Flow Control in Wireless Mesh Networks

Ton Hellings

Supervisors:
Prof. dr. ir. S. C. Borst (TU/e)
Dr. V. Shneer (TU/e)
Dr. T. J. J. Denteneer (Philips Research)

Eindhoven, September 2009
Acknowledgements

This thesis is the result of an eight-month internship at the department Distributed Sensor Systems of Philips Research in Eindhoven and it also serves as the final project of the master’s program Industrial and Applied Mathematics at Eindhoven University of Technology.

I would like to take this opportunity to thank my supervisors, Dee Denteneer, Sem Borst, and Seva Shneer, for guiding me through the process so well. They have always shown a lot of enthusiasm and interest and were never too busy to carefully read a partial report or think along about how to prove a theorem or two. I have often thought myself lucky to have such capable, accessible and friendly supervisors.

Also I would like to thank Johan van Leeuwaarden, who stepped up and acted as a fourth guide near the end of the internship. I especially want to express my gratitude for his contributions to Chapter 4.

During this internship I have met a lot of interesting people from all over the world, both in the office and outside of the office and I would like to thank my office mates Sina, Nikolai, Ermanna, Anal, Zubair, Miguel, Xiao-Lei, Bruh, Deheng, Andreas, David and Alberto for all the fun times that definitely got me through the eight months faster than expected.

Furthermore I want to thank my family and friends for all the support during the past five years of my studies.

Ton Hellings
Eindhoven
September 2009
Summary

Wireless mesh networks (WMNs) form a relatively new branch within the field of wireless networking. WMNs aim to provide reasonable substitutes of wired networks on a large scale. This would reduce wiring costs and would allow installation of new network routers to become much easier. One of the distinguishing features of WMNs is the multi-hop property, which means that data can use intermediate nodes to be forwarded from a source to a destination.

However, there are still some problems that need to be tackled before WMNs become considerable competition for the networks we see today. The main problem of our interest is flow control. Employing standard wireless network protocols to WMNs results in substantial congestion in the middle of a mesh chain. A few flow control mechanisms have been proposed to combat this congestion and ensure a better flow within a mesh network. Currently a task group called IEEE 802.11s is preparing to standardize protocols for WMNs and the aim of this thesis is to help in the decision making.

The thesis focuses on two types of protocols, namely windowed flow control and the method of extra back-off. We use mathematical models relying on queueing theory to gain a fundamental understanding of the various flow control mechanisms and we present methods to analyze throughput for small systems, which already turn out to be quite complex.

For windowed flow control, in which only a fixed number of packets is allowed to travel within the system, we show that the mesh chain with a window size tending to infinity is equivalent to the mesh system with cyclic topology in which there is a saturated node within the system. We derive a sharp upper bound to the throughput of the small system.

For the method of extra back-off we propose four different implementable schemes which are also compared to each other. The results vary significantly for these schemes. For the scheme with truncated exponential back-off times, the system behaves in a stable manner and no saturation occurs within the system. The methods used to analyze the back-off schemes heavily rely on matrix-geometric solutions to quasi-birth-death processes.

By means of simulations we compare the throughput achieved by these flow control protocols to the uncontrolled version and observe a great improvement. Another form of flow control, called express forwarding is also used in the comparisons. The fundamental properties found for small systems are shown to extend to larger systems as well.
Contents

Acknowledgements iii

Summary v

Contents ix

1 Introduction 1

1.1 Networks ......................................................... 2
  1.1.1 The OSI model ............................................. 2
  1.1.2 MAC protocols in wireless networks ......................... 3
     Wireless networks ........................................... 3
     MAC protocols ............................................. 4

1.2 Wireless mesh networks ......................................... 9
  1.2.1 Single-hop and multi-hop networks ......................... 9
  1.2.2 Ad-hoc networks .......................................... 10
  1.2.3 Mesh networks ........................................... 10

1.3 Flow control .................................................. 11
  1.3.1 General flow problems .................................... 12
     Loss of efficiency ......................................... 12
     Unfairness ............................................... 13
     Deadlocks ............................................... 13
  1.3.2 Flow problems in WMNs .................................. 13
  1.3.3 Flow control ............................................ 14

1.4 Flow control in WMNs ......................................... 15
  1.4.1 Windowed data packet control ............................. 16
  1.4.2 Extra back-off after a transmission ...................... 16
  1.4.3 Express forwarding and express retransmission ............ 17
     Express forwarding ........................................ 17
     Express retransmission .................................. 17

1.5 Thesis outline ................................................. 18

2 Model 19

2.1 General model description ..................................... 19
  2.1.1 Measures ................................................. 21
  2.1.2 Known results .......................................... 21

2.2 Windowed flow control model .................................. 21

2.3 Extra back-off model ......................................... 22
## 3 Windowed flow control

3.1 Windowed flow control with finite window size

3.1.1 Window size $W = 2$

3.1.2 Window size $W = 3$

3.1.3 Window size $W = 4$

3.2 Windowed flow control with window size tending to infinity

3.2.1 Increasing the window size

3.2.2 System with a saturated second node

3.2.3 Equivalence of both systems

3.3 Windowed flow control with infinite window size

3.3.1 Upper bound through cycle analysis

3.3.2 Upper bound through Markov process analysis

## 4 Extra back-off

4.1 Multiple back-off schemes and preliminaries

4.1.1 Different back-off schemes

4.1.2 Preliminaries

4.2 A matrix-geometric solution

4.3 Back-off scheme (i): basic back-off

4.3.1 Back-off scheme (i) as a QBD process

4.3.2 Proof of Proposition 3

4.4 Back-off scheme (ii): zero back-off node 3

4.4.1 Back-off scheme (ii) as a QBD process

4.4.2 Main results for scheme (ii)

4.5 Back-off scheme (iii): truncated back-offs

4.5.1 Back-off scheme (iii) as a QBD process

4.5.2 Main results for scheme (iii)

4.6 Deterministic truncated back-off times

4.6.1 Preliminary results

4.6.2 Closer investigation using cycles

4.6.3 An approximation to truncated deterministic back-off times

4.6.4 Extension to non-truncated deterministic back-off times

4.7 Comparison of the various back-off schemes

## 5 Generalizations supported by simulations

5.1 Outline of simulations

5.2 Uncontrolled protocol

5.3 Windowed flow control

5.3.1 The case of general $n$ and $k$

5.3.2 Simulations supporting Hypothesis 1

5.3.3 Simulations and regression for windowed flow control
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>Extra back-off</td>
<td>81</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Basic back-off scheme</td>
<td>81</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Truncated back-off scheme</td>
<td>85</td>
</tr>
<tr>
<td>5.5</td>
<td>Express forwarding</td>
<td>88</td>
</tr>
<tr>
<td>5.6</td>
<td>Comparison of the various flow control schemes</td>
<td>88</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions</td>
<td>93</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusions</td>
<td>93</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Windowed flow control</td>
<td>93</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Extra back-off</td>
<td>93</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Comparison</td>
<td>94</td>
</tr>
<tr>
<td>6.2</td>
<td>Recommendations</td>
<td>95</td>
</tr>
<tr>
<td>A</td>
<td>Calculations on Markov processes for windowed flow control</td>
<td>97</td>
</tr>
<tr>
<td>A.1</td>
<td>The case of $n = 3, k = 1, W = 3$</td>
<td>97</td>
</tr>
<tr>
<td>A.2</td>
<td>The case of $n = 3, k = 1, W = 4$</td>
<td>98</td>
</tr>
<tr>
<td>B</td>
<td>Proof of stability of the cyclic mesh chain with saturated node 2</td>
<td>103</td>
</tr>
<tr>
<td>B.1</td>
<td>Generalization of the reflection principle</td>
<td>103</td>
</tr>
<tr>
<td>B.2</td>
<td>Proof of Theorem 1</td>
<td>104</td>
</tr>
<tr>
<td>B.2.1</td>
<td>Analysis of case (i)</td>
<td>105</td>
</tr>
<tr>
<td>B.2.2</td>
<td>Analysis of case (ii)</td>
<td>112</td>
</tr>
<tr>
<td>B.2.3</td>
<td>Conclusion of proof</td>
<td>114</td>
</tr>
<tr>
<td>C</td>
<td>Determining an upper bound to $\theta_{3,1,\infty}$ by analysis of the Markov process</td>
<td>117</td>
</tr>
<tr>
<td>D</td>
<td>Calculations on truncated deterministic back-off</td>
<td>127</td>
</tr>
<tr>
<td>D.1</td>
<td>Simplifying the throughput function for truncated deterministic back-offs</td>
<td>127</td>
</tr>
<tr>
<td>D.2</td>
<td>Expected value of $R_{i+1}$ for truncated deterministic back-offs</td>
<td>129</td>
</tr>
<tr>
<td>D.3</td>
<td>Proof of convergence of ${a_i(\eta)}_{i=0}^\infty$</td>
<td>130</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>135</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In the past few decades the use of wireless networks has increased massively. Nowadays many households possess a wireless router and one or more computers to connect to it. But not only on this small scale technological developments have been huge. Much research has already been done on wireless mesh networks. These are wireless networks that can partially replace the wired network as we know it today. This can be done on the scale of a home, a corporate building, a neighborhood, and even on the scale of an urban area. In wireless mesh networks, routers all work autonomously, they forward each other’s data and not all of them have to be directly connected to the Internet. This can drastically reduce the costs of placing wires and it makes the introduction of new network routers easier. Instead of installing new wired connections, one only needs to register the new router in the wireless mesh network. However, multiple problems still need to be tackled before mass deployment can be realized.

These issues are currently being discussed in IEEE 802.11s, the task group that standardizes wireless mesh networks for the widely deployed IEEE 802.11 wireless networks standards. IEEE is a global organization that aims to advance technology in many fields related to electrical engineering, wireless networks being one of these. This thesis is aligned with an effort within Philips Research to help the defining of international standards for wireless mesh networks.

When deploying the protocols that are currently used in regular wireless networks to wireless mesh networks, a lot of congestion occurs. This means that data do not arrive timely at their destination and therefore flow control has to be applied. Flow control is designed to assure a certain steady flow of data, limiting the delay of a packet and limiting the amount of congestion within a network. Since every network entity works autonomously, one of the characteristic features of a wireless mesh network is that there is no overseeing mechanism controlling congestion; the algorithms within the network nodes must be altered such that they take on this function themselves. In this thesis we propose mathematical approaches to study several forms of flow control and we investigate their throughput improvements. We compare the various flow control schemes to the uncontrolled version and to each other.

This chapter first gives general information about networks in Section 1.1. From Section 1.2 on the emphasis is on wireless mesh networks. This introduction is kept relatively brief, since some extensive overviews can already be found in [2, 23, 27]. In Section 1.3 the problems of congestion and flow control are looked at in more detail and in Section 1.4 some possible solutions to implement flow control and thereby combat congestion are reviewed. In Section 1.5 we give a complete outline of the thesis, stating our goals in more detail.
1.1 Networks

In this section some basic information about networks and their history is given, focusing mainly on wireless networks.

1.1.1 The OSI model

In the late seventies of the twentieth century, the need to interconnect computers became more and more apparent. There were no standards for this and therefore problems arose trying to establish connections, especially between computers from different manufacturers. For this reason, in 1977 the International Organization for Standardization (ISO) decided to establish a committee with the purpose of creating international standards that would smoothen interconnection of computers [31]. The term for this type of connection was “Open Systems Interconnection”, in short OSI.

The committee soon came up with the OSI Architecture Model, a model that describes data flows in a network and is still used today, albeit with some modifications. The model shows the layered structure of a network, as can be seen in Figure 1.1. When two users on different computers want to communicate with each other, their data will have to follow a certain path. Every stage of the path has a different function. Data generated in the top layer of Host A will follow its way down to the bottom layer of Host A, then it will be transported to the bottom layer of Host B, after which it travels up to the top layer of Host B. The OSI model distinguishes seven layers:

1. *The Application Layer*. This is the highest layer in the OSI model, providing the user with applications such as a web browser or e-mail. This layer creates, transmits and receives data. The rest of the layers are only working for this layer.

2. *The Presentation Layer*. This layer serves to translate, protect and compress data. It prepares data for sending and it converts the received data back so it is useful for the Application Layer.

3. *The Session Layer*. The Session Layer keeps track of the sessions that are going on with other computers. It starts and ends communications and it delimits and synchronizes sessions.
4. The Transport Layer. The fourth layer of the model ensures a steady and fast data flow between network layers. It splits the data flows in packets and monitors the transmissions and the rates.

5. The Network Layer. This layer performs the routing in a network, it makes sure there is a connection path between two transport entities.

6. The Data Link Layer. This layer establishes, maintains and releases data links on the physical entities. It adds some bits to the beginning and end of data packets so that the receiving entity will be able to check whether the transmission has been completed successfully. It receives or sends acknowledgements (ACK) after a successful arrival. The Medium Access Control sublayer (MAC) of the Data link Layer tries to avoid collisions with packets from other entities and this sublayer will be the main focus of this project.

7. The Physical Layer. The Physical Layer (PHY) finally physically sends the packets to another entity. This can be done for example by means of a radio signal, a transmission through a wire, or optical transmission.

This model is still used today, although there have been discussions about combining or splitting layers [13], and about functions that make use of more than one layer, see [23].

1.1.2 MAC protocols in wireless networks

The difference between wired and wireless networking is not only a matter of the physical layer. Other layers will also be drastically influenced by the choice of the used medium. Especially the data link layer with its MAC sublayer might need to behave differently, as will be explained later on in this subsection.

This holds not only for wireless mesh networks, but for any type of wireless network, such as Bluetooth, Zigbee, WiMax to name a few. However, we will focus specifically on the MAC protocols of IEEE 802.11 (WiFi).

Wireless networks

In wired networks data are sent through a cable. In this case the medium is closed, so no undesired transmission can interfere. This is not the case in wireless networks. The transmissions go through the air by means of electromagnetic waves. Since no specific closed medium is reserved for a transmission, multiple transmissions can go through the same space. The signals are not directly sent to one receiver, they are just broadcast into the air, having no specific direction in general. This can cause interference, meaning that a receiver can happen to hear multiple signals at the same time. Furthermore, a continuous background noise can be present. These factors decrease the probability that a message is successfully delivered. Especially when the communication has to travel a long distance, the signal can become weak compared to the interfering transmissions. This property is quantified by the signal to interference and/or noise ratio (SINR, SIR or SNR):

\[
\frac{S}{I + N} > t.
\]
This value divides the received power $S$ of the signal by the power of the interference $I$ and/or noise $N$ and it should be above some threshold $t$ to guarantee a successful transmission.

The SINR is a function dependent on many variables. The distance between two communicating entities is one of them. The transmission distance is the most important factor, when one assumes the background noise and all power levels of transmissions to be constant, and furthermore that the interference depends only on the distance to the surrounding transmitters. Generally speaking, when the distance to the receiver is larger than a certain threshold, the transmission will not be successful. This distance is called the transmission range.

The range in which an entity can hear others transmitting is called the carrier sensing range. The size of this range need not be equal to the size of the transmission range.

We will assume that a transmission from one entity to another one within its transmission range is successful when there are no other entities transmitting within the carrier sensing range of the receiver. In this thesis we therefore use a blocking range, as discussed in Section 2.1.

**MAC protocols**

As was explained in Section 1.1.1, the second layer of most network models is the data link layer. This layer has a sublayer called the Medium Access Layer, which focuses on when to access the physical layer (PHY) and when not to. This way it tries to avoid collisions and in case of a collision it will decide how and when to retransmit a packet. Numerous MAC protocols for wireless networks have been developed over the last several decades and the most important ones will be discussed here in a chronological order.

A pioneering protocol was the ALOHAnet, developed in 1970 under leadership of Norman Abramson [1]. If in this protocol a transmitting node does not receive an acknowledgement within a certain time after finishing the transmission, it will try to resend the packet later. The time it waits before retransmitting the packet is random, so two consecutive collisions of the same packets will only happen when the random waiting times are close enough to each other. See Figure 1.2.

The main idea of waiting a random time before retransmitting a collided packet, called the back-off mechanism, is also the basis for CSMA, a protocol that is still used in today’s IEEE 802.11 protocols, which will be discussed later in this section. One of the problems of ALOHA is that all transmissions are initiated independently of each other. The stations just transmit whenever they have a packet to send. In the CSMA protocol, which stands for Carrier Sense Multiple-Access and was introduced by Kleinrock and Tobagi in 1975 [20], an entity first listens whether a transmission is already taking place in its carrier sensing range. If so, it will wait until the channel is free. There are variations of this principle, in which the stations try to transmit more or less aggressively. For example, whenever a node in $p$-persistent CSMA senses the channel free, it will transmit its data only with probability $p$. With probability $1 - p$ it will wait a certain amount of time and then reschedule the transmission. CSMA improves the throughput (the number of successfully delivered packets per time unit) compared to ALOHA and decreases the number of collisions, as can also be seen in [20].

CSMA/CA stands for CSMA with Collision Avoidance. Now a station that wants to transmit and senses the channel idle, will wait some more time, checking whether the channel stays idle, before starting the transmission. This brings down the number of collisions even
Figure 1.2: ALOHA; nodes 2 and k have a collision and resend their packet (Abramson, [1])
In the IEEE 802.11 protocol, a set of standards that are widely implemented in wireless routers, a technique called *Distributed Coordination Function*, abbreviated as DCF, is used. DCF is based on CSMA/CA and it uses two different mechanisms, the basic mechanism and RTS/CTS.

First the basic access mechanism will be explained in short, see Figure 1.3. Whenever a station that has a packet to send first senses the channel free, it waits for a certain time interval, called the DCF Interframe Space (DIFS), after which a backoff mechanism is initiated. The station picks a number in $\{0, 1, \ldots, W - 1\}$ where $W$ is an integer representing the back-off window size. It then starts counting down, until it reaches 0. If in the meantime it senses the channel becoming busy again, it will stop the back-off timer until it senses the channel free for another DIFS. During that period we call the station frozen. After it has sensed the channel free for a DIFS, it will continue decrementing the timer until 0 is reached. Then it will immediately start sending the data. After the sending of the data is completed, it will wait for a Short Interframe Space (SIFS), after which the receiving station should send an acknowledgement (ACK) to let the transmitting station know the data packet was received successfully. If it does not receive an ACK, the station concludes that the transmission was unsuccessful. This means it has to retransmit the packet. However, when the transmission interferes with another transmission, there is a probability that the other transmission also fails. If both stations wait for a small amount of time before retransmitting, their packets would have a high probability of colliding again. Therefore the back-off window size is increased. DCF doubles the window size every time a transmission fails. This process repeats until the transmission is successful or a maximum window size $W_{\text{max}}$ is reached. In the latter event, the window will not increase any further. As soon as the transmission is successful or a maximum number of failures has occurred, the back-off window size is reset to the initial value $W_0$. If the packet has not yet been delivered successfully, it will be dropped.

Of course, the window size does not necessarily have to be doubled. It can also be enough to multiply it by 1.5 or it might be necessary to multiply the window size by a number larger...
than 2. Also the initial window size $W_0$ can be changed. These values can be altered globally, but also generically, for example giving one station temporarily a lower $W_0$ increases the probability that this entity will be quicker to access the channel than other stations. Research in this direction has been done by Xiao [28], among others.

The other mechanism, called RTS/CTS, is depicted in Figure 1.4. After its back-off timer reached 0, a station first transmits a small Request To Send (RTS) packet. If the receiving station is ready to accept an incoming data packet from the transmitting station, it will wait for a SIFS and then send a Clear To Send (CTS) packet back. If the transmitter does not receive a CTS in time, it will not start the transmission, assuming the receiving station is busy with another transmission that it cannot hear. If the transmitting station does receive the CTS, it will send the packet after a SIFS and again wait for an ACK.

Not only the node that wants to transmit its data can receive the CTS. Other stations within the transmission range of the destination station will also hear the CTS message and they then know that another station will be sending. Just like other messages, the RTS and CTS messages also contain information about the duration of the transmission. The other stations now know how long the channel will be busy. They do not have to keep checking all the time. This RTS/CTS mechanism is also called Virtual Carrier Sensing. All stations have a Network Allocation Vector (NAV) in which they store information about the stations around them.

This mechanism tries to prevent the so-called hidden-node collisions. A hidden node occurs when two stations have a common neighbor in between them, but are not within each other’s carrier sensing ranges, as is seen in Figure 1.5. Node A is transmitting a packet to station B and at the same time station C wants to transmit a packet as well. The destination of this packet of node C is not important, it can be B or some other node D. So, when A is sending a packet to station B, C will not be able to hear this. In basic mode it will assume the channel is free and start transmitting its packet. Because B is within the transmission range of both nodes A and C, it will not be able to receive the data successfully, since stations A and C are interfering with each other. In RTS/CTS node C will have heard the CTS that
B replied on A’s RTS and know that node B is busy and for how long, even though it does not hear the transmission that node A is sending.

Another fundamental issue that is prevented by RTS/CTS is that of the exposed node, as is seen in Figure 1.6. In this scenario node B and node C are within each other’s carrier sensing range and they want to send a packet to node A and node D respectively. Node B is not in the carrier sensing range of node D and the same goes for nodes C and A. Now node B is transmitting to node A, while node C wants to start a transmission to node D. When there is no RTS/CTS, node C will first listen and hear that node B is transmitting and assume the channel is busy, so it will not start its transmission. However, since node C is not within the carrier sensing range of node A, the current transmission will not be disturbed by node C’s transmission. Now node C is unnecessarily waiting.

There are different views on whether or not the RTS/CTS mechanism really improves the throughput. According to research done by Bianchi [9, 10], the throughput significantly increases. However, there are still people [29] who believe RTS/CTS unnecessarily increases the transmission duration and therefore does not work well, as opposed to Bianchi’s findings. A RTS packet sent by a station requesting a transmission to a busy station can still, despite
1.2. Wireless mesh networks

its short length, ruin an ongoing transmission and therefore it will still cause collisions.

Another function in the original 802.11 MAC, which is not used as much as DCF, is the 
Point Coordination Function, PCF in short. This function uses a central Access Point (AP) 
which serves to tell the surrounding nodes which channels are free. The AP coordinates which 
nodes are allowed to transmit. This improves the Quality of Service. The AP in the middle 
practically doubles the carrier sensing range in one direction. Although this exact function is 
barely implemented in hardware, it does provide the basics for another access function, the 
HCF.

In 2005’s 802.11e standard a new access function has been described, called the Hybrid 
Coordination Function, HCF. This function uses both the DCF and PCF, but now in two 
new separate functions, EDCA and HCCA.

In DCF and PCF there is no distinction between traffic. There exist no priority classes, 
so all data is treated the same. However, for some applications it might be necessary to send 
packets with higher priority, so that they do not have too much delay. Enhanced Distributed 
Channel Access (EDCA) is a function implemented in 802.11e that does distinguish between 
data with different priorities. A station that sends data with high priority has a shorter 
back-off time than a station sending data with lower priority. In EDCA the stations reserve 
a channel for a fixed time interval, called a TXOP, in which they can send multiple packets.

In HCCA, HCF Controlled Channel Access, there is a lot of similarity with the PCF. The 
equivalent of the AP is called a Hybrid Coordinator (HC) and it also controls the transmis-
sions. In HCCA there is also the possibility of using different priority classes and it allows for 
sending a stream of packets, just like in EDCA. HCCA is considered to be the best coordina-
tion function that exists at the moment, although it does need an extra HC. Unfortunately, 
the availability of an HC is not assumed in wireless mesh networks, which is our main topic.

1.2 Wireless mesh networks

Most of the wireless networks that are deployed today are single-hop networks with a cen-
tralized access function. A centralized access function means that the nodes do not work 
automonomously and are given instructions by a node higher in the hierarchy (for example a 
wireless router telling laptops when they can send their data). In contrast, mesh networks 
are multi-hop networks with distributed self-configuring protocols. The difference between 
single- and multi-hop will be explained in Section 1.2.1.

Wireless mesh networks (WMNs) might be the future of the wireless Internet. Mesh 
applications can drastically reduce the number of access points needed. Instead of needing 
wired access points in every household and at multiple locations in a corporate environment, 
one wired Internet access point per street might be enough. Using WMNs the wiring costs 
can be reduced considerably. This can be important when new techniques, such as fiberglass 
cables, are implemented, when previously unconnected households in developing countries 
receive Internet connection, or when new neighborhoods are built.

1.2.1 Single-hop and multi-hop networks

In a single-hop network one station sends a message to another station and this completes 
the route of the sent packet. A packet is not forwarded by intermediate nodes.
Multi-hop, in contrast, is the setting in which a packet may have multiple intermediate stations on its route. In this case a station has not only transmitting and receiving capabilities, but also the capability to forward packets. This makes networking on a large scale possible, even when transmission ranges are not large. In this project we will mainly focus on multi-hop settings.

1.2.2 Ad-hoc networks

Ad-hoc networking is not new in the research world. The idea of networks without a centralizes organization has been around since approximately 1970. Although there is no scientific definition of an ad-hoc network, the main idea is practically the same everywhere. In ad-hoc networks the stations, which from now on we will call nodes, are not connected to a central router that handles all the traffic. Instead, each node configures its own actions. A node can join the network and schedule its own traffic, following a distributed protocol. The nodes are said to be self-organizing and when the infrastructure is changed, with nodes disappearing for example, the network can adapt itself.

In general, an ad-hoc network does not even need to be connected to the Internet. One can also use an ad-hoc network in a secluded area, with the only purpose of having the nodes communicate with one another.

In the literature, the term ad-hoc network is often used for describing mobile ad-hoc networks (MANET) only. In such networks the nodes may also have mobility, so that the topology of the network changes continuously.

Ad-hoc networking can be both single-hop and multi-hop. However, since the accessibility of the other nodes greatly depends on the position of a certain node in a single-hop setting, it is more reasonable to assume a multi-hop setting for the more advanced networks.

1.2.3 Mesh networks

A mesh network is assumed to comprise two types of nodes. First there are the mesh routers, which may or may not be connected to the Internet. If the routers are connected to the Internet, we speak of Internet gateways. Each router has the capability to transmit, receive and forward data. The second type of nodes are the mesh clients. These are also able to transmit, receive and forward data, but they are never directly connected to the Internet. One can think of laptops or advanced mobile phones. A schematic architecture of a mesh network is shown in Figure 1.7. This structure can be applied on different scales. For example in a city, with the mesh routers and access points on rooftops, or in a corporate building with the mesh routers in the hallways and the nomadic users in the offices.

Similar to ad-hoc networks, for mesh networks there is no generally accepted definition either. There is however a number of characteristics that WMNs should have. According to Akyildiz et al. [2] these are:

- Multi-hop wireless network. With multi-hopping the coverage range can be extended. The problem how to maintain a certain data flow in long routes will be the main problem of this project.

- Support for ad-hoc networking. The network should be able to form, heal and organize itself. Adding mesh points should be easy this way, without too much organization or
1.3 Flow control

The problem that we will be looking at is that of flow control in wireless mesh networks, incorporated in the MAC layer. Flow control means applying one or multiple protocols such that the flow is constrained. It puts restrictions on when packets can be transmitted. In this section flow control will be discussed, with first the motivation why flow control is necessary.

- Dependence of mobility and power-consumption constraints on the type of mesh node. Mesh routers are assumed to be connected to an electrical power source. Power is not an issue for them. Mesh clients do not have too much power available. Furthermore routers are assumed to be immobile, while clients do not necessarily have to be.

- Multiple types of network access. Both access to the Internet and peer-to-peer communication should be possible in WMNs.

- Compatibility and interoperability with existing wireless networks. Both conventional clients and mesh clients should be able to connect to WMNs.

A special group to define standards in the MAC layer for mesh networks was established by IEEE in 2003, under the name 802.11s. This task group is planning to deliver the new standards for mesh networking in the nearby future. The new protocol is supposed to be an extension to the current 802.11 MAC protocol, modifying and adding functionalities such that the protocol will be suitable for multi-hop networks. The goal is to ensure a high throughput, since the current 802.11 MAC protocol does not avoid congestion sufficiently well in a multi-hop environment.

Figure 1.7: A schematic architecture of a mesh network (Bruno et al. [11])
1.3.1 General flow problems

Generally in networks there are three main problems that arise when the offered load (i.e. the amount of data per time unit that is offered to the system) is increased. Those are the loss of efficiency, unfairness and the appearance of deadlocks [17].

Loss of efficiency

In wired networks the loss of efficiency is often a consequence of overflowing buffers, when buffer space is assumed to be finite. Buffers fill with packets being transmitted from one node, which monopolizes the buffer space. Gerla and Kleinrock [17] present the following example, as seen in Figure 1.8. The situation is simplified in the sense that we do not assume complicated protocols at work, but use a scheme where a node attempts to transmit a packet from its queue after exponential times. If there is place in the waiting queue (or buffer) of the receiving node, the transmission is successful, otherwise it will be rescheduled.

In this situation there are two crossing flows, one from node A to A', and one from node B to B'. The flows cross in a switch that has a finite buffer. Packets arrive at node A and B following Poisson processes with respectively rate 0.8 and rate $\lambda$. Whenever a packet is available in node A, it will be scheduled to be transmitted to the switch with a rate 1, and from the switch to node A' with rate 1 as well. For flow B−B' these numbers are 10 and 1 respectively. The rate $\lambda$ can vary and for $\lambda < 1$ the arrival rate is strictly lower than the service rate, so the system will not have problems processing the total load of 0.8 + $\lambda$. When $\lambda \geq 1$, however, the channel between the switch and node B' is not able to handle the load anymore, so the buffer of the switch will be filled. When this buffer is full, the packets from nodes A and B will not be able to enter right away, they have to wait for an open spot. Therefore queues will also build up at nodes A and B.

What happens now, is that the arrival process of the packets coming from node B has a rate that is 10 times higher than the rate of packets coming from A. This means that whenever there is a free spot in the buffer, node B will have a 10 times higher probability of taking it. Since packets from node B can only leave the switch at rate 1, in stable conditions they can also only enter the switch at this rate. This results in a throughput of only 0.1 for flow A−A',...
1.3. Flow control

![Figure 1.9: A part of a chain with three nodes with the second one starving](image)

... bringing the total throughput back to $\theta = 0.1 + 1 = 1.1$ for $\lambda \geq 1$. This is a loss in efficiency coming from throughput $\theta = 0.8 + \lambda < 1.8$ for $\lambda < 1$. The reason for this is the monopoly of flow B–B’ on the buffer space in the switch. It is obvious that some restrictions should be imposed on packets from node B.

**Unfairness**

Another problem is that of unfairness. Unfairness arises when some flows manage to take a larger part of a shared resource than others. In the example above unfairness also arose, combined with a loss of efficiency. Flow B–B’ managed to take 10 times more buffer space than flow A–A’. As shown in Section 1.3.2, for wireless networks this problem is also very relevant, albeit not because of sharing a limited buffer space, but because of sharing wireless links.

**Deadlocks**

A deadlock is a situation in which the network crashes because of overflow. This happens for example when two users cannot finish their own task before the other one has finished his task. So they are waiting for each other to finish, halting their own process and this way the system (partially) reaches a fixed state out of which it will not escape. One can think of a traffic intersection where four cars arrive at the same time and every car has to give priority to the car coming from the right. If all drivers wait for another car, they will be waiting forever.

We see that problems arise when the offered load increases. In this thesis, we will be concerned with saturated conditions, in which the rate of offered packets is taken to be infinity. In a less mathematical way, the nodes that are saturated always have a packet to send. Investigating this limiting behavior captures the essence of flow problems.

Flow problems can also be seen in WMNs, as will be explained in the following section.

### 1.3.2 Flow problems in WMNs

When applying the standard 802.11 MAC protocol to wireless mesh networks, there is a congestion problem. This problem already arises when a flow of packets has to travel from one node to another node through two or more intermediate nodes, as is seen in Figure 1.9, where node A transmits to some destination node through intermediate nodes B and C.

It is assumed that node C transmits the data to another (irrelevant) node. When the 802.11 MAC protocol is applied, each node that has a packet ready to send will compete with its neighbors in a fair way, as described in Section 1.1.2. It is assumed that every node will be able to transmit if and only if its direct neighbors are not transmitting. Now it is possible for node A to transmit whenever node B is not transmitting. Node B, however, has to wait until both node A and node C are silent. Therefore, node B has a fiercer competition than node
Chapter 1. Introduction

A. This will result in less transmission opportunities and subsequently a lower throughput for node B. This way data packets will pile up for node A and its queue will continue to grow, thus congestion occurs. This example has been explored in more detail by Denteneer et al. [15].

This problem is called the ‘node-in-the-middle’ problem. This means that the nodes in the middle of a chain will have lower throughput than the nodes closer to the edge, because of stronger competition. This problem is characteristic of wireless networks, because in wireless networks two transmissions can interfere, as opposed to transmissions in two distinct wired channels.

It can still be the case, however, that a network like this has quite a high throughput. A decent part of the amount of packets put into the system might be delivered successfully. But if, for example, the queue of the second node never stops growing, the mean amount of time the packets spend traveling from the source to the sink will also continue to grow. So, in case one has to transmit a very large file, comprising many packets, the first might be transmitted very rapidly, but the last ones will have a very long waiting time, because they have to take place in the back of the queue that has formed and has only increased in time. This should be improved by applying flow control.

In reality the waiting time of data packets will not keep growing, because the layers above the Data Link layer will receive information about what packets have and have not arrived at the destination. When the waiting time is too large, the transmission attempt is assumed to have failed and retransmission of the data packets is scheduled, discarding all data that are still waiting in the system. The session will be reinitiated and the unsent packets will be transmitted through the now less busy channel. However, for mesh networks the problem is even larger than for basic WLANs (wireless local area networks), since there is no control mechanism and every node will try to transmit its data autonomously. Therefore good flow control is also needed more.

1.3.3 Flow control

Flow control is the collection of algorithms and protocols designed to prevent throughput degradation, fairness and deadlocks. In the literature flow control protocols have been subdivided into four categories, operating on different levels [17]:

- **Hop level.** Flow control on the hop level is the collection of flow control algorithms that operate locally, between neighboring nodes.

- **Entry-to-exit level.** This level regulates how much data to accept at the first node based on information coming from the last node, in order to avoid congestion at the sink.

- **Network Access level.** Flow control on the network access level also regulates the incoming data, but now it is based on information from the entire internal network.

- **Transport level.** The algorithms classified as flow control on the transport level try to prevent congestion outside the network.

It commonly happens that more than one flow control algorithm is applied in a protocol. Especially flow control algorithms from different categories can easily be implemented together.
1.4. Flow control in WMNs

Flow control is important for various applications. Internet data can be divided into two main categories: elastic traffic and inelastic traffic. Elastic traffic is traffic for which a packet-level delay is not a major impediment, for example file transfers. Here it is more important that the data arrives, than when the data arrives. The rate of receiving may vary during the transmission without severe consequences. Inelastic traffic on the other hand does rely on transmissions without too much delay on packet-level. One can think of streaming audio and video, or VoIP (Voice over Internet Protocol). The elastic traffic should not cause unfairness, degrading the throughput of the inelastic traffic. Flow control can solve these fairness issues. In general it can also improve the overall throughput.

In Figure 1.10, the generally observed graphs are shown, in which the throughput is a function of the offered load. It shows that when load increases, the uncontrolled protocols have a catastrophic throughput decrease, resulting in a deadlock. The controlled protocol will have less throughput with a lower offered load, but becomes stable as the offered load is increased.

Deadlocks do not always occur. In the case they do not happen, the throughput of the uncontrolled version will most likely reach a stable limit as well, although a little lower than the controlled version.

Sofar, the problem of flow control in WMNs has mostly been explored experimentally. Not much research has been done on this topic yet. The main existing ideas will be discussed in the following section.

1.4 Flow control in WMNs

The problem of flow control in WMNs has not yet been extensively researched, but some possible solutions have already been proposed to ensure a better packet flow and higher end-to-end reliability. These are taken into consideration for the upcoming standardization of IEEE 802.11s.

We present three flow control mechanisms in this thesis. The first two of these solutions, windowed flow control in WMNs (proposed in Section 1.4.1), and extra back-off in WMNs (proposed in Section 1.4.2), will be our main research topics.
1.4.1 Windowed data packet control

One way of implementing flow control, is putting a window on (a part of) a network. This is the best known form of flow control and it operates on the network access level. The NAVs and back-off times are not changed in any way. This time the difference with a basic protocol is that only a certain number of packets is allowed into the system. The measure only affects the sources of a flow, because they can only send a packet into the system if there is a free space, after a successful arrival at a destination.

Using this method, a packet is guaranteed not to end up in an enormous queue. Its time in the system will always be expected to be below some constant value, no matter at what point in time the packet arrives, provided that all other flows in the system also have windowed flow control.

A disadvantage is that there is a limited amount of input. The congestion will shift from the inside of the system to the access point of the system, so this is a form of flow control at the network access level. However, as has already been shown briefly in [15], the throughput significantly increases, while a stable flow is maintained.

Another disadvantage is that this flow control mechanism requires a lot of feedback from the last node to the first node. Especially for large systems it seems very hard in the WMN scenario to give the first node continuous feedback on when there is room for the admission of another packet into the system.

When the system is in a reasonably stable state, however, the feedback from the last to the first node could be brought back to solely the end-to-end throughput rate over the past $t$ units of time for a fixed $t$. The input rate can be set equal to the feedback for the next time interval $t$. This is less accurate though and the main idea of a windowed control might not be satisfied this way, since the number of packets in the system is now allowed to fluctuate.

1.4.2 Extra back-off after a transmission

Another method, presented by Hiertz et al. [19], is the method of extra back-off after a transmission. The fundamental thought behind this method is the same as the idea behind express forwarding and retransmission, which will be discussed in Section 1.4.3. The key is to make sure it is quieter on the channel when a packet has just been transmitted and the next node on the route has to forward it.

After a transmission a node is forced into the back-off state, in which it cannot start a new transmission. The node in back-off will give the just transmitted packet the opportunity to travel out of its transmission range, before transmitting a new packet. This way an attempt is made to avoid a packet just transmitted from having interference with the next packet.

This method can be integrated in a basic setting, giving some flows higher priority than others. However, this flow control protocol only affects individual nodes and a node does not communicate with another node for which packet it just went into back-off. The other nodes might not use the time they are given to forward the packet to really transmit this certain packet, unless some other modifications are also made.

Some more analyses based on simulations can be found in [18]. This flow control scheme is of most interest for Philips Research, since their contribution has been substantial.
1.4.3 Express forwarding and express retransmission

The methods of express forwarding and express retransmission are due to Benveniste [6]. This is a form of flow control operating on the hop level. The idea is that the nodes that are busy with a transmission ‘cheat’ on the neighboring nodes and reserve extra extra time.

Express forwarding

Express forwarding is an extension to the basic IEEE 802.11 MAC protocol. In this latter protocol the RTS and CTS packets set the network allocation vector of their neighbors, so that they do not access the channel during the time reserved for the transmission. The channel is usually reserved until the ACK has been sent, as can be seen in Figure 1.4. After that there is a DIFS and following this space the nodes will start competing to be the first to send the next packet.

In express forwarding the NAV of the neighboring nodes is not set to right after the ACK. In this method the RTS and CTS packets extend the reservation by a short period, called $DT_0$ by Benveniste, see Figure 1.11. Only the receiving node knows that this extra duration is added. The other nodes (including the transmitting node) will still be waiting, and all except the transmitting node are assuming the channel is busy. In this short period of time the receiving node gets the opportunity to quickly start forwarding the packet it just received, updating the NAVs of the surrounding nodes. The node does not have to compete with its neighbors, since they will not attempt a transmission yet. This way a packet will have a much faster flow through a chain of nodes, because it will have a higher probability of immediately being forwarded by the next node.

Express retransmission

Express retransmission is a supplement of express forwarding, which makes sure a packet does not suffer too much from a failed transmission. After the ACK time has elapsed and the transmitting node has not received the ACK packet, it will know that the packet has to be retransmitted. In the $DT_0$ time that had been reserved to forward the packet the receiver has
nothing to send. The transmitter now grabs the opportunity to retransmit its packet. Again none of the surrounding nodes have the opportunity to start their own transmission.

The methods of express forwarding and express retransmission can also be used to prioritize certain data streams, so that their flow is kept at a certain level. The system can still handle the rest of the data streams without express forwarding, possibly with a lower throughput. If there is a queue of low-priority packets at the node that just received a high-priority packet, it must know that the extra $DT_0$ time is meant to start transmitting the packet it just received, not one of the packets with low priority. An analysis of express forwarding has also been provided by Zhou and Mitchell [30]. Furthermore, simulations have shown that this form of flow control greatly improves end-to-end reliability and substantially reduces packet delays [4, 5, 7, 8].

1.5 Thesis outline

The goal of this thesis is to analyze the throughput achieved by the various flow control mechanisms that were proposed in Section 1.4, and thus help in the process of making decisions for the upcoming standardization. We will make use of mathematical models to analyze the effect of windowed flow control and extra back-off on mainly the system with three nodes in a chain topology. As it turns out, these systems are already far from trivial, and greatly improve our fundamental understanding of the various flow control mechanisms. The mathematical models are introduced in Chapter 2. In Chapter 3 we investigate windowed flow control in more detail, presenting an equivalent system and analyze this further. In Chapter 4 we will treat extra back-off as a flow control mechanism and analyze four different back-off schemes in detail. In Chapter 5 we will extend the research to systems with more than three nodes, based on simulation results. We will see that what holds in the simplest case, essentially also holds in more general cases. We conclude in Chapter 6 and give recommendations for further research.
Chapter 2

Model

The protocols discussed in Section 1.4 are very complicated to analyze. We will make some simplifying assumptions and restrictions, such that the key characteristics of the process are still preserved. In this chapter the simplified model will be explained, both for the uncontrolled protocol and for the flow control measures mentioned in Section 1.4.

2.1 General model description

We will be looking at a chain of $n$ nodes, forming a mesh network. The first node will be saturated, which means it always has packets ready to be transmitted. The packets travel through the system, being forwarded by the intermediate nodes, as seen in Figure 2.1. The last node, node $n$, will forward the packet to a node outside of the system. The arrival process at node $1$ is not taken into account, while the departure process of node $n$ is.

The transmission time of a packet is assumed to be exponential with mean $1/\mu$, where the $n$ consecutive transmission times of a packet are all independent exponential variables. This means that a packet will not take the same amount of time at every node. Also, the transmission times needed by a certain node to transmit consecutive packets are independent, so that no node is faster than another on average. In general we will take $\mu = 1$.

The carrier sensing range and transmission range are replaced by a more general blocking range. When a node is within the blocking range of another node that is already transmitting, it is blocked. The nodes are assumed to be equidistant in the chain, and therefore the blocking range can be expressed as a number of nodes $k$ on each side of the transmitting node that have to be quiet during the transmission. For example, if node 1 is transmitting, nodes 2, 3, ..., $k+1$ will have to be silent. Nodes that can block each other are called interfering nodes. In the one-dimensional chain it holds that nodes $i$ and $j$ block each other if and only if $|i - j| \leq k$.

This simplification can be justified by a simple assumption. Namely, when the carrier sensing range $r_{cs}$ is at least the transmission range $r_t$ plus the distance between two neighboring

![Figure 2.1: A simple mesh chain with $n$ nodes, in which the first node is saturated.](image)
nodes, which we take 1, we have the following relation

\[ k = r_{cs} \geq r_t + 1. \]

This can be explained by noting that in this case a transmission from node \( i \) to node \( i + 1 \) can always interfere with transmissions received by the \( r_t \) neighbors of node \( i \) on both sides, nodes \( i - r_t, ..., i + r_t \). This means for the one-way traffic, in which each node \( j \) transmits to node \( j + 1 \), that nodes \( i - r_t - 1, ..., i + r_t - 1 \) must stay silent and therefore have node \( i \) in their carrier-sensing range. This leads to

\[ r_{cs} \geq \max\{|i - (i - r_t - 1)|, |i - (i + r_t - 1)|\} = r_t + 1. \]

Also in the case of two-way traffic, this relation holds, because of symmetry and the irrelevance of the current receiving node \( i + 1 \).

The access protocol of the IEEE 802.11 MAC, as was described in Section 1.1.2, will be replaced by a protocol that is simpler and easier to analyze. The original back-off timers are replaced by exponentially distributed back-off timers. Each node that has a packet ready to send and is not blocked by another transmitting node in its neighborhood, generates an exponentially distributed random number, each with the same mean \( 1/\lambda \). Whenever the back-off timer of a node reaches zero, the node checks whether or not it is blocked by one of its neighbors. If it is not blocked, it will start transmitting. When a node starts transmitting, it updates the NAVs of its neighboring nodes.

We will be investigating the case in which \( \lambda \to \infty \), so that the back-off times approach zero, and only serve to determine which node will be able to capture the channel next. This corresponds to a situation in which the periods of accessing the channels in the system are negligible compared to the busy periods.

Furthermore it is assumed that transmissions do not fail. Whenever a node has reserved the channel, it will send its message successfully in an exponential time. This is not a very strong constraint, since when assuming that a transmission would fail with probability \( p > 0 \) and a failed transmission will be immediately retried, the time until a successful transmission is still exponentially distributed. Indeed, the sum of a geometrically distributed number of exponential variables with mean \( 1/\mu^* \) is again an exponentially distributed variable, this time with mean \( 1/(1 - p)\mu^* \). Since \( \lambda \to \infty \), we can take \( \mu^* = \mu/(1 - p) \) and we would only have a scaling difference. In reality, this does require some form of express retransmission though, as was discussed in Section 1.4.3, because otherwise the failed transmission will not necessarily be retried immediately.

So now we have \( n \) servers in series, each serving customers with rate \( \mu = 1 \). The only difference with regular queueing systems is that the space of simultaneously busy servers is restricted. We have

\[ S_{n,k} = \{N \subseteq \{1, ..., n\} \mid \forall i, j \in N : |i - j| > k\}, \]

where \( S_{n,k} \) is the collection of possible states of transmitting nodes. There is still a small restriction to this, because in the \( N \in S_k \) in which \( \min(N) \geq k + 2 \), the first node, which is saturated, will be able to transmit another packet and will thus do so.
2.1.1 Measures

We will be focussing on throughput as the main performance measure, where throughput is defined as the average amount of data packets leaving the system per time unit. Another interesting measure is the average delay that a packet experiences from entering the system at node 1, to leaving the system from node \( n \). For an unstable system in which the throughput is not equal to the amount of packets that enter the system, however, queues will always continue to build up. In such a system the congestion will cause the average delay to keep increasing in time and therefore become infinite. Only for windowed flow control and for some settings in the extra back-off case, the system will be stable and therefore have finite packet delay.

2.1.2 Known results

For the uncontrolled case studies using the above-described model have already been carried out, such as in [15]. We will review these briefly.

We say that a node becomes saturated if its queue tends to infinity as time passes and that a node is stable if it does not and always returns to zero. First of all, for the system with \( n = 3 \) and \( k = 1 \), the second node is proven to become saturated, while the third node is stable. This results in a throughput of 0.7 packets/s for node 1 and 0.3 packets/s for nodes 2 and 3.

In general for \( n \geq 2k + 1 \), the first \( k + 1 \) nodes will become saturated, i.e. they have higher input than output. The rest of the nodes will be stable, making node \( k + 1 \) function as a bottleneck node and with \( \theta_i \) denoting the throughput of node \( i \) we have the following conjectures for \( n \geq 2k + 1 \).

\[
\theta_1 \geq \theta_2 \geq \ldots \geq \theta_{k+1} = \theta_{k+2} = \ldots = \theta_n,
\]

\[
\theta_{k+1} \leq \frac{1}{2k + 1}, \tag{2.1}
\]

\[
\theta_i \leq \frac{1}{2k + 1} \left(1 + \sum_{j=i}^{k} \frac{1}{j}\right), \quad 1 \leq i \leq k.
\]

These conjectures turned out to closely match the simulation results, with (2.1) providing a sharp bound to the throughput.

2.2 Windowed flow control model

When windowed flow control is applied, the first node is not always allowed to transmit a new packet into the system. Only when there are less packets in the system than the window size \( W \), the first node will be able to inject a packet into the system.

Since the number of packets in the system only decreases when a transmission is completed by the last node, this process can be designed as a closed circuit. The last node then sends packets to the first node. However, the interference structure of the one-dimensional array needs to be preserved.

As is shown in Section 5.3.1, there will be a small alteration in this model, to describe the limiting behaviour of the chain when the window size \( W \) increases to infinity. This
alteration is done to drastically lower the startup time the system needs before reaching a stable throughput.

### 2.3 Extra back-off model

To model the protocol in which extra back-off is used, only the NAV of the node that just finished a transmission needs to be altered. The node will need to be silent for the next few transmissions, so it must have an extra back-off time $t_{EB}$. This time should be in the order of $1/\mu = 1$ and intuitively it should be around $t_{EB} \approx k$, because then the node in back-off will give the packet the opportunity to get out of its blocking range before attempting to transmit another packet.

Since this form of flow control is highly experimental, we will implement a few different extra back-off schemes. We will mainly look into exponential back-off times, but we also explore a system with deterministic back-off times. Apart from the case in which a node will stay in back-off until its pre-decided back-off time has elapsed, we will also examine the slightly different model in which the back-off times are terminated by the reception of a new packet.

### 2.4 Express forwarding model

To implement express forwarding, the flow control protocol that was described in Section 1.4.3, some alterations have to be made to the previous model. In express forwarding the transmitting node fools itself and its surrounding nodes, except for the receiving node, by reserving a little extra time for a transmission. Now the receiving node can forward the packet it just got before another node can capture the channel. That is why the extra time should be in the order of $DT_0 = O(1/\lambda)$. Benveniste already gave a lower bound for this extra time for the non-simplified model, based on the length of a time slot, the length of the acknowledgement and the SIFS, among other things [6]. Since in our case the times between two transmissions are practically taken to be zero (because $\lambda \to \infty$), we do not have to implement a $DT_0$. Instead, we will let the next transmission start at the exact same time the previous one ends, so that the other nodes do not have to start their exponential back-off timer at all. This implies that the receiving node $i$ will always be able to forward the just received message, unless node $i + k$ is active when the previous transmission ended. This is because nodes $i - k - 1, i - k, \ldots, i + k - 1$ are all still silent because of the transmission that just ended, and the nodes that have to be silent for the new transmission are $i - k, i - k + 1, \ldots, i + k$, of which only $i + k$ is not in the range of the node that just transmitted the packet.

Since we assume that transmissions happen without failures, as was argued in Section 2.1, express retransmission is not implemented in our model. This protocol would only become active after a transmission has failed.

This model will not be extensively studied, but we will do simulations, as can be seen in Chapter 5. However, because of similarity between a special case of the extra back-off protocol and express forwarding, the case of $n = 3, k = 1$ is briefly analyzed in Section 4.1.
Chapter 3

Windowed flow control

In this chapter the throughput that is achieved using windowed flow control will be analyzed. We will be looking extensively at the simplest non-trivial case that still has interesting WMN properties, namely $n = 3$ and $k = 1$. In Section 3.1 we analyze the system for a few different window sizes, using Markov processes. The throughput functions are analyzed exactly. In Section 3.2 we investigate more closely what happens when the window size is increased. We see that the system tends to an equivalent system as $W \to \infty$. In Section 3.3 we analyze this equivalent system to find sharp bounds on the throughput.

For windowed flow control with general $n \geq 3$ and $k \geq 1$ we refer to Section 5.3.

3.1 Windowed flow control with finite window size

In this section we will concentrate on the case in which $n = 3$ and $k = 1$. This is a simple setting that can still be analyzed easily for certain window sizes.

The cases with either smaller $n$ or $k$ are less interesting because for $k = 0$, there is no interference. This corresponds to the well-known cases of wired communication, which have been extensively researched in the past decades. For $n = 1$, the throughput will be 1, because there is no interference either, independent of both $k$ and the window size. For $n = 2$ and $k = 1$, the two nodes will be competing for the channel fairly. At any point in time exactly one of the nodes will be active and therefore they will each be transmitting half of the time, resulting in a throughput of 0.5. Higher $k$ makes no sense for $n = 2$, because the situation will be the same. Therefore $n = 3, k = 1$ is the first case in which the situation is non-trivial but still relevant for our research.

Denteneer et al. already proved that the throughput for $n = 3$ and $k = 1$ is equal to 0.3 in the uncontrolled version of the model [15]. For window size $W = 1$, there is a single packet allowed in the system, which takes 3 time units on average to travel through the three nodes. The next packet waits for the first to leave the system before entering, so the throughput will be $\theta = \frac{1}{3}$. We will be analyzing the throughput for $W = 2, 3, 4$ in the following subsections.

3.1.1 Window size $W = 2$

In the case of $n = 3$ and $k = 1$, there are only four possible transmitting states, either the inner node, number 2, is transmitting, or one or more of the outer nodes are transmitting,
number 1 and/or number 3. We only distinguish between inner and outer node activity, because which of the outer nodes are active follows from the state of the packets.

When we take window size \( W = 2 \), the two packets can be divided over the 3 nodes in \( \binom{n+W-1}{W} = \binom{4}{2} = 6 \) ways. This number is found by looking at the problem of dividing \( W \) balls in \( n \) urns. We start by putting the \( W \) packets on a row and letting the nodes be represented only by the borders between subsequent nodes. So in the case of \( n = 3 \), \( k = 1 \) and \( W = 2 \), we have two packets to divide over 3 nodes. Therefore we take the border between nodes 1 and 2 and the border between nodes 2 and 3. If we place the first border after the first packet, it will mean that there is one packet in node 1 and the other packet is in node 2 or node 3. If the second border is placed immediately after the first border, it will result in no packets between these two borders and one packet behind the last border. This represents the state \( \{1,0,1\} \). So in general, there will be \( n-1+W \) objects in a row, of which \( W \) are packets and \( n-1 \) are the so-called borders. If the positions of the borders are determined, the state will also be fixed and vice versa. This means that we get the binomial coefficient \( \binom{n+W-1}{W} \).

For \( n = 3 \), \( k = 1 \) and \( W = 2 \), not all of these packet states allow both transmission states. When all packets are at the outer nodes, the inner node cannot be transmitting and vice versa. We will be using the notation \( i \) when the inner node is active and \( o \) for the outer nodes active. This results in states

\[
S_{3,1,2} = \{ \{x_1, x_2, x_3\}^a \mid \sum_{j=1}^{3} x_j = 2, x_j \in \mathbb{N}, a \in A(x_1, x_2, x_3) \},
\]

with

\[
A(x_1, x_2, x_3) = \begin{cases} 
\{o\}, & \text{if } x_2 = 0, \\
\{i\}, & \text{if } x_1 + x_3 = 0, \\
\{i, o\}, & \text{else},
\end{cases}
\]

where \( S_{n,k,W} \) represents the state space of the process with \( n \) nodes, blocking range \( k \) and window size \( W \). For example, the state \( \{0,1,1\}^o \) means that there are no packets at node 1, and there is one packet at both node 2 and node 3. The \( o \) indicates activity at the outer nodes (in this case only node 3). Now we find that

\[
|S_{3,1,W}| = \binom{n+W-1}{W} \cdot |\{i, o\}| - 1 - (W + 1) = 2 \cdot \binom{3+W-1}{3-1} - W - 2, \quad (3.1)
\]

for \( n = 3 \), \( k = 1 \) and general window size \( W \). Here the subtrahends represent the packet states in which only one activity state is possible, i.e. where \( x_1 + x_3 = 0 \) (1 state) or \( x_2 = 0 \) \((W + 1) \) states. We see that

\[
|S_{3,1,W}| = 2 \cdot \binom{2+W}{2} - W - 2 = (W + 2)(W + 1) - W - 2 = W^2 + 2W,
\]

which shows the quadratic behavior of the number of states as a function of the window size \( W \). For \( W = 2 \) we have:

\[
|S_{3,1,2}| = 2^2 + 2 \cdot 2 = 8 \text{ states}.
\]

Transitions from one state to another occur at the completion of a transmission. The next state only depends on the current state, which makes the system memoryless. Because of this property and since all transmission times are exponentially distributed random variables, this situation describes a Markov process.
We will start analyzing this Markov process and find the stationary distribution. This distribution exists, because the Markov process is aperiodic and has a finite state space, as has just been shown, and it is irreducible. The irreducibility is quite trivial, but can also be observed from Figure 3.1, which will be presented after the transition rates have been shown.

We see that the transition rates are given by

\[
\begin{align*}
    r_{\{0,2,0\}^c, \{0,1,1\}^c} &= 1/2, & r_{\{1,0,1\}^c, \{0,2,0\}^c} &= 1, \\
    r_{\{0,2,0\}^c, \{0,1,1\}^c} &= 1/2, & r_{\{0,0,2\}^c, \{1,0,1\}^c} &= 1, \\
    r_{\{0,1,1\}^c, \{0,0,2\}^c} &= 1, & r_{\{1,0,1\}^c, \{0,1,1\}^c} &= 1, \\
    r_{\{0,1,1\}^c, \{1,1,0\}^c} &= 1/2, & r_{\{1,0,1\}^c, \{2,0,0\}^c} &= 1, \\
    r_{\{0,1,1\}^c, \{1,1,0\}^c} &= 1/2, & r_{\{2,0,0\}^c, \{1,1,0\}^c} &= 1/2, \\
    r_{\{1,1,0\}^c, \{1,0,1\}^c} &= 1, & r_{\{2,0,0\}^c, \{1,1,0\}^c} &= 1/2. 
\end{align*}
\]

All other rates are equal to zero. The rates are found by reasoning, for example, if the process is in state \{0, 2, 0\}, it means that one packet from the second node will be transmitted to the third node in the next state, which means that we will have packet configuration \{0, 1, 1\}. The time spent in \{0, 2, 0\} is exponentially distributed with mean \(1/\mu = 1\), so the total outrate should be \(\mu = 1\). Now node 2 and node 3 will fairly compete for the next transmission opportunity, both having probability 1/2 of winning. This results in the fair division of the total rate 1 into rate 1/2 to \{0, 1, 1\} and also rate 1/2 to \{0, 1, 1\}. This explains the first two given rates. This Markov process is depicted in Figure 3.1. We will now express the stationary probabilities \(p_{\{x_1, x_2, x_3\}}\) as functions of \(p_{\{0,0,2\}}\). We will put \(p_{\{0,0,2\}} = x\). By equalizing inflow and outflow of specific nodes we can find the following
stationary probabilities respectively:

\[
\begin{align*}
P_{\{0,0,2\}}^o &= x, \\
P_{\{0,1,1\}}^i &= x, \\
P_{\{0,2,0\}}^i &= 2p_{\{0,1,1\}}^i = 2x, \\
P_{\{1,1,0\}}^o &= 2p_{\{0,2,0\}}^i = 2x.
\end{align*}
\]

Since states \(\{1,1,0\}^o\) and \(\{1,1,0\}^i\) have exactly the same inflow (although we do not know the stationary probabilities of the states that the inflows originate from) and the same rate of outflow, we have:

\[
P_{\{1,1,0\}}^i = p_{\{1,1,0\}}^o = 2x.
\]

Now we can calculate the rest:

\[
\begin{align*}
P_{\{1,0,1\}}^o &= \frac{1}{2} (p_{\{0,0,2\}}^o + p_{\{1,1,0\}}^i) = \frac{1}{2} (2x + x) = \frac{3}{2} x, \\
P_{\{0,1,1\}}^o &= \frac{1}{2} p_{\{0,2,0\}}^i + p_{\{1,0,1\}}^o = \frac{1}{2} \cdot 2x + \frac{3}{2} x = \frac{5}{2} x, \\
P_{\{2,0,0\}}^o &= p_{\{1,0,1\}}^o = \frac{3}{2} x.
\end{align*}
\]

Now because \(\sum_{s \in S_{3,1,2}} p_s = 1\), we have

\[
x = \left(1 + 1 + 2 + 2 + 2 + \frac{3}{2} + \frac{5}{2} + \frac{3}{2}\right)^{-1} = \left(\frac{27}{2}\right)^{-1} = \frac{2}{27}.
\]

Summing the stationary probabilities of all states with activity state \(a = i\), gives us the throughput of node 2 and thus of the whole system, because the throughput of every node is equal.

This gives us throughput \(\theta_{3,1,2} = \frac{4}{27} + \frac{2}{27} + \frac{4}{27} = \frac{10}{27} \approx 0.370370\) packets/unit of time.

In windowed flow control at any time there are \(W\) packets in the system. From the window size and the throughput we can now calculate the average delay. This is

\[
t_{3,1,2} = \frac{W}{\theta_{n,k,W}} = \frac{2}{\frac{10}{27}} = \frac{27}{5} = 5.4\text{ units of time.}
\]

### 3.1.2 Window size \(W = 3\)

For \(n = 3\), \(k = 1\) and window size \(W = 3\), the same approach is taken as for \(W = 2\). The size of the state space increases. Now we have \(|S_{3,1,3}| = 3^2 + 2 \cdot 3 = 15\) states.

Because of the increase in the number of states, the number of nonzero transition rates between states is also increased. Instead of the 12 we found for \(W = 2\), we now have 25 nonzero rates for \(W = 3\). Therefore the situation is shown in Figure 3.2, including the transition rates.

We will again find the stationary distribution by equalizing the inflow and the outflow for the nodes. This time we set \(p_{\{0,0,3\}}^o = 15x\), \(p_{\{1,1,1\}}^i = 33y\) and \(p_{\{2,0,1\}}^o = 16z\), in order to solve all equations. These numbers are chosen to avoid having too many fractions in the calculations. The complete deduction is shown in Section A.1.
Figure 3.2: The Markov process corresponding to $n = 3$, $k = 1$, $W = 3$
It turns out that \( x = y = z \) and that the throughput equals
\[
\theta_{3,1,3} = \frac{206}{537} \approx 0.383613...
\]
This means another improvement in the achieved throughput, compared to \( W = 2 \). We see that the average delay is increased to
\[
t_{3,1,3} = \frac{3}{206} \cdot \frac{1611}{206} \approx 7.82 \text{ units of time.}
\]

### 3.1.3 Window size \( W = 4 \)

For \( n = 3, k = 1 \) and window size \( W = 4 \), we can once again construct a Markov process to find the throughput. This time we have \( |S_{3,1,4}| = 4^2 + 2 \cdot 4 = 24 \) states.

Again the goal is to find the stationary distribution so that we can determine the throughput from that. We now put \( p_{0,0,4} = 1381x, p_{1,2,1} = 7413y \) and \( p_{1,1,2} = 3191z \). Again these values are taken to avoid most of the fractions and so that it will turn out that \( x = y = z \).

The complete calculations can be found in Appendix A.2.

The outcome is that the throughput is given by
\[
\theta_{1,3,4} = \frac{43666}{111987} \approx 0.389920...
\]
Again, this is an improvement. The average delay is increased though, to:
\[
t_{1,3,4} = \frac{4}{43666} \cdot \frac{447948}{43666} \approx 10.26 \text{ units of time.}
\]

It is worth mentioning that it hardly happens that the second queue becomes empty. We can calculate the stationary probability for this event by adding the stationary probabilities of all states in which \( x_2 = 0 \):
\[
P\{x_2 = 0\} = (1381 + 2286 + 1434 + 956 + 956)x = \frac{7013}{111987} \approx 0.063.
\]

We can compare this to
\[
P\{x_1 = 0\} = \frac{11048 + 5524 + 12937 + 2762 + 6670 + 1381 + 3667 + 1381}{111987} = \frac{45370}{111987} \approx 0.405,
\]
\[
P\{x_3 = 0\} = \frac{11048 + 11048 + 11048 + 9159 + 3492 + 1746 + 956 + 478}{111987} = \frac{48975}{111987} \approx 0.437,
\]
and see that there is indeed a major difference. The outer nodes have an empty queue far more often than the inner node.

For \( W > 4 \), the situation becomes harder and harder to analyze, because of the increase in the number of states and possible transitions. For \( W = 5 \) we see that we have \( 5^2 + 2 \cdot 5 = 35 \) states and since from every state 1 or 2 outgoing transitions are possible, there are between 35 and 70 transitions. For \( W = 6 \) the number of states increases to \( 6^2 + 2 \cdot 6 = 48 \) and the calculations will probably continue to become more and more complex.
3.1. Windowed flow control with finite window size

Figure 3.3: The Markov process corresponding to $n = 3$, $k = 1$, $W = 4$
Chapter 3. Windowed flow control

3.2 Windowed flow control with window size tending to infinity

In this subsection the limiting behavior of increasing window size \(W\) will be examined. We will show that the system with \(W \to \infty\) is equivalent to another system, in which not the first but the second node is saturated. The main part of this section will be proving the stability of this second system, which is found in Appendix B.

3.2.1 Increasing the window size

What we have seen so far is that the throughput \(\theta_{3,1,W}\) continues to increase as the window size \(W\) increases. This brings us to the following conjecture for general \(n\) and \(k\):

**Conjecture 1.** For fixed \(n\) and \(k \leq \frac{n-1}{2}\), the throughput sequence of a system with windowed flow control \(\{\theta_{n,k,W}\}_{W=1}^{\infty}\) is monotonically non-decreasing sequence, or:

\[
\theta_{n,k,W+1} \geq \theta_{n,k,W} \quad \forall W \in \mathbb{N}.
\]

**Argument.** We have seen this behavior for \(W = 0, 1, ..., 4\), with the throughputs given in the following table.

<table>
<thead>
<tr>
<th>(W)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\theta_{3,1,W})</td>
<td>0.333</td>
<td>0.370</td>
<td>0.384</td>
<td>0.390</td>
</tr>
</tbody>
</table>

An intuitive argument is that when one packet is added to a system with \(n\) nodes, blocking range \(k\), and window size \(W\), the nodes have more work to do. Since a system with windowed flow control is stable, each transmission helps to improve the throughput. Every packet must undergo \(n\) transmissions and after these the packet leaves the system. On average it holds that the more transmissions happen at the same time, the more work is done and the higher the throughput will be.

If a packet is added to the system, it will result in a higher packet occupation of the nodes. This means that a node that has the opportunity to start a transmission, because it is not blocked by a node in its blocking range, will have an empty queue less often. Then instead of being idle it will grab the opportunity and transmit a packet. Since every transmission is a contribution to the total amount of work done and therefore to the throughput, this will indeed result in a higher throughput.

The catch here is that an extra transmission can also block other transmissions that would have started if this node had stayed inactive. We believe this disadvantage is not large compared to the advantage.

In the case of \(n = 3\), \(k = 1\) and small window sizes it seems that an extra packet will reside in the queue of node 2 most of the time when the window size is incremented by one. We can compare the fractions of the time that queue 2 is empty to the fractions that either of the other two nodes are empty. These numbers are given in the table below. They follow from the stationary distributions as calculated in the previous sections.

<table>
<thead>
<tr>
<th>(W)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P_W{x_1 = 0})</td>
<td>0.667</td>
<td>0.481</td>
<td>0.428</td>
<td>0.405</td>
</tr>
<tr>
<td>(P_W{x_2 = 0})</td>
<td>0.667</td>
<td>0.296</td>
<td>0.132</td>
<td>0.063</td>
</tr>
<tr>
<td>(P_W{x_3 = 0})</td>
<td>0.667</td>
<td>0.556</td>
<td>0.471</td>
<td>0.437</td>
</tr>
</tbody>
</table>
3.2. Windowed flow control with window size tending to infinity

It seems that the stationary probability of finding an empty queue at the second node is shrinking with a factor a little larger than 1/2. This means that for increasing $W$ when node 2 starts transmitting it will not reach emptiness too often anymore before yielding to the outer nodes. Since the queue of node 2 does not go empty too often, it will keep the possibility to send another packet a larger share of the time. This results in node 2 having the opportunity to send more packets subsequently, so that the queue of node 3 will be filled further on average before it starts transmitting. Node 3 will transmit a packet to node 1, so that they can transmit packets together. The more packets there are in node 3, the higher the stationary probability becomes that nodes 1 and 3 are transmitting at the same time, resulting in higher throughput. We see in the table above that indeed nodes 1 and 3 are empty less often as $W$ increases, although they seem to approach some positive value, around 2/5.

Another noteworthy thing is that for general $k$ there is always an upper bound on the throughput equal to $\frac{1}{k+1}$.

**Lemma 1.** In any mesh chain with $n$ nodes and blocking range $k \leq n - 1$ there is an upper bound to the throughput equal to

$$\theta_{n,k} \leq \frac{1}{k+1}.$$ 

**Proof.** To find this upper bound we will be looking at a single packet starting at node 1. During the transmission of this packet to node 2, nodes 2, 3, ..., $k + 1$ must be inactive. Since $k \leq n - 1$, we have that $n \geq k + 1$, so all these nodes indeed exist.

Every packet requires $k + 1$ (possibly not consecutive) transmission times in the first $k + 1$ nodes, in which no other packet in this part of the system can be transmitted. Therefore every packet monopolizes the first $k + 1$ nodes for $k + 1$ transmission times. This means that on average at most one packet can leave the first $k + 1$ nodes per $\mu$ time units. So, with $\mu = 1$, the throughput can at most be $\frac{1}{k+1}$.

This means that for the case of $k = 1$, we can never achieve a throughput above $\frac{1}{2}$. Assuming Conjecture 1 is true, we see that the throughput is a non-decreasing function of $W$, limited by the upper bound $\theta_{n,k,W} \leq \frac{1}{k+1}$.

A monotonically non-decreasing sequence with an upper bound always converges to a certain limit. We want to investigate this limit for $n = 3$, $k = 1$. As mentioned in the argument of Conjecture 1, when we increase the window size, the assumption is that the packets will spend most of their waiting time at the second node, resulting in a larger queue for node 2.

With some intuitive arguments, we can also see that this is the case. We assume a large window size. When starting in the state where all packets are in the second node, this node will start transmitting packets to node 3. Its active time is then practically geometrically distributed with mean 2, because for every next transmission, the second node will have to compete with only the third node and win with probability $\frac{1}{2}$ every time. Of course, this can at most go on until all packets have been transmitted to node 3. This means that this is a truncated geometric distribution with mean a little less than 2. If the window size is taken large enough, it will approach the geometric distribution.

Once the third node has won, it will transmit one of its packets, allowing node 1 immediately after completion to accept a new entering packet. This can be seen as node 3 transmitting to node 1. If node 3 had more than one packet, the three nodes will all be competing with each other, resulting in the outer nodes winning in $\frac{2}{3}$ of the time. After the
outer nodes started transmitting, they will stay active until one has its queue empty, after which node 2 might grab another transmission opportunity. Since node 1 is on average receiving just as many packets as it is transmitting, both the events of receiving and transmitting happen at rate 1, it will probably be node 3 that reaches an empty queue first. If not, we are again in a state in which both nodes 2 and 3 have packets to transmit and will compete together, just like the situation after the very first transmission from node 2 to node 3. If node 3 does become empty, node 1 will keep transmitting for a geometrically distributed time with mean 2. If node 2 starts transmitting again before the queue of node 1 is empty, the process will also be reinitiated with the difference that node 2 has more competition. The geometric distribution will now have a mean of $\frac{3}{2}$ only. This shows that node 2 is always at a disadvantage to empty its queue, compared to either one of nodes 1 and 3. The stationary probability that the second queue becomes empty will remain positive, but as the window size increases, this probability will become smaller.

### 3.2.2 System with a saturated second node

The system that we will be looking at next is the cyclic mesh chain comprising three nodes, where the second node is saturated. Just as in the windowed flow control systems, either only one of the three nodes is active or the first and the third are active simultaneously. The second node now functions both as a source and a sink of packets. The saturation can be restated as having an infinite amount of packets at the second node, which already shows some similarity with the ‘infinite window size’. Since the notion of windowed flow control is only defined for finite $W$, we need to be careful talking about an infinite window size.

We will first show that the system with the second node saturated is stable i.e. that the queues of the first and third nodes do not tend to infinity.

**Theorem 1.** Consider a cyclic mesh chain comprising 3 nodes, in which the second node is saturated. If $k = 1$, so that either a single node may be active or both nodes 1 and 3 simultaneously, the queues at nodes 1 and 3 will be stable.

Now, to prove stability of queues 1 and 3, it suffices to show that when the sum of all packets at nodes 1 and 3 is above some value, it is expected to decrease within a certain time frame. This means that the drift of the number of packets is negative when this number is above some value. In other words, when the total number of packets at nodes 1 and 3 is outside of a certain bounded region $E_0$, it is always expected to return to this region within finite time and therefore the queues of both nodes 1 and 3 will not tend to infinity. This is essentially the implication of Proposition 1, which we will use to prove stability. This proposition is found in Section 1.5 of [3]:

**Proposition 1** (Foster-Lyapunov criteria). Suppose a Markov chain with transition probabilities $p_{ij}$ is irreducible and let $E_0$ be a finite subset of the state space $E$. Then the chain is positive recurrent if for some $h : E \rightarrow \mathbb{R}$ and some $\varepsilon > 0$ we have $\inf_x h(x) > -\infty$ and

$$
\sum_{k \in E} p_{jk}h_k < \infty, \quad j \in E_0, \quad \quad (3.2)
$$

$$
\sum_{k \in E} p_{jk}h_k \leq h(j) - \varepsilon, \quad j \notin E_0. \quad \quad (3.3)
$$
To apply this proposition to our case we have to do multiple things. First we have to embed the continuous-time Markov process at discrete-time moments. With \( h(x_1, x_3) = x_1 + x_3 \), it will turn out that (3.2) is quite trivially satisfied, while (3.3) needs much more arguing. The full proof is found in Appendix B.

### 3.2.3 Equivalence of both systems

We define \( p_{\{x_1, x_3\}}^a \) as the stationary probability of finding \( x_1 \) packets at node 1, \( x_3 \) packets at node 3, and activity state \( a \in \{i, o\} \), at arbitrary moments of the process. Here \( a = i \) means activity at the second node and \( a = o \) means activity at the first and/or third node. Because of the recurrence of the system, these probabilities exist and are positive.

On the other hand we define \( p_W^{\{x_1, x_3\}} \) as the stationary probability of the state in the system with \( n = 3 \), \( k = 1 \) and window size \( W \), in which there are \( x_1 \) packets at node 1, \( x_3 \) packets at node 3 and the system is in state \( a \), with \( a \in \{i, o\} \). This defines the state since it must now hold that \( x_2 = W - x_1 - x_3 \).

To show that the system with windowed flow control in which \( W \to \infty \) is equivalent to the system with the second node saturated, it suffices to show that for all \( x_1 \) and \( x_3 \) and \( a = i, o \):

\[
\lim_{W \to \infty} p_W^{\{x_1, x_3\}} = p_{\{x_1, x_3\}}^a.
\]

And equivalently, embedded at the start of the cycles we would need:

\[
\lim_{W \to \infty} p_W^{\{x_1, x_3\}} = p_{\{x_1, x_3\}}.
\]

We come to the following conjecture:

**Conjecture 2.** The system in which \( n = 3 \), \( k = 1 \) with window size \( W \to \infty \) is equivalent to a mesh circuit with preserved interference structure in which only the second node is saturated.

**Argument.** The arguments leading up to Theorem 1 already suggest the equivalence of both systems. It shows that in a system with an infinite number of packets at the second node the queues of nodes 1 and 3 remain stable. In the system with window size \( W \) and \( W \to \infty \), we see that if there are \( x_1 \) packets at node 1 and \( x_3 \) packets at node 3, node 2 must have \( x_2 = W - x_1 - x_3 \) packets.

While deriving the expected net gain in packets, similarly as was done in the proof, the only differences with the mesh chain in which the second node is saturated work in favor of the system with \( W \to \infty \). The geometric distributions that determine the number of packets transmitted at the start of cycle are now truncated, lowering the expected number of packets transmitted. Because node 2 can be empty after its transmission period, the competition of nodes 1 and 3 to keep the channel is less fierce, resulting in a higher number of packets transmitted back to node 2. This argument also holds for an increasing \( W \), or we may say that \( p_W^{\{x_1, x_3\}} \) is stochastically increasing in \( W \) and for all \( W \) they are stochastically dominated by \( p_{\{x_1, x_3\}} \). This is conform the results we proposed in the argument of Conjecture 1.

This suggests that the queues at node 1 and node 3 remain finite as \( W \to \infty \), while \( x_2 = W - x_1 - x_3 \to \infty \).

So, as \( W \) tends to infinity, the probability of finding the second queue empty at a random time will tend to zero. The probability of finding either of the other queues empty will remain positive. The just presented situation with an infinite supply of packets at the second node will be analyzed in the following section.
3.3 Windowed flow control with infinite window size

To model the limiting behavior of having a large window on a chain of three nodes, we now put an infinite pile of packets at the second node and make the last node transmit its packets to the first node, as was described in Section 3.2.

We will be deriving an upper bound for the throughput in two manners. First we will derive a loose bound by analyzing cycles. A cycle is assumed to start when node 2 grabs the channel after transmissions by the outer nodes. This derivation is short and straightforward. After this approach the bound will be sharpened by a more extensive and more complicated analysis of the Markov process that describes this system.

3.3.1 Upper bound through cycle analysis

Let us say that a cycle starts when node 2 begins a transmission after either node 1 or node 3 ends a transmission. Then the throughput of node \( i \) may be expressed as

\[
\theta_i \equiv \frac{\mathbb{E}\{N_i\}}{\mathbb{E}\{C\}},
\]

with \( N_i \) denoting the number of packets transmitted by node \( i \) during an arbitrary cycle and \( C \) the duration of an arbitrary cycle.

We first consider \( \mathbb{E}\{N_i\} \). By virtue of the stability of nodes 1 and 3, we have \( \mathbb{E}\{N_1\} = \mathbb{E}\{N_2\} = \mathbb{E}\{N_3\} \). We now distinguish two cases, depending on whether node 1 is empty at the start of a cycle or not: (i) In case node 1 is empty at the start of the cycle, the number of transmissions of node 2 is \( N_2 = 1 + \text{Geo}(1/2) \), with expected value \( \mathbb{E}\{N_2\} = 2 \); (ii) In case node 1 is not empty at the start of the cycle, the number of transmissions of node 2 is \( N_2 = 1 + \text{Geo}(2/3) \), with expected value \( \mathbb{E}\{N_2\} = 3/2 \). Here we assume the support of the geometric distribution to include 0. Denoting by \( p_0 \) the fraction of cycles that start with node 1 empty, we obtain \( \mathbb{E}\{N_2\} = 2p_0 + 3/2(1 - p_0) = 3/2 + 1/2p_0 \).

We now turn to \( \mathbb{E}\{C\} \). The average duration of a cycle may be expressed as

\[
\mathbb{E}\{C\} = \mathbb{E}\{T_2\} + \mathbb{E}\{T_1\} + \mathbb{E}\{U_1\} = \mathbb{E}\{T_2\} + \mathbb{E}\{T_3\} + \mathbb{E}\{U_3\},
\]

with \( T_i \) denoting the amount of time that node \( i \) is transmitting, and \( U_i \) the amount of time that node \( i \) is silent, while node \( 4 - i \) is transmitting, \( i = 1, 3 \). Note that \( \mathbb{E}\{T_i\} = \mathbb{E}\{N_i\} = 3/2 + 1/2p_0 \).

As before, we distinguish two cases, depending on whether node 1 is empty at the start of a cycle or not.

(i) In case node 1 is empty at the start of the cycle, node 3 must be as well. In order to give node 2 a chance to grab the channel, it had either just finished a transmission, in which case node 1 could not have been empty, or it was already empty. This means that after the transmission period of node 2, there are exactly \( 1 + \text{Geo}(1/2) \) packets at node 3.

Now node 1 will be silent for an expected amount of time 1 until node 3 completes its first transmission. Since the remaining number of packets at node 3 is geometrically distributed with parameter 1/2, node 3 will then still have a packet to transmit with probability 1/2. With probability 2/3, nodes 1 and 3 will then win the competition with node 2 for the channel, and with probability 1/2 node 1 will complete its transmission before node 3 does, in which case it will again be silent for an expected amount of time 1. Multiplying the probabilities
3.3. Windowed flow control with infinite window size

gives 1/6 chance of this happening. This can be repeated, so with probability \((1/6)^2\) there will be another silence period of length 1, and so on. This results in an expected silent time of at least \(\sum_{j=0}^{\infty} (1/6)^j = 6/5\).

(ii) In case node 1 is not empty at the start of the cycle, both nodes 1 and 3 begin transmitting. We distinguish two further scenarios, depending on whether node 1 or node 3 is the first one to run out of packets and stop transmitting.

(iia) In case node 1 stops transmitting first, it will be silent for an expected amount of time 1, while node 3 completes its transmission (and passes a packet to node 1). If node 3 is empty at that point, then node 1 will win the competition with node 2 for the channel with probability \(1/2\), and node 3 will be silent for a further expected amount of time 1. If node 3 is not empty at that point, then nodes 1 and 3 will win the competition with node 2 for the channel with probability \(2/3\), in which case we are back in the original situation.

(iib) In case node 3 stops transmitting first, it will be silent for an expected amount of time 1, while node 1 completes its transmission, and at that point must be non-empty because of the packet it received from node 3. Hence, node 1 will win the competition with node 2 for the channel with probability \(1/2\), and node 3 will be silent for a further expected amount of time 1.

Denote by \(p_a\) and \(p_b\) the fractions of cycles starting with node 1 non-empty, and either node 1 or node 3 the first one to stop transmitting, so that \(p_0 + p_a + p_b = 1\). Then we obtain that \(\mathbb{E}\{U_1\} \geq 6/5p_0 + p_a\) and \(\mathbb{E}\{U_2\} \geq 1/2p_a + 3/2p_b\), and find that the throughput may be bounded from above by

\[
\theta_{3,1,\infty} \leq \frac{3/2 + 1/2p_0}{3 + p_0 + \max\{6/5p_0 + p_a, 1/2p_a + 3/2p_b\}}.
\]

Maximizing the latter expression subject to the constraint \(p_0 + p_a + p_b = 1\) yields

\[
\theta_{3,1,\infty} \leq 8/19 \approx 0.42105,
\]

for \(p_0 = 5/9\), \(p_a = 0\), and \(p_b = 4/9\).

3.3.2 Upper bound through Markov process analysis

The Markov process describing the cyclic mesh system in which node 2 is saturated is depicted in Figure 3.4. We see that the size of the state space grows to infinity in two dimensions. Even though there is a clear pattern visible, this is not easily analytically solvable. We therefore present an upper bound looking closely at the process. This is done in Appendix C.

The result of this analysis is that

\[
\theta_{3,1,\infty} \leq \frac{557}{1358} \approx 0.41016 \text{ packets per time unit.}
\]

This is indeed sharper than (3.6). Using Conjecture 1, we can find a lower bound of 0.389920 from the case of \(W = 4\). This results in the bounds

\[0.389920 < \theta_{3,1,\infty} < 0.41016.\]
Figure 3.4: The Markov process corresponding to $n = 3$, $k = 1$, $W = \infty$
We see that the throughput is roughly $4/5$ of the theoretical upper bound for system with $k = 1$ and $n \geq 3$ in general, as derived in Lemma 1. From simulations done in Chapter 5 it turns out that $\theta_{3,1,\infty} \approx 0.396$, which indeed is between the two bounds.

Denteneer et al. found that the throughput of the uncontrolled system was equal to 0.3 [15], so windowed flow control gives a clear improvement of about 33%.

As was already argued, the system with $W = \infty$ is not useful to implement, but it does describe the limiting behavior when the window size is increased. Evaluation of the other measure we have discussed, the average waiting time, is not too useful in this case either. With a finite throughput and an infinite window size, the waiting time will be infinite too. As the window size increases, the throughput increases as well, but relatively not as fast, resulting in an increasing waiting time too. When the throughput is the main objective to improve, a balance should be found between a better throughput and a worse average waiting time. When the throughput of a system is close enough to the limit, but the waiting time is still small enough, the window size is seen as optimal.
Chapter 4

Extra back-off

In this chapter we will extensively look at the flow control mechanism that gives an extra back-off time to a node after it has transmitted a packet. Just as we did for windowed flow control, we will concentrate on the case with \( n = 3 \) and \( k = 1 \). It turns out that this case is already non-trivial.

We first introduce four back-off schemes and present some preliminaries in Section 4.1. After that we will discuss theory about a matrix-geometric solution to the various back-off schemes in Section 4.2. Using this theory we analyze the basic extra back-off scheme with exponential back-off times in Section 4.3. We will then discuss a simplification of this basic scheme in Section 4.4, where the third node does not have any back-off. After that we look into a new back-off scheme, in which back-off times are terminated by the reception of a new packet in Section 4.5. Finally, in Section 4.6 we will investigate the system in which back-off times are deterministic and terminated by the reception of a new packet, which is followed by a comparison and an overview of the chapter in Section 4.7. The work in Sections 4.1 to 4.5 was done in collaboration with Johan S.H. van Leeuwaarden.

4.1 Multiple back-off schemes and preliminaries

We still use the basic model as described in Section 2.1, but this time we employ extra back-off and to this extent use the small modification described in Section 2.3. Since we work in the regime with \( \lambda \to \infty \), we will not talk about extra back-off, but about back-off in general.

The distinguishing feature is that it incorporates a form a flow control by imposing back-offs. That is, after a transmission has ended, a node is forced to go into back-off for a certain period of time. We assume this back-off period to be exponentially distributed with mean \( \eta \) and rate \( \beta = 1/\eta \). During back-off, a node is not allowed to start a transmission, but it can still receive packets. We study back-offs since it is a flow control scheme that requires no additional communication between the nodes. Adding the back-off mechanism to existing distributed protocols requires only a small modification, which makes this very easy to implement.

We are mainly interested in the throughput, defined as the expected number of packets transmitted per unit of time. Denote the throughput of node \( i \) by \( \theta_i(\eta) \). In Figure 4.1 the throughput is shown as a function of the mean back-off period \( \eta \). This figure is generated using Matlab calculations, as described in Section 4.2. The upper curve depicts the throughput of node 1 and the lower curve is the throughput of nodes 2 and 3. The dotted line is \( \tau(\eta) \), a
Figure 4.1: Plots of $\theta_1(\eta)$ (upper curve) and $\theta_2(\eta)$ (lower curve). We clearly see that the dotted line, $\tau(\eta)$, in between these curves.

function that will be introduced in Theorem 1. Since we observe that $\theta_2(\eta) = \theta_3(\eta)$, we can conclude that node 3 is always able to process every packet it receives from node 2, on the long run, and thus remains stable.

For $\eta = 0$ our model reduces to the standard model without the back-off flow control, which was treated in Denteneer et al. [15], and in which case throughput of nodes 1 and 2 equal 0.7 and 0.3, respectively. This model is inherently different. If $\eta > 0$, a node that completes a transmission will never directly compete for the channel again, but will give priority to the next node.

We will briefly discuss the case in which $\eta \downarrow 0$. This is a different case than the one with $\eta = 0$, as long as $\lambda/\beta \to 0$. For $n = 3$ and $k = 1$, this back-off scheme is in fact equal to express forwarding, the flow control scheme discussed in Section 2.4. Indeed, once node 1 has transmitted a packet, it will go into back-off for an infinitesimal period of time, but enough to give node 2 the opportunity to grab the channel (which it only does if node 3 is silent at that point). Node 2 in turn gives node 3 priority over itself, and node 3 has just received a packet, so it can always start right away. After this infinitesimal time, nodes 1 and 2 should compete for the channel. However, since node 3 just started, node 2 is now blocked, so it will not start a transmission, whether it has won the competition from node 1 or not. This shows that the schemes are identical for $\eta \downarrow 0$. Since the extra back-off protocol allows other values of $\eta$ as well, we see for $n = 3$ and $k = 1$ that

$$\theta_{EF} \leq \theta_{EB},$$

(4.1)

where $\theta_{EF}$ denotes the throughput of express forwarding and $\theta_{EB}$ denotes the highest achievable throughput of extra back-off.

We note that node 3 always has either 0 or 1 packet(s) in its buffer, since after the transmission of node 2 it can always forward the packet right away. This means when node 2
is transmitting a packet, it finishes after an exponential time with mean 1. Then nodes 1
and 3 will grab the channel, and after an exponential time with mean 1 node 3 will be finished,
after which node 3 will on average need one more time unit before it gives the channel back
to node 2. This results in \( \lim_{\eta \downarrow 0} \theta_1(\eta) = 2/3 \) and \( \lim_{\eta \downarrow 0} \theta_2(\eta) = 1/3 \). Comparing this to the
results in [15], we indeed observe a discontinuity at the point \( \eta = 0 \). We will therefore assume
\( \eta > 0 \) throughout this chapter.

4.1.1 Different back-off schemes
We will describe three back-off schemes with exponential back-off times. Our basic back-off
scheme prescribes that (i) all nodes obey an exponential back-off period after transmitting
a packet. We consider two slightly modified back-off schemes: (ii) only nodes 1 and 2 have
an exponential back-off, while node 3 has no back-off, this scheme will show a very similar
throughput to scheme (i), but is easier to analyze. We make an improvement on scheme (i)
as we propose scheme (iii), in which all nodes have exponential back-offs, but the back-off
period of a node is terminated when a new packet arrives. We refer to this third back-off
scheme as truncated back-off.

We will see that back-off scheme (i) remains difficult to analyze, while the throughput of
scheme (ii) is a very accurate approximation to the throughput of scheme (i). Scheme (ii)
turns out to be easier to analyze and we will find an exact solution for the throughput.

Scheme (iii) will provide a higher throughput for node 2, and it even offers stability at
node 2 for \( \eta \) large enough. This is an improvement on scheme (i).

Eventually, we will briefly examine the case of truncated back-off, in which the back-
off times are deterministic and have length \( \eta \). As opposed to its exponential variant, for
deterministic truncated back-off times, no stability can be achieved at node 2.

4.1.2 Preliminaries
Let us start by an observation regarding the throughput of node 1. Every packet that is
transmitted by node 1 requires a total time that consists of a transmission time \( T_1 \), a back-off
time \( B_1 \), and a possible waiting time \( W_1 \) during which node 1 is neither active nor in back-off,
but waiting for node 2 to finish its transmission. This gives the following auxiliary result.

**Proposition 2** (throughput node 1).

\[
\theta_1(\eta) = \frac{\mathbb{E}\{T_1\}}{\mathbb{E}\{T_1\} + \mathbb{E}\{B_1\} + \mathbb{E}\{W_1\}} = \frac{1}{1 + \eta + \mathbb{E}\{W_1\}}
\]  

(4.2)

with

\[
\mathbb{E}\{W_1\} = \mathbb{P}[\text{node 2 is active when node 1 comes out of back-off}] \mathbb{E}\{T_2\}.
\]  

(4.3)

This proposition is a direct consequence of Little’s law. Equation (4.3) follows from
the fact that the transmission time of node 2 is memoryless. We note that the expected
remaining duration of the transmission of node 2 is 1, because of the memoryless property of
the exponential distribution. We see that \( \theta_1(\eta) \) depends on \( \mathbb{E}\{W_1\} \), which in turn depends
on \( \theta_2(\eta) \). The challenge is to find an exact expression for \( \mathbb{E}\{W_1\} \).

Recall that we call a node *i saturated* when its buffer content grows without bound and
for buffer size \( X_i \) it holds that \( \mathbb{P}[X_i = 0] \to 0 \) as time progresses, and we call a node *stable*
if it does not.
Proposition 3 (stability and saturation scheme (i)). When all nodes have exponential back-offs with mean $\eta$, we have $\theta_1(\eta) > \theta_2(\eta) = \theta_3(\eta)$, for all $\eta > 0$, which implies that node 2 is saturated. Node 3 will be stable.

Proof. See Section 4.3.2.

Proposition 3 follows from Figure 4.1 upon visual inspection. A more formal proof is presented in Section 4.3, and uses the theory of quasi-birth-death processes and matrix-analytic techniques, proposed in Section 4.2. A side-result of this proof is given below.

Corollary 1 (ordering of throughputs scheme (i)). When all nodes have exponential back-offs with mean $\eta$, there is the relation $\theta_1(\eta) > \tau(\eta) > \theta_2(\eta)$ with

$$\tau(\eta) = \frac{1}{1 + \eta + \frac{1}{1+\eta}}.$$  (4.4)

It turns out that an interesting relation holds for all exponential back-off schemes, presented in Theorem 2.

Theorem 2. In the 3-node system in which the nodes have exponential back-off times, either truncated, non-truncated or only nodes 1 and 2 having back-off, we have

$$\theta_1(\eta)\mathbb{E}\{W_1\} = \theta_2(\eta) \frac{1}{1 + \eta},$$  (4.5)

for all $\eta > 0$.

Proof. The relation (4.5) can be rewritten as

$$\mathbb{E}\{W_1\} = \frac{\theta_2(\eta)}{\theta_1(\eta)} \cdot \frac{1}{1 + \eta}.$$  (4.6)

Here the meaning of $\frac{\theta_2(\eta)}{\theta_1(\eta)}$ is the average number of packets transmitted by node 2 per packet transmitted by node 1. This is also the expected number of transmissions of node 2 in between two successive transmissions of node 1. We now have

$$\mathbb{E}\{W_1\} = \mathbb{E}\{X\} \mathbb{E}\{Y\},$$

with $\mathbb{E}\{X\} = \frac{\theta_2(\eta)}{\theta_1(\eta)}$ the expected number of transmissions of node 2 in between two successive transmissions of node 1, and $\mathbb{E}\{Y\}$ the expected waiting time of node 1 caused by a transmission of node 2, excluding its own back-off time. This gives us

$$\mathbb{E}\{Y\} = \mathbb{E}\{\max\{T_2 - B_1, 0\}\} = \mathbb{E}\{\max\{T_2, B_1\}\} - \mathbb{E}\{B_1\}.$$

We note that

$$\mathbb{E}\{\max\{\text{Exp}(\beta), \text{Exp}(1)\}\} = \mathbb{E}\{\text{Exp}(\beta) + \text{Exp}(1) - \min\{\text{Exp}(\beta), \text{Exp}(1)\}\} = \eta + 1 - \frac{\eta}{1 + \eta},$$

and thus

$$\mathbb{E}\{Y\} = \mathbb{E}\{\max\{T_2, B_1\}\} - \mathbb{E}\{B_1\} = \eta + 1 - \frac{\eta}{1 + \eta} - \eta = \frac{1}{1 + \eta}.$$

This results in the following expected waiting time of node 1

$$\mathbb{E}\{W_1\} = \mathbb{E}\{X\} \mathbb{E}\{Y\} = \frac{\theta_2(\eta)}{\theta_1(\eta)} \cdot \frac{1}{1 + \eta}.$$  \qed
4.2 A matrix-geometric solution

Our model with the basic back-off scheme described in Section 4.1 falls into the class of two-dimensional Markov processes, called quasi-birth-death (QBD) processes, whose transitions are skipfree to the left and to the right, with no restrictions upward or downward, in the two-dimensional lattice. The invariant distributions of QBD processes, under appropriate conditions, are well known to have a matrix-geometric form. More precisely, the stationary probability vector has a geometric solution in terms of a so-called rate matrix $R$. In this section we first present the general theory for QBD processes. In Section 4.3 we will describe the model with the basic back-off scheme (i) in terms of a QBD process, which will also be done later for the other two exponential back-off schemes.

Consider a continuous-time Markov process $\{X(t), t \in \mathbb{R}_+\}$ on the two-dimensional state space $\{(i, j) : i \in \mathbb{Z}_+, j \in \{1, \ldots, M\}\}$, which is partitioned as $\bigcup_{i=0}^{\infty} l(i)$, where $l(i) = \{(i, 1), (i, 2), \ldots, (i, M)\}$ and $\mathbb{Z}_+, \mathbb{R}_+$ denote the nonnegative integer and real numbers. The first coordinate $i$ is called the level and the second coordinate $j$ is called the phase of state $(i, j)$, with the set $l(i)$ referred to as level $i$. Each level has a finite number of states, $M$.

This Markov process is called a QBD process when its one-step transitions from each state are restricted to states in the same level or in the two adjacent levels, and a homogeneous QBD process when these transition rates are additionally level-independent for levels $l(i)$ with $i > 0$.

Let $\pi$ denote the stationary probability vector of this homogeneous QBD process. We construct $\pi$ by concatenating subvectors $\pi_i$, $i \in \mathbb{Z}_+$, where $\pi_i$ has $M$ components corresponding to the states of $l(i)$. This shows that vector $\pi$ is of infinite size. We shall assume throughout the chapter that the QBD process is irreducible and ergodic. Hence, we assume the stationary probability vector exists and therefore is uniquely determined as the solution of

$$
\pi_0 B + \pi_1 A_0 + \pi_2 A_2 = 0, \quad i \geq 1.
$$

(4.7)

where matrices $A_0, A_2$ are nonnegative and matrices $A_1, B$ have nonnegative off-diagonal elements and strictly negative diagonals. Matrix $A_0$ represents the transition rates from a level $i-1$ to $i$, while $A_1$ represents transitions within the same level and $A_2$ shows transitions from level $i$ to level $i-1$. Matrix $B$ serves as the rates within level $l(0)$. In our study the matrices all have dimension $M \times M$.

The infinite sized generator $Q$ of the Markov process now takes the block tridiagonal form

$$
Q = 
\begin{pmatrix}
B & A_0 \\
A_2 & A_1 & A_0 \\
& A_2 & A_1 & A_0 \\
& & & \ddots
\end{pmatrix}
$$

(4.9)
and thus (4.7), (4.8), and the fact that the sum of all stationary probabilities must equal 1 reduce to

\[
\begin{align*}
\pi Q &= 0, \\
\pi e^T &= 1,
\end{align*}
\]

where \(e\) denotes a row vector of appropriate dimension containing all ones. The matrix-

geometric solution of the stationary probability vector \(\pi\) partitioned into \(\pi_i, i \geq 0\), is given

by the following theorem.

**Theorem 3.** Consider a continuous-time QBD process with infinitesimal generator \(Q\) in the

form of (4.9). Suppose that the QBD process is irreducible and ergodic. Then its stationary

distribution \(\pi\) is given by

\[
\pi_i = \pi_0 R^i, \quad i \in \mathbb{N},
\]

(4.10)

where \(R\) is the minimal nonnegative solution of the nonlinear matrix equation

\[
A_0 + RA_1 + R^2A_2 = 0
\]

(4.11)

with spectral radius \(\text{sp}(R) < 1\). Furthermore, the stationary probability vector \(\pi_0\) exists and

is uniquely determined by solving the boundary condition

\[
\pi_0 B + \pi_1 A_2 = \pi_0 (B + RA_2) = 0
\]

(4.12)

and the normalization condition

\[
\sum_{i=0}^{\infty} \pi_i e = \pi_0 (I - R)^{-1} e^T = 1,
\]

(4.13)

where \(I\) denotes the identity matrix with dimension \(M \times M\).

This theorem is due to Neuts for the case \(M < \infty\) (see [24]); the theorem for the case

\(M = \infty\) follows from the results of Tweedie [26] (see also [22]). From Theorem 3 we know

that the stationary distribution is determined once \(R\) is obtained. Several iterative algorithms
exist for numerically solving (4.11); an overview of such algorithms is provided in [22].

A related matrix, typically denoted by \(G\), also plays an important role together with \(R\) in the general theory of matrix-analytic methods. This related \(G\) matrix is the minimal nonnegative solution of the nonlinear matrix equation

\[
A_0 G^2 + A_1 G + A_2 = 0
\]

(see [22, 24, 25]). Some recent algorithms for numerically solving (4.11) involve first computing

the matrix \(G\) and then computing the matrix \(R\) based on the relationship

\[
R = A_0 (-[A_1 + A_0 G])^{-1}
\]

(see [22]).

The QBD process driven by \(Q\) is ergodic if and only if it satisfies the mean drift condition

(see [24])

\[
\omega A_0 e^T < \omega A_2 e^T,
\]

(4.14)

where \(\omega\) is the equilibrium distribution of the generator \(A_0 + A_1 + A_2\) and \(e\) the unit vector. When (4.14) is satisfied, the stationary distribution of the QBD process exists.
4.3 Back-off scheme (i): basic back-off

We can model the basic back-off scheme (i) as a QBD process and then use the theory presented in Section 4.2 to prove some interesting features of the scheme.

4.3.1 Back-off scheme (i) as a QBD process

The meaning of level and phase in this specific model must first be defined. We shall already work under the assumption that node 2 is saturated, which in fact will be proved formally in Section 4.3.2. Since node 1 is saturated as well, the only buffer content that we need to keep track of is that of node 3. That is why the level \( l(x_3) \) represents the state of the system with \( x_3 \) packets at node 3. The phases that form the level will now be described by the possible states the system can be in when there are \( x_3 \) packets at node 3.

We denote each state by \( S_1 S_2 S_3 \), where \( S_i \) denotes the state of node \( i \). Since nodes 1 and 2 are saturated, they can only be transmitting \((S_i = T)\), in back-off \((B)\), or blocked by a neighbor (while not in back-off) \((X)\). Node 3 can be in one of these states, but can also be empty \((E)\). Now we obtain the following vectors:

\[
l(x_3) = \{XTX, XTB, BTX, BTB, BBB, BBT, TBB, BXT, BBT, TXT\}, \quad x_3 \geq 1. \tag{4.15}
\]

\[
l(0) = \{XTE, XTB, BTE, BTB, BBB, BBE, BBE, EEE, TBE, TXB, TXE\}, \tag{4.16}
\]

where we note that state \( S_1 S_2 S_3 = EEE \) of \( l(0) \), is only a dummy state, since the corresponding state for \( x_3 \geq 1 \) has node 2 blocked by only node 3, which is now empty and thus this state has no equivalent. The process is visualized in Figure 4.2, where state \( i \) corresponds to the
$i$-th coordinate of $l(x_3)$ in (4.15). The boundary conditions are shown in Figure 4.3, where states $i$ correspond to the $i$-th coordinates of $l(0)$ and $l(1)$ in (4.16) and (4.15).

The matrices $A_0$, $A_1$ and $A_2$ are now found to be:

$$A_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$
4.3. Back-off scheme (i): basic back-off

\[
A_1 = \begin{pmatrix}
\Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\beta & \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\beta & 0 & \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \beta & \beta & \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \beta & \Delta & \beta & \beta & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \Delta & 0 & \beta & \beta & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \Delta & 0 & \beta & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & \Delta & 0 & \beta & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & \Delta & 0 & 0 & \Delta \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \Delta & 0 & \Delta \\
\end{pmatrix},
\]

and

\[
B = \begin{pmatrix}
\Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\beta & \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\beta & 0 & \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \beta & \beta & \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \beta & \Delta & \beta & \beta & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \beta & 0 & \Delta & 0 & \beta & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & \Delta & 0 & \beta & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \Delta & 0 & 1 & 0 & 0 & \Delta \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & \Delta \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & \Delta & \Delta \\
\end{pmatrix},
\]

where \( \Delta \) is the shorthand notation for the element that makes all elements in the corresponding row in the matrix \( Q \) in (4.9) add up to zero. We note that \( \Delta \) is row-dependent.

### 4.3.2 Proof of Proposition 3

The proof consists of two separate parts: the saturation of node 2 and the stability of node 3.

#### Saturation node 2

Let \( \eta > 0 \). To prove the saturation of node 2 we will use a reductio ad absurdum, assuming that \( \theta_1(\eta) = \theta_2(\eta) \). Theorem 2 now tells us that

\[
\mathbb{E}\{W_1\} = \frac{\theta_2(\eta)}{\theta_1(\eta)} \cdot \frac{1}{1 + \eta} = \frac{1}{1 + \eta},
\]

yielding

\[
\theta_1(\eta) = \theta_2(\eta) = \tau(\eta) = \frac{1}{1 + \eta + \frac{1}{1 + \eta}}.
\]

We will now use an argument similar to the proof of Theorem 2, only this time we introduce \( W_2 \) as the time node 2 has to wait between a successive back-off time and a transmission and analyze the expected value of this random variable, instead of \( W_1 \). We see that

\[
\theta_2(\eta) = \frac{1}{1 + \eta + \mathbb{E}\{W_2\}},
\]
which would again imply $\mathbb{E}\{W_2\} = \frac{1}{1+\eta}$. However, we can also calculate $W_2$ in a different way. We define

\[ r_1 : \quad \text{expected number of transmissions of node 1 in between two successive transmissions of node 2, starting when node 2 is in back-off}, \]
\[ r_2 : \quad \text{expected number of transmissions of node 1 in between two successive transmissions of node 2, starting when node 2 is not in back-off}. \]

When node 1 starts a transmission during a back-off time of node 2, we have again that the expected waiting time contributed by the transmission of node 1 equals

$$\mathbb{E}\{Y_1\} = \mathbb{E}\{\max\{T_1 - B_2, 0\}\} = \frac{1}{1+\eta},$$

In case node 1 starts a transmission when node 2 is not in back-off, it will contribute

$$\mathbb{E}\{Y_2\} = \mathbb{E}\{T_1\}$$

to the waiting time. However, the waiting time of node 2 can also be increased by node 3, for example when node 1 is silent but node 3 is not. We call this extra waiting time $\mathbb{E}\{Y_3\} \geq 0$. Since $r_1 + r_2 = \theta_1(\eta)/\theta_2(\eta) = 1$ and $r_2 > 0$, we now have

$$\mathbb{E}\{W_2\} = r_1\mathbb{E}\{Y_1\} + r_2\mathbb{E}\{Y_2\} + \mathbb{E}\{Y_3\}$$

$$\geq (1 - r_2) \frac{1}{1+\eta} + r_2$$

$$= \frac{1}{1+\eta} + r_2 \frac{\eta}{1+\eta}$$

$$> \frac{1}{1+\eta}.$$

This leads to a contradiction, and thus we can conclude that node 2 is not stable, i.e. it is saturated and $\theta_1(\eta) \neq \theta_2(\eta)$. Because we always have $\theta_1(\eta) \geq \theta_2(\eta)$, we now see that $\theta_1(\eta) > \theta_2(\eta)$, so that the queue of node 2 grows without bounds at a linear rate.

**Stability node 3**

To prove that the queue of node 3 is stable, i.e., does not tend to infinity, it must be shown that the system satisfies the mean drift condition (4.14).

We first determine the equilibrium distribution $\omega$ of the generator $A_0 + A_1 + A_2$, which satisfies

$$\omega(A_0 + A_1 + A_2) = 0, \quad (4.19)$$
$$\omega e^T = 1. \quad (4.20)$$

Matrix $A_0 + A_1 + A_2$ describes a dependent set of equations, and therefore the last column of the matrix is replaced by $e^T$, so that the new matrix is invertible. This is standard work used in order to find an equilibrium distribution of a Markov process. Multiplying the inverted
matrix by \{0, 0, 0, 0, 0, 0, 0, 1\} gives \(\omega\):

\[
\omega = C^{-1} \cdot \left\{ 4\beta^3 + 6\beta^4, 2\beta^2 + 3\beta^3, 2\beta^2 + \beta^3, 2\beta + 5\beta^2 + 3\beta^3, 2 + 5\beta + \beta^2, 2\beta + 5\beta^2, 2\beta^2 + 6\beta^3 + 3\beta^2, 2\beta^2 + 6\beta^3, 2\beta^2 + 6\beta^3 + 3\beta^4, 3\beta^3 + 9\beta^4 + 3\beta^5 \right\},
\]

with \(C = 2 + 11\beta + 26\beta^2 + 34\beta^3 + 21\beta^4 + 3\beta^5\). Hence,

\[
\omega A_0^T e^T = \frac{2\beta + 9\beta^2 + 13\beta^3 + 6\beta^4}{2 + 11\beta + 26\beta^2 + 34\beta^3 + 21\beta^4 + 3\beta^5},
\]

\[
\omega A_2^T e^T = \frac{2\beta + 9\beta^2 + 15\beta^3 + 3\beta^4 + 3\beta^5}{2 + 11\beta + 26\beta^2 + 34\beta^3 + 21\beta^4 + 3\beta^5},
\]

from which we can easily see that \(\omega A_0^T e^T < \omega A_2^T e^T\) for all \(\beta > 0\), and hence the mean drift condition (4.14) is satisfied for all \(\eta > 0\). This means the system is ergodic for all values of \(\eta\) and therefore the queue of node 3 is stable.

### 4.4 Back-off scheme (ii): zero back-off node 3

The system in which node 3 does not go into back-off can only have 0 or 1 packet(s) at node 3, since any packet transmitted by node 2 can immediately be forwarded by node 3, as soon as node 2 goes into back-off. Node 2 cannot transmit another packet until node 3 is silent again after processing the previous packet.

We will first present the features of the QBD process corresponding to this system, followed by some conclusions.

#### 4.4.1 Back-off scheme (ii) as a QBD process

Since the buffer content of node 3 is always 0 or 1, and node 1 is saturated, we only have to keep track of the buffer of node 2.

We shall model the whole process as a QBD process, where we define the level \(l(x_2)\) as the state of the process when the buffer size of node 2 equals \(x_2\), and the phase as all states that are necessary to fully describe the process.

Consider a level \(l(x_2)\) for \(x_2 \geq 1\). To describe the phase, we again use states \(S_i\) where \(S_i\) denotes the state of node \(i\). Nodes 1 and 2 can be: transmitting a packet (T), blocked by a transmitting neighbor (X), or in back-off (B). Node 3 can either be transmitting (T) or have an empty buffer (E). This gives

\[
l(x_2) = \{\text{BTE}, \text{BBE}, \text{TBE}, \text{XTE}, \text{BBT}, \text{XTE}, \text{TBT}, \text{BET}, \text{TET}\}, \quad x_2 \geq 1.
\]

For \(x_2 = 0\) we have a slightly different vector, since node 2 cannot be transmitting. This rules out states BTE and XTE. We replace state BTE by a new state BEE, where node 1 is in back-off and nodes 2 and 3 are empty, and we replace state XTE by the dummy state EEE. This yields

\[
l(0) = \{\text{BEE}, \text{BEE}, \text{TBE}, \text{EEE}, \text{BBT}, \text{TEE}, \text{TBT}, \text{BET}, \text{TET}\}.
\]
Note that $S_2 = X$ is also replaced by $S_2 = E$, since it has no further consequences for node 2 whether or not a neighbor is transmitting when node 2 is empty. For $x_2 \geq 1$ the process is shown in Figure 4.4. The matrices with transition rates are as follows:

$$A_0 = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix},$$

$$A_2 = \begin{pmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix},$$

(4.23)

$$A_1 = \begin{pmatrix}
\Delta & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 \\
\beta & \Delta & \beta & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \Delta & 0 & \beta & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \Delta & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & \Delta & 0 & \beta & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & \Delta & 0 & \beta \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta & \beta \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta
\end{pmatrix},$$

(4.24)

where $\Delta$ is such that the elements in each of the rows in $A_0 + A_1 + A_2$ add up to zero.

$$B = \begin{pmatrix}
\Delta & 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 \\
\beta & \Delta & \beta & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \Delta & 0 & \beta & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & \Delta & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & \Delta & 0 & \beta & 0 & 0 \\
0 & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & \Delta & 0 & \beta \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta & \beta \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta
\end{pmatrix},$$

(4.25)
where $\Delta$ is such that the elements in each of the rows in $B + A_0$ add up to zero.

4.4.2 Main results for scheme (ii)

**Theorem 4** (zero back-off node 3). When nodes 1 and 2 have exponential back-offs with mean $\eta$, and node 3 has no back-off, the throughputs are given by

$$
\theta_1(\eta) = \frac{2 + 2\eta + \eta^2}{3 + 5\eta + 3\eta^2 + \eta^3},
$$

$$
\theta_2(\eta) = \theta_3(\eta) = \frac{1 + 2\eta + \eta^2}{3 + 5\eta + 3\eta^2 + \eta^3}.
$$

Node 2 will be saturated (not stable) for all values of $\eta$ and attains the maximal throughput

$$
\max_{\eta} \theta_2(\eta) = \frac{\sqrt{2}}{4}
$$

at $\eta = \sqrt{2} - 1$.

**Proof.** First we will prove that node 2 becomes saturated for every value of $\eta$. For this we use the mean drift condition (4.14). The goal is now to find the equilibrium distribution $\omega$ such that (4.19) and (4.20) hold for the matrices defined in Section 4.4.1. Standard matrix calculations give

$$
\omega = \frac{1}{1 + 3\beta + 5\beta^2 + 3\beta^3}\{\beta + \beta^2, 1, \beta, \beta^2 + \beta^3, \beta, \beta^2 + \beta^3, \beta^2, \beta^3\}.
$$

(4.29)

Now substituting $\eta = 1/\beta$ into (4.29) and calculating both sides of the mean drift condition (4.14) yields:

$$
\omega A_0 e^T = \frac{2 + 2\eta + \eta^2}{3 + 5\eta + 3\eta^2 + \eta^3} = \frac{1 + \eta}{3 + 2\eta + \eta^2} + \frac{1}{3 + 5\eta + 3\eta^2 + \eta^3},
$$

$$
\omega A_2 e^T = \frac{1 + \eta}{3 + 2\eta + \eta^2}.
$$

Since the system is only defined for $\eta > 0$, the denominator $3 + 5\eta + 3\eta^2 + \eta^3$ is always positive, so that $\omega A_0 e^T > \omega A_2 e^T$ for all values of $\eta > 0$. This QBD process is never ergodic and hence node 2 is saturated.

Hence, in order to find the stationary distribution, the Markov process describing the system with zero back-off for node 3 can be replaced by one that is considerably easier. Since node 3 can have only 0 or 1 packet(s) and nodes 1 and 2 are saturated, the state space required to describe the Markov process is finite. This results in the complete state space comprising only the 9 states that together formed one level:

$$
\{\text{BTE, BBE, TBE, XTE, BBT, TXE, TBT, BXT, TXT}\}.
$$

The generator matrix is given by $A_0 + A_1 + A_2$. This is because the interlevel transitions are now seen as transitions within the same level, since we no longer distinguish between different values of $x_2$. Figure 4.5 depicts this Markov process on a finite state space. The stationary distribution is given in (4.29). The throughput of node $i$ follows from adding the probability mass of all states for which $S_i = T$. This yields (4.26) and (4.27). We can now find the maximum throughput of the system by simply differentiating $\theta_2(\eta)$ and the result is $\theta_2(\sqrt{2} - 1) = \sqrt{2}/4$, as in (4.28).  \[\square\]
Figure 4.5: The reduced Markov process describing back-off scheme (ii)

We see that indeed (4.5) holds for scheme (ii), as can be deduced from (4.26) and (4.27). Observe that

\[
\theta_1(\eta) - \theta_2(\eta) = O(\eta^{-3})
\]

which shows that the difference in throughputs diminishes rapidly with \( \eta \).

**Corollary 2** (ordering and large back-off asymptotics). When nodes 1 and 2 have exponential back-offs with mean \( \eta \), and node 3 has no back-off, we have \( \theta_1(\eta) > \tau(\eta) > \theta_2(\eta) \), with the function \( \tau \) defined in (4.4). Moreover, since \( \theta_1(\eta) - \theta_2(\eta) = O(\eta^{-3}) \), there is the asymptotic relation

\[
\theta_1(\eta) \sim \theta_2(\eta) \sim \tau(\eta), \quad \eta \to \infty.
\]

(4.30)

Although back-off schemes (i) and (ii) are different, and their analysis requires different mathematical models, the throughputs of both schemes turn out to be almost identical, as can be seen in Figure 4.6. In Figure 4.7 the difference in throughput between both back-off schemes is shown for node 1 and node 2. The maximum difference is approximately 0.001636, which is less than one percent of the corresponding throughput.

Since node 3 is stable in back-off scheme (i), it will be able to transmit all the packets that it receives. When \( \eta \to \infty \), the periodicity of the system is lost, because the probability of node 3 blocking a transmission of node 2 goes to zero. This is because the fraction of time that node 3 is transmitting also goes to zero. This explains why the difference between schemes (i) and (ii) goes to zero for larger \( \eta \).

### 4.5 Back-off scheme (iii): truncated back-offs

In the case of truncated back-offs, the exponential back-off times are terminated by the arrival of a new packet. This results in the back-off of node 3 becoming completely irrelevant. Again node 3 can only have 0 or 1 packet(s), since it will start transmitting right after node 2 finishes a transmission. After its transmission, node 3 will go into back-off, and will become active again only after node 2 has transmitted a new packet. It will always become active immediately, regardless of whether it was in back-off or not. Hence, whenever it is in back-off, it does not have a packet to transmit anyway.
4.5. Back-off scheme (iii): truncated back-offs

Figure 4.6: $\theta_1(\eta)$ (upper curve) and $\theta_2(\eta)$ (lower curve) for back-off schemes (i) and (ii). The lines are the throughputs for scheme (ii), while the dots are the numerically found throughputs for scheme (i).

Figure 4.7: Difference between throughputs in back-off schemes (i) and (ii) for node 1 (the one approaching zero fastest) and node 2.
4.5.1 Back-off scheme (iii) as a QBD process

For modeling this system, we shall take an approach similar to Section 4.4.1. We model the system as a QBD process, when the level $l(x_2)$ denotes all states for which the buffer size of node 2 equals $x_2$. The phase description is the same as in Section 4.4.1, again given by

\[
l(x_2) = \{\text{BTE, BBE, TBE, XTE, TXE, TBT, BXT, TXT}\}, \quad x_2 \geq 1,\\
l(0) = \{\text{BEE, BBE, TBE, EEE, BBT, TEE, TBT, BET, TET}\}.
\]

This Markov process is shown in Figure 4.8. The matrices with transition rates are as follows:

\[
A_0 = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix},
\]

and $A_1$, $A_2$, and $B$ as in (4.23), (4.24) and (4.25), respectively.

The only difference between the two types of back-off is that for the truncated back-off model, a packet that has just been transmitted by node 1, will always arrive at node 2 to find it out of back-off. As argued, the possible back-off of node 3 does not matter in either system, so that only some transitions that increase the buffer size of node 2 are not identical. These transitions are captured by the difference in the matrix $A_0$.

4.5.2 Main results for scheme (iii)

We now find that this scheme makes node 2 stable from a certain value of $\eta$ onwards. This observation can be of great practical significance, especially if a similar phenomenon turns out to hold when applying the truncated back-off scheme to more general mesh networks.
Theorem 5 (stability node 2). In the case of truncated back-offs, node 2 is stable and hence 
\[ \theta_1(\eta) = \theta_2(\eta) = \theta_3(\theta), \] if and only if 
\[ \eta > \sqrt{5} - 1. \]  

Proof. Theorem 5 will be proved using the mean drift condition (4.14). We will now find the equilibrium distribution \( \omega \) such that (4.19) and (4.20) hold for the matrices defined in Section 4.5.1. Standard matrix calculations give 
\[ \omega = C^{-1}\{2\beta + 4\beta^2, 1, \frac{2\beta + 5\beta^2 + 4\beta^3}{2 + 3\beta + \beta^2}, 2\beta^2 + 4\beta^3, 2\beta, \frac{2\beta^2 + 11\beta^3 + 14\beta^4 + 4\beta^5}{2 + 3\beta + \beta^2}, \frac{4\beta^2 + 4\beta^3}{2 + \beta}, 4\beta^2, 6\beta^3 + 4\beta^4}{2 + \beta}\}, \]  
where \( C = 1 + 5\beta + 14\beta^2 + 12\beta^3 \). Using \( \eta = 1/\beta \) now yields 
\[ \omega A_0 e^T = \frac{8 + 4\eta + \eta^2}{12 + 14\eta + 5\eta^2 + \eta^3}, \]  
\[ \omega A_2 e^T = \frac{4 + 6\eta + 2\eta^2}{12 + 14\eta + 5\eta^2 + \eta^3}. \]  

We can conclude that this system is ergodic if and only if 
\[ \frac{8 + 4\eta + \eta^2}{12 + 14\eta + 5\eta^2 + \eta^3} < \frac{4 + 6\eta + 2\eta^2}{12 + 14\eta + 5\eta^2 + \eta^3}. \]  
Since the system is only well-defined for \( \eta > 0 \), we have \( 8 + 4\eta + \eta^2 < 4 + 6\eta + 2\eta^2 \), with the only valid interval being \( \eta > \sqrt{5} - 1 \). This means that node 2 is saturated if and only if 
\[ 0 < \eta \leq \sqrt{5} - 1. \]  

We will now treat the system in which node 2 is saturated differently from the system in which it is stable. In case node 2 becomes saturated, the throughput of all nodes can be determined exactly with help of a reduced Markov process, in which the number of packets at node 2 is not taken into account.

Theorem 6 (truncated back-offs). In the case of truncated back-offs, and assuming \( \eta \leq \sqrt{5} - 1 \), the throughputs are given by 
\[ \theta_1(\eta) = \frac{8 + 4\eta + \eta^2}{12 + 14\eta + 5\eta^2 + \eta^3}, \]  
\[ \theta_2(\eta) = \theta_3(\eta) = \frac{4 + 6\eta + 2\eta^2}{12 + 14\eta + 5\eta^2 + \eta^3}. \]  

Moreover, \( \theta_1(\eta) \geq \theta_2(\eta) = \theta_3(\theta) \) with equality if and only if \( \eta = \sqrt{5} - 1 \).

Proof. In case \( 0 < \eta \leq \sqrt{5} - 1 \), node 2 is saturated, and the Markov process describing the system with truncated back-off is replaced by one that is less complicated. Since node 3 can have only 0 or 1 packet(s) and nodes 1 and 2 are saturated, the state space is finite, and given by 
\[ S = \{\text{BTE, BBE, TBE, XTE, BBT, TXE, TBT, BXT, TXT}\}. \]
The transition matrix for this Markov process is given by $A_0 + A_1 + A_2$. This process is shown in Figure 4.9. We have already calculated the equilibrium distribution to determine the mean drift condition of the QBD process in the proof of Theorem 5. The stationary distribution is equal to $\omega$ as was found in (4.32).

Adding the stationary probabilities of the states in which $S_i = T$ gives throughput $\theta_i(\eta)$ of node $i$, resulting in:

$$\theta_1(\eta) = \frac{8 + 4\eta + \eta^2}{12 + 14\eta + 5\eta^2 + \eta^3},$$

$$\theta_3(\eta) = \theta_2(\eta) = \frac{4 + 6\eta + 2\eta^2}{12 + 14\eta + 5\eta^2 + \eta^3}.$$  

Furthermore, equating $\theta_1(\eta) = \theta_2(\eta)$ gives $-4 + 2\eta + \eta^2 = 0$, resulting in $\eta = -1 \pm \sqrt{5}$, of which only $\eta = -1 + \sqrt{5}$ is positive, thus equality in throughputs holds for this mean back-off time.

Observe that $\theta_1(\eta) \to 2/3$ and $\theta_2(\eta) \to 1/3$ as $\eta \downarrow 0$. Rewriting (4.37) as

$$\theta_1(\eta) = \frac{8 + 4\eta + \eta^2}{12 + 14\eta + 5\eta^2 + \eta^3} = \frac{1}{1 + \eta + \frac{2\eta + 4}{8 + 4\eta + \eta^2}},$$

and using (4.2), yields

$$\mathbb{E}\{W_1\} = \frac{4 + 2\eta}{8 + 4\eta + \eta^2}$$

and hence

$$\theta_1(\eta)\mathbb{E}\{W_1\} = \frac{4 + 2\eta}{12 + 14\eta + 5\eta^2 + \eta^3} = \frac{2(\eta + 1)(\eta + 2)}{12 + 14\eta + 5\eta^2 + \eta^3} \cdot \frac{1}{1 + \eta} = \theta_2(\eta) \frac{1}{1 + \eta}.$$

Indeed, we see that (4.5) does hold.
We note that the maximum throughput of node 2 is obtained for \( \eta \leq \sqrt{5} - 1 \), as can clearly be seen from Figures 4.10 and 4.11. Differentiating \( \theta_2(\eta) \) and setting the derivative equal to 0 gives as only positive real solution

\[
\eta_{\text{opt}} = -\frac{3}{2} - \frac{1}{2} \sqrt{z} + \frac{1}{2} \sqrt{13 - z + \frac{4}{\sqrt{z}}}
\approx 0.93328,
\]

with

\[
z = \frac{1}{3} \left( 13 - \frac{25}{(557 - 12\sqrt{2046})^{1/3}} - \left( \frac{557 - 12\sqrt{2046}}{12}\right)^{1/3} \right).
\]

This leads to a maximum throughput of

\[
\theta_2(\eta_{\text{opt}}) \approx 0.375132.
\]

We see that this is not a big difference with

\[
\theta_2(\sqrt{5} - 1) = \frac{1}{6} \sqrt{5} \approx 0.372678,
\]

although for the latter we are at the point of critical stability of node 2, while at the former the queue of node 2 does tend to infinity. Depending on priorities, it might be best in this situation to take \( \eta = \sqrt{5} - 1 + \varepsilon \) for some small \( \varepsilon > 0 \), so that the whole system is stable, while still maintaining a throughput close to the maximum.

**Theorem 7.** In the case of truncated back-offs, we have for \( \eta > \sqrt{5} - 1 \)

\[
\theta_1(\eta) = \theta_2(\eta) = \theta_3(\eta) = \tau(\eta) = \frac{1}{1 + \eta + \frac{1}{1+\eta}}. \tag{4.39}
\]

**Proof.** In Theorem 5 we saw that for \( \eta > \sqrt{5} - 1 \) we have stability for node 2, or \( \theta_1(\eta) = \theta_2(\eta) \). Now using (4.5) we find

\[
\mathbb{E}\{W_1\} = \frac{\theta_2(\eta)}{\theta_1(\eta)} \cdot \frac{1}{1 + \eta} = \frac{1}{1 + \eta},
\]

and substituting this into (4.2) yields

\[
\theta_1(\eta) = \frac{1}{1 + \eta + \frac{1}{1+\eta}}.
\]

Using the stability of node 3, this implies (4.39). \( \square \)

Figure 4.10 shows the throughputs of nodes 1 and 2. For \( 0 < \eta \leq \sqrt{5} - 1 \) exact solutions (4.35) and (4.36) have been used. For \( \eta > \sqrt{5} - 1 \) the throughput has been numerically calculated following Sections 4.2 and 4.5.1. Figure 4.11 depicts the difference between the numerical calculations and the function found in Theorem 7. The dots seem to be uniformly distributed in the square \((\sqrt{5} - 1, 5) \times (-5 \cdot 10^{-10}, 5 \cdot 10^{-10})\). This suggests that these differences are only caused by machine precision, instead of difference in function. This strengthens both the validity of the QBD calculations and the function found in Theorem 7.
Figure 4.10: $\theta_1(\eta)$ and $\theta_2(\eta)$ for back-off scheme (iii), combined with numerically obtained values for $\eta > \sqrt{5} - 1$. The dashed line is $\tau(\eta)$.

Figure 4.11: Difference in throughput between numerical calculation and approximation (4.39) for back-off scheme (iii)
4.6 Deterministic truncated back-off times

Until now we have only looked into the case of exponential extra back-off times, with mean $\eta$. We would like to extend this analysis to more general back-off times. Therefore we will present a preliminary analysis for deterministic back-off times. The situation has been simplified to truncated deterministic back-offs, so that the back-off of a node is terminated by the arrival of a new packet. The transmission times are still taken to be exponential with mean 1, since otherwise the process would have lost its randomness and become completely periodic as $\lambda \to \infty$.

4.6.1 Preliminary results

We will look at the process as a sequence of cycles, where a new cycle starts exactly when node 2 finishes a transmission. Truncating the back-offs makes the situation easier, since now node 3 always starts with a transmission at the start of the cycle. This is because node 2 has just transmitted a packet, terminated the back-off of node 3 and was forced into back-off itself, giving node 3 the opportunity to immediately start a transmission. We note that at the start of a cycle, node 2 always goes into back-off.

At the start of a new cycle $i$, however, node 1 can still have some back-off $R_i$ left from the previous cycle, with $0 \leq R_i < \eta$. This makes the situation at the start of a new cycle dependent on the previous cycle. In case we did not truncate the back-offs, node 3 could also have some residual back-off left, which would make the situation even harder.

Since the back-off time of node 2 can be cut short by node 1, this back-off time is not deterministic. We use the notation $H$ for the back-off time of node 2. After the back-off time node 2 can have a waiting time $W_2$. Together, the back-off time and the waiting time form the inactive time. Just as in Proposition 2, we note that

$$\theta_2(\eta) = \frac{1}{1 + \mathbb{E}\{H + W_2\}}.$$  

We denote the transmission time of node $i$ as $T_i$, for $i = 2, 3$. Since node 1 can have multiple transmissions during one cycle, as we will discuss below, we have $T_{1,j}$ denoting the $j$-th transmission time node 1.

**Theorem 8.** In the case of truncated deterministic back-offs with length $\eta$, node 2 becomes saturated, while node 3 remains stable, for all $\eta$.

**Proof.** We will investigate a single cycle, so that node 2 transmits exactly one packet. We will show that node 1 will always transmit at least one packet and has a positive probability of transmitting two or more packets. This means that the expected number of packets transmitted by node 1 during one cycle is strictly larger than 1, and thus larger than the expected number of packets transmitted by node 2 during one cycle.

First, however, we will show that node 3 transmits at least one packet as well. At the start of the cycle node 2 goes into back-off, while the possible back-off of node 3 is terminated. Since node 3 has a packet and is not blocked by node 2, it will start transmitting the packet it just received. Therefore node 3 will remain stable, having at least the same output as input. Assuming the system starts with node 3 empty, we can conclude that node 3 always has either 0 or 1 packets at its queue.
Denote the back-off time that remains for node 1 at the start of the cycle by \( R \). Since \( R < \eta \), the back-off time of node 1 will always elapse before that of node 2, which lasts \( \eta \) unless it is terminated earlier by a transmission of node 1. This tells us that node 1 will always get the opportunity to transmit a packet in every cycle independent of the value of \( \eta > 0 \).

Furthermore, if the transmission of node 3, which is exponentially distributed with mean 1, lasts longer than \( R + T_{1,1} + \eta \), node 1 will start transmitting another packet.

\[
P[T_3 > R + T_{1,1} + \eta] = P[T_3 > T_{1,1}]P[T_3 > R + \eta] = \frac{1}{2}e^{-R-\eta} > 0,
\]
and therefore the expected number of packets transmitted by node 1 is strictly larger than 1, resulting in the saturation of node 2 on the long run.

We compare Theorem 8 to Theorem 5. As it turns out, in contrast to exponential back-off times, deterministic back-off times do not facilitate the opportunity to stabilize node 2 by increasing the length of the back-off.

### 4.6.2 Closer investigation using cycles

We will investigate a single cycle, in which the remainder of the previous back-off period of node 1 is now fixed, say \( R \). To determine the expected inactive period of node 2, we condition on the state of node 1 at the moment node 3 finishes its transmission. Node 1 can either be active (a), in which case node 2 will start its transmission right after node 1 finishes, or it can be in back-off with the back-off of node 2 already over (b), in which case node 2 immediately starts its transmission, or node 3 finishes within the left-over back-off time of node 1 (c). Both cycles of type (a) and type (b) are shown in Figure 4.12. Here, cycle 1 depicts the situation in which node 3 finishes during a transmission of node 3 and cycle 2 shows the other situation. Note that the case

We will first treat case (c), in which \( T_3 < R \), so that node 3 finishes before node 1 starts its first transmission. This happens with probability \( 1 - e^{-R} \) and we have

\[
E\{H + W_2 \mid T_3 < R\} = R + 1,
\]

since node 2 has to wait for the transmission of node 1 to finish, which starts after \( R \) units of time and has mean duration 1. Now denote

- \( B_m \): The moment node 1 finishes its \( m \)-th back-off period, \( m \geq 0 \),
- \( C_m \): The moment node 1 finishes its \( m \)-th transmission, \( m \geq 1 \),

where \( B_0 = R \) and for ease of later notation, we put \( C_0 = 0 \). This means

\[
B_m = R + \sum_{i=1}^{m}(T_{1,i} + \eta), \quad m \geq 0,
\]

\[
C_m = R + \sum_{i=1}^{m-1}(T_{1,i} + \eta) + T_{1,m}, \quad m \geq 1.
\]

Note that \( C_0 < B_0 < C_1 < B_1 < C_2 < B_2 < \ldots \). For \( m \geq 0 \), we will now investigate events \( B_m < T_3 < C_{m+1} \), which correspond to case (a), and for \( m \geq 1 \) we look at \( C_m < T_3 < B_m \),
4.6. Deterministic truncated back-off times

We therefore need probabilities $P[T_3 < B_m]$ and $P[T_3 < C_m]$, for $m \geq 1$. We use the property that $T_3$ is memoryless to see that

$$P[T_3 < B_m] = 1 - P[T_3 > R + \sum_{i=1}^{m} (T_{1,i} + \eta)]$$

$$= 1 - P[T_3 > R + m\eta] \prod_{i=1}^{m} P[T_3 > T_{1,i}]$$

$$= 1 - e^{-R-m\eta} \left(\frac{1}{2}\right)^m, \quad (4.41)$$

and similarly we can show that

$$P[T_3 < C_m] = 1 - e^{-R-(m-1)\eta} \left(\frac{1}{2}\right)^m. \quad (4.42)$$

(a) The probability of event $\{B_m < T_3 < C_{m+1}\}$, is now easily found by combining (4.41) and (4.42):

$$P[B_m < T_3 < C_{m+1}] = P[T_3 < C_{m+1}] - P[T_3 < B_m]$$

$$= 1 - e^{-R-m\eta} \left(\frac{1}{2}\right)^{m+1} - 1 + e^{-R-m\eta} \left(\frac{1}{2}\right)^m$$

$$= e^{-R-m\eta} \left(\frac{1}{2}\right)^{m+1}. \quad (4.43)$$
The corresponding expected inactive time for node 2 needs to be determined. In this case node 2 only has to wait for node 1 to finish after node 3 finishes its transmission, before it can start, or

\[
\mathbb{E} \{ H + W_2 \mid B_m < T_3 < C_{m+1} \} = \mathbb{E} \{ T_3 \mid B_m < T_3 < C_{m+1} \} + 1.
\]

We can order the \(m\) finished back-off and transmission periods of node 1 any way we like and therefore, we will look at the event as if the back-off periods have all happened first, so that after \(R + m\eta\) time units, node 3 is still active and node 1 starts its transmissions. Since the transmission of node 3 will outlast another \(m\) transmissions of node 1, we continuously have two exponential variables with mean 1 competing. Since node 3 finishes between the \(m\)-th and \((m+1)\)-th transmission of node 1, we need the expected time at which the \((m+1)\)-th exponential time finishes. We do not have to discriminate between finishing transmissions of node 1 or 3. Since the expected minimum of two exponential distributions with mean 1 is \(1/2\), we now have

\[
\mathbb{E} \{ H + W_2 \mid B_m < T_3 < C_{m+1} \} = R + m\eta + \frac{m+1}{2} + 1 = R + m\eta + \frac{m+3}{2}, \quad m \geq 0. \quad (4.44)
\]

(b) We also use (4.41) and (4.42) to find the probability of event \(\{C_m < T_3 < B_m\}\):

\[
P \{C_m < T_3 < B_m\} = P \{T_3 < B_m\} - P \{T_3 < C_m\} = 1 - e^{-R-m\eta} \left( \frac{1}{2} \right)^m - 1 + e^{-R-(m-1)\eta} \left( \frac{1}{2} \right)^m = e^{-R-(m-1)\eta} \left( \frac{1}{2} \right)^m \left( 1 - e^{-\eta} \right). \quad (4.45)
\]

In this case, when node 3 finishes, node 2 can start immediately since node 1 is in back-off. We now have to find

\[
\mathbb{E} \{ H + W_2 \mid C_m < T_3 < B_m \} = \mathbb{E} \{ T_3 \mid C_m < T_3 < B_m \}.
\]

We will split this expected value, using the property that \(T_3\) is memoryless.

\[
\mathbb{E} \{ T_3 \mid C_m < T_3 < B_m \} = \mathbb{E} \{ C_m \mid T_3 > C_m \} + \mathbb{E} \{ T_3 - C_m \mid T_3 - C_m < B_m - C_m \} = \mathbb{E} \{ C_m \mid T_3 > C_m \} + \mathbb{E} \{ T_3^* \mid T_3^* < \eta \},
\]

where \(T_3^*\) denotes the rest of the transmission after \(C_m\), so \(T_3^* = T_3 - C_m\), given that \(T_3 > C_m\) and therefore \(T_3^* \sim \text{Exp}(1)\). For the same reasoning as in case (a), we can already see that \(\mathbb{E} \{ C_m \mid C_m < T_3 \} = R + (m-1)\eta + (m+1)/2\), since the \(m\)-th transmission of node 1 must also be the \(m\)-th in total to finish and after \(R\) there have been \(m-1\) completed back-off times.
4.6. Deterministic truncated back-off times

<table>
<thead>
<tr>
<th>Case</th>
<th>Event $E$</th>
<th>$\mathbb{P} [E]$</th>
<th>$\mathbb{E} {H_2 + W_2 \mid E}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>${B_{m-1} &lt; T_3 &lt; C_m}$</td>
<td>$e^{-R-(m-1)\eta} \left(\frac{1}{2}\right)^m$</td>
<td>$R + (m-1)\eta + \frac{m+2}{2}$</td>
</tr>
<tr>
<td>(b)</td>
<td>${C_m &lt; T_3 &lt; B_m}$</td>
<td>$e^{-R-(m-1)\eta} \left(\frac{1}{2}\right)^m (1-e^{-\eta})$</td>
<td>$R + (m-1)\eta + \frac{m+2}{2} - \frac{\eta e^{-\eta}}{1-e^{-\eta}}$</td>
</tr>
<tr>
<td>(c)</td>
<td>${C_0 &lt; T_3 &lt; B_0}$</td>
<td>$1 - e^{-R}$</td>
<td>$R + 1$</td>
</tr>
</tbody>
</table>

Table 4.1: An overview of the various possible events for truncated deterministic back-offs, where $m \geq 1$

For the other part we see, with $F_\eta(t) = \mathbb{P} [T^*_3 < t \mid T^*_3 < \eta]$ that

\[
\mathbb{E} \{T^*_3 \mid T^*_3 < \eta\} = \int_0^\infty t \, dF_\eta(t) = \int_0^\eta t \, d \left(\frac{1-e^{-t}}{1-e^{-\eta}}\right) = \frac{1}{1-e^{-\eta}} \int_0^\eta te^{-t} \, dt = \frac{1}{1-e^{-\eta}} (1 - (1 + \eta)e^{-\eta}) = 1 - \frac{\eta e^{-\eta}}{1-e^{-\eta}},
\]

so that

\[
\mathbb{E} \{H + W_2 \mid C_m < T_3 < B_m\} = R + (m-1)\eta + \frac{m+2}{2} - \frac{\eta e^{-\eta}}{1-e^{-\eta}}, \quad m \geq 1. \tag{4.46}
\]

In Table 4.1 an overview is given of all possible cases, as were found in (4.40), and (4.43)-(4.46). We now construct the expected cycle length as follows

\[
\mathbb{E} \{H_2 + W_2 + T_2 \mid R\} = 1 + (1-e^{-R}) (R+1) + \sum_{m=1}^\infty e^{-R-(m-1)\eta} \left(\frac{1}{2}\right)^m \left( R + (m-1)\eta + \frac{m+2}{2} \right) + \sum_{m=1}^\infty e^{-R-(m-1)\eta} \left(\frac{1}{2}\right)^m (1-e^{-\eta}) \left( R + (m-1)\eta + \frac{m+2}{2} - \frac{\eta e^{-\eta}}{1-e^{-\eta}} \right). \tag{4.47}
\]

As seen in Appendix D.1, this simplifies to

\[
C(\eta, R) = \mathbb{E} \{H_2 + W_2 + T_2 \mid R\} = 2 + R + \frac{e^{-R}}{2-e^{-\eta}}. \tag{4.48}
\]

So now we have determined the expected cycle length for the case with a fixed $R$. We see in (4.48) that when $R = 0$ always holds, we have

\[
\lim_{\eta \to \infty} \theta(\eta) = \lim_{\eta \to \infty} (C(\eta, 0))^{-1} = \lim_{\eta \to \infty} \left(2 + \frac{1}{2-e^{-\eta}}\right)^{-1} = \left(\frac{5}{2}\right)^{-1} = \frac{2}{5},
\]
and this limit is approached from below. This might seem a bit counterintuitive since a back-off time going to infinity would normally bring the throughput to zero. However, the practical meaning of $R = 0$ is that the back-off time of node 1 is also always terminated by the end of a transmission of node 2, as can be seen in Figure 4.12. This means that node 1 and 3 will both start transmitting a packet, and they are expected to be both finished after $3/2$ time units. Since the back-off time of node 1 tends to infinity, node 3 will always finish before node 1 can start another transmission, so after the $3/2$ time units, node 2 can always capture the channel and finish the cycle in an expected duration of 1 time unit, so that the expected cycle length is $5/2$.

However, we can not just take a fixed value $R$, since $R_i$ is a random variable that can assume a different value at the start of every cycle $i$. The distribution of this random variable depends on when node 3 finishes its transmission. In case (a), when node 3 finishes during a busy period of node 1, the back-off of node 1 will start at the moment the transmission of node 2 starts. In case (b), the back-off of node 1 has already been running since last time node 1 finished a transmission and it will still be running when node 2 starts its transmission. In case (c), node 3 finishes within the remainder of the back-off node 1 has from last cycle. After node 1 has finished transmitting, node 2 will start its transmission right away. This leads to

$$R_{i+1} | R_i = \begin{cases} \max\{\eta - T_2, 0\} & \text{in cases (a) and (c)}, \\
\max\{\eta - T_2 - T_3^* , 0\} & \text{in case (b)},
\end{cases}$$

where $T_3^*$ is the rest of the transmission of node 3, after node 1 transmitted its last packet. Since the transmission time is memoryless, we have $T_3^* \sim (T_3 | T_3 < \eta)$. So summing over all probabilities of case (a), as found in (4.43), yields

$$R_{i+1} | R_i = \begin{cases} \max\{\eta - T_2, 0\} & \text{w.p. } e^{-R_i} \frac{e^{-R_i}}{2 - e^{-\eta}} + 1 - e^{-R_i} = 1 - e^{-R_i} \frac{1 - e^{-\eta}}{2 - e^{-\eta}} \\
\max\{\eta - T_2 - T_3, 0\} | T_3 < \eta & \text{w.p. } e^{-R_i} \frac{1 - e^{-\eta}}{2 - e^{-\eta}}.
\end{cases}$$

The distribution $F(\cdot)$ of $\lim_{i \to \infty} R_i$ is yet to be found. When this distribution is known, the throughput can be calculated as

$$\theta(\eta) = \frac{1}{\int_0^\infty C(\eta, R) \ dF(R)}.$$ 

This will not be part of this thesis. We will, however, make an approximation in the following section.

### 4.6.3 An approximation to truncated deterministic back-off times

In this section we will approximate the throughput of truncated deterministic back-offs. To do this, we will not look at $\{R_i\}_{i=1}^{\infty}$ as a series of random variables, but instead use the expected value of $R_i$ to approximate the expected value of $R_{i+1}$, starting with an initial $R_0 = \eta/2$. In Appendix D.2 we show

$$\mathbb{E}\{R_{i+1} | R_i = R\} = f(R, \eta),$$

where

$$f(R, \eta) = e^{-R} \frac{e^{-2\eta} + 2\eta e^{-\eta} - 1}{2 - e^{-\eta}} + \eta - 1 + e^{-\eta}. \quad (4.50)$$
4.6. Deterministic truncated back-off times

<table>
<thead>
<tr>
<th>$\eta$</th>
<th>0.01</th>
<th>0.1</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_0(\eta)$</td>
<td>0.0050000000</td>
<td>0.05000000</td>
<td>0.500000</td>
<td>2.50000</td>
<td>5.00000</td>
<td>50.00000</td>
<td>500.00000</td>
</tr>
<tr>
<td>$a_1(\eta)$</td>
<td>0.0000495086</td>
<td>0.00457531</td>
<td>0.319975</td>
<td>3.96833</td>
<td>8.99668</td>
<td>99.00000</td>
<td>999.00000</td>
</tr>
<tr>
<td>$a_2(\eta)$</td>
<td>0.0000495070</td>
<td>0.00456313</td>
<td>0.310526</td>
<td>3.99789</td>
<td>8.99998</td>
<td>99.00000</td>
<td>999.00000</td>
</tr>
<tr>
<td>$a_3(\eta)$</td>
<td>0.0000495070</td>
<td>0.00456313</td>
<td>0.309982</td>
<td>3.99815</td>
<td>8.99998</td>
<td>99.00000</td>
<td>999.00000</td>
</tr>
<tr>
<td>$a_4(\eta)$</td>
<td>0.0000495070</td>
<td>0.00456313</td>
<td>0.309950</td>
<td>3.99815</td>
<td>8.99998</td>
<td>99.00000</td>
<td>999.00000</td>
</tr>
<tr>
<td>$a_5(\eta)$</td>
<td>0.0000495070</td>
<td>0.00456313</td>
<td>0.309948</td>
<td>3.99815</td>
<td>8.99998</td>
<td>99.00000</td>
<td>999.00000</td>
</tr>
<tr>
<td>$a_6(\eta)$</td>
<td>0.0000495070</td>
<td>0.00456313</td>
<td>0.309948</td>
<td>3.99815</td>
<td>8.99998</td>
<td>99.00000</td>
<td>999.00000</td>
</tr>
</tbody>
</table>

Table 4.2: Fast convergence of $\{a_i(\eta)\}_{i=0}^{\infty}$ for various values of $\eta$.

We will look at the following sequence

$$a_0(\eta) = \frac{\eta}{2},$$

$$a_{n+1}(\eta) = f(a_n(\eta), \eta), \quad n \geq 0.$$  

**Lemma 2.** The sequence $\{a_i(\eta)\}_{i=0}^{\infty}$ converges to $a(\eta)$ as $n \rightarrow \infty$, for all $\eta > 0$, with $0 < a(\eta) < \eta$.

**Proof.** See Appendix D.3. \hfill \Box

We now substitute $\mathbb{E}\{\lim_{i \rightarrow \infty} R_i | \eta\}$ by $a(\eta)$ and construct the approximation

$$\hat{\theta}(\eta) = \frac{1}{C(\eta, a(\eta))}.$$  \hspace{1cm} (4.51)

In Table 4.2 the first few terms of $a_i(\eta)$ are shown for various values of $\eta$, showing a fast convergence. We see for large $\eta$ that this sequence seems to converge to $\eta - 1$, which is logical, since the probability of having case (c) tends to one. This means that $R_i$ tends to max{$\eta - T_2, 0$}, and the expected value of this limiting variable tends to $\eta - 1$ for $\eta$ growing larger.

Since the sequence $\{a_i(\eta)\}_{i=0}^{\infty}$ converges, we must have

$$a(\eta) = e^{-a(\eta)} \frac{e^{-2\eta} + 2\eta e^{-\eta} - 1}{2 - e^{-\eta}} + \eta - 1 + e^{-\eta}. $$  \hspace{1cm} (4.52)

Setting

$$c_1(\eta) = \frac{e^{-2\eta} + 2\eta e^{-\eta} - 1}{2 - e^{-\eta}},$$

$$c_2(\eta) = \eta - 1 + e^{-\eta},$$

we can rewrite (4.52) to

$$(a(\eta) - c_2(\eta)) e^{a(\eta) - c_2(\eta)} = c_1(\eta) e^{-c_2(\eta)}.$$  

This equation has the solution

$$a(\eta) = c_2(\eta) + W\left(c_1(\eta) e^{-c_2(\eta)}\right),$$  \hspace{1cm} (4.53)
where the function $W$ is the so-called Lambert $W$ function. This function $W(x)$ is defined as the solution of $We^W = x$ for $W$. See [14], p. 23 and [12] for extensive treatments of this function. Among many other features, the function $W$ has just one positive real solution when $x > 0$, as the function $We^W$ increases from 0 to infinity when $W$ increases from 0 to infinity. Several (infinite)-series expressions exist for calculating this positive solution, and the numerical calculation of the $W$-function is standard. In fact, in Mathematica, the Lambert $W$ function is built in, and goes by the name “ProductLog”. In Table 4.2 the numerical outcomes are shown as $a(\eta)$. We will use these to approximate the throughput.

So, using $\hat{R} = a(\eta)$ and substituting this into (4.48), which is used for approximation (4.51), we can now make an approximation for this scheme. We use Java simulations to obtain Figures 4.13 and 4.14. In Figure 4.13 we see how the approximation follows the simulation results very accurately, except for the part $1 < \eta < 2$, where the approximations are clearly higher. In Figure 4.14 we see the differences, which clearly seem to follow a line. Indeed, in the interval $1 < \eta < 2$ there is a peak, but the maximum approximation error is smaller than 0.004, which is in the order of one percent of the throughput.

This probably indicates a higher variance in the distribution of $R$ for these values of $\eta$. In Figure 4.15 we see the estimated probabilities on all three cases, with $R$ being equal to $(\max\{\eta - T_2 - T_3, 0\} \mid T_3 < \eta)$ instead of $\max\{\eta - T_2, 0\}$ with increasing probability where the dashed line is increasing. The dashed line is still high between $\eta = 1$ and $\eta = 2$, and for these back-off times, the distribution of $(\max\{\eta - T_2 - T_3, 0\} \mid T_3 < \eta)$. It is our belief that this causes the approximation that is based on only taking the expected values to give higher throughput. This is supported by the fact that $C(\eta, R)$ is concave in $R$ for any fixed value of $\eta$.

For this approximation we numerically find the maximum achievable throughput to equal

$$\max_{\eta > 0} \hat{\theta}(\eta) \doteq 0.366514,$$

for $\eta_{\text{max}} \doteq 0.706093$. 

Figure 4.13: Throughput of truncated deterministic back-off times, with the line presenting the approximation and the dots the simulation results.
Figure 4.14: Approximation errors for truncated deterministic back-off times

Figure 4.15: Approximated probabilities of case (a) (dotted line), case (b) (dashed line), and case (c) (solid line)
4.6.4 Extension to non-truncated deterministic back-off times

We just provided an approach to the analysis of truncated deterministic back-off times and concluded with approximations. In this section we will briefly discuss what needs to be done to extend this analysis to non-truncated deterministic back-off times.

If we want to look at the case of non-truncated deterministic back-offs, we cannot only take $R_i$ into account as we did in this chapter, we have to take both $R_{1,i}$ and $R_{3,i}$ into account, the residual back-off times of nodes 1 and 3 at the start of cycle $i$. This makes it more difficult to analyze the system, since the distribution of $(R_{1,i+1}, R_{3,i+1})$ will depend on $(R_{1,i}, R_{3,i})$.

Also, node 2 is not always in the position to start a transmission when it sees both nodes 1 and 3 inactive after $R_{1,i}$, which it was in the case of truncated deterministic back-off times. In that case the back-off of node 2 only served the purpose of making sure node 1 can always transmit at least one packet per cycle, since it would always outlast $R_i$ and give node 1 the opportunity to start a transmission. This means that there is a positive probability $P[H + W_2 < \eta] > 0$, where $H$ is the time in back-off and $W_2$ is the time node 2 has to wait after back-off has elapsed. For non-truncated deterministic back-offs, the back-off of node 2 does have further use, since we always have $H + W_2 \geq \eta$.

However, apart from these complications, the analysis of non-truncated deterministic back-offs should be quite similar.

4.7 Comparison of the various back-off schemes

For $n = 3$ and $k = 1$ we have analyzed three different exponential back-off schemes. In scheme (i) and (ii) nodes are forced into an exponential back-off time after completing a transmission. The difference between the schemes is that in scheme (ii) node 3 does not have back-off. These schemes turn out to give nearly identical throughput functions, and scheme (ii) serves as an accurate approximation for the more realistic scheme (i). The third back-off scheme (iii) has a complete different mechanic, in which back-off times are terminated when a new packet arrives at the node. An interesting relation is found in Theorem 2. We find that the queue of node 2 is stable for certain values of the mean back-off time $\eta$.

We also look at a scheme in which the back-off times are taken deterministic, but can be terminated by the arrival of a new packet, just as in scheme (iii). Here we find an approximation to the throughput that can still be sharpened by further analysis of the distribution of $\lim_{i \to \infty} R_i$ in (4.49).

In Figure 4.16 we see a comparison between the throughput functions as found in this section, where as approximation for scheme (i) we used the function found in scheme (ii), and for the scheme with truncated deterministic back-off times we used the approximation as found in Section 4.6.3. Since the approximation errors are so small, we can use these functions to compare the throughput achieved by the various back-off schemes. We have depicted only the final throughput of node 3, but since $\theta_2(\eta) = \theta_3(\eta)$ for all $\eta > 0$ and all schemes, this also equals the throughput of node 2.

We see that for every $\eta$ in the graph scheme (i) performs worst of the three. Combined with the fact that this scheme does never give stability to node 2, it is arguably not be the best scheme to implement. Proceeding to scheme (iii) we see that this scheme achieves highest throughput of the three. Still after $\eta = \sqrt{5} - 1$, at which point a faster throughput drop starts, this scheme performs best for a while until it is overtaken by the scheme with truncated deterministic back-off times. However, stability is already achieved and therefore this scheme
4.7. Comparison of the various back-off schemes

Figure 4.16: The throughput functions as found in this chapter, for scheme (ii) (dotted line), for scheme (iii) (dashed line), and the approximation for truncated deterministic back-off times (solid line).

<table>
<thead>
<tr>
<th>Scheme (i)</th>
<th>Scheme (ii)</th>
<th>Scheme (iii)</th>
<th>Trunc. det.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum $\theta_2$</td>
<td>$\approx 0.353553$</td>
<td>$0.353553$</td>
<td>$0.375132$</td>
</tr>
<tr>
<td>$\eta_{\text{max}}$</td>
<td>$\approx 0.414214$</td>
<td>$0.414214$</td>
<td>$0.933280$</td>
</tr>
<tr>
<td>Node 2 stable</td>
<td>No</td>
<td>No</td>
<td>For $\eta &gt; \sqrt{5} - 1$</td>
</tr>
<tr>
<td>Node 3 stable</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 4.3: Overview of the results of this chapter, with the values that include an approximation-sign coming from approximations.

looks as the most useful of the three. It does show that the truncated deterministic back-off scheme is also worth considering. It can be questioned how reasonable the implementation of exponential back-off times is in reality. A more logical distribution would be the uniform or the deterministic distribution.

Either way, there is a clear distinction in the throughput achieved by the non-truncated scheme and the truncated schemes, in which the latter come out better. An explanation could be that these schemes react better to the positive circumstances found in the process. For example, node 2 can start being silent for a back-off time with mean $\eta$, so that it gives both nodes 1 and 3 the opportunity to transmit a packet. However, when it notices that node 1 has finished its transmission, the remainder of its back-off time will probably be useless, since node 1 is not transmitting now anyway. The fact that node 3 is always stable allows node 2 to only look at node 1 for new transmission opportunities.

In Table 4.3 a short overview is presented. It seems counterintuitive that a significantly higher value of $\eta_{\text{max}}$ might yield a higher maximum throughput, as is the case when comparing schemes (i) and (iii). Although scheme (iii) might need a higher value of $\eta$ to reach its maximum throughput, it does not mean that the actual time spent in back-off by nodes 2 and 3 is higher than the value $\eta_{\text{max}}$ that maximizes the throughput of schemes (i) and (ii).
Since the back-off in scheme (iii) can be terminated before it is finished, the average time spent in back-off might just as well be lower.
Chapter 5

Generalizations supported by simulations

In this chapter we will analyze the throughput for systems with $n \geq 3$ and $k \geq 1$, for both the uncontrolled system, described in Section 2.1, and the three flow control methods, described in Sections 2.2, 2.3, and 2.4. We base these analyses on simulation results. First we will describe in Section 5.1 the way the simulations have been done. After that we will look at the uncontrolled mesh chain in Section 5.2.

In Chapter 3 we analyzed windowed flow control in the case of $n = 3$ and $k = 1$. We found the behavior of the system as the window size tended to infinity. In Section 5.3 we will generalize these results for larger $n$ and $k$, using both heuristic arguments and simulation results.

Also for $n = 3$ and $k = 1$, we analyzed four forms of flow control in Chapter 4. We will extend the results for back-off schemes (i) and (iii) to the general case and discuss the differences between them, as well as other characteristic observations in the simulations. This happens in Section 5.4.

After this we will briefly analyze the simulations done for express forwarding in Section 5.5, and we end the chapter with a comparison between all different flow control mechanisms in Section 5.6.

For windowed flow control and the extra back-off schemes we will show that the influence of $n$ is minor as long as $n \geq 2k + 1$, while the influence of $k$ is very significant. As we will see, the average throughput for fixed $k$, taken over all tested values of $n$, can be approximated very well by a function of the form $f(k) = (a + bk)^{-1}$ for all schemes.

5.1 Outline of simulations

For the various models described in the Chapter 2, simulation programs have been written in Java. Each of them describes one of five models: the basic uncontrolled one, one with windowed flow control, two with extra back-off, of which one with non-truncated exponential back-off times and one with truncated exponential back-off times, and finally the model with express forwarding. Since the notion of $\lambda \to \infty$ is not implementable in Java without changing the original system dynamics, it is replaced by a very large number. In the test cases we use $\lambda = 10^{10}$, so that the average delay between transmissions caused by the access protocol is smaller than $10^{-10}$ time units. We regard this as negligible compared to the transmissions
with average duration of 1 time unit.

For all models we have parameters \( n \), the number of nodes, and \( k \), which represents the blocking range. We let \( k \) range from 1 to 8. The range of \( n \) is dependent of \( k \).

Windowed flow control normally depends on the used window size \( W \). However, as described in Conjecture 1, we can assume the throughput to increase as \( W \) grows. In Sections 5.3.1 we will see that for \( n \geq 2k + 1 \) the system with windowed flow control and \( W \to \infty \) is equivalent to the mesh chain in which node \((n - k)\) is saturated and functions as both the source and the sink for the packets, which are sent from node \( n \) to node 1. We will use this equivalent system to avoid long start-up times of the simulation and to immediately find the corresponding limiting throughput.

The optimal extra back-off scale \( \eta \) has not been determined yet for the general case of \( n \geq 3 \) and \( k \geq 1 \). Therefore we take \( \eta \) as a parameter too, letting it run from 0 to \( 2k - 1 \) for the non-truncated case, with step size 1 or 2, dependent on the value of \( k \). In case of truncated back-off times, we let \( \eta \) run from 0 to \( k \), with step size \( k/10 \).

All simulations have run for a simulation time of \( 10^6 \) time units, with a startup time of \( 10^5 \) time units. These simulations have all been repeated at least 5 times, to be able to construct good confidence intervals. An analysis has shown that this results in a maximum sample standard deviation of 0.000675695, for all simulations that have been done. The corresponding 95%-confidence interval has width 0.00118455, which will turn out to be insignificant in the comparison of the various results. We keep in mind that this is the worst case, which speaks in favor of the accuracy of the simulation results.

### 5.2 Uncontrolled protocol

For the uncontrolled protocol we have already shown in Section 2.1.2 that the throughput does not exceed \( 1/(2k + 1) \) for \( n \geq 2k + 1 \). We will not do exact calculations, but instead look at the simulation results. We have simulated this process for \( k = 1, 2, \ldots, 8 \) and \( 2k + 1 \leq n \leq 4k \).

In Figure 5.1 the average throughput taken over all simulated values of \( n \) is shown for \( k = 1, 2, \ldots, 8 \). In Section 3.1 of [15] the throughput of this system is argued to be bounded by functions of the form \( \theta(k) = (a + bk)^{-1} \). We will see in this chapter that all back-off schemes can be accurately approximated by a function of that form. We use linear regression on \( 1/\theta \) to find these constants \( a \) and \( b \) and this results in \( a = 1.27535 \) and \( b = 2.01349 \). The regression function \( f(k) = (1.27535 + 2.01349k)^{-1} \) is also shown in Figure 5.1. We see that these regression coefficients are indeed conform the values found in Conjecture 2 of [15], which states that \( 1 \leq a \leq 2 \) and \( b \approx 2 \).

### 5.3 Windowed flow control

In this section we will extend the results from Chapter 3 to general \( n \) and \( k \), with \( n \geq 2k + 1 \). We first present some heuristic arguments for the general variant of Conjecture 2, first for \( n = 2k + 1 \), then for \( n \geq 2k + 1 \). These are supported by simulations.

#### 5.3.1 The case of general \( n \) and \( k \)

In Chapter 3 we looked at a mesh system comprising \( n = 3 \) nodes with blocking range \( k = 1 \). For that particular system we found results on the throughput and the average waiting time.
This simple case could still be algebraically analyzed for small window sizes, because the state space stays small, having only two possible activity states (either the inner node or the outer nodes were active). For large window sizes the complexity grows very rapidly, as was seen in (3.1).

For larger values of $n$, the activity state space increases drastically in the non-trivial cases. For example, when $n = 4$ and $k = 1$, there are already 7 activity states. Either one of the four nodes can be active, or either the outer two nodes or the odd or even nodes are active. This results in a far greater complexity and the packet state space will increase as well because $\binom{n+W-1}{n-1}$ will be multiplied by $\frac{n+W}{n}$ as the number of nodes increases from $n$ to $n+1$.

It does not seem doable to find the throughput of these systems analytically. We can, however, follow the same course as in Section 3.2 and look at what occurs when the window size tends to infinity. We will first argue that a pile of packets will form at the middle node in a system with $n = 2k + 1$, for large $W$. After that we will generalize this to $n \geq 2k + 1$, in which the $(n-k)$-th node becomes “saturated”. In Section 5.3.3 we will look into the simulation results for these systems, presenting an approximation to the throughput for mesh systems with general $k$ and $n \geq 2k + 1$.

**Saturated node for $n = 2k + 1$**

As a generalization of what we saw in Section 3.2, in this section we will show that in a mesh chain comprising $n = 2k + 1$ nodes, with blocking range $k$, the packets will concentrate at node $k + 1$ as $W \to \infty$. This suggests equivalence to the mesh circuit with node $k + 1$ saturated.

We will write $\theta_i$ as the throughput of node $i$, for $i = 1, \ldots, n$. The throughput of node $i$ is equal to its output and also to the input of node $i \mod n + 1$, since the topology of the system with windowed flow control can be seen as a cycle instead of a chain.

As the window size increases, the number of packets in the system increases with it. Just as in the 3-node system, this might result in an expected number of $W - \varepsilon$ packets residing...
in a certain node, with
\[ \varepsilon < w(n, k) \] (5.1)
for some \( w(n, k) \) fixed for \( n \) and \( k \), independent of the window size \( W \). As \( W \to \infty \) the expected number of packets at this node, \( \mathbb{E}\{X_i\} \geq W - w(n, k) \), grows to infinity as well. Even stronger, as \( W \to \infty \), the probability of finding this node empty must tend to zero, otherwise we have \( \lim_{W \to \infty} \mathbb{P}\{X_i = 0\} = p > 0 \) and then \( \mathbb{E}\{X_i\} \geq W - w(n, k) \) can not be satisfied for \( W > w(n, k)/p \). Indeed, then we would have \( \mathbb{E}\{X_i\} \leq (1 - p)W < W - w(n, k) \) so we know that
\[ \lim_{W \to \infty} \mathbb{P}\{X_i = 0\} = 0. \] (5.2)
In general, this holds for the expected number of packets \( \mathbb{E}\{X_i\} \geq c(W - w(n, k)) \) for any \( 0 < c < 1 \) and \( w(n, k) \) independent of \( W \).

In the system with windowed flow control all throughputs \( \theta_i \) are equal to each other \( i = 1, 2, ..., n \). However, when we let \( W \to \infty \), just as in the 3-node case, we look at the system in a slightly different way. We say that one or more nodes will become saturated, meaning that the queues of these nodes will satisfy (5.2). These nodes will then have an “infinite” pile of packets at their disposal. Now, if we only assume the first node to be saturated, for example, we have the basic uncontrolled version. As shown in Section 2.1.2, this would lead to a violation of the principle that \( \theta_1 = \theta_2 = \ldots = \theta_n \), since the throughput of the last node is lower than the throughput of the first node. We realize that these arguments are not rigorous, but they do give an intuitive feel of what is going on.

**Conjecture 3.** For \( k \geq 1 \) and \( n = 2k + 1 \), the mesh chain with length \( n \) and blocking range \( k \) and windowed flow control with window size \( W \to \infty \) is equivalent to a mesh circuit with preserved blocking range in which node \( k + 1 \) is saturated.

**Argument.** A system is said to be in a stable configuration of saturated nodes, when every node has equal input and output. We will now show that any other configuration of saturated nodes than just node \( k + 1 \) saturated will lead up to a contradiction, under the assumption that the system with \( W \) increasing indeed tends to a stable configuration of saturated nodes.

We will first show that node \( k + 1 \) must be saturated. If we assume node \( k + 1 \) is not saturated, another node must be. Therefore we take node \( i < k + 1 \) to be saturated and we will show that this contradicts the requirement of equal throughputs at every node.

We split the transmissions started by node \( i \) into two categories. First we have those that commence after a competition in which node \( k + 1 \) participated, and the partial throughput of node \( i \) corresponding to these transmissions equals \( \theta_{i,1} \). Second we have the transmissions that node \( i \) started while node \( k + 1 \) was either empty or blocked. The throughput corresponding to these transmissions equals \( \theta_{i,2} \), so that \( \theta_{i,1} + \theta_{i,2} = \theta_i \). Whenever there is a competition between nodes \( i \) and \( k + 1 \) we have \( \theta_{i,1} = \theta_{k+1} \). However, since we have \( \theta_{k+i+1} > 0 \), node \( k+i+1 \) is sometimes transmitting a packet. Whenever it is, one of nodes \( 1, ..., i \) is also transmitting. The probability of node \( k+i+1 \) being the latter of the two to finish its transmission equals 1/2. This means with positive probability node \( i \) will be competing for the channel again, while node \( k+1 \) is still blocked. This results in \( \theta_{i,2} > 0 \), so that
\[ \theta_i = \theta_{i,1} + \theta_{i,2} > \theta_{i,1} = \theta_{k+1}, \]
which contradicts the assumption that all throughputs must be equal. This means that there is no stable configuration of saturated nodes with node \( k + 1 \) excluded from it.
5.3. Windowed flow control

If not only node $k+1$ is saturated, we take the first saturated node $i$ in the circuit before $k+1$. We assume again that $i < k$. For the exact same reasoning as above these nodes must have unequal throughput, which is a contradiction.

For $i > k+1$ the argument is analogous, except that we do not look at node $i+k+1$ as the node that blocks node $k+1$, but at node $i-k-1$.

**Saturated node for general $n$ and $k$**

For general blocking ranges $k$ in mesh chains with $n \geq 2k+1$ and large window sizes, the packets also seem to pile up. Now this interesting behavior seems to occur at node $(n-k)$, bringing us to the following hypothesis, which is a generalization of Conjectures 2 and 3. We do not have a proof for it, only a partial heuristic argument showing that the saturated node should be one of the nodes $k+1, \ldots, n-k$.

**Hypothesis 1.** *For $k \geq 1$ and $n \geq 2k+1$, the mesh chain with length $n$ and blocking range $k$ and windowed flow control with window size $W \to \infty$ is equivalent to a mesh circuit with preserved blocking structure in which node $(n-k)$ is saturated.*

**Heuristic argument.** We apply the same argument as for Conjecture 3 to show that any node $i$ with $i < k+1$ that is saturated will have higher throughput than node $k+1$ and likewise any saturated node $i$ with $i > n-k$ will have higher throughput than node $n-k$, which both lead to a contradiction. This means that the saturated node must be within nodes $k+1, k+2, \ldots, n-k$.

5.3.2 Simulations supporting Hypothesis 1

In this section simulations will be shown that support Hypothesis 1. We will see that for large window sizes the packets will first concentrate at node $k+1$ and then slowly shift to node $n-k$, which keeps a majority of the packets until the end of the simulations. This leads us to believe that this will be the saturated node.

Java simulations that follow the windowed flow control model as described in Section 2.2 have been conducted for $n = 21$ and $k = 4$. At regular time intervals, the distribution of the $W$ packets has been registered and put into graphs. The results of these simulations can be seen in Figure 5.2 for $W = 1024$ and in Figure 5.3 for $W = 8192$. Note that the nodes are now numbered from 0 to $n-1$.

The graphs for $W = 1024$ show the queues at intervals of 20000 units of time apart. All packets start out at node 1 (which is numbered 0 in these graphs), but at $t = 20000$ the large queue has already shifted to node $k+1$ (node 4 in the graphs). After this they seem to slowly shift to node $(n-k)$ and this process seems to be completed after approximately 140000 time units. The distribution of packets does not go through significant changes between this moment and the end of the simulations after 400000 units of time. We do notice in nearly every graph with $t \geq 140000$ that there seems to be a small pile of packets at one of the nodes $k+1, k+2, \ldots, n-k-1$, with at most around 200 packets there.

The same behavior is seen for $W = 8192$, although everything seems to take longer. Here the time between two graphs is 50000 units. The pile at node $k+1$ is not seen in good shape until $t = 150000$. After this the even slower process of shifting to node $(n-k)$ starts and this seems to finish around $t = 900000$. Again, after this time there are no significant changes to be seen anymore. However small, there seem to be some little piles around nodes 5 to 10.
Figure 5.2: The fluctuating queues over time, with $n = 21$, and $k = 4$, and $W = 1024$
Figure 5.3: The fluctuating queues over time, with $n = 21$, and $k = 4$, and $W = 8192$
at time $t = 950000$, but since the packet-axis has been extended to 8192, these are hardly noticeable. This is conform our beliefs that there is a $w(n, k)$ as in (5.1) that does not depend on $W$.

The window size is 8 times larger in the second scenario than it is in the first. The time it takes for the packets to reach a certain equivalent configuration is also of the order 8 times larger. We could see the moments $t \approx 140000$ and $t \approx 900000$ as the finish of start-up times, after which the system goes into a reasonably stable state.

We will now compare the throughput of the system with node $(n - k)$ saturated to that of the system with large window size $W$. For $k = 7$ and $15 \geq n \geq 29$ we find both simulation outcomes with $W = 10000$ and for the system with node $(n - k)$ saturated in Figure 5.4. We see the close resemblance between both systems. The noted differences are most likely simulation inaccuracies, since $W = 10000$ should give a good enough approximation of $W \to \infty$, so that during the simulation time it is unlikely that node $(n - k)$ becomes empty again when most of the $W = 10000$ have moved to that node. We calculate that for $k = 7$ the maximum relative difference in throughput is 0.22%, achieved at $n = 16$, which we regard as negligible. The system with $W = 10000$ needs a long startup time, testing showed that about $7 \cdot 10^6$ time units were necessary for the throughputs of all nodes to become approximately equal, while for the system with a saturated node $10^5$ units always sufficed in the tested scenario.

Using Conjecture 1, we can analyze the maximum achievable throughput for windowed flow control. We will first show that this conjecture is not violated by the simulation results for $n = 21$ and $k = 4$, of which the results are shown in Table 5.1. We see that the sequence in which the window size $W$ grows with steps of 5 indeed has an increasing throughput, with decreasing increments. We use Lemma 1 to note that the throughput converges to a maximum as $W \to \infty$.

Since we want to find the highest achievable throughput using windowed flow control, we will let $W \to \infty$ and use the equivalent model with the $(n - k)$-th node saturated. The results and regression are found in Section 5.3.3.
5.3. Windowed flow control

### Table 5.1: Throughput $\theta_{21,4,W}$ for increasing window size

<table>
<thead>
<tr>
<th>$W$</th>
<th>$\theta_{21,4,W}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.114906</td>
</tr>
<tr>
<td>10</td>
<td>0.130184</td>
</tr>
<tr>
<td>15</td>
<td>0.137869</td>
</tr>
<tr>
<td>20</td>
<td>0.142300</td>
</tr>
<tr>
<td>25</td>
<td>0.145459</td>
</tr>
<tr>
<td>30</td>
<td>0.147646</td>
</tr>
<tr>
<td>35</td>
<td>0.149387</td>
</tr>
<tr>
<td>40</td>
<td>0.150680</td>
</tr>
<tr>
<td>45</td>
<td>0.151925</td>
</tr>
<tr>
<td>50</td>
<td>0.152834</td>
</tr>
</tbody>
</table>

#### 5.3.3 Simulations and regression for windowed flow control

We conduct simulations for windowed flow control using the model with a cyclic topology and node $(n - k)$ taken to be saturated. First we will look into the influence of varying $n$, after that we will see the general influence of $k$.

In Figure 5.5 the relation between the throughput and $n$ is shown for $k = 1, 2, ..., 8$, with $n$ varying from $2k + 1$ to $4k + 1$. Note that, just as predicted in Section 3.3.2, the throughput for $n = 3, k = 1$ is $\theta_{3,1,\infty} \approx 0.396$ and is between the analytically found upper and lower bounds.

We immediately notice that for all $k$ the lowest throughput is achieved at $n = 2k + 1$. This might be counterintuitive, since one could argue that the less nodes the system has, the less resistance a packet will encounter while traveling through the system. However, if we look at $k = 1$ and $n = 3$, there is only once activity state in which node 2 is transmitting. Node 2 always has to beat both nodes 1 and 3 in order to capture the channel (provided that nodes 1 and 3 have a packet at their disposal). Node 2 is the bottleneck node here. For $n = 4$, we have a larger probability of node $(n - k) = 3$ capturing the channel, since its direct neighbor node 2 can be blocked by node 1, in which case node 3 does not have to compete with two other nodes, but with just one, resulting in a fairer share of the channel. A similar argument holds for node 2 having higher throughput in case $n = 4$.

At $n = 2k + 2$ the highest throughput is achieved for all $k$ except $k = 7$, after which the throughputs decrease again until approximately $n \approx 3k + 2$, where a new low is achieved. For small values of $k$ this is not noticeable. After this minimum we see another increasing part in the graph, which is followed by a new peak, lower than the previous one. It seems from these graphs that the throughput oscillates for $n$ increasing, with a decreasing amplitude.

Although the changes in throughput caused by $n$ are relatively small (maximums around 5% of the throughput for $k \leq 4$ and 2% for $k \geq 5$), they are consistent and not negligible.

Since the throughputs in Figure 5.5 oscillate around some seemingly limiting throughput value, we take the average over all tested values of $n$ and plot these in Figure 5.6. We see that these points also seem to admit to a function $\theta(k) = (a + bk)^{-1}$, with $a$ and $b$ yet to be determined. We do a linear regression on $1/\theta$ and find the regression function $f(k) = (1.23844 + 1.25705k)^{-1}$, which is also plotted in Figure 5.6. This function indeed fits the simulation results very well.
Figure 5.5: The varying simulation results for windowed flow control, with $k = 1, 2, ..., 8$ and $2k + 1 \leq n \leq 4k + 1$
5.4 Extra back-off

In Chapter 4 we analyzed various extra back-off schemes. In this section we will extend this analysis to general $n$ and $k$, for both the basic exponential back-off scheme, in which every node is forced into an exponential back-off time after the transmission of a packet, and for the truncated exponential back-off scheme, in which the modification is made that the back-off time of a node is terminated when it receives a new packet. We saw for $n = 3$ and $k = 1$ that the system incorporating truncated back-off times both allows stability for certain values of mean back-off duration $\eta$, and achieves higher throughputs.

5.4.1 Basic back-off scheme

For the basic back-off scheme, some simulation experiments already showed for $1 \leq k \leq 8$ that the maximum throughput is achieved for values of $\eta$ within $0 < \eta < k$. Therefore we simulated this system for $\eta = mk/10$ with $m = 0, 1, 2, ..., 10$. Instead of $\eta = 0$ we simulated $\eta = 10^{-6}$, so that the system is not reduced to the uncontrolled version. The number of nodes $n$ is taken to be dependent of $k$ and ranges from $2k + 1$ to $4k + 1$.

The outcome of such a simulation with $n = 21$, $k = 5$ and $0 \leq \eta \leq 5$ is shown in Figure 5.7. For all tested combinations $k = 1, ..., 8$ and $n \geq 2k + 1$, the simulation results show a graph with a clear maximum within $0 < \eta < k$.

Apart from the 11 dots that show the simulation results, we also see a short curve in Figure 5.7. This curve is used to find an approximation for both the maximum throughput and the value of $\hat{\eta}_{\text{opt}}$ for which this throughput is achieved. This is the quadratic function $f(\eta)$ that goes through the three highest points and it is an interpolation to find accurate coordinates of the top based on the simulation results, through standard analysis of quadratic functions. With this approximation of $\eta_{\text{opt}}$ we find $\hat{\theta}_{n,\text{opt}} = \hat{\theta}_n(\hat{\eta}_{\text{opt}}) = f(\hat{\eta}_{\text{opt}})$, where $\theta_n(\eta)$ represents the throughput of the last node in the chain. We use this method to analyze the influence of the increasing number $n$ on $\eta_{\text{opt}}$ and the results are shown in Figure 5.8.
Chapter 5. Generalizations supported by simulations

We see that the optimal $\eta$ does not show a clear consistent trend. For larger $k$ it seems that the optimal $\eta$ increases with $n$, but the graph corresponding to $k = 8$ does not show this behavior. Except for $k = 1$ they do all show that $\eta_{\text{opt}}$ is smallest for $n = 2k + 1$. Other than that it does not seem that $n$ has a clear effect on the optimal $\eta$.

In Figure 5.13 we see the same type of graphs as in Figure 5.8, only this time the maximum throughput found by interpolation is shown instead of the optimal value of $\eta$. Again, there is no clear consistent trend to be seen. It seems that the curves become more stable as $n$ increases. However, we see that the influence of $n$ is only of minor effect and the differences between the various throughput estimates may very well be a result of the inaccuracy of the simulation. We see that except for $k = 1$ the maximum variation caused by increasing $n$ is always smaller than 0.001. In Section 5.1 we saw that this is smaller than the maximum 95%-confidence interval, which is in favor of the belief that the variation of $\theta_{\text{opt}}$ is caused by the imprecision inherited by the simulation. We conclude that there is no recurring shape visible like the shape there was for windowed flow control for example.

For $k$ fixed we now take the average $\hat{\eta}_{\text{opt}}$ over all values of $n$ that we did simulations for and plot these averages against the blocking range $k$. In Figure 5.10 we see this is clearly an increasing function. By linear regression we find the straight line $f(k) = -0.762735 + 0.904537k$ which shows a good approximation for the values of $k$ plotted in the figure, but the pattern in the errors unveils that the true function is probably not linear. We note that the quadratic function $f(k) = -0.264321 + 0.605488k + 0.0332276k^2$ already approximates the line much better. However, since we only have values of $k$ from 1 to 8, we cannot say much more about the relation between $k$ and $\eta_{\text{opt}}$, since for example an exponential or higher-degree polynomial function could also very well perfectly fit the simulation results. The number of data points is too small to evaluate this more precisely.

Finally, in Figure 5.11 the maximum throughput is shown for $k = 1, \ldots, 8$. Again we have taken the average optimum throughput over all values of $n$ that we simulated and we see

![Figure 5.7: Throughput for $n = 21$, $k = 5$ and $\eta$ varying between 0 and 5, including interpolation between the three highest throughputs](image)
Figure 5.8: The estimated values of $\eta$ maximizing $\theta_k(\eta)$ for $k = 1, 2, \ldots, 8$ for an increasing number of nodes
Figure 5.9: The estimated values of maximum $\theta_n(\eta)$ for $k = 1, 2, \ldots, 8$ for an increasing number of nodes
that this clearly follows a function of the form \( f(x) = 1/(a + bx) \), with \( a \) and \( b \) to be determined. To find these values, we perform linear regression on \( 1/\tilde{\theta}_n(\tilde{\eta}_{opt}) \) and find \( a = 1.29552 \) and \( b = 1.63384 \).

### 5.4.2 Truncated back-off scheme

In Section 4.5 we analyzed truncated exponential back-off times, in which a node is forced into an exponential back-off time after it has finished a transmission, but the back-off time is terminated as soon as a packet arrives at the node. We saw that stability was achieved for node 2 and this happened from \( \eta = \sqrt{5} - 1 \) onwards. We now define \( \eta_c \) to be the critical mean back-off time such that (rate) stability is achieved for any \( \eta = \eta_c + \varepsilon \) with \( \varepsilon > 0 \).

Again, we have carried out simulations for \( k = 1, 2, \ldots, 8 \), with \( n \) varying between \( 2k + 1 \) and \( 4k \). The tested mean back-off times are taken to be in the order of \( k \), running from \( \eta = 0 \) to \( \eta = 2k - 1 \) with step size 1 or 2, depending on \( k \).

Analyzing the simulation results we see a stability result for general \( n \) and \( k \), with \( n \geq 2k + 1 \), much similar to the case of \( n = 3 \) and \( k = 1 \). It seems that nodes \( k + 2, \ldots, n \) are always stable, just like in the uncontrolled case that we discussed in Section 2.1.2. For all simulated combinations of \( n \) and \( k \) there was a value \( \eta_c \) such that for all values \( \eta > \eta_c \) that were simulated stability was achieved. Although it is not clear what the exact value of \( \eta_c \) is for general \( n \) and \( k \), we have been able to observe the trend that is depicted in Figure 5.12.

We see the effect of increasing \( \eta \) on the throughput for every node in the chain. First we see for \( \eta = 0 \) a system with nodes 2 to \( k+1 \) each having lower throughput than the previous node, resulting in their queues growing without bound at a linear rate. When \( \eta \) is increased, but still much lower than \( \eta_c \), we have \( \eta < \eta_c \) and there the first few nodes have already reached rate stability, having the same throughput as the node before them. It seems that the first node that does not have the same rate as its predecessors is in the order of \( \lceil \eta \rceil \) for \( \eta < \eta_c \).

When \( \eta = \eta_c - \varepsilon \) for some small \( \varepsilon > 0 \), we see a high throughput with the queue of only one node growing without bound at a linear rate. This changes when \( \eta_c = \eta + \varepsilon \), because then all nodes have rate stability. Finally we look at the figure with \( \eta > \eta_c \). Here the throughput is still equal for all nodes in the system, but it is lower than in the fourth diagram.

We cannot say in which of the cases \( \eta = \eta_c - \varepsilon \) and \( \eta = \eta_c + \varepsilon \) the throughput is highest. In the case of \( n = 3 \) and \( k = 1 \) we clearly saw that \( \tilde{\eta}_{opt} < \eta_c \), but it is uncertain if this has...
Chapter 5. Generalizations supported by simulations

Figure 5.11: Regression of the maximum achievable throughput $\hat{\theta}_n(\eta_{\text{opt}})$ versus blocking range $k$ for the basic extra back-off scheme

<table>
<thead>
<tr>
<th>$k$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lceil \eta_{c,k} \rceil$</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 5.2: An upper bound to the critical value of $\eta$ necessary for (rate) stability of the truncated extra back-off scheme

to be generalized for all $n \geq 2k + 1$. Intuitively we do have $\eta_{\text{opt}} \leq \eta_c$, since it seems that once stability has been achieved and the mean back-off times are still increased, node 1 will needlessly wait longer between introducing two new packets to the system. This would lead to the inequality

$$\theta(\eta) < \theta(\eta_c) \quad \text{for} \quad \eta > \eta_c,$$

so that the maximum throughput cannot be achieved for $\eta > \eta_c$ and thus $\eta_{\text{opt}} \leq \eta_c$.

Simulations have not been helpful in determining accurate approximations to values of $\eta_c$ or the maximum throughput achievable with this back-off scheme. As was also seen in Section 4.5, the throughput functions do not seem to have a smooth curve, making it harder to be analyzed, and interpolation as was used in Section 5.4.1 is not very useful. We therefore only state the smallest values of $\eta$ for which simulations showed the same throughput for every node. It is noteworthy that the value of $\eta_c$ does not depend on $n$ at all, as long as $n \geq 2k + 1$, and thus we speak of $\eta_{c,k}$. In Table 5.2 we see these values. We cannot conclude much from this, other than that $\eta_{c,k}$ could be a linear function of $k$. In that case we would have $\eta_{c,k} = a_1 k + a_2$, with $1 < a_1 < 9/7$ and $-2/7 < a_2 < 1$. Of course there are countless other explanations for $\eta_{c,k}$ to behave as in Table 5.2.

By means of one example we show that the throughput of the truncated extra back-off model is extremely robust with respect to $n$ when $n \geq 2k + 1$. In Figure 5.13 the throughput has been plotted against the number of nodes $n$ for fixed $\eta = 2$. We see that the largest difference in these values is around 0.0003, which is very insignificant compared to the throughputs that are 200 to 800 times as high. The difference is most likely solely the result of simulation
Figure 5.12: The effect of increasing $\eta$, with different situations shown, where the last two graphs show stability of all nodes.
imprecisions.

For every \( k \) we take the average throughput over all tested values of \( n \). Since we do not have means to approximate the optimal value, we will take the best evaluated \( \eta \) and set this as \( \hat{\eta}_{opt} \). We note that in Section 4.5 we saw that for \( n = 3 \) and \( k = 1 \) and \( \eta \) a little less than \( \eta_{opt} \) the optimal throughput was achieved, but that the throughput was relatively stable around \( \eta_{opt} \), so that \( \theta(\eta_c) \) and \( \theta(\hat{\eta}_{opt}) \) do not differ much. We illustrate this by evaluating the throughput functions from (4.36). According to the simulations, we have found \( \eta = 1 \) to deliver the best throughput among the evaluated mean back-off times. This value differs reasonably from the actual \( \eta_{opt} \approx 0.93328 \), although the found throughput of 0.3750 hardly deviates from \( \theta_{2}(\hat{\eta}_{opt}) \approx 0.3751 \).

Taking the best evaluated value of \( \eta \) yields Figure 5.14. Note that these should be seen as sharp lower bounds, as we just argued. Again, we perform a linear regression on \( 1/\theta \) to find the constants \( a \) and \( b \) that give the best regression curve \( f(k) = (a + bk)^{-1} \). These values turn out to equal \( a = 1.44445 \) and \( b = 1.37562 \).

5.5 Express forwarding

For express forwarding we do not have any analytical results, except for the nearly trivial case of \( n = 3 \) and \( k = 1 \). Therefore we will base our evaluation of express forwarding on simulation results only. We will again see that the throughput is accurately approximated by a function of the form \( f(k) = (a + bk)^{-1} \).

In Figure 5.15 the simulation results are depicted. The linear regression on \( 1/\theta \) is also shown and we obtain the function \( f(k) = (1.49408 + 1.46303k)^{-1} \). Except for the case \( k = 1 \) all simulation results show to be almost exactly on the curve of \( f(k) \) and we agree by visual inspection upon the accuracy of this regression.

5.6 Comparison of the various flow control schemes

In this section we will finally compare the different flow control mechanisms, based on simulations. We have plotted all regression functions and simulation results in Figure 5.16. We easily see the ordering of flow control schemes for \( k = 1, 2, \ldots, 8 \). The uncontrolled version performs worst, as expected, with basic extra back-off showing a good improvement, followed closely by express forwarding and truncated extra back-off. The protocol best tested in the simulations is windowed flow control. The regression constants \( a \) and \( b \) from \( f(k) = (a + bk)^{-1} \) are shown in Table 5.3. According to Figure 5.16 they seem valid for all values of \( k \geq 2 \). Only for \( k = 1 \) the simulation outcomes diverge significantly from the regressed curve for express forwarding and both extra back-off schemes. If we assume this regression to give an accurate extrapolation, we need only look at the value of \( b \) to order the throughput of the various back-off schemes for \( k \rightarrow \infty \). When we order the value of \( b \) for the various flow control protocols we see the same ordering as we already saw in the graph, suggesting that the graphs do not intersect as \( k \) grows. By comparing the values of \( b \) we can also calculate the improvement that each of the flow control schemes makes compared to the uncontrolled case, as \( k \rightarrow \infty \). These improvements are also seen in Table 5.3 and are denoted by \( \lim_{k \rightarrow \infty} \Delta \% \). We see that the expected gain in throughput of windowed flow control could increase up to over 60%, while truncated back-off times might reach over 45% improvement, express forwarding nearly 40% and basic extra back-off could increase the throughput with nearly a quarter. In Table 5.4
5.6. Comparison of the various flow control schemes

Figure 5.13: Throughput of the truncated back-off scheme with $\eta = 2$ for $k = 2, 3, \ldots, 8$ for an increasing number of nodes $n$
Figure 5.14: Regression of the best throughputs found versus the blocking range $k$ for the truncated extra back-off scheme

Figure 5.15: Regression of the best throughputs found versus the blocking range $k$ for express forwarding
5.6. Comparison of the various flow control schemes

Table 5.3: Constants $a$ and $b$ for various back-off scheme, obtained by means of regression, with $f(k) = (a + bk)^{-1}$ as shown in Figure 5.16

<table>
<thead>
<tr>
<th>Flow control mechanism</th>
<th>$a$</th>
<th>$b$</th>
<th>$\lim_{k \to \infty} \Delta %$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncontrolled</td>
<td>1.27535</td>
<td>2.01349</td>
<td>0.0%</td>
</tr>
<tr>
<td>Windowed flow control</td>
<td>1.23844</td>
<td>1.25705</td>
<td>60.2%</td>
</tr>
<tr>
<td>Basic extra back-off</td>
<td>1.29552</td>
<td>1.63384</td>
<td>23.2%</td>
</tr>
<tr>
<td>Truncated extra back-off</td>
<td>1.44445</td>
<td>1.37562</td>
<td>46.4%</td>
</tr>
<tr>
<td>Express forwarding</td>
<td>1.49408</td>
<td>1.46303</td>
<td>37.6%</td>
</tr>
</tbody>
</table>

we see the improvements as directly calculated from the simulation results, for $k = 1, 2, ..., 8$. We note that again the same ordering seems to hold.

What strikes us is that all interesting analytical results found for $k = 1$ and $n = 3$, such as equivalence of the systems for windowed flow control, the achieved stability for certain values of $\eta$ for truncated exponential back-off times, et cetera, seem to be valid for greater values of $n$ and $k$ as well.

The only thing that does not hold is the result of express forwarding compared to extra back-off, as seen in (4.1). For $n = 3$ and $k = 1$ we had the relation that the basic extra back-off scheme would always have at least the same throughput as express forwarding. This seems violated for general $n$ and $k$. There is a logical explanation to this. The special relation (4.1) depends on both properties $n = 3$ and $k = 1$, and is not necessarily valid for other values of these parameters, as we will now argue. As explained in Section 4.1 the back-off scheme with $\eta \downarrow 0$ is equivalent to express forwarding. It is easily seen that this only holds for $k = 1$; a node $i$ using express forwarding that finishes a transmission gives node $i+1$ priority to start the next transmission, while a node operating with extra back-off does not distinguish between
the 2\(k\) nodes surrounding it, giving all of them an equal chance of capturing the channel after it goes into back-off. This is an essential distinction. However, already for \(k = 1\) we see that the throughput of express forwarding averaged over all tested values of \(n\) is higher than the throughput of the basic extra back-off scheme, albeit only slightly. This shows that relation (4.1) might not be valid for \(n > 3\).

Now to see that (4.1) also relies on \(n = 3\), we will consider a system with \(n \geq 4\) nodes. When node 2 uses express forwarding and finishes a transmission, while node 4 is still active, it will block both node 1 and itself, while giving node 3 the opportunity to grab the channel, which node 3 would if it were not for the transmission of node 4. Seeing that node 3 does not start a transmission, node 2 will have probability 1/2 of transmitting another packet if available, since it will have an equal competition with node 1. If the same situation occurred for the basic extra back-off scheme with \(\eta \downarrow 0\), node 1 would start trying to access the channel at the same time as node 3, so that node 2 does not have any chance at all. Note that this reasoning heavily relies on the existence of node 4.

It shows in the case of \(\eta \downarrow 0\) that in extra back-off a node that has just finished gives more chance to previous nodes than express forwarding does, so that nodes earlier in the chain have an advantage compared to those in express forwarding. This would suggest queues to build up faster for extra back-off than for express forwarding. It is uncertain if this comparison also holds for general \(\eta\), although the simulation results seem to support it.
Chapter 6

Conclusions

6.1 Conclusions

In this thesis we have analyzed and compared various methods of flow control. We described a general mathematical model in Chapter 2 to which we applied flow control mechanisms. The emphasis was on windowed flow control and extra back-off schemes. We took the case of $n = 3$ nodes and blocking range $k = 1$ as a starting point for the analysis of both types of flow control.

6.1.1 Windowed flow control

Windowed flow control allows only a fixed number of packets to be in the system. This implies that the queue of no single node can grow without bound when the window size $W$ is finite. We analyzed the maximum achievable throughput for windowed flow control in Chapter 3. In Conjecture 1 we argued how increasing $W$ would lead to higher throughput. With this in mind, we aimed to find the limiting throughput for increasing $W$. We argued in Conjecture 2 that the cyclic mesh system with node 2 saturated, a system we showed to have stable queues for nodes 1 and 3 in Appendix B, is equivalent to the mesh chain with windowed flow control when $W \to \infty$. Subsequently we found an upper bound to the throughput of the equivalent system which, together with a lower bound, implied a relative improvement on the throughput of about 1/3 compared to the uncontrolled version.

In Section 5.3 we saw the generalization to larger values of $n$ and $k$. Similarly as in the case with a small system, it turned out that node $(n - k)$ would become saturated as the window size tended to infinity. The increase in throughput compared to the uncontrolled protocol grows larger when $k$ is increased, up to about 60%. The influence of $n$ on the throughput is significant, as observed from the fact that the throughput oscillates around some value. As $n$ grows, the throughput seems to stabilize.

6.1.2 Extra back-off

For the flow control mechanism based on forcing a node that just transmitted a packet into an extra back-off time, we also conducted analyses, which are found in Chapter 4. We proposed four back-off schemes, of which three with exponential back-off times. First there was the one (i) in which all nodes have an exponential back-off time, then another one (ii) in which only the first two nodes have an exponential back-off time and we also looked at the scheme (iii) in
which all nodes have an exponential back-off time, with the small modification that the back-off time can be terminated by the arrival of a new packet. The mean back-off time $\eta$ is taken variable. Schemes (i) and (iii) seem the most useful ones to implement. Since scheme (i) could only be numerically analyzed by a matrix-geometric method using quasi-birth-death theory, scheme (ii) was introduced as an accurate approximation. For scheme (ii) we succeeded in finding the exact throughput as a function of $\eta$. In both scheme (i) and scheme (ii) node 2 would become saturated, while node 3 would remain stable.

In scheme (iii) node 2 was found to become stable when $\eta$ was chosen large enough. We note the interesting critical value $\eta_c = \sqrt{5} - 1$, also found using QBD theory. Scheme (iii) performs better than the other two back-off schemes.

In general extra back-off times have two main advantages. They both allow the just transmitted packet to travel outside of the blocking range of the node in back-off, but they also allow a new packet to travel towards the node. Truncating the back-off times makes the idea of latter advantage more efficient, since it makes sure a node does not wait too long for the next packet to be able to arrive.

We did simulations for schemes (i) and (iii) and saw the features described above recur in the case of general $n$ and $k$. We saw that for both back-off schemes the exact value of $n$ with $n \geq 2k + 1$ had no significant influence on throughput. Scheme (iii) continued to perform better than scheme (i).

Finally we also presented a preliminary of an analysis of truncated deterministic back-off times, which follows the same protocol as scheme (iii), with the distinction that the exponential back-off times are replaced by deterministic ones. In comparison to scheme (iii), this scheme does not provide stability for node 2 for any value of $\eta$. We give an approximation in which the variation of one random variable $R$ is neglected. This still yields accurate approximations, as is shown with simulation results. The optimal throughput outcomes are a little lower for this scheme than those for the exponential scheme, but higher than for scheme (i).

### 6.1.3 Comparison

We also did simulations for express forwarding as a back-off scheme and used the uncontrolled version as a measure of improvement for the ones with flow control. Windowed flow control provides the highest throughput gain, followed by truncated back-off times. Express forwarding comes right after that, and the basic back-off scheme achieves the smallest improvement of the simulated flow control mechanisms. All schemes seemed to have $f(k) = (a + bk)^{-1}$ as an accurate approximation for the throughput, with $a$ and $b$ some constants.

Although windowed flow control might seem to be the most attractive flow control mechanism, it needs a constant feedback from the last node $n$ to node 1, since node 1 must know how many packets have left the chain when it wants to transmit a new packet. Feedback must either travel back through the same channels or through some external feedback channel. This is in conflict with the requirements of WMNs that every node needs to have a simple autonomous protocol.

Since the goal of IEEE 802.11s is to make a small modification to the access protocol and keep the autonomous behavior of the mesh nodes, we recommend the truncated version of extra back-off times. In the model that we used, stability can be achieved by choosing $\eta$ large enough. The achievable throughput gain comes in as second with an increase of over 45% as $k \rightarrow \infty$. 
6.2 Recommendations

The research in this thesis has provided a start to the general analysis of flow control protocols in WMNs. We have used a simplified model and we will propose some improvements and generalizations. Naturally, the research is not finished after this thesis and therefore we propose a few directions to continue.

- Throughout this thesis we have modeled the access algorithms of the various nodes as exponentially distributed times with rate $\lambda \to \infty$. In reality these access algorithms are more complicated, as shown in Section 1.1.2. A first improvement would be to take $\lambda = 5\mu = 5$, since the time access algorithms cost should not be completely diminished.

- The effect of more general topologies should be researched, starting with two-way traffic in a mesh chain or two chains crossing each other.

- We assumed transmissions to succeed with probability $p = 1$. The system dynamics might change when $p < 1$.

- We argued for windowed flow control that letting $W \to \infty$ was equivalent to taking node $(k + 1)$ saturated when $n = 2k + 1$. The rigorous proofs of Conjectures 2 and 3 are still absent, but the arguments we gave could function as a sketch of these proofs. Perhaps these could be extended to give a proof of Hypothesis 1.

- We assumed the first node to always be saturated. In contrast, the first node could also have an arrival process and that way have a maximum rate of letting packets enter the system. The influence of lowering the rate might show resemblance with the case of windowed flow control, when the input rate of the system is chosen approximately equal to $\theta_{n,k,\infty}$ and we might even observe node $(n - k)$ functioning as a bottleneck node.

- In this thesis we introduced a blocking range $k$ and based our results for the extra back-off schemes on the value of $k$. In reality the value of $k$ is something we need to be extremely careful with. In a WMN the nodes are not necessarily aware of the topology of the network, so they might not be aware of the number of nodes in their transmission and carrier-sensing range and thus cannot base their optimal extra back-off time on this.

- When discussing extra back-off, we have assumed the distribution of back-off to be free to choose. In reality this might not be the case. The idea of giving nodes extra back-off is still in a premature phase, but it is reasonable to assume that a standardization would also require one specific type of back-off. The most reasonable way might be to give the node a uniformly distributed extra back-off time, with as only parameter the mean packet length.

- Another point of interest is the correlation between the distributions used for extra back-off and the distribution used for the transmission times. It is interesting to see whether Theorem 2 still holds for more general distributions of the back-off time and/or the transmission time.

- We only presented a short introduction to the analysis of deterministic back-off times. This can be generalized, as proposed in Section 4.6.4.
• Systems with express forwarding could be analyzed for more general $n$ and $k$. The fact that the system is nearly trivial for $n = 3$ and $k = 1$ gives hope that this might be easier than the other schemes we discussed.

The most obvious continuation, however, is to extend our analysis to more general values of $n$ and $k$. As we saw in Chapter 5 most of the results for $n = 3$ and $k = 1$ also seem to be valid in more general settings.
Appendix A

Calculations on Markov processes for windowed flow control

A.1 The case of $n = 3$, $k = 1$, $W = 3$

In this subsection we will find the stationary distribution of the Markov process described in Section 3.1.2. The process is shown in Figure 3.2, with all states and all nonzero transition rates. As said in Section 3.1.2, we put $p_{\{0,0,3\}^o} = 15x$, $p_{\{1,1,1\}^i} = 33y$ and $p_{\{2,0,1\}^o} = 16z$. We will now show how to solve the equations by equalizing inflow and outflow. We chose not to construct transition rate matrices and solve the system in the general way, because this way we will see similarities between different values of $W$ that help us understand the system better.

From $p_{\{0,0,3\}^o} = 15x$ we readily find the following stationary probabilities:

\[
\begin{align*}
p_{\{0,1,2\}^i} &= p_{\{0,0,3\}^o} = 15x, \\
p_{\{0,2,1\}^i} &= 2p_{\{0,1,2\}^i} = 30x, \\
p_{\{0,3,0\}^i} &= 2p_{\{0,2,1\}^i} = 60x, \\
p_{\{1,2,0\}^o} &= p_{\{0,3,0\}^i} = 60x.
\end{align*}
\]

With the same argument as we used for $W = 2$ we obtain that the inflow of states $\{1,2,0\}^o$ and $\{1,2,0\}^i$ is equal and since they both have outflow rate 1, we find:

\[
p_{\{1,2,0\}^i} = p_{\{1,2,0\}^o} = 60x.
\]

Now we can continue to solve the system and find the value of $y$ expressed in $x$.

\[
\begin{align*}
p_{\{1,0,2\}^o} &= \frac{1}{2} (p_{\{1,1,1\}^i} + p_{\{0,0,3\}^o}) = \frac{33y + 15x}{2}, \\
p_{\{0,1,2\}^o} &= \frac{1}{2} p_{\{0,2,1\}^i} + p_{\{1,0,2\}^o} = \frac{1}{2} \cdot 30x + \frac{33y + 15x}{2} = \frac{33y + 45x}{2}, \\
33y &= p_{\{1,1,1\}^i} = \frac{1}{2} \frac{2}{3} p_{\{0,1,2\}^o} + \frac{2}{3} p_{\{1,2,0\}^i} = \frac{1}{3} \cdot \frac{33y + 45x}{2} + \frac{1}{3} \cdot 60x = \frac{11y + 55x}{2}.
\end{align*}
\]

From the last equation we can conclude that $x = y$, which simplifies the last array of equations.
We know that the sum of the stationary probabilities of all states must equal 1, so:

\[ p_{(1,0,2)^o} = 24x, \]
\[ p_{(0,1,2)^o} = 39x, \]
\[ p_{(1,1,1)^i} = 33x. \]

We continue by using \( p_{(2,0,1)^o} = 16z \). This results in:

\[ p_{(1,1,1)^o} = \frac{1}{2} \left( \frac{2}{3} p_{(0,1,2)^o} + \frac{2}{3} p_{(1,2,0)^i} + p_{(2,0,1)^o} \right) = \frac{1}{2} \left( 40x + 26x + 16z \right) = 33x + 8z, \]
\[ p_{(3,0,0)^o} = p_{(2,0,1)^o} = 16z, \]
\[ p_{(2,1,0)^i} = \frac{1}{2} p_{(3,0,0)^o} = 8z, \]
\[ p_{(2,1,0)^o} = \frac{1}{2} p_{(3,0,0)^o} + p_{(1,1,1)^o} = 33x + 16z, \]
\[ p_{(0,2,1)^o} = \frac{1}{2} p_{(0,3,0)^i} + p_{(1,1,1)^o} = \frac{1}{2} \cdot 60x + 33x + 8z = 63x + 8z, \]
\[ 60x = p_{(1,2,0)^o} = \frac{1}{2} p_{(2,1,0)^o} + \frac{1}{2} p_{(0,2,1)^o} = \frac{1}{2} (33x + 16z) + \frac{1}{2} (63x + 8z) = 48x + 12z, \]

and the last equation yields \( z = x \). This amounts to the following simplified stationary probabilities:

\[ p_{(1,1,1)^o} = 41x, \]
\[ p_{(3,0,0)^o} = 16x, \]
\[ p_{(2,1,0)^i} = 8x, \]
\[ p_{(2,1,0)^o} = 49x, \]
\[ p_{(0,2,1)^o} = 71x. \]

We know that the sum of the stationary probabilities of all states must equal 1, so:

\[ x = (15 + 15 + 30 + 60 + 60 + 60 + 33 + 24 + 39 + 16 + 16 + 8 + 41 + 71 + 49)^{-1} = \frac{1}{537}. \]

We have now determined the stationary distribution. We find the throughput of the chain by adding up the stationary probabilities of all states in which the inner node is active:

\[ \theta_{3,1,3} = \frac{1}{537} (15 + 30 + 60 + 60 + 33 + 8) = \frac{206}{537} \approx 0.383613. \]

### A.2 The case of \( n = 3, k = 1, W = 4 \)

The same approach as was used in the previous subsection is followed for the case in which \( W = 4 \). In Figure 3.3 the Markov process is depicted with all states and transition rates. We now put \( p_{(0,0,4)^o} = 1381x, p_{(1,2,1)^o} = 7413y \) and \( p_{(1,1,2)^i} = 3191z \). Again these values are taken to avoid most of the fractions. We can begin with:

\[ p_{(0,1,3)^i} = p_{(0,0,4)^o} = 1381x, \]
\[ p_{(0,2,2)^i} = 2p_{(0,1,3)^i} = 2762x, \]
\[ p_{(0,3,1)^i} = 2p_{(0,2,2)^i} = 5542x, \]
\[ p_{(0,4,0)^i} = 2p_{(0,3,1)^i} = 11048x, \]
\[ p_{(1,3,0)^o} = p_{(0,4,0)^i} = 11048x. \]
A.2. The case of $n = 3$, $k = 1$, $W = 4$

We see that $p_{(1,3,0)^o}$ is equal to $p_{(1,3,0)^i}$, because of the equal inflow and the same amount of outflow, so:

$$p_{(1,3,0)^i} = p_{(1,3,0)^o} = 11048x.$$ 

Now we see that

$$p_{(0,3,1)^o} = \frac{1}{2}p_{(0,4,0)^i} + p_{(1,2,1)^o} = 5524x + 7413y,$$

$$11048x = p_{(1,3,0)^o} = \frac{1}{2}p_{(0,3,1)^o} + \frac{1}{2}p_{(2,2,0)^o} = \frac{1}{2}(5524x + 7413y) + \frac{1}{2}p_{(2,2,0)^o}.$$ 

From the last equation we can conclude that

$$p_{(2,2,0)^o} = 16572x - 7413y.$$ 

In the same way we find $p_{(3,1,0)^o}$:

$$16572x - 7413y = p_{(2,2,0)^o} = p_{(1,2,1)^o} + \frac{1}{2}p_{(3,1,0)^o} = 7413y + \frac{1}{2}p_{(3,1,0)^o},$$

from which we can see that

$$p_{(3,1,0)^o} = 33144x - 29652y,$$

$$p_{(2,2,0)^i} = \frac{1}{2}p_{(3,1,0)^o} = 16572x - 14826y,$$

$$p_{(2,1,1)^i} = \frac{1}{3}p_{(2,2,0)^i} = 5524x - 4942y.$$ 

Now we will start using that $p_{(1,1,2)^i} = 3191z$:

$$p_{(1,0,3)^o} = \frac{1}{2}(p_{(0,0,4)^o} + p_{(1,1,2)^i}) = \frac{1}{2}(1381x + 3191z),$$

$$p_{(0,1,3)^o} = \frac{1}{2}p_{(0,2,2)^i} + p_{(1,0,3)^o} = \frac{1}{2}(2762x + 1381x + 3191z) = \frac{4143x + 3191z}{2},$$

$$p_{(2,0,2)^o} = \frac{1}{2}(p_{(2,1,1)^i} + p_{(1,0,3)^o}) = \frac{1}{2}(5524x - 4942y + \frac{1381x + 3191z}{2}) = \frac{12429x - 9884y + 3191z}{2}.$$ 

Furthermore we see that:

$$3191z = p_{(1,1,2)^i} = \frac{1}{3}p_{(0,1,3)^o} + \frac{1}{3}p_{(1,2,1)^i} = \frac{1}{3} \cdot \frac{4143x + 3191z}{2} + \frac{1}{3}p_{(1,2,1)^i},$$

so that we get

$$p_{(1,2,1)^i} = 9573z - \frac{4143x + 3191z}{2} = \frac{15955z - 4143x}{2}.$$ 

This leads to

$$\frac{15955z - 4143x}{2} = p_{(1,2,1)^i} = \frac{1}{3}p_{(0,2,2)^o} + \frac{1}{3}p_{(1,3,0)^i} = \frac{1}{3}(11048x + p_{(0,2,2)^o}),$$
which in turn gives
\[ p_{(0,2,2)^o} = 3 \cdot \frac{15955z - 4143x}{2} - 11048x = \frac{47865z - 34525x}{2}. \]

We continue by expressing stationary probabilities in terms of \( x, y \) and \( z \), until all states have been determined. After that we can eliminate two variables.

Thus
\[
P_{(1,2)^o} = \frac{1}{2} (\frac{2}{3} p_{(0,1,3)^o} + \frac{2}{3} p_{(1,2,1)^t} + p_{(2,0,2)^o})
\]
\[
= \frac{1}{3} \cdot 11048x + \frac{1}{3} \cdot 47865z - 34525x + \frac{1}{2} p_{(2,1,1)^o},
\]
so that
\[
P_{(2,1,1)^o} = 2 \cdot (7413y - \frac{1}{3} (11048x + \frac{47865z - 34525x}{2}))
\]
\[
= 4143x + 14826y - 15955z,
\]
\[
33144x - 29652y = p_{(2,2,0)^o} = p_{(2,1,1)^o} + \frac{1}{2} p_{(4,0,0)^o} = 4143x + 14826y - 15955z + \frac{1}{2} p_{(4,0,0)^o},
\]

which finally implies
\[
p_{(4,0,0)^o} = 2(33144x - 29652y - 4143x - 14826y + 15955z) = 58002x - 88956y + 31910z,
\]
\[
p_{(3,1,0)^t} = \frac{1}{2} p_{(4,0,0)^o} = 29001x - 44478y + 15955z,
\]
\[
p_{(3,0,1)^o} = p_{(4,0,0)^o} = 58002x - 88956y + 31910z.
\]

So now we have found all stationary probabilities expressed in terms of \( x, y \) and \( z \). If we now look at states \( \{0, 2, 2\}^o \) and \( \{2, 1, 1\}^o \), we will find the following set of equations:
\[
\frac{47865z - 34525x}{2} = p_{(0,2,2)^o} = \frac{1}{2} p_{(0,3,1)^t} + p_{(1,1,2)^o}
\]
\[
= \frac{1}{2} \cdot 5524x + \frac{12429x - 9884y + 28719z}{8},
\]
\[
8286x + 29652y - 31910z = 2p_{(2,1,1)^o} = p_{(1,1,2)^o} + p_{(3,0,1)^o} + \frac{2}{3} p_{(2,2,0)^t}
\]
\[
= \frac{12429x - 9884y + 28719z}{8} + 58002x - 88956y + 31910z.
\]

Or simplified:
\[
0 = -172625x + 9884y + 162741z,
\]
\[
0 = -498541x + 1037820y - 539279z.
\]

Simple linear algebra now shows that this set of equations is equivalent to \( x = y = z \). We can write all stationary distributions as a function of only \( x \):
A.2. The case of \( n = 3, k = 1, W = 4 \)

\[
\begin{align*}
P\{0,4,0\}^i &= 11048x, & P\{1,1,2\}^i &= 3191x, \\
P\{0,3,1\}^i &= 5524x, & P\{1,1,2\}^o &= 3908x, \\
P\{0,3,1\}^o &= 12937x, & P\{1,0,3\}^o &= 2286x, \\
P\{0,2,2\}^i &= 2762x, & P\{2,2,0\}^i &= 1746x, \\
P\{0,2,2\}^o &= 6670x, & P\{2,2,0\}^o &= 9159x, \\
P\{0,1,3\}^i &= 1381x, & P\{2,1,1\}^i &= 582x, \\
P\{0,1,3\}^o &= 3667x, & P\{2,1,1\}^o &= 3014x, \\
P\{0,0,4\}^o &= 1381x, & P\{2,0,2\}^o &= 1434x, \\
P\{1,3,0\}^i &= 11048x, & P\{3,0,1\}^o &= 956x, \\
P\{1,3,0\}^o &= 11048x, & P\{3,1,0\}^i &= 478x, \\
P\{1,2,1\}^i &= 5906x, & P\{3,1,0\}^o &= 3492x, \\
P\{1,2,1\}^o &= 7413x, & P\{4,0,0\}^o &= 956x.
\end{align*}
\]

Again, because of the property that all probabilities should sum up to 1, we have:

\[
x = (11048 + 5524 + 12937 + 2762 + 6670 + 1381 + 3667 + 1381 + 11048 + 11048 + 5906 + 7413 + 3191 + 3908 + 2286 + 1746 + 9159 + 582 + 3014 + 1434 + 956 + 478 + 3492 + 956)^{-1}
\]

\[
= \frac{1}{111987}.
\]

Now taking the sum over the stationary probabilities of all states in which node 2 is active, the states with an \( i \), we get the throughput:

\[
\theta_{3,1,4} = (11048 + 5524 + 2762 + 1381 + 11048 + 5906 + 3191 + 1746 + 582 + 478)x = \frac{43666}{111987} \approx 0.38992.
\]
Appendix A. Calculations on Markov processes for windowed flow control
Appendix B

Proof of stability of the cyclic mesh chain with saturated node 2

In this appendix we will prove Theorem 1. We first prove an auxiliary lemma in Appendix B.1, then proceed to the actual proof in Appendix B.2.

B.1 Generalization of the reflection principle

First, we will use a generalization of the reflection principle that Feller uses to prove the ballot theorem [16].

**Lemma 3** (Generalization of the reflection principle). Consider a discrete-time random walk, in which a particle either goes one up or one down every time unit. We take $x_1 > r, x_2 > r, t_1 < t_2,$ and $x_1 - x_2 \equiv t_1 - t_2 \pmod{2}$. By $(t, x)$ we mean that the particle at position $x$ at time $t$. The number of routes from $(t_1, x_1)$ to $(t_2, x_2)$, in which the particle never hits $(t, r)$ for $t_1 \leq t \leq t_2$ is equal to

$$N(t_2 - t_1, x_2 - x_1) - N(t_2 - t_1, x_2 - (2r - x_1)),$$

where $N(t, x) = \binom{t}{\frac{x}{2}}$.

**Proof.** To get from $(t_1, x_1)$ to $(t_2, x_2)$, the particle must take $t_2 - t_1$ steps, of which the number of steps up, denoted as $n_u$, minus the number of steps down, denoted as $n_d$, must equal $x_2 - x_1$. So we have the following system of equalities:

\[
\begin{align*}
n_u + n_d &= t_2 - t_1 \\
n_u - n_d &= x_2 - x_1.
\end{align*}
\]

From this we can readily see that

\[
\begin{align*}
n_u &= \frac{t_2 - t_1 + x_2 - x_1}{2} \\
n_d &= \frac{t_2 - t_1 - x_2 + x_1}{2}.
\end{align*}
\]

Note that from the constraint $x_1 - x_2 \equiv t_1 - t_2 \pmod{2}$ it follows that $n_u$ and $n_d$ are integers. This now means that there are $\binom{n_u + n_d}{n_d} = N(n_u + n_d, n_u - n_d)$ possibilities in which the steps up and down can be ordered, each corresponding to a unique path from $(t_1, x_1)$ to $(t_2, x_2)$. 

103
However, this might also include paths that do reach \((t, r)\) for some \(t\) with \(t_1 \leq t \leq t_2\). Therefore we need to count the paths that hit or cross \(r\) at some time and subtract this number from the total number of paths. The way to do this is depicted in Figure B.1.

![Figure B.1: A visualization of the reflection principle](image)

A path from \((t_1, x_1)\) hitting or crossing \(r\) will do so for the first time at a certain moment \(t^*\), at which the random walk will be at \((t^*, r)\). We will now introduce a bijective mapping from this type of paths to one that is easier to count. If we mirror the random walk up to the point \(t^*\) in \(r\), as is done in Figure B.1, we obtain a new starting position \((t_1, 2r - x_1)\), and in general every position of the walk \((t, y)\) with \(t_1 \leq t \leq t^*\) is now mapped onto \((t, 2r - y)\). The rest of the path, with \(t^* \leq t \leq t_2\), is unchanged.

This way every path from \((t_1, x_1)\) to \((t_2, x_2)\) hitting or crossing \(r\) is mapped onto a path from \((t_1, 2r - x_1)\) to \((t_2, x_2)\). This mapping is bijective, since every path hits or crosses \(r\) a first time and the only thing that happens is a partial mirroring in \(r\) up to this unique point.

So to find all paths from \((t_1, x_1)\) to \((t_2, x_2)\) hitting or crossing \(r\), we count all paths from \((t_1, 2r - x_1)\) to \((t_2, x_2)\). We get new \(n_u^*\) and \(n_d^*\) for these kinds of paths, namely:

\[
\begin{align*}
    n_u^* &= \frac{t_2 - t_1 + x_2 - 2r + x_1}{2}, \\
    n_d^* &= \frac{t_2 - t_1 - x_2 + 2r - x_1}{2}.
\end{align*}
\]

The conclusion is that there are \(N(n_u + n_d, n_u - n_d) - N(n_u^* + n_d^*, n_u^* - n_d^*)\) paths from \((t_1, x_1)\) to \((t_2, x_2)\) neither hitting nor crossing \(r\). Expressing this in \(t_j, x_j,\) and \(r\) for \(j = 1, 2\) gives the desired expression.

\[\square\]

### B.2 Proof of Theorem 1

We want to apply Proposition 1 to provide a proof to Theorem 1. We first need to discretize time in the Markov process. Then we will use Lemma 3 to find lower bounds to the probabilities seen in (3.3). In this we distinguish between two cases, found in Sections B.2.1 and B.2.2.
B.2. Proof of Theorem 1

In the end we conclude the proof in Section B.2.3.

We will approach this problem by analyzing cycles, in which a cycle is considered to start at the moment that node 2 wins the competition for the channel right after a transmission from node 1 and/or node 3. This brings us to a discrete-time Markov chain, in which the duration of a cycle is not of importance. The only variables that distinguish the states are the numbers of packets at nodes 1 and 3 at the start of a cycle, resulting in the state space

$$E \subset \{(x_1, x_3) \mid x_1, x_3 \geq 0\} \cap \mathbb{Z}^2.$$

We write strict subset, since we will now argue that not all combinations of \((x_1, x_3)\) are possible.

We distinguish between two cases. When node 2 grabs the channel, either node 1 or node 3 has just finished a transmission and before the end of this transmission the other one had gone empty. In case (i) node 3 was the last one active, so that at the moment the cycle begins we have \(x_1 = 1\) packets at queue 1 and \(x_3 \geq 0\) packets at queue 3. In case (ii) we have that node 1 was the last one active, resulting in \(x_1 \geq 0\) and \(x_3 = 0\). This gives us

$$E = \{(1, x_3) \mid x_3 \geq 0\} \cup \{(x_1, 0) \mid x_1 \geq 0\} \cap \mathbb{Z}^2.$$

For both cases we will show that the expected net change in the number of packets is negative.

B.2.1 Analysis of case (i)

We have \(x_1 = 1\) and assume \(x_3 = X \gg 0\). At the start of the cycle node 2 will start transmitting its packets to node 3. The number of packets it transmits after the first one is geometrically distributed with parameter \(2/3\). This means that the increase of packets at node 3 during node 2’s busy period is equal to \(1 + \text{Geo}(2/3)\), where the support of the geometric distribution is assumed to include 0. The expected increase then equals \(3/2\) and the new number of packets at node 3 is \(x_3 = X + 1 + \text{Geo}(2/3) \gg 0\).

We will now inspect the probabilities of node 3 transmitting \(j\) packets, with \(j \geq 1\), before node 2 wins the competition for the channel again. As either node 1 or node 3 must have stopped transmitting for node 2 to be able to take over, we only consider the events of this being the case for node 1. This reduction results in a lower bound to the desired probabilities, since it excludes the possibility of node 3 going empty first. We define:

$$p_j = P\{\text{node 3 transmits } j \text{ packets during one cycle and does not go empty}\}.$$

For \(j < X\) we see that \(p_j\) coincides with the desired probability. However, since we are mostly interested in the sum of packets at both node 1 and node 3, as will be seen later, we also want node 1 to transmit \(j\) packets, so that the total number of packets decreases with \(j\). This is only the case when node 3 does not go empty.

We define \(A_j\) as the set of possible orders of nodes finishing in a cycle resulting in node 1 going empty first and node 3 transmitting \(j\) packets in total. Since the number of packets transmitted by node 2 is unknown, for notation’s sake we will always start with a single 2, meaning that node 2 has finished its transmission period. The cycle ends when nodes 1 and 3 do not win the competition, so we always end an order with another 2. So the order 2132 represents the cycle in which after the \(1 + \text{Geo}(2/3)\) transmissions of node 2, nodes 1 and 3
start transmitting with node 1 finishing first, followed by node 3, after which node 2 grabs
the channel again starting the next cycle.

Now we will distinguish between two cases. First we will look at the case (i-a) in which
node 1 is the first to transmit its packet and thus becomes empty, then we will look at the
case (i-b) in which node 3 finishes first, resulting in 2 packets at node 1. According to this
partition of the possible orders, \( A_j \) can also be partitioned into \( A^a_j \) and \( A^b_j \), with \( A^a_j \cup A^b_j = P_j \)
and \( A^a_j \cap A^b_j = \emptyset \).

(i-a) We will now find all possible orders in which the nodes can finish so that \( j \) packets
are transmitted. Then we will calculate the probabilities that these orders occur. We will
be looking at the probabilities conditioned on the event that node 1 is the first to finish a
transmission and call these probabilities \( p^a_j \):

\[
p^a_j = \text{P}\{\text{node 3 transmits } j \text{ packets during one cycle and does not go empty}
and node 1 finishes first | node 1 finishes first}\}.
\]

Since in case (i-a) node 1 is the first to complete a transmission and node 3 does not go
empty, every order must start with order 2-1, after which node 1 is empty so that node 3 is
the next to finish its ongoing transmission. This results in 213. Since node 3 does not go
empty, node 1 must do so in order for node 2 to grab the channel at the end of the cycle.
Because of this every order must end with 132.

For \( j = 1 \) we see there is only one possible order, \( A^a_1 = \{2132\} \). We take for granted that
node 1 finishes first. Then with probability 1 node 3 is the next to finish. After that node 2
takes over the channel with probability 1/3 since there are packets at both nodes 1 and 3,
resulting in \( p^a_1 = 1/3 \).

For \( j = 2 \) there is also only one order satisfying the constraints, \( A^a_2 = \{213132\} \). The
probability of this one happening is \( p^a_2 = 2/3 \cdot 1/2 \cdot 1/3 = 1/9 \).

We will also show this for \( j = 3 \), in which case more than one order is possible. We find
\( A^a_3 = \{21313132, 21331132\} \). Only the fourth and fifth nodes to finish their transmissions are
different here. The first order has probability \( 2/3 \cdot 1/2 \cdot 2/3 \cdot 1/2 \cdot 1/3 = 1/27 \), whereas the
second one has probability \( 2/3 \cdot 1/2 \cdot 1/2 \cdot 1/2 \cdot 1/3 = 1/36 \), so that \( p^a_3 = 1/27 + 1/36 = 7/108 \).

We see that there need to be \( j \) 1’s and \( j \) 3’s in between two 2’s, which represent the \( j \)
packets both transmit. Since the first 1 and 3 and the last 1 and 3 are fixed, there are \( j - 2 \)
of both of them left that need to be placed in a certain order. We can see this as a random
walk comprising \( 2j - 4 \) steps, where the position is equal to the number of packets at node 1.
When node 3 finishes a transmission this number is increased by 1 and when node 1 finishes
a transmission it is decreased by 1. The random walk can never go below 0, since a negative
number of packets at node 1 is impossible. As is seen in Figure B.2, the non-fixed part of the
random walk starts at (2, 1) and ends at (2j - 2, 1).

We see that apart from the case in which node 1 is not transmitting, the probabilities
of either node 1 or node 3 finishing first are both equal to 1/2. So in general the random
walk goes up with probability 1/2 and down with probability 1/2 as well. However, when
node 1 does go empty, i.e. the random walk hits 0, the probability that belongs to going up is
different. With probability 1 the random walk goes up again, and then with probability 2/3
it continues, while with probability 1/3 the random walk ends. So to find the probability of a
certain order with \( j > 1 \), we need to count how often the random walk reaches zero after the
first transmission. For an order \( r \in P^n_j \), we define \( \gamma(r) \) as the number of times node 1 goes
empty, except for the fixed parts at the start and the end, as can be seen in Figure B.2. For $j > 1$ we start with probability $2/3$ for the event that nodes 1 and 3 grab the channel again after the first transmission of node 3, and end with probabilities $1/2$ for node 1 finishing its last transmission before node 3 does, and $1/3$ for the event of node 2 grabbing the channel at the end of the cycle. The probability of having order $r$ now becomes

$$P_r = \frac{2}{3} \left( \frac{1}{2} \right)^{2j-4-\gamma(r)} \left( \frac{2}{3} \right)^{\gamma(r)} \frac{1}{2} \cdot \frac{1}{3}$$

We still have $p_1 = 1/3$ as initialization. This sum, however, is hard to analyze and thus we will give a lower bound to $p^*_j$, by replacing (B.1) by:

$$P^*_r = \frac{1}{9} \left( \frac{1}{2} \right)^{2j-4},$$

so that $P\{r\} \geq P^*\{r\}$. Here we do not count the number of times that 0 is hit, so that the $(2/3)^{\gamma(r)}$ is replaced by $(1/2)^{\gamma(r)}$. Now we only have to count the paths from $(2, 1)$ to $(2j-2, 1)$ that do not cross 0. This number is, with Lemma 3, equal to

$$\binom{2j-4}{j-2} - \binom{2j-4}{j-4} = \frac{(2j-4)!}{(j-2)!(j-4)!} \left( \frac{1}{(j-2)(j-3)} - \frac{1}{j(j-1)} \right)$$

$$= \frac{(2j-4)!}{(j-2)!(j-4)!} \cdot \frac{(j^2 - j) - (j^2 - 5j + 6)}{j(j-1)(j-2)(j-3)}$$

$$= 2 \cdot \frac{(2j-4)!}{(j-2)!(j-4)!} \cdot \frac{2j - 3}{j(j-1)(j-2)(j-3)}$$

$$= 2 \cdot \frac{(2j-3)!}{(j-1)!(j-2)!} \cdot \frac{1}{j}$$

$$= 2 \cdot \frac{2j-3}{j(j-1)}.$$
This gives us the following lower bounds

\[ p_j^a = \sum_{r \in P_j^a} P\{r\} \geq \sum_{r \in P_j^a} P^*\{r\} = p_j^{a*} = \frac{1}{9} \left( \frac{1}{2} \right)^{2j-4} \frac{2}{j} \left( \frac{2j-3}{j-1} \right) \]

\[ = \frac{32}{9j} \left( \frac{1}{2} \right)^{2j} \left( \frac{2j-3}{j-1} \right) \]

(B.2)

for \(2 \leq j < X\). For \(j = 1\) we keep \(p_1^{a*} = p_1^a = 1/3\).

(i-b) What we have not yet treated are the cases in which node 3 is the first one to transmit its packet, resulting in two packets at node 1. We take the same approach as for case (i-a). We are now interested in the following probabilities:

\[ p_j^b = P\{\text{node 3 transmits } j \text{ packets during one cycle and does not go empty and node 3 finishes first} \mid \text{node 3 finishes first}\} \]

We see that every order now starts with 23 and still ends with 132. Every order contains \(j\) 1’s and \(j\) 3’s. This results in \(A_1^b = \emptyset\), \(A_2^b = \{231132\}\), and \(A_3^b = \{23113132, 23131132, 23311132\}\). The corresponding probabilities are now \(p_1^b = 0\), \(p_2^b = 1/2 \cdot 1/2 \cdot 1 \cdot 1/3 = 1/12\) and since \(P\{23113132\} = 1/2 \cdot 1/2 \cdot 1/2 \cdot 1/3 = 1/36\) and \(P\{23131132\} = P\{23311132\} = 1/2 \cdot 1/2 \cdot 1/2 \cdot 1/2 \cdot 1 \cdot 1/3 = 1/48\), we have \(p_3^b = 1/36 + 2/48 = 5/72\).

We will construct another random walk that keeps track of the number of packets at node 1. Now the first step is not down but up. This means the random walk will go from \((0, 1)\) to \((1, 2)\). After that, every step up will happen again with probability 1/2 and every step down with probability 1/2 as well, except for when the random walk reaches 0. Then the random walk goes up with probability 1, after which it ends with probability 1/3, and continues with probability 2/3. Since every order in \(A_j^b\) ends with 132, the last two steps are fixed, the random walk first goes down, then up. In total we have \(2j\) steps, with the first and the last two fixed. Therefore we need to find paths from \((1, 2)\) to \((2j - 2, 1)\) that never go below zero. Here we also need to count the number of times that the random walk hits 0. We will condition on the first time it hits 0.

The same simplification is made as in case (i-a). The amount of times the random walk reaches 0 is not counted for the new probabilities \(p_j^{b*}\), so that for every order \(r \in A_j^b\) we find a lower bound to the probability as follows

\[ P\{r\} \geq P^*\{r\} = \left( \frac{1}{2} \right)^{2j-2} \frac{1}{3}. \]

All we have to count is the number of paths from \((1, 2)\) to \((2j - 2, 1)\) that never go below 0.
and with Lemma 3 this turns out to be

\[
\binom{2j-3}{j-1} - \binom{2j-3}{j-4} = \frac{(2j-3)!}{(j-4)!(j-1)!} \left( \frac{1}{(j-3)(j-2)} - \frac{1}{j(j+1)} \right)
= \frac{(2j-3)!}{(j-4)!(j-1)!} \cdot \frac{j^2 + j - j^2 - 5j + 6}{(j-3)(j-2)j(j+1)}
= 3 \cdot \frac{(2j-3)!}{(j-2)!(j-1)!} \cdot \frac{2j-2}{j(j+1)}
= 3 \cdot \frac{(2j-2)!}{(j-2)!j!} \cdot \frac{1}{j+1}
= \frac{3}{j+1} \binom{2j-2}{j}.
\]

This results in the following lower bound to the probabilities:

\[
p_j^b = \sum_{r \in A_j^b} P\{r\} \geq \sum_{r \in A_j^b} P^*\{r\} = p_j^{b*} = \frac{1}{3} \left( \frac{1}{2} \right)^{2j-2} \frac{3}{j+1} \binom{2j-2}{j}
= \frac{4}{j+1} \left( \frac{1}{2} \right)^{2j} \binom{2j-2}{j}
\]

for \(2 \leq j < X\).

Figure B.3: A random walk corresponding to case (i-b) with \(2t+1\) the first moment 0 is hit.

Now we have found lower bounds to the probabilities of both cases. Since both cases occur equally likely, depending on whether node 1 or node 3 finishes first, we can combine (B.2) and (B.3) as follows:

\[
p_j = P\{\text{node 3 transmits } j \text{ packets during one cycle and does not go empty}\}
= \frac{1}{2} p_j^a + \frac{1}{2} p_j^b,
\]
for $1 \leq j < X$, so that $p^*_1 = 1/6$. Now we can also find lower bounds on the $p_j$’s for $2 \leq j < X$:

$$p_j \geq p_j^* = \frac{1}{2}p_j^{*n} + \frac{1}{2}p_j^{*s} = \frac{1}{2} \cdot \frac{32}{9j} \left( \frac{1}{2} \right)^{2j} \left( \frac{2j-3}{j-1} \right) + \frac{1}{2} \cdot \frac{4}{j+1} \left( \frac{1}{2} \right)^{2j} \left( \frac{2j-2}{j} \right) = \left( \frac{1}{2} \right)^{2j} \left( \frac{16}{9} \cdot \frac{1}{j} \cdot \frac{2j-2}{j} \right) \left( \frac{2j-3}{j-1} \right) + \frac{2}{j+1} \left( \frac{2j-2}{j} \right) = \left( \frac{1}{2} \right)^{2j} \left( \frac{2j-2}{j} \right) \left( \frac{8}{9(j-1)} + \frac{2}{j+1} \right) = \left( \frac{1}{2} \right)^{2j} \left( \frac{2j-2}{j} \right) \frac{26j-10}{9j^2-9}. $$

Note that the event of node 3 transmitting $j$ packets and not going empty is equal to the event of node 1 transmitting $j$ packets with node 3 not going empty. For every $X \gg 0$ packets at node 3 at the start of the cycle we have:

$$\mathbb{E}\{\text{packets transmitted by node 1} \mid x_3 = X\} \geq \sum_{j=1}^{N} jp_j \geq \sum_{j=1}^{N} jp_j^* \geq \sum_{j=2}^{N} j \left( \frac{1}{2} \right)^{2j} \left( \frac{2j-2}{j} \right) \frac{26j-10}{9j^2-9} \quad (B.4)$$

for any $N < X$.

We now require the following lemma:

**Lemma 4.** The following infinite sum diverges:

$$\sum_{j=2}^{\infty} \left( \frac{1}{2} \right)^{2j} \left( \frac{2j-2}{j} \right) = \infty.$$ 

**Proof.** To prove this we use Stirling’s approximation which says:

$$n! \sim \sqrt{2\pi n} \left( \frac{n}{e} \right)^n.$$

This means that as $n$ increases, the ratio of $n!$ and the right-hand expression tends to one. Now we will use this formula to show that

$$\left( \frac{1}{2} \right)^{2j} \left( \frac{2j-2}{j} \right) \sim a_j^{-\frac{1}{2}},$$
for some constant \( a \in \mathbb{R}^+ \).

\[
\left( \frac{1}{2} \right)^{2j} \binom{2j-2}{j} \sim \left( \frac{1}{2} \right)^{2j} \frac{\sqrt{2\pi(2j-2)} \left( \frac{2j-2}{e} \right)^{2j-2}}{\sqrt{2\pi(j-2)} \left( \frac{j-2}{e} \right)^{j-2} \cdot \sqrt{2\pi j^j}} \\
\sim \left( \frac{1}{2} \right)^{2j} \frac{\sqrt{2\pi(2j-2)} (2j-2)^{2j-2}}{\sqrt{2\pi(j-2)}(j-2)^{j-2} \cdot \sqrt{2\pi j^j}} \\
\sim \frac{1}{\sqrt{2\pi}} \left( \frac{1}{2} \right)^{2j} \frac{2j-2}{j^2-2j} \left( \frac{2j-2}{j-2} \right)^{j-2} \left( \frac{2j-2}{j} \right)^j \\
\sim 1 \frac{1}{2j} \frac{2j-2}{j^2-2j} \left( \frac{2j-2}{j-2} \right)^{j-2} \left( \frac{2j-2}{j} \right)^j \sim 1 \frac{1}{2j} \frac{2j-2}{j^2-2j} \left( \frac{2j-2}{j-2} \right)^{j-2} \left( \frac{2j-2}{j} \right)^j 
\]

We use the identity

\[
\lim_{n \to \infty} \left( 1 + \frac{c}{n} \right)^n = e^c
\]

to see that

\[
\left( \frac{2j-2}{j-2} \right)^{j-2} \sim 2^{j-2} \left( 1 + \frac{1}{j-2} \right)^{j-2} \sim 2^{j-2} e
\]

for the first term, and for the second term:

\[
\left( \frac{2j-2}{j} \right)^j \sim 2\left( 1 - \frac{1}{j} \right)^j \sim 2 e^{j-1}.
\]

Plugging these into the last step, we find:

\[
\left( \frac{1}{2} \right)^{2j} \binom{2j-2}{j} \sim \frac{1}{\sqrt{2\pi}} \left( \frac{1}{2} \right)^{2j} \left( \frac{2j-2}{j^2-2j} \cdot \frac{2j-2}{e} \cdot 2^{j-2} e \cdot 2 e^{j-1} \right)
\]

\[
\sim \frac{1}{2\sqrt{\pi}} \left( \frac{1}{2} \right)^{2j} \frac{2j-2}{j^2-2j} \cdot 2^{2j-2} e^{2j}
\]

\[
\sim \frac{1}{4\sqrt{\pi}} \sqrt{\frac{j-1}{j^2-2j}}
\]

so that \( a = \frac{1}{4\sqrt{\pi}} \). Because of this behaviour, we conclude that

\[
\sum_{j=2}^{\infty} \left( \frac{1}{2} \right)^{2j} \binom{2j-2}{j} = \infty.
\]
We can now use Lemma 4 to see that the expected number of packets transmitted by node 1, as found in (B.4), diverges to infinity as $N \to \infty$. We have

$$
\lim_{N \to \infty} \sum_{j=2}^{N} \left( \frac{1}{2} \right)^{2j} \binom{2j-2}{j} \frac{26j-10}{9j^2-9} = \sum_{j=2}^{\infty} \left( \frac{1}{2} \right)^{2j} \binom{2j-2}{j} \frac{26j^2-10j}{9j^2-9} \\
\geq \sum_{j=2}^{\infty} \left( \frac{1}{2} \right)^{2j} \binom{2j-2}{j} \cdot \frac{26}{9} \\
= \frac{26}{9} \sum_{j=2}^{\infty} \left( \frac{1}{2} \right)^{2j} \binom{2j-2}{j} \\
= \infty,
$$

where we note that the minimum of $\frac{26j^2-10j}{9j^2-9}$ for $j \geq 2$ and integer is reached at $j = 5$ and equals $25/9$.

Now let $\varepsilon_1 > 0$. Because of the divergence of the sum, there exists a value $X^*$, so that for all $X > X^*$ we can always find an $N < X$ such that $\sum_{j=1}^{N} j p_j^* > 3/2 + \varepsilon_1$. Combining this with (B.4) gives:

$$
\mathbb{E}\{\text{packets transmitted by node 1 during one cycle} \mid X_3 = X\} > \frac{3}{2} + \varepsilon_1.
$$

This means that when the number of packets at node 3 at the start of a cycle is large enough, the net flux is negative, or:

$$
\mathbb{E}\{\text{increase in packets at nodes 1 and 3 during one cycle} \mid X_3 > X^*\} \\
= \mathbb{E}\{1 + \text{Geo}(2/3)\} - \mathbb{E}\{\text{packets transmitted by node 1 during one cycle} \mid X_3 > X^*\} \\
< \frac{3}{2} - \frac{3}{2} - \varepsilon_1 \\
= -\varepsilon_1.
$$

(B.5)

After treating case (ii) the consequences of this will be discussed.

### B.2.2 Analysis of case (ii)

The second case, in which $x_3 = 0$ at the start of the cycle and $x_1 \geq 0$, is considerably easier. We will assume $x_1 = Y \gg 0$. Because of this, the number of packets transmitted by node 2 at the start of the cycle equals $1 + \text{Geo}(2/3)$, which has expectation $3/2$. We will now look at the number of packets that node 1 transmits before node 3 becomes empty. We condition on the number of packets transmitted by node 2.

Say node 2 transmits $z \geq 1$ packets, then node 3 has exactly $z$ packets in its queue when the outer nodes grab the channel and node 1 still has $Y \gg 0$ packets. This time we will assume node 3 to go empty during the cycle. We define

$$
q_j = \mathbb{P}\{\text{node 1 transmits } j \text{ packets and node 3 goes empty}\}.
$$

Note that the constraint of node 3 going empty is evident for $j < Y$, since it is impossible for node 1 to go empty transmitting less than $Y$ packets. We will concentrate on the case of $j < Y$. 
While both nodes 1 and 3 are transmitting, each of them finishes first with probability 1/2. Using the same notation as in case (i), we can derive the possible orders of finishing. Since node 3 must go empty, node 1 must be the last one transmitting, making every order end with 12. This means that the only possible order for \( j = 1 \) is \( 2 \, 3 \, 3 \ldots \, 3 \, 1 \, 2 \). The probability of this is \((1/2)^{z+1}\).

For \( j = 2 \) we have the same order, except that another transmission of node 1 must have ended before, during, or right after the \( z \) transmissions of node 3. This can happen in \( (z+1) \) ways and the probability for each of them is \((1/2)^{z+2}\). This includes the one with order \( 2 \, 3 \, 3 \ldots \, 3 \, 1 \, 2 \), but with a different reason. After node 1 finished its first transmission, it will lose the channel to node 2 with probability 1/2, and keep it with probability 1/2 as well. This happens each time that node 1 wants to transmit another packet while node 3 is already empty, so that the probability for all orders is the same.

This can be generalized to

\[
q^*_j = P\{\text{node 1 transmits } j, \text{ node 3 goes empty} \mid \text{node 2 transmits } z\} = \left(\frac{z + j - 1}{j - 1}\right) \left(\frac{1}{2}\right)^{z+j}.
\]

If we do not truncate at \( j = Y \), it is a probability distribution and therefore we note that \( \sum_{j=1}^{\infty} q^*_j = 1 \). We will solve the sum \( f(z) = \sum_{j=1}^{\infty} jq^*_j \) by means of induction. It is easily seen that \( f(0) = \sum_{j=1}^{\infty} (1/2)^j = 2 \). Next we will express \( f(z) \) in terms of \( f(z - 1) \):

\[
f(z) = \sum_{j=1}^{\infty} j\left(\frac{z + j - 1}{j - 1}\right) \left(\frac{1}{2}\right)^{z+j}
\]

\[
= \frac{1}{2} \sum_{j=1}^{\infty} j \cdot \frac{z + j - 1}{z} \cdot \frac{(z + j - 2)!}{(z-1)!(j-1)!} \left(\frac{1}{2}\right)^{z+j-1}
\]

\[
= \frac{1}{2} \sum_{j=1}^{\infty} j \cdot \frac{(z + j - 2)!}{(z-1)!(j-1)!} \left(\frac{1}{2}\right)^{z+j-1} + \frac{1}{2} \sum_{j=1}^{\infty} j \cdot \frac{j(j-1)}{z} \cdot \frac{(z + j - 2)!}{(z-1)!(j-1)!} \left(\frac{1}{2}\right)^{z+j-1}
\]

\[
= \frac{1}{2} \sum_{j=1}^{\infty} j \cdot \frac{(z + j - 2)!}{z!(j-2)!} \left(\frac{1}{2}\right)^{z+j-1} + \frac{1}{2} \sum_{j=1}^{\infty} j \cdot \frac{(z + j - 2)!}{z!(j-2)!} \left(\frac{1}{2}\right)^{z+j-1}
\]

\[
= \frac{1}{2} \sum_{j=1}^{\infty} jq^{z-1} + \frac{1}{2} \sum_{k=1}^{\infty} (k + 1) \cdot \frac{(z + k - 1)!}{z!(k-1)!} \left(\frac{1}{2}\right)^{z+k}
\]

\[
= \frac{1}{2} f(z - 1) + \frac{1}{2} \sum_{k=1}^{\infty} kq^*_k + \frac{1}{2} \sum_{k=1}^{\infty} q^*_k
\]

\[
= \frac{1}{2} f(z - 1) + \frac{1}{2} f(z) + \frac{1}{2}.
\]

This leads to \( f(z) = f(z-1) + 1 \). Combining this result with \( f(0) = 2 \) brings us to \( f(z) = z+2 \). This could have also been seen by noting that node 3 takes on average \( z \) time units for transmitting its packets, in which node 1 is also expected to transmit \( z \) entire packets. Then node 1 will finish the transmission it is busy with, after which it will battle with node 2 for the channel, transmitting another geometrically distributed number of packets with mean 1, which in total would also amount to an expected number of \( z + 2 \) transmitted packets.
Multiplying by the probability $P\{1 + \text{Geo}(2/3) = z\} = (\frac{1}{3})^{z-1}(\frac{2}{3})$ and summing over all possible values of $z$ gives:

$$
\sum_{z=1}^{\infty}(z+2)(\frac{1}{3})^{z-1}(\frac{2}{3}) = 2\sum_{z=1}^{\infty}(\frac{1}{3})^{z-1}(\frac{2}{3}) + \sum_{z=1}^{\infty}z(\frac{1}{3})^{z-1}(\frac{2}{3}) = 2 + \frac{1}{2/3} = \frac{7}{2}.
$$

Now we construct the sequence $b_n = \sum_{j=1}^{n}\sum_{z=1}^{\infty}(\frac{1}{3})^{z-1}(\frac{2}{3})jq_j^z$ for $n \geq 1$. This sequence gives a lower bound to the expected number of packets transmitted by node 1 when there are $Y$ packets at node 1 and $Y > n$. This sequence is non-decreasing and converges as follows:

$$
\lim_{n \to \infty} b_n = \lim_{n \to \infty} \sum_{j=1}^{n}\sum_{z=1}^{\infty}(\frac{1}{3})^{z-1}(\frac{2}{3})jq_j^z
= \sum_{z=1}^{\infty}(\frac{1}{3})^{z-1}(\frac{2}{3})\lim_{n \to \infty}\sum_{j=1}^{n}jq_j^z
= \sum_{z=1}^{\infty}(\frac{1}{3})^{z-1}(\frac{2}{3})(z+2)
= \frac{7}{2}.
$$

Now let $0 < \delta < 2$ and define $\varepsilon_2 = 2 - \delta > 0$. Then there is a $Y^*$ such that for all $Y > Y^*$ one can find an $N < Y$ such that $b_N > 7/2 - \delta = 3/2 + \varepsilon_2$. Again we can conclude that the net flux is negative:

$$
\mathbb{E}\{\text{increase in packets at nodes 1 and 3 during one cycle} \mid X_1 > Y^*\}
= \mathbb{E}\{1 + \text{Geo}(2/3)\} - \mathbb{E}\{\text{packets transmitted by node 1 during one cycle} \mid X_1 > Y^*\}
< \frac{3}{2} - \frac{3}{2} - \varepsilon_2
= -\varepsilon_2.
$$

\section{Conclusion of proof}

In Figure B.4 is a graph that shows the implication of the conclusions of cases (i) and (ii). It has values of $x_1$ and $x_3$ at the start of a cycle on the axes and shows that when they are outside the blue region they will return to it. The blue region represents the starting values with $x_3 < X^*$ and $x_1 < Y^*$. A cycle always begins and ends on an axis (where we neglect the one packet at node 3 for schematic reasons).

When a cycle starts at the $x_1$-axis, the expected position at the end of the cycle is either lower on the $x_1$-axis, or on the $x_3$-axis in case not node 3 but node 1 goes empty. In that case the expected increase in packets at node 3 is finite, namely $3/2$.

If the cycle starts at the $x_3$-axis, the expected position at the end of the cycle will be either lower on the $x_3$-axis or on the $x_1$ axis. If not node 1 but node 3 goes empty, the expected number of packets at the two nodes together is still lower than in the beginning, therefore the slope of the arrow going to the $x_1$-axis is larger than $-1$ and the new value of $x_1$ is expected to be lower than the old value of $x_3$.

We now use Proposition 1 to formalize this.
Proof of Theorem 1. To apply Proposition 1, we use the Markov chain as was described, with state space $E$ and we take the function $h : E \rightarrow \mathbb{R}$, as $h(x_1, x_3) = x_1 + x_3$. We furthermore define

$$E_0 = \{(1, x_3) \mid 0 \leq x_3 \leq X^* \} \cup \{(x_1, 0) \mid 0 \leq x_1 \leq Y^* \} \cap \mathbb{Z}^2 \subset E.$$ 

It is straightforward to see that $\inf_{(x_1, x_3) \in E} h(x_1, x_3) = 0 > -\infty$, because the number of packets is always non-negative.

The first constraint (3.2) says that the expected increase in packets is always finite for every state $(x_1, x_3) \in E$. Since the number of packets transmitted per cycle by node 2 follows a geometric distribution with finite mean, this constraint is satisfied.

Next take $\varepsilon = \min\{\varepsilon_1, \varepsilon_2\} > 0$, where $\varepsilon_1$ and $\varepsilon_2$ are such as found in (B.5) and (B.6), respectively. Then we apply (B.5) and (B.6) for $(x_1, x_3) \in E \setminus E_0$ to see that the average decrease in packets is larger than $\varepsilon$ and therefore also constraint (3.3) is satisfied.

The consequence is that the Markov chain with states $(x_1, x_3)$ is positive recurrent, so that the expected time of returning to $E_0$ is always finite after having left this region. This means that in this Markov chain, embedded on the start of the cycles, the queues of nodes 1 and 3 are stable. Since the expected duration of a cycle is finite, the start of another cycle will always occur within finite expected time from an arbitrary epoch. From that point on we can consider the embedded chain only and conclude that $E_0$ will always be revisited within finite expected time. Therefore the Markov process considered at arbitrary moments is also positive recurrent. $\square$
Appendix C

Determining an upper bound to $\theta_{3,1,\infty}$ by analysis of the Markov process

We will now present a sharp bound to $\theta_{3,1,\infty}$ by means of an extensive analysis of the Markov process corresponding to the system described. First we will introduce the state space and the transition rates.

We note that the state space is two-dimensional. Since the second node is saturated, we do not have to record how many packets are at that node. Instead, we only keep track of the number of packets at the first and third nodes. This gives us the following state space for this Markov process:

$$S_{3,1,\infty} = \{ \{x_1, x_3\}^a \mid x_1, x_3 \geq 0, a \in A(x_1, x_3) \},$$

with

$$A(x_1, x_2, x_3) = \begin{cases} \{i\}, & \text{if } x_1 + x_3 = 0, \\ \{i, o\}, & \text{else.} \end{cases}$$

We find that for $x_1, x_3 > 0$ the transition rates are now given by

$$r_{\{x_1, x_3\}^i, \{x_1, x_3+1\}^i} = 1/3,$$
$$r_{\{x_1, x_3\}^i, \{x_1, x_3+1\}^o} = 2/3,$$
$$r_{\{x_1, x_3\}^o, \{x_1+1, x_3-1\}^o} = 1,$$
$$r_{\{x_1, x_3\}^o, \{x_1-1, x_3\}^o} = 1.$$

For $x_1 = 0$ and $x_3 \geq 0$ we obtain for the states $\{0, x_3\}^a$ with $a = i$:

$$r_{\{0, x_3\}^i, \{0, x_3+1\}^i} = 1/2,$$
$$r_{\{0, x_3\}^i, \{0, x_3+1\}^o} = 1/2.$$

And for $x_3 > 0$ and $a = o$ we see:

$$r_{\{0, x_3\}^o, \{1, x_3-1\}^i} = 1/3,$$
$$r_{\{0, x_3\}^o, \{1, x_3-1\}^o} = 2/3.$$
Next, for \( x_3 = 0 \) and \( x_1 > 1 \), we have:

\[
\begin{align*}
    r_{\{x_1,0\}^i,\{x_1-1,0\}^i} &= 1/2, \\
    r_{\{x_1,0\}^i,\{x_1-1,0\}^o} &= 1/2.
\end{align*}
\]

This leaves us with the following initial rates:

\[
\begin{align*}
    r_{\{0,1\}^o,\{1,0\}^i} &= 1/2, \\
    r_{\{0,1\}^o,\{1,0\}^o} &= 1/2, \\
    r_{\{1,0\}^i,\{0,0\}^i} &= 1.
\end{align*}
\]

The Markov process corresponding to these states and transitions is shown in Figure 3.4. In Theorem 1 it was seen that this system is stable, i.e. that the queues of nodes 1 and 3 will not tend to infinity. This means that the state \( \{0,0\}^i \) has a positive stationary probability.

The Markov process for this system is so complicated that we will not calculate the exact throughput, but instead give bounds for it. We will start by analyzing the stationary distribution. First of all we see that

\[
p_{\{0,0\}^i} = p_{\{1,0\}^o} = p_{\{1,0\}^i}.
\]

The first equality we find by equalizing the inflow and outflow of \( \{0,0\}^i \). The latter is found in the same way as we have seen in the cases of \( W = 2, 3, 4 \). The inflow of states \( \{1,0\}^o \) and \( \{1,0\}^i \) is equal and they both have a total transition rate of 1 going out. This means that the stationary probabilities are equal. We will now investigate the rest of the states, starting with the ones in which the first queue is empty and the inner node is active:

\[
\begin{align*}
    p_{\{0,1\}^i} &= \frac{1}{2} p_{\{0,0\}^i}, \\
    p_{\{0,2\}^i} &= \frac{1}{2} p_{\{0,1\}^i} = \left( \frac{1}{2} \right)^2 p_{\{0,0\}^i}, \\
    &\vdots \\
    p_{\{0,x_3\}^i} &= \left( \frac{1}{2} \right)^{x_3} p_{\{0,0\}^i}.
\end{align*}
\]

For the states in which the outer nodes are active, we find:

\[
\begin{align*}
    p_{\{0,1\}^o} &= \frac{1}{2} p_{\{0,0\}^i} + p_{\{1,1\}^o}, \\
    p_{\{0,2\}^o} &= \frac{1}{2} p_{\{0,1\}^i} + p_{\{1,2\}^o} = \left( \frac{1}{2} \right)^2 p_{\{0,0\}^i} + p_{\{1,2\}^o}, \\
    &\vdots \\
    p_{\{0,x_3\}^o} &= \left( \frac{1}{2} \right)^{x_3} p_{\{0,0\}^i} + p_{\{1,x_3\}^o}.
\end{align*}
\]

Next we will be looking at the states in which there is one packet at the first node. These
probabilities are a little more complex.

\[
\begin{align*}
P_{(1,1)^i} &= \frac{1}{3} P_{(1,0)^i} + \frac{1}{3} P_{(0,2)^o} + \frac{1}{3} \left( \frac{1}{3} \right)^2 P_{(1,0)^i} + \frac{1}{3} \left( \frac{1}{2} \right)^2 P_{(0,0)^i} + \frac{1}{3} P_{(2,2)^o}, \\
P_{(1,2)^i} &= \frac{1}{3} P_{(1,1)^i} + \frac{1}{3} P_{(0,3)^o} \\
&= \frac{1}{3} \left( \frac{1}{3} P_{(1,0)^i} + \frac{1}{3} \left( \frac{1}{2} \right)^2 P_{(0,0)^i} + \frac{1}{3} P_{(1,2)^o} \right) + \frac{1}{3} \left( \frac{1}{3} \left( \frac{1}{2} \right)^3 P_{(0,0)^i} + P_{(1,3)^o} \right), \\
&= \left( \frac{1}{3} \right)^2 P_{(1,0)^i} + \left( \frac{1}{3} \right)^2 \left( \frac{1}{2} \right)^2 P_{(0,0)^i} + \left( \frac{1}{3} \right)^2 P_{(1,2)^o} + \left( \frac{1}{3} \right) \left( \frac{1}{2} \right)^3 P_{(0,0)^i} + \left( \frac{1}{3} \right) P_{(1,3)^o}, \\
&= \left( \frac{1}{3} \right)^2 P_{(1,0)^i} \left( \frac{1}{3} \right)^2 \left( \frac{1}{2} \right)^2 + \left( \frac{1}{3} \right) \left( \frac{1}{2} \right)^3 \right) P_{(0,0)^i} + \left( \frac{1}{3} \right)^2 P_{(1,2)^o} + \left( \frac{1}{3} \right) P_{(1,3)^o}, \\
\vdots \\
P_{(1,x_3)^i} &= \left( \frac{1}{3} \right)^x P_{(1,0)^i} + \left( \frac{1}{3} \right)^x \left( \frac{1}{2} \right)^2 + \ldots + \left( \frac{1}{3} \right) \left( \frac{1}{2} \right)^n \right) P_{(0,0)^i} \\
&\quad + \left( \frac{1}{3} \right)^x P_{(1,2)^o} + \ldots + \left( \frac{1}{3} \right) P_{(1,x_3+1)^o} \\
&= \left( \frac{1}{3} \right)^x P_{(1,0)^i} + \frac{1}{2} \sum_{n=1}^{x_3} \left( \frac{1}{3} \right)^n \left( \frac{1}{2} \right)^n P_{(0,0)^i} + \frac{1}{2} \sum_{n=1}^{x_3} \left( \frac{1}{3} \right)^n P_{(1,x_3+2-n)^o} \\
&= \left( \frac{1}{3} \right)^x P_{(1,0)^i} + \frac{1}{2} \sum_{n=1}^{x_3} \left( \frac{1}{3} \right)^n \left( \frac{1}{2} \right)^n P_{(0,0)^i} + \frac{1}{2} \sum_{n=1}^{x_3} \left( \frac{1}{3} \right)^n P_{(1,x_3+2-n)^o} \\
&= \frac{1}{2} \left( \left( \frac{1}{2} \right)^x + \left( \frac{1}{3} \right)^x \right) P_{(0,0)^i} + \frac{1}{2} \sum_{n=1}^{x_3} \left( \frac{1}{3} \right)^n P_{(1,x_3+2-n)^o}. \quad (C.2)
\end{align*}
\]

This now depends on \( P_{(0,0)^i} \) and \( P_{(1,x)^o} \) for \( x = 2, \ldots, x_3 + 1 \). We will now be looking at these latter states, in which the outer nodes are active with only one packet in the first queue:

\[
\begin{align*}
2p_{(1,x_3)^o} &= \frac{2}{3} p_{(1,x_3-1)^i} + \frac{2}{3} p_{(0,x_3+1)^o} + p_{(2,x_3)^o}, \\
p_{(1,x_3)^o} &= \frac{1}{3} p_{(1,x_3-1)^i} + \frac{1}{3} p_{(0,x_3+1)^o} + \frac{1}{2} p_{(2,x_3)^o} \\
&= p_{(1,x_3)^i} + \frac{1}{2} p_{(2,x_3)^o} \\
&= \frac{1}{2} \left( \left( \frac{1}{2} \right)^x + \left( \frac{1}{3} \right)^x \right) p_{(0,0)^i} + \frac{1}{2} \sum_{n=1}^{x_3} \left( \frac{1}{3} \right)^n p_{(1,x_3+2-n)^o} + \frac{1}{2} p_{(2,x_3)^o}. \quad (C.3)
\end{align*}
\]

Next we will look at the stationary probabilities of states \( \{x_1, x_3\}^i \) for general \( x_1 \geq 2 \) and
$x_3 \geq 0$. These are not too complicated:

$$p_{(x_1,1)^i} = \frac{1}{3} p_{(x_1,0)^i},$$

$$p_{(x_1,2)^i} = \frac{1}{3} p_{(x_1,1)^i} = \left(\frac{1}{3}\right)^2 p_{(x_1,0)^i},$$

$$\vdots$$

$$p_{(x_1,x_3)^i} = \frac{1}{3} p_{(x_1,1)^i} = \left(\frac{1}{3}\right)^{x_3} p_{(x_1,0)^i}.$$  

The probabilities corresponding to the states in which the outer nodes are active are more complicated:

$$2p_{(x_1,x_3)^o} = \frac{2}{3} p_{(x_1,x_3-1)^i} + p_{(x_1-1,x_3+1)^o} + p_{(x_1+1,x_3)^o},$$

$$p_{(x_1,x_3)^o} = \left(\frac{1}{3}\right)^{x_3} p_{(x_1,0)^i} + \frac{1}{2} p_{(x_1-1,x_3+1)^o} + \frac{1}{2} p_{(x_1+1,x_3)^o}. \quad (C.4)$$

Because of the recurrence of the system, we know that the throughputs of all nodes are equal. When we define $p_i$ to be the stationary probability of only node $i$ being active, for $i = 1, 2, 3,$ and $p_{13}$ the probability of both nodes 1 and 3 being active, we have that:

$$p_1 + p_{13} = p_2 = p_3 + p_{13}. \quad (C.5)$$

We will use this to express the throughput in terms of only $p_{(0,0)^i}$. We can write $p_1$, $p_2$, $p_3$ and $p_{13}$ as sums of stationary probabilities:

$$p_1 = \sum_{x_3 \geq 1} p_{(0,x_3)^o},$$

$$p_2 = \sum_{x_1, x_3 \geq 0} p_{(x_1,x_3)^i},$$

$$p_3 = \sum_{x_1 \geq 1} p_{(x_1,0)^o},$$

$$p_{13} = \sum_{x_1, x_3 \geq 1} p_{(x_1,x_3)^o}.$$  

We will continue by rewriting $p_2$:

$$p_2 = \sum_{x_1, x_3 \geq 0} p_{(x_1,x_3)^i},$$

$$= \sum_{x_3=0}^{\infty} p_{(0,x_3)^i} + \sum_{x_3=0}^{\infty} p_{(1,x_3)^i} + \sum_{x_1=2}^{\infty} \sum_{x_3=0}^{\infty} p_{(x_1,x_3)^i},$$

$$= \sum_{x_3=0}^{\infty} \left(\frac{1}{2}\right)^{x_3} p_{(0,0)^i} + \sum_{x_3=0}^{\infty} p_{(1,x_3)^i} + \sum_{x_1=2}^{\infty} \sum_{x_3=0}^{\infty} \left(\frac{1}{3}\right)^{x_3} p_{(x_1,0)^i},$$

$$= \frac{1}{1 - \frac{1}{2}} p_{(0,0)^i} + \sum_{x_3=0}^{\infty} p_{(1,x_3)^i} + \sum_{x_1=2}^{\infty} \frac{1}{1 - \frac{1}{3}} p_{(x_1,0)^i},$$

$$= 2p_{(0,0)^i} + \sum_{x_3=0}^{\infty} p_{(1,x_3)^i} + \frac{3}{2} \sum_{x_1=2}^{\infty} p_{(x_1,0)^i}. \quad (C.6)$$
The first sum in (C.6), \( \sum_{x_3=0}^{\infty} p_{\{x_3\}}\), can be simplified:

\[
\sum_{x_3=0}^{\infty} p_{\{x_3\}} = p_{\{0\}} + \sum_{x_3=1}^{\infty} p_{\{x_3\}}
\]

\[
= p_{\{0\}} + \sum_{x_3=1}^{\infty} \left( \frac{1}{3} p_{\{x_3-1\}} + \frac{1}{3} p_{\{0,x_3+1\}} \right)
\]

\[
= p_{\{0\}} + \frac{1}{3} \sum_{x_3=0}^{\infty} p_{\{x_3\}} + \frac{1}{3} \sum_{x_3=2}^{\infty} p_{\{0,x_3\}}
\]

\[
= p_{\{0\}} + \frac{1}{3} \sum_{x_3=0}^{\infty} p_{\{x_3\}} + \frac{1}{3} \sum_{x_3=1}^{\infty} p_{\{0,x_3\}} - \frac{1}{3} p_{\{0,1\}}
\]

\[
= p_{\{0\}} + \frac{1}{3} \sum_{x_3=0}^{\infty} p_{\{x_3\}} + \frac{1}{3} p_3 - \frac{1}{3} p_{\{0,1\}}
\]

so that

\[
\frac{2}{3} \sum_{x_3=0}^{\infty} p_{\{x_3\}} = p_{\{0\}} + \frac{1}{3} p_3 - \frac{1}{3} p_{\{0,1\}},
\]

\[
\sum_{x_3=0}^{\infty} p_{\{x_3\}} = \frac{3}{2} p_{\{0\}} + \frac{1}{2} p_3 - \frac{1}{2} p_{\{0,1\}}.
\]  \(\text{(C.7)}\)

The other sum in (C.6) can also be rewritten, using that \( p_{\{x_1,0\}} = \frac{1}{2} p_{\{x_1+1,0\}} \) for \( x_1 \geq 1 \):

\[
\sum_{x_1=2}^{\infty} p_{\{x_1,0\}} = \sum_{x_1=2}^{\infty} \frac{1}{2} p_{\{x_1+1,0\}}
\]

\[
= \frac{1}{2} \sum_{x_1=3}^{\infty} p_{\{x_1,0\}}
\]

\[
= \frac{1}{2} \sum_{x_1=1}^{\infty} p_{\{x_1,0\}} - \frac{1}{2} p_{\{2,0\}} - \frac{1}{2} p_{\{1,0\}}
\]

\[
= \frac{1}{2} p_1 - \frac{1}{2} p_{\{2,0\}} - \frac{1}{2} p_{\{0,0\}}.
\]  \(\text{(C.8)}\)

In these deductions we have used that \( p_{\{0,0\}} = p_{\{1,0\}} = p_{\{1,0\}} \). Now we can substitute (C.7)
and (C.8) into (C.6), as follows, using that $p_1 = p_3$:

\[
p_2 = 2p_{(0,0)^i} + \left(\frac{3}{2}p_{(0,0)^i} + \frac{1}{2}p_3 - \frac{1}{2}p_{(0,1)^o}\right) + \frac{3}{2} \left(\frac{1}{2}p_1 - \frac{1}{2}p_{(2,0)^o} - \frac{1}{2}p_{(0,0)^i}\right)
\]

\[
= \frac{7}{2}p_{(0,0)^i} + \frac{5}{4}p_1 - \frac{1}{2}p_{(0,1)^o} + \frac{3}{4}p_{(2,0)^o} - \frac{3}{4}p_{(0,0)^i}
\]

\[
= \frac{11}{4}p_{(0,0)^i} + \frac{5}{4}p_1 - \frac{1}{4}p_{(0,1)^o} - \frac{3}{4}p_{(2,0)^o}
\]

Now $p_2$ has been expressed as a function of two stationary probabilities and $p_1$. Since the throughput can be written as $\theta_{3,1,\infty} = p_2 = p_1 + p_{13}$, we see that:

\[
p_1 + p_{13} = \frac{7}{4}p_{(0,0)^i} - \frac{1}{4}p_{(2,0)^o} + \frac{5}{4}p_1,
\]

\[
p_{13} = \frac{7}{4}p_{(0,0)^i} - \frac{1}{4}p_{(2,0)^o} + \frac{1}{4}p_1. \tag{C.9}
\]

Also, since there is always at least one active node in the system, it holds that $p_1 + p_2 + p_3 + p_{13} = 1$, and by writing $p_2 = p_1 + p_{13}$ and $p_3 = p_1$, this simplifies to $3p_1 + 2p_{13} = 1$, so that $p_{13} = \frac{1 - 3p_1}{2}$. Substituting this into (C.8) gives:

\[
\frac{1 - 3p_1}{2} = \frac{7}{4}p_{(0,0)^i} - \frac{1}{4}p_{(2,0)^o} + \frac{1}{4}p_1,
\]

\[
1 - 3p_1 = \frac{7}{2}p_{(0,0)^i} - \frac{1}{2}p_{(2,0)^o} + \frac{1}{2}p_1,
\]

\[
\frac{7}{2}p_1 = -1 + \frac{7}{2}p_{(0,0)^i} - \frac{1}{2}p_{(2,0)^o},
\]

\[
p_1 = \frac{2}{7} - p_{(0,0)^i} + \frac{1}{7}p_{(2,0)^o}.
\]

Because $\theta_{3,1,\infty} = p_1 + p_{13} = p_1 + \frac{1 - 3p_1}{2} = \frac{1 - p_1}{2}$, we can now compute the throughput as a function of $p_{(0,0)^i}$ and $p_{(2,0)^o}$:

\[
\theta_{3,1,\infty} = \frac{1 - p_1}{2}
\]

\[
= \frac{1}{2} - \frac{1}{2}\left(\frac{2}{7} - p_{(0,0)^i} + \frac{1}{7}p_{(2,0)^o}\right)
\]

\[
= \frac{1}{2} - \frac{1}{7} + \frac{1}{2}p_{(0,0)^i} - \frac{1}{14}p_{(2,0)^o}
\]

\[
= \frac{1}{14} \left(5 + 7p_{(0,0)^i} - p_{(2,0)^o}\right). \tag{C.10}
\]
Next we will find bounds for $p_{(2,0)^o}$ expressed in $p_{(0,0)^i}$. We note that:

\[
p_{(0,1)^o} = \frac{1}{2}p_{(0,0)^i} + p_{(1,1)^o}, \quad (C.11)
\]

\[
p_{(2,0)^o} = \frac{1}{2}p_{(3,0)^o} + p_{(1,1)^o}. \quad (C.12)
\]

If we now subtract (C.12) from (C.11), we get

\[
p_{(0,1)^o} - p_{(2,0)^o} = \frac{1}{2}p_{(0,0)^i} - \frac{1}{2}p_{(3,0)^o}.
\]

Since we know that $\frac{1}{2}p_{(0,1)^o} + \frac{1}{2}p_{(2,0)^o} = p_{(1,0)^o} = p_{(0,0)^i}$, we can get rid of the $p_{(0,1)^o}$ term:

\[
\frac{1}{2}p_{(0,0)^i} - \frac{1}{2}p_{(3,0)^o} = p_{(0,1)^o} - p_{(2,0)^o}
= (p_{(0,1)^o} + p_{(2,0)^o}) - 2p_{(2,0)^o}
= 2p_{(0,0)^i} - 2p_{(2,0)^o}.
\]

This, in turn, implies that

\[
\frac{1}{2}p_{(0,0)^i} - \frac{1}{2}p_{(3,0)^o} = 2p_{(0,0)^i} - 2p_{(2,0)^o}
= \frac{3}{2}p_{(0,0)^i} + \frac{1}{2}p_{(3,0)^o}
= 3p_{(0,0)^i} + 3p_{(0,0)^i}
> 3p_{(0,0)^i}
= \frac{3}{4}p_{(0,0)^i}.
\]

So we can see that (C.10) becomes

\[
\theta_{3,1,\infty} = \frac{1}{14} \left( 5 + 7p_{(0,0)^i} - p_{(2,0)^o} \right)
< \frac{1}{14} \left( 5 + 7p_{(0,0)^i} - \frac{3}{4}p_{(0,0)^i} \right)
= \frac{1}{14} \left( 5 + \frac{25}{4}p_{(0,0)^i} \right). \quad (C.13)
\]

Now we will find an upper bound for $p_{(0,0)^i}$ by expressing as many stationary probabilities as possible in terms of $p_{(0,0)^i}$. Adding those cannot result in a value bigger than 1, so this would give us an upper bound. First, it is seen that $p_{(1,0)^i} = p_{(1,0)^o} = p_{(0,0)^i}$, so

\[
p_{(1,0)^i} + p_{(1,0)^o} + p_{(0,0)^i} = 3p_{(0,0)^i}.
\]

Next, we see $\frac{1}{2}p_{(0,1)^o} + \frac{1}{2}p_{(2,0)^o} = p_{(1,0)^o}$, so that

\[
p_{(0,1)^o} + p_{(2,0)^o} = 2p_{(0,0)^i}.
\]
Appendix C. Determining an upper bound to $\theta_{3,1,\infty}$ by analysis of the Markov process

Next, with help of (C.1), it is clear that

$$\sum_{x_3=1}^{\infty} p_{(0,x_3)}^i = \sum_{x_3=1}^{\infty} \left( \frac{1}{2} \right)^{x_3} p_{(0,0)}^i$$

$$= \frac{1}{2} \frac{p_{(0,0)}^i}{1 - \frac{1}{2}}$$

$$= p_{(0,0)}^i.$$

In the same way, using (C.2), it follows that

$$\sum_{x_3=1}^{\infty} p_{(1,x_3)}^i = \sum_{x_3=1}^{\infty} \left( \frac{1}{2} \left( \left( \frac{1}{2} \right)^{x_3} + \left( \frac{1}{3} \right)^{x_3} \right) p_{(0,0)}^i + \frac{1}{2} \sum_{n=1}^{x_3} \left( \frac{1}{3} \right)^n p_{(1,x_3+2-n)}^i \right)$$

$$> \sum_{x_3=1}^{\infty} \left( \frac{1}{2} \left( \left( \frac{1}{2} \right)^{x_3} + \left( \frac{1}{3} \right)^{x_3} \right) p_{(0,0)}^i \right)$$

$$= \frac{1}{2} \left( \left( \frac{1}{2} \right)^{x_3} + \left( \frac{1}{3} \right)^{x_3} \right) p_{(0,0)}^i$$

$$= \frac{3}{4} p_{(1,0)}^i.$$

Now we will be looking at the states that are of the form $\{0,x_3\}$ with $x_3 \geq 2$, and the ones that are $\{1,x_3\}$ and $\{2,x_3\}$ for $x_3 \geq 1$, where we use (C.2), (C.3) and (C.4):

$$p_{(0,x_3)}^o = p_{(1,x_3)}^o + \left( \frac{1}{2} \right)^{x_3} p_{(0,0)}^i,$$

$$p_{(1,x_3)}^o = \frac{1}{3} p_{(0,x_3+1)}^o + \frac{1}{2} p_{(2,x_3)}^o + \left( \frac{1}{3} \right)^{x_3} p_{(1,0)}^i,$$

$$p_{(2,x_3)}^o > \frac{1}{2} p_{(1,x_3+1)}^o.$$

Summing the first equation over all possible values of $x_3$, gives:

$$a = \sum_{x_3=2}^{\infty} p_{(0,x_3)}^o$$

$$= \sum_{x_3=2}^{\infty} p_{(1,x_3)}^o + \sum_{x_3=2}^{\infty} \left( \frac{1}{2} \right)^{x_3} p_{(0,0)}^i$$

$$> \sum_{x_3=2}^{\infty} \left( \frac{1}{3} \right)^{x_3} p_{(1,0)}^i + \frac{1}{2} \left( \frac{1}{3} \right)^2 p_{(0,0)}^o i$$

$$= \frac{1}{3} \frac{p_{(1,0)}^i}{1 - \frac{1}{3}}$$

$$= \frac{1}{6} + \frac{1}{2} p_{(0,0)}^i$$

$$= \frac{2}{3} p_{(0,0)}^i.$$

$$= 2 \frac{2}{3} p_{(0,0)}^i.$$
We shall call this $a$ and let it help us finding the sums over the other equations, which we call $b$ and $c$:

\[
b = \sum_{x_3=1}^{\infty} p_{(1,x_3)^i} = \frac{1}{3} \sum_{x_3=2}^{\infty} p_{(0,x_3)^o} + \frac{1}{2} \sum_{x_3=1}^{\infty} p_{(2,x_3)^o} + \sum_{x_3=1}^{\infty} \left( \frac{1}{3} \right)^{x_3} p_{(1,0)^i}
\]

\[
= \frac{1}{3} a + \frac{1}{2} c + \frac{1}{3} \frac{1}{1 - \frac{1}{3}} p_{(1,0)^i}
\]

\[
> \frac{1}{3} \cdot \frac{2}{3} p_{(0,0)^i} + \frac{1}{2} c + \frac{1}{2} p_{(0,0)^i}
\]

\[
= \left( \frac{2}{9} + \frac{1}{2} \right) p_{(0,0)^i} + \frac{1}{2} c
\]

\[
= \frac{13}{18} p_{(0,0)^i} + \frac{1}{2} c.
\]

Now $b$ can be used to obtain a bound on $c$:

\[
c = \sum_{x_3=1}^{\infty} p_{(2,x_3)^o}
\]

\[
> \frac{1}{2} \sum_{x_3=2}^{\infty} p_{(1,x_3)^o}
\]

\[
= \frac{1}{2} b
\]

\[
> \frac{1}{2} \left( \frac{13}{18} p_{(0,0)^i} + \frac{1}{2} c \right)
\]

\[
= \frac{13}{36} p_{(0,0)^i} + \frac{1}{4} c.
\]

From this we can derive a lower bound on $c$ that only depends on $p_{(0,0)^i}$:

\[
\frac{3}{4} c > \frac{13}{36} p_{(0,0)^i}
\]

\[
c > \frac{13}{48} p_{(0,0)^i}.
\]

Finally, $b$ can also be bounded from below:

\[
b > \frac{13}{18} p_{(0,0)^i} + \frac{1}{2} \cdot \frac{13}{48} p_{(0,0)^i}
\]

\[
= \frac{13}{18} p_{(0,0)^i} + \frac{1}{2} \cdot \frac{13}{48} p_{(0,0)^i}
\]

\[
= \frac{211}{288} p_{(0,0)^i}.
\]
Adding all probabilities will give an upper bound on $p\{0,0\}^i$ as follows:

\[
1 = \sum_{x_1, x_3 \geq 0} (p\{x_1,x_3\}^i + p\{x_1,x_3\}^o) \\
> \sum_{x_3 \geq 0} (p\{0,x_3\}^i + p\{0,x_3\}^o + p\{1,x_3\}^i + p\{1,x_3\}^o + p\{2,x_3\}^o) \\
= p\{0,0\}^i + \sum_{x_3 = 1}^{\infty} p\{0,x_3\}^i + p\{0,1\}^o + \sum_{x_3 = 2}^{\infty} p\{0,x_3\}^o + p\{1,0\}^i + \sum_{x_3 = 1}^{\infty} p\{1,x_3\}^i + p\{1,0\}^o + \sum_{x_3 = 1}^{\infty} p\{2,x_3\}^o \\
> \left( 3 + 2 + \frac{3}{4} + \frac{2}{3} + \frac{13}{48} + \frac{211}{288} \right) p\{0,0\}^i \\
= \frac{2425}{288} p\{0,0\}^i.
\]

Now we can see that

\[ p\{0,0\}^i < \frac{288}{2425} \approx 0.1188. \]

When we substitute this into (C.13), we finally get:

\[
\theta_{3,1,\infty} < \frac{1}{14} \left( 5 + \frac{25}{4} \cdot \frac{288}{2425} \right) \\
= \frac{557}{1358} \approx 0.41016 \text{ packets/s},
\]

which shows that the throughput for an infinite number of packets is bounded from above by 0.41016 packets per time unit.
Appendix D

Calculations on truncated deterministic back-off

D.1 Simplifying the throughput function for truncated deterministic back-offs

In this section we will deduce (4.48) from (4.47), which are used in Section 4.6. We start with

\[ \mathbb{E} \{ H_2 + W_2 + T_2 \mid R \} = 1 + (1 - e^{-R}) (R + 1) \]

\[ + \sum_{m=1}^{\infty} e^{-R-(m-1)\eta} \left( \frac{1}{2} \right)^m \left( R + (m - 1)\eta + \frac{m + 2}{2} \right) \]

\[ + \sum_{m=1}^{\infty} e^{-R-(m-1)\eta} \left( \frac{1}{2} \right)^m (1 - e^{-\eta}) \left( R + (m - 1)\eta + \frac{m + 2}{2} \right) - \frac{\eta e^{-\eta}}{1 - e^{-\eta}}. \]

Next, we rewrite the sums by combining equal terms

\[ \mathbb{E} \{ H_2 + W_2 + T_2 \mid R \} = 1 + (1 - e^{-R}) (R + 1) \]

\[ + \sum_{m=1}^{\infty} e^{-R-(m-1)\eta} \left( \frac{1}{2} \right)^m \left( R + (m - 1)\eta + \frac{m + 2}{2} \right) (2 - e^{-\eta}) \]

\[ - \sum_{m=1}^{\infty} e^{-R-(m-1)\eta} \left( \frac{1}{2} \right)^m \eta e^{-\eta} \]

\[ = 1 + (1 - e^{-R}) (R + 1) \]

\[ + e^{-R} \sum_{m=1}^{\infty} e^{-m\eta} \left( \frac{1}{2} \right)^m \left( R + (m - 1)\eta + \frac{m + 2}{2} \right) (2e^{\eta} - 1) \]

\[ - e^{-R} \sum_{m=1}^{\infty} e^{-m\eta} \left( \frac{1}{2} \right)^m \eta. \]
Then we join both sums and split them again into one part with and one part without the term $m$.

$$
\mathbb{E}\{H_2 + W_2 + T_2 \mid R\} = 1 + (1 - e^{-R})(R + 1)
+ e^{-R} \sum_{m=1}^{\infty} \left(\frac{e^{-\eta}}{2}\right)^m
\cdot \left(2Re^\eta - R + 2(m - 1)\eta e^\eta - (m - 1)\eta + (m + 2)e^\eta - \frac{m + 2}{2} - \eta\right)
= 1 + (1 - e^{-R})(R + 1)
+ e^{-R} (2Re^\eta - R - 2\eta e^\eta + 2e^\eta - 1) \sum_{m=1}^{\infty} \left(\frac{e^{-\eta}}{2}\right)^m
+ e^{-R} \left(2\eta e^\eta - \eta + e^\eta - \frac{1}{2}\right) \sum_{m=1}^{\infty} m \left(\frac{e^{-\eta}}{2}\right)^m.
$$  

(D.1)

Now we can easily evaluate the sums, since $0 < e^{-\eta}/2 < 1$ for all $\eta > 0$, namely

$$
\sum_{m=1}^{\infty} \left(\frac{e^{-\eta}}{2}\right)^m = \frac{\frac{e^{-\eta}}{2}}{1 - \frac{e^{-\eta}}{2}} = \frac{1}{2e^\eta - 1}
$$

$$
\sum_{m=1}^{\infty} m \left(\frac{e^{-\eta}}{2}\right)^m = \frac{\frac{e^{-\eta}}{2}}{(1 - \frac{e^{-\eta}}{2})^2} = \frac{2e^\eta}{(2e^\eta - 1)^2}.
$$

Substituting these identities into (D.1) results in

$$
\mathbb{E}\{H_2 + W_2 + T_2 \mid R\} = 1 + (1 - e^{-R})(R + 1)
+ e^{-R} (2Re^\eta - R - 2\eta e^\eta + 2e^\eta - 1) \frac{1}{2e^\eta - 1}
+ e^{-R} \left(2\eta e^\eta - \eta + e^\eta - \frac{1}{2}\right) \frac{2e^\eta}{(2e^\eta - 1)^2}.
$$

And finally, simplifying this gives

$$
\mathbb{E}\{H_2 + W_2 + T_2 \mid R\} = 1 + (1 - e^{-R})(R + 1) + e^{-R}(R + 1) - e^{-R} \frac{2\eta e^\eta}{2e^\eta - 1} + e^{-R} \frac{(\eta + \frac{1}{2}) 2e^\eta}{2e^\eta - 1}
= 1 + R + 1 + e^{-R} \frac{e^\eta}{2e^\eta - 1}
= 2 + R + e^{-R} \frac{1}{2 - e^{-\eta}}.
$$

where the expression is identical to (4.48).
D.2 Expected value of $R_{i+1}$ for truncated deterministic back-offs

In this section we will show how (4.50) in Section 4.6 is calculated. We start by repeating (4.49):

$$R_{i+1} | R_i = \begin{cases} 
\max\{\eta - T_2, 0\} & \text{w.p. } 1 - e^{-R_i \frac{1-e^{-\eta}}{2}} \\
\max\{\eta - T_2 - T_3, 0\} & \text{w.p. } e^{-R_i \frac{1-e^{-\eta}}{2}}.
\end{cases} \tag{D.2}$$

We now need to find $E\{\max\{\eta - T_2, 0\}\}$ and $E\{\max\{\eta - T_2 - T_3, 0\} | T_3 < \eta\}$. The first one is relatively straightforward:

$$E\{\max\{\eta - T_2, 0\}\} = \eta - E\{\min\{T_2, \eta\}\}$$
$$= \eta - \int_0^\infty \min\{t, \eta\} e^{-t} \, dt$$
$$= \eta - \int_0^{\eta} te^{-t} \, dt - \eta e^{-\eta}$$
$$= \eta - (1 - (1 + \eta)e^{-\eta}) - \eta e^{-\eta}$$
$$= \eta - 1 + e^{-\eta}. \tag{D.3}$$

The other one is a bit harder to evaluate, since we need to find the following probability first for $t < \eta$

$$P[T_2 + T_3 < t | T_3 < \eta] = \int_0^t P[T_2 < t - x_3] \, dF_{X_3}(x_3)$$
$$= \int_0^t P[T_2 < t - x_3] \, d\left(\frac{1-e^{-x_3}}{1-e^{-\eta}}\right)$$
$$= \int_0^t \left(1 - e^{-(t-x_3)}\right) \frac{e^{-x_3}}{1-e^{-\eta}} \, dt$$
$$= \frac{1}{1-e^{-\eta}} \int_0^t (e^{-x_3} - e^{-t}) \, dt$$
$$= \frac{1}{1-e^{-\eta}} (1 - e^{-t} - te^{-t}).$$

We can now see that

$$E\{\min\{T_2 + T_3, \eta\} | T_3 < \eta\} = \int_0^\infty \min\{t, \eta\} \, dP[T_2 + T_3 < t | T_3 < \eta]$$
$$= \int_0^{\eta} t \, d\left(\frac{1-e^{-t} - te^{-t}}{1-e^{-\eta}}\right) + \eta \left(1 - \frac{1-e^{-\eta} - \eta e^{-\eta}}{1-e^{-\eta}}\right)$$
$$= \int_0^{\eta} \frac{t^2 e^{-t}}{1-e^{-\eta}} \, dt + \frac{\eta^2 e^{-\eta}}{1-e^{-\eta}}$$
$$= 2 - \frac{\eta(2+\eta)e^{-\eta}}{1-e^{-\eta}} + \frac{\eta^2 e^{-\eta}}{1-e^{-\eta}}$$
$$= 2 - \frac{2\eta e^{-\eta}}{1-e^{-\eta}}.$$
Thus
\[
\mathbb{E}\{\max\{\eta - T_2 - T_3, 0\} \mid T_3 < \eta\} = \eta - \mathbb{E}\{\min\{T_2 + T_3, \eta\} \mid T_3 < \eta\}
\]
\[
= \eta - 2 + \frac{2\eta e^{-\eta}}{1 - e^{-\eta}}
\]
\[
= \frac{\eta + e^{-\eta} - 1}{1 - e^{-\eta}} - 2.
\]
\[\text{(D.4)}\]
We will now calculate the total expectation multiplying the probabilities in (D.2) by the expectations in (D.3) and (D.4).
\[
\mathbb{E}\{R_{i+1} \mid R_i = R\} = \left(1 - e^{-R} \frac{1 - e^{-\eta}}{2 - e^{-\eta}}\right)(\eta - 1 + e^{-\eta}) + e^{-R} \frac{1 - e^{-\eta}}{2 - e^{-\eta}} \left(\frac{\eta + e^{-\eta} - 1}{1 - e^{-\eta}} - 2\right)
\]
\[
= e^{-R} \frac{1 - e^{-\eta}}{2 - e^{-\eta}} \left(\frac{\eta + e^{-\eta} - 1}{1 - e^{-\eta}} - 2 - \eta + 1 - e^{-\eta}\right) + \eta - 1 + e^{-\eta}
\]
\[
= \frac{e^{-R}}{2 - e^{-\eta}} \left(\eta + \eta e^{-\eta} - 1 + e^{-\eta} - \eta + \eta e^{-\eta} - e^{-\eta} + e^{-2\eta}\right) + \eta - 1 + e^{-\eta}
\]
\[
= e^{-R} e^{-2\eta} + 2\eta e^{-\eta} - 1 \frac{1}{2 - e^{-\eta}} + \eta - 1 + e^{-\eta}.
\]
This is indeed exactly the expression found in (4.50).

### D.3 Proof of convergence of \(\{a_i(\eta)\}_{i=0}^{\infty}\)

In this section we will prove Lemma 2. We have the sequence \(\{a_i(\eta)\}_{i=0}^{\infty}\) with \(a_{i+1}(\eta) = f(a_i(\eta), \eta)\), where
\[
f(R, \eta) = e^{-R} e^{-2\eta} + 2\eta e^{-\eta} - 1 \frac{1}{2 - e^{-\eta}} + \eta - 1 + e^{-\eta}.
\]
We now write
\[
b_1(\eta) = \frac{e^{-2\eta} + 2\eta e^{-\eta} - 1}{2 - e^{-\eta}},
\]
\[
b_2(\eta) = \eta - 1 + e^{-\eta}.
\]
To prove convergence, we first show that (i) \(b_1(\eta) < 0\), (ii) \(b_2(\eta) > 0\), (iii) \(f(0, \eta) > 0\), and (iv) \(\frac{df}{d\eta}(R, \eta) > 0\) for all \(\eta > 0\).

(i) Since \(2 - e^{-\eta} > 0\) for all \(\eta > 0\), we only have to look at the numerator of \(b_1(\eta)\) to determine the sign. We see that \(b_1(0) = 0\) and
\[
b_1'(\eta) = -2e^{-2\eta} + 2e^{-\eta} - 2\eta e^{-\eta} = -2e^{-\eta}(e^{-\eta} - 1 + \eta) < 0,
\]
for \(\eta > 0\). This last inequality follows from the well-known identity \(1 + x < e^x\) for all \(x \neq 0\). We can conclude that \(b_1(\eta) < 0\) for all \(\eta > 0\).

(ii) Because of the same identity, we see that \(b_2(\eta) = \eta - 1 + e^{-\eta} > 0\) for all \(\eta > 0\).

(iii) First we evaluate \(f(0, \eta)\).
\[
f(0, \eta) = \frac{e^{-2\eta} + 2\eta e^{-\eta} - 1}{2 - e^{-\eta}} + \eta - 1 + e^{-\eta}
\]
\[
= \frac{e^{-2\eta} + 2\eta e^{-\eta} - 1 + (\eta - 1 + e^{-\eta})(2 - e^{-\eta})}{2 - e^{-\eta}}
\]
\[
= \frac{\eta e^{-\eta} + 3e^{-\eta} + 2n - 3}{2 - e^{-\eta}}.
\]
D.3. Proof of convergence of $\{a_i(\eta)\}_{i=0}^{\infty}$

Again, we only have to convince ourselves that the numerator is greater than or equal to 0, since it is clear that the denominator is always positive. We define

$$b_3(\eta) = \eta e^{-\eta} + 3e^{-\eta} + 2\eta - 3,$$

and see that $b_3(0) = 0$. To prove that $f(0, \eta) > 0$ for $\eta > 0$, it suffices to prove $b_3(\eta) > 0$, and since $b_3(0) = 0$, we look at the derivative.

$$b'_3(\eta) = e^{-\eta} - \eta e^{-\eta} - 3e^{-\eta} + 2 = -2e^{-\eta} - \eta e^{-\eta} + 2,$$

which shows that $b'_3(0) = 0$, so we proceed to the second derivative:

$$b''_3(\eta) = 2e^{-\eta} - e^{-\eta} + \eta e^{-\eta} = e^{-\eta} + \eta e^{-\eta}.$$

The second derivative clearly shows that $b''_3(\eta) > 0$ for all $\eta \geq 0$, and thus $b_3(\eta) > 0$ for all $\eta > 0$.

(iv) We note that $\frac{df}{dR}(R, \eta) = -e^{-R}b_1(\eta)$ and because of (i) and $e^{-R} > 0$, we conclude that $\frac{df}{dR}(R, \eta) > 0$ for all $\eta > 0$.

Because of (i) and (ii) we see that the function $f(R, \eta)$ is concave in $R$ for every value of $\eta > 0$. As explained in the text below (4.53), the equation $a(\eta) = f(a(\eta), \eta)$ has only one positive real solution $a(\eta)$ for every $\eta > 0$. Because of (iii) and (iv), which show that $y = f(R, \eta)$ is strictly increasing in $R$ and starts out at $\eta = 0$ above the line $y = R$, the derivative $\frac{df}{dR}(R, \eta)$ at the point $R = a(\eta)$ is within 0 and 1, making this a stable point of convergence. In Figure D.1 a visualization of this proof is shown. Since the other solution to $a(\eta) = f(a(\eta), \eta)$ yields $a(\eta) < 0$ and for the same reasoning has an instable point of convergence, we can conclude that the sequence $\{a_i(\eta)\}_{i=0}^{\infty}$ converges to $a(\eta)$ for all $\eta > 0$.

We now want to show that when we start in the interval $0 < a_0(\eta) < \eta$, we will always stay there. Suppose $a_i(\eta) = x > 0$, then we have because of (i) and (ii):

$$a_{i+1}(\eta) = f(x, \eta) = e^{-x}b_1(\eta) + b_2(\eta) > b_2(\eta) > 0,$$
where we also used that \( e^{-x} > 0 \). Next, suppose \( a_i(\eta) = x < \eta \), then

\[
a_{i+1}(\eta) = f(x, \eta) = \frac{e^{-x}e^{-2\eta} + 2\eta e^{-\eta} - 1}{2 - e^{-\eta}} + \eta - 1 + e^{-\eta} < \frac{e^{-a}e^{-2\eta} + 2\eta e^{-\eta} - 1}{2 - e^{-\eta}} + \eta - 1 + e^{-\eta} = \eta + \frac{e^{-3\eta} + 2\eta e^{-2\eta} - e^{-\eta} + (e^{-\eta} - 1)(2 - e^{-\eta})}{2 - e^{-\eta}} = \eta + \frac{e^{-3\eta} + (2\eta - 1)e^{-2\eta} + 2e^{-\eta} - 2}{2 - e^{-\eta}}.
\]

Since again the denominator of the last term is always positive, we must show that the numerator is negative for \( \eta > 0 \). We see that with \( b_4(\eta) = e^{-3\eta} + (2\eta - 1)e^{-2\eta} + 2e^{-\eta} - 2 \), we have \( b_4(0) = 0 \), so we will look at the derivative:

\[
b'_4(\eta) = -2e^{-\eta} - (4\eta - 4)e^{-2\eta} - 3e^{-3\eta}.
\]

We immediately see for \( \eta \geq 1 \) that this expression is negative. We also see \( b'_4(0) = -1 \) and we note that this expression is continuous in \( \eta \). If the expression is positive for some \( 0 < \eta < 1 \) the equation

\[
-2e^{-\eta} - (4\eta - 4)e^{-2\eta} - 3e^{-3\eta} = 0 \quad (D.5)
\]

must have one or more solutions within \( 0 < \eta < 1 \). Substituting \( p = e^{-\eta} \), we simplify this to

\[
2 + (4\eta - 4)p + 3p^2 = 0,
\]

and calculate the discriminant of this quadratic equation

\[
D = (4\eta - 4)^2 - 4 \cdot 2 \cdot 3 = 16\eta^2 - 32\eta - 8.
\]

Setting \( D = 0 \) gives solutions \( \eta_{1,2} = 1 \pm \frac{1}{2}\sqrt{6} \), so that \( \eta_1 < 0 \) and \( \eta_2 > 1 \). This means that \( D < 0 \) for the whole interval \( 0 < \eta < 1 \) and therefore (D.5) has no solution.

We conclude that from \( 0 < a_i(\eta) < \eta \) we know that also \( 0 < a_{i+1}(\eta) < \eta \) and since the sequence converges, we have \( 0 < a(\eta) < \eta \). This concludes the proof.
Bibliography


