Technical University of Denmark

Energy Efficiency in Future PONs

Reschat, Halfdan; Laustsen, Johannes Russell ; Wessing, Henrik; Ruepp, Sarah Renée; Yan, Ying

Published in:
Proceedings of OPNETWORK2012

Publication date:
2012

Energy Efficiency in Future PONs

Halfdan Reschat, Johannes Russell Laustsen, Henrik Wessing, Sarah Renée Ruepp, and Ying Yan
Department of Photonics Engineering, Technical University of Denmark
Kongens Lyngby, Denmark, 2012
E-mail: {s072222,s072226}@student.dtu.dk, {hewe,yiya,srru}@fotonik.dtu.dk

Abstract

There is a still increasing tendency to give energy efficiency a high priority, even in already low energy demanding systems. This is also the case for *Passive Optical Networks* (PONs) for which many different methods for saving energy are proposed. This paper uses simulations to evaluate three proposed power saving solutions for PONs which use sleep mechanisms for saving power. The discovered advantages and disadvantages of these methods are then used as a basis for proposing a new solution combining different techniques in order to increase the energy efficiency further. This novel solution is also presented in this paper.

All simulations are done using OPNET Modeler where an EPON model is implemented with the necessary mechanisms for simulating basic functionality and for expanding the model with sleep methods. The individual sleep methods are implemented on top of the basic EPON model, making it easy to switch between them and to let each method seamlessly be an integrated part of the EPON model without affecting other sleep methods or the normal EPON functionality. For simulating different kinds of approximately realistic traffic patterns, traffic generators are implemented in an easy interchangeable and customizable way, making it possible to simulate both generic traffic following general traffic patterns and individual types of traffic following documented real life traffic patterns.

Introduction

Energy efficiency is becoming more and more important on a worldwide scale, with e.g. the European Union having the goal of reducing their projected energy consumption by 20 % by 2020 [1]. On a global scale, the information and communication technology industry is estimated to be responsible for 2 % of all CO_2 emissions [2]. This is to a large degree due to the access networks, which are the places where most of the energy is used [3]. As the number of broadband subscribers increase on average more than 20 % per year from 2005 to 2010 [4], the access network is a very relevant area to strive towards making more energy efficient.

PONs have for a while been one of the preferred choices when deploying new access networks, and though DSL still is the most used access network technology, PONs are becoming more and more used [5]. It is for this reason becoming increasingly relevant to optimize the energy efficiency of PONs even though PON already is the most energy efficient access network technology [3]. Improving the energy efficiency of PONs helps reduce the emission of greenhouse gasses which contributes to saving the environment and also shows corporate social responsibility for the Internet service providers which can be a good selling point.

This paper is focused on *Time Division Multiplexed* (TDM) PONs, which comprise an *Optical Line Terminal* (OLT) at the service provider's central office and an *Optical Network Unit* (ONU) at each endpoint near the end-users. A single fiber is used to transmit the data and a passive optical splitter is used to split the signal to distribute it to all the ONUs. In the downlink direction the OLT broadcasts the data, which then is analyzed by every ONU to determine if the data is intended for it. All other ONUs discards the data. This means that all the ONUs' receivers must be on all the time, even when the data that they are receiving are not for them. In the uplink direction the ONUs are assigned timeslots in which they are allowed to transmit. The method of uplink bandwidth allocation depends on the standard, of which the most common currently are *Ethernet PON* (EPON) and *Gigabit-capable PON* (GPON). In both cases the uplink transmission is controlled by the OLT.

This paper presents and evaluates three different proposed sleep methods through modeling of an EPON system in OPNET Modeler [6]; along with the fourth *novel sleep method*. The sleep methods are implemented in an integrated, though easily interchangeable, way with the EPON model with traffic generators implemented, generating both individual type traffic and generic traffic.

The remainder of this paper is organized as follows: In section II the three previously proposed sleep methods are described. In section III the design and implementation of the EPON model with the traffic generators is described, followed by the design and implementation of the three sleep methods in section IV and the description of the *novel sleep method* in section V. The simulation results of all sleep methods are presented and evaluated in section VI. Finally, the conclusion is given in section VII.

ONU Sleep Methods

The way the PON sleep methods are saving power is by turning off all non-essential electronics of the sleeping ONUs in periods with absence of traffic; meaning that incoming uplink traffic to the ONU still is observed while all the downlink electronics and the optical transmitter and receiver are turned off. There are several different methods proposed for doing this, whereof the three methods selected in this paper are described in general terms in this section, all suitable for both EPON and GPON.

The ***first sleep method*** of *Sleep and Periodic Wake-up* (SPW) is proposed by R. Kubo et al. in [7]. Since the distribution of downlink capacity is done entirely by the OLT, according to the individual ONU downlink traffic flows, the SPW mechanism is able to determine when an absence of downlink traffic can be expected for a specific ONU. This knowledge is then used by the OLT to instruct the ONU without traffic to sleep for a calculated amount of time, where the absence or presence of traffic and the sleep amount are determined based upon some predefined parameters and the *interval-based algorithm*:

$$i_{a,k} = \alpha i_{k-1} + (1-\alpha) i_{a,k-1}$$

This algorithm is executed on each downlink packet arrival and uses exponential smoothing to calculate a weighted average interframe period ($i_{a,k}$) using the last downlink interframe period (i_{k-1}), the last average interframe period ($i_{a,k-1}$), and a smoothing factor (α). The ONU being requested to sleep decides to accept the sleep request if it does not have uplink traffic queued and otherwise rejects the sleep request. After going to sleep the ONU either wakes up according to the instructed sleep time or wakes up early if uplink traffic arrives at the ONU. During the sleep period, all downlink packets for the sleeping ONU are buffered by the OLT. When waking up, the ONU has to perform clock recovery and synchronization before being able to send and receive downlink packets. The first packet the ONU sends after waking up is a message informing the OLT that it has woken up. The OLT then responds by sending the packets that were buffered while the ONU was sleeping to the ONU. In order to handle the situation of early wakeup, there is during the sleep period allocated exactly enough uplink bandwidth for the sleeping ONU to be able to send a wakeup message. Using these mechanisms of SPW, the OLT is able to put ONUs to sleep in accordance with the patterns of their downlink traffic.

The ***second sleep method*** is proposed by J. Zhang and N. Ansari in [8]. This method takes advantage of the fact that the OLT only sends downlink traffic to one ONU at a time, and only one ONU is sending uplink traffic at a time. Because of this, an ONU can for instance for a PON with 16 ONUs potentially sleep 15/16 of the time. This kind of sleep will then, contrary to the SPW sleep method, take place within a *Dynamic Bandwidth Allocation* (DBA) cycle. The sleep is asynchronous for the optical transmitter and the optical receiver where the transmitter considerably easily can be put to sleep in periods outside of the ONU's allocated uplink times. The receiver sleep time is, on the other hand, more difficult to determine due to the ONU not knowing when downlink data for it will arrive. For this reason a function is proposed which defines when data is not allowed to be sent to an ONU. The function $f(\Delta)$, which for instance can be $0.8 * \Delta$ or another fraction of Δ, describes how much time must pass from an ONU finishes receiving downlink traffic from the OLT until the OLT again is allowed to send traffic to the ONU. The Δ parameter is the length of the previous period between the ONU finishing receiving traffic from the OLT and the following time when the ONU began receiving traffic again. Using this method allows both the ONU and the OLT to know in which periods downlink traffic for the ONU is not being sent and thereby making the ONU able to sleep in these periods.

The ***third sleep method*** is proposed by S. Lee and A. Chen in [9] and uses two different modes of bandwidth management to make sleep possible. In the *Normal mode*, DBA is performed as usual and without sleep but when the traffic load becomes below a certain threshold, the system switches to *Power Saving mode* and turns into a slotted system. In this mode *Fixed Bandwidth Allocation* (FBA) is used instead of DBA, and each ONU gets one downlink and one uplink slot for data where the ONU can sleep in an asynchronous way. After the switch to FBA all ONUs are informed of their timeslots, and are then able to sleep outside of this timeslot. Furthermore it is possible to end the timeslot if no more traffic should be sent or received, and thereby extend the sleep length.

EPON OPNET Model

The implemented EPON system consists of one OLT and 64 ONUs, where the number of active ONUs in the network is determined by a simulation parameter. Not every EPON feature is implemented, only the parts necessary for basic functionality and relevant for the expansion with sleep methods. The main control protocol of EPON is the *Multi-Point Control Protocol* (MPCP), using GATE and REPORT messages, which is implemented in a simplified form with primarily the resource allocation aspects implemented. In addition to the implementation of the EPON, two groups of traffic generators are also implemented. The ONU and the OLT are each implemented as a single process model and fed with traffic from the traffic generators as shown in Figure 1 for the ONU, where the OLT node model almost is identical.

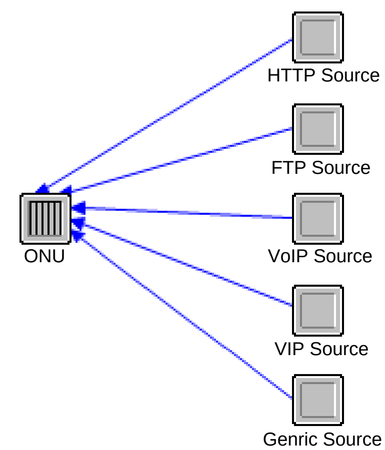

Figure 1: ONU node model

The traffic generators are used to get an insight into the effectiveness of the sleep methods for both specific types of traffic as well as for basic traffic patterns. The five traffic generators are divided into two groups, the individual type generators and the generic traffic generator. The individual type generators are generating HTTP, FTP, *Voice over IP* (VoIP), and *Video over IP* (VIP) traffic based upon real-life examples of these types of traffic, with a client-server structure for HTTP and FTP, and a client-client structure for VoIP and VIP. These generators are used to produce somewhat specific traffic patterns in accordance with the different types of traffic. The generic traffic generator is using a Pareto On/Off generator, thereby generating traffic with basic traffic patterns.

All the necessary functionality of the ONU is implemented in the ONU process model, shown in Figure 2, with the sleep methods implemented as described in the next section.

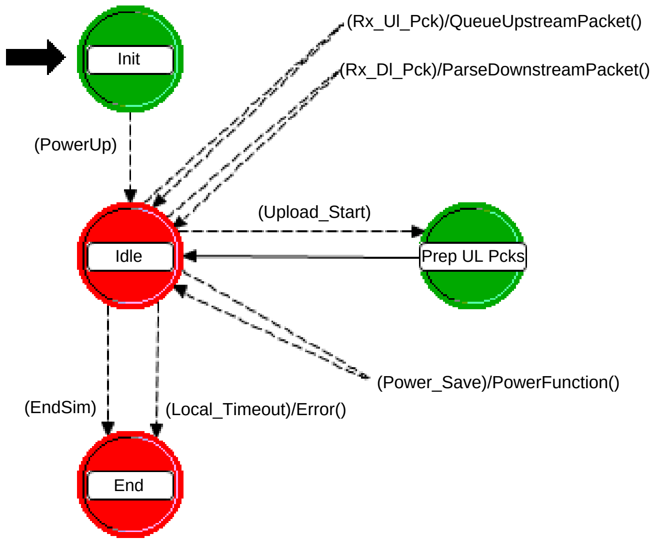

Figure 2: Process model for the ONU

Incoming packets to the ONU, both uplink and downlink, are queued, removed, or parsed as they arrive, depending on their type and destination. GATE messages from the OLT are parsed in order to set the timer for the next allocated uplink period with the start time and length from the GATE message. Uplink packets from the traffic generators are queued for delivery to the OLT and at the ONU's allocated timeslot the ONU calculates how many packets to send, taking the transmission time of the REPORT message and the optical receiver's on/off times into account. The REPORT message to the OLT includes information about how much time it would take for the ONU to send the remaining packets, in order for the OLT to be able to schedule the next round of uplink bandwidth allocation.

The OLT is implemented using a seemingly similar structure as the ONU, but with clear differences in functionality. The process model of the OLT is shown in Figure 3.

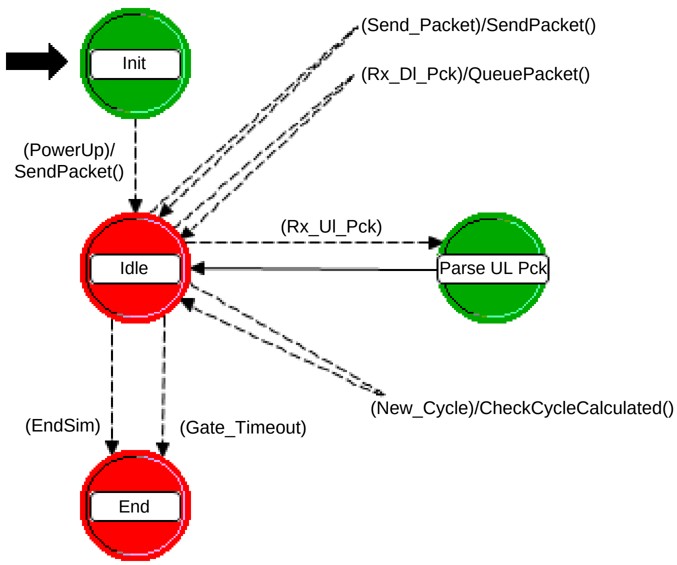

Figure 3: Process model for the OLT

On arrival of a REPORT message at the OLT, the list containing information about all ONUs is updated with the information from the REPORT message. If the received REPORT message is the last expected from the ONUs in the current round, the GATE messages can be created and sent out. The information about the uplink bandwidth allocation for the GATE message is calculated by the DBA algorithm. The DBA algorithm is implemented in a fair way, meaning that it is possible for all ONUs to get the same amount of capacity, but if an ONU does not need it, other ONUs can use the excess bandwidth. The allocated uplink DBA periods are informed to the ONU using GATE messages. This handling of incoming REPORT messages from the ONUs and GATE messages to the ONUs makes it possible for traffic to be sent both in uplink and downlink directions.

Sleep Method OPNET Models

In addition to the mechanisms implemented to make basic operations, such as bandwidth allocation and traffic transport, possible, the three mentioned sleep methods are also implemented. These are implemented as an integrated part of the OLT and ONU process models in such a way that a single simulation parameter determines which – if any - of the sleep methods should be used. This makes it very simple to run simulations with the same setups and traffic scenarios while shifting between sleep methods, in order to compare for instance the packet delay and energy consumption of the different methods.

The ***first sleep method***, SPW, is fully implemented in accordance with [7], as previously described. The OLT calculates the average interframe period on each downlink packet arrival for the individual ONUs, and for each generation of GATE messages it is checked whether or not the average interframe period is above a threshold defined by a simulation parameter. Due to the possibility for very bursty traffic or no traffic at all, both are initially not handled very well by the sleep method, an alternative average interframe period is also implemented, which is calculated on GATE message generation and includes the GATE message as a downlink packet in its calculation. If this alternative period is bigger than the normal average interframe period, it is used instead to determine if an ONU should sleep and for how long. When the amount of sleep has been calculated using a SPW algorithm, the wakeup time is then compared with and changed according to future GATE message generation times. This ensures that the wakeup time instructed to the ONU fits with the arrival of a GATE message. By doing so, the ONU can sleep as long as possible as it does not have to wait a long time for a GATE message after waking up. A GATE message is necessary in order to get information about when the ONU can send its wakeup message to the OLT. Through these subtle changes, and a few other smaller modifications, the sleep method is implemented making the ONU sleep in accordance with the SPW algorithms and several SPW specific simulation parameters.

The ***second sleep method*** is implemented in accordance with the way it is specified in [8], as previously described. In this method no exchange of sleep method control messages between the OLT and the ONUs is needed due to the sleep mechanism relying on the OLT and the ONUs having the same algorithm implemented, thereby knowing the behavior of each other. The implementation of the mechanism for ONU optical transmitter sleep is done using the normal uplink DBA cycles where the transmitter is sleeping outside of the ONU's timeslots. The sleep of the optical receiver is determined using the function $f(\Delta) = \beta * \Delta$, where

Δ is the time between the two previous uplink transmissions for the ONU and β is a simulation parameter between 0 and 1. This part is implemented so that the OLT keeps track of Δ for each ONU and then uses a queue for each ONU to buffer traffic until $\beta * \Delta$ time has passed since the ONU's last transmission ended. In cases where Δ exceeds the cycle time, the cycle time is used as Δ instead to ensure that all ONUs wake up in each cycle to receive their GATE message. Because there are periods where no downlink traffic is allowed to be sent to an ONU, it is important that the OLT is scheduling the traffic in an effective way, which is implemented similar to the DBA algorithm. A visualization of the sleep periods necessary to take into account when scheduling downlink timeslots is shown in Figure 4. Here an algorithm is implemented for scheduling the timeslots using the sleep periods and the lengths of the downlink queues for the different ONUs. The algorithm needs to take the fair distribution of capacity into account as well as the periods where the ONUs are not allowed to receive traffic. In this way it is possible for the ONUs to sleep according to the predefined parameters, while other ONUs are receiving traffic, thereby saving power.

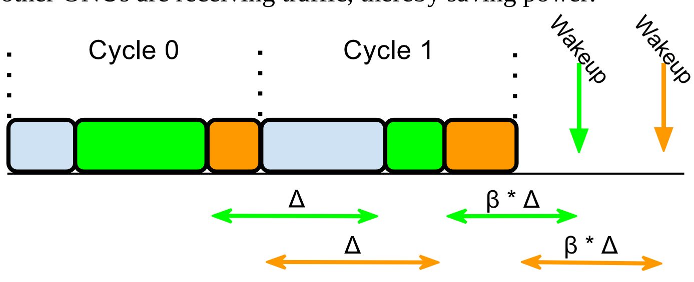

Figure 4: Scheduling sleep according to ONU sleep times

The ***third sleep method*** is implemented according to [9], as previously described. The OLT monitors all traffic and at the end of each cycle it checks if every ONU has used less bandwidth than the total bandwidth divided by the number of active ONUs during an observation time period of a predefined length. If all ONUs have done so, a switch is made from the DBA mode to the FBA mode, which is the power saving mode. MPCP messages are not used in the FBA mode and instead a FBA message is sent out, when shifting to the FBA mode, which informs the ONUs of their slot start time, slot length, and cycle length, which is the slot length multiplied by the number of ONUs plus the length of the discovery window. All downlink packets are in the FBA mode arranged in individual queues instead of one collective queue for all downlink traffic as in the DBA mode. If, during an ONU's timeslot, no pending traffic is left for the ONU, or not enough time to send the next packet, a STOP-GATE message is sent, informing the ONU that its timeslot can be considered over. When sending a STOP-GATE due to there not being enough time for the next packet to be sent, the OLT also checks if the ONU in any of a predefined number of previous cycles the ONU has emptied its queue. In case the ONU in every of these checked consecutive cycles has not been able to send all its packets there is made a shift back to the DBA mode. This is also the case for the ONU, which also sends a stop message to the OLT when no more traffic will be sent from it. Changing back to the DBA mode can also be done as per request by an ONU or if the queue at either the ONU or the OLT gets above a certain threshold. In this way shifting back and forth between the normal mode and the power saving mode is possible, where ONUs in the power saving mode know when they will receive traffic and thereby sleep outside of their timeslots.

Novel Sleep Method

The creation of a novel sleep method is done utilizing all three sleep techniques presented in previous sections, combined in order to get even better energy efficiency. The choices related to optimizing and combining the different sleep methods are made in accordance with sleep methods' advantages, disadvantages, and opportunities learned through both the implementation of the sleep methods and the simulation of them in different scenarios, as presented in the next section.

The *third sleep method*, which shifts between a slotted power-saving mode and a normal mode, is used as the primary sleep mechanism for the *novel sleep method*. In the normal mode the *first sleep method* is used, which then provides the opportunity to save power outside the slotted power-saving mode. On top of the *first sleep method* the *second sleep method* is also used; though only for the uplink part of the ONU to have the transmitter be turned off when not needed. The *second sleep method* is only utilized for the *novel sleep method* in periods where the process is in the normal mode and an ONU is not asleep due to the *first sleep method*. In this way the three sleep methods are combined to complement one another and also provide greater control concerning the way power is saved. This flexibility can for instance be used to customize the *novel sleep method* to a certain type of traffic patterns or to save more power by allowing higher packet delay.

As mentioned, the three previously proposed sleep methods are not used for the *novel sleep method* as is, but are rather chosen to be optimized or to only have parts of them used. The *first sleep method* is optimized in relation to subsequent sleep periods which previously required the exchange of two GATE messages and two REPORT messages in order for an ONU to wake up and go right back to sleep at times in absence of both uplink and downlink traffic. This operation is improved for the *novel sleep method* so that only a single GATE message and no REPORT messages are needed in order to put an ONU back to sleep. As previously described, only the uplink part of the *second sleep method* is used due to it being very easily integrated with the existing EPON MPCP system, and thereby increasing the energy efficiency without increasing the packet delay. The *third sleep method* is changed for the *novel sleep method* so that now the cycle time is constant, determined by customizable parameters, while the individual time slots are dynamically calculated each time the power-saving mode is entered and then stays constant until the normal mode is entered again.

Through the optimization and integration of the three sleep methods the *novel sleep method* is created, which the effectiveness of is shown in the next section compared to the other three methods.

Simulation Results

In this section a selection of results generated through the simulations is presented.

A scenario with 64 ONUs and three different traffic loads is chosen. Downlink average traffic loads of 1.28%, 2.56%, and 50% are used to represent low, high, and extremely high traffic loads, respectively. A load of half the size is used in the uplink direction. An exponential distribution is used to distribute the load among the ONUs, to ensure as realistic a scenario as possible. This is the case for both types of traffic generators. The network bandwidth is set to 1 Gbit/s and the simulations are run without sleep, as well as with the three presented sleep methods.

During the simulations the amount that each ONU sleeps and the number of times it sleeps are observed. From this it is calculated how much energy the system uses compared to when no sleep method is used. When calculating the energy consumption for the sleep methods it is taken into account that going to sleep is not an instantaneous drop in power usage but rather a slope [11]. In addition, the time it takes to regain network synchronization when waking from sleep is also taking into account. Besides collecting information about sleep aspects during the simulations, the queue delay of packets is also logged in order to see the effect caused by the introduction of the different types of sleep methods. Looking at an average of the traffic generated, it is observed that a simulation time of one hour is enough to get results with an acceptable confidence interval. The results are as follows.

In Figure 5 the power usage is shown for the different sleep methods with the two different types of generated traffic. The percentage of power they use is compared to when no sleep method is used.

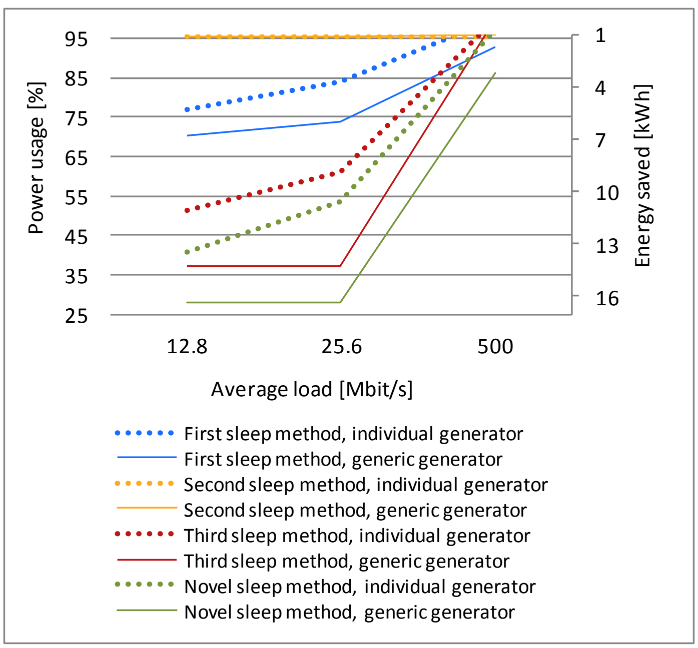

Figure 5: Power efficiency of sleep methods

By looking at the first three sleep methods, it is clear that the *third sleep method* gives the best results for the 12.8 Mbit/s and 25.6 Mbit/s loads for both generic traffic and individual type traffic. The reason why generic traffic gives a better result can be explained by the fact that it varies within a smaller interval as can be seen in Figure 6, where the individual type traffic, even though the average is the same, has many larger peaks originating from the FTP and HTTP traffic. These peaks makes sleep method 1 go into normal mode for a while before switching back into power saving mode again. When the traffic load becomes very high, sleep method 1 will never go into power saving mode, which is why there is not saved any power for the 500 Mbit/s scenario.

The *first sleep method*, while not as effective as the *third sleep method*, is still considerably more power saving than the *second sleep method* and is even able to save about 7% power on very high traffic loads for the generic generator. The *second sleep method* is only able to save around 4% power; however, it does so for all loads.

Looking at the *novel sleep method*, it can be seen in the figure that the combination of the sleep methods and the improvements of the individual methods results in almost an additional 10 percentage points of energy saved compared to the *third sleep method*, which so far gave the highest energy improvement. It is also observed that especially for the scenario with generic traffic the *novel sleep method* saves an impressive amount of energy even under very high load.

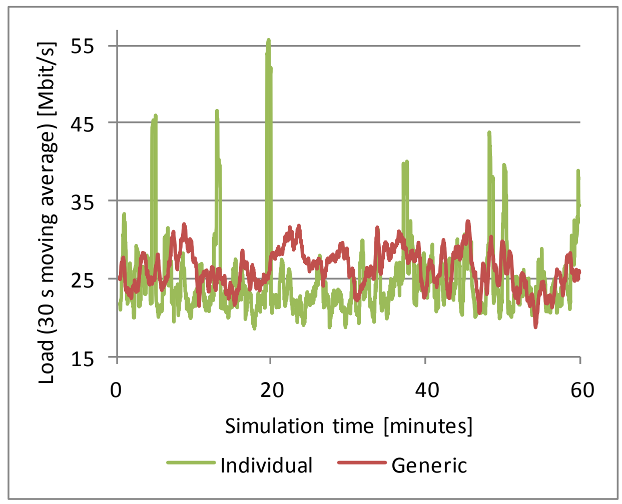

Figure 6: Generic traffic compared to Individual traffic

As described, saving power is not the only important factor and it is also necessary to look at the potential drawback; the increase in queue delay. Even a very power sleep method is useless if it causes unacceptably high packet delays.

Delay-sensitive real-time traffic like VoIP can tolerate a total packet loss of 1% [10], which means that a packet loss of 0.5% in each side's access network should be the maximum. For this reason, it is interesting to observe the delay of the 99.5% "fastest" packets, while the remaining 0.5% can be considered lost without causing too much harm.

The delay which 99.5% of packets experience is found using cumulative distribution functions for the packet delays in the simulation. In Table 1 the maximum delay in milliseconds that 99.5% of the packets experience is shown. Comparing the *third sleep method* with the first, where the third was the most power efficient of the three previously proposed sleep methods, it is seen from the table that in all the scenarios the *first sleep method* affects the delay much less than the *third sleep method*. When further comparing the delay of the *first sleep method* to when no sleep method s used, it is seen that the delay is only a few milliseconds larger. Looking at the *second sleep method*, which did not save much energy, it is observed that it, with regard to delay, has a poor performance too – at least concerning individual type traffic – and that it generally has a large delay compared to the power saved.

Finally when looking at the *novel sleep method* it is seen that a substantial improvement has been made. The delay still is not as low as the *first sleep method*, but compared to the *third sleep method the delay has clearly improved*.

Scenario	No sleep	Sleep method			
		1	2	3	Novel
12.8 Mbit/s, generic	0.036	4.3	0.9	25.5	25.6
12.8 Mbit/s, individual	0.074	7.1	21.1	76.3	26.3
25.6 Mbit/s, generic	0.036	4.4	4.5	25.6	25.6
25.6 Mbit/s, individual	0.37	6.2	21.9	66.0	25.9
500 Mbit/s, generic	0.04	0.04	2.7	N/A	1.8
500 Mbit/s, individual	51.6	53.1	181.8	N/A	51.6

Table 1: Max delays for 99.5% of the packets in milliseconds

This means that when looking at the first three methods if a delay of up to 76 ms can be accepted, then the *third sleep method* clearly is the better choice; even though it does not work at very high traffic loads but here the other methods do not work very efficiently either. If very low delays are required then the first method is preferred since it only causes few milliseconds of packet delay. The novel method reduces the delay significantly and also provides an even lower energy usage, saving more than 16 kWh; so if causing a delay of 25 ms can be tolerated then the *novel sleep method* is clearly preferable over the other three methods.

Conclusion

In this paper three previously proposed methods for saving energy in *Passive Optical Networks* have been presented and a new sleep method has been proposed. The implementation of a PON and the sleep methods has been shown as well as the simulation results for these models.

The results of the simulations rely heavily on the traffic generators – which is why two distinctively different traffic generators were used. From the results concerning the three previously proposed sleep methods it can be concluded that either the *first sleep method* or the *third sleep method* should be chosen, depending on the delay that can be accepted. Furthermore, insights have been gained into the different sleep methods in relation to the techniques they use, their power saving potential, and their effect on packet delay. These insights were then used to design a new solution, combining the different sleep methods in order to increase the energy efficiency further and reduce the delay. The simulation results for the *novel sleep method* showed that its energy efficiency was better than the other three sleep methods and that the delay was reduced compared to the *third sleep method*, which was the one with the best energy efficiency.

Overall, the insight gained from implementing and simulating the three previously proposed solutions was shown to be useful through the design and simulation of the *novel sleep method* which with the selected traffic patterns outperformed the other methods.

References

[1] Energy Efficiency Plan 2011 [Online], http://ec.europa.eu/energy/efficiency/action_plan/action_plan_en.htm

[2] Report from Gartner [Online], http://www.gartner.com/it/page.jsp?id=530912

[3] A. Gladisch et al., Tu.4.A.2, ECOC ’08 [Online], http://goo.gl/IOy74 and http://goo.gl/PqLg5

[4] Fixed broadband internet subscription 00-10 [Online], http://www.itu.int/ITU-D/ict/statistics/material/excel/2010/FixedBroadbandInternetSubscriptions_00-10.xls

[5] OECD broadband statistics (June 2011) [Online], http://goo.gl/MhCwV

[6] OPNET Modeler v16.1A PL1

[7] R. Kubo et al., “Adaptive Power Saving Mechanism for 10 Gigabit Class PON Systems,” IEICE Transactions on Communications. 2010, Vol. E92-B, 2

[8] J. Zhang and N. Ansari, “Toward Energy-Efficient 1G-EPON and 10G-EPON with Sleep-Aware MAC Control and Scheduling,” IEEE Communications Magazine. 2011, Vol. 49, 2.

[9] S. Lee and A. Chen, “Design and Analysis of a Novel Energy Efficient Ethernet Passive Optical Network,” 2010

[10] QoS - voip-info.org [Online], http://www.voip-info.org/wiki/view/QoS

[11] L. Shi et al., “Energy-Efficient PON with Sleep-Mode ONU: Progress, Challenges, and Solutions,” IEEE Network Magazine. 2012, Vol. 26, 2.