x is denoted P_x, x $\in$ {a,b,c}. Furthermore, let W_x^i denote the average waiting time for the ith arriving packet in that SI, given that $packet_1$ arrives in interval x. Thus,

$$P_a = \frac{1000}{3000} \approx 0.333, \quad P_b = \frac{1080}{3000} \approx 0.360, \quad P_c = \frac{920}{3000} \approx 0.307$$

Given an arrival in interval x, the average remaining time of that interval after the arrival is half of the length of the interval. Thus, the average delay of an arrival in interval x consists of the remaining time of interval x (the first term), plus the transmission time of the remaining packets in the queue (any term in the middle), plus its own transmission time (the last term):

$$W_a^1 = \frac{1000}{2} + 562 + 456 = 1518\mu s, \quad W_b^1 = \frac{562}{1080} \times \frac{562}{2} + 456 \approx 602\mu s$$

$$W_c^1 = \frac{920}{2} + 7000 + 814 = 8274\mu s$$

Note that those packets that arrive during interval b will experience an average waiting time equal to 562/2 μs and that with the probability 562/1080. In addition, note that remaining packet arrivals, on average 2.64, will be transmitted first during the next SI. The second packet arriving exactly 3000 μs after the first one, given that the first one arrived in interval x, will experience an average delay equal to:

$$W_a^2 = 7000 - \frac{1000}{2} + 814 = 7314\mu s, \quad W_b^2 = 7000 - 1000 - \frac{1080}{2} + 814 = 6274\mu s$$

$$W_c^2 = W_c^1 - 3000 + 456 = 5730\mu s$$

where the last term is the transmission time. Similar reasoning gives the waiting time for the third packet:

$$W_x^3 = W_x^2 - 3000 + 456 = W_x^2 - 2544\mu s$$

A fourth packet will arrive during the SI only if $packet_1$ arrived in interval a. Then the waiting time for the fourth packet is:

$$W_a^4 = W_a^3 - 3000 + 456 = 2226\mu s$$

The average delay for all packets arriving in one and the same SI given that the first packet arrives in interval x is:

$$W_x = \begin{cases} \frac{1}{4}\sum_{i=1}^{4} W_x^i, & x = a \\ \frac{1}{3}\sum_{i=1}^{3} W_x^i, & x = b, c \end{cases}$$

This gives $W_a = 3957\ \mu s$, $W_b \approx 3535\ \mu s$ and $W_c = 5730\ \mu s$. The total average waiting time W is given by:

$$W = \sum_{x \in \{a,b,c\}} P_x \times W_x$$

Straightforward calculations then give W $\approx 4349\ \mu s$. To that comes the time it takes to process the packets, which is 25 μs for both the source and the destination, i.e. 50 μs in total. This gives a lower bound of 4399 μs for the average end-to-end delay, which should be compared to the simulation result of about 4900 μs. The missing microseconds are due to the error-prone wireless medium causing retransmissions and which in turn increase the average end-to-end delay.

Providing QoS Guarantees in Ad Hoc Networks through EDCA with Resource Reservation

Ali Hamidian

Abstract

As the use of WLANs based on IEEE 802.11 increase, the need for QoS becomes more obvious. The upcoming IEEE 802.11e aims at providing QoS, but its contention-based medium access mechanism *enhanced distributed channel access* (EDCA), provides only service differentiation, i.e. soft QoS. In order to provide hard QoS, we have proposed an extension called *EDCA with resource reservation* (EDCA/RR), which enhances EDCA by offering also hard QoS through resource reservation. This report focuses on EDCA/RR with the aim to enhance the scheme further in single-hop scenarios but also to present an idea of how to extend the scheme to be useful also in multi-hop ad hoc networks.

1 Introduction

The widespread use of portable devices equipped with *wireless local area network* (WLAN)-capability is likely to increase the popularity of ad hoc networks. As the WLAN technique continues to grow and mature, the users expect to use the wireless network the same way as they use an ordinary personal computer connected to a *local area network* (LAN). Thus, the users want to have the possibility to use the same demanding applications as they run on their personal computers; e.g. to see and talk to friends using an instant messaging program.

To meet with such demands and support multimedia applications with *quality of service* (QoS) requirements, the upcoming IEEE 802.11e standard [1] introduces the new *hybrid coordination function* (HCF). It is called hybrid because it has both a contention-based and a contention-free medium access method in a single medium access protocol. The contention-based *enhanced distributed channel access* (EDCA) provides QoS by delivering traffic based on differentiating user priorities while the contention-free *HCF controlled channel access* (HCCA) provides QoS by allowing for reservation of transmission time.

Although the HCCA is an important enhancement that aims at providing hard QoS in WLANs, it is the EDCA that has received most attention so far, and it is possible that EDCA's destiny will be similar to the one of its predecessor DCF, i.e. it will be implemented by the majority of the vendors, whereas the HCCA might be somewhat neglected just as the PCF - despite the fact that HCCA is a great improvement compared to its predecessor. In addition, the EDCA is a distributed channel access method and can be used in ad hoc networks while the HCCA is centralized and thus only usable in infrastructure networks. Therefore, the focus of this report lies on the EDCA.

There has been a lot of research on providing QoS to ad hoc networks. However, many of these suggest proprietary protocols - based on times division multiple access, multiple channels, etc. It is our belief that any realistic proposal must be based on the widely spread de facto standard IEEE 802.11 [2]. Hence, in this report we propose a mechanism, supporting QoS in ad hoc networks, based on IEEE 802.11 and IEEE 802.11e. Consequently,

it can be integrated into existing systems without much difficulty.

The remainder of this report is organized as follows: Section 2 gives an overview of our previous work enhancing the EDCA. In Section 3 we discuss further enhancements applied to the scheme and present some ideas for extending the scheme for multi-hop networks. Finally, Section 4 concludes this report and gives some directions for future work.

2 The Original EDCA/RR

In a previous work we have enhanced the EDCA medium access mechanism to provide QoS guarantees by reserving *transmission opportunities* (TXOPs) for traffic streams with strict QoS requirements [3]. Before starting with the enhancements based on that work, it is necessary to give an introduction to our proposed scheme in order to facilitate the reading and understanding of the rest of this report. Although not named in [3], our scheme is called *EDCA with resource reservation* (EDCA/RR) in this report.

The EDCA/RR works like the EDCA as long as there is no station that needs to reserve TXOPs for its high-priority traffic stream. Once a station (sender) wishes to reserve TXOPs to be able to send traffic with strict QoS requirements, it requests admission for its traffic stream. The admission control request is not sent to any central station such as a *QoS access point* (QAP), but is handled internally within the sender by an admission control algorithm. The sender either admits or rejects its own requested traffic stream according to the admission control algorithm. At this point, we should point out that our scheme is not dependent on any specific admission control or scheduling algorithm; thus, it is possible to use any proposed enhancement (such as those presented in [4, 5, 6, 7]) to the reference design algorithms provided in the IEEE 802.11e specification (for details see [3] or [1]). However, in our EDCA/RR implementation, the reference admission control and scheduling algorithms have been used - partly because they are specified in the IEEE 802.11e specification making them widely known and giving them certain acceptance, and partly because they are relatively easy to implement.

In case the traffic stream is rejected, the sender can try to lower its QoS demands and retry. On the other hand, if the traffic stream is admitted,

the sender schedules its traffic by setting the *scheduled service interval* (SI) and the *service start time* (SST) parameters. Details about the calculation of these parameters can be found in [3]. Next, the sender broadcasts an *add traffic stream* (ADDTS) request containing a *traffic specification* (TSPEC) element with information such as mean data rate, nominal frame size, SST and SI. All stations that receive the ADDTS request store the information of the sender's SST and SI, and schedule the new traffic stream exactly as the sender. This ensures that no station starts a transmission that cannot be finished before a reserved TXOP starts and thus collision-free access to the medium is offered to the streams with reserved TXOPs. In order to make sure that all stations have similar schedules, all neighbours have to unicast an ADDTS response back to the sender to acknowledge a received ADDTS request.

Every time the sender receives an ADDTS response from a neighbour, it stores the address of the neighbour. After receiving a response from all neighbours, the sender waits until the SST specified in the TSPEC element and initiates a transmission. If the time instant when all responses are received occurs later than the advertised SST, the transmission is delayed until the next TXOP. During a TXOP, the sender can transmit multiple frames but it must stop sending when the remaining time of the TXOP is less than the transmission time of another data frame plus its corresponding ACK. Once the TXOP is finished, the station waits until the next TXOP, which occurs after an SI. A station that has reserved TXOPs for a traffic stream with strict QoS requirements, is not allowed to transmit frames belonging to that stream at time instants other than during the reserved TXOPs. Of course the station is allowed to transmit frames from other traffic streams, in other ACs, by contending for access to the medium. However, these streams and other low-priority streams from other stations must ensure to finish their transmission before a TXOP starts; otherwise the contending station must backoff and the frames are not allowed to be sent until after the reserved TXOP(s).

When a transmission failure occurs during a TXOP, the station does not start a backoff procedure. Instead, it retransmits the failed frame after SIFS if there is enough time left in the TXOP to complete the transmission.

3 Enhancing the EDCA/RR

In the previous section we described EDCA/RR that works fine in a WLAN operating in ad hoc mode, i.e. in a single-hop ad hoc network where all stations are within each other's transmission range. Although single-hop ad hoc networks might be seen as limited, we must remember that the main application area for the EDCA is a WLAN and not a multi-hop ad hoc network. To give an example of the application area for single-hop ad hoc networks where our scheme can be used, we can mention network gaming where players can use their laptops to play demanding network games with each other at no cost anywhere they want; i.e. without needing to worry about (neither wired nor wireless) Internet connections. However, since providing QoS in a multi-hop ad hoc network is also desirable, besides enhancing our scheme for the single-hop case, in this report we aim at enhancing the EDCA even further such that it can be used to provide QoS guarantees in a multi-hop ad hoc network.

In this section we start by describing our solution to a problem in EDCA/RR related to hidden stations. Next we identify some problems that might occur due to mobile stations leaving and entering a network. Finally we present a conceivable solution to extend EDCA/RR such that it can be used in multi-hop ad hoc networks.

3.1 The Hidden Station Problem in EDCA/RR

In the original version of EDCA/RR, the hidden station problem was handled through the exchange of *request to send* (RTS) and *clear to send* (CTS) frames, i.e. the same way as in IEEE 802.11(e). However, this method is not sufficient since in EDCA/RR, stations hidden to a station that has reserved TXOPs can cause other problems than the well-known hidden station problem (causing collisions). Contending stations that have received a TSPEC from the reserving station do not start a transmission unless it finishes before a TXOP starts. But unfortunately, stations hidden from the reserving station do not receive any TSPEC so they do not know when the TXOPs start. Therefore, they might start a transmission that extend across a TXOP.

To illustrate the problem with a hidden station in an ad hoc network using our scheme, suppose there are three stations in a row (see Figure 1):

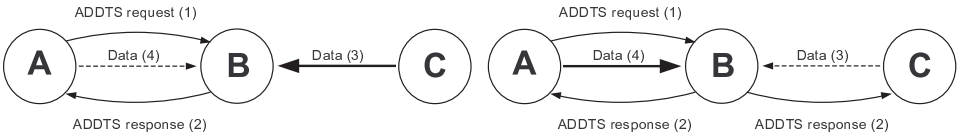


Figure 1: C is hidden from A and can start transmitting just before A's TXOP starts.

Figure 2: C is informed about the TXOP reservation of A and defers during A's TXOP.

A, B and C, where A and B as well as B and C are within each other's transmission range but A and C cannot hear each other. Assume further that A wants to send QoS traffic to B so it has broadcasted an ADDTS request and B has replied with an ADDTS response. However, C (that is hidden from A) is unaware of A's TXOP reservation since it has not received A's ADDTS request so there is a chance that C starts transmitting just before a TXOP reserved by A is about to start. In that case a collision would occur during A's reserved TXOP meaning that A would no longer have collision-free access to the medium. In order to prevent C from transmitting just before a reserved TXOP is about to start, it must become aware of A's TXOP reservation. In other words, the reservation schedule of any sender must be known by all stations within two hops from the sender.

There are different ways of achieving this goal, i.e. to spread the TSPEC to stations outside of the reserving station's transmission range. One approach can be to rebroadcast the ADDTS request sent by the reserving station during the TXOP reservation. Hence, in our example B would rebroadcast the ADDTS request of A to let also C receive the request frame. However, there are many problems related to this approach. First, should C send an ADDTS response to B just like B has to send an ADDTS response to A? Before answering this question we must remember that there might be many stations at the same distance from A as B and C respectively (i.e. one and two hops away from A respectively). This means that if C has to respond to B then every other station two hops away from A should also respond to B because those are also hidden stations. Moreover, this procedure would continue until all stations one hop away from A rebroadcast the ADDTS request from A, and all stations two hops away from A send back an ADDTS response. Obviously, this would lead to a lot of overhead

and a significant increase in the reservation delay. On the other hand, if C does not have to send a response to B, then B cannot be sure whether the rebroadcasted ADDTS request was received by C or not.

Another approach to spread the TSPEC is to let the ADDTS responses contain the TSPEC and let all stations overhear these frames (see Figure 2). This way, the TSPEC is known to all stations within two hops from the sender with no additional signaling frame and with limited increase of overhead. Thus, when B sends an ADDTS response back to A, C will hear this frame and save the information included in the TSPEC, i.e. SST and SI of A. This approach is much less complex and results in less overhead than the previous approach. However, again B cannot be sure whether the ADDTS response was received correctly by C or not. Therefore, we let reserving stations transmit special RTS/CTS frames extended to contain a TSPEC (RTS_TSPEC and CTS_TSPEC), in the beginning of a TXOP. This way, a station with an out-of-date reservation schedule has the chance to update its schedule. Although one might think that this increases the overhead too much, we must remember that the RTS_TSPEC and CTS_TSPEC frames are sent only at the beginning of a TXOP and not for every single data frame.

3.2 Leaving and Entering the Network

An important issue that needs special attention is mobility and in particular stations leaving and entering the network. For example, if a station with reserved TXOPs leaves the network, the other stations must become aware of that because otherwise they will defer from transmitting although they should not and the network capacity will be wasted. On the other hand, if a station enters a network where other stations have reserved TXOPs, it cannot start contending for access to the medium and transmit because such a transmission might collide with the transmission in a reserved TXOP.

A conceivable solution to these problems is to let the stations in the network listen for frames in the beginning of each TXOP in order to determine whether the reserved TXOPs are still in use. If there is no transmission within the first time period equal to DIFS, the TXOP is considered to be unused and can be used for transmission by other stations. This kind of situations might occur occasionally when a station with reserved TXOPs

has no frames to send during certain TXOPs. If several consecutive TXOPs are determined to be unused, the receiver can ask the sender if it still has something to send. If the sender does not respond despite several attempts, the receiver and other stations can assume that the sender has left the network and delete the TXOP reservation completely. In that case, the TXOPs must be deallocated. Furthermore, possible traffic streams after the terminated stream, shall be moved back to use the unused time so that a reserved TXOP starts just after another has finished in order to avoid time gaps between two reserved TXOPs. Moving the streams is done pretty easily thanks to the distributed characteristic of EDCA/RR. There is no need for any signaling; each station performs the rescheduling itself in a distributed manner.

A station that enters the network must update its schedule before it is allowed to transmit any frame. This can be done by setting a schedule update bit in beacons or other frames exchanged during the initialization process.

3.3 QoS Provisioning in Multi-hop Ad Hoc Networks

The first goal of this report was to enhance the EDCA/RR scheme operating in single-hop ad hoc networks. In particular, we wanted to solve the problems that could occur due to hidden stations. We have achieved this goal and presented our solution above. Another goal was to propose an extension to the scheme such that it can be used to provide QoS guarantees in a multi-hop ad hoc network. For this purpose, we need an ad hoc routing protocol that can find a route between the communicating stations. In this report we assume that the routing protocol is reactive, i.e. the route discovery process is performed on-demand. Examples of two popular reactive routing protocols are *Ad hoc On-Demand Distance Vector* (AODV) [8] and *Dynamic MANET On-demand* (DYMO) [9]. During the route discovery process the source broadcasts a *route request* (RREQ) throughout the network to find the destination. When the destination receives the RREQ, it responds with a *route reply* (RREP) unicast toward the source. In this section we present an idea of how to extend EDCA/RR in order to be able to provide QoS in a multi-hop ad hoc network.

To illustrate the idea, let us assume there are three stations in a row (see

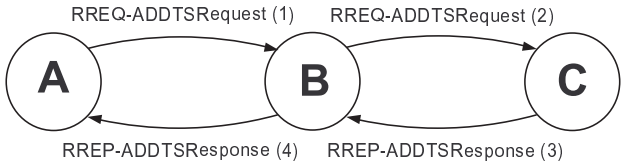


Figure 3: Simultaneous multi-hop route discovery and resource reservation.

Figure 3): A, B and C, where A and B as well as B and C are within each other's transmission range but A and C cannot hear each other. Assume further that A wants to send high-priority traffic (with QoS requirements) to C.

To reserve resources along a multi-hop route, the QoS requirements of A's traffic stream must be known by the routing protocol so that it can use the requirements during the route discovery process. However, the routing protocol shall not start its route discovery process before the traffic stream has been admitted by the admission control mechanism at A's MAC sublayer. In EDCA/RR the resource reservation process (including admission control) starts when the first packet of a traffic stream with QoS requirements reaches the MAC sublayer, i.e. after it has been handled by the routing protocol at the network layer. To prevent the routing protocol searching for a route using its usual metrics (and thus not considering the QoS requirements of the traffic stream), it must be modified to co-operate with the protocol at the MAC sublayer (EDCA/RR in our case). Therefore, once the first high-priority frame of a traffic stream in station A reaches its network layer, the routing protocol must be modified to signal EDCA/RR to check whether the requested traffic stream can be admitted or not. If the traffic stream is rejected, A's application can either try to lower its QoS demands and retry or accept the fact that there are not enough resources to be reserved. Thus, in case of rejection, the MAC sublayer must inform the network layer about this fact in order to trigger the routing protocol to find a (normal, i.e. non-QoS) route to the destination. On the other hand, if the traffic is admitted, the MAC sublayer shall notify the routing protocol and send it the necessary information regarding the required QoS. Since the QoS requirements are gathered in a TSPEC, it is suitable to use the TSPEC (or possibly a part of it) in order to inform the routing protocol about the QoS

requirements of the admitted traffic stream.

At this point A has determined that it has enough resources to reserve so it can start its route discovery process to search for a route that can handle its QoS requirements. Therefore, A broadcasts a RREQ-ADDTSRequest, i.e. a RREQ message including a TSPEC.

When B receives the RREQ-ADDTSRequest, its MAC sublayer handles the ADDTSRequest part of the message to check whether the traffic can be admitted or not. In case the traffic is rejected, the RREQ-ADDTSRequest will be dropped. However, although some applications need a certain minimum level of QoS for functioning, others can function despite that the QoS level is not sufficient. Therefore, in the latter case, B may broadcast an ordinary RREQ searching for a normal route to the destination. On the other hand, if the traffic is admitted, the MAC sublayer notifies the routing protocol to rebroadcast the RREQ-ADDTSRequest. Station C processes the RREP-ADDTSRequest mainly as in B. However, if the traffic is admitted, the MAC sublayer schedules the traffic stream of A and notifies the routing protocol to send a RREP-ADDTSResponse back to the source (i.e. station A).

When B receives the RREP-ADDTSResponse, its MAC sublayer handles the ADDTSResponse part of this message to schedule the traffic stream (since now the traffic stream has been admitted by all stations from the source to the destination). Then the network layer forwards the RREP-ADDTSResponse to A. Station A processes the RREP-ADDTSResponse just as in B and thus, the resource reservation is finished and the traffic stream can start transmitting during its reserved TXOPs.

4 Conclusion

This report has considered the QoS issues in ad hoc networks. In particular, the aim was to extend and enhance the previously proposed EDCA/RR scheme, which allows multimedia applications to reserve medium time according to their needs (specified in a TSPEC). The scheme has been enhanced to prevent hidden stations causing collisions during reserved TXOPs. The main idea was to spread the information about the TXOP reservation (included in a TSPEC) such that also hidden stations become aware of the

reservation and thus, defer during the reserved TXOPs.

However, the EDCA/RR scheme was designed for WLANs operating in ad hoc network configuration. Although such a scheme can be useful in application areas such as network gaming, our aim was to extend it to be useful also in multi-hop ad hoc networks since these networks are expected to offer new communication possibilities. Thus, we have presented an extension to the scheme such that it can be used together with an ad hoc routing protocol to find multi-hop QoS-enabled routes between the communicating stations. Thus, a station with traffic requiring QoS will be able to reserve TXOPs for deterministic medium access along a multi-hop route to the destination. As part of our future work, this extension will be tuned and incorporated into the existing enhanced EDCA/RR implementation.

References

[1] IEEE P802.11e/D13.0, "Part11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 7: Medium Access Control (MAC) Quality of Service (QoS) Enhancements", September 2004.

[2] ANSI/IEEE Std 802.11, "Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1999.

[3] A. Hamidian, U. Körner, "An Enhancement to the IEEE 802.11e EDCA Providing QoS Guarantees". Telecommunication Systems Journal, 2005.

[4] D. Skyrianoglou, N. Passas and A. Salkintzis, "Traffic Scheduling in IEEE 802.11e Networks Based on Actual Requirements", Mobile Venue '04 Mobile Location Workshop Athens, May 2004.

[5] A. Grilo, M. Macedo and M. Nunes, "A Scheduling Algorithm for QoS Support in IEEE 802.11e Networks". IEEE Wireless Communications Magazine, June 2003, pp. 36-43.

[6] W. F. Fan, D. Gao, D. H. K. Tsang, "Admission Control for Variable Bit Rate traffic in IEEE 802.11e WLANs". 2004.

[7] P. Ansel, Q. Ni, T. Turletti, "FHCF: A Fair Scheduling Scheme for 802.11e WLAN". Research report number 4883, INRIA Sophia Antipolis, July 2003.

[8] C. Perkins, E. M. Belding-Royer and S. Das. *"Ad hoc On-Demand Distance Vector (AODV) Routing"*. Experimental RFC 3561.

[9] R. Ogier, M. Lewis and F. Templin. *"Dynamic MANET On-demand (DYMO) Routing"*. IETF Internet draft.

Reports on Communication Systems

101. **On Overload Control of SPC-systems**
Ulf Körner, Bengt Wallström, and Christian Nyberg, 1989.

102. **Two Short Papers on Overload Control of Switching Nodes**
Christian Nyberg, Ulf Körner, and Bengt Wallström, 1990.

103. **Priorities in Circuit Switched Networks**
Åke Arvidsson, Ph.D. thesis, 1990.

104. **Estimations of Software Fault Content for Telecommunication Systems**
Bo Lennselius, Lic. thesis, 1990.

105. **Reusability of Software in Telecommunication Systems**
Anders Sixtensson, Lic. thesis, 1990.

106. **Software Reliability and Performance Modelling for Telecommunication Systems**
Claes Wohlin, Ph.D. thesis, 1991.

107. **Service Protection and Overflow in Circuit Switched Networks**
Lars Reneby, Ph.D. thesis, 1991.

108. **Queueing Models of the Window Flow Control Mechanism**
Lars Falk, Lic. thesis, 1991.

109. **On Efficiency and Optimality in Overload Control of SPC Systems**
Tobias Rydén, Lic. thesis, 1991.

110. **Enhancements of Communication Resources**
Johan M Karlsson, Ph.D. thesis, 1992.

111. **On Overload Control in Telecommunication Systems**
Christian Nyberg, Ph.D. thesis, 1992.

112. **Black Box Specification Language for Software Systems**
Henrik Cosmo, Lic. thesis, 1994.

113. **Queueing Models of Window Flow Control and DQDB Analysis**
Lars Falk, Ph.D. thesis, 1995.

114. **End to End Transport Protocols over ATM**
Thomas Holmström, Lic. thesis, 1995.

115. **An Efficient Analysis of Service Interactions in Telecommunications**
Kristoffer Kimbler, Lic. thesis, 1995.

116. **Usage Specifications for Certification of Software Reliability**
Per Runeson, Lic. thesis, May 1996.

117. **Achieving an Early Software Reliability Estimate**
Anders Wesslén, Lic. thesis, May 1996.

118. **On Overload Control in Intelligent Networks**
Maria Kihl, Lic. thesis, June 1996.

119. **Overload Control in Distributed-Memory Systems**
Ulf Ahlfors, Lic. thesis, June 1996.

120. **Hierarchical Use Case Modelling for Requirements Engineering**
Björn Regnell, Lic. thesis, September 1996.

121. **Performance Analysis and Optimization via Simulation**
Anders Svensson, Ph.D. thesis, September 1996.

122. **On Network Oriented Overload Control in Intelligent Networks**
Lars Angelin, Lic. thesis, October 1996.

123. **Network Oriented Load Control in Intelligent Networks Based on Optimal Decisions**
Stefan Pettersson, Lic. thesis, October 1996.

124. **Impact Analysis in Software Process Improvement**
Martin Höst, Lic. thesis, December 1996.

125. **Towards Local Certifiability in Software Design**
Peter Molin, Lic. thesis, February 1997.

126. **Models for Estimation of Software Faults and Failures in Inspection and Test**
Per Runeson, Ph.D. thesis, January 1998.

127. **Reactive Congestion Control in ATM Networks**
Per Johansson, Lic. thesis, January 1998.

128. **Switch Performance and Mobility Aspects in ATM Networks**
Daniel Søbirk, Lic. thesis, June 1998.

129. **VPC Management in ATM Networks**
Sven-Olof Larsson, Lic. thesis, June 1998.

130. **On TCP/IP Traffic Modeling**
Pär Karlsson, Lic. thesis, February 1999.

131. **Overload Control Strategies for Distributed Communication Networks**
Maria Kihl, Ph.D. thesis, March 1999.

132. **Requirements Engineering with Use Cases - a Basis for Software Development**
Björn Regnell, Ph.D. thesis, April 1999.

133. **Utilisation of Historical Data for Controlling and Improving Software Development**
Magnus C. Ohlsson, Lic. thesis, May 1999.

134. **Early Evaluation of Software Process Change Proposals**
Martin Höst, Ph.D. thesis, June 1999.

135. **Improving Software Quality through Understanding and Early Estimations**
Anders Wesslén, Ph.D. thesis, June 1999.

136. **Performance Analysis of Bluetooth**
Niklas Johansson, Lic. thesis, March 2000.

137. **Controlling Software Quality through Inspections and Fault Content Estimations**
Thomas Thelin, Lic. thesis, May 2000.

138. **On Fault Content Estimations Applied to Software Inspections and Testing**
Håkan Petersson, Lic. thesis, May 2000.

139. **Modeling and Evaluation of Internet Applications**
Ajit K. Jena, Lic. thesis, June 2000.

140. **Dynamic traffic Control in Multiservice Networks - Applications of Decision Models**
Ulf Ahlfors, Ph.D. thesis, October 2000.

141. **ATM Networks Performance - Charging and Wireless Protocols**
Torgny Holmberg, Lic. thesis, October 2000.

142. **Improving Product Quality through Effective Validation Methods**
Tomas Berling, Lic. thesis, December 2000.

143. **Controlling Fault-Prone Components for Software Evolution**
Magnus C. Ohlsson, Ph.D. thesis, June 2001.

144. **Performance of Distributed Information Systems**
Niklas Widell, Lic. thesis, February 2002.

145. **Quality Improvement in Software Platform Development**
Enrico Johansson, Lic. thesis, April 2002.

146. **Elicitation and Management of User Requirements in Market-Driven Software Development**
Johan Natt och Dag, Lic. thesis, June 2002.

147. **Supporting Software Inspections through Fault Content Estimation and Effectiveness Analysis**
Håkan Petersson, Ph.D. thesis, September 2002.

148. **Empirical Evaluations of Usage-Based Reading and Fault Content Estimation for Software Inspections**
Thomas Thelin, Ph.D. thesis, September 2002.

149. **Software Information Management in Requirements and Test Documentation**
Thomas Olsson, Lic. thesis, October 2002.

150. **Increasing Involvement and Acceptance in Software Process Improvement**
Daniel Karlström, Lic. thesis, November 2002.

151. **Changes to Processes and Architectures; Suggested, Implemented and Analyzed from a Project viewpoint**
Josef Nedstam, Lic. thesis, November 2002.

152. **Resource Management in Cellular Networks -Handover Prioritization and Load Balancing Procedures**
Roland Zander, Lic. thesis, March 2003.

153. **On Optimisation of Fair and Robust Backbone Networks**
Pål Nilsson, Lic. thesis, October 2003.

154. **Exploring the Software Verification and Validation Process with Focus on Efficient Fault Detection**
Carina Andersson, Lic. thesis, November 2003.

155. **Improving Requirements Selection Quality in Market-Driven Software Development**
Lena Karlsson, Lic. thesis, November 2003.

156. **Fair Scheduling and Resource Allocation in Packet Based Radio Access Networks**
Torgny Holmberg, Ph.D. thesis, November 2003.

157. **Increasing Product Quality by Verification and Validation Improvements in an Industrial Setting**
Tomas Berling, Ph.D. thesis, December 2003.

158. **Some Topics in Web Performance Analysis**
Jianhua Cao, Lic. thesis, June 2004.

159. **Overload Control and Performance Evaluation in a Parlay/OSA Environment**
Jens K. Andersson, Lic. thesis, August 2004.

160. **Performance Modeling and Control of Web Servers**
Mikael Andersson, Lic. thesis, September 2004.

161. **Integrating Management and Engineering Processes in Software Product Development**
Daniel Karlström, Ph.D. thesis, December 2004.

162. **Managing Natural Language Requirements in Large-Scale Software Development**
Johan Natt och Dag, Ph.D. thesis, Febuary 2005.

163. **Designing Resilient and Fair Multi-layer Telecommunication Networks**
Eligijus Kubilinskas, Lic. thesis, Febuary 2005.

164. **Internet Access and Performance in Ad hoc Networks**
Anders Nilsson, Lic. thesis, April 2005.

165. **Active Resource Management in Middleware and Service-oriented Architectures**
Niklas Widell, Ph.D. thesis, May 2005.

166. **Quality Improvement with Focus on Performance in Software Platform Development**
Enrico Johansson, Ph.D. thesis, June 2005.

167. **On Inter-System Handover in a Wireless Hierarchical Structure**
Henrik Persson, Lic. thesis, September 2005.

168. **Prioritization Procedures for Resource Management in Cellular Network**
Roland Zander, Ph.D. thesis, November 2005.

169. **Strategies for Management of Architectural Change and Evolution**
Josef Nedstam, Ph.D. thesis, December 2005.

170. **Internet Access and QoS in Ad Hoc Networks**
Ali Hamidian, Lic. thesis, April 2006.