CrossMark

Automatic Hidden Bypasses in Software-Defined Networks

**Jerzy Domżał[1] · Zbigniew Duliński[2] ·
Jacek Rząsa[1] · Piotr Gawłowicz[1] · Edyta Biernacka[1] ·
Robert Wójcik[1]**

Abstract As global internet traffic continues to increase, network operators face
challenges on how to efficiently manage transmission in their networks. Even
though attempts are underway to make optical networks automatic, the majority of
actions related to traffic engineering are still performed manually by the adminis-
trators. In this paper we propose an Automatic Hidden Bypasses approach to
enhance resource utilization in optical networks. Our solution uses the software-
defined networking concept to automatically create or remove hidden bypasses
which are not visible at the network layer. The mechanism increases throughput and
reduces transmission delays.

✉ Jerzy Domżał
   jdomzal@kt.agh.edu.pl

   Zbigniew Duliński
   dulinski@th.if.uj.edu.pl

   Jacek Rząsa
   rzasa@agh.edu.pl

   Piotr Gawłowicz
   gawlowicz.p@gmail.com

   Edyta Biernacka
   ebiernac@agh.edu.pl

   Robert Wójcik
   robert.wojcik@kt.agh.edu.pl

[1] Department of Telecommunications, AGH University of Science and Technology, Al.
     Mickiewicza 30, 30-059 Kraków, Poland

[2] Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University,
     Łojasiewicza 11, 30-348 Kraków, Poland

✷ Springer

1 Introduction

Total IP traffic continues to increase globally. It is forecast to increase over threefold from 2012 to 2017, reaching 120 exabytes (10^{18} bytes) per month in 2017 [1]. This puts pressure on network operators to efficiently and effectively manage their optical network infrastructure. Despite many attempts to automate the process, optical networks are usually still managed manually. Optical paths are created by the administrators based on the traffic distribution.

The concept of software-defined networking (SDN) becomes more and more popular in the context of network management in teleinformatics networks. In SDN, it is assumed that the control plane and the data plane are separated to simplify the management of traffic in the network. At the control plane, usually the central controller decides where packets should be sent at the data plane. This concept is also used in optical networks, especially to improve the effectiveness of routing and wavelength assignment.

To increase efficiency in optical networks optical bypasses can be used. A bypass is a dynamically set path which offloads traffic from the permanently established and regularly used links. Bypasses are created and removed based on the existing demands. Others have proposed several ways to realize optical bypasses. They are presented in Sect. 4.

Our approach, presented in this paper, is the Automatic Hidden Bypasses (AHB). This method uses a hidden bypass functionality as presented in [2], and adds components known from SDN, making the process completely automated. This means that bypasses are created and removed down based on existing demands. The network decides when and how to create a new bypass as well as which transmissions should use it. Analysis presented in this paper shows that AHB can provide lower delays and higher throughput. The mechanism yields excellent results in both low and high loaded networks.

Many solutions related to optical bypassing have been proposed in the literature in recent years. Bypasses can be created manually by network operators or automatically in centralized or distributed systems. Usually, bypasses consume lambdas reserved for their implementation and not used during the standard network operation (without bypasses). Sometimes the setup of bypasses is broadcasted at the IP layer and sometimes bypasses are hidden. The detailed description of selected state-of-art solutions is presented in Sect. 4. We have chosen some representative mechanisms to explain existing possibilities for setting up bypasses and present a short comparison of those solutions with the AHB mechanism.

The remainder of the paper is organized as follows. Section 2 presents the Automatic Hidden Bypasses concept. After the concept is introduced, Sect. 3 shows an analysis performed in the ns-3 network simulator. The analysis is divided into two parts which show the results of two examined scenarios under multiple configurations. Section 4 describes selected existing approaches for providing optical bypasses in various forms and highlights differences compared to AHB. Finally, Sect. 5 concludes the paper.

2 Automatic Hidden Bypasses

This section describes the Automatic Hidden Bypasses (AHB) algorithm. An illustration of a core network is presented in Fig. 1. We assume that each IP router is bound with an optical network switch (OXC), which is typically the case with existing carrier networks. In order to provide the AHB functionality, we employ traffic offloading with an optical bypass. An optical bypass can be established on demand, and it is transparent to the IP layer, i.e. the IP layer is not aware of the existence of this bypass. In this way the routing tables in routers do not need to be updated. The bypass ingress node must be informed that certain transmissions should be forwarded into the bypass rather than to the interface indicated by the routing table [2].

The AHB algorithm, described in this section, is a traffic offloading method in which optical bypasses are created and torn down automatically based on the network congestion status. We assume that the operator has mechanisms of
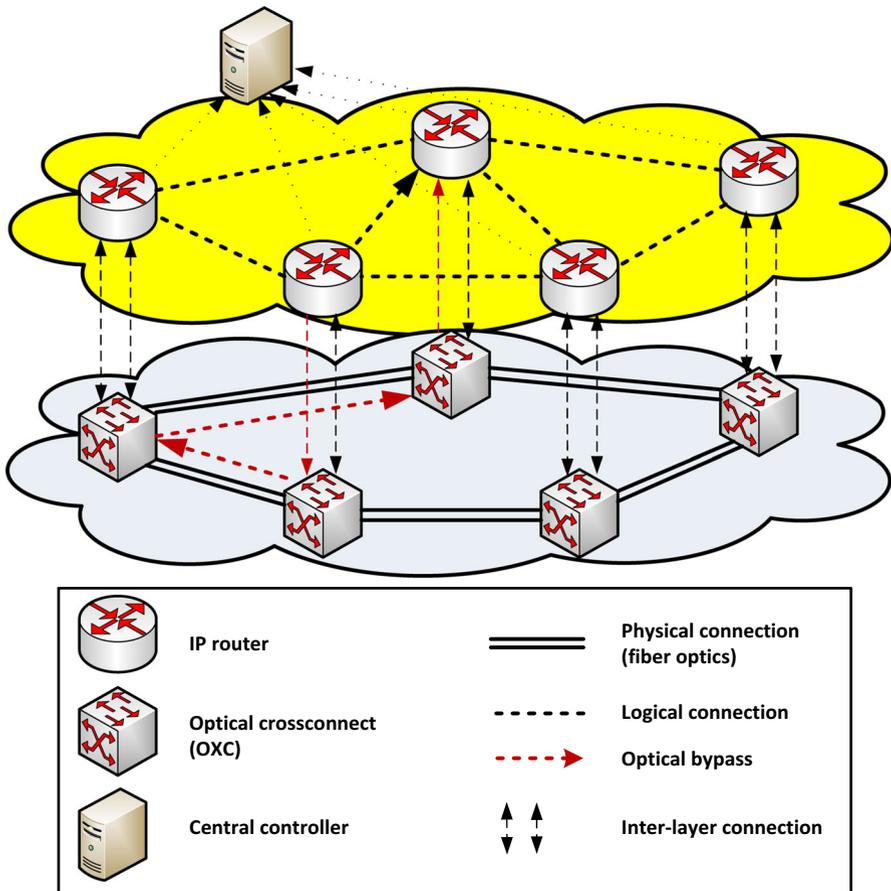

Fig. 1 IP and optical network

establishing optical paths in the network. AHB is possible due to the potential of SDN and flow-based traffic treatment. It is very important to note that, unlike other typical mechanisms to set up bypasses, in our proposal traffic matrix does not need to be known in advance. Only a part of physical resources is available at the IP layer. The rest is reserved for bypass creation and is used only when congestions occur in the network. This ensures a partial separation between both layers and enables more efficient resource management.

AHB provides the following functionalities:

- monitoring of every link's congestion status,
- finding an optimal bypass in response to current congestion status,
- offloading certain flows through bypasses.

The main advantage of using AHB is that the operator does not need to use excessive optical resources to guarantee a congestionless network. Instead, we propose using as many optical resources as necessary in each situation. When the usage of certain links exceeds a certain threshold, new optical paths are created. When the demand ceases, resources are freed. This way, not all optical links are used. Moreover, optical paths are established based on the current traffic matrix and can be better tailored to the existing demands.

In the remainder of this section, the AHB algorithm is presented in details. In particular, the following are presented: the components required to realize the mechanism, the packet processing procedure, and the method of calculating an optimal bypass.

2.1 Components

On top of classic networks, AHB requires additional components that are typical for SDN networks. These include:

1. *Central controller* This is the entity which receives and stores information from the nodes about the ongoing transmissions. Moreover, when such a demand occurs, the central controller is responsible for finding the optimal optical bypass, creating it and tearing it down. The process of finding the optimal bypass is described in Sect. 2.3.
2. *Flow table (FT) on each IP node* Every IP node stores the information about all the flows that are active on its interfaces. The information about each flow includes:

    - *Hashed flow ID:* the identifier (ID) is hashed from the usual 5-tuple, i.e. IP addresses, transport layer protocol and its port numbers.
    - *Flow time stamp:* the time of the last received packet of this flow. This is needed to erase flows that are no longer active.
    - *TTL (time to live) value in IPv4 or hop limit in IPv6:* a value stored for the first packet of a flow. This is needed for the central controller to know the path of each flow. The controller will receive information about each new

flow from every node that the flow traverses. By comparing TTLs, the controller can establish the exact path.

- *Bypass interface identifier:* the identifier of an optical interface on which the bypass originates. For most flows, this field is empty, which means that those flows should be forwarded classically—according to the routing table. However, when a bypass is created, some flows will be forwarded using this bypass. Those flows have the identifier set.

3. *Communication protocol* This is used to provide communication between the central controller and nodes in the network. This protocol is outside of the scope of this paper. Following minor modifications, most standard protocols can be used, e.g. OpenFlow. OpenFlow is able to transmit data between nodes and the central controller. In AHB it is necessary to inform the central controller periodically about active flows and their rates. On the other hand, the central controller can indicate the necessity to setup a bypass by a node. This communication between nodes and controller may be ensured by dedicated signalling protocol. However, we propose to use the OpenFlow. This protocol meets the requirements of the AHB, is stable and implemented in network devices available on the market.

These three additional components are fairly standard when the notion of SDN appears.
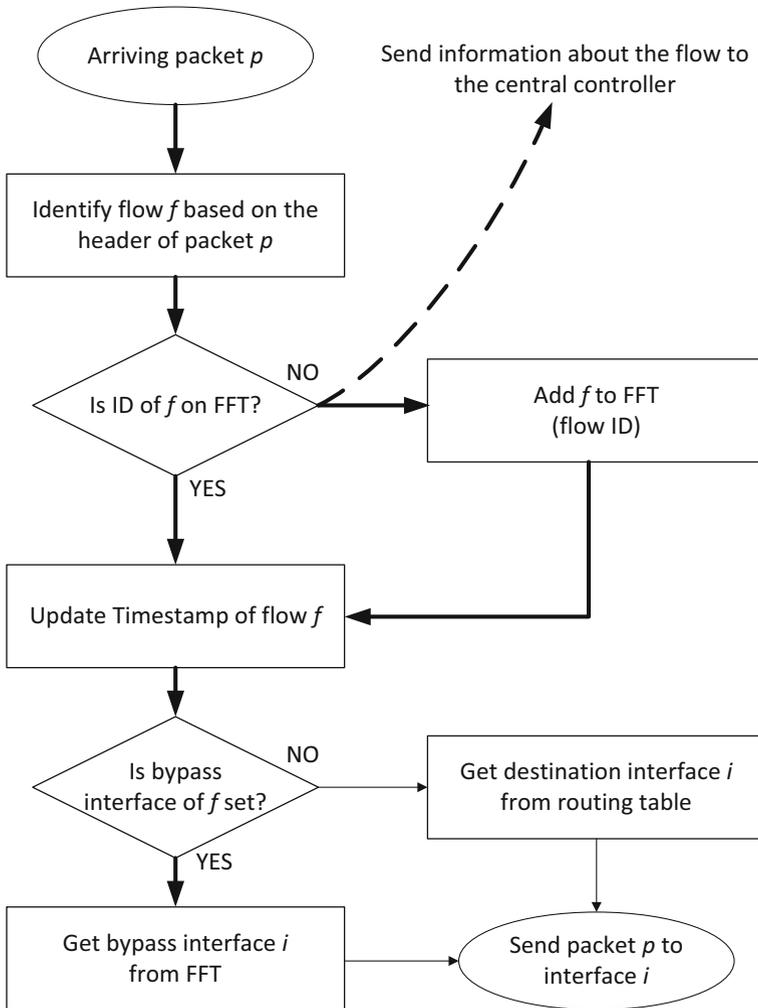
2.2 Packet Processing Procedure

Figure 2 presents the packet processing procedure in routers implementing the AHB concept. Firstly, the flow is identified by hashing certain fields in the packet header. Then, it is checked whether the flow is already present on the FT. If it is not, the flow ID is added to the table, and information about the new flow is sent to the central controller. Note that the information does not need to be sent on every new flow. It is sufficient to gather more data and distribute this information once a while. After the flow has been added to the FT, or if it was already present there, its time stamp is updated.

  The last action in the process is to determine the outgoing interface for the packet. This is performed by checking the *bypass interface identifier* field in the FT. If the field is empty, it means that this flow is to be forwarded normally, i.e. according to the routing table. However, when the field is not empty, this means that the flow should be forwarded through a bypass, which is identified by the value in the field. Finally, the packet is forwarded to the appropriate interface.

2.3 Optimal Bypass Calculation

When a link in the network crosses a congestion threshold a bypass creation procedure starts. The node which detects the congestion informs the central controller about the situation. We have to note that in this case a congestion does not

Fig. 2 Packet processing procedure

mean that the link is overburdened. We assume that a link is congested when its throughput exceeds the fixed threshold, e.g. 80% of the link capacity. Along with the congested link's ID the node sends information about the rate of each flow that is active on the link. This information, combined with the knowledge the controller already has, is sufficient to start the selection of a new bypass. The central controller has the information about all the flows in the network, especially about their exact paths. Additionally, the controller is aware of the currently available optical resources that can be used to form hidden bypasses. The flows' bit rate can be estimated, e.g. by using techniques from [3]. In this solution, basic flows' properties, which can be inferred from a sampled stream in an easy way, are analyzed. The

results presented in [3] confirm the usefulness of the algorithm and its high effectiveness.

The amount of total traffic transmitted in a link is observed continuously for the purpose of quick congestion detection. The state and the information about particular flows which is required in the bypass creation process may also be collected constantly. However, to reduce the workload of the routers, the information gathering may be triggered when a link exceeds a pre-congestion threshold which is slightly lower than the congestion threshold. This way, under light loads the information is not gathered, but when there is a threat that congestion can occur, the flow state information starts to be collected.

Upon receiving the congestion notification, the controller searches for all possible bypass configurations and chooses the one with the highest gain calculated according to the following formula:

$$gain = w_1 * T - w_2 * n_\lambda + w_3 * n_{IP} \tag{1}$$

where: T is the amount of traffic that can be pushed into a bypass, n_λ is the number of optical links that form a bypass, n_{IP} is the number of IP hops that a bypass traverses, and w_1, w_2, w_3 are the weight coefficients.

There are three factors that are evaluated. The most important is the amount of traffic present on the congested link that can be transferred to the assessed bypass (T). The more data can be offloaded, the more visible will be the effect of the action. Another factor that contributes to the gain is the number of IP hops that a bypass traverses (n_{IP}). All data which travels via a bypass is transmitted optically without electric conversion in the nodes. This means that from the IP layer perspective a bypass is a one hop jump. This factor is important, as optical-electrical conversion and queuing take time which is saved by the bypass. The final factor that contributes to cost rather than gain (hence the minus sign) is the amount of optical resources that needs to be used to create a bypass (n_λ). Obviously, the more resources that are required, the less attractive the bypass becomes. In the process of finding a bypass, existing bypasses are also taken into consideration. One of them may be selected as the optimal solution.

In the formula, these three factors are meaningless without properly set weights. It is the operator's responsibility to modify these weights accordingly. For example, when there are plentiful spare resources, the greatest focus can be placed on the number of saved IP hops. On the other hand, since resources are scarce, the efficiency of weights w_1 and w_2 may be important. In the simulation experiments which are presented later in the paper, we show various configurations of the weights.

Once the optimal bypass is found, the request to establish it is sent to the proper network devices (if the bypass does not exist already). Also, a list of flows that are to be forwarded through a bypass is sent to the bypass ingress node. The node adds information to the respective flows' rows in the FT. This is how a node knows whether an incoming packet is to be forwarded to a bypass or via a standard IP originated route.

The final action is to remove the unneeded bypass. Each bypass takes only a portion of traffic that existed at the moment of its creation. New flows are not forwarded through bypasses. Because of this, each bypass is bound to lose rather than gain traffic. Normally, a bypass should be removed when transmission on it ends. This releases resources for later reuse. However, in order to avoid inefficiency, a bypass can be unset earlier, i.e. when the utilization of its resources falls below a certain threshold, referred to as bypass remove threshold (BRT).

All actions taken by the controller and routers do not need to be realized immediately. Nodes in the network observe traffic load in links in a real time and when a node notices that a link is likely to become congested (by exceeding pre-congestion threshold) this is reported to the central controller. From this point on, detailed statistics about flows being served in this link are collected by the router, and when the link becomes congested, they are sent to the central controller. This package of data can be successfully sent between a node and the controller, even if millions of flows are served at the time. As a result, the central controller can begin the procedure of selecting an optimal bypass just after the link becomes congested. However, the list of all possible bypasses is created when the pre-congestion threshold is exceeded. In this way, when the link becomes congested, the central controller already has candidates for bypasses and all data needed to calculate Eq. 1. The key assumption is to set the thresholds that indicate actions properly. Even if the number of candidate bypasses is large, the calculation is simple and can be executed quickly. To do this, a dedicated resources can be reserved for such operations in the central controller.

As we can see, the whole process of selecting an optimal bypass does not need to be performed in a very short time. The central controller has enough time for all operations. Of course, the process of selecting candidate bypasses can be improved by selecting the fixed limited number of candidates (e.g. paths having no more than fixed number of hops, with limited maximum delay, etc.). As a result, the calculation process of an optimal bypass can be even shorter. In scenarios analyzed
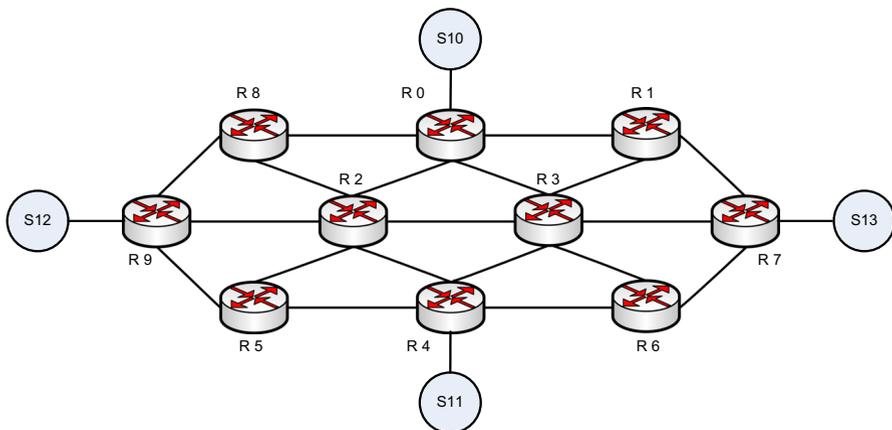
Fig. 3 Simulation topology

in this paper, based on the topology presented in Fig. 3, the process of selecting bypasses was very short, negligible from the whole AHB algorithm operation process time.

There is also no need for synchronization among routers because traffic sent through bypasses is not visible at the IP layer. Under normal conditions, bypasses are set up to cover a part of flows' path at the IP layer. This means that after O-E conversion (at the end point of a bypass) packets of flows have to be served again at the IP layer in a node which belongs to the original path. In this way, loops cannot occur. On the other hand, when the topology changes, some bypasses might have been torn down. The central controller is aware of the current topology and can react accordingly. In such a case, the controller analyzes whether the change affects bypass and can cause loops. The bypasses which may result in loops have to be re-arranged.

The AHB mechanism is proposed to be used for bypass selection by the central controller. Of course, this solution may be generalized to other network solutions like, e.g. Multiprotocol Label Switching (MPLS) or to make choice only among existing optical paths. Also other factors can be taken into consideration when finding a bypass. It depends on operator's needs. Moreover, the AHB algorithm can cooperate with solutions, where, e.g. congestion in links is indicated in different way. For example, in Flow-Aware Networks (FAN) a link is considered as congested when the coefficient representing the amount of elastic or streaming traffic in this link exceeds border threshold. FAN was originally proposed in [4] as an architecture to ensure quality of service for transmissions based on flows. However, substantial work was necessary to be conducted to serve priority traffic in congestion. There are some mechanisms to solve this problem, like Efficient Congestion Control Mechanism proposed recently in [5]. Such a solution can be more effective when the AHB algorithm is also implemented. To estimate the flows' throughputs the structure of two-dimensional model proposed in [6] can be used. This is an efficient method to estimate the average throughputs of elastic flows, which usually consume most of link bandwidth.

3 Simulation Results

In this section, we present the results of simulation experiments performed in the ns-3 simulator. The functionality of the AHB algorithm was implemented in the simulator and the source code is publicly available at [7]. The goal is to show the efficiency of the proposed mechanism. Over 500 simulation runs were performed in the topology presented in Fig. 3. Every core node has an OXC+IP functionality. It is possible to implement up to four lambdas between OXCs. Some of them may not be visible to nodes at the IP layer (if AHB is used). The capacity of links between two nodes at the IP layer is equal to the total capacity of lambdas implemented between these nodes at the optical layer and reported to the IP layer. For example, if all four lambdas between two nodes are visible at the IP layer, the capacity of the link between these nodes at the IP layer is equal to four times the capacity of one lambda. The propagation delay for each link in the network was set to 1 ms. This

topology is relatively simple but realistic. It is assumed that at least few paths can be established between any nodes and additional lambdas are available in the optical layer. In the proposed topology, it is easy to explain and analyze the obtained results. However, the AHB mechanism will work in each topology with multiple paths between nodes and with enough lambdas in the optical layer.

In the first scenario, traffic was sent from S12 to S13. 40 users were connected to S12 and another 40 users were connected to S13 (each by a 0.125 Gbit/s link). Every 0.01 s a source user was randomly selected and began a TCP flow transmission to a randomly selected destination user. Flows were generated with Pareto distribution. The mean size of each flow was 5 MB, shape factor was set to 1.5, and the packet size was set to 1500 B. For every core link four lambdas were available at the optical level (each with capacity equal to 0.25 Gbit/s). We used first-in first-out (FIFO) queues sized to 250 packets and the Open Shortest Path First (OSPF) routing protocol. The simulation duration was set to 200 seconds and the warm-up time was 30 seconds. The simulations were repeated at least 10 times for each configuration. 95% confidence intervals were calculated using Student's t-distribution.

The routing protocol chose the path between S12 and S13 through routers R9-R2-R3-R7. While the cost of each core link was the same, it was the shortest path. We observed several parameters during simulations. They are listed in Table 1. Transmitted data is a mean value of data transmitted during a simulation run. The delay shows a mean packet delay, and the number of hops shows the mean number of links at the IP layer which were traversed during a simulation run. The resource usage means a ratio (presented in %) of available (active) capacity to total capacity of all links in a network. The number of bypasses describes how many bypasses were set up during a simulation, and bypass length is the mean number of hops at the optical layer for a bypass. The value of bypass activity shows how long on average a bypass was active.

First, we assumed that for every core link all lambdas were visible at the IP layer and bypasses were not used. As a result 1 Gbit/s in each core link was available. We can see that 21.17 ± 0.01 GB was transmitted in the network. Some traffic was lost due to congestion of links which composed the path. The mean delay of a packet was equal to 13.36 ± 0.01 ms and the resource usage was 100%.

Next, we analyzed the network with only two lambdas visible at the IP layer (the capacity of each link at the IP layer was 0.5 Gbit/s). Two other lambdas for each link were used to build optical bypasses. Each router in the network periodically (5s) informed the central controller about new flows it was serving. As a result, the central controller was able to build full paths for all flows. When a link in the network crossed a pre-congestion threshold (70% of link capacity) the node corresponding to this outgoing link started to collect throughputs of all flows it was serving in this link. When the congestion threshold was reached in the link (80% of the link capacity), a call for a bypass was sent to the central controller with statistics for all flows transmitted through this link. We analyzed three cases:

– case (a): $w_1 = 1000$, $w_2 = 1$, $w_3 = 100$
– case (b): $w_1 = 1000$, $w_2 = 1$, $w_3 = 1$
– case (c): $w_1 = 1000$, $w_2 = 100$, $w_3 = 1$

**Table 1** Network performance under various configurations (scenario 1)

Parameter	w/o AHB	Case	Bypass remove threshold				
			0.1	0.2	0.3	0.4	0.5
Transmitted data (GB)	21.17 ± 0.01	(a)	41.39 ± 0.14	41.39 ± 0.14	41.39 ± 0.14	41.38 ± 0.12	41.38 ± 0.13
		(b)	41.09 ± 0.06	41.12 ± 0.10	41.06 ± 0.06	41.14 ± 0.11	40.89 ± 0.53
		(c)	41.09 ± 0.09	41.10 ± 0.09	41.02 ± 0.04	41.09 ± 0.09	41.07 ± 0.06
Delay (ms)	13.36 ± 0.01	(a)	12.56 ± 0.47	12.56 ± 0.47	12.56 ± 0.47	12.49 ± 0.37	12.51 ± 0.39
		(b)	14.23 ± 0.41	14.21 ± 0.43	14.03 ± 0.66	14.27 ± 0.36	14.11 ± 0.48
		(c)	14.63 ± 0.08	14.54 ± 0.26	14.55 ± 0.28	14.47 ± 0.36	14.28 ± 0.28
No. of hops	5	(a)	4.42 ± 0.09	4.42 ± 0.09	4.42 ± 0.09	4.41 ± 0.08	4.41 ± 0.08
		(b)	4.81 ± 0.07	4.80 ± 0.09	4.79 ± 0.10	4.81 ± 0.07	4.79 ± 0.08
		(c)	4.87 ± 0.01	4.86 ± 0.03	4.86 ± 0.03	4.85 ± 0.04	4.83 ± 0.03
Resource usage (%)	100	(a)	69.53 ± 0.12	69.53 ± 0.12	69.53 ± 0.12	69.53 ± 0.38	69.65 ± 0.37
		(b)	70.09 ± 0.13	70.09 ± 0.23	69.83 ± 0.39	70.07 ± 0.41	69.55 ± 0.62
		(c)	70.50 ± 0.01	70.38 ± 0.28	70.37 ± 0.30	70.25 ± 0.37	69.87 ± 0.47
No. of bypasses	n/a	(a)	10.40 ± 0.99	10.40 ± 0.99	10.40 ± 0.99	11.40 ± 1.27	13.20 ± 2.14
		(b)	15.80 ± 0.50	15.80 ± 0.50	16.60 ± 0.99	18.80 ± 4.54	20.00 ± 3.42
		(c)	16	16.80 ± 1.45	17.40 ± 2.30	19.00 ± 2.83	24.40 ± 2.16
Bypass length	n/a	(a)	2.94 ± 0.27	2.94 ± 0.27	2.94 ± 0.27	2.85 ± 0.17	2.80 ± 0.26
		(b)	2.01 ± 0.06	2.04 ± 0.11	1.98 ± 0.03	2.01 ± 0.07	1.94 ± 0.07
		(c)	2	2.03 ± 0.05	2.03 ± 0.05	2.07 ± 0.06	2.15 ± 0.12
Bypass activity (s)	n/a	(a)	195.05 ± 0.40	195.05 ± 0.40	195.05 ± 0.40	175.60 ± 17.53	149.78 ± 19.54
		(b)	192.74 ± 5.80	190.32 ± 11.82	183.90 ± 14.61	167.20 ± 35.74	151.24 ± 32.08
		(c)	195.43 ± 0.18	186.95 ± 14.81	181.49 ± 21.16	166.95 ± 25.00	127.29 ± 12.92

$Case(a)$: $w_1 = 1000$, $w_2 = 1$, $w_3 = 100$, $Case(b)$: $w_1 = 1000$, $w_2 = 1$, $w_3 = 1$, $Case(c)$: $w_1 = 1000$, $w_2 = 1000$, $w_3 = 1$

We present the simulation results for three cases just to show the effectiveness of the mechanism. However, in a real network an operator should decide on the values of the weights in formula (1) taking into consideration available resources and goals to be achieved. We decided to set the congestion threshold to 80% which allowed us not to saturate links in the network. We made other analysis (not presented in the paper) which proved that this value was set rationally (delay in communication between controller and routers was acceptable). For each case, the BRT was changed in range from 0.1 to 0.5. The case without BRT (BRT=0) was not considered. We assumed that such a threshold is necessary to eliminate the unfavourable case when one slow flow may occupy the bypass unnecessarily. The results are presented in Table 1. The values of the most important parameters (traffic transmitted, resource usage and bypass activity) are also shown in Figs. 4, 5 and 6.

Firstly, we note that in each case the amount of transmitted traffic is significantly higher when bypasses are used. Moreover, also in each case, less resources were used than in the case without bypasses. The case (a) gives the best results for both mentioned criteria, especially when the value of the BRT is low. We should also note that in this case the mean number of hops at the IP layer is lower and the mean bypass length is higher than in the other cases. Based on the presented results, we conclude that the results obtained for case (a) are the best. It means that in the analyzed experiment the implementation of the bypass algorithm allows us to send more traffic with a lower usage of active resources. The results also show that the coefficient related to the number of hops at the IP layer (w_3) should be greater than the similar coefficient related to the number of links composing a bypass (w_2).

In the second scenario, we checked how the proposed mechanism performs in a more loaded network. We assumed that traffic was sent among nodes S10, S11, S12 and S13. Two groups of users were connected to those nodes. In the first group three users were connected to each source node (each by a 0.125 Gbit/s link). Every 0.02 s four users (one for each source node) were randomly selected and began to transmit TCP flows to randomly selected nodes. The mean flow size was 2 MB. This
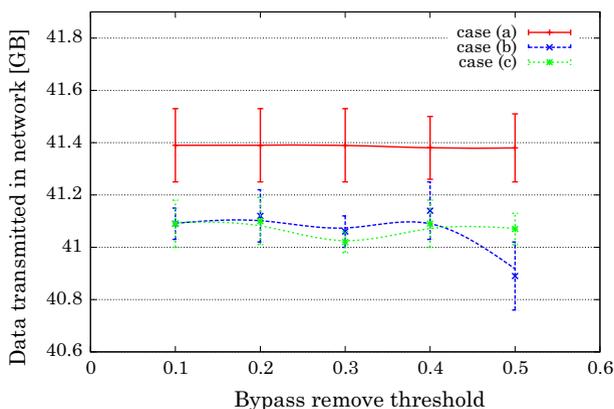
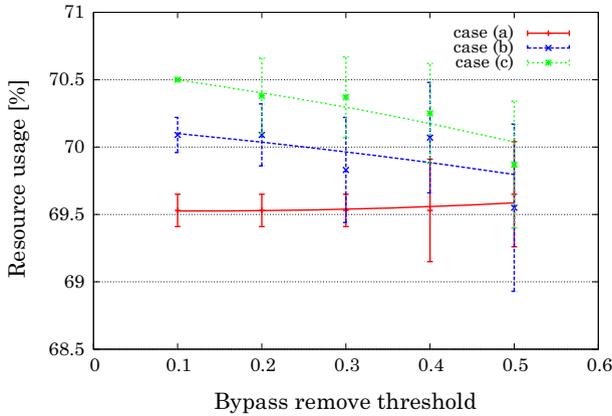**Fig. 4** Traffic transmitted in the network (scenario 1)

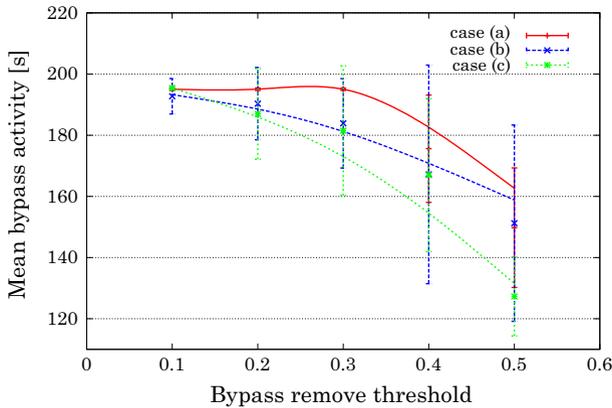**Fig. 5** Resource usage in the network (scenario 1)

Fig. 6 Mean bypass activity in the network (scenario 1)

ensured that traffic was constantly transmitted in the network but the links were not congested. In the second group, 100 users were connected to each source node (each by a 0.1 Gbit/s link). During the respective time periods, UDP flows were generated by users connected to the source node one by one, each every 0.1 s. The aim of such traffic assignment was to simulate a scenario in which traffic in a network increases in given time periods, e.g. such as during streaming of popular video content, which is observed in networks from time to time. Traffic was generated as shown in Fig. 7. The remaining simulation parameters were set as in the previous scenario.

First, we assumed that bypasses were not used. As a result, 1 Gbit/s in each core link was available. We observed the same parameters as in scenario 1. The obtained results are presented in Table 2. We observed that 40.69 ± 0.03 GB was transmitted in a network. The mean delay of a packet was equal to 11.81 ± 0.01 ms and the resource usage was 100%.
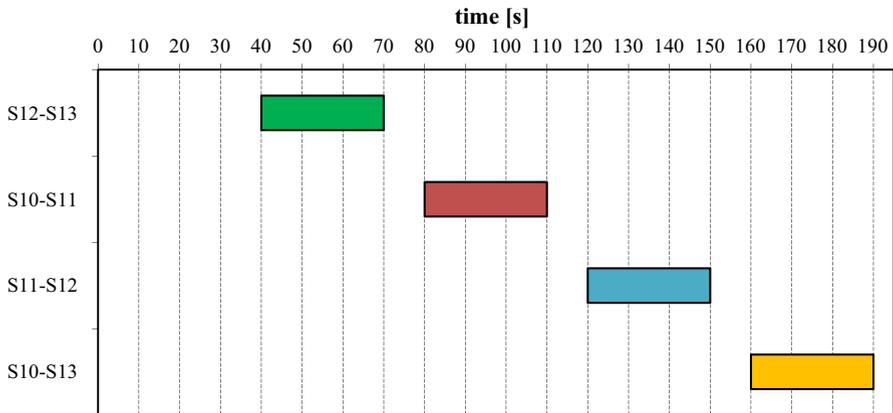
Fig. 7 Source-destination traffic in scenario 2

Next, we examined the network with the option to set up bypasses in the same cases as in scenario 1. The results are shown in Table 2 and in Figs. 8, 9 and 10. In the analyzed cases, the transmitted traffic is higher than in the network without bypasses. Moreover, less resources are consumed than in the analyzed cases. It is a significant advantage of the AHB algorithm. All three cases with bypasses yield similar results; this means that in a highly loaded network, the algorithm selects similar bypasses independently of the assigned weights. It is consistent with our predictions, because under such conditions the options of setting up a bypass are limited. It is also interesting to note that for higher values of BRT less resources are used. On the other hand, bypasses need to be established more frequently, what results in consuming more computing power of the central controller. It is a challenge for network operators to set the value of BRT correctly, i.e. to ensure efficient operation of the central controller as well as not wasting resources. The average bypass activity was relatively low in both analyzed scenarios. However, we should be aware that the whole simulation time was 200 s. In the first scenario, bypasses were active almost all the time. In the second scenario, bypasses were active only when they were required (around 20 s) during the simulation, which means around 10%. In real networks these times can be much longer.

To send data between the central controller and routers, the communication protocol, e.g. OpenFlow needs to be used. We did not observe the amount of traffic sent between the controller and routers. However, it is simple to estimate the maximum amount of such traffic. In the most complex scenario, we generated 39,436 flows. ID of a flow was written to memory by using 4 bytes. Flow IDs were sent to the controller when they were registered in routers. We have 10 routers in the network, so maximum 1.58 MB was sent. Moreover, maximum 50 bypasses were set up in the network during a simulation run. If we assume that all flows were active in the congested links, $50*39436*8=15.77$ MB (we needed additional 4 bytes to send a flow rate to the controller) was sent in the network. This value could be twice more greater if we assume that similar information was sent back from the controller to routers. Concluding, maximum 33.12 MB was sent between the

Table 2 Network performance under various configurations (scenario 2)

Parameter	w/o AHB	Case	Bypass remove threshold				
			0.1	0.2	0.3	0.4	0.5
Transmitted data (GB)	40.69 ± 0.03	(a)	46.92 ± 0.77	46.65 ± 0.75	46.33 ± 1.14	46.22 ± 1.23	46.59 ± 0.59
		(b)	47.73 ± 1.94	48.17 ± 1.48	48.34 ± 2.13	49.05 ± 1.86	47.89 ± 2.02
		(c)	47.98 ± 2.15	48.41 ± 0.90	48.06 ± 1.51	48.00 ± 0.87	48.91 ± 1.13
Delay (ms)	11.81 ± 0.01	(a)	11.20 ± 0.07	11.10 ± 0.06	11.06 ± 0.05	11.04 ± 0.05	11.06 ± 0.04
		(b)	11.79 ± 0.20	11.91 ± 0.11	11.81 ± 0.11	11.85 ± 0.12	11.74 ± 0.18
		(c)	12.07 ± 0.14	11.99 ± 0.14	11.90 ± 0.08	11.89 ± 0.12	11.95 ± 0.09
No. of hops	4.16 ± 0.01	(a)	3.88 ± 0.01	3.87 ± 0.01	3.88 ± 0.01	3.88 ± 0.01	3.87 ± 0.01
		(b)	4.03 ± 0.01	4.03 ± 0.02	4.01 ± 0.02	4.01 ± 0.02	4.02 ± 0.01
		(c)	4.08 ± 0.01	4.06 ± 0.01	4.06 ± 0.02	4.06 ± 0.01	4.05 ± 0.01
Resource usage (%)	100	(a)	58.18 ± 0.28	57.12 ± 0.55	56.59 ± 0.47	56.49 ± 0.53	56.55 ± 0.29
		(b)	58.81 ± 1.17	58.32 ± 1.02	57.74 ± 1.22	58.04 ± 0.85	57.11 ± 0.94
		(c)	58.98 ± 1.40	58.36 ± 0.61	57.43 ± 0.86	57.44 ± 0.46	57.93 ± 0.80
No. of bypasses	n/a	(a)	23.80 ± 0.93	23.20 ± 2.14	22.60 ± 1.68	22.60 ± 1.86	24.00 ± 1.57
		(b)	36.60 ± 3.81	38.40 ± 2.30	37.20 ± 2.65	39.40 ± 1.68	36.80 ± 2.76
		(c)	41.00 ± 4.58	42.20 ± 1.99	40.80 ± 2.41	41.00 ± 1.76	42.80 ± 2.41
Bypass length	n/a	(a)	2.98 ± 0.05	2.94 ± 0.06	2.96 ± 0.04	2.92 ± 0.04	2.93 ± 0.06
		(b)	2.23 ± 0.07	2.20 ± 0.19	2.25 ± 0.18	2.24 ± 0.17	2.14 ± 0.12
		(c)	1.98 ± 0.20	2.02 ± 0.09	1.94 ± 0.10	2.02 ± 0.06	2.06 ± 0.14
Bypass activity (s)	n/a	(a)	38.93 ± 0.66	33.09 ± 0.61	30.38 ± 1.27	29.98 ± 0.41	28.22 ± 1.18
		(b)	37.82 ± 0.96	32.23 ± 0.54	29.81 ± 0.78	29.08 ± 1.25	28.56 ± 1.36
		(c)	38.57 ± 1.56	32.49 ± 0.97	30.14 ± 0.50	29.40 ± 0.19	28.86 ± 0.31

Case(a): $w_1 = 1000$, $w_2 = 1$, $w_3 = 100$, *Case(b)*: $w_1 = 1000$, $w_2 = 1$, $w_3 = 1$, *Case(c)*: $w_1 = 1000$, $w_2 = 100$, $w_3 = 1$
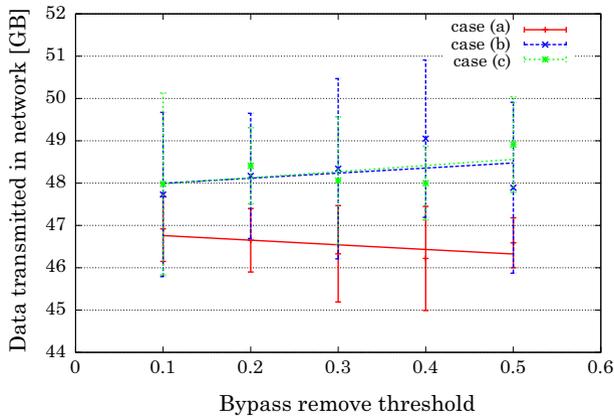
**Fig. 8** Traffic transmitted in the network (scenario 2)
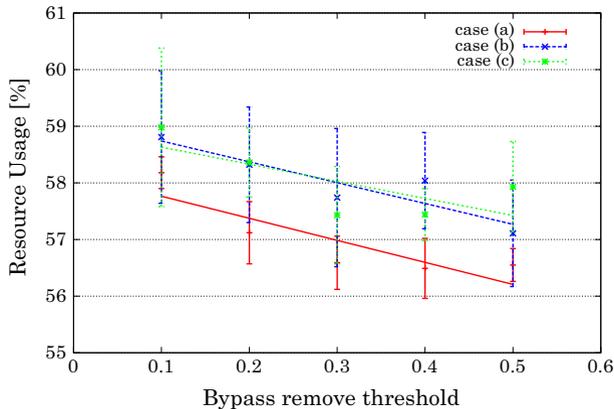
Fig. 9 Resource usage in the network (scenario 2)

controller and routers in the network in one simulation run. This value is negligible when we compare it with the value of transmitted traffic (over 6 GB gain). Moreover, this is the maximum value, and for sure lower amount of traffic was transmitted between the controller and routers in each case. What is very important, we did not observe any unfavorable impact of signalling delay or lack of synchronization among routers on setting up and releasing bypasses.

4 Related Work

There are several papers dealing with multilayer traffic management. Some mechanisms described in this section allow for setting up optical bypasses in both, hidden (not visible at IP layer) and announced (visible at IP layer) versions. In some cases, it is assumed that a centralized controller is used, while others use a
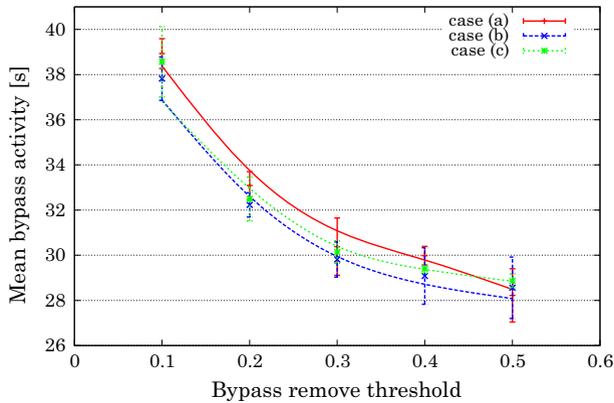
Fig. 10 Mean bypass activity in the network (scenario 2)

distributed approach. Of course, we selected only a fraction of mechanisms presented in the literature. In our opinion, they are the most representative solutions which can be compared to the proposed AHB mechanism. We denoted them as mechanism 1–mechanism 8 to enable presentation of differences between these mechanisms and the AHB approach. The differences are summarized in Table 3.

[Mechanism 1] In [8], the author provides evaluation of four dynamic bypass mechanisms in a network without a centralized controller. The mechanisms are compared to one another and to two reference cases, i.e. networks without an ability to create bypasses. In general, a bypass can be created between a pair of nodes. It can be a one hop or a multihop bypass. Authors evaluate two options: (a) only new connections are routed through the bypass, (b) existing connections might also be rerouted to the bypass. However, it is important to note that the bypasses may not be established optimally, since without a centralized controller it is not possible to choose the best one. The simulation results presented in the paper confirm that the best results were obtained by using multihop bypass mechanisms, albeit at a cost of creating a high number of bypasses and removing operations in the network. The analyzed mechanism operates based on a distributed approach. The advantage over AHB is a minimal signalling and the lack of a single point of failure (controller). On the other hand, less efficient bypasses are established in distributed solutions compared to methods with a central controller. The authors of [9] confirm advantages of centralized solutions over distributed ones regarding efficiency. Moreover, they also confirm that centralized solutions, currently available for implementation, are fast and scalable.

[Mechanism 2] In [10], it is assumed that in a network, each node monitors traffic through a predefined period of time. At the end of the period, if the volume of traffic towards a node exceeds a given threshold, the node may create a bypass and reroute traffic correctly. It is assumed that the information about the new bypass is not announced to a routing protocol. This operation is similar to that proposed in this paper. However, in [10] a decision to create a hidden bypass is made locally in a node and the central controller is not used. The authors also consider bypasses

Table 3 Stronger (+) and weaker (−) sides of each mechanism compared to AHB

Mechanism no.	Stronger and weaker sides
1	(+)(−) Distributed management
	(+) Minimal signalling
	(+) Single point of failure (controller) is not present
	(−) Less efficient bypasses are established
	(−) Traffic in real time is not analyzed
2	(+)(−) Distributed management
	(+) Minimal signalling
	(+) Single point of failure (controller) is not present
	(−) Less efficient bypasses are established
3	(+)(−) Distributed management
	(+) Minimal signalling
	(+) Single point of failure (controller) is not present
	(−) Less efficient bypasses are established
	(−) Peak rates of traffic in network must be known in advance
4	(+)(−) Distributed management
	(+) Minimal signalling
	(+) Single point of failure (controller) is not present
	(−) Less efficient and short bypasses are established
5 and 6	(+)(−) Centralized management
	(−) Complex algorithm to find bypasses
	(−) Whole traffic matrix must be known in advance
7	(+)(−) Centralized management
	(+) Simple algorithm based on users requests
	(−) Less efficient bypasses are established
	(−) Traffic in real time is not analyzed
8	(+)(−) Centralized management
	(+) Can be effectively used in access networks
	(−) Not scalable to huge networks

extended by an end node. If a bypass is extended, information about it is sent to a source node of the bypass. The node then selects streams for a new longer bypass. If the amount of traffic in a bypass is below a given threshold, the bypass is released. The proposed distributed solution is compared to the network with a centralized controller responsible for establishing end-to-end paths. It is shown that, for low to medium amount of offered load, establishing a distributed bypass makes it possible to achieve a higher amount of data switched optically. For a high amount of offered load, both solutions give similar results. However, the centralized approach consumes fewer lightpaths, which are needed to create bypasses. This means that the bypasses are selected more efficiently. For the centralized solution, is was assumed that the traffic demands generated by source nodes were available in a

central database, which is not necessary in our proposal. We monitor traffic load continuously and observe selected flows' transmission characteristics in links which are likely to become congested. As a result, the central controller has full knowledge about flows to set up optimal bypasses. The authors of [10] indicate that the most important advantage of their system is scalability in terms of network size and provisioning time. They also say that centralized algorithms do not scale well as a number of constraints and network dimension increase. This is especially important for short-time-scale traffic variations. Moreover, in their opinion, centralized approaches are too slow. The scalability issue should be considered as one of the most important challenges of controller-based systems to setup bypasses. However, as shown in the next section, such limitations are not observed for our proposal.

[Mechanism 3] The authors of [11] present hidden and announced versions of bypasses. They show three methods for adapting a virtual topology to current needs. In the first method, a virtual topology is built for the network with peak traffic that is known in advance; then, during a low load period, some lightpaths are released to reduce the amount of consumed energy. For the remaining mechanisms, it is assumed that a virtual topology is built for low load and, during peak periods, additional bypasses are added with or without announcing their existence to a routing protocol. The aim is to reduce the amount of energy consumed in the network. In the paper, an evaluation of three approaches to adaptive optical network is provided. The authors show that the hidden bypass mechanisms have the lowest impact on the established topology, since the lowest number of routing changes was observed for this mechanism. The number of topology changes, as is shown, depends on the difference between the peak and low load values. As well as the mentioned studies, the authors compare and contrast mechanisms in a network with very high data streams. For various number of high data flows, capacities requested for flows and demanded paths lengths, the best results—i.e. the lowest number of topology changes and the greatest energy savings—are once again obtained for the network with hidden bypass mechanisms. However, the limitation of the proposed mechanisms is the need to know the flow peak rates in advance. In our solution, the central controller does not need to know peak rates or any other flows characteristic in advance. The necessary data is estimated 'on the fly'. Furthermore, the results presented in [11] confirm that our decision of using hidden bypasses is profitable when considering the number of route changes in electrical domain.

[Mechanism 4] A valuable approach to multilayer FAN is found in [12]. The authors propose to extend the original FAN concept by introducing the option to route traffic flows through a newly established bypass. So far, mechanisms proposed for FAN, e.g. in [13] were focused on IP network layer traffic management. In [12], it is assumed that the optical layer may be used if the IP layer nears congestion. Therefore, when a new request to carry a given flow appears, first a check is made whether the IP layer is able to carry the flow or if it is redirected to the optical layer. At the optical layer, in order to accept the flow, unoccupied wavelength must be available. If such a lightpath is present, three different policies are used to accept the flow for a bypass. They are the newest flow, the most active flow, and the oldest flow. In the first, a new flow is sent through the optical layer if an optical queue is below a given threshold. In the second, the most active elastic flow at the electrical

layer (flow which has the highest number of bytes in the queue) is redirected to the optical layer to make room for future flows at the electrical layer. Finally, in the third policy, the oldest elastic flows are sent through the optical layer. The proposed solutions are examined for a very limited network comprising three nodes. Several UDP and TCP flows are sent through the nodes. In the paper, the goodput, the packet delay and the rejection ratio by the admission control are examined. The best rejection ratio is obtained for the most active and the oldest policies, while the newest policy gives the worse results. Moreover, the lowest delay was observed in the networks with the most active and the oldest policies. This solution is proposed for FAN, but may be extended for other QoS architectures. The main disadvantage is that only one hop bypasses may be set up. In the AHB mechanism, bypasses can be selected more intelligently and can be composed of more links.

[Mechanism 5] The author of [14] proposes and analyzes a dynamic algorithm to find a set of bypasses for the Atlanta 15-node network. It is assumed that the network is periodically updated after each 15–30 min. It is shown that it is possible to find decent network configurations in the network and, at the same time, by using a penalty, the number of reconfigurations remains low. However, as stated in that paper, such an approach requires total network traffic matrix. The heuristic algorithm used for bypass selection is complex. As a result, while in smaller networks the algorithm results can be obtained quickly, in larger networks it needs yet to be evaluated. The solution proposed by us in this paper is scalable and its implementation is not limited. Our algorithm is simple and the results can be obtained quickly even in large networks. The central controller knows the topology and all possible bypasses between any two nodes. The number of bypasses can be high in large network; however, it is not a problem for currently used devices to operate on tables with even millions of entries. Moreover, the controller needs only to keep information on flows transmitted through congested links—total network traffic matrix is not necessary.

[Mechanism 6] In [15] the authors present the concept of automatically switched optical bypasses. The created bypasses are not reported to the IP layer. The central controller is used to decide how the bypasses should be established. The authors use integer linear programming to optimize the decisions taken by the central mechanism. The proposed solution is rather complex. Moreover, to set up a bypass optimally, the exact traffic matrix needs to be known in advance. The authors also attempt to use the tomogravity model to estimate the traffic matrix; however, the computational errors are reported to be around 10%, which may not be acceptable. These problems are not observed in our solution. Information about the traffic is delivered to the controller only when congestions are expected and the bypass selection algorithm is simple.

An example of using the SDN concept in optical networks is presented in [16]. The authors present the centralized management system of optical networks. It is based on the path computation element, which in this proposal is the OpenFlow controller. As a result the optical networks may be controlled in an efficient way and the reliability of transmission is improved. In [17] the authors explain that the implementation of the OpenFlow controller results in an intelligent control plane for optical networks. They also propose a dynamic transparent wavelength-switched

optical network where flexible transmitters and receivers controlled by an OpenFlow-based control plane are used. The analysis presented in the paper proves that the solution improves reliability of transmission and is scalable. The authors of both cited papers do not consider the bypass implementation. In our paper, we propose to use the controller to select a bypass when a link becomes congested. This possibility may additionally improve the effectiveness of the mechanisms presented in [16] and [17].

[Mechanism 7] A unified control plane architecture for optical SDN is proposed in [18]. This architecture, based on OpenFlow, is tailored to cloud services. In this proposal, an SDN controller is able to request a path or compute a new path for flows that need to be transmitted. For such paths, new lightpaths can be established in the optical domain. For example, the Generalized Multi-Protocol Label Switching (GMPLS) control plane can be used for a detailed path computation to set up or tear down lightpaths. Traffic demands are classified based on user requests. The authors of [18] demonstrated their mechanism in a testbed built on ADVA fixed Reconfigurable optical add-drop multiplexers (ROADMs). The experiments were conducted in the University of Bristol laboratory. The results show that paths can be established in the hardware ranging from a few seconds to dozens of seconds including setting up a new lightpath. The operation of the controller takes additional seconds. This solution can be extended by a possibility of setting up bypasses as proposed by us in this paper. The operational time will be extended only by additional selection of optimal bypass. As we show in the next section, this time is short even in large networks.

[Mechanism 8] Originally, SDN was proposed for datacenters. However, possible significant expense reductions and service-oriented income growth caused that SDNs are also attractive for optical solutions. The challenge in this case is a unified control plane which is able to virtualize the network in an intelligent manner and to manage a great number of requests from network elements [19]. The potential benefits of an SDN-based multilayer approach are lower latency and on-demand optical paths setup. To make this possible, the SDN controller must be able to understand and analyze the key abilities of the optical layer. However, as it is shown above, it is possible to set up and tear down optical paths (bypasses) as a result of the SDN controller's demand. The process of selection of optimal virtual subspace is an open challenge which is currently analyzed by standardization bodies, e.g. the Open Networking Foundation (ONF) or the Optical Internetworking Forum (OIF). The authors of [19] proposed an experimentally demonstrated OpenFlow1.0-based flex-grid architecture which is promising for future access networks. In core networks, the scalability problems may occur. However, for our needs, the concept presented in [19] can be extended to enable efficient bypass selection. The key challenge in this case is a method for flow aggregation. Our algorithm is based on traffic analysis in real time and can operate efficiently even in large networks.

As we can see, many mechanisms which use optical bypasses were proposed in the literature. They have advantages and disadvantages over the AHB algorithm. Based on data presented in Table 3, we can conclude that the solution proposed by us is novel and offers possibilities not present in other known approaches. Of

course, some improvements or modifications may be needed before the AHB algorithm could be implemented in real devices. In our future work, we plan to work on possible implementations of the AHB mechanism.

5 Conclusions

Several methods of implementing optical bypasses in multilayer networks have been proposed so far. They assume the use of distributed or centralized algorithms for selecting bypasses. However, the methods are usually complex, or they do not yield satisfactory results.

The method of automatically setting up hidden bypasses proposed in this paper is a promising solution to be implemented in software-defined networks, and it is simple to implement. The simulation results confirm that such a mechanism allows for efficient transmission in a network. The proposed solution has two main advantages: we are able to transmit more traffic, and the resource usage is minimized. Today, with network devices consuming vast amounts of energy, this is an extremely important issue. In our proposal, the unused links (lambdas) are not active until needed. Transmission in a network with hidden bypasses is more intelligently organized. As a result, traffic is transmitted faster and with an insignificant amount of packet losses. The mechanism gives satisfactory results both in low and high loaded networks and for each different traffic type. It should be noted that the AHB mechanism may be the most useful in networks where traffic is generated in an unpredictable way. In our analysis we used links with low capacity. However, the proposed solution is scalable and may be successfully implemented in large networks with high speed links.

References

1. CISCO, Cisco Visual Networking Index: Forecast and Methodology, 2012–2017, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf (2013)
2. Caria, M., Chamania, M., Jukan, A.: Trading IP routing stability for energy efficiency: A case for traffic offloading with optical bypass, in: 15th International Conference on Optical Network Design and Modeling (ONDM), 2011, pp. 1–6 (2011)
3. Vilardi, R., Grieco, L., Barakat, C., Boggia, G.: Lightweight enhanced monitoring for high speed networks. Trans. Emerg. Telecommun. Technol. **25**(11), 1095–1113 (2014)

4. Kortebi, A., Oueslati, S., Roberts, J.W.: Cross-protect: implicit service differentiation and admission control. In: Proceedings of High Performance Switching and Routing, HPSR 2004, Phoenix, AZ, USA, pp. 56–60 (2004)
5. Domzal, J., Wojcik, R., Cholda, P., Stankiewicz, R., Jajszczyk, A.: Efficient Congestion Control Mechanism for Flow-Aware Networks. Int. J. Commun, Syst (2015). doi:10.1002/dac.2974
6. Gero, B.P., Palyi, P.L., Racz, S.: Flow-level performance analysis of a multi-rate system supporting stream and elastic services. Int. J. Commun. Syst. **26**(8), 974–988 (2013)
7. Source code of AHB, http://www.kt.agh.edu.pl/∼jdomzal/bypass/ns-3-bypass.tar.gz (2015)
8. Joachim, S.: Efficiency analysis of distributed dynamic optical bypassing heuristics. In: IEEE International Conference on Communications (ICC), pp. 5951–5956 (2012)
9. Jain, R., Paul, S.: Network virtualization and software defined networking for cloud computing: a survey. IEEE Commun. Mag. **51**(11), 24–31 (2013)
10. Ruffini, M., O'Mahony, D., Doyle, L.: Optical IP switching: a flow-based approach to distributed cross-layer provisioning. IEEE/OSA J. Opt. Commun. Netw. **2**(8), 609–624 (2010)
11. Caria, M., Chamania, M., Jukan, A.: A comparative performance study of load adaptive energy saving schemes for IP-over-WDM networks. IEEE/OSA J. Opt. Commun. Netw. **4**(3), 152–164 (2012)
12. Lopez, V., Cardenas, C., Hernandez, J.A., Aracil, J., Gagnaire, M.: Extension of the flow-aware networking (FAN) architecture to the IP over WDM environment. In: 4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks, pp. 101–106 (2008)
13. Domzal, J., Wojcik, R., Lopez, V., Aracil, J., Jajszczyk, A.: Efmp: a new congestion control mechanism for flow-aware networks. Trans. Emerg. Telecommun. Technol. **25**(11), 1137–1148 (2014)
14. Adibi, S., Erfani, S.: An optimization-heuristic approach to dynamic optical bypassing. In: Photonische Netze (ITG-FB 233) (2012)
15. Chamania, M., Caria, M., Jukan, A.: Achieving IP routing stability with optical bypass. Opt. Switch. Netw. **7**(4), 173–184 (2010)
16. Casellas, R., Martinez, R., Munoz, R., Vilalta, R., Liu, L., Tsuritani, T., Morita, I.: Control and management of flexi-grid optical networks with an integrated stateful path computation element and OpenFlow controller [invited]. IEEE/OSA J. Opt. Commun. Netw. **5**(10), A57–A65 (2013)
17. Liu, L., Choi, H.Y., Casellas, R., Tsuritani, T., Morita, I., Martinez, R., Munoz, R.: Demonstration of a dynamic transparent optical network employing flexible transmitters/receivers controlled by an OpenFlow-stateless PCE integrated control plane [invited]. IEEE/OSA J. Opt. Commun. Netw. **5**(10), A66–A75 (2013)
18. Simeonidou, D., Nejabati, R., Channegowda, M.: Software defined optical networks technology and infrastructure: enabling software-defined optical network operations. In: Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC), 2013, pp. 1–3 (2013)
19. Cvijetic, N., Tanaka, A., Ji, P., Sethuraman, K., Murakami, S., Wang, T.: Sdn and openflow for dynamic flex-grid optical access and aggregation networks. J. Lightwave Technol. **32**(4), 864–870 (2014)

Jerzy Domżał received his M.Sc. and Ph.D. degrees in telecommunications from AGH University of Science and Technology, Kraków, Poland in 2003 and 2009, respectively. Currently, he works as an assistant professor at the Department of Telecommunications of AGH. His current research interests focus on Software-Defined Networks, Flow-Aware Networking and Quality of Service. He is an author or co-author of several journal and conference papers, two books and one patent.

Zbigniew Duliński received the Ph.D. degree in theoretical physics from the Jagiellonian University. His research interests include distributed computing systems, network management mechanisms and traffic engineering for inter-domain traffic, optimization of inter-cloud communication, SDN and NFV application for inter-domain communication.

Jacek Rząsa received an M.Sc. degree in telecommunications in 2001. He has participated in research ordered by telecommunication operators and worked in many international projects. He is author and co-author of several research papers. His research interests focus on energy aware optical transport networks, traffic engineering in optical networks and Carrier Ethernet.

Piotr Gawłowicz received his M.Sc. and B.Sc. in Electronics and Telecommunications from AGH University of Science and Technology, Kraków, Poland in 2012 and 2014, respectively. He works as researcher at TKN Group at TU Berlin, Germany. His research interests include software defined networking, wireless networks and simulation tools.

Edyta Biernacka received the M.Sc. degree in telecommunications from AGH University of Science and Technology, Kraków, Poland in 2012. She is currently working towards the Ph.D. degree in the Department of Telecommunications of AGH. Her research interests include aspects of multilayer optical networks.

Robert Wójcik received his M.Sc. and Ph.D. (with honors) degrees in telecommunications from AGH University of Science and Technology, Kra-ków, Poland in 2006 and 2011, respectively. Currently, he works as an assistant professor at the Department of Telecommunications of AGH. His current research interests focus on Multipath routing, Flow-Aware Networking and Quality of Service. He serves as a technical editor in IEEE Communications Magazine.