Performance Analysis of a MAC Protocol for Ethernet Passive Optical Networks

by

Lieven VERSLEGERS

Promoters: prof. dr. ir. M. PICKAVET and prof. dr. ir. M. GAGNAIRE (ENST Paris)
Thesis Counsellor: Ir. B. LANNOO

Thesis submitted for the degree of electrical engineer

Academic year 2005-2006
Word of thanks

I would like to thank everyone who contributed to this thesis.
In the first place Un Grand Merci to my promoter in Paris, Professor Gagnaire, for providing me with a thesis subject and guiding me throughout the year. Furthermore, thanks to my promoter and my thesis counsellor in Ghent, Professor Pickavet and Bart Lannoo, for their follow-up from the home front. Thanks also to Kurt Haelvoet for giving me the opportunity of an internship at Telenet Mechelen and Elie for the nice companionship there. Inevitably, I would like to express gratitude to my parents for their moral support. I can not forget about all my friends, in Paris as in Belgium, for taking my mind off analytical modelling from time to time, and most in particular Sam and Dan who corrected my English language mistakes.
Permission to loan

The undersigned, acknowledging that he is the author of this thesis, gives Ghent University and ENST Paris permission to publicly archive this thesis and copy parts of it for personal use.
For every other use, the copyright limitations apply, especially concerning the obligation to mention explicitly the source, when quoting results from this thesis.

Date                      Signature
Performance Analysis of a MAC Protocol for Ethernet Passive Optical Networks

by

Lieven VERSLEGERS

Thesis submitted for the degree of electrical engineer

Academic year 2005-2006

Promoters: prof. dr. ir. M. PICKAVET and prof. dr. ir. M. GAGNAIRE (ENST Paris)
Thesis Counsellor: Ir. B. LANNOO

Ghent University
Faculty of Engineering
Department of Information Technology
Chairman: prof. dr. Ir. P. LAGASSE

Summary

With ever-increasing bandwidth demands, the access has become the network’s bottleneck. A possible and promising solution for next-generation access networks is to deploy optical fibre closer to the end-user. This thesis focuses on the Ethernet Passive Optical Network (EPON), its Multipoint Control Protocol (MPCP), the dynamic bandwidth allocation algorithm IPACT (Interleaved Polling with Adaptive Cycle Time) and its service disciplines. Simulations were performed, from which interesting observations were made. Attention was given especially to the mechanisms of cycle times. The observations provided new insights and form the basis of the analysis. They are also a useful step up to the analytical model.

This analytical model constitutes the core of this thesis. The system’s stability is discussed in detail, followed by an analysis of the cycle time. Ultimately packet delay is derived. Hereby it is assumed that traffic load is symmetric, that packet arrivals are Poisson distributed and that packets have fixed size. Some extensions and limitations of the model are treated, including asymmetric traffic load, packet size distribution and self-similar traffic.

Differentiated services are discussed as well. An existing analytical model (which formed the actual starting point of this thesis) is discussed in detail and some of the shortcomings are shown. Furthermore, the future of the EPON is discussed, with attention to different upgrade scenarios.

Index terms: EPON, MPCP, IPACT, packet delay
Performance Analysis of a MAC Protocol for Ethernet Passive Optical Networks

Lieven Verslegers

Supervisors: prof. dr. ir. M. Pickavet, prof. dr. ir. M. Gagnaire and ir. B. Lannoo

Abstract - This article describes the research that has been performed in the field of the Multipoint Control Protocol (MPCP) for the Ethernet Passive Optical Network (EPON) and more in particular the dynamic bandwidth allocation algorithm IPACT (Interleaved Polling with Adaptive Cycle Time). The main focus has been on modeling packet delay analytically.

Keywords - EPON, MPCP, IPACT, packet delay

I. INTRODUCTION

With ever-increasing bandwidth demands, the access has become the network’s bottleneck. Whereas optical fibre is already omnipresent in WANs and MANs, most broadband access networks still rely on coaxial cable and twisted pair. The use of optical fibre in the access seems the logical next step. Optical access networks can offer bandwidths that are many times larger than older technologies and can therefore support a variety of services at the same time: video on demand, two-way videoconferencing, video email, etc.

II. TERMINOLOGY

A. Passive Optical Networks (PONs)

A PON is made up of an OLT (Optical Line Terminal), connected to the ONUs (Optical Network Units) by optical fibre that splits up at a passive optical splitter. The OLT is located at the local exchange (central office) and connects the access to the metro backbone. The ONU can reside at the curb (Fiber To The Curb or FTTC) or at the end-user location (Fiber To The Building or Home, FTTB/FTTH). Due to the use of a passive component, a PON is MP2P (Multi-Point-To-Point) in the upstream direction and P2MP (Point-To-Multi-Point) downstream.

There are several standards for the PON: BPON (Broadband PON), GPON (Gigabit PON) and EPON (Ethernet PON). This last standard will be discussed in this paper.

B. EPON, MPCP and IPACT

As the name suggests, EPONs use Ethernet frames to encapsulate data. The EPON’s standardized bit rate is 1 Gb/s.

Since in the upstream direction multiple ONUs share a common channel, an arbitration mechanism is necessary for upstream transmission. Such a mechanism is delivered by the Multipoint Control Protocol (MPCP). It supports timeslot allocation to the ONUs by the OLT. MPCP is not concerned with a specific bandwidth allocation scheme; it provides a framework, intended to facilitate the implementation of bandwidth allocation algorithms, by providing signalling infrastructure for coordinating upstream data transmission. A REPORT control message is sent to the OLT by an ONU to report its queue status. The OLT then uses this knowledge of queue status to assign a transmission window.

Interleaved Polling with Adaptive Cycle Time (IPACT) is a possible bandwidth allocation scheme, in which the OLT polls the ONUs individually and issues grants to them in a round-robin fashion [1]. At the end of a transmission window, an ONU reports its queue size(s). The OLT uses this information to determine the next granted transmission window. The knowledge of the distance between OLT and ONUs allows the OLT to schedule transmission windows so that packets from different ONUs do not overlap in time. In fact, transmission windows are only separated by a guard time. Several variants exist, the following of which were studied in detail:

- Fixed service: the OLT always grants the maximum window to all ONUs.
- Gated service: the OLT grants the amount of bytes the ONU requested.
- Limited service: the OLT grants the requested window, so long as it is not more than the maximum transmission window.

III. ANALYTICAL MODELING

In the analysis, packets are assumed to follow a Poisson arrival process with bit rate \( \lambda \) [Mbit/s] and to have a fixed size (B bits). Traffic is assumed symmetric, i.e. traffic charge is the same for all ONUs (which are at the same distance from the OLT). No formulas will be shown here: the focus is on the mode of thought.

A. Fixed service

Since for the fixed service discipline the cycle time is constant, the system can be considered at discrete moments that are \( T_{cycle} \) apart and located immediately after an ONU has sent packets in its granted transmission window. If \( Q(n) \) is the queue size [packets] of an ONU at \( t = n.T_{cycle} \), then \( Q(n) \) is a discrete homogenous Markov chain, which means that transition probabilities can be defined. Poisson properties allow formulas for these probabilities to be obtained. In order to derive the stationary distribution of queue sizes, a linear system of equations must be solved. The queue size distribution then allows the average queue size at the discrete moments in time to be calculated. Additional terms must be taken into account to obtain the average queue size in
continuous time. The average packet delay follows from Little’s law. Figure 1 shows that results from simulation and analysis practically coincide for an EPON with 16 ONUs at 20 km from the OLT, a maximum transmission window of 10 1500 byte IP packets and a guard time of 1.5 and 5 µs.

**B. Gated service**

For the gated service discipline, analysis becomes even more complex, due to the varying cycle times. Figure 2 gives an idea of this strong variation for a (high) 57.5 Mbit/s traffic load per ONU. The figure also clearly shows how there exist periods of shorter and longer cycle times. This leads to the conclusion that successive cycle times influence each other.

The analysis for gated service takes the cycle time as its starting point. A first approximation, in which the cycle time is assumed to tend to a length in which as many packets arrive as can be sent, proves to be only partially correct.

A more correct analysis distinguishes between low and high traffic load. For low traffic load the cycle time can not become lower than a minimum value, which is determined by the distance between OLT and ONUs. This is because a grant message needs the information contained in the previous request. At low traffic loads, it turns out that the ONU’s traffic mostly determines its cycle time. What complicates analysis even more, is the fact that ONUs can cluster, which causes ONUs to influence cycle times of ONUs that are polled successively.

For high traffic load, the cycle time is most often determined by the aggregate traffic load. In this case statistical properties of Poisson traffic allow an approximate distribution of cycle times to be derived. This distribution can then be used to derive average packet delay. For this, one must consider that the probability that a packet falls in a cycle of a certain length is also proportional to this length.

Figure 3 shows how the analysis suits the simulated results, again for an EPON with 16 ONUs at 20 km and a guard time of 1.5 µs.

**C. Limited service**

Limited service shows some properties similar to fixed service and some similar to gated service: since the transmission window can not become bigger than a certain maximum value, the possibility exists that a packet can not be sent with its first requested window; since the granted window is based on the requested window, the cycle time is variable. All processes together cause the system to become too complex for a numerical analysis similar to the fixed or gated service.

**D. Extensions and limitations**

Possible extensions of the analytical model have been investigated. For asymmetric traffic load the fixed service analysis still applies and also the gated service analysis still has its value. An analysis that includes packet size distribution is difficult for fixed service, due to an unused remainder of the transmission window, whereas for gated service, this problem does not exist. Other types of traffic, such as self-similar traffic, are too complex for analytical methods to be used.

Extending the EPON to support differentiated services was investigated. A Weighted Round Robin (suggested in [2]) proved not to be a good solution, compared to priority scheduling.

**IV. CONCLUSIONS**

The EPON with MPCP-IPACT proves to be a very complex system to analyze, yet it still allows packet delay to be derived in the case of fixed and gated service and Poisson traffic with constant size packets. This analysis also turns out to be valuable in more general cases.

**ACKNOWLEDGEMENTS**

The author would like to acknowledge the suggestions of prof. dr. ir. M. Gagnaire, prof. dr. ir. M. Pickavet and ir. B. Lannoo.

**REFERENCES**


# Table of contents

1. Introduction .................................................................................................................. 1
   1.1 Setting ...................................................................................................................... 1
      1.1.1 The future of access networks ................................................................. 1
      1.1.2 Passive Optical Networks ................................................................. 2
   1.2 Structure of thesis ................................................................................................. 4

2. EPONs and MPCP – IPACT .......................................................................................... 5
   2.1 Ethernet PONs ....................................................................................................... 5
      2.1.1 EPON Optics .............................................................................................. 5
      2.1.2 Point-to-Point Emulation ......................................................................... 5
      2.1.3 Message format ......................................................................................... 6
      2.1.4 Multipoint Control Protocol ..................................................................... 7
   2.2 IPACT ...................................................................................................................... 8
      2.2.1 The algorithm .............................................................................................. 8
      2.2.2 Service disciplines ................................................................................. 10

3. Simulations and observations ...................................................................................... 12
   3.1 Introduction ......................................................................................................... 12
   3.2 Fixed service ....................................................................................................... 13
   3.3 Gated service ..................................................................................................... 13
   3.4 Limited service ................................................................................................. 21
   3.5 General conclusion ........................................................................................... 23

4. Analytical model ......................................................................................................... 24
   4.1 Introduction ......................................................................................................... 24
   4.2 Stability analysis ............................................................................................... 25
      4.2.1 Fixed service ............................................................................................ 26
      4.2.2 Gated service .......................................................................................... 26
      4.2.3 Limited service ....................................................................................... 27
   4.3 Cycle time analysis ............................................................................................ 28
      4.3.1 Fixed service ............................................................................................ 28
      4.3.2 Gated service .......................................................................................... 29
      4.3.3 Limited service ....................................................................................... 36
   4.4 Access delay analysis ......................................................................................... 38
      4.4.1 Fixed service ............................................................................................ 39
      4.4.2 Gated service .......................................................................................... 45
      4.4.2 Limited service ....................................................................................... 48
   4.5 Conclusions ......................................................................................................... 50

5. Extensions and limitations of the model ................................................................... 51
   5.1 Introduction ......................................................................................................... 51
   5.2 Asymmetric traffic load ...................................................................................... 51
      5.2.1 Fixed service ............................................................................................ 51
      5.2.2 Gated service .......................................................................................... 52
   5.3 Packet size distribution ...................................................................................... 54
      5.3.1 Fixed service ............................................................................................ 54
      5.3.2 Gated service .......................................................................................... 56
      5.3.3 Limited service ....................................................................................... 59
   5.4 Self-similar traffic ............................................................................................... 59
      5.4.1 Fixed service ............................................................................................ 59
      5.4.2 Gated service .......................................................................................... 60
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACT</td>
<td>Adaptive Cycle Time</td>
</tr>
<tr>
<td>APD</td>
<td>Avalanche Photodiode</td>
</tr>
<tr>
<td>APON</td>
<td>ATM Passive Optical Network</td>
</tr>
<tr>
<td>AF</td>
<td>Assured Forwarding</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>AWG</td>
<td>Arrayed Waveguide Grating</td>
</tr>
<tr>
<td>BE</td>
<td>Best Effort</td>
</tr>
<tr>
<td>BPON</td>
<td>Broadband Passive Optical Network</td>
</tr>
<tr>
<td>CO</td>
<td>Central Office</td>
</tr>
<tr>
<td>CPON</td>
<td>Composite Passive Optical Network</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>DFB</td>
<td>Distributed Feedback</td>
</tr>
<tr>
<td>DSL</td>
<td>Digital Subscriber Line</td>
</tr>
<tr>
<td>EF</td>
<td>Expedited Forwarding</td>
</tr>
<tr>
<td>EPON</td>
<td>Ethernet Passive Optical Network</td>
</tr>
<tr>
<td>FCFS</td>
<td>First Come First Served</td>
</tr>
<tr>
<td>FCS</td>
<td>Frame Check Sequence</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out</td>
</tr>
<tr>
<td>FTTB</td>
<td>Fibre To The Building</td>
</tr>
<tr>
<td>FTTC</td>
<td>Fibre To The Curb</td>
</tr>
<tr>
<td>FTTH</td>
<td>Fibre To The Home</td>
</tr>
<tr>
<td>GFP</td>
<td>General Framing Protocol</td>
</tr>
<tr>
<td>HDTV</td>
<td>High-Definition Television</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IFG</td>
<td>Inter-Frame Gap</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPACT</td>
<td>Interleaved Polling with Adaptive Cycle Time</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunication Union</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LARNET</td>
<td>Local Access Router Network</td>
</tr>
<tr>
<td>LED</td>
<td>Light Emitting Diode</td>
</tr>
<tr>
<td>LLID</td>
<td>Logical Link Identifier</td>
</tr>
<tr>
<td>MAN</td>
<td>Metropolitan Area Network</td>
</tr>
<tr>
<td>MPCP</td>
<td>Multipoint Control Protocol</td>
</tr>
</tbody>
</table>
MP2P  Multi-Point-To-Point
NAM   Network Animator
OLT   Optical Line Terminal
ONU   Optical Network Unit
PIN   P-type, Intrinsic, N-type
PON   Passive Optical Network
P2MP  Point-To-Multi-Point
P2P   Point-To-Point
RF    Radio Frequency
RITENET Remote Interrogation of Terminal Network
RTT   Round-Trip Time
SPD   Start of Packet Delimiter
SUCCESS-DWA Stanford University Access Dynamic Wavelength Assignment
TDM   Time Division Multiplexing
VOD   Video On Demand
WAN   Wide Area Network
WDM   Wavelength Division Multiplexing
WRR   Weighted Round Robin
Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Packet size (network layer)</td>
<td>[bits]</td>
</tr>
<tr>
<td>B_{req}</td>
<td>REPORT/request message size</td>
<td>[bits]</td>
</tr>
<tr>
<td>c</td>
<td>Speed of light</td>
<td>[m/s]</td>
</tr>
<tr>
<td>C</td>
<td>Clustering factor</td>
<td>[-]</td>
</tr>
<tr>
<td>\phi_x</td>
<td>Class weight (x = EF, AF or BE)</td>
<td>[-]</td>
</tr>
<tr>
<td>\ell</td>
<td>Distance between OLT and ONU</td>
<td>[m]</td>
</tr>
<tr>
<td>\lambda</td>
<td>ONU arrival rate</td>
<td>[bit/s]</td>
</tr>
<tr>
<td>\lambda_x</td>
<td>Class ONU arrival rate (x = EF, AF or BE)</td>
<td>[bit/s]</td>
</tr>
<tr>
<td>\Lambda</td>
<td>Aggregate traffic load</td>
<td>[bit/s]</td>
</tr>
<tr>
<td>\Lambda_{MIN}</td>
<td>Minimum guaranteed bandwidth</td>
<td>[bit/s]</td>
</tr>
<tr>
<td>\Lambda_{MAX}</td>
<td>Maximum obtainable bandwidth</td>
<td>[bit/s]</td>
</tr>
<tr>
<td>n</td>
<td>Refractive index</td>
<td>[-]</td>
</tr>
<tr>
<td>N</td>
<td>Number of ONUs</td>
<td>[-]</td>
</tr>
<tr>
<td>N_T</td>
<td>Number of packet arrivals in a time T</td>
<td>[packets]</td>
</tr>
<tr>
<td>P</td>
<td>Number of packets sent in a window</td>
<td>[packets]</td>
</tr>
<tr>
<td>P_i</td>
<td>Granted transmission window for ONU i</td>
<td>[packets]</td>
</tr>
<tr>
<td>P_{max}</td>
<td>Maximum transmission window</td>
<td>[packets]</td>
</tr>
<tr>
<td>P_x</td>
<td>Class priority (x = EF, AF or BE)</td>
<td>[-]</td>
</tr>
<tr>
<td>Q</td>
<td>Queue size</td>
<td>[packets]</td>
</tr>
<tr>
<td>R</td>
<td>Unused remainder</td>
<td>[bytes]</td>
</tr>
<tr>
<td>R_U</td>
<td>Upstream bandwidth on the EPON</td>
<td>[bit/s]</td>
</tr>
<tr>
<td>\rho</td>
<td>Traffic load</td>
<td>[-]</td>
</tr>
<tr>
<td>T_{cycle}</td>
<td>Cycle time</td>
<td>[s]</td>
</tr>
<tr>
<td>T_{cycle_{min}}</td>
<td>Minimum cycle time</td>
<td>[s]</td>
</tr>
<tr>
<td>T_{ fibre}</td>
<td>Two-way delay on the EPON</td>
<td>[s]</td>
</tr>
<tr>
<td>T_{guard}</td>
<td>Guard time</td>
<td>[s]</td>
</tr>
<tr>
<td>T_{MAX}</td>
<td>Maximum cycle time</td>
<td>[s]</td>
</tr>
<tr>
<td>T_{proc}</td>
<td>Processing time</td>
<td>[s]</td>
</tr>
<tr>
<td>V_i</td>
<td>Requested window size</td>
<td>[bits]</td>
</tr>
<tr>
<td>W</td>
<td>Packet delay</td>
<td>[s]</td>
</tr>
<tr>
<td>W_{MAX}</td>
<td>Maximum window size</td>
<td>[bits]</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Setting

1.1.1 The future of access networks

Optical fibres are now omnipresent in Wide Area Networks (WANs) and Metropolitan Area Networks (MANs), providing a backbone with enormous bandwidth capacity. Most broadband access networks, however, still rely on coaxial cable or twisted pair (Digital Subscriber Line or DSL) and consequently remain the bottleneck. Using optical fibre in the access seems to be the logical next step. Optical access networks can offer bandwidths that are much higher than the old technologies and can therefore support an enormous variety of services simultaneously: video on demand, two-way videoconferencing, interactive gaming and IP telephony, to name but a few.

One can distinguish between several implementations, differing in the extent to which the optical fibre is deployed: with Fibre To The Home (FTTH) or Fibre To The Building (FTTB), the fibre reaches the user’s house or building; Fibre To The Curb (FTTC) brings the fibre to a service node near the user. Several other FTTx acronyms exist to designate often similar implementations. FTTH and FTTB require enormous investments, whereas FTTC seems to be a more economical solution. Combinations of optical access networks with traditional copper or cable networks or wireless networks make up intermediate solutions.

Whole books can be and have been written about the many different access technologies and possible implementations (for instance [1]). There is no universal solution for the access, rather several more or less suitable options depending on the situation. However, the aim of this thesis is not to provide such an overview.
1.1.2 Passive Optical Networks

Passive Optical Networks or PONs, as the name suggests, are optical networks that do not contain active elements between source and destination, only passive optical components, such as optical fibre, couplers, splices and splitters. A PON can be deployed in several topologies: for instance tree, ring or bus (figure 1.1). The (most common) tree topology will be the one discussed in the analysis.

![PON topologies](image)

As can be seen in the figure, a PON is made up of an OLT (Optical Line Terminal), connected to the ONU5 (Optical Networks Units) by optical fibre that splits up at a passive optical splitter. The OLT is located at the local exchange and connects the access to the metro backbone (figure 1.2). The ONU can reside at the curb (FTTC implementation) or at the end-user location (FTTH/FTTB). Due to the use of passive components, a PON is MP2P (Multi-Point-To-Point) in the upstream direction and P2MP (Point-To-Multi-Point) in the downstream direction.

The advantages of PON are often highlighted in PON literature (e.g. in [2] and [3]):

- PONs provide high bandwidths due to deep fibre penetration.
- PONs allow for longer distances between the central office and customer premises (20 km compared to 5.5 km for DSL).
• PONs minimize the fibre deployment in both the local exchange office and the local loop.
• PONs eliminate the necessity to install active multiplexers at splitting locations, relieving network operators of the task of maintaining active curb-side units and providing power to them.
• Operating in the downstream direction as a broadcast network, PONs allow for video broadcasting.
• Being optically transparent end to end, PONs allows easy upgrades to higher bit rates or additional wavelengths (these upgrades will be briefly discussed in chapter 7).

Currently, there are three important PON standards [3]: Ethernet PON (EPON), Broadband PON (BPON, based on the ATM PON or APON standard) and Gigabit PON (GPON). All standards use two wavelengths, one for downstream (at 1490 nm) and one for upstream data (at 1310 nm). The wavelengths are shared in time among the users, which means that all standardized PONs are Time Division Multiplexed PONs (TDM-PONs). A third wavelength (1550 nm) can optionally be used for downstream analogue video broadcasting (RF\textsuperscript{1}). Table 1.1 briefly compares some of the key characteristics of the different standards. More detailed descriptions can be found throughout the literature, for example in [1]. The focus of this thesis is on the EPON.

\textsuperscript{1} Radio Frequency
<table>
<thead>
<tr>
<th></th>
<th>EPON</th>
<th>BPON</th>
<th>GPON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>IEEE 802.3ah</td>
<td>ITU G.983</td>
<td>ITU G.984</td>
</tr>
<tr>
<td>Framing</td>
<td>Ethernet</td>
<td>ATM(^2)</td>
<td>GFP(^3)/ATM</td>
</tr>
<tr>
<td>Maximum bandwidth</td>
<td>1 Gb/s</td>
<td>622 Mb/s</td>
<td>2,488 Gb/s</td>
</tr>
<tr>
<td>ONUs/PON</td>
<td>16</td>
<td>32</td>
<td>64</td>
</tr>
<tr>
<td>Average bandwidth per ONU</td>
<td>60 Mb/s</td>
<td>20 Mb/s</td>
<td>40 Mb/s</td>
</tr>
<tr>
<td>Video</td>
<td>RF/IP</td>
<td>RF</td>
<td>RF/IP</td>
</tr>
</tbody>
</table>

Table 1.1: Characteristics of PON standards [4]

It should be noted that not all optical access networks are PONs. There exist solutions with active components as well. Using a switch instead of a splitter causes the architecture to become P2P (Point-To-Point). [5] considers the advantages that active networks have over PONs.

### 1.2 Structure of thesis

This first chapter briefly described the setting of this thesis. In a second chapter more details will be given about the Ethernet PON, its Multipoint Control Protocol (MPCP) and the Interleaved Polling with Adaptive Cycle Time (IPACT) bandwidth allocation scheme. The third chapter deals with observations, made from simulations, leading naturally to the analytical model, described in the elaborate fourth chapter. In a fifth chapter extensions and limitations of the model will be discussed. The sixth chapter introduces differentiated services and discusses another analytical model (from [3]). In chapter seven the possible upgrades for EPON are investigated. Chapter eight concludes.

Appendix A treats Poisson and self-similar traffic properties. Appendix B gives the necessary information about simulations and calculations.

\(^2\) Asynchronous Transfer Mode  
\(^3\) Generic Framing Protocol
2. EPONs and MPCP – IPACT

2.1 Ethernet PONs

Amongst the different optical access networks, EPON is often quoted as the most cost-effective solution. This lower cost results from the fact that passive fibre architectures are cheaper than networks with active elements. Besides this, Ethernet equipment is low-cost (compared to ATM equipment) which means that, amongst the different PON standards, EPON seems the most inexpensive solution for next-generation broadband access networks ([6]). However, when deploying new access networks, existing infrastructure can cause other solutions to be more suitable, i.e. more cost-effective.

In addition to cost, another argument is to the EPON’s credit: EPONs carry their data encapsulated in Ethernet frames. Ethernet encapsulation is more suitable to carry IP packets than ATM encapsulation, which requires the segmentation of variable-size IP packets into fixed-size ATM cells (causing extra overhead). Also, most of today’s data traffic originates and terminates in Ethernet LANs. Because of this, EPONs ease interoperability.

2.1.1 EPON Optics

As with Gigabit Ethernet, EPON has a nominal bit rate of 1.25 Gb/s (physical layer). Due to 8B/10B encoding, it is effectively 1 Gb/s (data layer). Optical transceivers (historically the highest cost component) make use of low-cost Distributed Feedback (DFB), Fabry-Perot lasers and APD and PIN detectors.

2.1.2 Point-to-Point Emulation

The Ethernet standard has been defined for both shared media and full-duplex P2P links. EPON is in some respects a combination of both (figure 1.2): in the downstream direction, the EPON behaves as a shared medium (frames transmitted by the OLT reach all ONUs). In the upstream direction, due to the directional properties of the passive coupler, frames sent by an ONU only reach the OLT, not the other ONUs. This is comparable to P2P (or MP2P, if one considers all ONUs together). However, there is a
difference with real P2P links: collisions can occur between traffic originating from different ONUs. Therefore, a protocol is required to manage the access to the network upstream.

Figure 1.2: Upstream/downstream transmission on EPON [8]

A point-to-point emulation (P2PE) mechanism causes the EPON to behave as a collection of P2P links downstream. For this purpose the Logical Link Identifier (LLID) is used. It is a 2-byte tag in the preamble. One bit designates the mode (point-to-point or broadcast), the remaining 15 bits identify the ONU. Each ONU adds its LLID to the preamble of all the frames it sends upstream. Similarly, the OLT will add the LLID of the destination ONU. All ONUs will filter frames, based on the LLID. Only the frames that belong to the ONU will be retained, the other ones will be rejected.

2.1.3 Message format

Figure 1.3 shows the fields of the EPON frame [9]. Most of them are identical to those of the 1000BASE-X frame: the 14 byte Ethernet header (containing the destination and source address, length and type), the payload and the Frame Check Sequence (FCS). The difference lies in the first part of the frame, the preamble. Traditionally, the preamble was used for clock synchronization and to inform a device that a frame was on the way. For EPONs the preamble contains a significant amount of information: the third byte makes up the Start of Packet Delimiter (SPD) which does what its name suggests, the sixth and seventh byte are the Logical Link ID, containing the LLID and the mode bit associated with either an
ONU or the OLT. The CRC8 field contains a value that is computed as a function of the preamble from the first bit of the SPD through the last bit of the LLID.

![Figure 1.3: EPON frame structure [9]](image)

Between two adjacent frames, a 12 byte Inter-Frame Gap (IFG) is left. This causes the total overhead to be equal to 38 bytes.

### 2.1.4 Multipoint Control Protocol

The Multipoint Control Protocol, developed by the IEEE 802.3ah task force, is the arbitration mechanism, supporting timeslot allocation to the ONUs by the OLT. MPCP is not concerned with a specific bandwidth allocation scheme; it provides a framework, intended to facilitate the implementation of bandwidth allocation algorithms for the EPON, by providing signalling infrastructure (64 byte Ethernet control messages: GATE and REPORT) for coordinating upstream data transmission. Besides that, MPCP provides auto-discovery, registration and ranging operations for newly added ONUs (REGISTER_REQUEST, REGISTER and REGISTER_ACK), but this last functionality will not be treated here, because it is of no importance for the analysis in the following chapters.

The bandwidth allocation approach is TDM (Time Division Multiplexing): every ONU will be allowed to send its data in a specific timeslot, according to the adopted algorithm. The REPORT message, sent upstream (from ONU to OLT), informs the OLT about the state of the queues (containing packets ready for upstream transmission) at the ONU and can report up to 13 queue occupancies for one ONU (queues can be prioritized). Upon receiving a REPORT message, the OLT
relies on the bandwidth allocation algorithm for determining the upstream transmission schedule. After the execution of the algorithm, the GATE message, sent downstream (from OLT to ONU) is used to issue transmission grants. A grant contains the transmission start time and the transmission length.

In order to accomplish the scheduling of ONU transmission slots, a mechanism to synchronize distributed events to a central master counter (the OLT) is required. For this purpose, the control messages contain time-stamps. The time-stamps, generated by the OLT, are used as a global time reference in the network. ONUs synchronize to the OLT by resetting their local counter according to the time-stamped GATE messages. Thanks to this mechanism of synchronization, the OLT can order ONUs to send their data so that transmission windows will not overlap.

2.2 IPACT

It was mentioned in the previous paragraph that MPCP does not specify a bandwidth allocation algorithm, it only provides the framework. Several bandwidth allocation algorithms exist. They are not part of the EPON standard. An overview can be found in [6]. Interleaved Polling with Adaptive Cycle Time (IPACT), proposed in [10], is the scheme that is treated further on.

2.2.1 The algorithm

The best way of describing the algorithm is by means of an example (based on [10]), in which, for simplicity, only three ONU will be considered:

1. Assume that at some time $t_0$ the OLT knows exactly how many bytes are waiting in each ONU’s buffer and the round-trip time (RTT) to each ONU. The OLT keeps this data in a polling table, shown in figure 1.4. At time $t_0$ the OLT sends a GATE message to ONU1, allowing it to send 6000 bytes (figure 1.4a).

2. Upon receiving the GATE from the OLT, ONU1 will adjust its local time according to the timestamp. It will send its data in the granted window (figure 1.4b). In the meanwhile the ONU keeps receiving new data packets from users. At the end of its transmission window, ONU1 will generate a REPORT message, telling the OLT how many bytes were in ONU1’s buffer at the moment the request was generated. In this case there were 550 bytes.
3. Even before the OLT receives a reply from ONU1, it knows when the last bit of ONU1’s transmission will arrive, because it knows when the transmission window starts and what size it is. Then, knowing the RTT for ONU2, the OLT sends a GATE to ONU2 such that the first bit from ONU2 will arrive soon after the last bit from ONU1, with only a small guard interval between (figure 1.4b). The guard intervals provide protection for fluctuations of RTT. Additionally, the OLT receiver needs some time to readjust its sensitivity due to the fact that every ONU can be located at a different distance from the OLT.

4. After some time, the data from ONU1 arrives. At the end of the transmission from ONU1, there is a new REPORT message that contains information on how many bytes remained in ONU1’s buffer after transmission. The OLT will use this information to update its polling table (figure 1.4c).

5. Similarly to the above step, the OLT can calculate the time when the last bit from ONU2 will arrive. Hence, it will know when ONU3’s transmission window should start so that its data is appended to the end of ONU2’s data. After some more time, the data from ONU2 will arrive. The OLT will again update its table, this time the entry for ONU2 (figure 1.4d). Note that if an ONU emptied its buffer completely, it will report 0 bytes back to the OLT. Correspondingly, in the next cycle, the ONU will be granted 0 bytes, that is, it will be allowed to send a REPORT, but no data.
2.2.2 Service disciplines

There exist several different services disciplines, i.e. ways for the OLT to determine the granted window size $W_i$ [bits] for ONU i, depending on the requested window $V_i$ [bits] [3]. The cycle times and the packet delay for the following three disciplines will be analyzed in chapter 4:

**Fixed service**

This service discipline ignores the window, requested by the ONU; instead, the OLT will always grant the maximum window to all ONUs. This causes the cycle time to be constant and maximal.

\[ W_i = W_{\text{MAX}} \]
**Gated service**

This approach imposes no limit on the size of the granted transmission window; the ONU is always authorized to send the amount of bytes it requested. This means that the cycle time will grow with the traffic load.

\[ W_i = V_i \]

**Limited service**

With this service discipline, an ONU is granted its requested transmission window, but not more than \( W_{\text{MAX}} \). With this approach the cycle time is variable, but it will not surpass a certain limit.

\[ W_i = \min \left\{ V_i, W_{\text{MAX}} \right\} \]

Other possible service disciplines consist of trying to predict how many bytes an ONU will hold at the moment its transmission window begins. If the OLT manages to do so, all packets arriving in a cycle will be sent in the first transmission window (counting from their arrival). This way, one can decrease packet delay. The simplest approaches here are to add a constant credit to the requested window or to multiply the requested window size by a constant (linear credit). Obviously, much more complicated prediction mechanisms can be thought of, depending on the type of traffic. However, it should be noted that overestimating will cause bandwidth to be lost.
3. Simulations and observations

3.1 Introduction

An analytical model for IPACT does not fall from the sky. It takes simulations and observations of both animated packet flows (in NAM) and analysis of traces. It is therefore not sufficient to just concentrate on end results (i.e. packet delays); examining intermediate processes often turns out to be helpful to understand the mechanisms that play a part in it. The importance of simulation should be stressed: leaving simulations aside (as in [3] and [11]) can result in losing sight of certain limitations or processes that are not immediately obvious. A model can then seem perfect in a theoretical sense, without describing the system as it works in reality.

The aim of this section is to describe some of the observations that lead to the analytical model that will be introduced in the next chapter. Also, some small calculations will be made. Knowledge hereof should help to understand the analysis in later sections; derivations should look more intuitive afterward and the concrete meaning of formulas should be clearer. The focus will be on cycle times, which turn out to be much more complex than might have been expected. This cycle time is defined as the time between the start of two successive transmission windows for a fixed ONU (in simulations most often chosen to be ONU 1).

Table 3.1 shows the parameters used throughout the simulations (in NS-2) described in this chapter: More details about the simulations can be found in Appendix B.1.

<table>
<thead>
<tr>
<th>N</th>
<th>Number of ONUs</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>λ</td>
<td>ONU arrival rate (Poisson traffic)</td>
<td>from 5 to 65 Mbit/s</td>
</tr>
<tr>
<td>T_{fibre}</td>
<td>Two-way delay on the EPON</td>
<td>200 µs</td>
</tr>
<tr>
<td>T_{proc}</td>
<td>Processing time</td>
<td>35 µs</td>
</tr>
<tr>
<td>T_{guard}</td>
<td>Guard time</td>
<td>1.5 µs</td>
</tr>
<tr>
<td>B</td>
<td>Packet size (network layer)</td>
<td>12000 bits</td>
</tr>
<tr>
<td>R_U</td>
<td>Upstream bandwidth on the EPON</td>
<td>1 Gbit/s</td>
</tr>
<tr>
<td>P_{max}</td>
<td>Maximum transmission window (fixed/limited)</td>
<td>10 packets</td>
</tr>
</tbody>
</table>

Table 3.1: Simulation parameters
3.2 Fixed service

From the description of fixed service in a previous section, it is obvious that the cycle time will be constant and equal for all traffic loads.

3.3 Gated service

In simulations, the granted window size of each ONU’s first grant is zero. This means that each ONU will be allowed to send a REPORT message, but no actual data, during the first cycle. Packet arrivals happen from the very beginning of the simulation, which means that, as from the second cycle, ONUs will receive grants allocating transmission windows. For this reason, it can take some cycles for the cycle time to attain its regime.

The evolution of cycle times will be considered for increasing traffic loads: 5, 25, 50, 57.5 and 65 Mb/s per ONU. These traffic loads were chosen because they clearly show some of the cycle time’s different characteristics and processes. Simulations are all 1 s long.

5 Mb/s

When the traffic load is 5 Mb/s per ONU, the cycle time from simulations shows the pattern seen in figure 3.1. The cycle time will stay at or slightly above the minimum value of 235 µs. A regime is attained immediately.

For a traffic load as low as 5 Mb/s, it seems intuitive that a minimum cycle time is long enough for all packets (having arrived early enough to generate a REPORT and receive a GATE) to be sent. This means that there are fewer packets arriving in a minimum cycle time than can be sent.

A quick calculation shows that the expected number of packet arrivals (all 16 ONUs together) in a minimum cycle time (235 µs) is equal to (see Appendix A.1 for an overview of properties of a Poisson process):

\[ E[N_{\tau_{\text{min}}}] = \frac{\Lambda}{B} T_{\text{cycle}}^{\text{min}} = \frac{16 \cdot 5 \cdot 10^6}{12000} \cdot 235 \cdot 10^{-6} = 1.57 \]
On the other side, the number of packets that can be sent in a minimum cycle time is equal to:

\[
T_{\text{cycle}}^\text{min} = N \cdot \left( \frac{T_{\text{guard}} + \frac{B_{\text{req}}}{R_U}}{B + 304} \right) \cdot R_U
\]

\[
\frac{235 \cdot 10^{-6} - 16 \cdot \left( 1.5 \cdot 10^{-6} + \frac{576}{10^9} \right)}{12304} \cdot 10^9 = 16
\]

The intuitive assumption is true: much less packets arrive than could be sent. Moreover, the probability of having more than 16 packet arrivals is (see again Appendix A.1):

\[
1 - \sum_{k=0}^{16} \exp \left( -\frac{5 \cdot 10^6}{12000} \right) \cdot \frac{\left( \frac{5 \cdot 10^6}{12000} \right)^k}{k!} = 4 \cdot 10^{-8}\% 
\]

This probability is close to zero. One could then wonder why the cycle time is not constant and equal to 235 µs. An initial explanation is fairly simple: a REPORT message is generated at the end of the transmission window; therefore, the time consumed by sending data packets in the transmission window needs to be added to the minimum cycle time of 235 µs. The average number of packet arrivals for all ONUs together in a minimum cycle time was calculated as 1.57, which means the average number of packet arrivals for one ONU in a minimum cycle time is 1.57/16 = 0.098. Concretely, an ONU has approximately one packet arrival every ten cycles, at 5 Mb/s traffic load. Since a packet is 12304 bits long (including Ethernet overhead and inter-frame gap), the average cycle time could then be expected to be:
whilst the simulated average cycle time is 237.34 µs. One could think this value is due to the arrivals’ stochastic nature, but repeating the simulation invariably leads to a value that is slightly higher than the calculated value. Taking a closer look at the trace file reveals a difference between the expected and the simulated distribution of cycle times. Theoretically, assuming the cycle time as experienced by the ONU is only determined by the packet arrivals at that ONU, the probability of having a 235 µs cycle time is equal to the probability of having zero arrivals; the probability of having a 235 + 12.304 = 247.304 µs cycle time is equal to the probability of having one packet arrival, etc. Those theoretical probabilities are calculated as (Appendix A.1):

\[
\Pr[N_{\text{min}} = 0] = \exp \left( -\frac{5 \times 10^6}{12000} \cdot \frac{235 \times 10^{-6}}{0!} \right) = 0.906
\]

\[
\Pr[N_{\text{min}} = 1] = \exp \left( -\frac{5 \times 10^6}{12000} \cdot \frac{235 \times 10^{-6}}{1!} \right) = 0.089
\]

\[
\Pr[N_{\text{min}} = 2] = \exp \left( -\frac{5 \times 10^6}{12000} \cdot \frac{235 \times 10^{-6}}{2!} \right) = 0.004
\]

... However, the distribution obtained from simulation is 82.4% (↔ 90.6% theoretical) cycles of 235 µs, 16.1% (↔ 8.9%) cycles of 247.304 µs, 1.4% (↔ 0.4%) cycles of 259.608 µs. It appears that the probability of having a cycle time greater than the minimum cycle time, is about twice that obtained using the assumption that only the traffic at the ONU determines the cycle time. The process that explains this discrepancy is the clustering of transmission windows: consider the example of two ONUs (called ONU n and ONU n+1) that receive successive transmission windows every cycle. It is than clear that if those ONUs receive the minimum transmission window, only separated by a guard time, in a certain cycle, the number of packets that ONU n can send in the next cycle will also cause the cycle time of ONU n+1 to be longer by the time necessary to send one packet. This reasoning can be extended to more than two ONUs.
Conclusion

At low traffic load, the cycle time as experienced by an ONU will almost certainly be determined by the packet arrivals at that ONU, and in addition to that, by the packet arrivals at the ONUs that are polled and granted transmission windows precedingly.

25 Mb/s

Consider now a traffic load that is five times higher than the previous one: 25 Mb/s per ONU. With the same reasoning as in the previous case, the expected number of packets arrivals in the minimum cycle time is 7.83, or 0.49 packets per ONU. This is still considerably lower than 16, the number of packets that could be sent in the minimum cycle. The probability of having more than 16 arrivals in a minimum cycle is as low as 0.3%. Even at this traffic load (41% of the instability limit), it seems that it is still mainly the traffic at the ONU and, when ONUs are clustered, the ONUs polled right before that ONU, that determines the cycle time almost completely. This is also clear from the trace in figure 3.2, where one can see that cycle times most often take values $235 + 12.304n$ µs ($n$ is a natural number). Again, the regime is attained immediately.

![Figure 3.2: Evolution of cycle times for 25 Mb/s traffic load per ONU and gated service discipline](image)

The average cycle time from the simulations is 250 µs, a little higher than the theoretical value of 241 µs (based on the assumption that only the ONU itself determines its cycle time). Similar calculations as for 5 Mb/s lead to probabilities of having 0, 1, 2, 3, … packet arrivals in a minimum cycle time (and consequently probabilities for cycle times of length 235 µs, 247.304 µs,
259.608 µs, 271.912 µs, …) equal to 61.3%, 30.0%, 7.3%, 1.2%, … Compare these values to simulated probabilities of having these respective cycle times: 28.8%, 37.6%, 17.9%, 6.4%. In addition to clustering ONUs causing cycle times to be longer, another mechanism that was negligible at 5 Mb/s comes into play here: with more packet arrivals, the probability of having cycle times longer than the minimum, becomes bigger. A higher probability for longer cycle times also means a higher expected number of packet arrivals per cycle time. This causes successive cycle times to become more and more correlated, as the traffic load increases. This can also be observed in figure 3.2: it sometimes happens that a couple of successive cycles are longer or shorter than average. To be exact, it should be mentioned that, from the point of view of an ONU, packet arrivals in a cycle time do not influence the next cycle time (the cycle in which the REPORT message is generated), only the cycle time after that (the cycle in which the GATE message arrives and the packets are sent). A detailed analysis follows in the next chapter.

**Conclusion**

At a medium traffic load and for the given parameters, the evolution of cycle times becomes more complex: it is still mainly the ONU (and the preceding ONUs in the cluster) that determine the cycle time, but the process of longer (respectively shorter) cycle times inducing longer (respectively shorter) cycle times is no longer negligible.

**50 Mb/s**

For a 50 Mb/s traffic load (82% of the stability limit), the expected number of packet arrivals is 15.7, close to the number of packets that can be sent in the minimum cycle time. Only a few cycle times will still be determined by the ONU traffic load (and the load of preceding ONUs). Instead, the aggregate EPON traffic load is very likely to be the determining factor, since the probability for having over 16 arrivals is not negligible (already 40% in the minimum cycle time). A phenomenon that arises more clearly here is long cycle times inducing even longer cycle times and short cycle times resulting in short cycle times. This is very clear in figure 3.3 where one can see periods of longer and periods of shorter cycle times. Successive cycle times are clearly correlated. The logical explanation is that, as the cycle becomes longer, the expected number of packet arrivals $\frac{\Lambda}{B}T_{cycle}$ increases as well. In the figure one can also see that at its regime, the cycle time fluctuates heavily around some average value. The fact that the cycle time tends to this average can be understood qualitatively as follows: there exists a cycle time in which exactly as many packets arrive as can be sent. The process being stochastic, sometimes there are more packet arrivals, sometimes less. If more packets arrive, this will cause a
longer cycle time and, therefore, a higher probability of greater than average packet arrivals. The fact that the system does not “explode” (i.e. an ever increasing cycle time) can be explained by the fact that the increase of the expected number of packet arrivals is not proportional to the increase of the cycle time; the number of packet arrivals increases more slowly. The explanation for lower-than-average cycle times is similar: the number of packet arrivals does not decrease proportionally to the cycle time, it decreases more slowly, which causes that the cycle time is more likely to evolve back to its average level than to continue to decrease.

![50 Mb/s](image)

Figure 3.3: Evolution of cycle times for 50 Mb/s traffic load per ONU and gated service discipline

**Conclusion**

For higher traffic load, the cycle time is mainly determined by the aggregate traffic load, i.e. the traffic load of all ONUs together, with successive cycle times being correlated

**57.5 Mb/s**

For this traffic load, which is already close to instability (94.4% of the maximum traffic load), the average simulated cycle time is 609 µs. Again the cycle time fluctuates heavily around this average value, with successive periods of longer and shorter cycles (figure 3.4). The expected number of packet arrivals in an average cycle time is equal to:

$$E\left[N_{T_{cycle}}\right] = \frac{\Lambda}{B} T_{cycle} = \frac{57.5 \cdot 10^6}{12000} \cdot 609 \cdot 10^{-6} = 2.92$$

or 46.7 packet arrivals for the entire EPON. On the other hand, the number of packet that can be sent in a 609 µs cycle is:
This is almost the same number of packets that arrive in 609 µs. This shows that at this high traffic load, the cycle time is indeed almost completely determined by the aggregate traffic load, because in the reasoning above, the limitation of a minimum cycle time was not considered.

65 Mb/s

It is also interesting to look at what happens when the traffic load passes the instability limit. Consider therefore a traffic load of 65 Mb/s (106.7% of the maximum traffic load). From figure 3.5 it is clear that then the system “explodes”: the cycle time keeps increasing. How can this be explained? Consider the first cycle time to be 235 µs; the expected aggregate number of packet arrivals is then:

\[
E[N_{T_{cycle}}] = \frac{\Lambda}{B} T_{cycle} = \frac{16 \cdot 65 \cdot 10^6}{12000} \cdot 235 \cdot 10^{-6} = 20.4
\]

A cycle in which 20.4 packets are sent, takes:

\[
N \cdot \left( T_{guard} + \frac{B_{req}}{R_U} \right) + 20.4 \cdot \frac{B + 304}{R_U} = 16 \cdot (1.5 \cdot 10^{-6} + \frac{576}{10^9}) + 20.4 \cdot \frac{12304}{10^9} = 284.2 \mu s
\]

Repeat this same reasoning for the new cycle time and a couple more times:

284.2 µs → 24.6 packets
From these numerical values, it is clear that the cycle time does not just increase, its rate of growth increases, as can be seen by the shape of the graph in figure 3.5. On the contrary, for traffic loads under the stability limit, the rate of growth decreases. Combined with the probabilistic nature of the traffic, this causes that, for a stable system, the cycle time to fluctuate around an average value. In a stable system, this phenomenon, combined with the probabilistic nature of the traffic causes the cycle time to fluctuate around an average value.

Figure 3.5: Evolution of cycle times for 65 Mb/s traffic load per ONU and gated service discipline

The cycle time is an intermediate process (which will prove to be important in the analysis), but packet delay is a more important parameter still. Consider therefore the following trace (figure 3.6) of packet delays from the same simulation for ONU 1; additional insight can be gained from it. One can observe the sawtooth pattern, that can be explained as follows: the “discontinuities” find place between packets sent in different cycles and packets arriving earlier in the same cycle experience longer delays. It is also clear that for this instable situation, cycle times become longer and longer. A last observation is that on average (not in one cycle), packet delays increase over time.
3.4 Limited service

The most interesting traffic loads to consider for limited service are those close to instability, because there the difference with gated service becomes clear. For low traffic loads the cycle time behaves the same as for gated service. This can be explained by the very low probability of having a request for more than the maximum transmission window.

60 Mb/s

It will be calculated in the next chapter that, for the parameters used, the gated service system is still stable for 60 Mb/s; the limited service system, on the contrary, is not (the stability limit is 58.8 Mb/s).

In the figure 3.7, one can see that the gated service cycle time is again fluctuating around some average value, whereas the limited service cycle time tends to 2 ms, the maximum cycle time. With the same reasoning as for 57.5 Mb/s, the average cycle time for gated service can be estimated to be 2.118 ms: for this traffic load, there are as many packets arriving as can be sent in one cycle. As was shown in the discussion of stability, the efficiency is greater for larger transmission windows/longer cycle times.

The limited service system also tries to evolve to this cycle time, but gets clamped and can’t reach the required efficiency to send 60 Mb/s. The fact that the cycle time still shows small fluctuations under 2 ms, is explained by the stochastic nature of the traffic, which causes certain ONUs to remain within a stable domain.

Figure 3.6: Evolution of packet delays past the stability limit for gated service
60 Mb/s

Figure 3.7: Evolution of cycle times for 60 Mb/s traffic load per ONU and gated or limited service discipline

65 Mb/s

Increasing the traffic load even further, causes the cycle time to reach its limit and clamp faster; fluctuations disappear almost immediately (figure 3.8). Compare this to the ever increasing cycle time for the gated service case.

Figure 3.8: Evolution of cycle times for 65 Mb/s traffic load per ONU and limited service discipline
3.5 General conclusion

In this chapter, it has been shown from simulations that, whereas the cycle time is constant for fixed service, the process is much more complex for gated and limited service. For those last two the cycle time varies much more for higher traffic loads than for low traffic loads. Furthermore, it has been observed how there exist correlations between cycle times. This knowledge will prove very useful in the next chapter.
4. Analytical model

4.1 Introduction

The system to be modelled looks fairly simple at first sight: the traffic sources are chosen to generate constant sized Poisson traffic; further on an EPON, consisting of an OLT and N ONUs and the dynamic bandwidth allocation scheme IPACT, regulate the upstream bandwidth. However, trying to capture all functionality and interdependencies into formulas will prove to be rather complex and approximations will have to be made to allow numerical results to be obtained.

To give only a brief overview of the complexity of the mechanisms that come into play (they will be treated in detail further on): cycle times influence each other (apart from the simplest case, fixed service) and have an impact on packet delay. The stochastic process of packet arrivals has its effect on the cycle time in return. For limited service as for fixed service, packets are not sure to end up in the first transmission window that was requested after their arrival. Such a backlog influences the system behaviour over multiple cycle times, affecting the number of packet arrivals in those cycles. It may be clear that simple queuing analysis does not capture all these specific properties: e.g. the best known and simplest queuing systems have underlying Markov chains, which means their future state only depends on the past through the present. In general, this is not the case for IPACT. In queuing systems literature no advanced models, modelling IPACT correctly, were found. Even though one could probably capture all interdependencies in formulas, this would lead to a system of equations that is far too complex to be solved (analytically and numerically). What follows in this chapter is an attempt to model IPACT for the EPON, as detailed as possible, but taking a more pragmatic point of view when necessary.

An important remark that one has to keep at the back of one’s mind at all times to prevent confusion is the point of view that is taken throughout the analysis. Sometimes this is the system as a whole (the EPON with N ONUs), at other times this is the ONU.

Generally speaking, for gated and limited service, the analysis of cycle times, queue sizes and packet delays will take the ONU’s viewpoint for lower traffic loads and the system’s viewpoint for higher traffic loads. For fixed service the ONU’s viewpoint is taken. This distinction in approach should be
clear from the use of $\lambda_i$ for the ONU $i$’s traffic load [Mb/s] and $\Lambda$ for the aggregate traffic load, logically calculated as:

$$\Lambda = \sum_{i=1}^{N} \lambda_i$$

where $N$ designates the number of ONUs in the EPON. In the majority of cases, all ONUs will be assumed to have the same traffic load $\lambda$, so that:

$$\Lambda = N \cdot \lambda$$

### 4.2 Stability analysis

Due to physical properties of optical fibre and lasers, transmission links are limited in bandwidth. For the EPON this bandwidth is standardized at 1 Gb/s (at data link layer). Flooding the EPON, by pushing the traffic load beyond this limit, will cause the system to become instable. At that point, no meaningful results for average queue size and packet delay can be obtained any more in simulations or numerical/analytical calculations. For gated service, cycle times will keep growing, causing time delays to increase as well. For limited and fixed service, buffers won’t run empty an infinite amount of times, also causing time delays to increase continuously. Buffers of limited size will experience overflow, resulting in high packet loss. Therefore, in this section the maximum allowed traffic load for the system to be stable will be calculated. In order to be able to calculate this maximum traffic load, the efficiency needs to be discussed.

The 1 Gb/s bandwidth has to be shared amongst the $N$ ONUs. This would mean that, if all ONUs have the same service level agreement, in a first approximation the bandwidth per ONU is equal to $1/N$ Gb/s. However, one must take several sources of overhead into consideration, which cause the available bandwidth to be lower. A big source of overhead comes from the guard time $T_{\text{guard}}$, the time between successive transmission windows. Furthermore, there is the time consumed by REPORT messages, sent upstream at the end of the transmission window. Thirdly, there is the Ethernet overhead (header and inter-frame gap).
4.2.1 Fixed service

The fixed service discipline makes up the simplest case, because each ONU is granted the maximum window $P_{\text{max}}$ [packets]. The maximum obtainable efficiency is then calculated as the proportion of time used to send the actual (network layer) data in one cycle to the total cycle time:

$$
\text{efficiency} = \frac{N \cdot B \cdot P_{\text{max}}}{R_U \left( N \cdot B_{req} + N \cdot (B + 304) \cdot P_{\text{max}} \right) + N \cdot T_{guard}}
$$

From the second chapter, it is known that the upstream bandwidth $R_U$ is standardized as 1 Gb/s and that a REPORT message counts 576 bits. Consider then the numerical example of an EPON with 16 ONUs each obtaining a window for 10 IP-packets of 12000 bits and a guard time of 1.5 µs. The number 304 in the formula accounts for the Ethernet overhead. The maximum efficiency is then found to be 95.9%. This means each ONU obtains $\frac{959}{16} = 59.9$ Mb/s (network layer). Formula (4.1) can easily be extended for ONUs with different service level agreements, therefore obtaining different maximum transmission windows.

Note: if one considers the network layer efficiency vis-à-vis the physical layer bandwidth, the efficiency is a factor 0.8 lower (due to 8/10B encoding). For the numerical example, the efficiency becomes 76.7%.

4.2.2 Gated service

For gated service, the efficiency can be higher than for fixed service, since there is no limitation for the granted window size. Basically, this means that the maximum efficiency is calculated as the limit of formula 4.1 for $P_{\text{max}}$ going to infinity, which gives:

$$
\text{efficiency} = \frac{B}{B + 304}
$$

For 12000 bit packets this results in 97.5% or 61 Mb/s per ONU. It is clear that bigger transmission windows result in higher channel utilization: a higher percentage of bandwidth is used on packets carrying actual data. However, big transmission windows also cause the cycle time to increase. This will result in high packet delay, which is undesirable.
Of course, due to the dynamic bandwidth allocation, if certain ONUs send less than they could, others can now obtain higher data rates.

### 4.2.3 Limited service

For limited service, the absolute maximum efficiency (for the entire EPON) is the same as for fixed service. In general, the efficiency in a cycle can be expressed as (assuming the cycle time is well past the minimum cycle time):

\[
efficiency = \frac{B \cdot \sum_{i=1}^{N} P_i}{R_U} = \frac{N \cdot B_{req} + (B + 304) \cdot \sum_{i=1}^{N} P_i}{R_U} + N \cdot T_{guard}
\]  

(4.2)

Here, \( P_i \) is the transmission window [packets] obtained by ONU \( i \). Consider, as a numerical example, an EPON with 16 ONUs having the same service level agreement and consequently having the same maximum transmission window. The same numerical values are used as in the fixed service example. This time, however, only three ONUs generate traffic, the other 13 send out requests for 0 bytes. Applying formula (4.2), results in an efficiency of only 89.5%. 10.5% of the cycle time is now “wasted” on overhead.

The previous example clearly illustrates that one has to be careful not to render a limited service system unstable if different ONUs undergo different traffic loads. This is because efficiency drops when certain ONUs send less than what is allowed for their maximum transmission window, while other ONUs can not compensate completely for this because they can not exceed their maximum transmission window.

Throughout the simulations and calculations that will be discussed later, maximum traffic loads will be limited to values slightly under the instability boundary (most often a maximum of 57.5 Mb/s for all ONUs). One has to keep in mind that, due to the stochastic nature of the traffic and the fact that queue sizes become infinite at the stability limit, traffic loads close to instability can give unexpected results over too short simulation periods.
4.3 Cycle time analysis

The last three letters in IPACT stand for « Adaptive Cycle Time »; therefore a thorough analysis of the cycle time (both theoretically and by means of simulation), makes a logical subject for this section.

Remember, the cycle time was defined as the time between the start of two successive transmission windows for a fixed ONU. One must realize that, for gated and limited service, different ONUs do not necessarily experience the same cycle time. For example, two ONUs that are polled successively will only experience the same cycle time if they are clustered and receive the same transmission window in two successive cycles, assuming that this cycle time is bigger than the minimum cycle time (distance being the physical limitation). However, over a long time, all ONUs experience the same average cycle time. These and other subtleties of cycle times will be explained in this section. The cycle time will be analyzed and analytical results will be compared to simulations for fixed, gated and limited service. As in the original model ([3]), this analysis assumes constant packet sizes (B bits) and Poisson arrivals. For details about Poisson traffic, have a look at Appendix A.1.

4.3.1 Fixed service

As mentioned earlier, fixed service leads to a constant cycle time. This cycle time is equal to:

\[
T_{\text{cycle}} = \frac{N \cdot B_{\text{req}} + N \cdot (B + 304) \cdot P_{\text{max}}}{R_U} + N \cdot T_{\text{guard}}
\]

(4.3)

with again N the number of ONUs in the EPON, B_{req} the size of a REPORT/request message [bits], P_{max} the maximum transmission window [packets] for each ONU, R_U is the upstream bandwidth [bits/s] and T_{guard} the guard time.

As a numerical example, consider the case where each of the 16 ONUs receive a maximum transmission window for 10 maximum sized IP-packets (1500 bytes). Taking into account Ethernet-headers and 12 byte inter-frame gaps, the standard REPORT message being 64 bytes (plus 8 bytes preamble) and the guard time chosen to be 1.5 µs, this results in a 2.0 ms cycle time.
4.3.2 Gated service

4.3.2.1 Average cycle time: a simple model

A first and rather simplistic approach to analyze the cycle time (similar, but more detailed than the one described in [11]) is the following:

Suppose the cycle time settles around an average value, for which the average number of arrivals equals the number of packets that can be sent. This can be captured in formulas:

\[
T_{cycle} = N \left( T_{guard} + \frac{B_{req}}{R_U} \right) + E[N_{t_{cycle}}] \frac{B + 304}{R_U}
\]  

The expected number of packet arrivals can be easily calculated for Poisson traffic (appendix A.1):

\[
E[N_{t_{cycle}}] = T_{cycle} \frac{\Lambda}{B}
\]  

Substituting formula 4.4 in formula 4.5 and solving for \( T_{cycle} \) gives:

\[
T_{cycle} = \frac{N \left( T_{guard} + \frac{B_{req}}{R_U} \right)}{1 - \frac{\Lambda \cdot (B + 304)}{R_U \cdot B}}
\]

Even though this approach will turn out to be incomplete, some interesting features arise from it:

- Leaving aside the time consumed by REPORT messages, the cycle time is proportional to the guard time. Reducing the guard time causes the average cycle time to become much smaller. This corresponds to what will be observed in figure 4.2 when comparing 1.5 and 5 µs guard times.

- The cycle time is a hyperbolic function of the aggregate traffic load. For \( \Lambda = \frac{R_U \cdot B}{B + 304} \), the average cycle time becomes infinite. This is another way of deriving the instability limit for gated service. For traffic loads beyond this limit, the formula loses its value, because obtained cycle times are negative. Of course, in reality a cycle time can not be negative; instead, what will happen was shown in figure 3.5: the cycle time grows continuously, so it has no average value.

Compare now this theoretical cycle time with simulated cycle times for \( T_g = 5 \) µs (figure 4.1).
For high traffic loads (as from 45 Mb/s), this formula manages to predict the average cycle time almost perfectly, whereas for lower traffic loads there is a big discrepancy. The discussion thereof and a more detailed analysis of the cycle time are the subject of the next sections.

### 4.3.2.2 Cycle time: complexity

A first explanation for the discrepancy in figure 4.1, lies in the fact that for gated service, the physical limitation comes into play at low traffic load: the time between the start of two successive transmission windows can not be smaller than the time between the generation of the REPORT and the arrival of the next gate message. If all the ONU are separated from the OLT by an optical fibre of length $l$, this means the cycle time will always be bigger than $T_{\text{fibres}} = \frac{2nl}{c}$, where $c$ is the speed of light in vacuum.

As optical fibre has a refractive index $n = 1.5$, a 20 km fibre between ONU and OLT causes the cycle time to be at least 200 $\mu$s. To this value must still be added the time it takes to generate and interpret the REPORT and GATE message, called $T_{\text{proc}}$, in simulations chosen to be 35 $\mu$s. The absolute minimum for the cycle time is thus given by:

\[ T_{\text{min}} = T_{\text{fibres}} + T_{\text{proc}}. \]

---

4 In the analysis and in simulations, the ONUs are chosen to be at 20 km from the OLT, because this makes up the worst case scenario.
\[ T_{\text{cycle}}^{\text{min}} = T_{\text{fibre}} + T_{\text{proc}}. \]

There is no upper limit for the cycle time here, since for gated service every request is granted.

The minimum cycle time under traffic load differs slightly from the physical minimum mentioned above, and can have different values in different cycles because it is also determined by \( P \), the number of packets that are sent. The time consumed by sending those packets must be added to \( T_{\text{cycle}}^{\text{min}} \), because it is the time that expires between the arrival of the GATE message and the sending of the REPORT at the end of transmission. The minimum cycle time under traffic load becomes in a first approximation:

\[ T_{\text{cycle}}^{\min'} = T_{\text{cycle}}^{\text{min}} + \frac{P \cdot (B + 304)}{R_{t}}. \]

One could have expected this formula to be correct, but it turns out that the ONUs that are polled right before a specific ONU also have their influence on the cycle time, as experienced by that ONU. The easiest way of explaining this is by means of an example: consider two ONUs that are polled successively and suppose that in a certain cycle they do not send actual data and their REPORT messages are separated by only the guard time. It is then clear that in the next cycle the granted transmission window for the first ONU will also have its influence on the cycle time the second ONU experiences, because this ONU will only be allowed to send after the first ONU’s transmission window has ended. This reasoning can be extended to multiple ONUs that are “clustered”, by which is meant that the REPORT messages they send are separated by only a very small time (most often, the guard time). This process will be treated into more detail in the next section. It should be clear that the phenomenon of clustering is especially important at low traffic loads. At high traffic load the aggregate traffic load (all ONUs) determines the cycle time, as was observed in simulations in the previous chapter.

Remark: REPORT messages are sent at the end of the transmission window. If this were not the case, analysis would be easier, because then the minimum cycle time under traffic load would be equal to the absolute minimum cycle time.

### 4.3.2.3 Cycle time: analysis

In the previous section, it became clear that even the cycle time is too complex for an exact analysis. Therefore, the analysis, proposed in this section, is an approximation. However, this approximation
will lead to results very close to the ones obtained through simulation. Not only will the average cycle
time be derived, also the distribution of cycle times will be estimated (for high traffic loads). This will
prove to be extremely useful when deriving packet delays later on.

It is necessary to distinguish low and high traffic loads, because, as was shown in the previous chapter,
of the different mechanisms in play. A rule of thumb will be used to make the distinction. Calculate the
probability of having more packet arrivals than can be sent in a minimum cycle time (appendix A.1):

\[
1 - \sum_{k=0}^{P} \exp \left( - \frac{\Lambda}{B} T_{\text{cycle}}^\text{min} \right) \left( \frac{\Lambda}{B} T_{\text{cycle}}^\text{min} \right)^k \frac{1}{k!}
\]  

(4.7)

Here \( P \) is the maximum number of packets that can be sent in a minimum cycle time:

\[
P = \left( T_{\text{cycle}}^\text{min} - N \cdot (T_{\text{guard}} + 576 \cdot 10^{-6}) \right) + \frac{R_U}{B + 304}
\]

If this probability is lower than 5%, the analysis for low traffic is the most appropriate, for higher
probabilities, the more complex analysis for high traffic loads is more suitable.

**Low traffic load**

As was observed in the previous chapter, for low traffic loads, it is mainly the ONU and (in the case of
clustering) the ONUs that are served before that ONU, that will determine the cycle time (as observed
by the ONU). For low traffic loads it suffices to know the average cycle time to derive packet delay. It
can than be estimated as:

\[
\frac{T_{\text{cycle}}}{T_{\text{cycle}}} = T_{\text{cycle}}^\text{min} + (1 + C) \frac{\Lambda}{B} T_{\text{cycle}}^\text{min} \frac{B + 304}{R_U}
\]

(4.8)

This is the minimum cycle time plus the time necessary for sending the packets. The factor of \( C \) takes
the effects of clustering into account.

**Clustering**

What follows here, is an approximate derivation for the clustering factor from the formula
above.

At low traffic load, it was observed that cycle times are only slightly bigger than the minimum
cycle time. Here it will be assumed that the cycle time is constant and corresponds to a whole
number of packets, i.e. the maximum number that can be sent in the cycle time \( P \), derived
earlier.
To focus attention, consider two successive ONUs, sending out REPORTs to the OLT in a certain cycle, that are only separated by a guard time. It is then clear that the data sent upstream by the first ONU will have its effect on the start of the transmission window for the second ONU. For instance, if the first ONU sends one packet, the transmission window for the second ONU will be delayed by the duration of sending one packet. This reasoning can be extended to several successive ONUs. Because of this, the beginnings of the different transmission windows are normally a whole number of packets apart. This allows for an approximate discrete analysis: REPORTs can only fall at a certain number of places.

In order to derive how big a cluster of packets is on average, consider a certain ONU, with its REPORT at the end of the cycle, that was assumed constant and a whole number of packets. Because traffic is symmetric, the whole system is symmetric, and the other N-1 REPORTs have equal probability to fall at any of the discrete points. The problem has been abstracted to putting N-1 REPORTs at P+1 places that each have \((P+1)^{-1}\) probability. It should be clear that several REPORTs can take the same place (separated by a guard time). The number of interest is the average number of REPORTs right before the considered ONU. This number is the average of a binomial distribution with N-1 trials and a chance 1/P of success. This average is known to be \(C = \frac{N-1}{P+1}\). For the 16 ONU EPON and the guard time of 5 respectively 1.5 μs, this value is calculated as 1.25 and 0.88.

Remark: this analysis does not take into account that successive ONU, separated by more than a guard time can influence each other, if the first one sends more than one packet. The probability of such an event, however, can be neglected at low traffic load.

**High traffic load**

For higher traffic loads, the aggregate traffic plays the most important role. The physical limitation and the effect of the ONU’s traffic load still have their influence and will be taken into account, by considering the minimum cycle time \(T_{\text{cycle}}^{\text{min}}\) to be:

\[
T_{\text{cycle}}^{\text{min}} = T_{\text{cycle}}^k
\]

with \(k = \min \left\{ n \cdot \frac{(B+304) + N \cdot B_{\text{req}}}{R_U} + N \cdot T_{\text{guard}} > T_{\text{cycle}} \right\} \)
where $T_{\text{cycle}}$ is the value, obtained from formula 4.8. Another assumption, necessary for rendering the analysis feasible, is that the cycle time only takes the following discrete values:

$$T_{\text{cycle}}^n = \frac{n \cdot (B + 304) + N \cdot B_{\text{req}}}{R_U} + N \cdot T_{\text{guard}}$$

for $n \geq k$

The basic idea for deriving the distribution is to abstract the evolution of cycle times as a Markov chain: call $T_{\text{cycle}} = (T_{\text{cycle}}(n), n \in N)$ the series of cycle times with the designated discrete values in $[T_{\text{cycle}}^{\text{min}'}, \infty[$, then:

$$P(T_{\text{cycle}}(n + 1) = j | T_{\text{cycle}}(k) = i, 0 \leq k \leq n) = P(T_{\text{cycle}}(n + 1) = j | T_{\text{cycle}}(n) = i)$$

One can estimate the probability of transition from one specific cycle time to another, when the arrival rate for all ONUs together is $\Lambda$ [bits/s]:

$$P(T_{\text{cycle}}(n + 1) = j | T_{\text{cycle}}(n) = i) = \exp\left(-\frac{\Lambda}{B} T_{\text{cycle}}^i \right) \frac{\left(\frac{\Lambda}{B} T_{\text{cycle}}^i \right)^j}{j!}$$

In words: the probability of transition from a cycle time in which $i$ packets are sent to a cycle time in which $j$ packets are sent is equal to the probability of having $j$ arrivals in a time $T_{\text{cycle}}^i$. Note that this only accounts for transitions to cycle times that are bigger than the minimum cycle time (under traffic load). The probability of transition to the minimum cycle time (under traffic load) is logically given by:

$$P(T_{\text{cycle}}(n + 1) = T_{\text{cycle}}^{\text{min}'}) = \sum_{j:T_{\text{cycle}}^j \leq T_{\text{cycle}}^{\text{min}'}} \exp\left(-\frac{\Lambda}{B} T_{\text{cycle}}^i \right) \frac{\left(\frac{\Lambda}{B} T_{\text{cycle}}^i \right)^j}{j!}$$

These transition probabilities allow a transition matrix $P$ to be defined:

$$P = \begin{pmatrix}
p_{0,0} & p_{0,1} & \ldots & p_{0,M} 
p_{1,0} & p_{1,1} & \ldots & \ldots 
p_{2,1} & p_{2,2} & \ldots & p_{2,3} 
\vdots & \vdots & \ddots & \vdots 
p_{M,0} & p_{M-1,M-2} & \ldots & p_{M-1,M-1} & p_{M-1,M} 
p_{M-1,0} & p_{M-1,M-1} & \ldots & p_{M-1,M} & p_{M,M}
\end{pmatrix}$$

with elements:

$$p_{i,j} = P(T_{\text{cycle}}(n + 1) = T_{\text{cycle}}^j | T_{\text{cycle}}(n) = T_{\text{cycle}}^i)$$

for $i > 0$ and $j > 0$
To obtain numerical results, one has to constrain the size of the matrix to a sufficiently large value. Limiting the matrix’s dimension causes the sum of the elements of a line in the transition matrix no longer to be unity, a necessary property for a Markovian matrix. This problem is solved by defining:

\[ P_{i,M} = 1 - \sum_{j=0}^{M-1} P_{i,j} \]

The distribution of cycle times can then be found by solving the following set of equations:

\[
\begin{cases}
  P\pi = \pi \\
  \sum_{i=0}^{M} \pi_i = 1
\end{cases}
\]

Here \( \pi \) is an \((M+1) \times 1\) matrix with elements \( \pi_i \) \((0 \leq i \leq M)\), giving the probabilities for different cycle times: \( \pi_0 \) is the probability of having the minimum cycle time (under traffic load), \( \pi_i \) is the probability that the cycle time is \( T_{i+1}^{\text{cycle}} \). To solve this system, one has to take into account that \( P\pi = \pi \) is a system of rank \( \text{dim}(P)-1 \). Therefore, one can suppress one column of \( P \) (for instance the last one) and replace it with a column entirely composed of ones. Call the so obtained matrix \( \hat{P} \). One now has to solve the following set of equations:

\[ \pi (\hat{P} - \hat{I}) = b \]
\[ b = (0 \ldots 0 \ 1) \]
\[ \hat{I} = \begin{pmatrix}
  1 & 0 & 0 \\
  0 & \ldots & \ldots \\
  \ldots & \ldots & 1 & 0 \\
  0 & 0 & 0 
\end{pmatrix} \]

The average cycle time is then easily obtained as:

\[ \overline{T_{\text{cycle}}} = \sum_{i=0}^{M} \pi_i \cdot T_{i+1}^{\text{cycle}} \]

### 4.3.2.4 Simulations and analytical results

This section treats simulations investigating the evolution of the cycle time as well as analytical results for gated service under various traffic loads.
Whereas the simple analysis could only estimate the cycle time for high traffic load, this more thorough analysis gives good results for all traffic loads (figure 4.2). The cycle time grows very slowly for the first part of the graph. This is the part where it is mainly the ONU’s traffic load (and the traffic load of the ONUs served right before that ONU) that determines the cycle time. From a certain point, the aggregate traffic load starts to determine the average cycle time and the average cycle time increases faster.

As was predicted by the simple analysis, lowering the guard time reduces average cycle time drastically: at 57.5 Mb/s the average cycle times for 1.5 and 5 µs guard time are respectively 610 and 1569 µs, a factor 2.57 different.

![Gated Service](image)

Figure 4.2: Average cycle time: comparison simulation – analytical model

### 4.3.3 Limited service

#### 4.3.3.1 Analysis

The cycle time for limited service can, in a first approximation, be analyzed in a way that resembles the gated service analysis. This analysis will turn out to have its limitations, but will nevertheless provide insight in the matter. The details of the gated service will not be repeated here, instead only the difference in approach will be highlighted.
Since for limited service, ONU i’s transmission window is limited to \( P_{\text{max}} \) packets, the cycle time will, contrary to gated service, not become bigger than the following value:

\[
T_{\text{cycle}}^{\text{max}} = \frac{N \cdot B_{\text{req}} + N \cdot P_{\text{max}} \cdot (B + 304)}{R_U} + N \cdot T_{\text{guard}}
\]

Note that this is the cycle time for fixed service. In the analysis of the previous section, the cycle time for gated service was only constrained to a sufficiently large value in order to be able to obtain numerical results. For gated service, it makes sense to limit the dimension of the transmission matrix to a value that corresponds to \( T_{\text{cycle}}^{\text{max}} \). A similar set of equations will have to be solved in order to calculate the approximate cycle time distribution, from which the average cycle time will follow. Results of this analysis will be compared to simulation and discussed in the next section.

### 4.3.3.2 Simulations and analytical results

For a limited service simulation with parameters \( P_{\text{max}} = 10 \), and guard time 1.5 or 5 µs, the analysis gives near-perfect results (figure 4.3). Note that the results are practically equal to the results for gated service, which can be explained by the fact that even for high traffic load the average cycle time is still under its maximum value, equal to 2.0 ms: 1.565 ms for \( T_{\text{guard}} = 5 \) µs and 0.609 ms for \( T_{\text{guard}} = 1.565 \) µs.

![Figure 4.3: Average cycle time: comparison simulation – analytical model and P_{\text{max}} = 10](image-url)
Contrary to gated service, limited service has an additional parameter that should be varied to check the correctness of the analysis. Choose $P_{\text{max}} = 3$, this reduces the maximum cycle time to 623.8 $\mu$s. It follows from figure 4.4 that there is a significant difference between the simulation and analytical result for the highest traffic load. A qualitative explanation for this is found in the fact that, for this smaller transmission window, the probability of having more packet arrivals than can be sent in the transmission window is no longer negligible. In fact, backlogs start to influence cycle times. If a packet cannot be sent in its first requested transmission window, it will have an influence over multiple cycle times. The analysis does not take this into account and there does not seem to be a feasible way of extending the analysis to include this feature.

![Limited Service](image)

Figure 4.4: Average cycle time: comparison simulation – analytical model for $P_{\text{max}} = 3$ and $T_{\text{guard}} = 1.5$ $\mu$s

### 4.4 Access delay analysis

The cycle time analysis from the previous section will prove to be useful in predicting packet delays. This access delay, as a function of traffic load, is one of the most important performance parameters in access networks. The aim of this part of the analytical model is to evaluate this access delay. To check the correctness of the model, analytical results will be compared to simulations.

The definition of access delay gives less ground for confusion than the definition of cycle time. For EPONs, access delay is the time between reception and emission (in the requested transmission
window) of a packet by an ONU. However, there are still some subtleties: depending on the bandwidth, it takes more or less time to receive or send a packet. Therefore, the packet arrivals at the ONU are considered to be the moments when the packet has arrived completely, the time of emission is the moment the packet starts being sent to the OLT.

As for the cycle time, the (increasingly more complex) cases of fixed, gated and limited service will be discussed.

4.4.1 Fixed service

4.4.1.1 Analysis

Remember that this service discipline ignores the window requested by the ONU. It always grants the maximum window to all ONUs. The performance for this service discipline serves as a reference for the other service disciplines.

The cycle time $T_{\text{cycle}}$ for fixed service was found earlier (formula 4.3). Since this cycle time is constant, the system will be considered at discrete moments that are $T_{\text{cycle}}$ apart and right after the ONU has sent packets in the granted transmission window. When deriving the packet delay, the sending of packets will be considered not to consume time, in a first approximation. An extra term will be ultimately added to compensate for this.

Suppose ONU $i$ is granted its first transmission window, starting at $t = 0$. Its transmission window in the second cycle will than start at $t = T_{\text{cycle}}$ and its transmission window in the $n^{\text{th}}$ cycle will start at $t = (n-1).T_{\text{cycle}}$. Therefore, call $Q(n)$ the queue size [packets] of the ONU at $t = n.T_{\text{cycle}}$. $Q(n)$ is a discrete Markov chain because:

$$\Pr(Q(n+1) = j|Q(k) = i_k, 0 \leq k \leq n) = P(Q(n+1) = j|Q(n) = i_n)$$

In words: the next queue size (at $t = (n+1).T_{\text{cycle}}$) only depends on the past through the present (instant $n.T_{\text{cycle}}$). Moreover, this Markov chain is homogenous because $\Pr(Q(n+1) = j|Q(n) = i)$ does not depend on $n$. This makes that it is possible to define transition probabilities

$$p_{i,j} = \Pr[Q(n+1) = j|Q(n) = i]$$

that make up the elements of the transition matrix that describes the evolution of the system in a probabilistic way:
These transition probabilities are found to be:

\[ p_{i,0} = \sum_{k=0}^{P_{\text{max}}-i} \Pr[N_{\text{cycle}} = k] \quad \text{for } i \leq P_{\text{max}} \]

This formula says that it is only possible for a queue to send all its packets in one transmission window if the number of packets present is equal to or smaller than the maximum transmission window. The probability for such a transition is then given by the sum of the probabilities of having not more packet arrivals than the maximum transmission window minus the number of packets present.

\[ p_{i,j} = \Pr[N_{\text{cycle}} = P_{\text{max}} + j - i] \quad \text{for } i \geq 0, j > 0 \text{ and } j - i \geq -P_{\text{max}} \]

\[ p_{i,j} = 0 \quad \text{for } i, j \geq 0 \text{ and } j - i < -P_{\text{max}} \]

More concretely, the probability of having a transition of \( i \) packets in the queue at an instant \( n \) to \( j \) packets at an instant \( n + 1 \), is equal to having \( P_{\text{max}} + j - i \) arrivals in a period \( T_{\text{cycle}} \), where \( P_{\text{max}} \) is the number of packets in a maximum transmission window. The probability of having a transition from \( i \) to \( j \) is zero, if \( j \) is smaller than \( i - P_{\text{max}} \), because the transmission window is restricted. From the Poisson properties (appendix A.1), it follows that:

\[ p_{i,j} = \exp\left(-\frac{\lambda}{B} T_{\text{cycle}}\right) \left(\frac{\lambda}{B} T_{\text{cycle}}\right)^{P_{\text{max}}+j-i} \frac{P_{\text{max}}+j-i}{(P_{\text{max}}+j-i)!} \quad \text{for } i \geq 0, j > 0 \text{ and } j - i \geq -P_{\text{max}} \]

and

\[ p_{i,0} = \sum_{k=0}^{P_{\text{max}}-i} \exp\left(-\frac{\lambda}{B} T_{\text{cycle}}\right) \left(\frac{\lambda}{B} T_{\text{cycle}}\right)^{k} \frac{1}{k!} \quad \text{for } i \leq P_{\text{max}} \]

Note that \( \lambda \) is the ONU’s arrival rate [Mb/s].

For an infinite buffer, there is no limit for the number of packets in the queue. For the analysis this means that the transition matrix would have to be of infinite dimension. Because no results could be
obtained in that way, the matrix’ dimension has to be limited to some sufficiently large value.
However, limiting the dimension of the matrix causes the sum of the elements of a line in the transition matrix no longer to be one, a necessary property for a Markovian matrix. This problem is solved by defining:

\[
P_{i,M} = 1 - \sum_{j=0}^{M-1} P_{i,j}
\]

To find the average queue sizes for this stationary system, one has to solve the following set of equations:

\[
P \pi = \pi
\]

\[
\sum_{i=0}^{M+1} \pi_{i,1} = 1
\]

where \( \pi \) is the \( 1 \times (M+1) \) matrix giving the probabilities of the queue occupancies 0 to M. It is clear that the system’s dimension necessitates numerical analysis. To solve this system, one has to take into account that \( P\pi = \pi \) is a system of rank M. Therefore, one can suppress one column of \( P \) (for instance the last one) and replace it by a column entirely composed of ones. Call the so obtained matrix \( \hat{P} \). The following set of equations now must be solved:

\[
\pi(\hat{P} - \hat{I}) = b
\]

\[
b = \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix}
\]

\[
\hat{I} = \begin{pmatrix}
1 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 1 & 0 \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
\end{pmatrix}
\]

The average queue size \( \hat{Q} \) at the discrete moments in time considered is given by:

\[
\hat{Q} = \sum_{i=1}^{M+1} \pi_{i,1} (i - 1)
\]

To convert \( \hat{Q} \) to the average queue size in continuous time two additive terms \( Q_1 \) and \( Q_2 \) must be adopted. Since Poisson arrivals are uniformly distributed over time (i.e. they are not bursty), the average queue size in continuous time will be equal to the average queue size exactly in between the discrete moments considered earlier (assuming the time it takes to send the packets is negligible).

Therefore the first additive term is given by the average number of arrivals in \( \frac{T_{\text{cycle}}}{2} \) seconds:
To be even more correct, one can add a second term $Q_2$, that accounts for the time consumed by sending the packets:

$$Q_2 = \frac{\lambda}{B} \times \frac{\lambda T_{\text{cycle}}}{B + 304} \times \frac{B}{2}$$

This last formula can be interpreted as the percentage of time the ONU is sending multiplied by the average number of packets that are to be sent in the transmission window, divided by two. This term is, generally speaking, much smaller than the sum of $\hat{Q}$ and $Q_1$.

The average queue size $\overline{Q}$ in continuous time is given by the following sum:

$$\overline{Q} = \hat{Q} + Q_1 + Q_2$$

To make this more concrete, have a look at figure 4.5, a partial trace of queue size at a traffic rate of 57.5 Mb/s. $\hat{Q}$ accounts for the average number of packets that stay in the queue at the end of a transmission window, this is right after the steep descent. $Q_1$ allows converting to continuous time by considering the number of packets right in between two transmissions. $Q_2$ indeed does not contribute considerably to the average queue size. Therefore, the descent is too steep. This last term could be neglected without introducing a considerable error.
The average time delay [s] follows from Little’s law:

\[ \bar{W} = \frac{B\bar{Q}}{\lambda} \]

### 4.4.1.2 Simulations - analytical results

In this section analytical and simulated results for queue sizes and packet delays will be compared and briefly discussed for two different guard times: 1.5 µs and 5 µs.

From figure 4.6, it is clear that the fixed service analysis is almost perfect: the two curves nearly overlap. The only small discrepancies appear at very high traffic load, but even this can be explained by the fact that near instability, the system becomes more susceptible to small variations, due to the probabilistic nature of the Poisson traffic.

For both guard times, the relationship between traffic load and average queue size is approximately linear for traffic loads ranging from 5 to 40 Mb/s; for higher traffic loads the average queue size approaches infinity. This behaviour can be explained as follows: up to 40 Mb/s, packets are very likely to be sent in the next transmission window (term \( Q_1 \) is dominant), whereas near instability the probability for packets to queue for several cycles becomes important (\( \hat{Q} \) becomes dominant).
Figure 4.6: Average queue size: comparison simulation – analytical model for $P_{\text{max}} = 10$

The explanation for the packet delay (figure 4.7) is similar: if the traffic load is sufficiently low, packets will most likely be sent in the next transmission window, which starts on average half a cycle (1.0 ms) later. This causes the first part of the graph to be nearly constant. For higher traffic loads, packets are more likely to wait for several cycles, so the average packet delay can surpass the cycle time.

Figure 4.7: Average packet delay: comparison simulation – analytical model for $P_{\text{max}} = 10$

Comparing the difference in results for the two guard times leads to the conclusion that this parameter can improve the system’s performance, in terms of packet delay, especially at high traffic loads. This is
because lowering the guard time increases the stability limit. This, combined with the asymptotic
behaviour near instability explains the large improvement in packet delay.

The analysis can also be checked for a smaller transmission window, for instance $P_{\text{max}} = 3$ for all ONUs
(figure 4.8). Again, theory and simulation coincide and the interpretation of the curve is completely
similar.

![IPACT Fixed - $P_{\text{max}} = 3$](image)

Figure 4.8: Average packet delay: comparison simulation – analytical model for $P_{\text{max}} = 3$

### 4.4.2 Gated service

#### 4.4.2.1 Analysis

As for the cycle time analysis, one must distinguish low and high traffic loads in the same way as
before.

**Low traffic load**

A packet that arrives will not be sent in the first transmission window (counted from its arrival).
Indeed, the ONU has to send a REPORT for the packets that have arrived during the cycle and wait for
the GATE to arrive, so a packet is sent in the second transmission window. Given the uniform
distribution of Poisson arrivals (by uniform is meant not bursty), a packet arrives at the queue on
average half-way between two transmission windows. Therefore on average it stays in the queue for
one and a half cycle. For low traffic loads, the cycle time shows only small fluctuations above its minimum value, which means that a good approximation of the mean waiting time is given by:

$$\bar{W} = \frac{3}{2} \frac{T_{cycle}}{2}$$

From Little’s law the average queue size follows immediately:

$$\bar{Q} = \frac{\lambda}{B} \cdot \bar{W}$$

**High traffic load**

In a first approximation, one could take the same approach as for low traffic load. However, this leads to serious underestimations. The explanation lies in the fact that for high traffic load, big fluctuations in cycle time take place. Knowledge of only the mean cycle time is then no longer sufficient, for the following reason: the probability for a packet to arrive in a cycle of some length is not only proportional to the probability of that cycle length, but also to the cycle length itself. This is logical: leaving the distribution of cycle times aside, supposing all cycle times have equal probability, a packet is more likely to arrive in a longer than in a shorter cycle.

Remember, from the cycle time analysis, the cycle time distribution was found to be (section 4.3.2.3):

$$\pi_i = \Pr\left(T_{cycle} = T^{i+k}_{cycle}\right)$$

The probability for a packet to arrive in a cycle of duration $T^{i+k}_{cycle}$ (again using the discrete approximation), is then:

$$\tilde{\pi}_i = \frac{\pi_i T^{i+k}_{cycle}}{\sum_j \pi_j T^{j+k}_{cycle}}$$

The denominator normalizes the probability. With the same reasoning as in the previous case, one can then estimate the average waiting time to be:

$$\bar{W} = \frac{3}{2} \sum_i \tilde{\pi}_i T^{i+k}_{cycle}$$
4.4.2.2 Simulations and analytical results

Simulations and lengthy numerical calculations lead to the results in figures 4.9 and 4.10. The analysis predicts packet delay well, but slightly overestimates for the higher traffic loads, which is explained by the fact that the analysis is approximative.

As for fixed service, one can see that for the first part of the graph, packet delays increase very slowly; this is the domain determined by the ONU’s traffic and by the traffic of the ONUs that are polled right before that ONU. From figure 4.2, one can see that in this domain, the average cycle time is still very close to its minimum value. For higher traffic loads, aggregate traffic load becomes the determining factor and packet delay increases quickly. Comparing between different guard times leads to the conclusion that this parameter strongly influences packet delay. The explanation follows naturally from the simple model in section 4.3.2.1 and the fact that a packet stays on average one and a half cycles in the queue.

![Gated Service - Tg = 5 µs](image)

Figure 4.9: Average packet delay: comparison simulation – analytical model
4.4.2 Limited service

4.4.2.1 Analysis and simulations

It was shown in section 4.3.3.2 how the cycle time analysis is no longer valid for traffic loads near instability for small maximum transmission windows. Consequently, an analysis of packet delay, similar to the one for gated service (using again the cycle time distribution, but this time derived from the transition matrix with its dimension limited by the maximum cycle time), can be expected to fail in predicting packet delays for high traffic loads. However, for $P_{\text{max}} = 10$, such an analysis still gives good results for a large range of traffic loads (figure 4.11). From this simulation, however, one should not conclude that the analysis is correct. Indeed, if one observes the scenario of a much smaller transmission window, one sees that, the model is no longer correct for high traffic load (figure 4.12).

This can be explained as follows: the smaller the transmission window, the sooner instability is reached and the higher the probability that packets will not be sent with their first requested transmission window.

A reasonable idea seems to model the process of packets waiting for multiple cycles in the same way as for fixed service. However, in combination with the varying cycle times, this is no longer feasible. Nonetheless, the fixed and gated service analysis provide the insight to understand limited service qualitatively. The quantitative approach becomes too complex.
Limited Service

Figure 4.11: Average packet delay: comparison simulation – analytical model

Limited Service - Pmax = 3

Figure 4.12: Average packet delay: comparison simulation – analytical model
4.5 Conclusions

Constructing an analytical model proved to be far from straightforward. A good understanding of the cycle time turned out crucial. For fixed service, this cycle time is constant, which allowed an approach in which queue sizes were calculated at discrete moments in time, leading to a packet delay that corresponded to simulation. For gated service, a distribution of cycle times was derived, which proved very useful in determining the packet delay. For limited service, a similar analysis became infeasible, but the knowledge of the fixed and gated service provides qualitative insight.
5. Extensions and limitations of the model

5.1 Introduction

The analysis in the previous chapter clearly showed that, even under the assumption of symmetric traffic load (\(\lambda\) Mb/s for every ONU), Poisson arrivals and constant sized packets, an EPON with the IPACT protocol is a very complex system. This caused the analysis to be approximate for gated service. From this, it should be clear that a similar system with more complex assumptions concerning traffic sources, will not allow exact analysis either. Therefore, the aim of this chapter is to investigate for these other traffic sources the validity, possible extensions and limitations of the analytical model as it was established earlier (for fixed and gated service). First, asymmetric traffic load will be investigated, followed by Poisson arrivals with packets having a certain distribution of sizes. The last part of the chapter deals with self-similar traffic.

5.2 Asymmetric traffic load

5.2.1 Fixed service

For fixed service, the analysis remains valid. This is explained by the fact that the transmission window that an ONU is granted does not depend upon its own traffic load nor on the traffic load of the other ONUs. Remember that the entire derivation for average packet delay from section 4.4.1.1 took the point of view of the ONU. This was obvious from the use of the symbol \(\lambda\) (\(\Lambda\)), the aggregate traffic load, never appeared. Consequently, all formulas could (but will not) be repeated here. The graph in figure 5.1 compares between model and simulation and confirms the validity of the analysis. The parameters that are used here are the ones from table 3.1, except for the ONU arrival rates. These were chosen to be 55 Mb/s for all ONUs but one. The arrival rate for ONU 1 (the tagged ONU) was varied over its domain of stability (which remains unchanged), from 5 to 57.5 Mb/s.
5.2.2 Gated service

For gated service, the situation is more complicated and the analysis can not merely be repeated. Consider again the case of all ONUs having the same packet arrival rate of 55 Mb/s, except for one tagged ONU, for which the traffic charge is varied.

Recall the formula that allowed low and high traffic load to be distinguished (formula 4.7):

\[
1 - \sum_{k=0}^{p} \exp\left(-\frac{\Lambda}{B} T_{\text{cycle}}^{\text{min}}\right) \frac{\left(\frac{\Lambda}{B} T_{\text{cycle}}^{\text{min}}\right)^k}{k!}
\]

With P the maximum number of packets that can be sent in a minimum cycle time:

\[
P = \left( T_{\text{cycle}}^{\text{min}} - N \cdot (T_{\text{guard}} + 576 \cdot 10^{-6}) \right) \frac{R_T}{B + 304}
\]

If this probability was over 5%, the analysis for high traffic analysis was applied. This is clearly the case for the considered example, for which this probability of having more packet arrivals than can be sent in a minimum cycle (when the tagged ONU has no packet arrivals at all, hence 15 \cdot 55 \text{ Mb/s} in the following formula) is:
If the traffic load for the tagged ONU is increased, this probability will also increase, since the aggregate traffic load is calculated as:

\[ \Lambda = \lambda_i + 15 \cdot 55 \cdot 10^6 \]

Another formula that needs to be adapted is formula 4.8, used to estimate the minimum cycle time in the high traffic load analysis. It can reasonably be assumed to be:

\[ \overline{T_{cycle}} = \overline{T_{cycle}}^{min} + (1 + C) \cdot \frac{\Lambda}{N \cdot B} \cdot \frac{T_{cycle}^{min} B + 304}{R_U} \]

Furthermore, the analysis can just be repeated. Figure 5.2 shows a comparison between packet delay from simulations and analytically calculated delay. Even though serious approximations are made throughout the analysis, the results still show a fairly good prediction.

This last graph is clearly less steep than the one in figure 4.2. This can be explained by the fact that, as is also assumed in the analysis, it is the aggregate traffic load that determines the system’s behaviour. Varying only one ONU, even over a range of 100 Mb/s, does not affect the aggregate traffic load as much as varying all 16 ONUs traffic load over 10 Mb/s. In particular, in the lower part of the traffic load range for the tagged ONU (10 till 50 Mb/s), the average packet delay hardly increases.
5.3 Packet size distribution

So far, packets were always assumed to have a constant size (B bits). In this section, the possible extension of the analytical model to include packet size distribution will be investigated. It has been shown that packet size distribution is, generally speaking, trimodal in access networks [12]. The three modes correspond to the most frequent packet sizes. In addition to those (as in [11]), two other packet sizes will be considered here as well. Thus the packet size distribution is 64 bytes (47%), 300 bytes (5%), 588 bytes (15%), 1300 bytes (5%) and 1518 bytes (28%). The quoted packet sizes are for the data link layer. Recall, the packet size was previously assumed to be 1500 bytes (IP), which corresponds to 1518 bytes at data link layer. It was specified in chapter 2 that an 8 byte preamble and a 12 byte inter-frame gap form part of the EPON standard. This causes the service times to be equal to 0.672, 2.56, 4.864, 10.56 and 12.304 µs.

Firstly, the analytical model for fixed service will be discussed and secondly, an existing model for gated service is discussed and an attempt is made to extend the gated service model from chapter 4. For limited service, some general remarks will be given.

5.3.1 Fixed service

When dealing with fixed size packets, fixed service allows the easiest analysis. Since MPCP does not allow packet fragmentation, transmission windows were always naturally chosen to contain a whole number of packets (plus a REPORT message). Having mentioned this, it should be clear that the requirement of non-fragmentation complicates analysis when dealing with a packet size distribution. If a packet is bigger than the remainder of the transmission window, it will have to wait until the next cycle. An important consequence is that part of the transmission window will stay unused.

In [13], it was derived that the unused remainder R under high traffic load can be approximated as:

\[
E(R) = \frac{1}{E(X)} \sum_{r=1}^{n-1} r \cdot (1 - F_X(r))
\]

with B the maximum packet size, X the random variable representing the packet size and \(F_X(x)\) the cumulative packet size distribution. For the given packet size distribution, this leads to an unused remainder of 632 bytes.
An important consequence of the unused remainder is that the maximum available bandwidth per ONU drops by the following factor:

\[
\frac{W_{\text{max}}}{8} - \frac{E(R)}{8} - \frac{W_{\text{max}}}{8}
\]

For a maximum transmission window of 10 1500 byte IP packets, this factor is 95.9%; for a transmission window of 3 1500 byte IP packets it is 86.3%. This means the system already becomes instable for much lower traffic charge, or equivalently, the packet delay will be higher for the same traffic load. This is clear from figure 5.3, in which the simulated results for both constant size and variable size traffic are displayed. Another reason for the higher average delay is that with IP packets smaller than 1500 bytes, the Ethernet overhead becomes more significant, in percentage terms.

![Figure 5.3: Comparison between fixed service with constant size or variable size packets](image)

One could think of an extension where the unused remainder is filled up with small packets, but that is outside the scope of this study, which focuses on packet delay.
5.3.2 Gated service

5.3.2.1 Existing model

In [11], an attempt is made to model the packet delay for gated service for Poisson packet arrivals with a certain packet size distribution. This analytical model relies on *Queuing Analysis of Polling Systems* ([14]), according to which the EPON with IPACT can be classified as a cyclic server system with multiple queues. In [14], the gated service discipline is described as follows: the server serves only those messages that are queued at the moment of polling. Since this does not correspond completely to the functioning of IPACT, two-stage buffers need to be introduced (figure 5.4). Ethernet packets wait in the first stage buffer of the ONU, until this ONU is allowed to send all packets of the second stage buffer. This transmission is triggered by the arrival of the GATE message from the OLT. Upon this arrival, the gate of the first stage buffer is closed as well. At the end of the transmission window, the report message is sent to the OLT to report the total size of the packets that remain in front of the gate of the first stage. At the same time, those packets are advanced into the vacant space of the second stage buffer.

![Figure 5.4: Gated polling model of DBA scheme with two-stage buffer at an ONU [11]](image)

The proposed model is a continuous time queuing model. Packets are assumed to arrive according to a Poisson process with rate $\lambda [s^{-1}]$. The system is assumed to be symmetric in the sense that arrival rates and packet size distribution are the same for all ONUs. The detailed and extremely mathematical analysis can be found in [11]. Later in this section the necessary information will be provided and a shortcoming of the model will be discussed.
The guard time (or switchover time) between two ONUs is assumed to be independent and identically distributed, with a mean value equal to $r$ and second moment $r^{(2)}$. Furthermore, the following condition is assumed to guarantee stability:

$$\rho \equiv N \cdot \lambda \cdot b < 1$$

where $b$ is the mean packet length. If $C$ is a cycle time at steady state, then:

$$\rho = \frac{E[\text{service time during a cycle}]}{E[C]} = \frac{E[C] - N \cdot r}{E[C]}$$

From this relation the mean cycle time is obtained:

$$E[C] = \frac{N \cdot r}{1 - \rho} \quad (5.1)$$

**Comments**

A first remark is that formula (5.1) strongly resembles formula (4.6), the result of the simple cycle time analysis from chapter 4. It is in fact a simplified version: the time consumed by REPORT messages is neglected (or can be considered part of the switchover time); furthermore, the traffic load $\rho$ is written in terms of service time, instead of upstream bandwidth, but that is an equivalent formulation. Formula (4.6) is also more general in the sense that it considers bit rate at the network layer.

The analysis from section 4.3.2.1 turned out to be an oversimplification and the same applies to this analysis, for the following reasons: it is inherent to IPACT that a GATE can not be sent before the previous REPORT from the same ONU is received, causing the interval between successive grants to the same ONU to be at least the round-trip time to that ONU. This is because the grant needs information (requested window size) contained in the previous request [10]. Therefore, taking the average guard time equal to 5 $\mu$s, as is done in [11], causes the predicted cycle time to be much lower than the round-trip time. One could argue that increasing the guard time in the model would solve this problem, but this is not true. From the discussion in chapter 4, it should be clear that these switchover times depend on traffic load. In fact, the nomenclature guard time/switchover time is a source of confusion. The OLT is a cyclic server, but it serves its multiple queues (the ONUs) from a distance, which makes the approach based on [14] not suitable and even incorrect. Guard time, as the name suggests, is a necessity due to imperfections in synchronization and properties of transceivers, whereas the time between two transmission windows (certainly at low traffic load) will also be determined by the distance between OLT and ONU.

Consequently, the analysis from [11] underestimates packet delay, especially at low traffic loads.
5.3.2.2 Extending the model

An extension of the model to include the packet size distribution will prove to be more complex and approximate than the extension for asymmetric constant sized traffic from section 5.2.2. The analysis of chapter 4 considered the cycle time to take discrete values, this value being a consequence of the number of packets sent. This assumption was justified by the fact that the packet size was constant. With varying packet sizes, the cycle time can take many more values, it can range over a practically continuous domain if enough different packet sizes are considered. At first sight, this no longer allows the discrete analysis anymore. However, one could repeat the analysis with a packet of average size:

$$\overline{B} = \sum_i p_i \cdot B_i$$

where $p_i$ is the probability for having a packet of $B_i$ bits from the packet size distribution. Applying this formula to the distribution from the introduction leads to an average packet of 605.32 bytes (network layer). Lengthy calculations and simulations then lead to the graph from figure 5.5. The prediction is not bad, but not perfect either.

![Gated Service](image)

Figure 5.5: Comparison between model and analysis for gated service with variable sized packets
5.3.3 Limited service

It was mentioned in chapter 4 that the limited service discipline combines elements of fixed and gated service. It is therefore not surprising that the problem of the unused remainder (section 5.3.1) appears for limited service as well. The problem can be solved by making ONUs request not more than their maximum transmission window. That way, there will be no unused remainder.

5.4 Self-similar traffic

This section briefly deals with another type of traffic, called self-similar traffic. Appendix A.2 treats some of its basic properties and the way to generate it. Because of its complexity, the description that follows will be qualitative, rather than quantitative. One cannot find a closed form analytical expression due to the fact that for Pareto distribution the variance is infinite (even though the mean is still finite). It will be shown to what extent the system differs under this other type of traffic.

General observations

At simulation timescales, the traffic generated per ONU shows more fluctuations for self-similar traffic than for Poisson traffic. Whereas for Poisson traffic the ONU with the highest load out of the N ONUs (due to the stochastic nature of traffic) usually had the highest delay, for self-similar traffic there is no clear relation.

5.4.1 Fixed service

Figure 5.6 shows a trace for fixed service with a maximum cycle time of 2.0 ms and $T_g = 1.5 \, \mu s$. The pattern shown is much more irregular than in figure 4.5. For the last quarter of the trace the instantaneous traffic is clearly higher, causing the system to become temporarily unstable: there are much more bytes/packets arriving in one cycle than can be sent in the maximum transmission window. Because of this many more unpredictable behaviour, an analysis similar to the one from section 4.4.1.1 is no longer possible.
5.4.2 Gated service

Figure 5.7 shows how cycle times evolve, for both self-similar and constant size packet Poisson traffic, a symmetric traffic load of 40 Mb/s per ONU and $T_g = 1.5 \mu s$. Remember the analysis of cycle times for Poisson traffic in section 4.3.2.3 for high traffic load; it relied on the derivation of the cycle time distribution. It is clear from the figure that a similar approach is no longer possible for self-similar traffic. The graph for self-similar traffic shows spikes, due to the burstiness: for self-similar traffic the instantaneous traffic load is much higher during some periods of time than for Poisson traffic. Because of this, one can not easily capture the behaviour into formulas.
5.4.3 Conclusions

The previous discussion made clear that mathematical analysis is not feasible (with the same approach as before). One must turn to simulation. If one was to analyze IPACT with self-similar traffic into more detail and try to model it analytically, long simulations would have to be done, varying all parameters. An interesting first investigation could be to see how the behaviour varies for different Hurst parameters (or accordingly, different shape parameters, see formula A.1).

However, if it proves to be impossible to model IPACT with self-similar traffic analytically, one might as well turn to simulations with real traffic traces, instead of generating traffic.

5.5 General conclusion

Extending the fixed service and (to a certain extent) the gated service model was shown to be possible for asymmetric traffic. Including packet size distributions proved to be difficult for fixed service, due to an unused remainder of the transmission window. However, for gated service, fairly good results were obtained. For self-similar traffic, it was made clear that one can probably only rely on simulation, because analysis seems to become infeasible.
6. Differentiated services

6.1 Introduction

EPONs are essentially conceived to support differentiated services: voice communications, standard and high-definition video, video conferencing and data traffic. How this can be realized, is investigated and analyzed in this chapter. The starting point is a brief summary of the ONU architecture (using weighted round robin scheduling) as it was suggested in [3], and the first analytical model followed by a critical discussion thereof. The analytical model from chapter 4 will be helpful in understanding some of the shortcomings of this first model. In the second part of the chapter another ONU architecture, using priority scheduling, will be analyzed.

6.2 Weighted round robin scheduling

6.2.1. Architecture

The ONU architecture, as suggested in [3], introduces differentiated services by categorizing traffic at the ONU into three classes. Expedited forwarding (EF) is the highest priority class; it is delay sensitive and requires bandwidth guarantees. The medium priority class is called assured forwarding (AF); it requires bandwidth guarantees, but is not delay sensitive. The lowest priority class is best effort (BE), which is neither delay sensitive nor requires bandwidth guarantees.

The ONU consists of two stages (figure 6.1). The three queues in the first stage correspond to the different classes. Packets in these three queues are served in a weighted round robin (WRR) fashion and placed in the second stage queue. For the purpose of the WRR, each class is assigned a weight \( \varphi_x \) according to its priority \( P_x \) (\( x = \text{EF, AF or BE} \)):

\[
\varphi_x = \left[ \frac{\lambda_x}{\lambda} P_x \right]
\]  

(6.1)

Here \( \lambda_x \) is the arrival rate [bits/s] for class \( x \) at the ONU. The total arrival rate at the ONU \( \lambda \), is given by:
6.2.2 Analysis

The following assumptions are made in order to obtain analytical results:

- Packet arrivals are Poisson distributed.
- Packets have a fixed size of $B$ bits.
- Queues have infinite size, which is justified because the aim of the model is to examine packet delay, not packet loss.

At the first stage, priority is modelled by means of service time: the higher the priority of a class, the shorter its service time and, consequently, the higher its service rate. Service times are modelled as having an exponential distribution of parameter $\mu_x$. This allows the assigning of the ONU’s guaranteed bandwidth to its different queues, according to their priority. ONU i’s minimum guaranteed bandwidth $\Lambda_{MIN}^i$ can then be written as:

$$\Lambda_{MIN}^i = \sum_{x=1}^{3} \mu_x^i$$

Assuming all ONUs have the same service level agreement, this becomes:
\[ \Lambda_{MIN}^i = \Lambda_{MIN} = \sum_{x=1}^{3} \mu_x \]

Introducing the class weight, leads to:

\[ \Lambda_{MIN} = \sum_x \frac{\varphi_x}{\varphi_j} \mu = \mu \]

\[ \mu_x = \frac{\varphi_x}{\sum_j \varphi_j} \Lambda_{MIN} \]

The Poisson arrival process, the exponential service time, the FIFO service discipline and the infinite size of the buffer allow modelling every first stage queue as M/M/1\(^5\).

It is a known fact that the process of packets leaving an M/M/1 queue is also Poisson distributed (having the same rate as the input process, when the system is stable). Also, when adding different Poisson processes, one obtains a Poisson process, having as rate the sum of the different rates. Therefore, the process of packet arrivals at the second stage queue will also be Poisson distributed and, consequently, will have an arrival rate \( \lambda \), given by:

\[ \lambda = \sum_x \lambda_x \]

The OLT authorizes the ONU to send a transmission window having a size \( W_{MAX} \) [bits] at maximum. Since a group of packets is sent, the second stage’s service discipline is considered to be FIFO as well. The average time between two successive GATE messages is estimated to be \( T_{MAX}/2 \). The transmission window being 0 at minimum and \( W_{MAX} \) at maximum, the average transmission window is supposed to be \( W_{MAX}/2 \). The service rate of the second stage queue can thus be expressed as:

\[ \mu = \frac{W_{MAX}}{2} \frac{W_{MAX}}{T_{MAX}} = \frac{W_{MAX}}{T_{MAX}} \]  \hspace{1cm} (6.2)

This seems to make sense, because this is the ONU’s minimum guaranteed bandwidth. The service time can be considered to have an exponential distribution of parameter \( \mu \).

\(^5\) Kendall notation
Since the arrival process of the second stage queue is modelled as having a Poisson distribution, the service time considered exponential and packets are served in group, the second stage queue can be modelled as M/M^Y/1 (figure 6.2).

![Queuing network modelling the ONU](image)

In queuing systems terminology, a network such as this is referred to as an open BCMP\(^6\) network without change of class, or an open Jackson network. This system is stable under the assumption that for each ONU \(i\):

\[
\lambda_i < \mu_i
\]

which means the packet arrival rate is inferior to the service rate.

If \(p_i(n_i)\) denotes the probability that queue \(i\) holds \(n_i\) packets, then, from the properties of Jackson networks, it follows that the probability designating all four queues together can be written as a product:

\[
p(n_1, n_2, n_3, n_4) = \prod_{i=1}^{4} p_i(n_i)
\]

This means the average number of packets can be calculated for all four queues separately. The average number of packets for a first stage queue (M/M/1) is known to be:

\[
Q_i = \frac{\rho_i}{1 - \rho_i} \quad \text{with} \quad \rho_i = \frac{\lambda_i}{\mu_i}
\]

---

\(^6\) Named after Baskett, Chandy, Muntz and Palacios
The average number of packets in the second stage queue (M/M/Y/1), on the other hand, is equal to:

\[ Q_4 = \frac{r_0}{1 - r_0} \]

Where \( r_0 \) is the root, with modulus smaller than 1, of the following equation:

\[ (\mu \cdot p^\circ \rho_{\text{MAX}} + 1 - (\lambda + \mu)p^\circ + \lambda)p(n) = 0 \]

Here \( p^\circ \) is the operator, defined by \( p^\circ(p(n)) = p(n+1) \), that can be found to be:

\[ p(0) = 1 - r_0 \]
\[ p(n) = (1 - r_0)r_0^n \]

A more thorough analysis of M/M/1 and M/M/Y/1 can be found in the appendix of [3].

Little’s law then allows the average packet delay to be calculated as:

\[ \bar{W} = \frac{r_0}{\lambda(1 - r_0)} + \frac{\rho_{\text{EF}}}{\lambda(1 - \rho_{\text{EF}})} + \frac{\rho_{\text{AF}}}{\lambda(1 - \rho_{\text{AF}})} + \frac{\rho_{\text{BE}}}{\lambda(1 - \rho_{\text{BE}})} \]

### 6.2.3 Numerical example

Consider an EPON consisting of 16 ONUs, having the standardized upstream bandwidth of 1 Gb/s. The guard time is chosen to be 5 µs and \( T_{\text{MAX}} \) is 2 ms. All ONUs have the same service level agreement and can obtain a transmission window:

\[ W_{\text{MAX}} = R_U \left( \frac{T_{\text{MAX}}}{N} - T_{\text{guard}} \right) = 10^9 \left( \frac{2 \cdot 10^{-3}}{16} - 5 \cdot 10^{-6} \right) = 120000 \]

This corresponds to 10 packets of 1500 bytes. The minimum and the maximum obtainable bandwidth for an ONU can then be calculated as:

\[ \Lambda_{\text{MIN}} = \frac{120000}{2 \cdot 10^{-3}} = 60 \text{ Mb/s} \quad (6.3) \]
\[ \Lambda_{\text{MAX}} = \frac{120000}{16 \cdot 5 \cdot 10^{-6} + 120000/10^9} = 600 \text{ Mb/s} \quad (6.4) \]

\( \lambda \) is chosen to be 50 Mb/s (stable system), 20% of which is EF, 25% AF and 55% BE traffic.
If the priorities are chosen to be $Pr_{EF} = 5$, $Pr_{AF} = 3$ and $Pr_{BE} = 2$, the analysis leads to an expected number of packets of 2, 5 and 11 in the EF, AF and BE queue respectively, and 1 in the second stage queue. The average packet delay follows from Little’s law:

$$W = \frac{Q_1 + Q_2 + Q_3 + Q_4}{\lambda} = \frac{19}{4200} = 4.5 \text{ ms}$$

### 6.2.4 Comments

This section deals with shortcomings and imperfections of the architecture and its analytical model, described in the previous section. In the next section a possible solution (from [12]) will be formulated and analyzed.

- The ambition of the ONU architecture is to support differentiated services. Therefore traffic is categorized in three classes, one that requires bandwidth and delay guarantees (EF), one that only requires bandwidth guarantees (AF) and a last one that has no special requirements (BE). The weighted round robin turns out not to be the correct manner in which to fulfil those requirements.

The description of the WRR in the previous section was very concise; a more detailed explanation (as described in [15]) is necessary to see where the problems arise. The basic idea of the WRR is that each category gets a different amount of service time. Therefore each category obtains a weighing factor $\phi_i$. Packets from a certain category $i$ are then guaranteed to be sent during a certain part of time, equal to $\frac{\phi_i}{\sum \phi_j}$, where the sum in the formula contains all the classes that have packets ready to be sent. In the worst case (this is when all categories have packets ready to be sent), a category will still be guaranteed its part $\frac{\phi_i}{\sum \phi_j}$ of the bandwidth. For a link with transmission speed $R$, this means that each category will obtain a throughput capacity of at least $R \cdot \frac{\phi_i}{\sum \phi_j}$. This description is idealized, because it does not take into account the fact that packets are entities whose sending will not be interrupted.

Knowing this, a shortcoming of the model arises. In formula (6.1) the weighing factor is determined, based on the different $\lambda_x$. Normally, however, these weighting factors have to be
estimated in advance (which turns out to be difficult, because packet size distributions may vary). Also, \( \lambda \) is not a constant for real traffic. This would cause the weighting factors to vary over time, a difficulty that is not addressed in the model. If one would implement a WRR with those weighting factors, one will have to choose the time period over which the \( \lambda \) are averaged out. A consequence of this approach would be, for instance, that if no packets arrive for a certain category during this time period, the weighting factor becomes zero. Once packets start arriving again, it would take some time for the weighting factor to build up, which is undesirable because it will cause bigger delays. In fact, the difficulty of estimating the priorities \( P_x \) remains, a problem that is not addressed either. All this causes the model to lack causality: packet arrivals from a certain category should not determine the weighting factor for that category.

- Another remark about a WRR is that this scheduling algorithm is intended and more suited for links that are permanently available. Concretely, in the context of the EPON, different ONUs share upstream bandwidth. Each ONU is only allowed to send during a fraction of time, its transmission window. Whereas a WRR may lead to fair distribution of bandwidth between different queues that share a permanently available link, the case of the EPON is more complex. One could think of using the WRR for sending the packets during the period of the transmission window. Chapters 3 and 4 should make clear that this is not a good option: for instance, under low traffic load, transmission windows turned out to be very small, often only one packet (referring to the analysis and simulations with constant size packets). If another packet arrives shortly after the first one (the one for which the request was generated), the WRR makes it possible that this second packet takes the first packet’s place. This is undesirable, but not dramatic, since the packet is likely to be sent in the next window. However, for high traffic loads, transmission windows are still fairly small. It was shown in chapter 4 that transmission windows for 10 packets of 12000 bits caused the cycle time of an EPON containing 16 ONUs to be 2.0 ms, which is already fairly high. It is clear that a window for 10 packets is short, potentially too short, for applying a WRR and may not allow the right percentage of time to be assigned to each queue. On a scale of one cycle, WRR does not provide fair link sharing. This complicates the process of sending packets, because a mechanism should then be present to keep track of how much bandwidth each of the queues received over the past cycles. Also, for successive transmission windows and for a given ONU, it might be necessary to remember which queue was served last so that in the next cycle, packets from that same queue or the next queue could be sent first. Another option would be to always start sending packets from the highest
priority (EF) queue first. Again, it is clear that the use of a WRR makes the implementation of differentiated services more complex than necessary.

- The WRR, described in the previous point, is a process that is only activated during the transmission window. The queuing network, modelling the ONU, on the contrary, is a continuous process. Packets, categorized in the first stage queues are continuously served by the second stage queue. Priority is modelled by means of service times. This is an unusual approach: most often, in queuing theory, service times, having an exponential distribution of parameter $1/\mu$, are used to model the packet distribution. Apart from this, the fact that the service times for the different queues are independent, causing packet arrivals at the second stage queue to be Poisson distributed (being a sum of Poisson processes), lacks clarity. It makes it possible that (in the model) different queues are served at the same time. This feature does not reflect a real WRR and this approximation was not justified.

A WRR is work-conserving, by which is meant the scheduler is only idle when no packets are awaiting service. This characteristic is also not clearly dealt with in the analysis. In queuing systems literature, polling models can be found that describe the WRR in a more correct way (see [14]).

- As was said earlier, a WRR is suited for a permanently available link. However, if one was to consider a WRR which polls and serves the queues continuously, this would complicate the process. The question that should be answered then is at what speed the WRR should serve the first stage queues to put packets into the second stage queue. In the case of the EPON, the speed of the upstream link is not an option, because it is too high, which would leave packets practically in the same order as they arrived, thus not implementing differentiated services. Lowering the speed would solve this problem, but, on the other hand, it would increase packet delay and is therefore not a good option.

- The analysis of chapter 4 showed how complex the process of cycle times is. In this model, the assumption is made that the average time between two successive GATE messages is $T_{\text{MAX}}/2$ and transmission windows are estimated at $W_{\text{MAX}}/2$, regardless of the traffic load. It is clear that this assumption is an oversimplification of the problem. Cycle times and transmission windows strongly depend upon the traffic load.

Consider the numerical example of the EPON (with $T_{\text{guard}} = 5 \mu$s) from section 4.3.3.2: the average cycle time in simulations for 50 Mb/s traffic load per ONU turned out to be 482.7 $\mu$s,
less than a quarter of the maximum cycle time. This means the transmission window will also be approximately a quarter of the maximum transmission window on average.

The choice of these values for $T_{\text{MAX}}$ and $W_{\text{MAX}}$ is motivated by the fact that these values lead to the minimum guaranteed bandwidth. This argument does not make sense, because it could be extended to every pair of cycle times and transmission window having that same proportion (formula 6.2).

- Modelling the second stage queue as M/M$^Y$/1 may seem correct at first sight, but proves not to be the perfect model if one has a closer look (for details about M/M$^Y$/1, see [3]). The process of packets being sent in a batch of $Y$ packets at maximum does reflect the functioning of limited service IPACT; however, modelling the service time of the second stage queue as having an exponential distribution does not. In reality, the aggregated EPON traffic will determine when a transmission window is granted and a past state of the queue will determine the size of the transmission window. The transmission window will only be granted once a cycle, whereas M/M$^Y$/1 models serving the second stage queue continuously. Again one could argue that service time is usually used to model packet size distribution, which is not the case here.

- The model does not mention in what way reality and model differ, which makes it difficult to gain insight or understand to what extent the model is an approximation. For instance, the process of REPORT messages generating transmission windows with a certain time delay is not reflected in the model. Even though the time consumed by REPORT messages may be negligible, compared to the time used for sending data packets, one could expect them to be treated in some way by an analytical model describing IPACT.

- The calculation of $\Lambda_{\text{MIN}}$ (formula 6.3) does not take into account the REPORT messages or Ethernet overhead and consequently, slightly overestimates the minimum available bandwidth. Nonetheless, this gives a good first approximation of the bandwidth. On the contrary, it was shown in chapter 4 that maximum available bandwidth is somehow more difficult to calculate than in formula 6.4, due to the difficulty of the minimum cycle time.

- It was shown that the distance between ONU and OLTs influence cycle times and consequently, packet delays especially at low traffic load. This is not dealt with in this analytical model.
• A last and important remark deals with the results that follow from the analytical model. Average packet delay is calculated to be 4.5 ms. This is the average packet delay for all classes. One would expect the analysis to give different results for different classes. The analytical model should at least be extended to include this distinction. Even then, results would still be erroneous. To understand this, compare the 4.5 ms to the 0.73 ms obtained from simulations in chapter 4. Introducing differentiated services should not cause packet delays to increase on average, certainly not by a factor of approximately 6. Also, the values for the priorities, chosen for the WRR, seem to be rather random. In reality, they should be based on knowledge of network traffic.

Conclusion
From the previous remarks, it is clear that the original model has unsolvable shortcomings. The main reason is that the model tries to apply classical queuing theory too literally: it tries to cram the complex system of the EPON with IPACT into the framework of Jackson networks. However, Jackson networks do not manage to capture all the features of the system. Therefore, the mathematical description does not reflect reality, which means that, even though derivations are completely correct in a mathematical way, the description will not provide real insight into the matter, nor correct results. Another problem is that the analysis tries to model too much at once: IPACT at itself is already very complex, differentiated services left aside. An approach which is more likely to be successful, is to begin by analyzing simpler cases and to verify (by means of simulation) if they are valid.

6.3 Priority scheduling

6.3.1 Problem statement
From the previous section, it should be clear that WRR at ONU level is not a good solution for implementing differentiated services for the EPON. An important reason that was not stated in the previous section is that the WRR is not compliant with the IEEE 802.1D standard. This condition should be fulfilled because MPCP is part of the IEEE 802 family of standards. According to this standard, intra-ONU scheduling is by default strict priority scheduling [12].
In this section, the necessary information about the standard will be provided, followed by a description of how it can be applied to the EPON. After that, the system will be analyzed. The analysis will rely on the analytical model of chapter 4.

### 6.3.2 Standard requirements

The 802.1 standard specifies that by default for a certain station, packets corresponding to a certain class will only be sent if no packets are present in queues corresponding to higher priority classes. The classes of traffic that are mentioned in this paragraph are also defined in the standard. Table 6.1 gives an overview of the classes, as well as a recommendation of how different classes should be grouped, if a station has less than seven queues. From this table, one can conclude that for ONUs with three priority queues (as in the original model) the traffic types would be mapped to the following classes:

- Expedited Forwarding: network control and voice
- Assured Forwarding: video and controlled load
- Best Effort: excellent effort, best effort and background

<table>
<thead>
<tr>
<th>Number of queues</th>
<th>Traffic types queue assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Network Control</td>
</tr>
<tr>
<td>2</td>
<td>Network Control</td>
</tr>
<tr>
<td>3</td>
<td>Network Control</td>
</tr>
<tr>
<td>4</td>
<td>Network Control</td>
</tr>
<tr>
<td>5</td>
<td>Network Control</td>
</tr>
<tr>
<td>6</td>
<td>Network Control</td>
</tr>
<tr>
<td>7</td>
<td>Network Control</td>
</tr>
</tbody>
</table>

Table 6.1: Grouping traffic classes into priority queues [12]
6.3.3 ONU architecture

An architecture for implementing differentiated services, using priority scheduling (suggested in [12]), is shown in figure 6.3. It is very similar to the architecture from section 6.2: it has two stages; the first stage consists of priority queues, the second stage consists of one FCFS queue. When the ONU’s transmission window starts, packets from the second stage queue are sent. At the same time, packets present in the second stage are transferred to the first stage (higher priority packets being served before lower priority packets). They will determine the REPORT message that is generated at the end of the transmission window. The second stage buffer can have exactly the size of the maximum transmission window.

![Figure 6.3: ONU with two-stage queue][12]

6.3.4 Analysis

The implementation of differentiated services only works in combination with fixed and limited service. For gated service, packets are sure to be sent in their first requested transmission window. Therefore, priority scheduling would only put higher priority packets up front in the window, but this does not cause a big difference in quality of service between the different priority types.

6.3.4.1 Fixed service

Remember the packet delay analysis from section 4.4.1.1. The analysis for differentiated services follows logically from it. First class packets (EF) are the first ones to be served in the transmission window. Therefore, the probability for an EF packet not to be sent in the first window is, as a good approximation, equal to the probability of having more than \( P_{\text{max}} \) packet arrivals for this class in the fixed cycle time:

[12]: #
\[
\Pr(N_{EF}^{T_{cycle}} > P_{\max}) = 1 - \sum_{k=0}^{P_{\max}} \exp\left(-\frac{\lambda_{EF} T_{cycle}}{B}\right) \frac{\left(\frac{\lambda_{EF}}{B} T_{cycle}\right)^k}{k!}
\]

For the cases that were considered earlier, for which \(P_{\max}\) was equal to 10 or 3, \(T_{\text{guard}} = 1.5 \mu s\) and \(\lambda_{EF} = 10 \text{ Mb/s}\), these probabilities are calculated to be 0.00015 % and 0.2 %.

The reasoning from above can be extended to for the medium priority class (AF). The probability for an AF packet not to be sent in its first window is approximately equal to the probability of having more than \(P_{\max}\) arrivals for the EF and AF classes together in the fixed cycle time:

\[
\Pr(N_{EF}^{T_{cycle}} + N_{AF}^{T_{cycle}} > P_{\max}) = 1 - \sum_{k=0}^{P_{\max}} \exp\left(-\frac{\lambda_{EF} + \lambda_{AF}}{B} T_{cycle}\right) \frac{\left(\frac{\lambda_{EF} + \lambda_{AF}}{B} T_{cycle}\right)^k}{k!}
\]

For the same parameters as above and \(\lambda_{AF}=12.5 \text{ Mb/s}\), these probabilities are now found to be 0.18 % and 3.1 %.

The reasoning can be extended to any number of classes. One might wonder why the results were said to hold only approximately. That is because this analysis does not take into account that packets can stay in a queue for several cycles. Note that this is only likely to happen to the lower priority packets, or to high priority packets if they make up the major part of traffic. If one wants take this into consideration, a more complex analysis, similar to the one from chapter 4, becomes necessary.

### 6.3.4.2 Limited service

It was shown in chapter 4 that for fixed service accurate analysis was no longer possible. However, since for limited service the cycle time is at maximum equal to the constant cycle time for fixed service, the above formulas serve as upper limits. This is because, evidently, the expected number of packet arrivals within a cycle time increases with the cycle time.

All formulas followed from Poisson properties, but it is clear that the same mechanism of prioritizing works just as well for other traffic. Consider for instance telephone traffic, which is even more predictable than Poisson traffic. From the standard requirements from table 6.1 it follows that voice traffic will always acquire high priority, which means it is sure to be sent in its first requested transmission window. For burstier traffic, the probability of packets being delayed for several cycles becomes higher, but analytical modelling is then no longer feasible.
7. The Future of EPON

A thesis would not be complete without a glance at the future. Therefore, this chapter treats, if only briefly, some of EPON’s perspectives. With ever-increasing bandwidth demands, it can be expected that at some point the bandwidths provided by the current EPON standard will become a limiting factor. It was already mentioned in chapter 1 that EPONs (and by extension, all PONs) allow for easy upgrades: the PON fibre infrastructure is transparent to bit rate and packet format, which means upgrades do not require replacing components in the field.

Several upgrade scenarios have already been suggested, but remain cost-prohibitive with the current state of technology. Therefore, a logical and cost-effective scenario is one in which a subset of users is migrated to a separate EPON. Other solutions, discussed into more detail in the subsequent paragraphs, are an increase of the bit rate and the use of multiple wavelengths.

7.1 10 Gb/s EPON

Very recently, on March 9th 2006, a 10 Gb/s EPON study group was formed within the IEEE 802.3 working group. Several requirements have been suggested and criteria have been set. They will be briefly discussed here. More details can be found at the study group’s website at [16].

Their first criterion is a broad market potential. The following list of new bandwidth-intensive applications shows that the 1 Gb/s EPON will indeed prove to be insufficient and a more advanced standard is desired: broadcast TV (HDTV), IPTV, VOD, 3D online gaming, ultrahigh speed internet, business Ethernet access, virtualized multimedia network applications, medical imaging, video conferencing, video email, etc.

Furthermore, since 1 Gb/s EPONs have already been deployed, a migration scenario should be considered. This means that distance limits between OLT and ONUs should be chosen to be the same as in the existing standard and existing fibre plant characteristics should be considered for a new standard. In this way, 1 Gb/s and 10 Gb/s EPONs could coexist, allowing subscribers to upgrade at different points in time. Apart from the symmetric option (10 Gb/s upstream and downstream) an asymmetric variant (10 Gb/s downstream and 1 Gb/s upstream) has been studied. As with the 1 Gb/s EPON, MPCP can be used for bandwidth assignment.
Then there is the question of economical feasibility. Because 10 Gb/s Ethernet and 1 Gb/s EPON are widely deployed for commercial services, cost factors are well known. 10 Gb/s can be shared by many subscribers, which results in a reduction of infrastructure cost. The PON architecture is well suited for broadcasting services, so IP-based TV, one of the main drivers in the broadband market, can be delivered cost-effectively. The installation costs of the optical distribution network will be the same for 10 Gb/s as for 1 Gb/s.

7.2 WDM upgrade

Another technology that will allow much higher bandwidths to be supported is the incorporation of wavelength-division multiplexing (WDM), in place of just one wavelength upstream and a separate one downstream. The idea of a WDM-PON is not new. Architectures had already been proposed already ten years ago, yet no standardization exists. The fact that they have not been deployed is obvious: there was no demand for such high bandwidths nor was the technology mature enough to provide the required components.

7.2.1 WDM-PON architectures

There are numerous variations of WDM-PON. The simplest approach employs a separate wavelength channel from the OLT to each ONU, for each of the upstream and downstream directions. In this way, a point-to-point link is created between the OLT and each ONU. In the downstream direction an arrayed waveguide grating (AWG) replaces the passive splitter and routes the different wavelengths. The drawbacks of this first approach are its cost and the fact that whenever an ONU is idle its wavelength cannot be used. The Composite PON (CPON) is more economical because it only uses one wavelength upstream, and so is the LARNET (Local Access Router Network) architecture, because it uses a cheap broad-spectrum source (such as an LED) at the ONU and slices its spectrum. RITENET (Remote Interrogation of Terminal Network) avoids the transmitter at the ONU by modulating the OLT’s downstream signal before sending it back upstream. The Super-PON can cover a range of over 100 km with a splitting ratio of 2000 with the aid of optical amplifiers. One final architecture is the Stanford University Access Dynamic Wavelength Assignment (SUCCESS-DWA) PON. As its name suggests, it offers scalability by dynamically assigning wavelengths, amongst multiple physical ONUs. More detailed descriptions and a comparison between these architectures can be found in [17].
7.2.2 A gradual WDM upgrade for EPON

From the previous section, it is clear that there are many options for a WDM upgrade. However, all architectures mentioned before differ (some only slightly, others to a greater extent) from the current standardized EPON architecture, making them less suitable for an evolutionary WDM upgrade. They are not backward-compatible. A method for a gradual upgrade in EPONs was suggested in [18] and will be discussed in this section.

The WDM upgrade for the EPON requires an array of fixed transceivers at the OLT, thus making a larger part of the optical spectrum available. The ONUs consist of either an array of fixed transceivers or one or more tunable transceivers. No particular WDM architecture is imposed on the ONUs, so current ONUs (with fixed wavelengths) are still supported and WDM capabilities can be added on an evolutionary basis. This means the OLT should know for each ONU which architecture it has and which wavelength it supports. This information can be exchanged during the discovery and registration process.

For a WDM PON, the OLT must now assign bandwidth in two dimensions (time and wavelength). In addition to the start time of the transmission window, the OLT now also has to convey to the ONU which of its supported wavelengths will be used for transmission and reception. This means MPCP must be extended in order to be able to communicate ONU WDM architectures and coordinate upstream and downstream transmission.

As for the existing standardized version of MPCP, different algorithms could be considered for dynamic bandwidth assignment. TDM could be separate from the wavelength assignment or could be combined with it. WDM IPACT falls under this last category. The round robin nature of the original IPACT algorithm is kept intact, but now an ONU does not have to wait for the completion of the previous ONU’s transmission window. Instead it can be served by another wavelength.
8. Conclusions

Any master’s thesis is subject to change, and this one makes no exception: what originally started as a techno-economical study, investigating the deployment of fibre to the home in Europe, evolved to merely technical research and analysis at protocol level.

An internship at Telenet Mechelen and an extensive literature study provided me with a clear view on the technical possibilities and different implementations for the access. Even though this knowledge only finds itself under extremely condensed form in this final thesis book, its importance can not be overestimated, because it has shown the reality as it is: economical considerations most often dominate; the best solution, technically speaking, is seldom the cheapest and therefore not the most suitable in the short term.

Most of this thesis book reflects the research done in the domain of one out of several promising solutions for the access, i.e. the Ethernet PON and, in particular, its protocol MPCP. The actual starting point was an analytical model, developed at ENST Paris, describing packet delay for MPCP/IPACT, a dynamic bandwidth allocation algorithm, the results of which were to be checked by simulation. The model was briefly summarized in chapter 6, where some of its shortcomings were also discussed in detail.

In order to be able to establish an improved analytical model, the study domain of queuing systems was explored and existing polling models were examined. Because none of these managed to capture the specific EPON/IPACT details, a model then had to be built from the ground up. Observations from traces were analyzed in detail in chapter 3, which formed the basis of the analytical models proposed in chapter 4.

Analytical modelling IPACT turned out to be far from straightforward. The importance of a good understanding of the cycle time was shown. For fixed service, an approach was suggested in which queue sizes are considered at discrete moments in time, resulting in an analytical packet delay that corresponded to simulation. For gated service, a distribution of cycle times was derived. This proved useful when determining packet delay. Again simulation and analysis matched well. For fixed service, it was shown how a similar analysis is not feasible. However, knowledge of fixed and gated service provides additional insight.
Some general remarks about the analytical model: it is an engineer’s model, its main goal and achievement is to provide insight into the mechanisms that determine delay. Striving for a more mathematically polished model would have meant neglecting certain features. It would have resulted in a discrepancy between model and reality or, in this case, simulation. It may be clear that this approach increased complexity, compared to fitting IPACT into an existing polling model.

In chapter 5, the model was investigated to see if and how it could be extended to include asymmetric traffic and packet size distribution. This also raised the key question of to what extent mathematics can capture reality, and when the quantitative analysis should be replaced by just qualitative analysis. It was then demonstrated how, at a certain point, a mathematical analysis becomes infeasible and needs to be replaced by simulation.

Since a thesis would be incomplete without having a look at the future, chapter 7 briefly introduced EPON perspectives, i.e. how EPON will allow transition to even higher bandwidth in access networks in a more distant future.

As a general conclusion, this thesis broadened my perspective of the multidisciplinary telecom world and gave me general knowledge of the field the as well as some very specific skills in analytical modelling.
Appendix A: Traffic sources

A.1 Poisson traffic

In the context of queuing systems, the Poisson process is the most frequently used model for packet arrivals, because it is the only one that allows real analysis. In this section, some basic properties, used throughout the derivations, will be repeated.

Packet inter-arrival times follow an exponential law. This means that, for constant sized packets (B bits), the distribution function of the inter-arrival time is given by:

$$F(t) = 1 - e^{-\frac{\lambda}{B}}$$

where \( \lambda \) is the arrival rate [bits/s].

The probability of having \( k \) packet arrivals in a time \( t \) is:

$$P(N_t = k) = \exp\left(-\frac{\lambda}{B}t\right)\frac{\left(\frac{\lambda}{B}t\right)^k}{k!}$$

These arrivals are uniformly distributed over the interval, by which is meant that there are no bursts. Obviously, this does not mean that inter-arrival times are approximately constant.

The expected number of packet arrivals in an interval of length \( t \) is:

$$E(N_t) = \sum_{k=0}^{\infty} k \cdot P(N_t = k) = \frac{\lambda}{B} t$$
A.2 Self-similar traffic

Self-similarity describes the phenomenon where a certain property is preserved with respect to scaling in space or, as is here the case, in time. This section introduces basic properties of self-similar traffic, the traffic that was used for the simulations in section 5.4 and the method to generate this traffic type. A much more complete overview can be found in [19].

A.2.1 Properties

Consider $X(t)$ to be a second-order stationary discrete time process, interpreted as the traffic volume at a time instant $t$. The aggregated process $X^{(m)}$ can then be defined as:

$$X^{(m)}(i) = \frac{1}{m} \sum_{t=m(i-1)+1}^{mi} X(t)$$

and its auto-covariance function as:

$$\gamma^{(m)}(k) = E[(X^{(m)}(t) - \mu)(X^{(m)}(t+k) - \mu)]$$

with $\mu = E(X(t))$.

The type of self-similarity that is usually considered, regarding network traffic, is the second order self-similarity. It requires:

$$\gamma^{(m)}(k) = \gamma(k) \quad \text{for all } m \geq 1$$

If this equality only holds in the limit for $m$ going to infinity, the process is called asymptotically second-order self-similar.

An important parameter, called the Hurst parameter $H$ (or self-similarity parameter) describes the variance of the aggregated process as:

$$\text{var}(X^{(m)}) = \sigma^2 \cdot m^{2H-2}$$

where $\sigma$ is the variance of $X(t)$. When restricting $H$ to the domain $\frac{1}{2} < H < 1$, this parameter describes how fast the variance decays when aggregating traffic over time.
A.2.2 Traffic generation

Self-similar traffic can be generated by the aggregation of several sources having Pareto-distributed ON/OFF periods.

The Pareto distribution function is given by:

$$\Pr(Z \leq x) = 1 - \left( \frac{b}{x} \right)^\alpha, \quad b \leq x$$

with $b$ the location parameter and $\alpha$ the shape parameter. In the network context, the following holds: $1 < \alpha < 2$.

The following relation exists between the Hurst parameter and the shape parameter:

$$H = \frac{3 - \alpha}{2} \quad (A.1)$$
Appendix B: Simulations and calculations

In the context of analytical modelling, simulations serve several purposes: on the one hand, their results can accord a degree of correctness to an established model; on the other hand, they can be the starting point for analytical modelling, providing a basis for a frame of mind, because throughout simulations, questions that may not be obvious at first sight can arise. Whereas an ad hoc analytical model risks to oversimplify a problem, this danger decreases when using simulations.

This section briefly deals with how the simulations have been performed, first to check an existing analytical model (discussed in chapter 6), later to observe the process into detail (chapter 3) and establish new models (described in chapter 4) and their extensions (chapter 5). An overview will be given of the software that was used and the functionality that was added to it. Also the method of doing calculations (for the analytical model) will be briefly discussed.

B.1 Simulator

There are several software tools available for simulating networks. Working with OPNET turned out to be unfeasible due to restrictions in extending the student version with the necessary functionality. Therefore, it was decided to work with NS-2, which does not have this problem because it is open source code. Programming was done in OTcl, allowing faster implementation. An elaborate manual for NS-2 can be found at [20]. A more comprehensible manual is found at [21].

In the next sections, traffic generation, the implementation of IPACT on the EPON, and the output and different traces that were produced will be briefly treated.

B.1.1 Input

Poisson traffic with fixed size packets
Some of the properties of Poisson traffic are repeated in appendix A.1. The easiest way to generate Poisson arrivals is by successively drawing random variables having an exponential distribution. This
distribution only has one parameter, whose inverse is the average of the variable. Therefore, in order to obtain a bit rate of $\lambda$ [bits/s], the average inter-arrival time needs to be set to $\frac{B}{\lambda}$ [s]. The packet size $B$ was chosen to be 1500 bytes (as in the original model [3]), which corresponds to an IP packet with maximum size. The script generator uses this method to generate a Poisson traffic trace. The random generator requires special attention, as it must be seeded correctly in order to obtain truly random results.

**Poisson traffic with a packet size distribution**

The generation of Poisson packet arrivals with a certain packet size distribution is similar to the previous case. If a packet size $B_i$ occurs with a probability $p_i$, the average of the inter-arrival time needs to be set to $\frac{\sum_i p_i \cdot B_i}{\lambda}$. The script generator_var generates this type of traffic.

**Self-similar traffic generation**

The necessary properties of self-similar traffic are treated in Appendix A.2. There it is also explained how this traffic type can be generated by aggregating Pareto ON/OFF sources. An existing program (obtained from [22]) was used to generate self-similar traces. Some small changes were performed in order to obtain a traffic trace of the same format as the Poisson traces.

**B.1.2 EPON with IPACT**

**Choice of parameters**

The parameters were chosen according to the standard: there are 16 ONUs connected to a node representing the passive optical splitter by 1 Gb/s links with a 0.05 ms delay. The passive optical splitter at its turn is connected by a 1 Gb/s link with 0.05 ms delay to the OLT. This makes a total delay of 0.1 ms between ONU and OLTs (one way). For optical fibre, this delay corresponds to a distance of 20 km between all ONUs and the OLT. This constitutes the worst case scenario in terms of delay, and is therefore the most interesting case to study.

The delay that occurs at the ONU between the reception of a GATE message and the actual start of the transmission window is not negligible and is estimated at 17.5 µs. The delay at the OLT between the reception of a REPORT message and the sending of a GATE based on this REPORT is chosen the same value.
Furthermore, when a packet is sent, 26 bytes of Ethernet overhead are added and the 12 byte inter-frame gap is respected. A REPORT message consumes 72 bytes (64 bytes plus the preamble). In between transmission windows, the guard time was chosen as 5 µs or 1.5 µs.

**IPACT**

There are several scripts, implementing fixed, gated and limited service. All scripts take ONUi as input traces (i goes from 1 to 16) and generate several output traces, treated in the next section. The first versions handle packets of constant size (1500 bytes), the later versions handle packets of various sizes as well. These scripts also implement a procedure that monitors the queues (treated in the next section).

The IPACT procedure serves the ONUs in a round-robin fashion, allowing an ONU to send during its transmission window, according to the implemented service discipline.

For the fixed service discipline, transmission windows are constant, which means the time between the start of successive transmission windows is also constant and equal to:

\[
\frac{P_{\text{max}} \cdot (B + 304) + B_{\text{req}}}{R_{\text{U}}} + T_{\text{guard}}
\]

For instance, for the EPON with a transmission window of 10 maximum size IP packets (1500 bytes) this gives 125 µs. Whenever an ONU’s transmission window expires, the procedure switches to the next ONU, which can then start sending the packets that are in its queue, until its transmission window ends.

For gated service, the number of packets/bytes that is left in the queue at the end of a transmission window is saved, as well as the end time of the transmission window, and the procedure switches to the next ONU. The start time of the next ONU’s transmission window will then be the end of the previous ONU’s window plus the guard time. In case this start time falls before a GATE message could have arrived, the knowledge of the end time of the ONU’s transmission window allows its start to be postponed exactly long enough. The ONU will then be allowed to send the number of packets that was saved previously.

For fixed service, apart from the fact that no more packets/bytes can be sent than a specified maximum, the functioning is the same.
The names of these scripts are self-explanatory: epon_16ONU_fixed, epon16ONU_gated, epon16ONU_limited, epon16ONU_fixed_var, epon16ONU_gated_var and epon16ONU_limited_var.

B.1.3 Output

One could argue that a program that outputs the average delay, given the input traces for a certain traffic load, is sufficient for checking the correctness of an analytical model. This may well be true, but if there is a discrepancy between model and simulation, such an approach does not allow the locating of errors. For the purpose of constructing an analytical model, the average delay itself does not add much: delay can only be expected to increase with increasing traffic load. For this reason, various forms of output have been generated. They will be discussed briefly throughout this section.

NAM

NAM (Network Animator) is an animation tool for viewing network simulation traces and real world packet traces. It is mainly intended as a companion animator to the ns simulator. Figure 2.1 shows an animation of an EPON with 16 ONUs. This way of visualizing is more helpful than numerical traces: it gives an idea of how many packets an ONU sends in a cycle and how variable this number is. Apart from this, it is also a good check for the simulation’s correctness.
Cycle time traces
Cycle times play a prominent role in the analysis of chapter 4. The inspiration for this approach was the observation that cycle times show much more variation than one might expect (a detailed discussion hereof was given in chapter 3). The trace that is generated during a simulation is the sequence of cycle times [s] for a fixed ONU, chosen to be ONU1. The names given to the traces are self-explaining (ACT stands for Adaptive Cycle Time): $ACT_{\text{fixed}}$, $ACT_{\text{gated}}$ and $ACT_{\text{limited}}$.

Packet delay traces
The ultimate focus of the analysis is on packet delay and this is true for the simulation as well. Therefore, several traces are generated (one per ONU), giving all packet delays. This allows the average packet delay to be calculated as well as the evolution of packet delay over time to be followed (for some examples, see chapter 3). In order to obtain the best results, one can average over all ONUs (in case of symmetric traffic), this way minimizing statistic variations as much as possible. The names given to the traces are: $delay_{\text{ONU}i \text{fixed}}$, $delay_{\text{ONU}i \text{gated}}$ and $delay_{\text{ONU}i \text{limited}}$ where $i$ is the number of the ONU.

Monitoring queue sizes
Since simulations are lengthy, it is a good idea to monitor the process in some way during execution, to give an idea of the progress. The most suitable parameter is the queue size. Since queue sizes vary
quickly over time, the time average value is calculated and this for each ONU, as well as the average over all ONUs. These averages allow to be kept track of the convergence (or divergence, for an unstable system).

Apart from the fact that they show how the system evolves, the average queue sizes also allow the easy calculation of average packet delay for constant size packets, thanks to Little’s law, which states that the average number of customers in a stable system (over some time interval) is equal to their average arrival rate, multiplied by their average time in the system. A mathematical formulation is:

\[ \bar{W} = \frac{\bar{Q}}{\lambda} \]

Therefore, knowledge of the average queue size \( \bar{Q} \) and the arrival rate \( \lambda \) is equivalent to knowledge of average packet delay \( \bar{W} \) (for a stable system). In order to obtain good results, the queue size needs to be monitored at a sufficiently high rate. \( 10^5 \text{ s}^{-1} \) is more than high enough (\( 10^{-5} \text{ s} \) is of the same order as a packet’s duration at 1 Gb/s).

**B.2 Calculations**

The analysis relies on numerical matrix calculations. The analytical model in chapter 4 is symbolic. The formulas have been used in Maple to produce analytical results for different parameters, allowing comparison with the results obtained from simulation. The most important Maple files, used for calculating packet delay for fixed and gated service, are *Fixed_calculation* and *Gated_calculation*. 
References


