FFI-rapport 2013/01899

NATO Narrowband Waveform (NBWF) – network layer flow control protocols

Tore J. Berg

FFI Forsvarets forskningsinstitutt

FFI-rapport 2013/01899

NATO Narrowband Waveform (NBWF) – network layer flow control protocols

Tore J. Berg

Norwegian Defence Research Establishment (FFI)

24 July 2013

FFI-rapport 2013/01899

1295

P: ISBN 978-82-464-2280-0
E: ISBN 978-82-464-2281-7

Emneord

Modellering og simulering

Datanett

Radionett

Trådløse nett

Approved by

Torunn Øvreås	Project Manager
Anders Eggen	Director

English summary

NATO has an ongoing activity with the objective to develop a narrowband waveform (NBWF) standard for the VHF/UHF band. This is a single-channel mobile ad-hoc network (MANET) which shall serve voice and data traffic simultaneously over a 25 kHz radio channel. As a member of the NBWF working group, FFI is working on a proposal for a network layer flow control protocol in multihop networks.

A narrowband network has low throughput capacity, and to experience network congestion during usage must be regarded as an ordinary event. Therefore it is important to have a robust flow control protocol that prevents network collapse. The first part of this document addresses the challenges we are faced with in a multihop network when the current NBWF link layer is used. Then we propose two flow control protocols and implement these protocols in the NBWF network simulator. The solutions proposed are based on a cross layer design where we also consider services at the lower layers, their buffer structure and the layer interface flow control. A number of simulation experiments are conducted with the objective to study protocol behaviour under different conditions.

This report concludes that NBWF must implement a network layer flow control protocol, that is, some signalling between adjacent nodes is required. However, which of the two flow-control protocols to select demands further study. Based on the simulation experiments, we conclude that the link layer must implement an additional function (LLC exponential backoff) as a countermeasure to the hidden-node problem.

Sammendrag

Nato har en pågående aktivitet for standardisering av en smalbåndsbølgeform (NBWF) for bruk i VHF/UHF-området. En modellerings- og simuleringsaktivitet ved FFI skal bidra til NBWF ved å vurdere alternative protokollfunksjoner for betjening av tale- og datatrafikk i distribuerte mobile nett. Dette dokumentet omhandler flytkontroll på nettnivå over luftgrensesnittet.

Et smalbåndsnett har lav trafikkapasitet og metningssituasjoner kan lett oppstå. Flytkontroll skal beskytte nettets kjerneprotokoller mot overbelastning slik at nettet kan opprettholde en stabil tjenestekvalitet for alle brukere under høytrafikkperioder. Den første delen av dokumentet gir en oversikt over noen utfordringer ved bruk av gjeldende NBWF linklagsprotokoller. Deretter beskrives to mulige protokoller for flytkontroll. Disse implementeres i NBWF-simulatoren. En serie simuleringseksperimenter er utført for å studere kandidatenes oppførsel under varierende trafikkforhold.

Rapporten konkluderer med at NBWF må ha flytkontroll på nettverksnivå. Å velge hvilken av de to foreslåtte protokollene som gir best ytelse for NBWF, krever bruk av mer komplekse nett-topologier enn de som er simulert. Under arbeidet ble det observert en svakhet i det gjeldende NBWF-linklaget. Rapporten forslår en løsning (”LLC exponential backoff”) og estimerer nettets kapasitetsøkning etter implementasjon av ”LLC exponential backoff”.

Contents

1 Introduction

NATO has an ongoing activity with the objective to develop a narrowband waveform (NBWF) standard. This is a single-channel mobile ad-hoc network (MANET) which shall serve voice and data traffic over a 25 kHz radio channel. NBWF uses TDMA and a dynamic reservation protocol to allocate transmission capacity for IP traffic, and this reservation protocol is based on a random access protocol [7].

The purpose of this document is to propose a NBWF flow control protocol over the air interface. This protocol operates at sublayer 3a according to the NBWF reference model [7, figure 3.1].

A multihop network refers to a network where *routing* and *relaying* functions must be implemented to increase the network service coverage area beyond the radio coverage area. Routing and relaying are two different network functions. The aim of a routing function is to determine appropriate routes between the network's edge-nodes. A relaying function forwards a data packet to another node after its next hop node address has been assigned by the routing functions. A routing function demands implementation of one or more routing protocols. This document does not consider routing, but deals with network layer relaying protocols.

We need flow control rules and mechanisms in multihop networks to: 1) Protect against transit queue overflow, 2) Maintain an efficient military priority handling service [9], 3) Prevent loss of transmission capacity due to packet lifetime expire, 4) Maintain fairness between fresh traffic and relay traffic; and 5) Soften the hidden-node problem.

A well designed network keeps the transit queues smaller than the fresh traffic queues, that is, packets shall be queued at the entry-nodes and not at the relay nodes. With regard to buffer space, transit buffer overflow is of little concern in NBWF. Due to the low traffic capacity and the low maximum packet lifetime[1], only a small memory space is required in each relay node. However, there are a number of other reasons to keep the transit buffer queues small. If the transit queues become long during network congestion, the packet lifetime control function (explained in section 4.4) starts to delete packets in the transit queue. Transit packets have consumed transmission capacity and we surmise fast throughput degradation when a network starts to drop transit packets. Consider Figure 1.1, which depicts a saturated network region some radio hops away from two traffic sources. To maintain the quality of service for the highest priority traffic, the low priority traffic stream should be decelerated before reaching the saturated region. This is important to provide the same priority handling efficiency as we experienced in an "all-hearing-all"-network [9].

As a collision avoidance scheme, NBWF has two mechanisms in the MAC layer. The first is a Preamble Sense Multiple Access (PSMA)[2] scheme that effectively coordinates neighbour nodes, that is, nodes in direct radio contact. The second scheme is based on a reservation technique

[1] The NBWF core protocols are designed to delete a packet when reaching a certain age.

[2] All the transmission bursts are prefixed by a fixed unique bit pattern named a preamble. A node which starts to listen for a transmission after the preamble cannot detect the transmission and act as if the channel is idle.

similar to the IEEE 802.11 MAC Request-To-Send/Clear-To-Send technique[3], and this scheme reduces the interference among hidden-nodes, see chapter 2. This document deals with flow control in multihop networks where hidden-nodes may be cumbersome and lead to throughput degradation. A flow control protocol which reduces the hidden-node problem improves the traffic conditions in multihop networks.

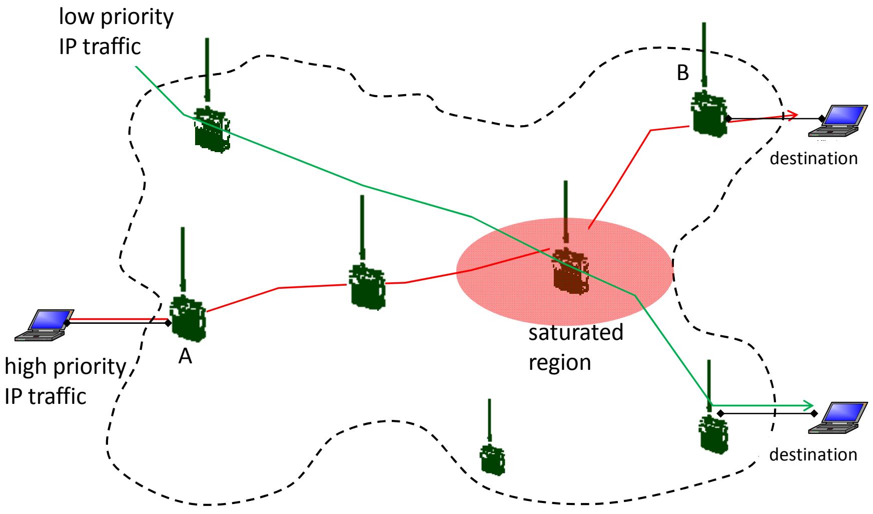

Figure 1.1 A low priority and a high priority traffic stream passes a saturated region. This document proposes a flow control mechanism based on back pressure, that is, the information about saturation propagates backwards towards the entry-node.

Our simulation experiments must consider NBWF networks at different congestion levels. Congestion can be obtained by increasing the IP payload size, increasing the IP packet arrival rate, or both. In NBWF, it is the MAC connection setup phase that suffers most from the hidden-node problem. This means that NBWF is less sensitive to long IP payloads as in networks without a reservation phase. Therefore we change the offered network traffic by changing the packet arrival rate while keeping the IP payload size fixed at approximately 500 bytes.

The purpose of the simulation experiments is not to predict the absolute throughput/delay-performance of NBWF, but to detect any anomalies and test the efficiency of the protocols proposed.

Chapter 3 proposes two different network level flow control protocols for NBWF and both were implemented in the NBWF simulator. These flow control protocols use additional network level functions which are not especially designed for assisting flow control functions. For completeness, we have added chapter 4 "Auxiliary Network Level Functions" that outlines the auxiliary functions required.

[3] NBWF sends connect request and connect confirm packets. They are always sent even in a fully meshed network.

Chapter 5 and 6 present our first simulation experiments and are based on the two simplest multi-hop networks possible; a three-node chain and a four-node chain, respectively. The purposes of these experiments are to get a basic understanding of the proposed flow control protocols.

Chapter 6 discovers low efficiency of the NBWF connection setup phase due to the hidden-node problem. As a countermeasure to this problem, chapter 7 studies if an LLC exponential backoff function can improve the efficiency.

1.1 Terminology

The first part of this section defines the most important terms used in this report, while the second part specifies the probes used and describes what they measure. A network is a stochastic process and a probe is a tool for observing the network behaviour. In the simulator, a probe is a software component/object which collects data (e.g., end-to-end packet delays) and produces an estimate of the first order moment.

MAC entity
The active process in a radio node which executes MAC layer functions. For example, it is the MAC entity that executes the MAC protocol.

AHAnN
All-hearing-all (AHA) refers to a network topology where all the nodes have overlapping radio coverage areas (fully connected topology). n*N* specifies an AHA-network containing *N*-nodes (e.g. AHAn25).

Capture model
A (packet) capture model gives the air frame failure rate (loss due to bit-error) as function of the SNR during reception.

Zero capture model
The preamble of an incoming air frame is handled according to the normal NBWF procedure [6]. When the receiver is locked onto an air frame (successful SOM event), any overlapping transmission(s) introduce bit-errors in the payload.

Perfect capture model[4]
The preamble of an incoming air frame is handled according to the normal NBWF procedure [6]. When the receiver is locked onto an air frame (successful SOM event), this air frame is always correctly received regardless of the SNR condition of the radio channel.

NBWF capture model
This is the normal operating mode as described in [6].

[4] Simulation experiments are sometimes executed with three capture models: normal, zero and perfect. This is an efficient method to test the effect of the near-far problem.

The near-far problem
A receiver is locked to a weak signal from a distant node when a node in the vicinity starts to emit a high energy signal. The stronger signal overrides the weaker signal and the first packet is lost.

Sink-node
An end-destination for an IP traffic stream.

Entry-node
A radio node which is the end-source node for an IP traffic stream (fresh input traffic) from an IP client.

Edge-node
A node taking the role as *sink-node* and/or *entry-node*.

Throughput capacity
When the IP traffic requests use of ARQ, the offered traffic and the throughput shall follow a straight line up to the point where the radio channel becomes congested, see Figure 1.2. The throughput capacity is defined as the point on the curve where the deviation between the offered traffic and the throughput becomes higher than approximately 1%.

Maximum throughput
The highest point on a throughput plot, see Figure 1.2. Only loss tolerant IP applications can operate at this load level.

Below we specify the probes used in this report.

P(receive CC), p_{CC}
This measurement is taken by each node that has sent a CR PDU and expects to receive a CC PDU. When a CC PDU is received, the value sampled is one. Otherwise, zero. Two or more CR PDUs may be sent simultaneously[5] (i.e., a packet collision event) but then the probability to get a CC PDU is very low since NBWF demands a positive signal-to-noise ratio (SNR) to generate a CAS, reference [6, table B.4].

[5] This report uses a fixed pathloss model. Two overlapping transmissions with equal signal levels at the receiver give 0 dB SNR.

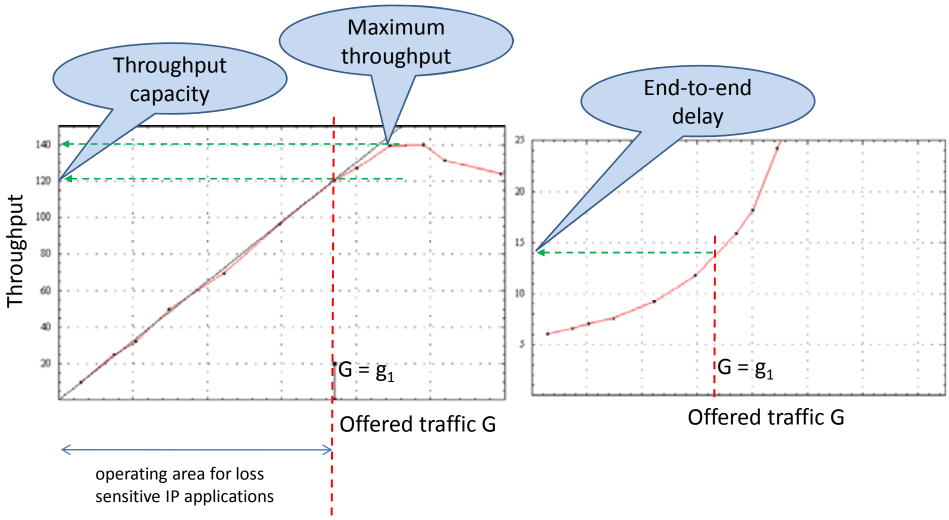

Figure 1.2 Throughput and delay plot examples.

Throughput [bytes/s]
All the network layer entities in the sink-nodes reports the payload size to this probe when they receive a packet destined for the IP client. This probe measures the average number of bytes received over a time window of 1 second and sends this value to a batch-means module [11].

End-to-end delay [sec]
In the simulator, all packets get a timestamp when they are created and the sink-nodes are then able to calculate their age. Of course, lost packets due to buffer overflow or lifetime expiry are not included.

Input buffer queue size [#packets]
The entry-node makes a sample each time it inserts a new packet in this queue.

Transit buffer queue size [#packets]
The relay-node makes a sample each time it inserts a new packet in this queue.

Measured Forward Delay [sec]
The pacing protocol specified in section 3.1 measures the packet forwarding delay according to equation (3.1) in chapter 3. In the simulator, the MFD-probe collects samples from this equation.

2 The Hidden-Node Problem

The NBWF MAC protocol, outlined in [7], is a connection oriented protocol which means that a MAC connection must be established before a data packet can be sent over the air interface. Consider node A in Figure 2.1, where node A has data traffic to node B. A number of nodes (H_B) is neighbours to B but is outside node A's radio coverage area. The nodes in the node set H_B are referred to as *hidden-nodes* since they cannot detect any transmission from node A. Similarly, node A has the hidden-node set H_A. Node A serves an IP packet stream towards B, which implies that the two nodes must exchange CR/CC PDUs. Any CC PDU may be destroyed by transmission(s) from the nodes in the hidden-node set H_A when one or more nodes in the set failed to receive the CR PDU. If one of the two events below occurs, the CR PDU from A to B is lost:

e_1: Node B is locked onto a transmission from H_B when A starts to transmit; or
e_2: Any node in the H_B -set starts to transmit while node B demodulates the packet from A.

Equally, the CC PDU from B to A is lost when:

e_3: Node A is locked onto a transmission from H_A when B starts to transmit; or
e_4: Any node in the H_A -set starts to transmit while node A demodulates the packet from B.

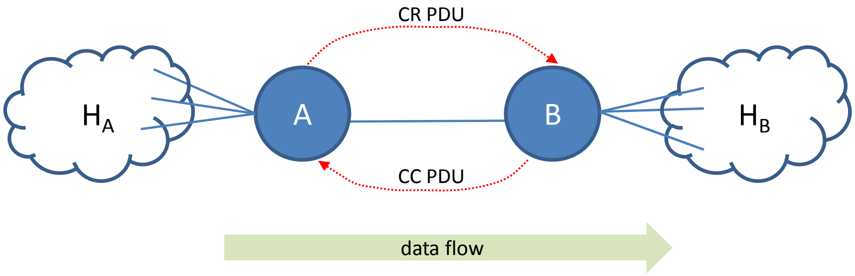

Figure 2.1 Two nodes in a multi-hop network where the data traffic is directed from A to B. Both nodes have hidden nodes.

However, a packet may survive event e_2 and e_4 if the overlapping transmission is weak enough to give a signal-to-noise ratio (SNR) above approximately 4 dB[6] [6]. The success rate is therefore also affected by the relative received signal level from the transmitters. To account for this effect, some of the simulation experiments in this document include different packet capture models (defined in section 1.1).

We have now explained the term hidden-nodes and set focus on additional challenges we meet when relaying IP packets over multiple hops. To ease the presentation, we select a four node chain where node A in Figure 2.2 is the only node with a packet destined for node D. Below we take a review of the most significant time instances and events to get a single packet from A to D.

[6] The value depends on the interleaver length used and this value refers to the N1 mode.

Actions between t_0 and t_1 (forwarding on the first hop)

Node A initiates the MAC connection establishment procedure at time instance t_0 by sending an MAC CR PDU. Node B responds with a CC which is overheard by node C, a hidden-node to A. Node C remains silent until it registers a disconnection packet, or its connection lifetime timer expires. At time instance t_1, node A takes down the connection after a successful delivery to node B.

Actions between t_1 and $t_1 + Q_B$ (queuing delay in node B)

After B has sent the MAC Disconnect Confirm (DC) PDU, B starts to serve the relay packet by initiating a MAC scheduling process. At this point in time, node C and all other nodes in the neighbourhood may also start a MAC scheduling since A's MAC reservation has ended. The connect request packet for the transit packet is sent on the air at $t_1 + Q_B$. Node A is a hidden-node to C while node D is a hidden-node to B. In this simple network Q_B includes the MAC disconnection delay and the MAC random access delay. If the network had been large and node B had many nodes to compete with, Q_B would have included many MAC delivery cycles.

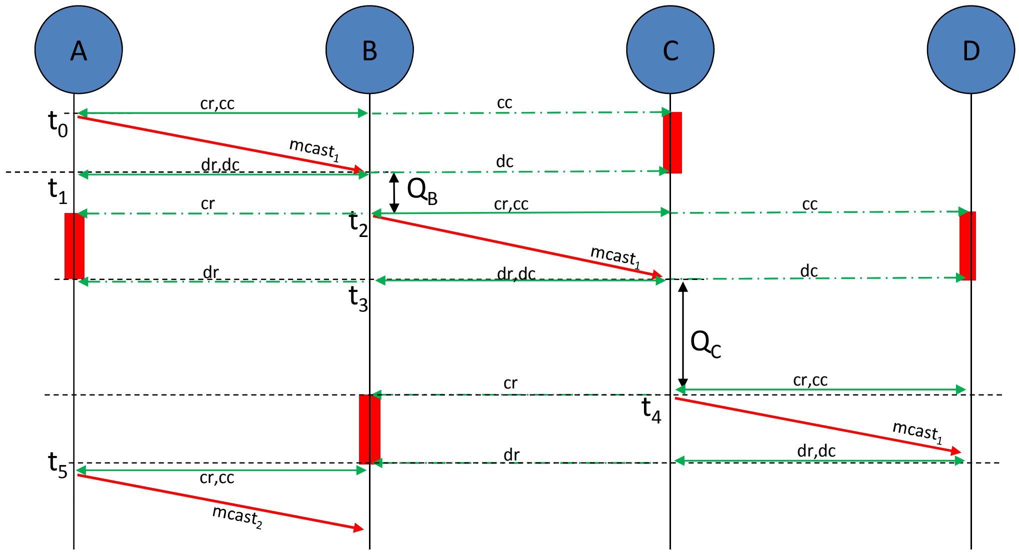

Figure 2.2 Time-sequence diagram for packet relaying. Entry-node is A and sink-node is D. To ease the drawing, the MAC signalling (green arrows) is set to take zero time.

Actions between t_2 and t_3 (forwarding on the 2^{nd} hop)

The MAC CR/CC PDU exchange forces the {A,D}-nodes to be idle in this time period. C receives the data packet at t_3 and B disconnects the MAC connection.

Actions between t_3 and $t_3 + Q_C$ (queuing delay in node C)

No MAC connections exist here and all busy nodes may access the channel. Node A and C are hidden nodes, and if A initiates a new connection setup, it may interfere with the forwarding of its own packet at time instance t_4.

Actions between t_4 and t_5 (forwarding on the last hop)
The CR PDU sent by node C blocks node B, a hidden-node to D, and B will not disturb the packet forwarding process. On the other hand, node A is not blocked by the MAC protocol and may start to send. This is an invidious event since B cannot respond and node A should defer from further transmissions to after time instance t_5.

Now we have completed the description of the packet relay process and shown that the NBWF MAC reservation process (connection setup) is an effective countermeasure to the hidden-node problem for two-hop paths with *one-way traffic* (A -> C or C->A) since the CR/CC-signalling prevents collisions between A and C. However, three-hop paths remain a challenge since the MAC protocol does not prevent node A from sending in [t_3, t_5] which interfere with the packet delivery on the third hop. When a packet has reached more than 3-hops away from the source node, the source does not longer interfere with the packet forwarding process.

In networks with *two-way traffic* (e.g., A -> C and C->A), two connect setup phases are initiated simultaneously if two nodes get a fresh packet simultaneously. The MAC collision avoidance function does not prevent a collision at node B.

In NBWF, the relay function of a multicast packet and a unicast packet is identical in a chain network. The only difference is that a unicast packet may request use of ARQ in the LLC layer.

3 3a Layer Flow Control

This chapter deals with *Network* Level Flow Control (NLFC) for multihop unicast/multicast traffic. One purpose of NLFC is to prevent overflow in the network transit buffers when a fast source-node (node A in Figure 3.1) sends multihop traffic via a saturated region (relay-node R2). Saturation may be caused by high local traffic streams, or background noise. Deletions of transit packets are very costly since they have consumed transmission capacity. During congestion, a well designed network shall do most of the queuing in the input buffers at the entry-node, while the transit buffers shall store a few packets only. Another purpose of NLFC should be to reduce the probability that node A in Figure 2.2 interfere with the forwarding of its own packet in the time interval [t_3, t_5].

Due to the network dynamics (node mobility and an unpredictable traffic volume/pattern) we do not base the flow control on a reservation technique in the forward direction. A reservation technique may generate much traffic depending on many factors such as network topology and the IP streams' lifetime. Instead we apply a backpressure technique where the information about saturation propagates in the backward direction towards the entry-node.

We look at two different flow control algorithms named pacing [10] and Periodic Explicit Congestion Notification (PECN). PECN is a new solution designed for NBWF to use the collision free data channel provided by the NBWF super frames. Both pacing and PECN are described in detail in this chapter. In addition to pacing/PECN, the NLFC requires the following elements of procedure to operate:

1) *Admission control in the entry-node.*
   A packet counter at layer 3a in the entry-node counts the number of packets per (end-destination, priority)-pair. If this number is higher than Q_{max}, the admission control does not accept more (end-destination, priority)-packets from the local IP client.
2) *Control of the remaining lifetime (L_{min}).*
   Upon arrival at the entry-node, each packet is assigned a maximum lifetime (L_{max}). Any packet is deleted if they do not reach the end-destination within this time limit. That means, the maximum end-to-end delay we shall measure is L_{max}. When the 3a layer entity takes a packet from the fresh traffic queue or the transit queue, it tests the remaining lifetime against L_{min}. If the remaining lifetime is lower than L_{min}, the packet is deleted silently.
3) *Maintenance of a GUID-cache.*
   This is explained in section 4.3.
4) *Layer interface flow control.*
   This is explained in section 4.1.

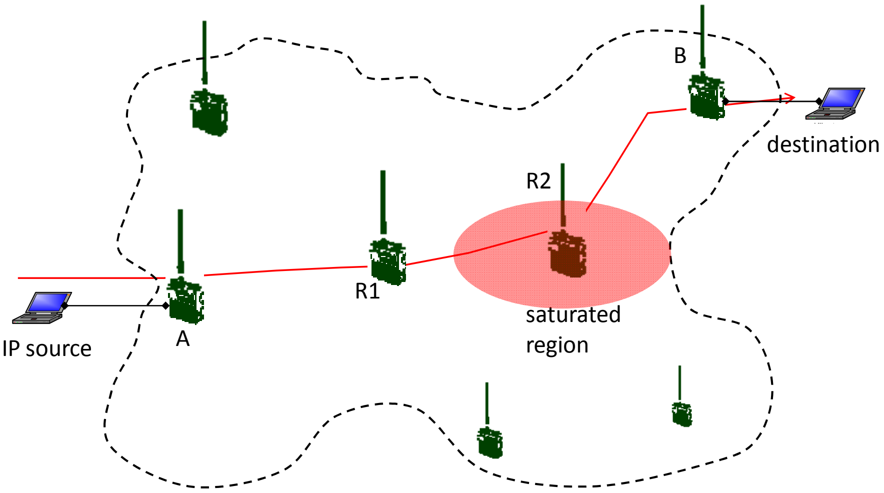

Figure 3.1 An IP traffic stream passes through a saturated network region.

3.1 Pacing

A pacing protocol shall provide flow and congestion control over multihop paths. Based on [10], this section specifies a pacing protocol for NBWF. The NBWF node's buffer system is scaled to hold one packet below layer 3a (see Figure 4.1), and the 3a layer entities can therefore effectively choke the outgoing traffic when a saturation situation occurs. A well scaled crosslayer buffer system is essential for the pacing protocol.

Consider the three hop route in Figure 3.2. As the first rule, the 3a protocol should not allow more than one outstanding data packet to **each 3a peer-entity**. This is achieved by adding a forced idle period (pacing) after serving packet 1 in the figure. B starts to relay A's data packet at t_2. In principle, node A may send in $\langle t_3, t_4 \rangle$ but the likelihood that the two hidden-nodes A and C interfere with each other is high (Q_C is a stochastic variable). Node B is blocked by MAC protocol in the interval $\langle t_4, t_5 \rangle$, so A should not send the next packet here. Node A shall desist from further transmissions to B until the pacing time delay *($t_5 - t_1$)* has elapsed[7]. The 3a protocol entity measures the forwarding delay *($t_3 - t_1$)* to each of its neighbours and uses these samples to calculate pacing intervals.

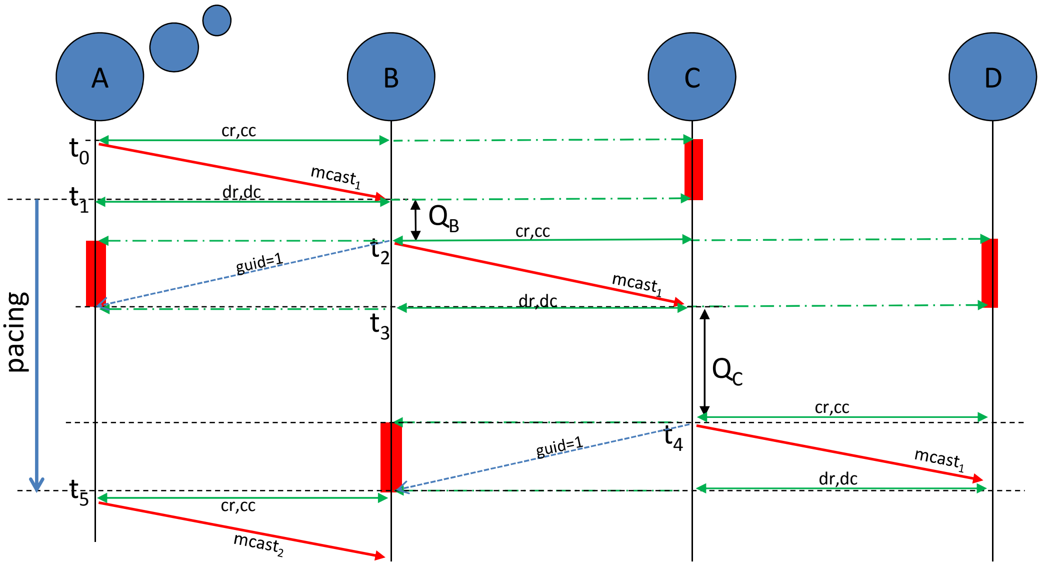

Figure 3.2 Time-sequence diagram for packet forwarding from entry-node A to sink-node D. Related data packets are tagged with the same number (GuId). The red rectangles mark the time periods where the MAC protocol turns off the hidden-nodes.

Let $D_{fd,B}$ denote the packet forwarding delay *($t_3 - t_1$)* through node B. $D_{fd,B} = Q_B + Tx_{B \to C}$ where the queuing delay *($t_2 - t_1$)* in node B is Q_B and $Tx_{B \to C}$ is the transmission time *($t_3 - t_2$)* for packets from node B to C. If the pacing timer is started at time instance t_1 in the figure, the pacing interval should be $T_{PI} = D_{fd,B} + D_{fd,C}$. However, node A cannot measure the forwarding delay $D_{fd,C}$ through C, so A assumes the traffic condition within node C's neighbourhood is similar as at node B, and sets the pacing interval to $T_{PI} = 2 \cdot D_{fd,B}$.

Node A shall measure the forwarding delay $d_{fd,B,i}$ for relay packet number[8] *i* and estimates the forwarding delay based on the exponential moving average:

$$\hat{D}_{fd,B,i} = (1-\gamma) \cdot \hat{D}_{fd,B,i-1} + \gamma \cdot d_{fd,B,i}, \qquad 0 < \gamma < 1, \qquad i\text{:integer} > 0 \tag{3.1}$$

[7] Pacing is not actually needed to prevent collisions in $\langle t_1, t_2 \rangle$ because the MAC protocol eliminates collisions (≈0) between A and B.

[8] Each error-free packet received gives a sample.

The constant γ determines the weight between new and old samples and sets the adaptation time to new load levels. Individual $\hat{D}_{fd,i}$ shall be maintained per relay node and per priority, and $\hat{D}_{fd,0}$ shall be assigned the initial forwarding delay given in Table 3.1.

Parameter	Value
Smoothing factor γ	1/3
Initial forwarding delay per priority {IFD$_{P0}$, IFD$_{P1}$, IFD$_{P2}$, IFD$_{P3}$}	{0.5,0.5,0.5,0.5} [sec]
EFD inactivity period	60 sec

Table 3.1 Pacing parameters used by the simulator.

Assume node A in Figure 3.3 issues an *LLC-Data.request* at time instance t_2.
This packet shall use node B as a relay and node A must measure the forwarding delay $d_{fd,i} = t_8 - t_6$. MAC has not established a connection at t_2. Hence, the packet gets a local waiting time delay Q_A at the LLC layer which is the time MAC uses to set up a connection[9]. Q_A is a stochastic variable. This delay may be long in a large network since MAC must compete during many MAC access cycles before it wins. Therefore the LLC service provider shall not store any outgoing packet before the MAC layer is ready to send a packet on the air.

Node B saves the packet in the transit queue at t_6 and sends the packet on the air at t_7. The transmission is completed at t_8 and node A receives the passive acknowledgement via an *LLC-Data.indication*.

We now take a detailed look at the signalling sequence over the 3a/LLC-interface:

Actions between t_2 and t_3 (packet arrival event in node A)
The LLC service provider has signalled earlier (t_1 in Figure 3.3) that it is idle when a new IP packet arrives at t_2. The 3a entity informs the LLC entity about an awaiting packet by issuing an *LLC-Data.request*. This primitive does not contain the packet but does only signal the parameters needed by the lower layer entities, see Table 4.1. Upon receiving this primitive, the LLC entity must, without any delay, issue a Xoff to tell the 3a entity that it is busy and cannot handle new requests.

Actions between t_4 and t_5 (MAC connection established)
A MAC connection is ready for use and we have reached the time instance in the TDMA frame where a data burst shall be sent on the air.

[9] The simulator cheats when it comes to implementation of packet lifetime update since we carry the remaining lifetime as LLC PCI. Hence we do not need a ”real time” service over the 3a/LLC interface. A real NBWF node must carry this information as 3a PCI and the 3a entity cannot send the packet down before the MAC entity has established the connection and requested the data packet.

This information is brought to the 3a entity by an *LLC-Status.indication* primitive [10] and the *LLC-Data.request* primitive hands over the payload which is immediately sent on the air.

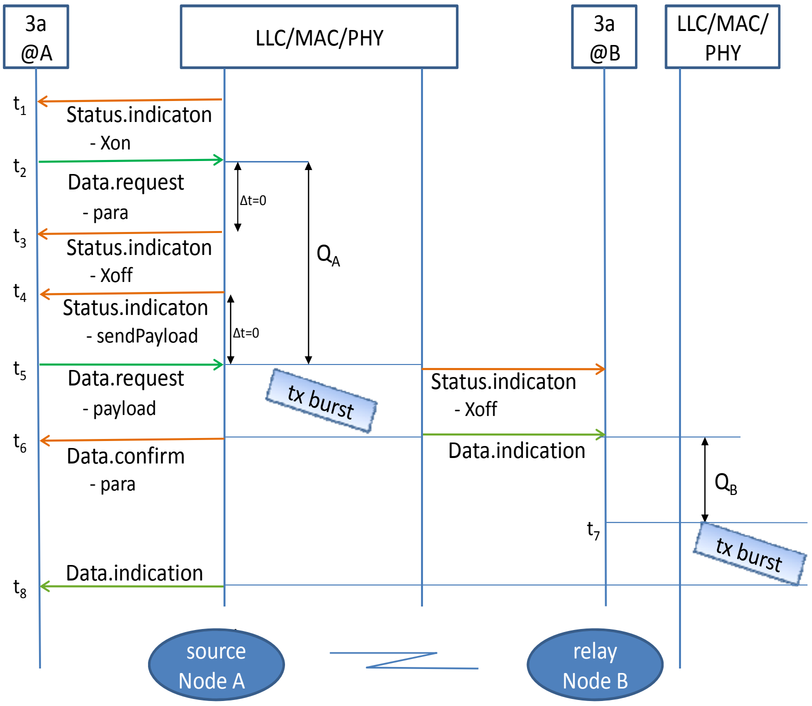

Figure 3.3 Detailed time-sequence diagram for packet forwarding as observed by the 3a entity in node A. The box marked "LLC/MAC/PHY" is the LLC service provider by which the two peer-layer entities 3a@A and 3a@B communicate. Q_A is the local delay in the source node added by the layers below layer 3a.

Actions at time instance t_6 (first hop completed)
a1: Node A's 3a entity receives an *LLC-Status.indication(sduSent)* when the 3a DT PDU has been sent over the air interface. The 3a entity saves the current time since the t_6 value is needed later. If the packet lifetime expires while under service in the LLC service provider, no *LLC-Status.indication* primitive will be received. However, a missing primitive is of no concern because the lifetime expires in both layers and both layers delete the packet simultaneously.

a2: Node A's 3a entity starts the pacing timer using the timeout interval
$T_{PI} = \min(2 \cdot \hat{D}_{fd,B,i}, \textit{ remaining lifetime})$ [11], see equation (3.1).
a3: After the LLC entity has sent the *sduSent*-signal, it issues a *MAC-Disconnect.request* to release the MAC connection.

[10] A real NBWF radio node should inform the 3a layer about this event some millisecond before this point is reached to take into account the processing delay at layer 3a (the simulator has zero processing delay).
[11] Timers shall not run for packets that are deleted and the value to use is the smallest one.

Actions at time instance t_8 (second hop completed)

Node A overhears the forwarding of its own packet identified by the GUID and calculates the packet forwarding delay $d_{fd,i} = t_8 - t_6$ and updates the estimated forwarding delay for node B according to equation (3.1).

3.2 Periodic Explicit Congestion Notification (PECN)

Explicit congestion notification[12] means that the saturation level is signalled explicitly on the air by the nodes which experience **transit** traffic congestion. *Periodic* expresses that this is done periodically. We use broadcast in a super frame slot. The adaptive MAC scheduling function [9, chapter 3] is based on a periodic broadcast of a MAC Load Level (MLL) report. To minimise the transmission capacity consumed by PECN, the PECN reports are included in the same transmission bursts as the MLL-reports. This gives a report cycle period of $0.2025 \cdot n$ seconds, which is 10 seconds in a 50-node network.

A PECN-report contains the following attributes[13]:

```
PecnReport
{
   3bits   flowControlLevel; // 5 signalling states used
   8bits   sourceNode;       // MAC address
}
```

The *flowControlLevel* signals *Xon and Xoff* for the priority level $P_i \in \{P0, P1, P2, P3\}$, where *P0* is the lowest priority level. A node receiving *Xoff@P_i* from the *sourceNode*[14] is only allowed to use this node as a relay for traffic having priority strictly larger than P_i. A *Xoff@P3* blocks all **transit**[15] traffic via the *sourceNode*. In the opposite end we have the *Xon* which opens for all priority levels. Below we describe how the PECN operates in a four-node chain, see Figure 3.4. Here node A is the entry-node and node D is the sink-node.

Actions between t_0 and t_1 (forwarding on the first hop)

Node A initiates the MAC connection establishment procedure at time instance t_0 by sending an MAC CR PDU. Node B responds with a CC PDU which is overheard by node C, a hidden-node to A. Node C remains silent until it registers a disconnect packet, or its MAC connection lifetime timer expires. At time instance t_1, node A takes down the connection after a successful delivery to node B.

[12] Do not confuse this term with the IETF’s ECN.

[13] Source node addresses are carried as MAC PCI.

[14] This address is also required by the MLL-reports [9, chapter 3].

[15] Single-hop and ”last-hop” traffic to a Xoff-node is allowed.

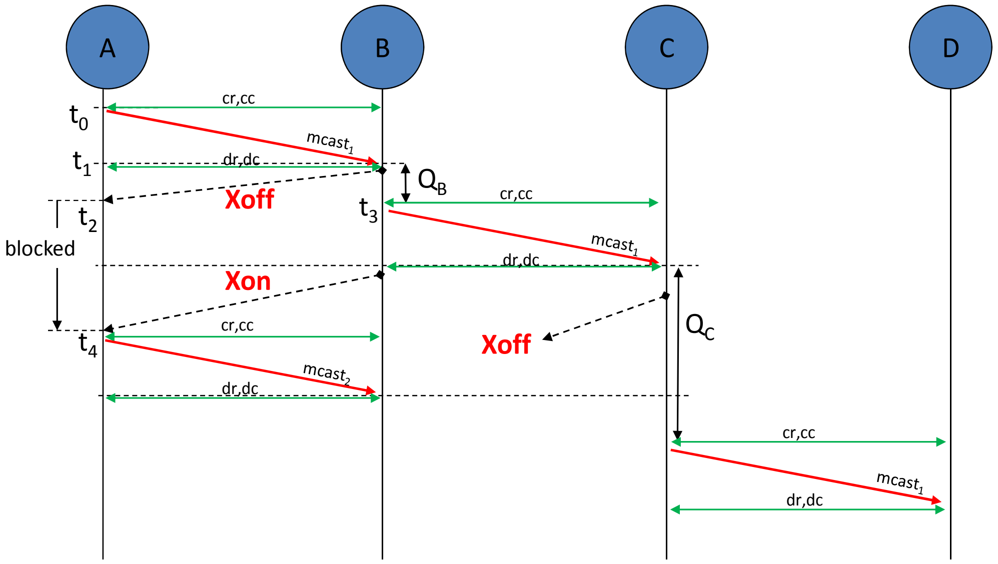

Figure 3.4 Time-sequence diagram for packet relaying in a four node chain using PECN. Node A is the entry-node and node D is the sink-node.

Receiving Xoff at t_2.

When node B inserts the relay packet in its transit queue, it counts the number of packets stored. If this number is larger than q_{trans}[16], it signals Xoff@priority in the next PECN-report. Otherwise, node B signals Xon. The drawing here assumes that the queue size passes the threshold level and node B must signal Xoff. A node cannot skip the emission of a Xoff/Xon signal in its dedicated slot since background noise may have destroyed one or more of the signalling packets sent earlier. The Xon/Xoff signalling process is identical for all the priority levels.

The figure depicts a Xoff just after node B has received the multicast packet (t_1), but the signal might be sent much later in large networks. Until node A receives the Xoff, node A is allowed to forward more packets to node B, and A might continue to fill up the transit buffers in B for a significant period of time. The outgoing transit traffic from node A to node B is blocked from time instance t_2, and the blocking period last until a Xon with node B as the source node is received, or the maximum blocking time period timer expires. A timer is needed as a protection against Xon-losses in a noisy radio environment.

Actions between t_3 and t_4 (forwarding on the second hop)

After a random waiting time Q_B, node B establishes a MAC connection, sends the packet on the air and then releases the connection. Here we assume that the transit queue size becomes lower than the threshold level and then node B issues a Xon.

Receiving X_{on} at t_4.

Node A is now allowed to send a new packet to node B. If a packet is awaiting service, node A repeats the same time-sequence as described from t_0.

[16] An integer value to be determined from simulation experiments.

3.3 Unicast versus Multicast NLFC

PECN treats multicast and unicast traffic very similarly in pure chain networks, but operates differently on unicast and multicast streams in other topologies, see Figure 3.5. In the NBWF simulator, a radio node has one 3a entity for each adjacent node that serves unicast packets while one common 3a entity serves all the multicast packets. The unicast 3a entities (one instance for each neighbour node) and the single multicast 3a entity are unaware of each other. One unicast stream is therefore served by one 3a entity. No local information is exchanged between the unicast 3a entities depicted in the figure. This implies that the node 0 may send unicast packets to node 2 and possibly disturb the packet forwarding by relay node 1. As shown in the figure, multicast streams to different relay nodes are handled by the same 3a entity. Regardless of which multicast relay that did send a flow control signal, the last Xon/Xoff-signal received determines the flow control state of the 3a entity.

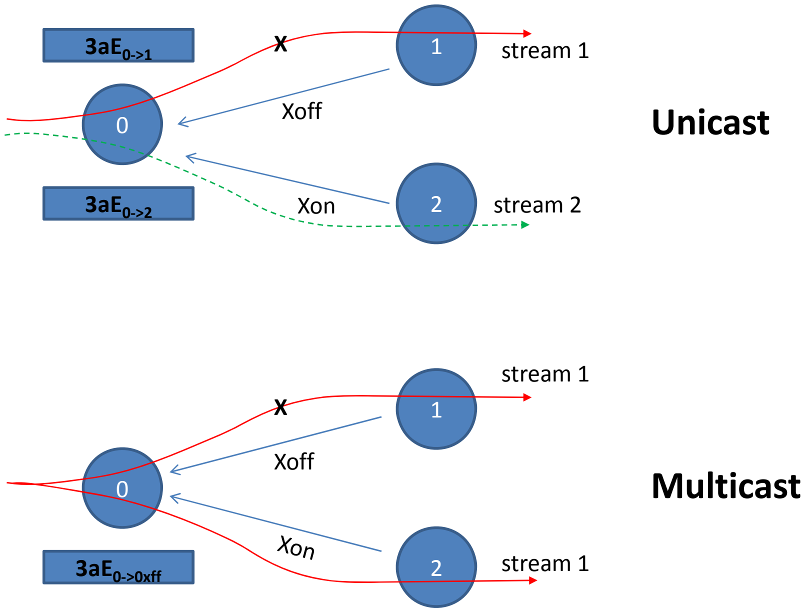

Figure 3.5 PECN peer-to-peer flow control on unicast and multicast streams. In case of multicast stream, the originating 3a entity has a one-to-many relationship to its neighbours and one node may signal Xoff while another signals Xon.

Also pacing treats multicast and unicast traffic very similarly in pure chain networks, but operates differently on unicast and multicast streams in other topologies, see Figure 3.6. However, the difference is not as noticeable as for PECN since only the MFD sampling process is affected and no explicit signalling of flow control states are required. With multicast streams, node 0 acquires multiple MFD samples for each relay packet sent.

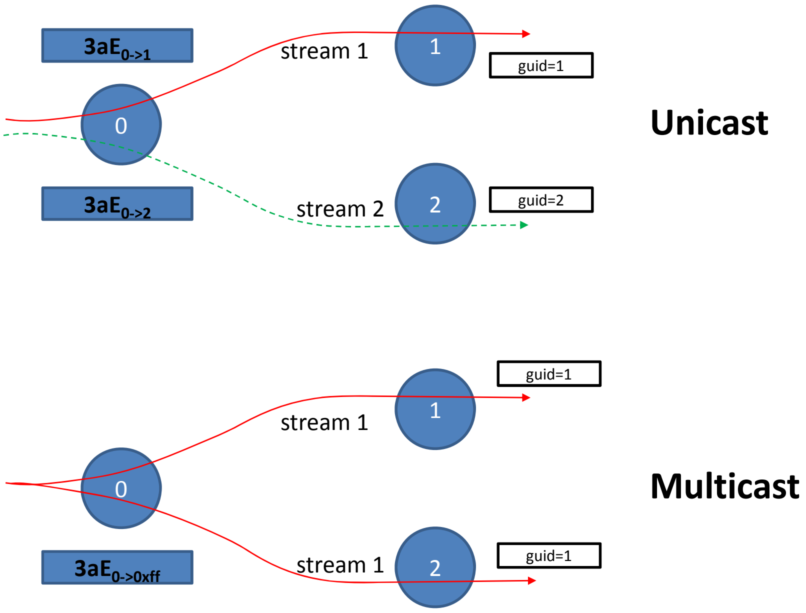

Figure 3.6 Pacing and packet forwarding.

4 Auxiliary Network Level Functions

To implement an efficient network flow control function, we need additional protocol functions and must introduce some requirements to the lower protocol layers. This chapter addresses these topics.

The LLC layer provides a connectionless service to the 3a layer. We have a technical complication that increases the complexity of the interface; the remaining lifetime (see section 4.4 below) must be carried as 3a PCI[17], but the service time delay in the LLC service provider is not known at the time when the packet is taken under service. Our solution to this problem is to split the *LLC-Data.request* in two phases, see Figure 3.3. Phase 1 delivers the information required by the local LLC/MAC entities to start serving the packet, see Table 4.1. Phase 2 sends the payload downwards from the network layer just before the packet goes on the air.

[17] LLC applies segmentation of LLC SDUs and to carry the remaining lifetime below 3a complicates the lower layers.

Parameter	request phase 1	request phase 2	indication	confirm
Priority	M			
Local GUID	M			M
LlcAddress Destination	M			
Remaining lifetime [sec]	M			
QoS: useARQ	M		M(=)	
RelayList	M			
ccList	M			
Payload		M	M(=)	
Measured delivery delay [sec]			M	
sduServedAt [local node time]				M

Table 4.1 LLC-Data service primitives.

Below we explain the usage of these parameters:

Priority: Used by MAC entity to select a scheduling priority.
Local GUID: References the 3a packet under service. LLC includes this identifier when the LLC entity requests the payload later.
Destination (next hop address): The 3a routing function selects the LLC entity to serve this packet.
Remaining lifetime: This is the packet lifetime and is required by the LLC/MAC packet lifetime control functions.
QoS::useARQ: Applies to unicast packets only. The 3a entity may enable/disable LLC ARQ on a per packet basis.
RelayList: Applies to multicast packets only. The 3a entity determines the node set to be used as relays.
ccList: Applies to multicast packets only. The 3a entity selects the node set which shall respond with a MAC CC PDU.
Payload: This is the packet sent transparently to the destination(s).
Measured delivery delay: This delay is used by the receiving 3a entity to update the remaining lifetime at the receiving side. The sending 3a entity compensates for the delay up to the first bit sent on the air, while the receiving side compensates for the other delays such as transmission burst lengths and retransmission(s), see section 4.5.
sduServedAt: The time instance when the MAC entity has sent the last bit of the packet on the air.

LLC-Status service primitives provide for the coordination between LLC service provider and the 3a entity, see Table 4.2. Below we explain the usage of these parameters:

Local GUID: This parameter is only valid when the signal is *sendPayload*. It is a packet reference number used over the 3a/LLC-interface.
QStatus: PECN use this parameter to inform about the transit queue status.

Parameter	request	indication
signal {Xon,Xoff,sendPayload}		M
Local GUID		M
QStatus	M	M(=)

Table 4.2 LLC-Status service primitives.

4.1 Buffering and Interface Flow Control

A radio node must comply with the following rules:

1) The 3a layer entity must not pass a 3a PDU down to the lower layer before the MAC layer entity has finished the MAC scheduling process and is ready to send the first data bit on the air.
2) The LLC layer must issue an Xoff(priority=P3) immediately upon receiving a 3a SDU from the upper layer.

Rule 1 is a result of the 3a PDU lifetime update function as explained in section 4.4. Rule 2 states that the layers below 3a shall serve one-and-only-one packet at a time and that no pre-emption is implemented. Also remember that the MAC protocol cannot handle two or more connections concurrently. To have an efficient multihop flow control function at the 3a layer, a radio node must store only one outgoing packet below layer 3a. However, rule 1 fulfil rule 2 implicitly.

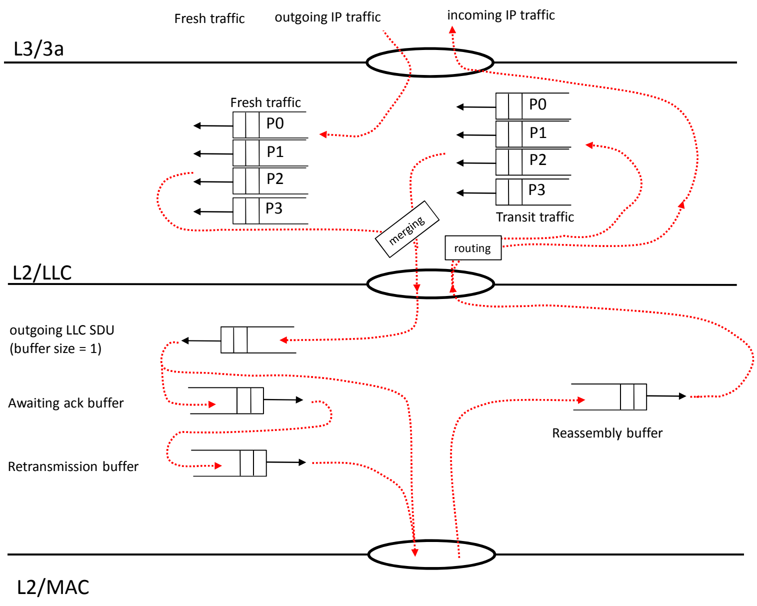

Figure 4.1 Illustration of the node buffer system. An NBWF node shall have one buffer system for each 3a entity and outgoing traffic shall be queued shall be queued on the outgoing link. The simulator has implemented separate buffers for each priority level in the 3a layer. All buffers are served according to their priority. Within the same priority level, the fresh traffic buffer and the transit traffic buffer shall be served on a round robin basis. Incoming traffic over the air interface is split in two by a 3a layer routing function.

4.2 Data Packet Duplicate Filtering

Each IP packet is identified by a unique global unit identifier (GUID) on the air interface. This GUID is composed from the following three fields in the 3a DT PDU PCI:

```
3aDtPdu GUID
{
            int destAddr;   // Exit-node address
            int srcAddr;    // Entry-node address
            int dataUnitId; // A unique identifier within the scope (dest,src)
}
```

For all IP packets arriving over the terminal interface, the 3a layer entity in the entry-node assigns a *dataUnitId*. This dataUnitId must be a unique number and have a validity period equal to the maximum packet lifetime (see section 4.4). After this time period, the entry-node can reuse this number.

If a 3a layer entity receives a 3a DT PDU with a GUID it has seen before, this data packet shall be deleted immediately without any further actions. Duplicate filtering protects against routing errors (loops) as well as multiple forwarding of the same unicast/multicast packet.

4.3 Maintenance of a GUID Cache

Section 4.2 specifies a GUID as a unique identifier (time limited serial number) for each 3a DT PDU (IP packet). When a node receives a 3a DT PDU over the air interface, the node shall extract the GUID and the PDU's remaining lifetime. The GUID shall be inserted in a cache ("database") and a timer shall be started with an expiry time equal to the remaining lifetime. Upon timeout this GUID shall be removed from the cache. The same process shall be applied for each GUID assigned locally, that is, IP traffic arriving over the local terminal interface.

4.4 Lifetime Control

The purpose of the lifetime control function is to guarantee a maximum packet lifetime in the NBWF core network, typically set to 60 seconds. (This value must be dimensioned according to the radio transmission capacity and the *dataUnitId* length to avoid running out of numbers).

As an aspect of lifetime control, the 3a layer entity demands a minimum remaining lifetime to serve a packet. This is a fixed threshold value, typically set to 15 seconds. With a maximum lifetime of 60 seconds, a packet is guaranteed to be deleted if it is delayed more than 45 seconds in the fresh traffic input buffer (delay at layer 3a in the edge-node). However, 45 seconds is a long delay and the network is certainly in a saturation state.

4.5 LLC Service Time Delay Measurement

This is an LLC layer service provided to the 3a layer for calculating the remaining lifetime. It is impossible to predict the LLC SDU delivery delay at the sending side for some reasons: 1) The MAC scheduling delay is a random number. 2) The MAC SDU segment size is a random number and hence, the number of LLC PDUs to send is a random number. 3) One or more segments of a unicast LLC SDU may need retransmission due to background noise.

We select a solution where the originating 3a entity updates the remaining lifetime field in the 3a PCI when it sends the IP payload down to LLC; cf. the phase 2 row in Table 4.1. The receiving LLC entity receives the last bit of the data first segment at time instance t_4 in Figure 4.2. It knows the transmission speed and byte length, and calculates the point in time where the remote radio turned to transmission mode (t_3). Another solution is to use the CAS as a time reference. The B-side (receiver) calculates the LLC SDU delivery delay as the difference between t_5 and t_3.

If segment 1 is lost and segment 2 is received then t_3 is set to the time instance when segment 2 is sent. This gives an underestimation of the delivery delay ($t_5 - t_3$). This is indifferent for packets not using ARQ because they are deleted by the receiving LLC entity when one or more segments are missing. Lost segments in packets using ARQ will be retransmitted by the LLC ARQ protocol and layer 3a may receive LLC SDUs where the delivery delay is underestimated. The consequence is that a packet may live longer than the maximum lifetime. This is not considered to be a problem since: 1) The underestimation of the delivery delay is small compared to the maximum lifetime 2) If the segment loss rate is high, the retransmission period will also be long compared to the underestimation.

The NBWF simulator has no processing delay at layer 3a and $t_3 - t_2 = 0$. This is impossible to achieve in a real system. However, it is possible to issue the *sendPayload*-signal at a fixed point in time before the time instance t_3 (the radio's point-of-no-return) since we have no random component between these two time instances. If the time gap is large enough to allow incoming packets, new functionality must be introduced to "send" the payload back to the 3a layer.

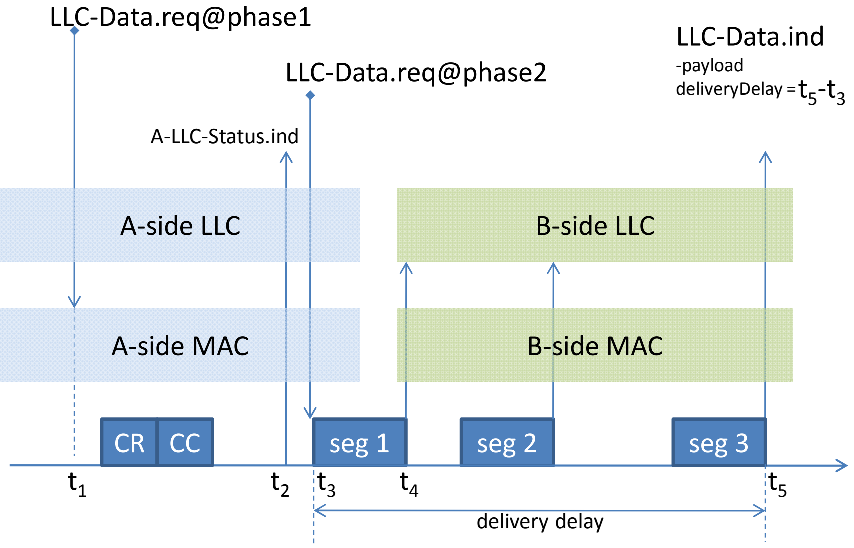

Figure 4.2 Time-sequence diagram for serving a multisegment LLC SDU.

5 ChainN3 Networks

The previous chapters have described what challenges we meet in conjunction with unicast/multicast-forwarding in multihop ad-hoc networks and some solutions have been proposed. The purpose of this chapter is to specify a simple scenario where it is easy to get a basic insight into the behaviour of the pacing and PECN protocols. Any multihop network cannot be simpler than a chainN3 network. A chainN3 network refers to a three node chain topology. We start with this simple topology where it is easier to discover implementation errors and increases the complexity later as we get an understanding on how these flow control protocols behave.

The most important performance metric for the IP clients is the throughput/delay-performance, and all the simulation experiments in this document measures the throughput and the end-to-end delay[18]. We have earlier expressed that a flow control protocol shall press the saturation back to the entry-node. Hence it is important to measure both the fresh traffic queue size and the transit queue size. The Measured Forward Delay (MFD) probe may indicate traffic conditions seen by a relaying node and this probe is also activated. NBWF uses a connection setup procedure and the p_{CC} -probe expresses directly how efficient the connection setup phase operates; a low p_{CC} -value tells us that it is difficult to establish a connection.

[18] The probe which measures throughput terminates a simulation run when the accuracy is better than 10% at a confidence level of 90%. No confidence control is applied to the other probes.

A real network must guarantee a maximum packet lifetime. For NBWF, the maximum packet lifetime is set to $L_{\max} = 60$ seconds[19]. Below we give an overview of the other scenario parameters. They express default values. If other values are used, the text expresses the new values in use.

Fixed network parameters:

Number of voice relays: 0
Admission control threshold q_{max}: 10 packets
3aLayer::L_{min}: 15 seconds
LlcLayer::L_{min}: 10 seconds
Transit queue buffer space: infinite
Pathloss: Fixed 10dB (low loss since the network shall operate under excellent SNR conditions)

Fixed traffic parameters:

Packet arrival distribution: Exponential
Payload length: Fixed size 500 bytes
Priority distribution: Single level at priority P2
ARQ: Not in use

The radio parameters are specified in [9, table 1.1]. All the experiments in this document are conducted on networks operating in an excellent radio environment. In the scenarios simulated, the network is configured to carry IP traffic on 6 of the 9 slots, see Figure 5.1.

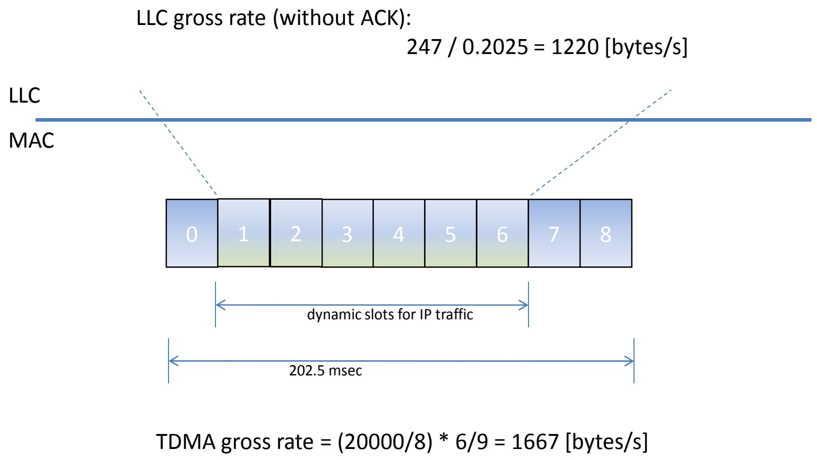
LLC gross rate (without ACK):
247 / 0.2025 = 1220 [bytes/s]
LLC
MAC
0
1
2
3
4
5
6
7
8
dynamic slots for IP traffic
202.5 msec
TDMA gross rate = (20000/8) * 6/9 = 1667 [bytes/s]

Figure 5.1 IP traffic can use the slots numbered 1 to 6 only since the TDMA allocation scheme reserves slot 0 for multicast voice and 2 slots for other application such as network management and routing. The LLC gross rate is calculated from the overhead we currently have in the NBWF simulator.

[19] The NBWF core protocols demand a limit for reusing unique identifiers.

Our choice of using mode N1 (20kbps) doesn't affect the conclusions since the focus is the shape and the relative magnitude of the performance plots, and not the absolute network throughput capacity.

The m-factor: Section 3.1 specifies the pacing interval function $T_{PI} = 2 \bullet \hat{D}_{fd,B}$ node A shall use when it forwards a packet to relay node B. Node A implements one function for each of its neighbours. This chapter introduces an *m*-factor such that $T_{PI} = m \bullet \hat{D}_{fd,B}$ and uses *m* as a simulation parameter. The *m*-factor has a similar impact in multihop net as the t_u-parameter in an AHA-net [9, equation 3.1]. An increasing *m* enlarges the average pacing delay and the collision rate decreases. By decreasing *m*, the opposite effect is achieved. The optimum *m*-value depends on the traffic conditions and we cannot find a single value which gives maximum performance for all scenarios (topology, packet lengths, etc.).

The q-factor: Section 3.2 specifies q_{trans} as the transit queue size threshold at which a Xoff signal shall be emitted. This chapter sets $q_{trans} = q$ and uses *q* as a simulation parameter.

In multihop networks, the traffic pattern may have nearly the same high impact on the performance as the traffic volume. For this reason, the forthcoming sections study one-way and two-way traffic separately.

Many experiments in this document are based on unicast traffic instead of multicast traffic. To collect and analyse data from unicast traffic is easier. In many of the scenarios simulated, unicast and multicast give the same protocol behaviour even though some performance metrics are different. For example, in a chainN3 network where only the edge-node generates traffic, the multicast throughput is twice the unicast throughput. This in contrast to the end-to-end delay which is different since the multicast sample set includes one-hop links.

5.1 Pacing with One-way Traffic

A multihop network with a single traffic source should be an easy traffic case to solve since the entry-node does only compete with its own relay traffic, see Figure 5.2. By setting the *m*-factor to a very high number, we are guaranteed to have collision free network since any packet reaches the sink node before a new packet is taken under service. How large *m* depends on the traffic level, and as the traffic increases, we must increase *m* to have a network without colliding packets. The simulation variables in this section are the offered traffic and the *m*-factor taken from the set *{0, 0.1, 1, 2, 3}*. With *m=0*, a zero pacing delay is added and hence the flow control mechanism is disabled.

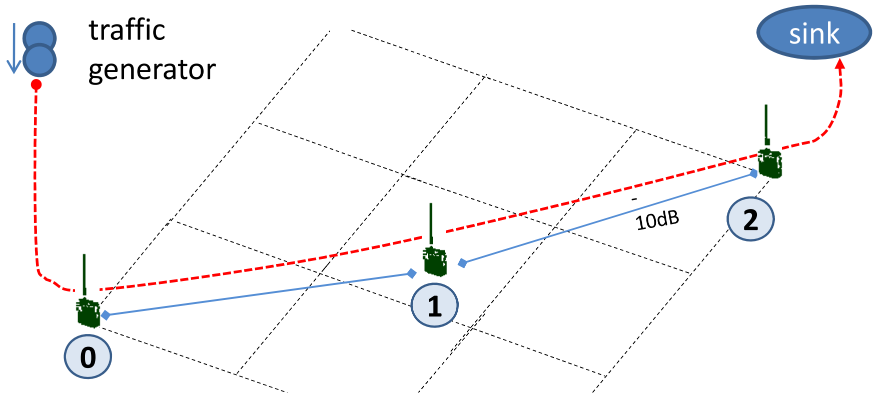

Figure 5.2 A chainN3 network where node 0 is the entry node and node 2 is the sink node.

Figure 5.3 verifies that the pacing protocol has an effect on the traffic but gives no throughput capacity enhancement. However, $m \in \{0.1, 1\}$ seems to give a lower end-to-end delay under saturation.

What is the expected throughput capacity? To answer this question, we simulated an AHAn2 network with two-way traffic (two-way since node 0 and node 1 compete). This AHAn2 network had 900 bytes/s maximum throughput and 800 bytes/s throughput capacity. The throughput of the chainN3-network cannot be larger than one half of the AHAn2 throughput. Two arrows mark these upper bounds in the throughput plot and show that the NBWF protocols have excellent efficiency in this multihop scenario.

As shown in Figure 5.4, the transit queue remains short for all load levels. Even the small *m=0.1* leads to a transit queue near zero which means that the relay usually has served the relay packet before the next arrives. p_{CC} is close to one (Figure 5.5) for all *m* and load levels, and we conclude that the connection setup phase works efficiently in this scenario.

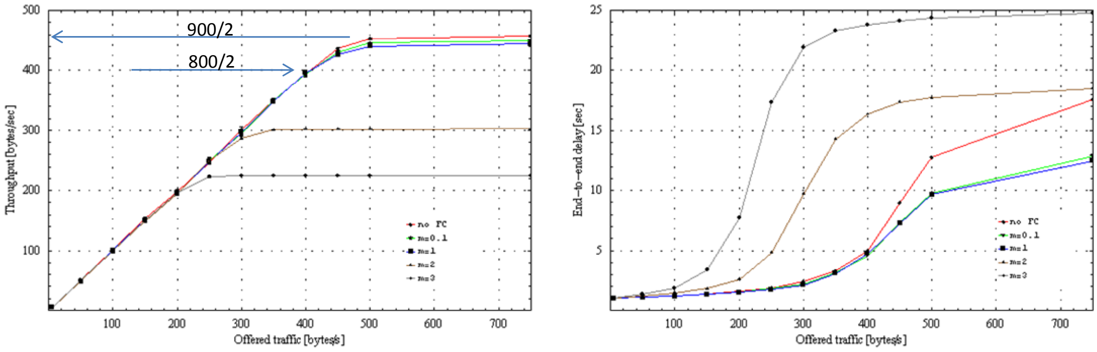

Figure 5.3 Sensitivity of m on throughput and end-to-end delay performance. "no FC" means no flow control (m=0) (chainN3a1).

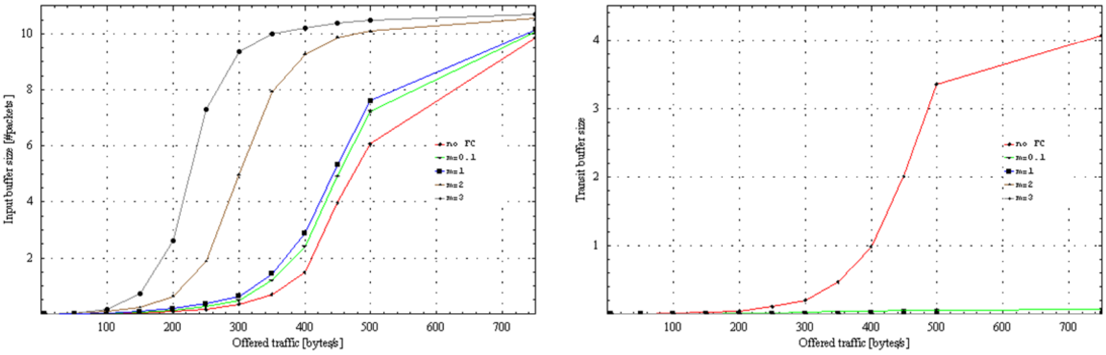

Figure 5.4 Fresh traffic queue size and transit buffer queue size [number of packets].

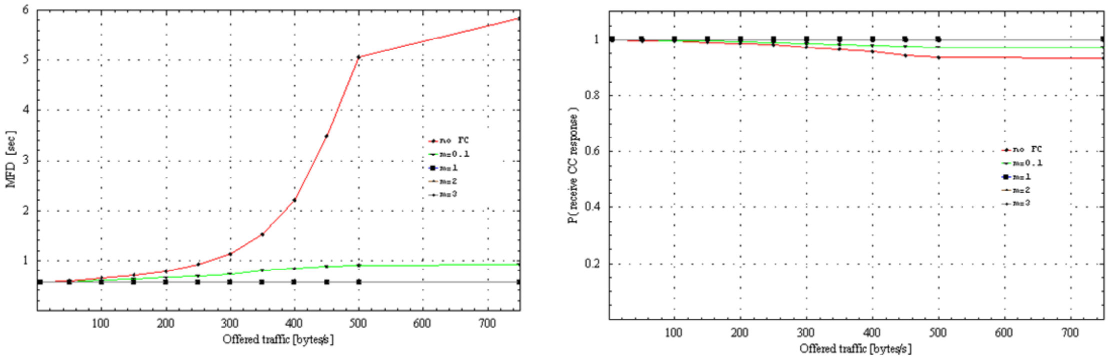

Figure 5.5 MFD and p_{CC} (chainN3a1).

5.2 Pacing with Two-way Traffic

We anticipated a low collision rate in the previous scenario, but expect a significant collision rate when the traffic becomes two-way, see Figure 5.6. Here the MAC CR PDUs sent by the two edge-nodes collide frequently at node 1, depending on the packet arrival rate. The pacing protocol is not designed to solve this problem. However, by increasing *m*, fewer packets become available to MAC and the collision rate is expected to drop.

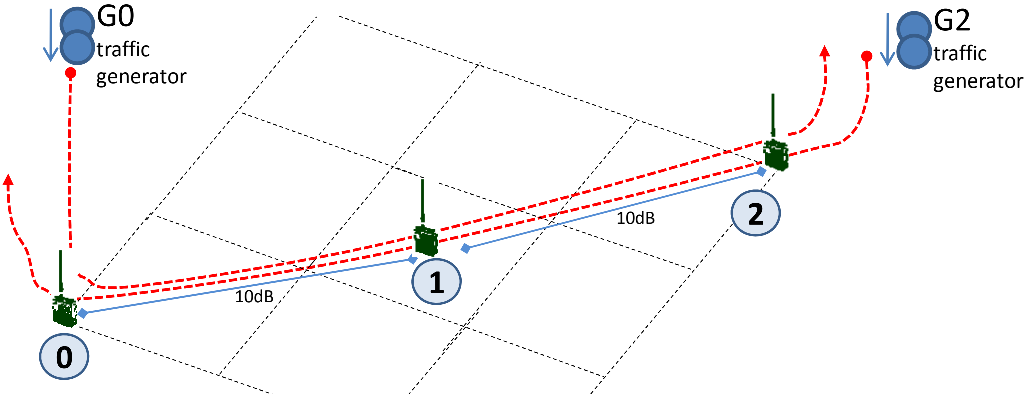

Figure 5.6 A chainN3 network where the nodes 0 and 2 both operate as source and sink nodes. The traffic generator G0 = G2. Node 1 is a relay node and does not generate fresh traffic.

Figure 5.7 verifies that the throughput capacity is very close to the performance of the one-way case presented earlier, but the maximum throughput is slightly lower. We observe a large deviation in the optimum *m*-value. While the one-way network benefits from a zero or small *m*, the two-way network needs a much larger *m*. *m* is a constant and a single value cannot optimise the throughput for both traffic conditions.

From Figure 5.8 we see that we can move the buffering from the transit queues to the input buffers by increasing *m*. Plots of more interest are the p_{CC}-plots in Figure 5.9. For small *m*-values, p_{CC} drops dramatically when going from the one-way to the two-way traffic pattern. A low p_{CC} means that much transmission capacity is consumed by the connection setup process. From the throughput plot we conclude that $m \in \{1, 2, 3\}$ is the best choice, and by selecting $m \geq 2$, p_{CC} gets a more favourable value. The last plot included is the pacing delay versus the offered traffic, see Figure 5.10. At the saturation point, the packet inter arrival time is 1.25 seconds ($500/400$). As indicated by the figure, the pacing delay should be in the same order to give proper throughput.

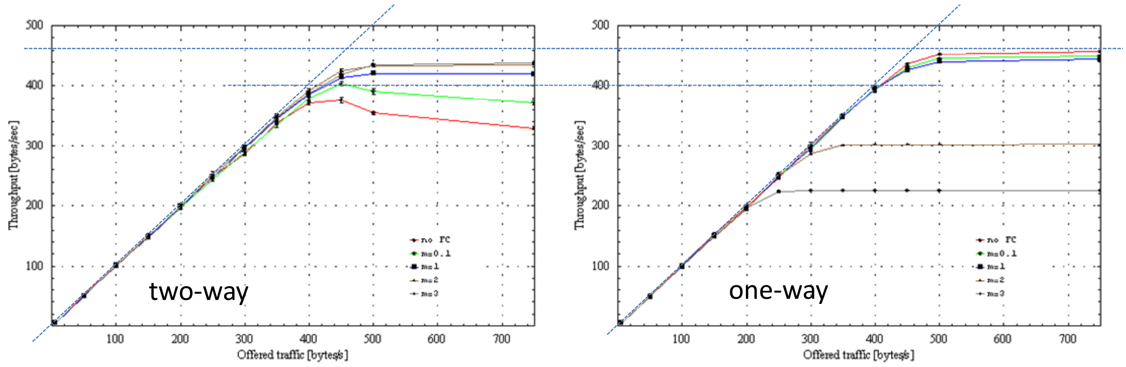

Figure 5.7 Throughput performance comparison (chainN3a2).

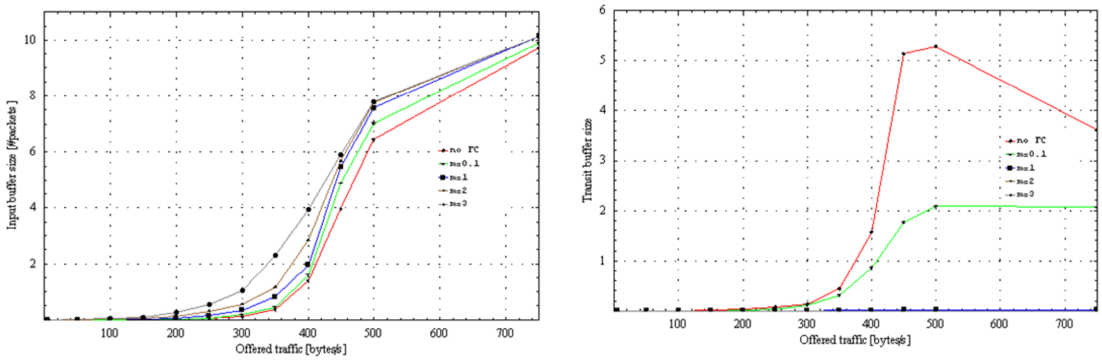

Figure 5.8 Fresh traffic queue size and transit buffer queue size [number of packets].

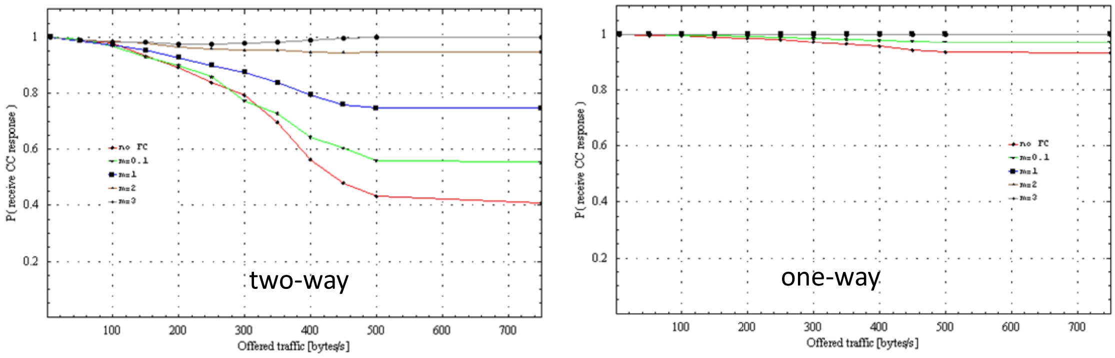

Figure 5.9 *p_{CC} performance comparison.*

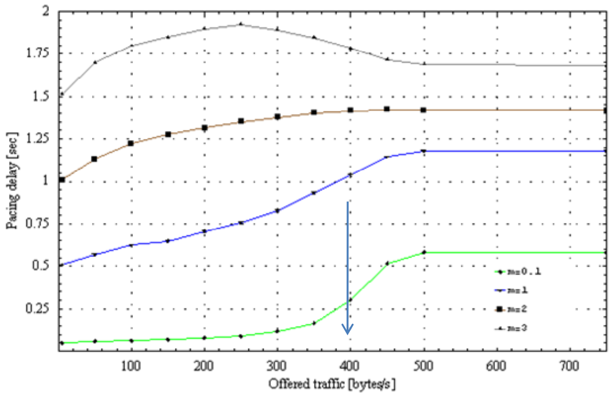

Figure 5.10 Measured pacing delay vs. offered traffic. The vertical arrow indicates the saturation point.

5.3 PECN with One-way Traffic

This section repeats the experiments in section 5.1with pacing replaced by PECN. PECN is a competitor to pacing and we use the latter as a reference when we discuss the results. Remember that the simulation parameter for pacing is the *m*-factor, while PECN protocol is controlled by the *q*-factor; the transit queue threshold parameter Q_{trans} as explained in section 3.2. Figure 5.11 indicates that both NLFC protocols have similar performance and changing of *q*-values has no impact on the PECN throughput performance. They also have approximately identical link delay statistics for the {*m,q*}-values which maximise the throughput. The only interesting statistics to comment on is the transit buffer queue size in Figure 5.12, which shows that the *q*-factor gives a better chance of setting the buffer size. However, this is not an important property in the current network.

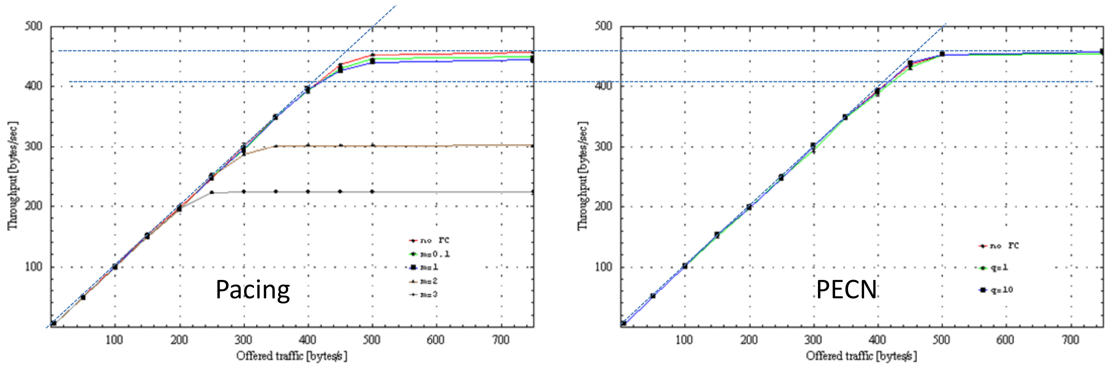

Figure 5.11 Simulated throughput comparison between pacing and PECN.

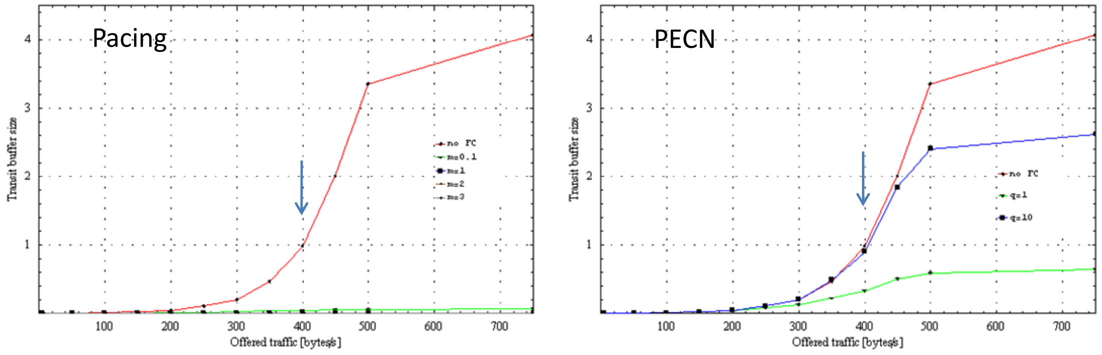

Figure 5.12 Simulated transit buffer size comparison.

5.4 PECN with Two-way Traffic

This section repeats the experiments in section 5.2 for PECN. Figure 5.13 indicates 5% (390 vs. 370 bytes/s) lower throughput for the PECN protocol. The *q*-parameter has nearly no effect on the throughput or the p_{CC} in Figure 5.14. This in contrast to the *m*-parameter used by the pacing protocol, which is able to provide a much higher p_{CC}. We conclude that the connection setup process operates with low efficiency under PECN, but the transit queue size in Figure 5.15 is affected by the *q*-factor.

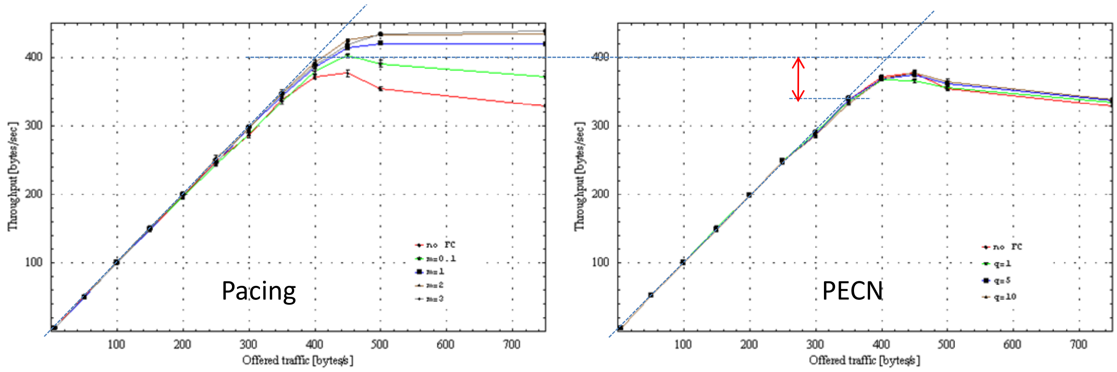

Figure 5.13 Throughput comparison between pacing and PECN.

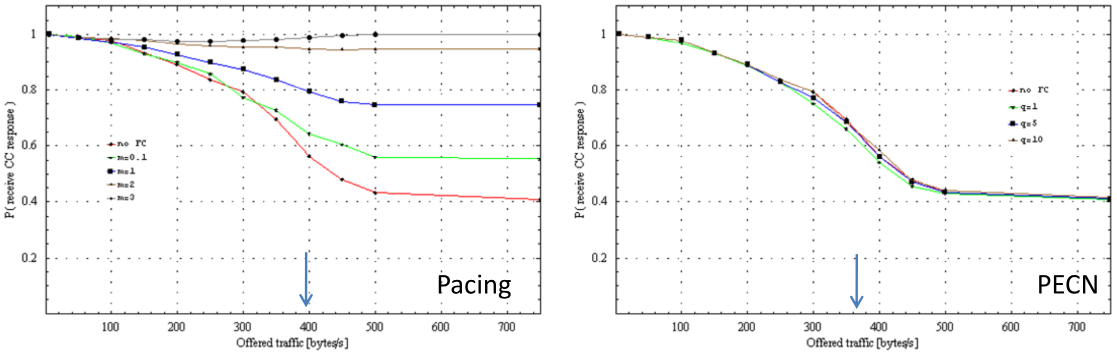

Figure 5.14 Simulated p_{CC} comparison.

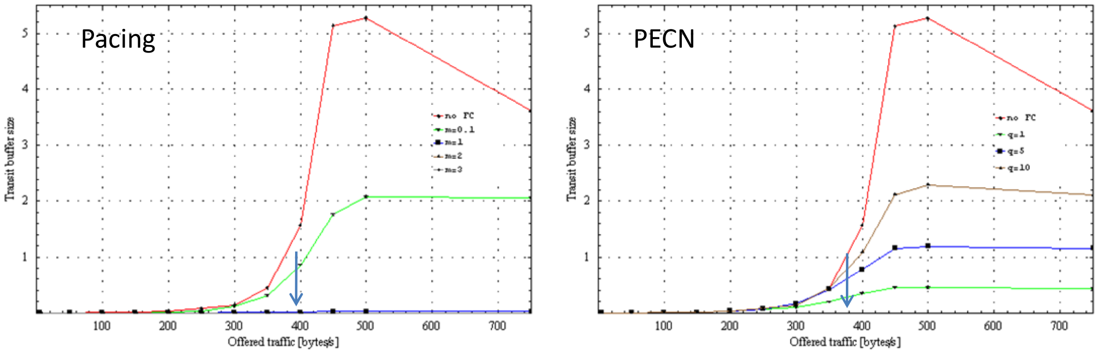

Figure 5.15 Simulated transit buffer size.

5.5 Conclusions and Remarks

In this chapter, we have studied pacing and PECN in the simplest multihop topology possible. We did experience reasonable transit queue sizes under all traffic conditions even with the flow control disabled. The major traffic problem in the scenarios was a low p_{CC}, indicating that the connection setup process consumed significant transmission capacity. A pacing protocol handles this situation better since it measures the packet forwarding delay and adds a backoff delay which again reduces the MAC CR PDU collision rate as a side effect. The *m*-parameter showed to improve the p_{CC}, while the *q*-parameter had nearly no impact. However, remember that the {*m*,*q*}-parameters are not implemented to regulate p_{CC}. Essentially, pacing is the best protocol for the scenarios in this chapter. Also note that the PECN protocol operates in a beneficial environment because it has faster access to the shared broadcast slot (TDMA super frame) in a small network than in a network with many nodes.

6 ChainN4 Networks

This chapter repeats the experiments from chapter 5 for a chainN4 network. A chainN4 refers to the four node chain topology in Figure 6.1. All other parameters remain unchanged and the subsections below are sequenced as in the previous chapter.

6.1 Pacing with One-way Traffic

Figure 6.1 specifies the network topology and traffic pattern for this section. The AHAn2 network referenced in the previous chapter had the maximum throughput capacity 900 bytes/s. Under the assumption that the hop-by-hop delay in the chainN4 is the same as the link delay in the AHAn2 network, the maximum throughput capacity of the chainN4 network must be lower than 900/3=300 bytes/s. The simulated throughput performance in Figure 6.2 estimates the maximum throughput to be 280 bytes/s, which is 6.7% reduction compared to the upper limit (300 bytes/s). From the throughput plot we should use *m=0.1*. However, Figure 6.5 illustrates how difficult it is to send a MAC CR PDU from node 0 to node 1, while the link 1→2 operates at high p_{CC}–values. We have omitted the p_{CC}-plot for the link 2→3 since it is very similar to the link 1→2 . The MAC protocol works efficiently on the link 1→2 because node 1 has no hidden-nodes since node 3 does not send CR PDUs. It is the MAC CR PDUs from node 0 and node 2 that collide at node 1 which causes low p_{CC}-values.

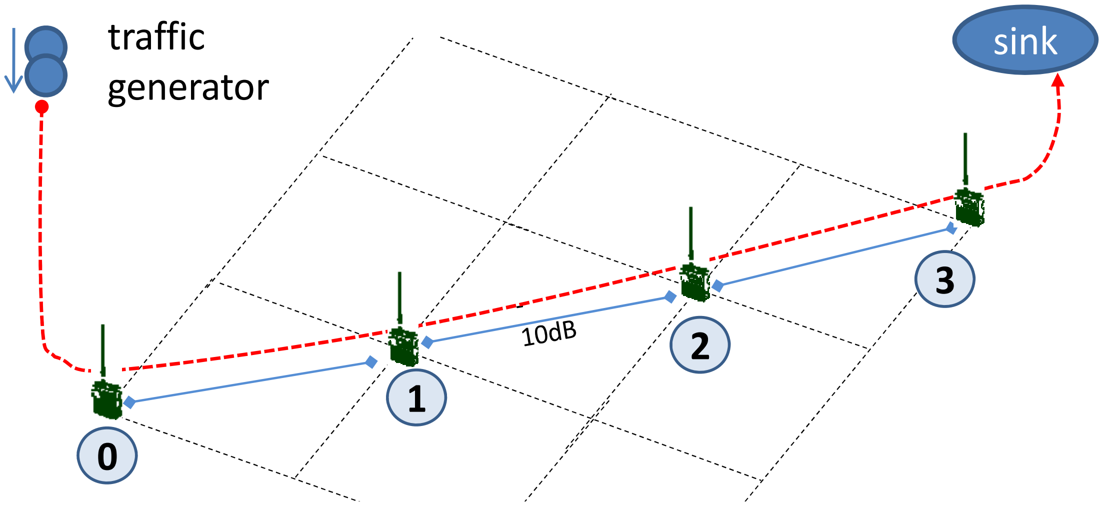

Figure 6.1 A chainN4-network with a single traffic generator (chainN4a1).

Observe from Figure 6.4 that the *m*-parameter has a significant impact on the transit traffic queue size; even a small *m* gives a large drop of the queue size. The link 1→2 has the longest queue[20] but the magnitude is acceptable even for *m=0* (i.e., pacing disabled).

[20] The simulator uses outbound queuing which means that incoming relay traffic on the link 0→1 is stored on the outgoing link 1→2.

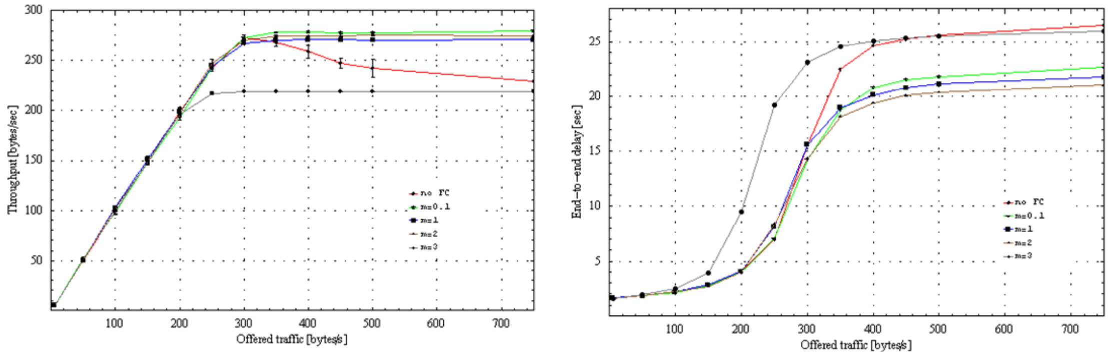

Figure 6.2 Throughput and delay performance (chainN4a1).

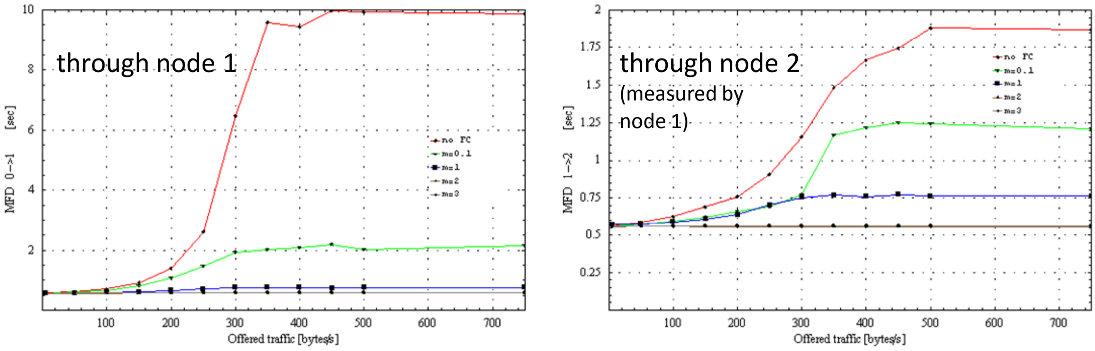

Figure 6.3 Simulated MFDs (chainN4a1). Note the different y-scales on the figures.

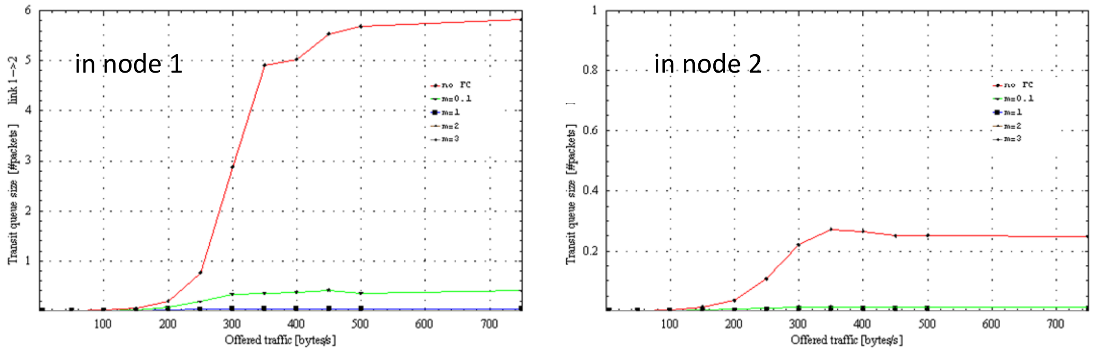

Figure 6.4 Simulated transit queue buffer sizes.

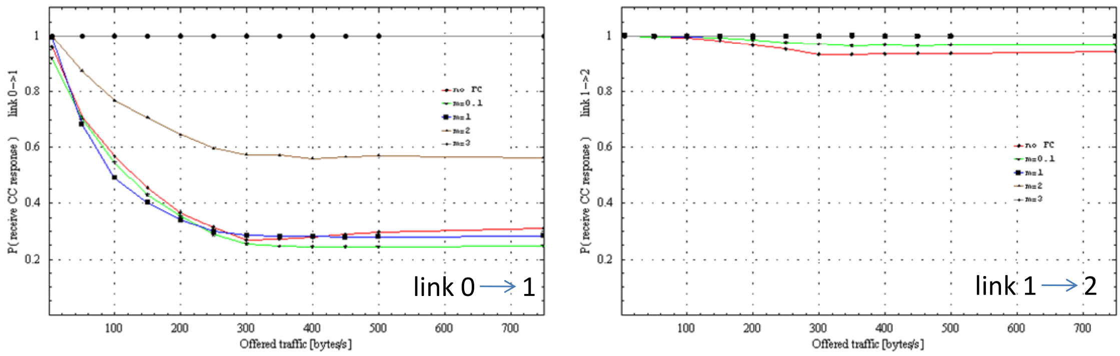

Figure 6.5 Simulated p_{CC}. If m=3 is used, p_{CC} =1 and we have a collision free network.

6.2 Pacing with Two-way Traffic

Two traffic generators are now activated in the chainN4 network, see Figure 6.6. Figure 6.7 shows a large degradation of the throughout performance when the traffic pattern switches from one-way to two-way; the throughput capacity is reduced from 240 to 80 bytes/s (67% reduction). This throughput drop is caused by the connection establishment problem on the link 1→2, see Figure 6.8. With one-way traffic, p_{CC} was better than 0.95 but drops below 0.2. Now we have two links where the signalling traffic consumes much transmission capacity.

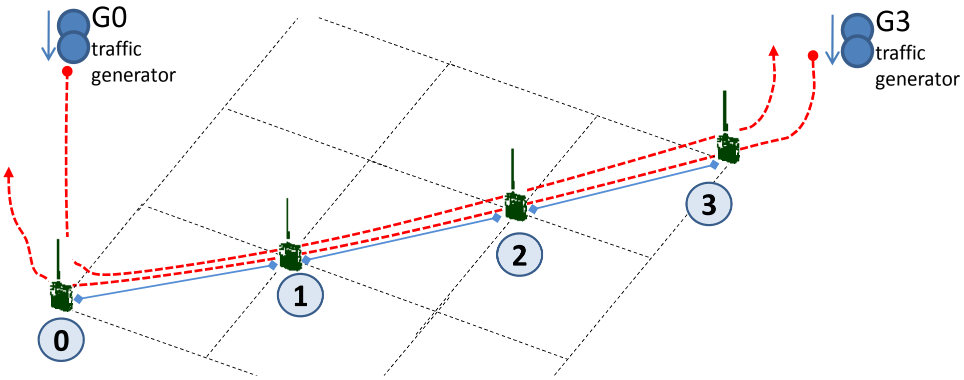

Figure 6.6 A chainN4 network where the edge-nodes 0 and 3 both have a role as source and sink nodes. G0 = G3.

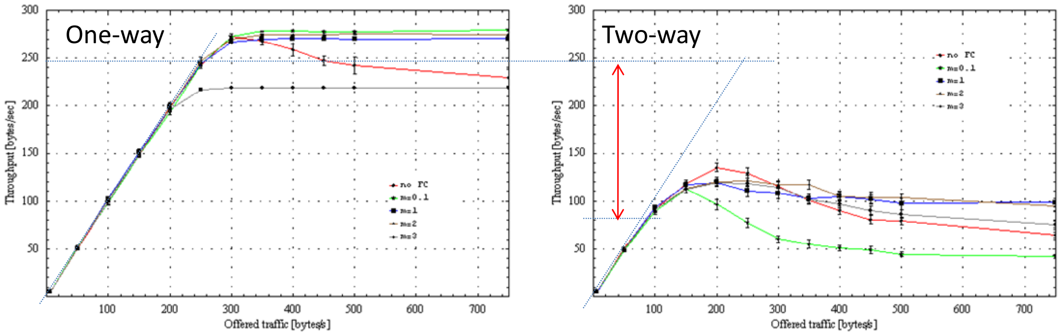

Figure 6.7 Throughput comparison for pacing in a chaninN4 network (chainN4a2).

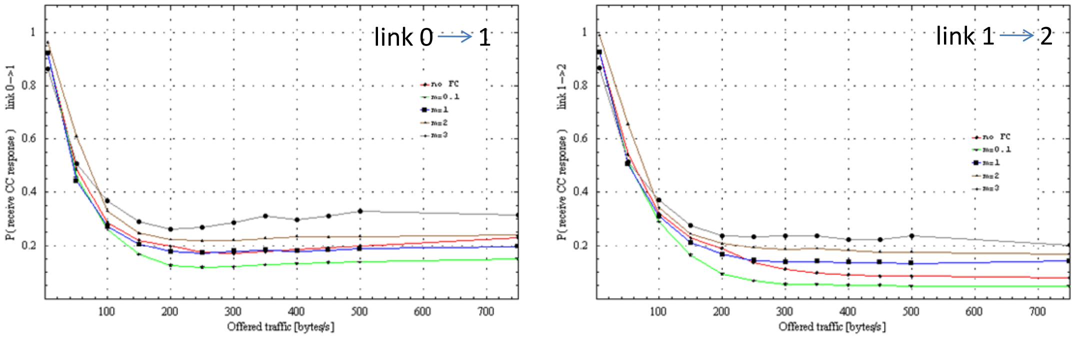

Figure 6.8 Simulated p_{CC}.

6.3 PECN with One-way Traffic

This section repeats the experiments in section 6.1 with pacing replaced by PECN. PECN has the same throughput performance and the comments given in section 5.3 for the chainN3-case are also valid for the chainN4 network.

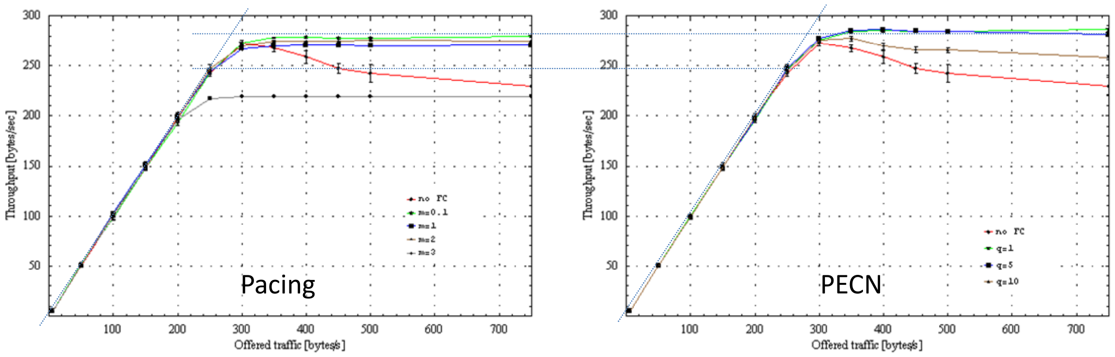

Figure 6.9 Throughput comparison between pacing and PECN in a chainN4 network with one-way traffic.

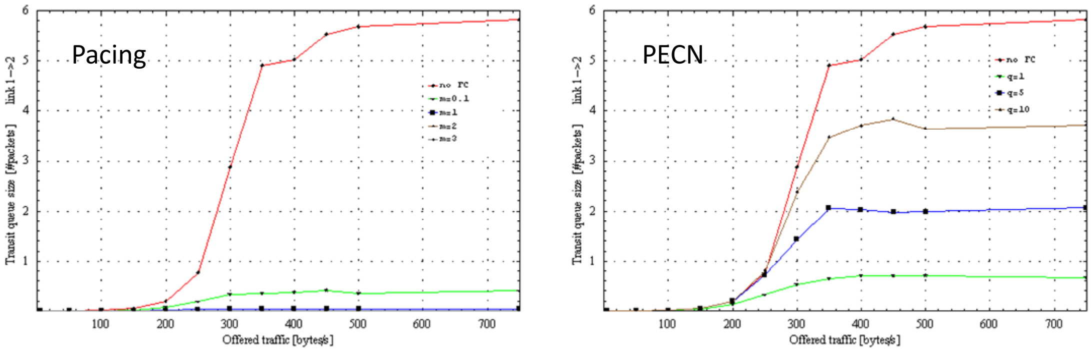

Figure 6.10 Transit queue size [#packets] on the link from node 1 to 2.

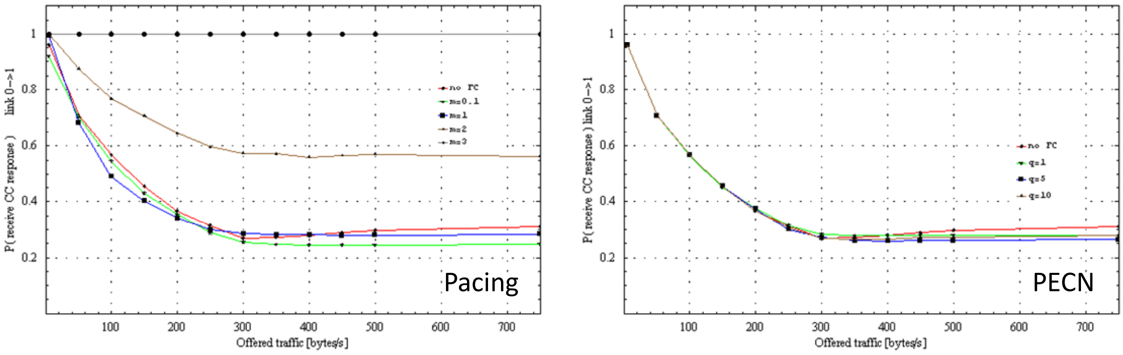

Figure 6.11 p_{CC} on the link from node 1 to 2.

6.4 PECN with Two-way Traffic

In this section, we repeat the experiments in section 6.2 for PECN. Figure 6.12 verifies that also PECN experiences a high throughput capacity drop when both edge-nodes generate traffic, but the throughput capacity and the maximum throughput have the same magnitude as the pacing protocol. Under maximum load, the pacing protocol may deliver 35% higher throughput (100 vs. 65 bytes/s).

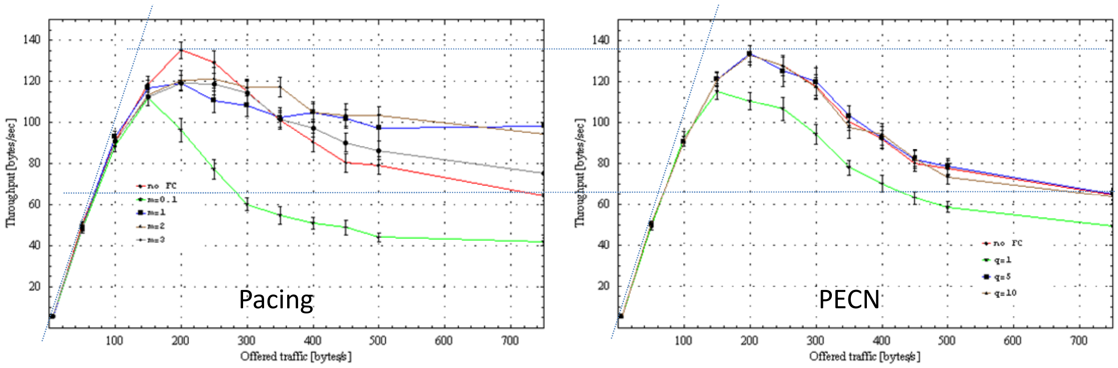

Figure 6.12 Throughput comparison between pacing and PECN.

6.5 Conclusions and Remarks

As expected, when we switched from a chainN3 topology to a chainN4 here, the hidden-node problem became more dominant. However, the performance degradation was moderate in the one-way traffic scenario; 6.7% drop in throughput capacity. Pacing and PECN had identical throughput capacity.

Dramatic performance degradation occurred when we turned on two-way traffic. Both flow control protocols experienced 67% reduction in the throughput capacity. The main cause of degradation is the interference between the MAC CR PDUs because the LLC connection setup process persistently generates new PDUs upon MAC connection failure. Flow control is conducted at layer 3, so the NLFC cannot decide the MAC CR PDU interspace time delay, but the LLC layer can. MAC sends only one CR PDU at a time, and issues a local loss event to the LLC layer each time a CC PDU is not returned by a peer-entity within a specific time limit. Then the LLC entity issues a new *MAC-Connect.request*. In the same manner as the adaptive MAC scheduling function (described in [9, chapter 3]) increases its random delay during high load periods, the LLC entity can insert an adaptive random delay when the entity applies the recovery function after a connection setup failure. This is the subject for the next chapter.

The two flow control protocols are designed to protect the transit buffer from overflow and both fulfil this task. In addition, pacing is designed to reduce/prevent interference between its own data packets during the packet forwarding process.

When a node in a PECN network signals Xoff, its 3a entity will not feed the lower layer with a new packet, but any packet that already is under service by the LLC is served as usual. This delay comes in addition to the interspace delay between the periodic reports, which are only 0.8 seconds here since the network has four nodes only.

PECN operates in favourable scenarios in the multihop network simulated up to now since the interspace delay between the periodic reports are short; only 0.8 seconds in the network with four nodes. Remember also that when a node in a PECN network receives a Xoff from its neighbourhood, its 3a entity will not feed the lower layer with a new packet, but a packet that already is under service[21] by the LLC is served as usual. Also this increases the latency time of the PECN flow control process, but the LLC service time is short in the networks considered here; 0.6 seconds under low load and typically 2.4 seconds during high traffic periods. The LLC service time increases when the number of nodes increases because more nodes compete for channel access.

In chapter 5 we argued for using unicast traffic instead of multicast in the initial experiments. Unicast and multicast produce different statistics, but under certain conditions we can tell what to expect based on unicast experiments. In a three node chain network, a single multicast packet contributes twice[22] to the network throughput as long as the network operates below its throughput capacity, and the multicast throughput shall be twice the unicast throughput. As shown in Figure 6.13, the simulator produces the expected result. When the network operates above its capacity and starts to lose two-hop traffic (i.e., lifetime expiry events in the transit buffer), the multicast case can achieve higher throughput. A scenario where we shall measure very different throughput for multicast and unicast is the chainN4 network with two-way traffic since few packets survive more than one hop. Figure 6.14 verifies this statement.

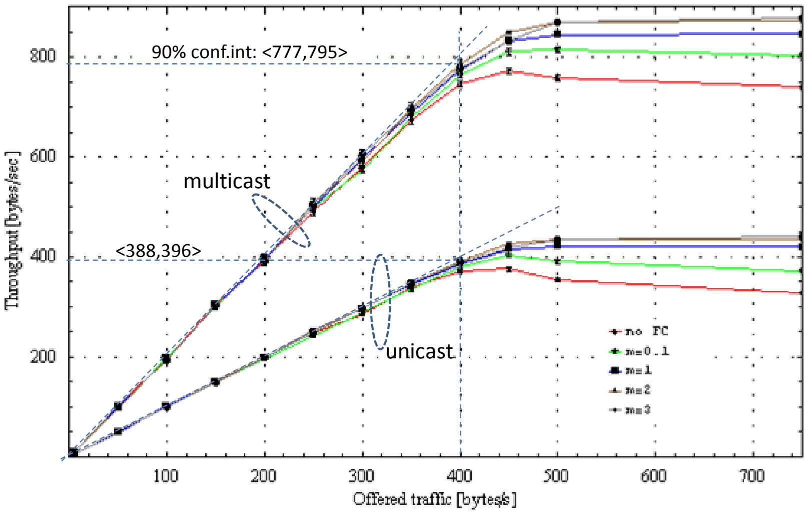

Figure 6.13 Simulated throughput comparison between multicast and unicast in a chainN3 network with two-way traffic.

[21] "Under service" means that the MAC scheduling has started. Any PECN X_{off} signal will not affect this process.

[22] Once at the relay node and once at the exit node.

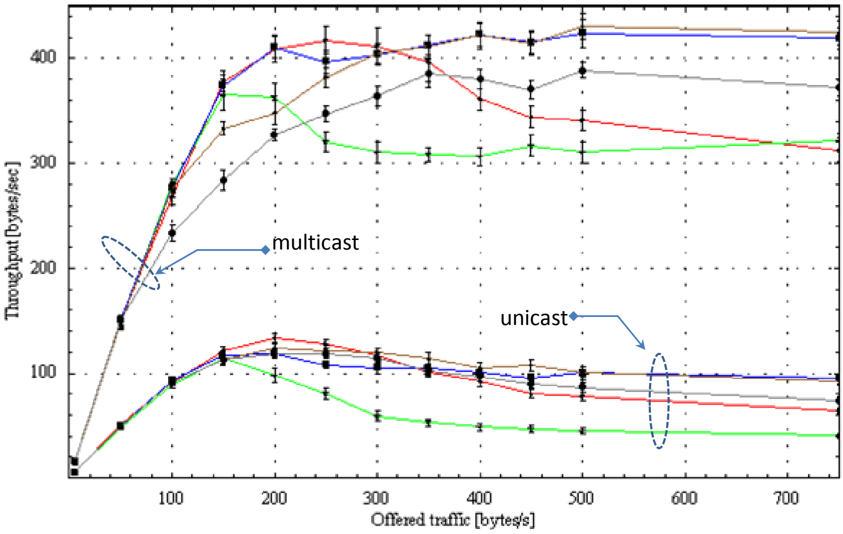

Figure 6.14 Simulated throughput comparison between multicast and unicast in a chainN4 network with two-way traffic.

7 LLC Exponential Backoff

The previous chapter discovered high performance degradation in a chainN4-network in case of two-way traffic and argued for adding a random backoff delay at the LLC layer each time a CC PDU is lost. This chapter studies the network performance when the following backoff delay function is applied:

$$randExp(n_{CR}, n_{CR0}, x) = randUniform[0,\ t_u(n_{CR}, n_{CR0}, x)] \qquad (7.1)$$

where

$$t_u(n_{CR}, n_{CR0}, x) = \Bigg[\begin{array}{ll} 0 & for\ n_{CR} \le n_{CR0},\ n_{CR0} \ge 1 \\ (n_{CR} - n_{CR_0} + 1)^x - 1 & for\ n_{CR} \ge n_{CR_0} + 1,\ \ x \ge 0 \end{array} \qquad (7.2)$$

n_{CR} is the number of CR PDUs sent during the MAC connection establishment phase. Figure 7.1 shows how the backoff delay is affected by two constants n_{CR0} and x. $n_{CR} = 2$ means that two CR PDUs are sent; one for the first setup attempt and one for the error recovery. n_{CR0} determines the n_{CR} –threshold, the starting point to add random delays, while the x-constant determines how fast the backoff delay shall increase.

Be aware of that the random delay added by the MAC layer operates independently of the LLC backoff delay, but the LLC backoff process influences the adaptive MAC scheduling process since the process regulates the MAC offered traffic. Small n_{CR0}–values and large x-values gives increased backoff delays.

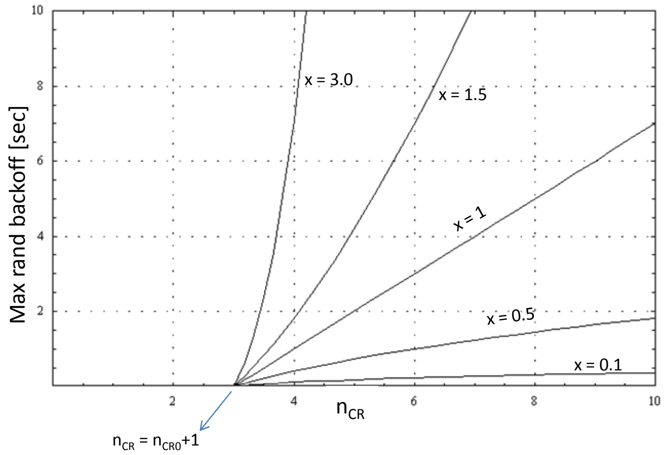

Figure 7.1 *Illustration of the exponential backoff delay parameters. If n_{CR0} is set to one then the second connection setup attempt will be given a random waiting time delay before the LLC entity issues a MAC-Connect.request.*

Compared to section 6.2, Figure 7.2 verifies that the LLC exponential backoff increases the network throughput with 222% (from 65 to 144 bytes/s) for $n_{CR0} = 1, x \in \{1, 1.5\}$. The link with the lowest quality in this chainN4-network is the link $1 \leftrightarrow 2$, and Figure 7.3 shows a very low p_{CC} for $x \in \{0, 0.15\}$ regardless of n_{CR0}. A much better quality may be achieved by setting $x = 3$. However, this adds more idle time than needed since the throughput capacity drops.

Figure 7.4 shows that the throughput is not very sensitive to n_{CR0}. Based on the experiments in this chapter, we should select $x \in \{1, 1.5\}$ and $n_{CR0} = 1$. The latter choice is based on the p_{CC}-plot.

One drawback of adding a backoff delay may be loss of throughput capacity in an AHA-network. The performance drop increases with increasing x and decreasing n_{CR0} since a large random delay is not needed in an AHA-network where the vulnerability period is $t_{v,cas}$*=2.5* msec [9, table 1.1] and not the CR PDU length (≈20 msec). This problem is a subject for the next section.

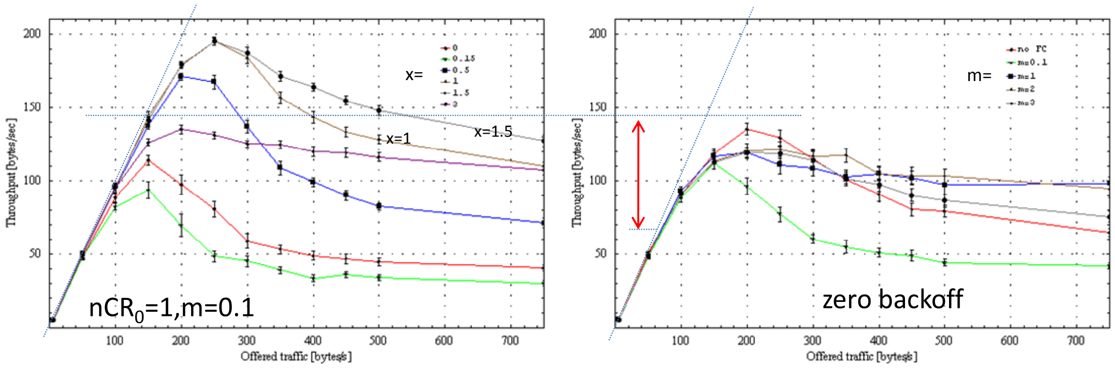

Figure 7.2 *Simulated pacing throughput with x as a parameter. The right plot is a copy of Figure 6.7 (two-way) and uses the pacing parameter m as a parameter.*

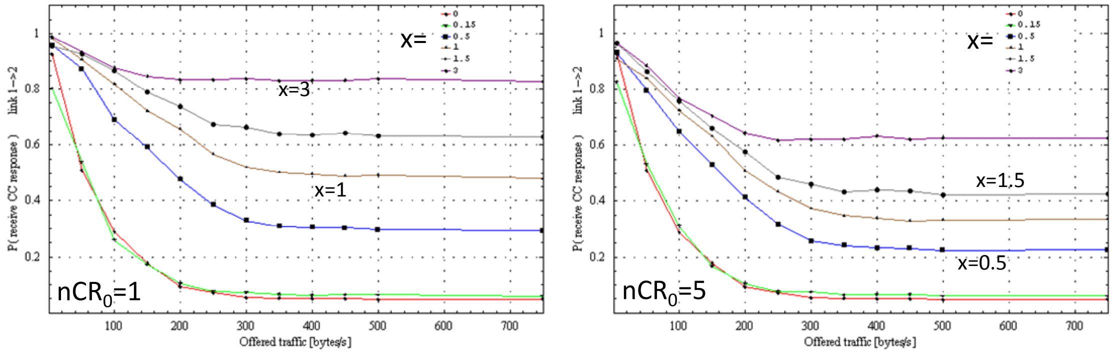

Figure 7.3 p_{CC} on the link from node 1 to node 2.

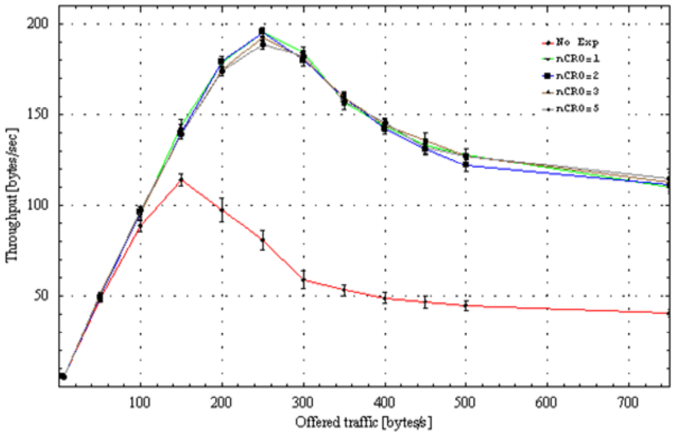

Figure 7.4 Throughput for the pacing protocol for a set of n_{CR0}–values when x = 1 and m = 0.1.

7.1 AHA-Networks

The downside of using LLC exponential backoff delay may be reduced throughput capacity in an AHA-network where an additional backoff is not required since the MAC PSMA protocol prevents collisions. The MAC PSMA protocol parameters shall be optimised to maximize the throughput capacity in an AHA-network and any random delay added by other protocol layers will extend the average channel idle time and hence give degraded throughput performance and increased link delays. This section looks at the negative effects of LLC exponential backoff in an AHA-network.

The influence by LLC exponential backoff is determined by n_{CR0} and *x*. To maintain the MAC protocol efficiency, we expect that n_{CR0} must be set strictly larger than one such that the second connection establishment attempt does not introduce additional random delays. The previous experiments indicate that *x=1* is a good choice. We keep *x* constant at this value and use n_{CR0} as the simulation parameter; $n_{CR0} \in \{1,2,3,5,2000\}$. $n_{CR0} = 2000$ disables the LLC exponential backoff function since the packet lifetime expires before reaching this number of recovery attempts. $n_{CR0} = 1$ is included because this value is expected to give throughput degradation in an AHA-network.

An AHAn10 **single-level** priority network is the first case considered[23]. Figure 7.5 shows that our expectation of having throughput degradation for $n_{CR0} = 1$ is wrong. In the priority {P0,P2}-networks, n_{CR0} has nearly no impact while $n_{CR0} = 1$ gives throughput improvements in the P3-network! The P3-network operates under a high collision rate since the MAC random delay is not dimensioned to handle 10 active nodes. When a CR PDU collision event occurs, the LLC layer adds an additional random delay. This reduces the collision probability to a more optimum value with regard to throughput.

7.2 Background Noise

When an NBWF network operates in a noisy radio environment (jamming and/or background noise), the LLC exponential backoff protocol will act as if this is a hidden-node problem and increases its random delay. This action introduces an additional setup delay without giving any benefit such as reduced CR PDU loss rate.

New experiments are done for the AHAn10-network in a radio environment with 10% background noise, that is, the packet loss probability for any transmission burst is 0.1 regardless of its size. Figure 7.6 verifies that a decreasing n_{CR0} reduces the throughput[24]. A small n_{CR0} value increases the fraction of the CR PDUs that are given an additional random delay. This again increases the channel idle time without decreasing the packet loss probability; the MAC protocol cannot affect the background noise.

As can be seen from Figure 7.6, 10% background noise leads to significant throughput degradation. By selecting $n_{CR0} \geq 5$, the LLC exponential backoff gives modest throughput degradation in noisy radio environments. We repeated all the AHAn10-experiments for a 25-node network, and these results supported our conclusion.

Background noise may, of course, also be present in multihop networks and the LLC exponential backoff will then add an unnecessary delay. The consequence is reduced throughput. We are not able to differentiate between packet loss due to hidden-nodes or background noise. However, the degradation due the exponential backoff (compare p_{noise}=10% for {n_{CR0}=disabled, n_{CR0}=1}) is much lower than the influence from the background noise (compare p_{noise}=0% and {p_{noise}=10%, n_{CR0}=disabled}). To conclude, we do not exclude LLC exponential backoff based on these experiments.

[23] Different priority levels are used but not in the same experiment.

[24] The results for the P2- and P3-networks are not shown.

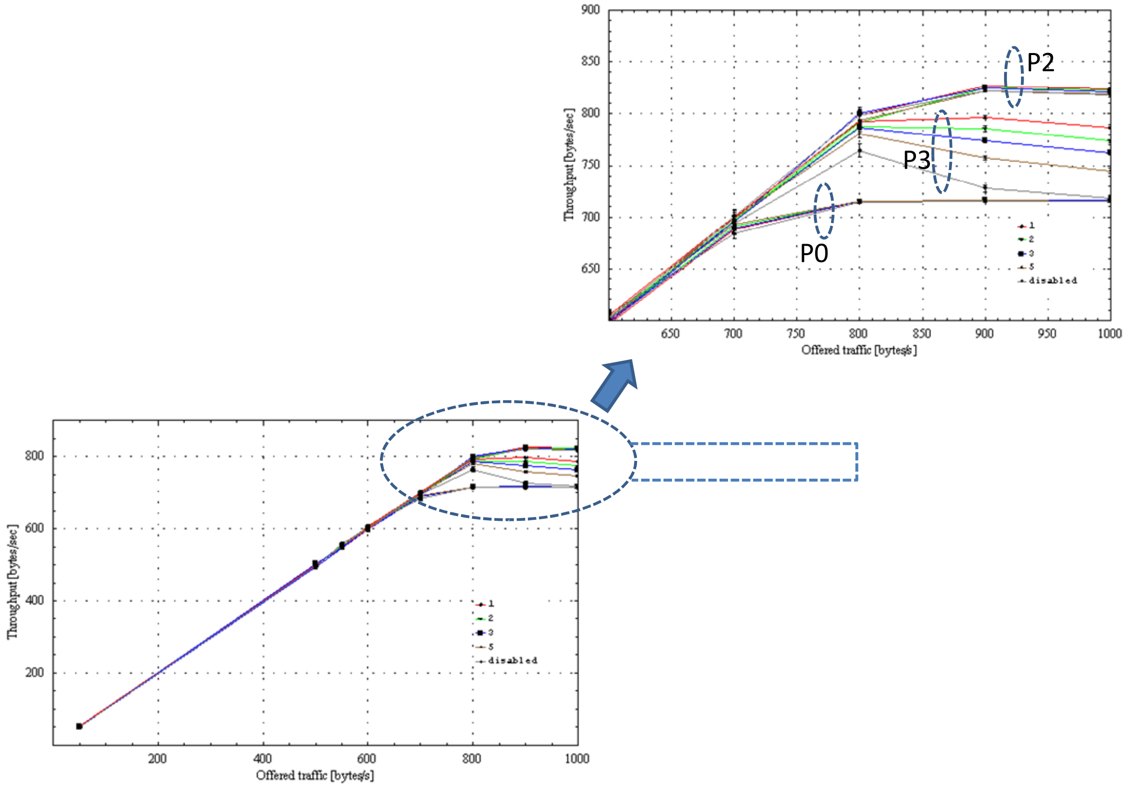

Figure 7.5 AHAn10-throughput for three single-level priority networks sending {P0,P2,P3}-traffic, respectively. n_{CR0}={1,2,3,5,2000(disabled)} and x=1 (simFeb4b). The P2-network gets higher throughput than the P3-network because its MAC protocol operates with a better balance between the channel idle period and the collision rate.

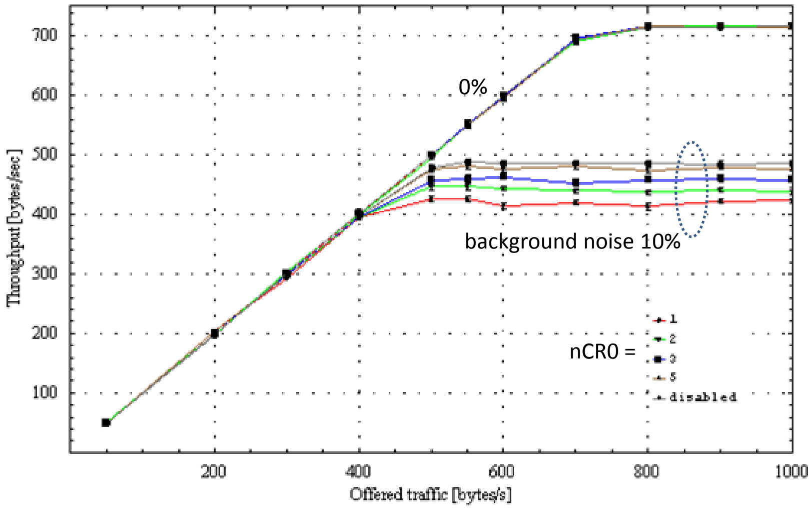

Figure 7.6 P0-throughput in presence of background noise. The curves marked with 0% is the same as for the P0-network in Figure 7.5.

7.3 PECN in a Large Network

When the four node chain operates in a 50-node network, the PECN report period increases from 0.8 to 10.1 seconds, and the flow control protocol reacts slower to changing traffic conditions. The pacing protocol does not have this problem since it does not signal the saturation level explicitly. To test the PECN behaviour in a large network, we added 46 silent nodes[25]. Then we repeated the simulation experiments in section 0 after enabling the LLC exponential backoff protocol with *x=1* and *q=1*.

Figure 7.7 presents the throughput performance. Steady-state simulation shows no significant difference between short and long report periods. We observed the same result for the end-to-end delay and the p_{CC}.

Figure 7.8 verifies that the transit queue length is slightly longer for the longest report period. However, the queue length is short and the difference is insignificant. To conclude, the PECN protocol is surprisingly insensitive to the report period rate. This is possibly due to the fact that the simple network simulated behaves well with flow control disabled.

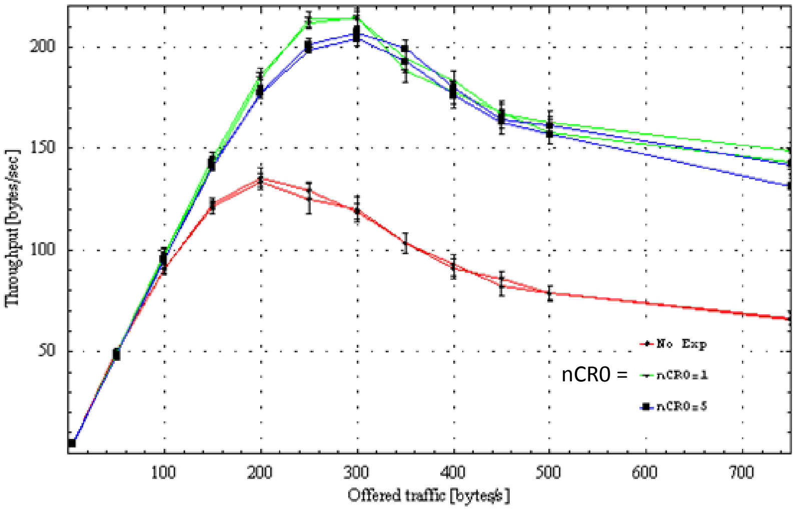

Figure 7.7 Throughput comparison between short and long report period.

[25] The only effect of adding these nodes are longer PECN report periods.

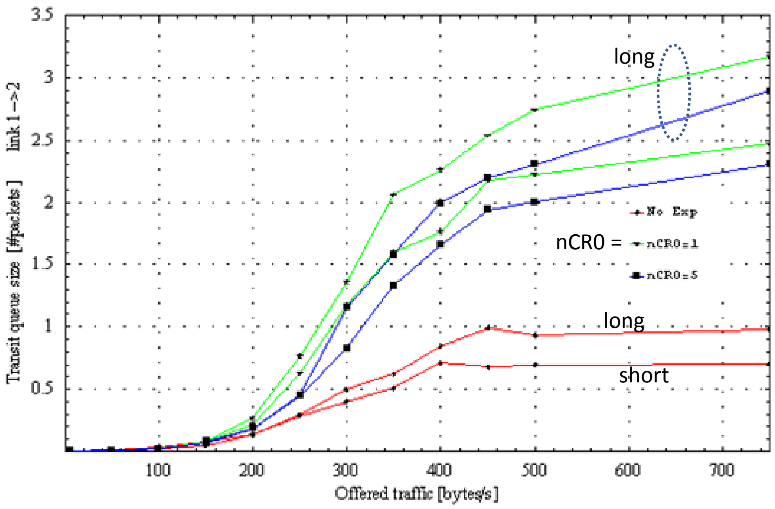

Figure 7.8 Transit queue size [#packets] between short and long report periods. Data applies to the link 1->2.

7.4 Conclusions and Remarks

LLC exponential backoff improved the performance significantly in the scenarios simulated. Proposed parameter values are $(n_{CR0}, x) = (1,1)$ since we did not observe negative effects by using $n_{CR0} = 1$ in AHA-networks. Only a modest throughput drop was detected in a radio environment suffering from background noise.

8 Conclusions and Remarks

The initial scope of this study was the network layer. We detected problems with the protocols proposed and new functions were introduced in the link layer. The LLC exponential backoff function introduced in chapter 7 improved the overall network performance more than expected and should therefore be mandatory in NBWF. This function may reduce the signalling traffic in the presence of hidden-nodes giving increased network throughput capacity.

Pacing can be regarded as a stream-based flow control mechanism because this protocol performs traffic control on a (source, destination)-pair. This in contrast to PECN, which is a node-based mechanism; that is, Xon/Xoff turns on/off all streams using the Xon/Xoff originating node as a relay. A stream-based flow control gives a more fine-grained adaptation to the traffic conditions. PECN may be modified to operate as a stream-based flow control mechanism by extending the PECN signalling packet. However, this may exceed the payload size available in a single TDMA super frame slot. To send one single Xon/Xoff signal over two slots is certainly a disadvantage.

A traffic source that neglects resistance, or increases its traffic when it experiences resistance, is referred to as a *persistent traffic source*. A traffic source applying a backoff mechanism to reduce its traffic when it meets resistance is referred to as an *elastic traffic source*. An example of a traffic process belonging to the elastic traffic source category is a pacing process which serves the multihop traffic streams that have two or more hops left to the end-destination. An example of a persistent traffic source is a single-hop traffic stream, or the last-hop in a multihop stream (pacing is not applied on the last hop). When the pacing process “meets” a persistent traffic source, pacing allocates indirectly more bandwidth to its competitor by adding a pacing delay to the packet stream under service. The pacing protocol needs further analysis in complex network topologies where persistent and elastic traffic sources are present.

Below we give a short overview/comparison of the two flow control protocols analysed in this document:

Implementation complexity: None of the flow control protocols need to be executed as real-time processes in a real-world NBFW node. In the simulator, they have identical software complexity. With regard to implementation complexity, we regard these protocols to be equal.

Use of signalling traffic: Pacing operates without explicit signalling traffic while PECN sends signalling traffic in a TDMA super frame slot. However, only a few bits are required and they are sent in the same transmission burst as the periodic MAC reports [9, chapter 3].

Scalability: Pacing is insensitive of the network size. The PECN protocol’s report period delay increases as the number of network nodes increases and this protocol is therefore less scalable. However, steady-state simulations of short and long report periods did not discover difference in performance.

Robustness: Consider a PECN-network where the two neighbours A and B serve two traffic streams A→B and B→A, respectively. If both node A and node B have signalled Xoff, both traffic streams are blocked until one of them sends Xon. A Xon signal is trigged when the transit queue size becomes lower than a threshold but here this occurs only upon packet lifetime expiry. We suspect PECN may have long “deadlock”-periods that cannot be observed in a steady-state simulator. This potential problem demands further analysis.

Jamming: PECN is considered to be more vulnerable to network jamming due to the fact that a periodic dedicated signalling slot is used. Jamming leads to lost Xon/Xoff-packets. Pacing does not rely on explicit control packets. However, jamming leads to lost user data packets resulting in lost pacing MFD-samples. Pacing will then resort to the IFD-values but continues to operate, possibly with less efficiency.

Both pacing and PECN fulfil their task since both behaved well in the simulation experiments conducted. NBWF should implement one of the protocols. Based on the simulations done in this study, we are unable to decide which of the two candidates that is the preferred protocol for NBWF. Simulations of more complex networks are required to differentiate between the two candidates. The robustness of PECN is also a task for further study.

References

[1] "Requirements for a Narrowband Waveform", AC/322(SC/6-AHWG/2)M(2008)0003, August 2008.

[2] Svein Haavik, "Initial link layer protocol design for NBWF – input to NATO SC/6 – AHWG/2", FFI-rapport 2009/01895.

[3] Vivianne Jodalen, "Modelling the NBWF radio", TIPPER/FFI project document, FFI June 2008.

[4] Tore J Berg, "The design of an initial NBWF network simulator", FFI-report 2008/01921, FFI November 24th 2008.

[5] Bjørnar Libæk, et.al, "Enhancements to the Narrowband Waveform (NBWF) network simulator", FFI-report 2009/01765, FFI June 10th 2008.

[6] Bjørnar Libæk and Bjørn Solberg, "A simulator model of the NATO Narrowband Waveform physical layer", FFI-notat 2011/00533, FFI October 19th, 2011.

[7] Vivianne Jodalen, et.al, "NATO Narrowband Waveform (NBWF) – overview of link layer design", FFI-report 2009/01894, FFI March 28th, 2011.

[8] Tore J Berg, "Design of an initial LLC Data Protocol for the NBWF Simulator", FFI-report 2011/00537, FFI March 21st, 2012.

[9] Tore J Berg, "NATO Narrowband Waveform (NBWF) – multilevel precedence and preemption for IP traffic", FFI-report 2012/01884, FFI October 22st, 2012.

[10] John Jubin and Janet D. Tornow, "The DARPA Packet Radio Network Protocols", proceedings of the IEEE, January 1987.

[11] Tore J. Berg, "oProbe – an OMNeT++ Extension Module", http://sourceforge.net/projects/oprobe, 2007.

[12] Tore J. Berg, "oTWLAN – a tool to simulate tactical ad-hoc networks", http://sourceforge.net/projects/otwlan, 2010.

Terms and Acronyms

AHA	All hearing all
ARQ	Automatic Repeat Request
CAS	Carrier sense
CC	Connect Confirm
CC PDU	Connect Confirm PDU
CCCH	Common Control Channel
CEID	Connection Endpoint Identifier
CL	ConnectionLess
CNR	Combat Net Radio
CO	Connection Oriented
CODTC	Connection oriented data traffic channel
CR	Connect Request
CR PDU	Connect Request PDU
CTS	Clear To Send
DC	Disconnect Confirm
DR-PDU	Disconnect Request PDU
DSSS	Direct Sequence Spread Spectrum
DT PDU	Data PDU
EFD	Estimated Forward Delay
EPM	Electronic Protective Measures
FSM	Finite State machine
GUID	Global Unique Identifier
H_A	Hidden node set for node A
ICI	Interface Control Information
IFD	Initial Forward Delay
IP	Internet Protocol
IP-SAP	Internet Protocol SAP
LBN	Last Bit Number
LLC	Logical Link Control
LLC-AM	LLC Acknowledged Mode
LLCE	LLC Entity
MAC	Medium Access Control
MACE	MAC Entity
MAC-SP	MAC Service Provider
MANET	Mobile Ad-hoc NETwork
MFD	Measured Forward Delay
MLL-report	MAC Load Level Report
MLPP	Multi-Level Precedence and Preemption
NBWF	Narrow Band Wave Form
NC3B	NATO C3 Board
NLFC	Network Level Flow Control
NM-SAP	Network Management SAP

OSI	Open System Interconnection
PCAS	Premature CAS
p_{CC}	Probability to receive a CC PDU after sending a CR PDU
PCI	Protocol Control Information
PDP	Packet Data Protocol
PDU	Protocol Data Unit
PECN	Periodic Explicit Notification
PHY	Physical
PSMA	Preamble Sense Multiple Access
PTT	Push To Talk
QoS	Quality of Service
RATCH	Random Access Traffic CHannel
RF	Radio Frequency
RLC	Radio Link Control
RM	Reference Model
RRC	Radio Resource Control
RTS	Request To Send
SAP	Service Access Point
SDU	Service Data Unit
SNR	Signal to Noise Ratio
SOM	Start Of Message
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access