SWEDISH DEFENCE
RESEARCH AGENCY

FOI-R- -1156- -SE
January 2004
ISSN 1650-1942
Scientific Report

Otto Tronarp

Quality of Service in Ad Hoc Networks by Priority Queuing

Command and Control Systems
P.O. Box 1165
SE-581 11 LINKÖPING
SWEDEN

FOI- SWEDISH DEFENCE RESEARCH AGENCY
Command and Control Systems
P.O. Box 1165
SE-581 11 LINKÖPING
SWEDEN

FOI-R- -1156- -SE
January 2004
ISSN 1650-1942
Scientific Report

Otto Tronarp

Quality of Service in Ad Hoc Networks by Priority Queuing

Issuing organization

FOI- Swedish Defence Research Agency
Command and Control Systems
P.O. Box 1165
SE-581 11 LINKÖPING
SWEDEN

Report number, ISRN

FOI-R- -1156- -SE

Report type

Scientific Report

Research area code

4. C4ISR

Month year

January 2004

Project No.

E7035

Customers code

5. Contracted Research

Sub area code

41. C4I

Author/s

Otto Tronarp

Project manager

Mattias Sköld

Approved by

Martin Rantzer

Sponsoring Agency

Swedish Armed Forces

Scientifically and technically responsible

Jan Nilsson

Report title

Quality of Service in Ad Hoc Networks by Priority Queuing

Abstract

The increasing usage of information technology in military affairs raises the need for robust high capacity radio networks that will be used to provide several different types of services, such as group calls and situation awareness services. As all services have specific demands on packet delays and packet losses in order to be fully functional, there is a need for a *quality-of-service* (QoS) mechanism in the network.

In this report we examine the possibility of providing a QoS mechanism in ad hoc networks by using priority queues. The study includes two different queuing schemes, namely fixed priority queuing and weighted fair queuing. The performance of the two queuing schemes is evaluated and compared with respect to their ability to provide differentiation in network delay, i.e. provide high-priority traffic with lower delays than low-priority traffic. The study is mainly done by simulations, but for fixed priority queuing we also derive an analytical approximation of the network delay.

Our simulations show that fixed priority queuing provides a sharp delay differentiation between service classes, whereas weighted fair queuing provides the ability to control the delay differentiation. One of those queuing schemes alone might not be the best solution for providing QoS. Instead we suggest that a combination of them be used.

Keywords

quality of service, ad hoc networks, priority queues, weighted fair queuing

Further bibliographic information

Language English

Also published at Linköpings University LiTH-ISY-EX-3343-2003

ISSN 1650-1942

Pages 48 p.

Price acc. to pricelist

Utgivare Totalförsvarest Forskningsinstitut-FOI Ledningssystem Box 1165 581 11 LINKÖPING	**Rapportnummer, ISRN** FOI-R- -1156- -SE	**Klassificering** Vetenskaplig Rapport
	Forskningsområde 4. Spaning och ledning	
	Månad, år Januari 2004	**Projektnummer** E7035
	Verksamhetsgren 5. Uppdragsfinansierad verksamhet	
	Delområde 41. Ledning med samband och telekom och IT-system	
Författare Otto Tronarp	**Projektledare** Mattias Sköld	
	Godkänd av Martin Rantzer	
	Uppdragsgivare/kundbeteckning Försvarsmakten	
	Teknisk och/eller vetenskapligt ansvarig Jan Nilsson	

Rapportens titel

Quality of Service in Ad Hoc Networks by Priority Queuing

Sammanfattning

Den ökande användingen av informationsteknologi inom den militära sektorn skapar ett behov av robusta radionätverk med hög kapacitet. Genom nätverket kommer en mängd olika tjänster att erbjudas, såsom gruppsamtal och positionsförmedling. De olika tjänsterna ställer skilda krav på paketfördröjning och paketförlust, för att fungera tillfredställande. Det finns därför ett behov att kunna styra tjänstekvaliteten som en specifik tjänst erhåller.
I det här arbetet undersöker vi möjligheten att förse nätverket med en funktion för att styra den tjänstekvalitet, som en specifik tjänst erhåller, med hjälp av prioritetsköer. Vi studerar två olika kösystem, fixed priority queuing och weighted fair queuing.
De två kösystemen utvärderas och jämförs med avseende på förmåga att styra nätverksfördröjningen, det vill säga ge högprioriterad trafik lägre fördröjning än lågprioriterad trafik. Resultaten har i första hand tagits fram genom simuleringar, men för fixed priority queuing har vi även härlett en analytisk approximativ formel för nätverksfördröjningen.
Simuleringsresultaten visar att fixed priority queuing ger en skarp fördröjningsdifferentiering mellan tjänsteklasser, medan weighted fair queuing ger möjlighet att styra fördröjningsdifferentieringen. Ett av dessa kösystem är förmodligen inte ensamt den bästa lösningen, i stället föreslår vi en kombination av de två kösystemen.

Nyckelord

tjänstekvalitet, ad hoc-nätverk, prioritetsköer

Övriga bibliografiska uppgifter	**Språk** Engelska
Har även publicerats av Linköpings universitet LiTH-ISY-EX-3343-2003	
ISSN 1650-1942	**Antal sidor:** 48 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Contents

1 Introduction **9**
1.1 Background 9
1.2 Problem overview 11
1.2.1 Medium Access Control 11
1.2.2 Queuing systems 13
1.3 Problem definition 14
1.4 Outline 15

2 Network Model **17**
2.1 OSI Model 17
2.2 Data Link Layer 19
2.2.1 Medium Access Control 20
2.3 Transport and Network Layer 21
2.3.1 Traffic model 21
2.3.2 Routing 21
2.4 Performance Measures 22

3 Queuing Systems **25**
3.1 Fixed Priority Queuing 25
3.2 Weighted Fair Queuing 30
3.2.1 Generalized Processor Sharing 30
3.2.2 Virtual Time 32

4 Results **35**
4.1 Scenarios 35
4.2 Analytical Results 37

4.3 Simulation Results 39
4.3.1 WFQ 39
4.3.2 WFQ vs. FPQ 39

5 Conclusions 43
5.1 Future work 44

Chapter 1

Introduction

1.1 Background

Since the early eighties there have been an increasing adoption of information technology in military affairs. This innovative development of warfare is often referred to as *revolution in military affairs* (RMA). As a part of this ongoing RMA the Swedish Armed Forces are beginning to adopt and develop the concept of *network based defense* (NBD) [1]. Previously there was a platform-centric view of warfare. The focus was on abilities and performance bound to individual platforms such as a fighter plane or a warship. In NBD the abilities of a platform are seen as a set of services that the platform can provide. By interconnecting several platforms into a network structure the collective ability will increase. One requirement for achieving NBD is the availability of high capacity networks that can distribute information between all entities in the network.

Several different types of services will be provided through the network including group calls, positional services, and situation awareness. All services place specific demands on packet delays and packet losses in order to be fully functional. For example, as the human ear is very sensitive to delays, voice transmission demands low delays. File transfer and e-mail, on the other hand, have a much higher delay tolerance. Those types of demands are commonly referred to as *quality-of-service* (QoS) demands.

In traditional networks all traffic receives the same treatment, which leads to a poor utilization of available resources. Consider a network that provides a set of services with different QoS demands, for example voice transmission, with its high demand on low delays, and e-mail, with considerably better delay tolerance. If the network treats all traffic equally, all services will receive the same QoS, even though they have very different QoS demands. When the network has enough capacity to give all services the QoS demanded for voice transmission, everything is fine. Yet it is a serious waste of resources to give that kind of QoS to e-mail. However, if the network load increases to a level where the network no longer can provide such high QoS, it will lead to increased packet delays and packet losses. The quality of the voice transmissions will start to degrade and finally be inoperable. E-mail, on the other hand, might still work or even be given far better QoS than it requires. If the network instead could provide the e-mail service with the minimum capacity that it needs to fulfill its QoS requirements, there might be enough capacity left for the voice transmissions to function under this higher network load.

Much research is being done on the provision of QoS guarantees on the Internet, such as the *differentiated services* (DiffServ) architecture [2], a proposed standard from the *internet engineering task force* (IETF) for service differentiation on the Internet. In DiffServ individual traffic flows with similar QoS demands are tagged as members of the same service class, and the service classes are given differentiated treatment in the form of different *per-hop* behavior, i.e. traffic from different service classes is given different forwarding treatment when relayed through the network. One of the suggested per-hop behaviors is *assured forwarding*, where service classes are guaranteed a minimum forwarding rate and are also guaranteed a minimum buffer capacity. Another suggested per-hop behavior is *relative service*, where the network simply guarantees that higher classes will be provided with better QoS than lower classes [3].

The nature of military operations places greater demands than high capacity on such a network: it must also be robust and mobile. Fixed infrastructures are very vulnerable because of their centralized structure. Therefore they cannot be relied on solely. Instead we need a mobile network that can be established quickly in any environment. To further increase robustness the network should have a decentralized structure because if a node becomes inoperable it will have the least impact on the total network performance if the network is decentralized. One technology that realizes a network that might fulfill those demands is mobile ad hoc networks.

The phrase *ad hoc* is Latin and literally means *for this* or *for this purpose only* and is often used to denote temporary solutions. The term mobile ad hoc networks refers to wireless networks that are created dynamically through cooperation between the participating wireless nodes, i.e. without the aid of a central administrative node or fixed infrastructure [4].

The lack of a fixed infrastructure in the network together with its dynamic and mobile property can cause the network topology to change rapidly and in an unpredictable manner. To establish node-to-node communication over a large area under such harsh conditions, each node must also function as a router for the network to provide multihop functionality. In that way traffic between two nodes that do not have a direct connection between each other can take multiple hops over other nodes in the network to reach its destination. However, this creates a new problem, namely how the packets know what route to take through the network. This is solved with a routing protocol that determines how traffic is routed through the network, see [5] for an overview of routing protocols for ad hoc networks. Another important problem in mobile ad hoc networks is how to control the node's access to the transport medium; the radio channel in the case of wireless transmissions. This is known as the *medium access control* (MAC) problem.

1.2 Problem overview

1.2.1 Medium Access Control

The *medium access control* (MAC) protocol is the set of agreed rules used to prevent or resolve conflicts that occur when more than one node tries to use the channel at the same time. MAC protocols can be classified into two classes: contention-based and conflict-free protocols [6].

Contention-based protocols Contention-based protocols do not guarantee that a transmission is successful. Instead they describe a set of rules used to resolve conflicts when they occur. The pure ALOHA is an example of a very simple contention-based protocol that solves conflicts by random retransmission. That is, whenever a transmission fails due to collision the node waits for a random amount of time and then tries again. This process is repeated until the transmission is successful. It is obvious that under high loads a lot of network capacity is wasted on solving conflicts. To overcome those problems several modifications of the pure ALOHA algorithm, which enhances the performance under high loads, have been suggested, e.g. slotted ALOHA, where the time is divided into time slots and a transmission is only allowed to start at the beginning of a time slot.

Another more elaborate extension of the ALOHA scheme is the *carrier sense multiple access* (CSMA) protocol. The fundamental idea behind CSMA is to sense the channel before each transmission and only start the transmission if the channel is idle. If the channel is busy, the node waits for a random amount of time and then repeats the same procedure. This scheme clearly prevents some conflicts but not all. For example, two nodes could sense the channel at the same time and find it idle and then start to transmit, with a collision as result. When conflicts occur, they are solved in the same way as in the ALOHA protocol.

The problem with contention-based protocols is that collisions will occur, and under high network loads an increasing amount of capacity is wasted on resolving conflicts. This makes it difficult to make any QoS guarantees, especially delay bounds.

Conflict-free protocols Conflict-free protocols are designed to avoid conflicts, i.e. all transmissions are guaranteed to be successful, at least in the sense that a transmission will not fail due to interference from other nodes in the network. In radio networks where the channel is the radio channel, the conflict-free property can be obtained by frequency multiplexing, time multiplexing or a combination of both.

Time Division Multiple Access The *time division multiple access* (TDMA) protocol is a conflict-free MAC protocol that is based on

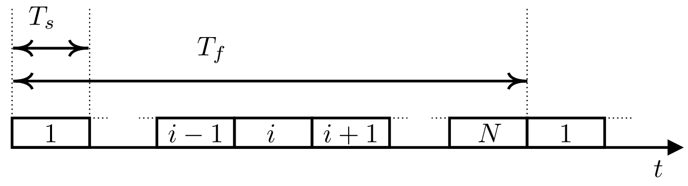


Figure 1.1: TDMA slot allocation for a N-node network with the *frame length* T_f and the *slot length* T_s.

time multiplexing. The time is divided into several *time slots*, and each node is assigned one of those slots according to a periodically repeating pattern as shown in Fig. 1.1. One such period is refered to as a *frame* or *cycle*. During the assigned time slot, that node is the only one that is allowed to use the channel. To uphold this rule the nodes must be synchronized. The TDMA protocol provides a reasonable utilization of the channel when the nodes are geographically close and have similar communication requirements, but if the nodes are geographically scattered or if there are nodes that have greater communication requirements than the others, the utilization starts to degenerate.

There exists several modifications to the basic TDMA protocol that deals with its deficiencies, for example *generalized TDMA* (GTDMA), where nodes with higher load may be assigned more than one time slot. Another TDMA derivative is *spatial-reuse TDMA* (STDMA) [7]. In STDMA time slots can be reused, i.e. two nodes that are sufficiently spatially separated for their transmissions not to interfere with each other can be assigned the same time slot. This provides a better utilization of the channel when the nodes are geographically scattered.

1.2.2 Queuing systems

When a node generates traffic, or receives relay traffic, at a higher rate than it can transmit, a *queue* is formed. The most common way to deal with those queues are the *first-come-first-serve* (FCFS) discipline, where the packets are served in the order of their arrival.

However, there exists a wide variety of different queuing disciplines, such as the equally simple *last-come-first-serve* (LCFS) discipline and the random service discipline. An entire family of queuing disciplines are the priority queues, where the packets are given differentiated treatment according to which service class they belong to. In priority queues the packets are assigned a priority that is a function of service class, and then the packets are served in decreasing order of priority.

Fixed Priority Queuing

The *fixed priority queuing* (FPQ) discipline is probably the simplest form of priority queues. As the name suggests, the packets are assigned a fixed priority according to service class membership, i.e. if the packet belongs to service class c, it is assigned the priority q_c.

There is no sense of fairness in this strategy since packets that belong to the service class with the highest priority are always served first. So packets that do not belong to that service class are not guaranteed any service at all; they are merely given what is left after the highest priority class has been served.

Weighted Fair Queuing

Weighted fair queuing (WFQ) was first introduced in [8]. It was also developed in parallel under the name *packet-by-packet generalized-processor sharing* (PGPS) in [9], and is a packet approximation of the *generalized-processor sharing scheme* (GPS) [9]. GPS allows a minimal percentage of the total capacity to be allocated to a service class and uses proportionally fair sharing of any excess capacity. In other words every service class is guaranteed a minimum service rate, and any excess capacity is shared fairly between active service classes.

1.3 Problem definition

The major contribution to packet delays in multihop radio networks comes from the queuing time in the individual nodes when the packets are routed through the network. The same holds for packet losses. They arise when the queue in a node grows over its buffer capacity and it starts dropping packets. Hence packet loss and delay are to a high degree a *local* or *per-hop* problem. Therefore it is a natural strategy to change the per-hop behavior in an attempt to overcome the problems.

The purpose of this work is to investigate the possibility of providing such differentiated per-hop behaviors among service classes by employing priority queues in the MAC protocol. More specifically we will study the effects of fixed priority queuing and weighted fair queuing in TDMA-networks. Since the focus is on aggregated traffic flows (service classes) and not individual traffic flows, we will not be able to give absolute QoS guarantees, but rather a relative service differentiation.

1.4 Outline

In chapter 2 we build up the network model, with its assumption and simplifications, that we will use in this report. Here, we also define the performance measures that we will use. In chapter 3 we give a more in-depth description of our two queuing schemes: fixed priority queuing and weighted fair queuing. In this chapter we also derive an analytical approximation for the network delay in a TDMA network with fixed priority queues. In chapter 4 we present the results from the simulations and the analytical approximation. Finally we draw our conclusions in chapter 5, where we also present possible topics for future work.

Chapter 2

Network Model

In this chapter we provide a layout of the network model with the assumptions and simplifications that we use throughout this work. We start by introducing the OSI reference model, which is used to describe the network architecture, in section 2.1. In subsequent sections we describe the relevant layers of our network model.

2.1 OSI Model

The *open systems interconnection* (OSI) model is a reference model for network architectures developed by the *international organization for standardization* (ISO) in the late 1970s [10]. The OSI model breaks down the network functionality into a hierarchy of seven layers, as shown in Fig. 2.1.

Each layer provides a specified set of functions to higher layers by encapsulating the next lower layer and adding functionality to it. In this way the next higher layer is provided with a virtual communication link with a specified set of properties.

The seven layers in the OSI model in ascending order are:

Physical Provides a virtual link for bits, i.e. it handles the transmission of raw bits over the communication medium.

Data link Provides a virtual link for reliable transmission and reception

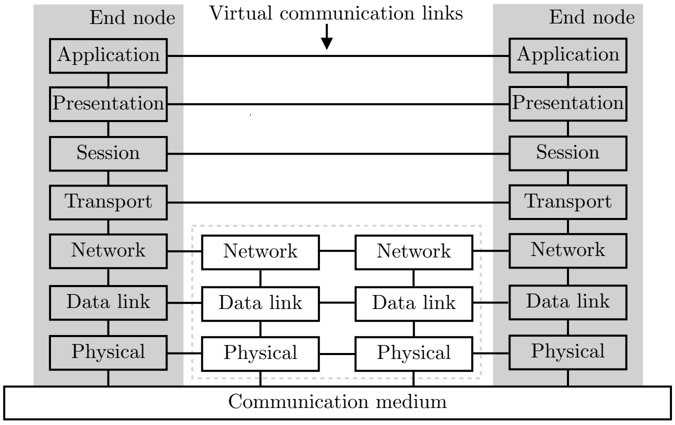


Figure 2.1: The seven layers in the OSI model. The lower three layers must be implemented in all nodes in the network, whereas the upper four layers are needed only in end nodes. Each layer provides the next higher layer with a virtual communication link with a specified set of properties.

of packets, i.e. it handles error correction and MAC.

Network Provides a virtual link for end-to-end packets, i.e. it handles the routing of packets in the network.

Transport Provides a virtual link for end-to-end messages, i.e. in the source node it breaks down messages into packets, and at the destination node it reconstructs the messages from the packets.

Session Provides a virtual link for the information exchange that is necessary to establish a session.

Presentation Provides a virtual link that is independent of data representation, i.e. it handles encryption, for example.

Application Provides a virtual link for applications to interact with each other, for example, the *file transfer protocol* (FTP) is used by file transfer applications.

The three lowest layers must be implemented in all intermediate nodes such as routers and switches. In contrast the four upper layers are needed only in end nodes. However, in an ad hoc network all nodes can act as an intermediate node, since they can all function as a router. Therefore all the seven layers are needed in every node.

2.2 Data Link Layer

We let the directed graph $G = (\mathcal{V}, \mathcal{E})$ represent the network, where $\mathcal{V}$ is the set of *vertices* and $\mathcal{E}$ is the set of directed *edges*. The vertices represent the nodes in the network, and the edges represent the links between nodes. In order to specify $\mathcal{E}$ we need some definitions.

We define P_i as the transmission power from node i, i.e. the signal power that the transmitter antenna in node i is fed with. G_{ij} is defined as the link gain from node i to node j, i.e. the power gain of the signal as it passes the transmitter antenna, the radio channel and the receiver antenna. Thus, the received signal power in node j when node i transmits is given by $P_i G_{ij}$. Further, we define N_j as the noise power in node j, i.e. the noise power that is received in node j. With the above definitions we can define the *signal-to-noise ratio* (SNR), Γ_{ij}, in node j when node i transmits. As the name suggests Γ_{ij} is the quotient between the received signal power, $P_i G_{ij}$, and the received noise power, N_j.

$$\Gamma_{ij} = \frac{P_i G_{ij}}{N_j} \tag{2.1}$$

If Γ_{ij} is sufficiently large, then node i can transmit reliable, i.e. error free, to node j. In this case we say that there exists a link (i, j) from node i to node j. Let γ_R denote this *reliable communications threshold*. Then the set of links $\mathcal{E}$ (or set of directed edges) is given by:

$$\mathcal{E} = \{(i, j) : \Gamma_{ij} \geq \gamma_R, \quad \forall (i, j) \in \mathcal{V} \times \mathcal{V}\} \tag{2.2}$$

The absolute level of γ_R depends on several of the radio system's properties, such as the modulation, data rate, and the channel coding.

Note that $(i, j) \in \mathcal{E}$ does not necessarily imply that $(j, i) \in \mathcal{E}$, but if we make the following assumptions:

- All nodes transmit with equal power.
- The channel is reciprocal and all nodes use isotropic antennas. An isotropical antenna is a theoretical antenna that radiates equally well in all directions. This assumption gives us $G_{ij} = G_{ji},\ \forall i, j \in \mathcal{V}$.
- The noise power in all nodes is equal, i.e. $N_i = N_j,\ \forall i, j \in \mathcal{V}$.
- All links operate at the same fixed data rate.

Then it will hold that $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$.

2.2.1 Medium Access Control

For simplicity we will use the basic time division multiple access (TDMA) protocol in the analytical analysis of fixed priority queues (see section 3.1). The basic TDMA protocol has a poor channel utilization when there are nodes with prominent greater communication needs than most of the other nodes. This is a situation that often arises when the nodes are geographically scattered, since this gives rise to *bottlenecks* when a major part of the traffic is routed over a few nodes. For that reason, we will use the *generalized time division multiple access* (GTDMA) protocol with perfect traffic adaptation in the rest of our work.

In a GTDMA network, nodes with greater communication needs can be assigned more time slots than other nodes and thus increase their capacity. Perfect traffic adaptation means that the nodes are assigned time slots corresponding to the average traffic load that they are exposed to. Further, to minimize the network delay the time slots for each node should be evenly spaced in the GTDMA frame. That, however, is a tough problem. To circumvent this, we will permute the slot allocation at the start of each new frame. In that way we obtain the evenly spaced property on average over time.

The only deviation we make from the standard (G)TDMA protocols is that we will not use the common first-come-first-serve queuing discipline. Instead we will use the queuing disciplines described in chapter 3.

For simplicity we make the following assumptions on the (G)TDMA protocol:

- Perfect slot synchronization, i.e. every node has access to a perfectly synchronized time reference.
- All packets are of equal length and it takes a whole time slot to transmit a packet.

2.3 Transport and Network Layer

2.3.1 Traffic model

Traffic that arrives at a network can be categorized in two categories: traffic with a single source and destination (*unicast*) and traffic that has multiple destinations (*multicast*). Unicast is obviously used for point-to-point communication such as file transfers, telephone calls or e-mail, whereas multicast is used when the information needs to be distributed to multiple nodes, which is the case with group calls and situation awareness services. Even though multicast is an important traffic type we will focus on unicast traffic in this work because it is easier to analyze analytically.

Unicast traffic can be modeled as a stream of packets where each packet enters the network at a *source node* $i \in \mathcal{V}$ according to a probability function $p_s(i)$ and leaves the network at a *destination node* $j \in \mathcal{V}$. The destination node for a packet can be modeled as a conditional probability, i.e. given that the source node is i, the probability that the destination node is j is $p_d(j|i)$.

We will use a uniform traffic model where the packets from service class c arrive at the network according to a Poisson process with intensity λ_N^c. That the traffic model is uniform means that each node is equally probable as source node, hence $p_s(i) = 1/N$, and that each node except the source node is equally probable as destination node, and hence $p_d(j) = 1/(N-1)$, where $N = |\mathcal{V}|$ is the number of nodes in the network.

2.3.2 Routing

Since the network should provide multihop functionality, each node also functions as a router and therefore must have a routing table. For routing we will use the shortest-path algorithm, i.e. a packet will be routed along the route that traverses the least number of nodes. This algorithm minimizes the channel utilization, i.e. it requires the least number of retransmissions of a packet for it to reach its destination. If more than one shortest route exists between two nodes, then all traffic between those two nodes always uses the same route. The routing table can be calculated with Dijkstra's algorithm [11], for example. Denote this routing table R where the table entry $R(k, l)$ is a route r_{kl} from node k to node l. We will assume that the graph $(\mathcal{V}, \mathcal{E})$ forms a connected graph, i.e. there exists a route between every node pair. Hence the number of routes in the network is given by $|R| = N(N-1)$.

With the routing table given, we can calculate the average traffic load λ_i^c from service class c on node i. First define Λ_{ij} as the number of paths that contain the directed link (i, j). Since there is a total of $N(N-1)$ routes in the network the quotient $\Lambda_{ij}/N(N-1)$ represents the relative load on link (i, j). With this we can write the average traffic load λ_{ij}^c from service class c on link (i, j) as

$$\lambda_{ij}^c = \lambda_N^c \frac{\Lambda_{ij}}{N(N-1)}, \tag{2.3}$$

where λ_N^c is the average traffic load on the network from traffic class c. We note that Eq. (2.3) is only valid when all traffic from service class c is forwarded through the network. If packets from service class c are dropped then Eq. (2.3) will give an overestimation of the link load.

Now, summing over all nodes that node i has a link to, we get

$$\begin{aligned} \lambda_i^c &= \lambda_N^c \sum_{j:(i,j)\in\mathcal{E}} \frac{\Lambda_{ij}}{N(N-1)} \\ &= \lambda_N^c \frac{\Lambda_i}{N(N-1)}, \end{aligned} \tag{2.4}$$

where Λ_i is the number of routes in R that start in or pass node i.

2.4 Performance Measures

Since we are interested in QoS from a delay perspective, we will use the *end-to-end* packet delay as a performance measure for the different service classes. More specifically we will look at the *network delay*. We define the network delay as the expected value of the average end-to-end packet delay over all routes. We let the stochastic variable d_i^c denote the node delay, i.e. the delay that a packet from service class c experiences when it passes node i. Further, let D_{kl}^c denote the end-to-end packet delay for route r_{kl}. D_{kl} can then be written as the sum of all node delays along the route r_{kl}. We get

$$D_{kl}^c = \sum_{i:(i,j)\in r_{kl}} d_i^c. \tag{2.5}$$

Since there are N possible start nodes for a route and for each start node there are $N-1$ possible end nodes, there is a total of $N(N-1)$ different routes in the network. With that we get the average end-to-end packet delay over all routes as

$$D^c = \frac{1}{N(N-1)} \sum_{k\in\mathcal{V}} \sum_{l\in\mathcal{V}\setminus k} D_{kl}^c. \tag{2.6}$$

The network delay for service class c is the expected value of Eq. (2.6)

$$\overline{D}^c = E[D^c]. \tag{2.7}$$

Chapter 3

Queuing Systems

In this chapter we give a more detailed description of the two queuing systems that we use. We start with fixed priority queuing in section 3.1, where we also derive an analytical expression for the network delay. Then we move on to weighted fair queuing in section 3.2 and its implementation details.

3.1 Fixed Priority Queuing

In fixed priority queuing the packets are assigned a fixed priority according to which service class they belong to. Packets from service class 1 are assigned the highest priority, packets from service class 2 are assigned the next highest priority, etc. The packets are then transmitted in decreasing order of priority as shown in Fig. 3.1.

The merging of packet streams that takes place in a multihop network when packets are relayed through the network complicates the properties of the arrival processes at the nodes in the network. The problem is that the merging can create a strong correlation between the packets' inter-arrival times and the packets' transmission time. A common approximation when packets arrive according to a Poisson process and have a exponentially distributed transmission time is the Kleinrock independence assumption [12]. It simply states that the relay traffic is of Poisson type and independent of transmission time. In a TDMA network

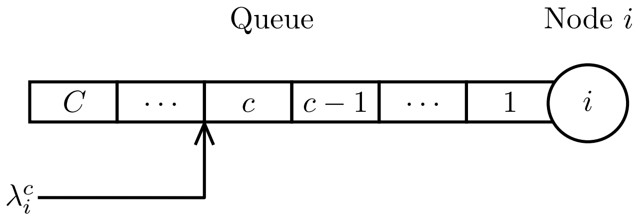


Figure 3.1: Packets from service class c join a fixed priority queue behind all packets from service classes that have a higher priority and before all packets from service classes with lower priority.

the transmission times are clearly not exponentially distributed. Therefore there might be second thoughts about applying the assumption to TDMA network. However, in [13] Grönkvist show that the Kleinrock independence assumption is a fairly good approximation when applied to STDMA networks. Further, since the objections to applying Kleinrock's independence assumption in a TDMA network are essentially the same as in a STDMA network we will use the assumption in the following.

This assumption enables us to treat the queue in node i independently of all the other queues and as if the arrival process for the different service classes is of Poisson type with intensity λ_i^c given by Eq. (2.4).

To calculate the expected delay, $E[d_i^c]$, that a packet from service class c experiences when it passes node i in the network we will use the general scheme for calculating delays in priority queues that is presented in [12]. We study the system from the point of view of a newly arrived packet from service class c and denote this packet the tagged packet.

The expected delay that the tagged packet experiences when it passes node i in the network can be broken down in two parts: the expected waiting time in the queue and the transmission time. Since it is a TDMA network and all packets are of equal length, and take a whole time slot to transmit, the transmission time is deterministically given by the *slot length* T_s. Let $\overline{W}_i^c$ denote the expected waiting time in the queue for node i for packets from service class c. $\overline{W}_i^c$ can be broken down into three parts:

1. The expected synchronization time $\overline{T}_{sync}$, i.e. the expected time to

the next allocated time slot for node i.

2. The expected waiting time due to previously arrived higher, or equal, priority packets $\overline{T}_{php,i}$, i.e. packets that already are in the queue and that have a higher or equal priority and thus will be transmitted before the tagged packet.

3. The waiting time due to successive arrivals of higher priority packets $\overline{T}_{shp,i}$, i.e. packets that arrive while the tagged packet are in the queue and that have a higher priority and thus will be transmitted before the tagged packet.

The expected synchronization time $\overline{T}_{sync}$ depends on whether node i is the source node for the packet or whether the packet is relayed via node i. If node i is the source node for the packet, then the packet arrival time is uniformly distributed over the frame T_f. Hence $\overline{T}_{sync} = T_f/2$.

When the packet is relayed via node i, we note that packets cannot arrive in node i's time slot, because according to the TDMA protocol node i is the only node that is allowed to transmit during this time slot. Further, packets can arrive only at the end of time slots, because a packet transmission takes exactly one time slot and starts at the beginning of a time slot. This is shown in Fig. 3.2, where time slots that relay traffic to node i can arrive in are shaded gray. The time between the start of two adjacent time slots allocated to node i is T_f. Because packets cannot arrive in node i's time slot nor in node $i+1$'s time slot, we see that the synchronization time for relay traffic is in the interval $[0, T_f - 2T_s]$. Here we assume that the synchronization time for relay traffic is uniformly distributed in that interval, and we get $\overline{T}_{sync} = (T_f - 2T_s)/2$. To determine the expected synchronization time for both types of traffic, we must know how much of the total traffic that originates from the node itself and how much is relay traffic.

There is a total of Λ_i routes that pass node i, and $N-1$ of them have node i as start node. Hence, $(N-1)/\Lambda_i$ of the traffic that passes node i originates from the node itself, and the rest of the traffic, $1-(N-1)/\Lambda_i$, must be relay traffic. Thus the expected synchronization time for the combined traffic is given by

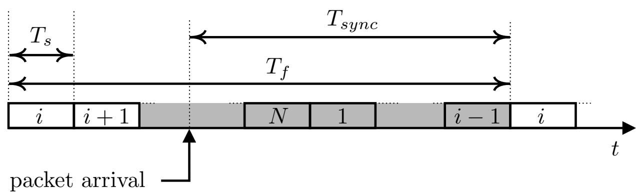


Figure 3.2: Slot allocation for a N-node TDMA network. Traffic that is relayed via node i can only arrive in the shaded area.

$$\overline{T}_{sync} = \frac{N-1}{\Lambda_i} \cdot \frac{T_f}{2} + \left(1 - \frac{N-1}{\Lambda_i}\right) \cdot \frac{T_f - 2T_s}{2}. \tag{3.1}$$

For the second part of the waiting time we will need *Little's result* [12], which relates the expected number of packets in the queue, $\overline{M}$, to the expected arrival rate, λ, and the expected time spent in queue, $\overline{W}$, as

$$\overline{M} = \lambda \overline{W}. \tag{3.2}$$

This expresses the intuitive feeling that a system with a long queue is associated with long delays and high arrival rates.

Each packet transmitted before the tagged packet contributes a delay of T_f. So every service class ξ with a priority that is the same as or higher than our tagged packet contributes a delay of

$$T_f \overline{M}_i^{\xi}, \tag{3.3}$$

where $\overline{M}_i^{\xi}$ is the expected number of packets from service class ξ in node i's queue. Summing Eq. (3.3) over all service classes that have a priority higher than or equal to the tagged packet and together with Eq. (3.2), we get the second part of the waiting time:

$$\overline{T}_{php,i} = T_f \sum_{\xi=1}^{c} \lambda_i^{\xi} \overline{W}_i^{\xi}. \tag{3.4}$$

The tagged packet spends an average of $\overline{W}_i^c$ in the queue because the queue size and the arrival process are independent. On average, $\lambda_i^{\xi} \overline{W}_i^c$

packets arrive from service class ξ during that time. All packets with a higher priority than the tagged packet that arrive during that time, introduce a delay of T_f. Thus, for the third part of the waiting time we get

$$\overline{T}_{shp,i} = T_f \sum_{\xi=1}^{c-1} \lambda_i^\xi \overline{W}_i^c. \tag{3.5}$$

Finally summing Eqs. (3.1), (3.4) and (3.5), we get the total waiting time:

$$\overline{W}_i^c = \overline{T}_{sync} + T_f \sum_{\xi=1}^{c} \lambda_i^\xi \overline{W}_i^\xi + T_f \sum_{\xi=1}^{c-1} \lambda_i^\xi \overline{W}_i^c. \tag{3.6}$$

Solving for $\overline{W}_i^c$ we get the following set of recursive equations:

$$\overline{W}_i^c = \frac{\overline{T}_{sync} + T_f \sum\limits_{\xi=1}^{c-1} \lambda_i^\xi \overline{W}_i^\xi}{1 - T_f \sum\limits_{\xi=1}^{c} \lambda_i^\xi}.$$

Here, we observe that this is a triangular set of equations. We can easily solve for $\overline{W}_i^1$ and obtain the following solution

$$\overline{W}_i^1 = \frac{\overline{T}_{sync}}{1 - T_f \lambda_i^1}.$$

Then we can solve for $\overline{W}_i^2, \ldots, \overline{W}_i^C$ recursively. The general solution is given by

$$\overline{W}_i^c = \frac{\overline{T}_{sync}}{(1 - T_f \sum\limits_{\xi=1}^{c} \lambda_i^\xi)(1 - T_f \sum\limits_{\xi=1}^{c-1} \lambda_i^\xi)}. \tag{3.7}$$

With Eq. (3.7) we get the expected node delay as

$$E[d_i^c] = \frac{\overline{T}_{sync}}{(1 - T_f \sum\limits_{\xi=1}^{c} \lambda_i^\xi)(1 - T_f \sum\limits_{\xi=1}^{c-1} \lambda_i^\xi)} + T_s. \tag{3.8}$$

With Eqs. (3.8) and (2.4) in Eq. (2.6) we now have an analytical expression for the network delay $\overline{D}^c$ in a TDMA network with fixed priority queues. In chapter 4 we will evaluate how good this approximation is by comparing it with simulation results.

3.2 Weighted Fair Queuing

Weighted fair queuing (WFQ) is packet approximation of the *generalized-processor sharing scheme* (GPS). It was developed in parallel in [8] and, under the name *packet-by-packet generalized processor sharing* (PGPS), in [9]. The GPS queuing scheme has a very attractive property. One can allocate a specific percentage of the total system capacity to a service class. Further, if some service classes do not utilize their full share, the excess capacity is fairly shared between those classes that need it. Thus, every service class is guaranteed a minimum service rate, but they may experience a better service rate if the system is not fully utilized.

3.2.1 Generalized Processor Sharing

GPS is a flow-based scheme that serves multiple service classes synchronously at a fixed rate r. If we associate each service class c with a positive real number ϕ_c and let $S_c(\tau, t)$ denote the amount of service that service class c received during the time $]\tau, t]$, then the GPS server is defined as a server that satisfies

$$\frac{S_c(\tau,t)}{S_j(\tau,t)} \geq \frac{\phi_c}{\phi_j}, \quad j = 1, 2, \ldots, C \tag{3.9}$$

for each service class c that is continuously backlogged during $]\tau, t]$. Thus, the actual amount of service that service class c receives relative to service class j is always greater than or equal to the quotient ϕ_c/ϕ_j. Multiplying with $S_j(\tau, t)$ and ϕ_j on both sides in Eq. (3.9) and summing over all js, we get

$$S_c(\tau,t) \sum_{j=1}^{C} \phi_j \geq \phi_c \sum_{j=1}^{C} S_j(\tau,t).$$

Here, we note that $\sum_{j=1}^{C} S_j$ is the total service given by the system during the interval $]\tau, t]$. Since the system is continuously working during that period that sum must be equal to the system's capacity integrated over that interval, which is $r(t-\tau)$. With that, and by dividing by $t-\tau$ and $\sum_{j=1}^{C} \phi_j$ on both sides, we get

$$\frac{S_c(\tau, t)}{t-\tau} \geq \frac{\phi_c}{\sum_{j=1}^{C} \phi_j} r.$$

If we let $\tau \to t$, we see that the service rate $r_c(t)$ for service class c has a lower bound given by

$$r_c(t) = \lim_{\tau \to t} \frac{S_c(\tau, t)}{t-\tau} \geq \frac{\phi_c}{\sum_{j=1}^{C} \phi_j} r = g_c.$$

Hence, service class c is guaranteed the minimum service rate g_c independently of the amount of load the server experiences from other service classes. That minimum service rate can be adjusted, with the parameter ϕ_c, to give favorable treatment to certain service classes. If $\phi_i = \phi_j, \ \forall i, j$ the scheme degenerates to uniform processor sharing and the service classes are given their fair share, an equally big part, of the systems capacity. As stated above, the GPS scheme is flow-based and is therefore not suitable for systems where the smallest entity is packets, but it can work as a foundation on which to build a packet-based queue.

The most straightforward way to make a packet approximation of GPS is a work-conserving server that serves packets in the order that they would have finished if they were served by a GPS server. In other words, if we let F_p denote the finishing time of packet p under GPS, then the packetized server should serve the packets in increasing order of F_p. Unfortunately that is not possible. Consider the case when a server that has queued traffic completes the service of one packet at time τ and is ready to serve the next packet. It should pick the packet with the lowest F_p, but that packet may not have arrived and the server does not know if or when a packet with a lower F_p will arrive. To serve packets in strictly increasing order of F_p it should pick the packet with the lowest F_p in the queue and if during that packet's service time, a packet with a lower F_p arrives, the packet in service gets pushed back to the queue, and the server starts serving the newly arrived packet instead. This scheme is clearly not work-conserving and is the preemptive version of WFQ. In this work we will use non-preemptive WFQ, which is work-conserving and serves the packets in increasing order of F_p under the assumption that no packets will arrive after time τ.

3.2.2 Virtual Time

To implement weighted fair queuing we need to keep track of the finishing times, F_p, the packets would have if they were served by a GPS server. This can be done with a virtual time that simulates the progress of time in the fictitious GPS server. We will use the virtual time implementation from [9].

The virtual time $V(t)$ is a function of real time t and is defined as zero when the server is idle. When the server becomes busy, the virtual time starts to progress and the rate of change of V is updated on every event that occurs in the system. By event we mean a packet arrival or departure. Let t_j denote the real time when event j occurs in a busy period. The event counter j is also set to zero when the server is idle. Further, let B_j denote the set of busy service classes, i.e. classes that have packets in the queue or in service during the open interval $]t_{j-1}, t_j[$. Since there are no arrivals or departures in that interval, the set, B_j, must be fixed in that interval. In a busy period the virtual time is then defined to progress as

$$\begin{aligned} V(0) &= 0 \\ V(t_{j-1} + \tau) &= V(t_{j-1}) + \frac{\tau}{\sum_{i \in B_j} \phi_i}, \end{aligned}$$

for $0 \leq \tau \leq t_j - t_{j-1}$ and $j = 2, 3, \ldots$, i.e. the virtual time increases at the same rate as the backlogged sessions receive service.

Now we define L_c^ξ as the real time it takes from when the transmission of packet number ξ from service class c starts until the next transmission can start, i.e. in a standard TDMA network it is equal to the frame length T_f. With that we can write the finish time for packet number ξ

from service class c that arrives at the real time a_c^ξ as

$$F_c^\xi = \max\left(F_c^{\xi-1}, V\left(a_c^\xi\right)\right) + \frac{L_c^\xi}{\phi_c}, \tag{3.10}$$

where $F_c^{\xi-1}$ is the finish time for the previous packet of the same service class and F_c^0 is defined as zero for all c. The first part of Eq. (3.10) is the arrival time plus the time spent in queue; the last part is the service time in the fictitious GPS server.

Chapter 4

Results

4.1 Scenarios

We will use three different scenarios to evaluate the performance of the queuing systems. Common to all scenarios is the fact that all service classes are modeled as uniform unicast Poisson traffic, as described in section 2.3.1. Further, they all use a test network consisting of 40 nodes and with the topology shown in Fig. 4.1.

The network was generated by placing 40 nodes randomly within a quadratic area, with the sides 1 km, in the neighborhood of Skara, Sweden. Then the link gain, G_{ij}, between nodes was calculated with Detvag-90® [14], a two dimensional deterministic wave propagation model. With the link gain known, the transmission power, P_i, was chosen to be the smallest possible value such that the graph $(\mathcal{V}, \mathcal{E})$ is a connected graph, i.e. there exists a way through the network between all node pairs.

In some of the scenarios we will use a special service class: the best effort (BE) class. Packets that belong to the BE class are not queued with the same queuing scheme as packets from other service classes. Instead they end up in their own FCFS queue. Packets in this special queue are transmitted only if the other queuing system does not have any queued packets.

Scenario I Will be used for comparing the analytical results for FPQ that we derived in section 3.1 with results from simulations. It consists of three service classes: class 1, class 2 and class 3. Each has an average arrival rate of $\lambda_N/3$ packets/time slot. Here we use the standard TDMA as MAC protocol, since the analytical expression for the network delay is derived for that.

Scenario II Will be used for analyzing the effect that the resource allocation parameters, ϕ_c, in WFQ has on the network delay. It consists of two service classes: class 1 and class 2. Each has an average arrival rate of $\lambda_N/2$ packets/time slot. To get a sufficiently low variance in the simulation results we choose a simulation length of $1.5 \cdot 10^6$ time slots. In this scenario we use GTDMA as MAC protocol.

Scenario III Will be used for comparing the performance of FPQ against WFQ. It consists of three service classes, class 1, class 2 and class BE, each has an average arrival rate of $\lambda_N/3$ packets/time slot. To get a sufficiently low variance in the simulation results we choose a simulation length of $1.5 \cdot 10^6$ time slots. In this scenario we use GTDMA as MAC protocol.

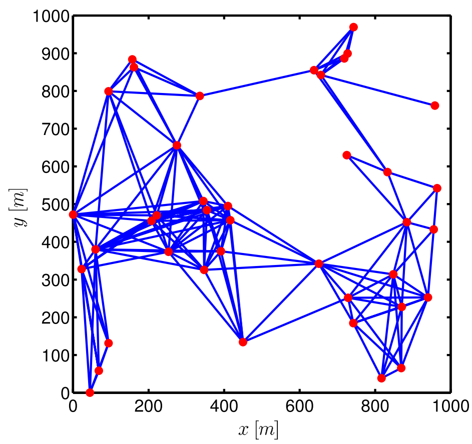


Figure 4.1: Network topology for the 40-node test network used in the simulations.

4.2 Analytical Results

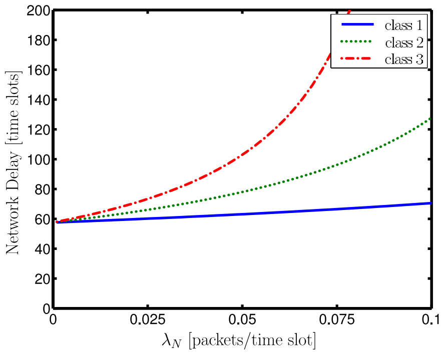
To evaluate how well the combination of Eqs. (2.4), (2.7) and (3.8) approximates the network delay in a TDMA network with FPQ we use scenario I and compare the result with computer simulations. The result is shown in Fig. 4.2(a) for the analytical expression and in Fig. 4.2(b) for the computer simulation. As we see, the approximation seems to work fairly well, especially for the high-priority class. However, we observe a rather large discrepancy for the low-priority classes. To get a better view of the discrepancy we look at the relative error, ϵ_{rel}, which we define as

$$\epsilon_{rel} = \left| \frac{\overline{D}_a^c - \overline{D}_s^c}{\overline{D}_s^c} \right|,$$

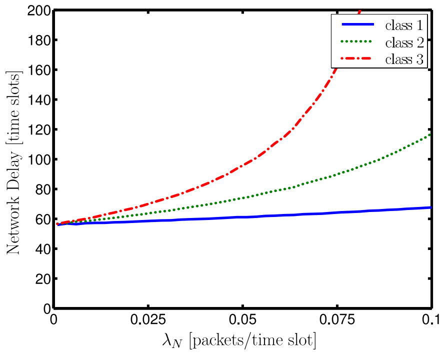
where $\overline{D}_s^c$ is the simulated value of the network delay for class c and $\overline{D}_a^c$ is the result of the analytical approximation. The relative error for scenario I is shown in Fig. 4.3.

There we see that the relative error for service class 1 starts at around 2.5% and grows slightly with increasing λ_N, whereas the relative error for classes 2 and 3 grows more rapidly with increasing λ_N. Since most nodes only have a few links to them, the process that describes the relay traffic will become more and more deterministic when λ_N increases in the simulation. In the analytical approximation, however, we assume that the relay traffic arrives according to a Poisson process. Because the Poisson process has a much higher variance than the deterministic process, the analytical approximation will overestimate the network delay, which increases with increasing λ_N.

The large discrepancy for low priority classes can be explained by the expression for the expected waiting time in queue, $\overline{W}_i^c$. As we recall from Eq. (3.6), the expected waiting time in queue i for packets from service class c, $\overline{W}_i^c$, contains a sum of the scaled expected waiting times for all service classes with a priority higher than class c. Hence, errors in the high-priority classes are accumulated and propagated to the low-priority classes.



(a) Analytical



(b) Simulations

Figure 4.2: Network delay in a TDMA network with FPQ.

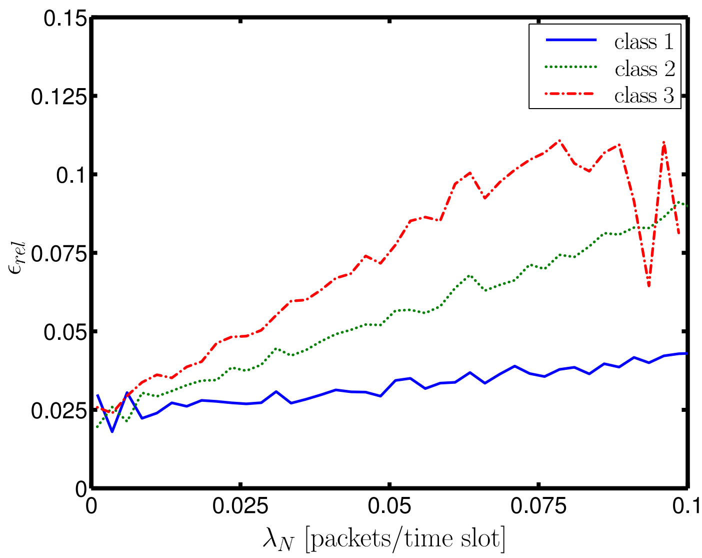


Figure 4.3: Relative error.

4.3 Simulation Results

4.3.1 WFQ

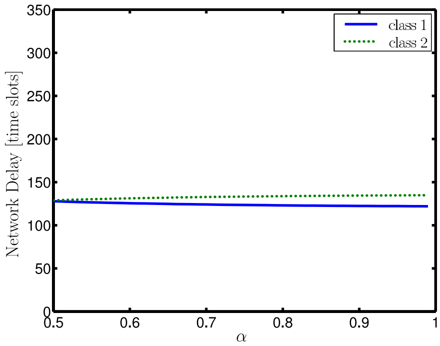
To see the effect the resource allocation parameters, ϕ_i, in WFQ have on the network delay we use scenario II. In the simulation the two service classes in the scenario are given the resource allocation $\phi_1 = \alpha$ and $\phi_2 = 1-\alpha$, respectively, and the total network load, λ_N, is fixed. The result for four different network loads is shown in Fig. 4.4, where the network delay is viewed as a function of the parameter α. There we see that the ϕ_is give us the means to control the resource allocation. As one would expect, the network delay for the two classes is equal when they have 50% each of the resources. Then as α increases, and consequently more resources are allocated to class 1, the network delay for class 1 decreases, whereas it increases for class 2. We can also see that when the network load increases and we come closer to the asymptote for class 2, the difference in network delay between the two classes increases.

4.3.2 WFQ vs. FPQ

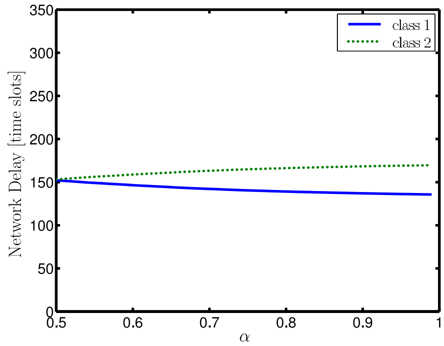
For comparison of FPQ against WFQ we use scenario III. In the simulations the resource allocation parameters in the WFQ are $\phi_1 = 0.7$ and $\phi_2 = 0.3$. The result is shown in Fig. 4.5, where the network delay is viewed as a function of the total network load, λ_N.

We see that the behavior of BE class is essentially equal for the two queuing systems. This is expected since from the BE point of view the two queuing systems work as an FPQ with 2 service classes, the low-priority BE class and the high-priority class consisting of the original class 1 and class 2. The mutual ordering between classes 1 and class 2 in the high-priority class is done with the corresponding queuing system.

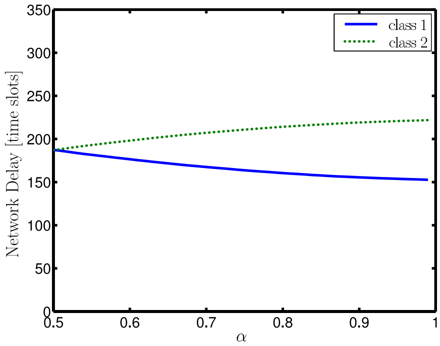
To better see the differences between class 1 and class 2 with the two queuing schemes, we look at Fig. 4.6, which is a zoomed in copy of Fig. 4.5. There we see that, in FPQ, the performance of class 2 is suppressed in favor of class 1, whereas in WFQ the resources are shared between the two service classes according to the resource allocation. Here, service class 1 is suppressed, compared with FPQ, to give service class 2 its fair share of the resources. With the resource allocation parameter, ϕ_i, we can adjust



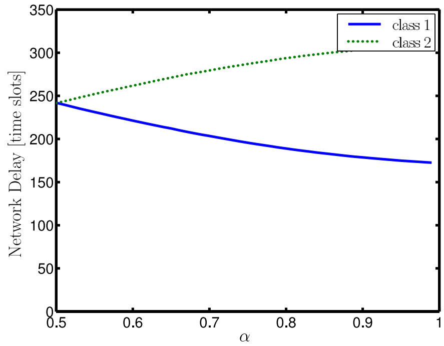
(a) $\lambda_N = 0.05$



(b) $\lambda_N = 0.10$



(c) $\lambda_N = 0.15$



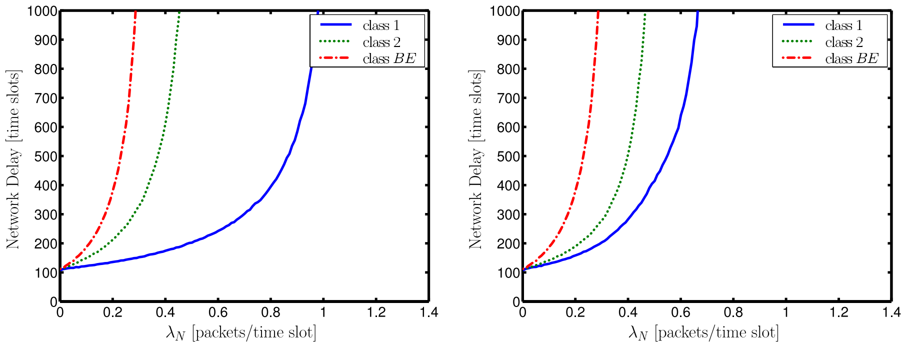
(d) $\lambda_N = 0.20$

Figure 4.4: Network delay for different network loads λ_N in a GTDMA network with WFQ as a function of the parameter α for two service classes with the resource allocation $\phi_1 = \alpha$ and $\phi_2 = 1 - \alpha$.

allocation and, in the limiting case, when $\phi_1 \to 1$ and $\phi_2 \to 0$, the WFQ will behave much like a FPQ.

Another interesting measure to look at is the throughput, which we define as the average number of packets per time slot delivered to their final destination. The throughput for the three service classes is shown in Fig. 4.7 as a percentage of total throughput in the network.

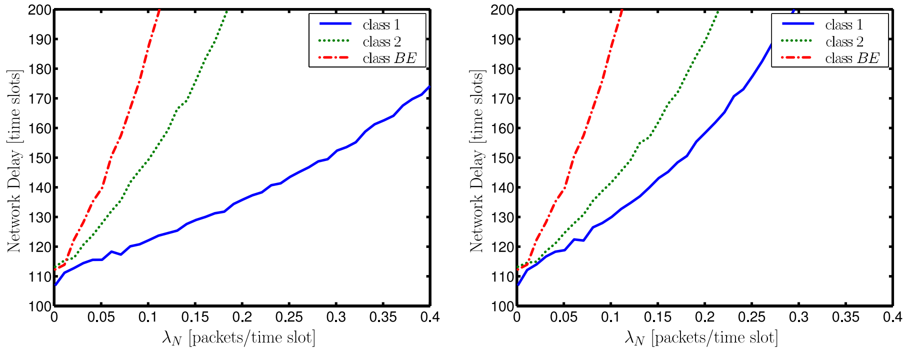
There we more clearly see how the low-priority classes are suppressed in FPQ in favor of class 1. They are even suppressed to the extent that class 1 can take all the capacity in the network, whereas in WFQ the throughput for the two prioritized classes levels out on their specific resource allocation, which in this case is 70% for class 1 and 30% for class 2.



(a) FPQ

(b) WFQ with $\phi_1 = 0.7$ and $\phi_2 = 0.3$

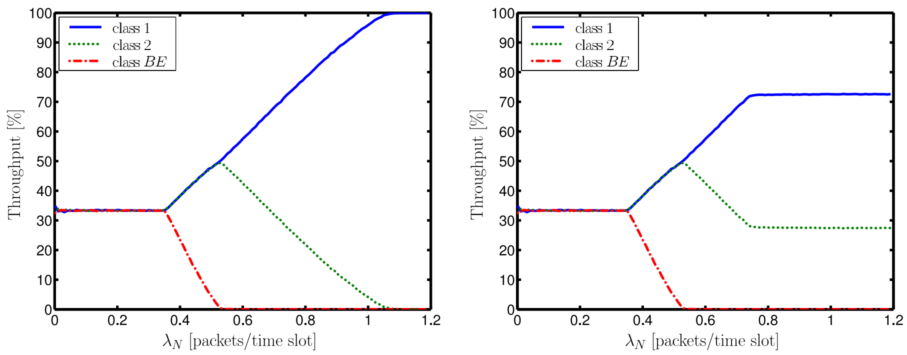
Figure 4.5: Network delay in a generalized TDMA network with FPQ 4.5(a) and WFQ 4.5(b).



(a) FPQ

(b) WFQ with $\phi_1 = 0.7$ and $\phi_2 = 0.3$

Figure 4.6: Network delay in a generalized TDMA network with FPQ 4.6(a) and WFQ 4.6(b).



(a) FPQ

(b) WFQ with $\phi_1 = 0.7$ and $\phi_2 = 0.3$

Figure 4.7: Throughput for the three service classes as percent of total throughput in a generalized TDMA network with FPQ 4.7(a) and WFQ 4.7(b).

Chapter 5

Conclusions

In this report we have examined the possibility of providing a QoS mechanism in ad hoc networks by using priority queues in the MAC layer. More specifically we have studied the problem of providing QoS by the use of *fixed priority queuing* (FPQ) and *weighted fair queuing* (WFQ) in TDMA networks. The results have mainly been obtained by simulations, but for FPQ we have also derived an analytical approximation for the network delay.

Our simulations show that the analytical approximation of the network delay for fixed priority queuing works fairly well for low traffic loads. However, the assumption that the relay traffic can be described as a Poisson process probably leads to an overestimation of the network delay, which increases with the traffic arrival intensity, λ_N. Therefore the error in the approximation increases with increasing λ_N. For moderate traffic loads it still works fairly well for the high-priority class, but for low-priority classes the error grows more rapidly because the errors are accumulated and propagate from classes with higher priority.

The evaluation of fixed priority queuing shows that it gives a very distinct delay differentiation, i.e. there is a very distinct difference in network delay between high-priority classes and classes with lower priority. The high-priority class can in fact dominate so much that no other traffic can pass through the network. This is because in fixed priority queuing, high-priority classes always take precedence over low-priority classes. Is this a desirable property? It certainly has its applications in a military context where, for example, *priority messages*[1] should always take precedence over all other traffic. However, it might not be the best way to differentiate between traffic that has different priorities for technical reasons because in this case the priorities do not indicate the importance of the traffic and therefore it is no longer obvious that the prioritized traffic should always take precedence over other traffic.

Weighted fair queuing, on the other hand, provides the means to control how much of the resources that are dedicated to a specific service class. Consequently no service class can totally dominate the network. One interesting property with WFQ is that if it is combined with an *admission control policy* that controls how much traffic that allowed to enter the network, we can give absolute end-to-end guarantees if the arrival process fulfills certain constraints [9]. This might be better suited to giving QoS for technical reasons.

We conclude that both of the evaluated queuing schemes have their advantages and disadvantages, and none of them alone is likely to be the answer to providing QoS. Instead, a combination of them could be used. For example; there could be a FPQ with three service classes on top: class 1 for *priority messages* and the like; class 2 for traffic that is prioritized for technical reasons; and class 3, a best effort class. Class 2 could then be divided into subclasses, and a WFQ could be used to determine the mutual ordering within that class.

5.1 Future work

In this work we have only studied the performance of the queuing systems for one network topology, it would be interesting to investigate if and how the performance is effected by network size, network connectivity and mobility.

Furthermore, we have used a very simple Poisson model for the arriving traffic. A natural extension of our work would be to use a more realistic traffic model that models a *real-time* application such as a video conference or phone calls. With such a model it would be interesting to study the sample probability distribution for the end-to-end packet de-

[1] Military term for messages that are allowed to interrupt all other messages.

lay for individual sessions, like a single phone call, and see how that is affected by different queuing schemes.

Bibliography

[1] *Militärstrategisk doktrin*, Försvarsmakten, 2002, M7740-774002.

[2] Steven Blake, David L. Black, Mark A. Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss, "RFC 2475: An architecture for differentiated services," ftp://ftp.rfc-editor.org/in-notes/rfc2475.txt, Dec. 1998, Status: PROPOSED STANDARD.

[3] Constantinos Dovrolis and Parameswaran Ramanathan, "A Case for Relative Differentiated Services and the Proportional Differentiation Model," *IEEE Network*, vol. 13, no. 5, pp. 26–34, 1999.

[4] Laura Merie Feeney, Bengt Ahlgren, and Assar Westerlund, "Spontaneous Networking: An Application-Oriented Approach to Ad Hoc Networking," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 176–181, June 2001.

[5] Elizabeth M. Royer and Chai-Keong Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications Magazine*, vol. 6, no. 2, pp. 46–55, Apr. 1999.

[6] Raphael Rom and Moshe Sidi, *Multiple Access Protocols Performance and Analysis*, Springer-Verlag, 1989.

[7] Randolph Nelson and Leonard Kleinrock, "Spatial-TDMA: A Collision-Free Multihop Channel Access Protocol," *IEEE Transactions on Communication*, vol. 33, no. 9, pp. 934–944, Sept. 1985.

[8] Alan Demers, Srinivasan Keshav, and Scott Shenker, “Analysis and Simulation of a Fair Queueing Algorithm,” in *SIGCOMM*. ACM, 1989, pp. 3–12.

[9] Abhay K. Parekh and Robert G. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344–357, June 1993.

[10] Hubert Zimmermann, “OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection,” *IEEE Transactions on Communications*, vol. 28, no. 2, pp. 425–432, Apr. 1980.

[11] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, *Network Flows. Theory, algorithms, and applications.*, Prentice Hall, 1993.

[12] Leonard Kleinrock, *Queuing Systems, Volume II: Computer Applications*, Wiley-Interscience, 1976.

[13] Jimmi Grönkvist, *Assignment Strategies for Spatial Reuse TDMA*, Licentiate thesis, Royal Institute of Technology, Stockholm, Mar. 2002.

[14] Börje Asp, Gunnar Eriksson, and Peter Holm, “Detvag-90 – Final report,” Scientific Report FOA-R–97-00566–SE, Swedish Defence Research Establishment (FOA), 1997.