A Testbed for Validation and Assessment of Frame Switching Networks

Arthur Mutter[1], Sebastian Gunreben[1],
Wolfram Lautenschläger[2], and Martin Köhn[1]

[1] Universität Stuttgart - IKR, Pfaffenwaldring 47, 70569 Stuttgart, Germany
mutter@ikr.uni-stuttgart.de
[2] Alcatel-Lucent Deutschland AG, Bell Labs, Stuttgart, Germany

Abstract. Packet assembly at the network edge is one solution to reduce high packet rates in core network switches. Literature discusses this topic controversially because of three reasons: (1) potential negative impact of packet assembly on the traffic characteristics, (2) disruptive integration into existing networks and (3) lack of support of packet assembly in existing control plane environment.

In this paper, we introduce our testbed with 10 Gbps packet assembly nodes at the network edge and a GMPLS (Generalized Multi-Protocol Label Switching) control plane for its management. Our testbed integrates transparently in existing Ethernet based networks and allows comprehensive studies on the impact of packet assembly on the traffic characteristics by measurements. Beside feasibility, for early findings, we setup a measurement scenario and quantified the impact of packet assembly in terms of packet latency. For realistic traffic conditions, we found that the additional delays and jitter introduced by the assembly process are negligible.

1 Introduction

The popularity of the Internet causes growth of data volume and interface rates in packet core networks, i.e., 100 Gbps interfaces are currently on the way to standardization [11]. About half of the packets[1] in these networks is smaller than 200 Byte. Reasons are the acknowledgement packets of upper layer transport protocols, like the Transmission Control Protocol (TCP, [1]). With minimum size Ethernet packets, the packet rate on a 100 Gbps link is as high as 149 Million packets per second (preamble and inter-framing gap considered). A 100 Gbps capable network node has to be able to accept and deliver a minimum size packet every 6.72 ns. Current network processors operate at frequencies of several hundred MHz which translates to clock periods of 1 to 5 ns. This means a network processor has to accept and deliver a packet every few clock cycles. This puts a high burden on the realization. Consequently, higher data rates dramatically increase the processing and switching effort within core switches and raise their complexity.

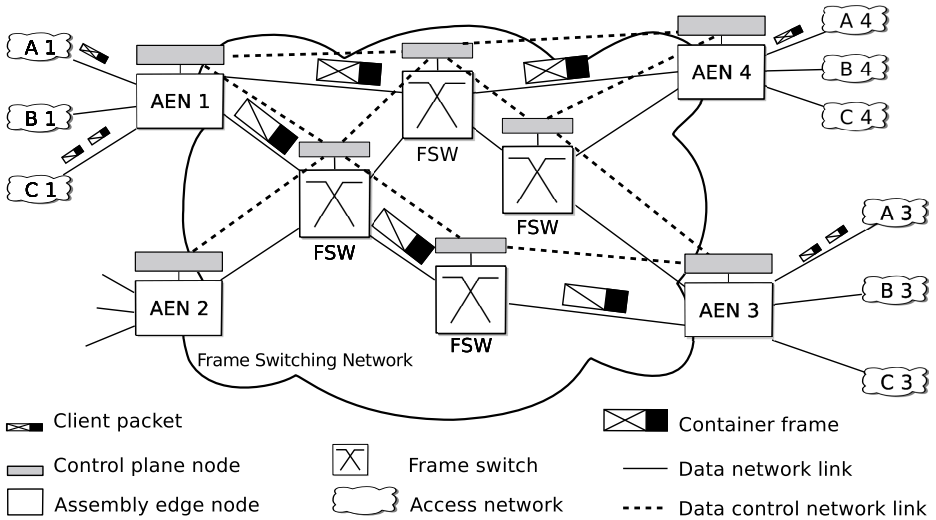[1] caida.org, the Cooperative Association for Internet Data Analysis, 2008.

Fig. 1. Frame Switching network architecture

One approach to alleviate this problem is to reduce the number of packets in these networks, i. e., to reduce the packet rate. A prominent solution is the assembly of client layer packets into server layer containers at the network edge. The Frame Switching (FS) architecture relies on this principle [19]. FS is a connection oriented packet switching (CO/PS) network technology for metro and core networks. Fig. 1 depicts a FS network. It consists of edge nodes, namely Assembly Edge Nodes (AEN) and core nodes, namely Frame Switches (FSW). A path through the network includes two AENs – one at each end – and a sequence of FSWs between them. The CO character of this technology is ensured by a preceding connection setup phase. The signaling phase propagates traffic engineering properties along the path, e. g., relative Quality of Service (QoS) and required bandwidth.

At network edge, the AEN assembles in ingress direction multiple client packets of the same forward equivalent class (FEC) into containers and forwards these containers to the next FSW according to the next hop of the signaled path. Packets of one FEC belong together according to a superior order, e. g., belong to the same flow or need to traverse the same target AEN. The FSWs forward these containers according to a forwarding table and maintain relative QoS among different paths. The egress AEN disassembles the containers to individual packets and forwards them individually towards their destination client.

FS avoids any cross layer interactions and operates on layer 2 transparently to upper layer protocols. Because of the layering, we use the term *frame* for the containers carrying assembled packets. The originally foreseen layer 2 technology for FS networks was an extended Optical Transport Network (OTN, [12]). The server layer frame then corresponds to a G.709 frame [13]. However, the concept of packet assembly at the network edge is also applicable to other layer 2

technologies. The most prominent layer 2 technology for cheap and simple implementation is Ethernet [8].

Packet assembly maps multiple smaller packets into larger frames. In case of fixed size frames, packet segmentation is necessary to provide 100 % fill ratio. Additionally, in case of low load situations and fixed frame sizes, also padding is required. Besides segmentation and padding, frame assembly at the network edge changes the original traffic characteristic of assembled packets. Fig. 2 depicts this property. Arriving packets show a certain inter-arrival time characteristic (left side of Fig. 2). The AEN assembles these packets into one frame (middle). At the egress side, the AEN disassembles the frame to packets and releases them in a back-to-back manner without the original inter-arrival time characteristic (right side of Fig. 2).

It is assumed that the additional latency or jitter introduced by the assembly process leads to temporary buffer overflows at the network edge due to the burstification of the traffic or affect the upper layer protocol performance negatively. These assumptions are the major argument against any packet assembly technology in packet based networks. On the other hand, these assumptions are hard to quantify without a real network environment and real network conditions. Another open point of packet assembly networks is their controllability as packet assembly requires additional signaling because of the assembly process. This missing support of packet assembly in present control plane protocols leads to additional skepticism.

In this paper, we present, to the best of our knowledge, the first time a FS testbed including both the data plane and the control plane (CP). Our FS testbed enables studies on both topics, i. e., the packet assembly process and its impact on the traffic characteristics as well as the controllability of these networks. Our testbed relies on Ethernet technology and includes bidirectional AENs. Our AEN prototype supports various packet assembly schemes, i. e., combined timer and threshold based assembly. The resulting frames may be of fixed or variable size supporting packet segmentation and padding. The testbed and the AEN operate at 10 Gbps. Additionally, our testbed implements a CP for path management. Our CP bases on the DRAGON Generalized Multi-Protocol Label Switching (GMPLS, [30]) implementation and shows extensions to support the packet assembly functionality of FS networks. This especially includes the signaling protocols to signal service classes and the parameters of the assembly scheme.

1.1   Related Work

Considering the data plane, literature presents several architectures and implementations of assembly nodes as well as testbeds within the context of Optical Burst Switching (OBS) networks [21, 24, 32]. They focus in their studies mainly on technological problems and less on the realization of packet assembly. Thereby they support only variable length bursts and thus do not consider packet segmentation or padding. Additionally to the optical layer, Kögel et al. implement in [16] an assembly node on a network processor. However, on network processors, scalability and throughput are limited due to the software implementation.

In our earlier publication [23], we present our assembly edge node, exclusively focusing on the data plane part. We show its architecture and give detailed information about its implementation and supported features.

Kornaros et al. propose in [15] an assembly node architecture based on G.709 frames. For a detailed comparison to our assembly edge node we refer to [23]. However, the authors in [15] do neither report on any control plane interface for extensive path management support nor describe a testbed for validation and testing.

Selected testbeds related to the control plane of FS networks apply the GM-PLS protocol suite. This includes both, Optical Burst Switching networks, e. g., [26, 29] as well as transport networks, e. g., [22]. The focus on the control plane studies in OBS lies on the interaction of both control planes: connection-oriented GMPLS CP for lightpath reservation and CPs for connection-less OBS for temporary lightpath occupancy. However, the frame switching network is connection oriented while OBS is connectionless and thus may apply any GMPLS CP directly. Additionally, none of the proposals in literature realizes the parameterization of the assembly edge node via the control plane. The focus on the control plane studies for transport networks lies on the support of optical technologies, inter-domain and multi-layer issues. These studies neither focus on the support of assembly units nor its impact on the control plane.

Summarizing, literature intensively studies the field of packet assembly on selected aspects. However, none of these studies provide a testbed, which (a) transparently integrates in existing networks, (b) provides equivalent capabilities for packet assembly strategies or (c) supports network operation by a standard compliant decentralized control plane.

1.2   Organization of the Paper

In Section 2, we introduce the principles of our testbed architecture. In Section 3, we introduce the prototype architecture and functionality of our frame assembly unit. Section 4 presents the control plane architecture of the frame switching network. Besides the functional validation of the testbed, we provide preliminary measurements to quantify the impact of the assembly process on the traffic characteristics in Section 5. Section 6 summarizes our work.

2   Introduction to Our Frame Switching Testbed

Our FS testbed uses Ethernet as the transport technology for the FS network between two AENs, cf. Fig. 1. We use Ethernet jumbo of fixed size (9 KByte) as container frames to transport the assembled packets. For easy and transparent integration in existing networks, the AEN also provide Ethernet interfaces to the client networks at the edge. The frame switches (FSW) along a path from one AEN to the destination AEN are manageable Ethernet switches. The following paragraphs detail the principles of the FS testbed architecture.
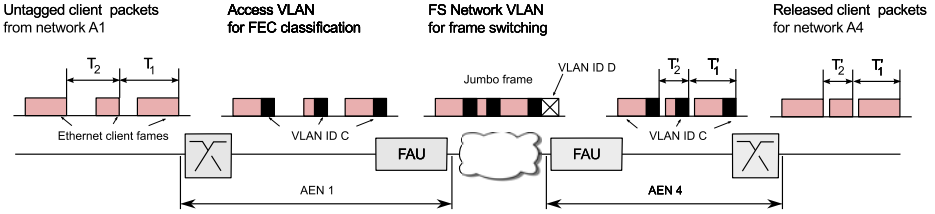
Fig. 2. Packet classification and assembly

The AEN shows two independent functions: switching and assembly. Therefore, the realization of an AEN shows two options. (1) One monolithic device incorporating both functionalities or (2) a modular device separating the functionalities. The advantage of (1) is the single device and potential resource savings due to common usage of components, e.g., a common packet buffer. However, the common components have to suffice a larger number of requirements and are therefore more complex.

We follow the modular approach (2) and separate the functionalities. Fig. 3 depicts our modular AEN architecture. It shows one Ethernet switch and multiple Frame Assembly Units (FAU). Both are interconnected to the control plane via a Control Channel Interface (CCI). Our modular approach is highly flexible as it allows adding FAU ports to an AEN incrementally.

The Ethernet switch serves two purposes. (a) Classification of packets on a per port basis, i.e., packets on one port belong to one FEC. (b) Switching of classified packets to the correct FAU.

The classification process at the AEN Ethernet switch assigns all packets on certain ports to one Virtual LAN (VLAN, cf. Ethernet VLAN extension of [7]). Packets that belong to the same VLAN have the same VLAN identifier (VLAN ID). The specific VLAN ID is selected in the connection setup procedure (cf. Section 4). The selected VLAN ID per port serves as a classification criteria for the subsequent FAU. The FAU assembles packets per FEC (i.e., per VLAN ID) and creates frames (cf. Section 3). The switch in the egress AEN will also use this VLAN ID to switch the packets to the appropriate client network. Additionally, the utilization of VLAN at this point enables to decouple the classification process in the FAU from the Ethernet MAC addresses of the clients. This drastically simplifies configuration.

Ethernet is a connectionless packet-switching technology with a separate control plane (Spanning Tree Protocol, [6]) for link management. This forbids any traffic engineering in Ethernet based FS networks. The virtual LAN extension of Ethernet (VLAN, [7]) alleviates this problem and provides connections to FS networks. In the core one connection corresponds to one VLAN. The ports of the FSW define the path of a connection, i.e., two ports per FSW along the path belong to the same VLAN. Please note that the VLAN ID of the classification process is not necessarily the same as the VLAN ID in the FS network. Ethernet limits the number of VLAN to 2^{12}. This restricts the application of this concept in large networks. However, the Provider Bridge (PB, [9]) and the Provider
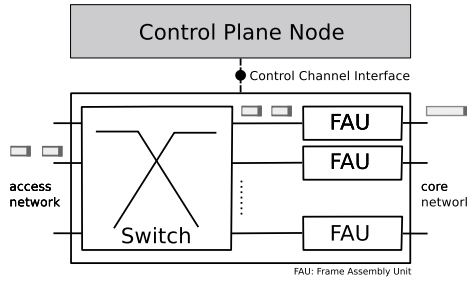
Fig. 3. Architecture of the assembly edge node (AEN)

Backbone Bridge (PBB, [10]) proposals alleviate this limitation by additional Ethernet VLAN and frame headers, respectively. Consequently, this paper does not elaborate on this problem.

Fig. 2 depicts from left to right the whole classification, encapsulation and forwarding process from the connection A1 to A4 of Fig. 1. Additionally, it distinguishes the different VLAN IDs. The Ethernet packets of the client network A1 arrive at the AEN 1 switch. The AEN 1 switch classifies these packets per port and assigns these packets the VLAN ID C. The VLAN tagged packets leave the AEN 1 switch and face the FAU for packet assembly. The FAU assembles packets of one FEC (i. e., same VLAN ID) to Ethernet jumbo frames of 9 KByte. The FAU assigns these Ethernet jumbo frames a VLAN ID (here VLAN ID D), which is used in the FS network to switch the jumbo frames. The FAU in the destination AEN 4 disassembles the Ethernet jumbo frames. This FAU forwards the original client packets including the classification VLAN ID (VLAN ID C) to the AEN 4 switch. The switch removes the classification VLAN ID and forwards the individual packets to the client network A4.

The big advantage of this MAC in MAC encapsulation is the transparency to any upper layer protocol. For instance, ARP (Address Resolution Protocol, [27]) and IP (Internet Protocol, [28]) protocols are transparently switched without interfering with the FS transport layer.

Additionally, each connection belongs to a selected class of service. The FS testbed realizes service differentiation on a Differential Service basis (DiffServ, [3]). The information on the service class is included in the header of the assembled frame (3 Bit user priority of the VLAN header, [7]). The Ethernet switches in the FS network (i. e., FSW) use these priority bits to relatively prioritize the jumbo frames.

The next sections introduce the architecture and implementation of the FAU as well as the implemented control plane instance to manage this testbed.

3   FAU Prototype Architecture

The FAU assembles packets to frames in ingress direction and disassembly frames to packets in reverse direction. Fig. 4 depicts the ingress direction of the FAU prototype architecture. The lower part shows the pipelined architecture of the
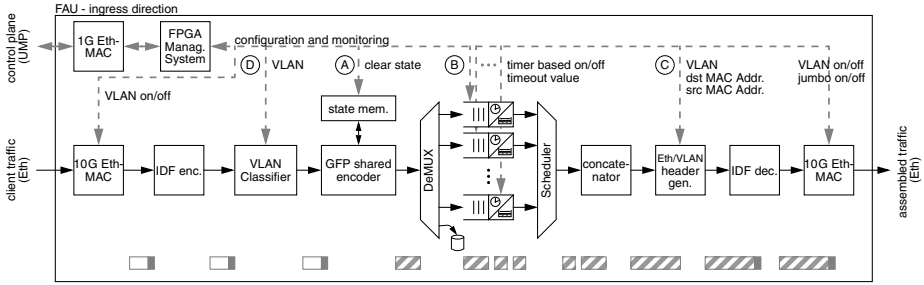
Fig. 4. Architecture of FAU prototype: ingress direction

packet processing. The upper part shows the interface to the control plane for runtime configuration. The following two sections describe both parts in detail.

We realized the FAU on an evaluation board with two Xilinx Virtex-4 FX100 FPGAs, two optical 10 Gigabit Ethernet interfaces for data plane and a 1 Gigabit Ethernet interface for CP connection. We designed the FAU prototype in VHDL supporting simultaneously seven FECs per direction.

3.1   Data Plane Part

The description in this section introduces the data plane part of the ingress direction of the FAU. For a more detailed introduction the reader is referred to our earlier publication [23].

The description of Fig. 4 follows the assembly process from left to right, i. e., from an access to a core network. The first stage, the 10 G Ethernet interface, receives packets, checks their frame checking sequence (FCS) and forwards them without the FCS. The next stage converts the packets into an internal data format (IDF) and therefore adds an IDF header.

The VLAN classifier classifies client packets according to their VLAN ID (12 Bit field in the VLAN header). Packets belonging to one VLAN belong to the same forwarding equivalent class (FEC). The VLAN classifier stage stores the information to which FEC a packet belongs in the IDF header.

The next stage encodes packets with the ITU-T Generic Framing Procedure (GFP, [14]). GFP encapsulates each packet by adding meta information (the GFP core and payload headers) that allows delineation at the destination. GFP supports packet segmentation as the delineation process relies only on the meta information and is independent of container frame boarders. Additionally, GFP operates stateful per FEC. Consequently, a new connection requires a clear state of the GFP encoder before operating.

The assembly stage consists of many assembly components. Each component serves one FEC. Packets not assigned to any FEC are dropped here. Each assembly component supports combined timer and threshold based packet assembly. The buffer of each assembly component collects packets of one FEC. In every component a control block monitors the fill level of this buffer. When the size threshold is exceeded or in case of timeout, the control block signals the scheduler

that a frame is ready to be sent. Since frame switching uses constant size frames, the assembly components segment packets to fill frames completely. Further, it appends padding if the amount of collected packet data is below the frame size.

If more than one assembly component signals a ready frame, the scheduler assigns the outgoing link in Round Robin manner. On request of the scheduler the assembly component transmits the GFP encoded packets or segments of a packet to the concatenator stage to build the frame. The concatenator stage removes the IDF headers of the GFP encoded packets, aligns the data and creates a continuous data block, which represents the frame's payload.

The next stage finalizes the Ethernet jumbo frame by adding to the payload an Ethernet and a VLAN header according to the frame's FEC. The IDF decoder stage removes the IDF header of the frame. The last stage, the 10 G Ethernet interface, appends the jumbo frame a FCS and transmits it to the core.

The data plane part of the egress direction is similar. [23] shows it in detail.

3.2   Control Plane Part

At several stages of the data path, the FAU requires status information and configuration data. For runtime configuration, the FAU foresees a management interface for interactions with the control plane. We developed an FPGA Management System (FMS) as CP interface. Communication with the CP is realized by utilizing an additional 1 Gigabit Ethernet interface (cf. Fig. 4). The FMS enables online read and write access to register and memory values via our self-developed low-level configuration protocol UMP (Universal Hardware Platform Management Protocol). UMP resides on address value pairs.

After powering up the testbed, the CP configures the 10 Gigabit Ethernet interfaces of each FAU to enable support for Ethernet VLAN header extension as well as Ethernet jumbo frames. Additionally, it assigns to each Ethernet interface a unique MAC address that is used as source address in the Eth./VLAN header generator stage. The number of simultaneously supported connections (i. e., FECs) is limited by the available hardware resources in the device. A new connection requires the necessary configuration steps to be done in a fixed order. The setup requires in the ingress FAU the following two steps to be done for the specific FEC:

1. Configure individual stages in the FAU pipeline.
   – Clear state of this FEC in GFP encoder stage (A in Fig. 4).
   – Enable or disable timer-based assembly in the corresponding assembly component. If enabled, then set the timeout value for the timer in units of 10 ns (B in Fig. 4).
   – Set VLAN ID, VLAN priority and Ethernet destination address in the Eth./VLAN header generation stage for this FEC (C in Fig. 4).
2. Set the VLAN ID in the VLAN classifier stage for this FEC (D in Fig. 4).

The order of these two steps is mandatory as the last step acts like a gatekeeper for the incoming packets. Before the first packet enters the FAU, the FAU has to be configured properly. After this step, the classifier labels packets for a certain

FEC. To tear down this connection the control plane has again to keep a fixed order of the necessary two steps. The classifier step first stops incoming packets, while the second step empties the assembly component of this FEC.

1. Remove the VLAN ID mapping of this FEC in the VLAN classifier (D in Fig. 4).
2. Enable timeout based assembly in this FEC to transmit all remaining data in case of only threshold based assembly (B in Fig. 4).

In the egress FAU similar configuration steps are required for connection setup and tear down.

4   Control Plane for Frame Switching Networks

This section introduces the control plane for frame switching networks as well as the control plane implementation of our testbed. The major requirements on any control plane are traffic engineering capabilities and path management support. The GMPLS CP architecture [30] is a promising candidate to control future networks and new network technologies. The CP of the FS testbed resides on the GMPLS framework of the US project DRAGON and has been extended to support FS networks. Extensions include the signaling protocols as well as the interfaces between the control plane and the data plane node. The first section details the DRAGON GMPLS CP. The second section introduces both control plane extensions.

4.1   The DRAGON GMPLS Control Plane

The FS CP resides on the GMPLS implementation of parts of the US project DRAGON (Dynamic Resource Allocation via GMPLS Optical Networks, [18]). This project targeted to support non-GMPLS capable network nodes like Ethernet switches, optical and Time Division Multiplex (TDM) devices. Fig. 5 shows our setup. In the upper part are the elements of the DRAGON CP: The Virtual Label Switching Routers (VLSR), the DRAGON User Network Interface (UNI) and the Network Aware Resource Broker (NARB, [34]).

The NARB enables constraint based path computation and interacts with other NARB elements in inter-domain scenarios. For instance, constraint path computation occurs if a connection request already includes the VLAN ID along the path. For non-constraint path computation, NARB selects one VLAN ID among the announced. The NARB partly fulfils the functionality of the IETF Path Computation Element (PCE, [5]).

The DRAGON UNI provides the interface for rudimentary path management. It enables setup and teardown of paths (VLAN connection). It triggers the path computation and signaling steps, which configure the switch ports appropriately. The DRAGON UNI implements the Optical Internet Forum interface (OIF 1.0, [25]) and realizes an UNI for an overlay model with a GMPLS environment as described in [33]. Additionally for manual operation, the UNI implements a command line interface (CLI) to manage connections.
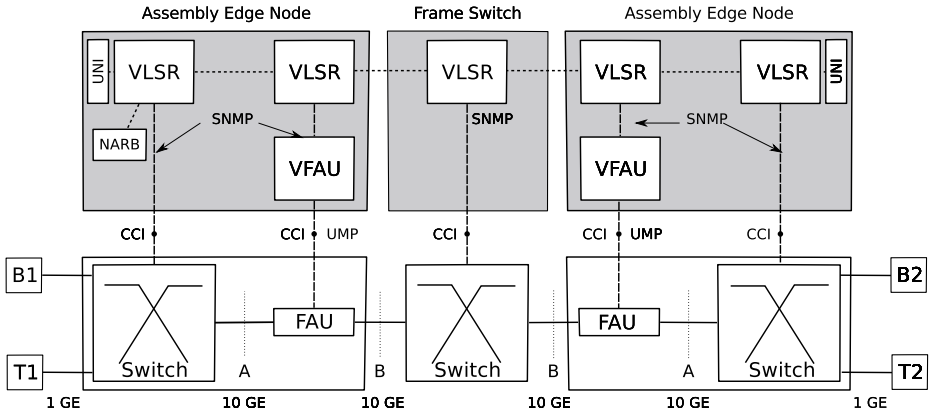
Fig. 5. Frame Switching Testbed Architecture

The VLSRs represent the non GMPLS-capable AENs (Switch, FAU) and the FSW in the CP and operate signaling and routing protocols, i.e., the Resource Reservation Protocol (RSVP-TE, [33]) and the Open Shortest Path Protocol (OSPF, [31]). On the control channel interface, VLSR and AEN/FSW exchange configuration data, i.e., requesting or changing the device state. The communication protocol is the Simple Network Management Protocol (SNMP, [4]). The manageable switches (in the AEN and FSW) support SNMP and implement the manageable objects for Ethernet VLAN extensions according to [2]. In detail, the VLSR performs the following actions on the CCI:

– Query of available or used VLANs per port.
– Assignment of ports to a VLAN.
– Removal of ports from a VLAN.
– Change of port state with respect to trunk and access ports.

While the manageable switches implement the corresponding MIB (Management Information Base) and the SNMP application interface, the FAU does not. The FAU implements the proprietary Ethernet based UMP protocol of Section 3.2. For easy integration in the framework (i.e., without changing the interface to the VLSR), Section 4.2 introduces our gateway, which translates SNMP messages to the proprietary UMP protocol.

4.2 Extensions to the DRAGON GMPLS Control Plane

We extended the DRAGON GMPLS control plane to support the parameters of the FAU, i.e., assembly scheme configuration and class of service (CoS). This includes changes in the signaling protocol RSVP and requires an application layer gateway at the CCI between VLSR and FAU. The following sections address these two extensions in detail.
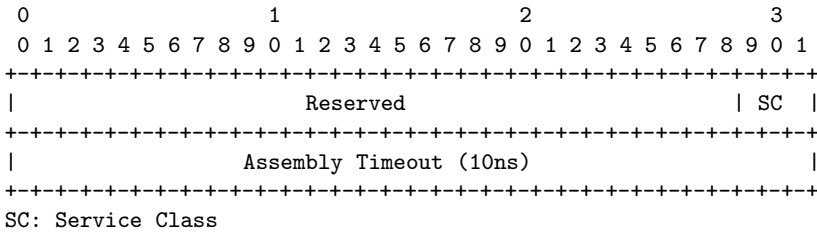
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Reserved                       | SC  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Assembly Timeout (10ns)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
SC: Service Class
```

Fig. 6. Extended ATM service class object, [20]

Additional signaling parameters. The path setup procedure in our testbed requires signaling of the VLAN ID, CoS and the parameters of the assembly process. The VLAN ID corresponds to the signaling of labels in a GMPLS framework and the signaling protocol RSVP. The GMPLS control plane inherently supports this without any modification. However, RSVP does not support directly any CoS and the assembly parameters. To signal these parameters we implemented a modified RSVP ATM service class object, [20]. The ATM Service Class object extends the RSVP-TE Path message and includes 3 Bit to signal ATM service classes for any label switched path, i. e., UBR (Unspecified Bit Rate), VBR-NRT (Variable Bit Rate, Non-Real Time), VBR-RT (Variable Bit Rate, Real Time), CBR (Constant Bit Rate). As the aim is exactly the same as for FS networks, we reuse this object for the FS CoS. The 3 Bit for the ATM service classes perfectly correspond to the 3 Bit VLAN priority of the VLAN header.

We add the information on the assembly timeout parameter to the ATM service class object. Therefore, we extended the original ATM service class object by another 32 Bit indicating the timeout parameter in units of 10 ns. A value of 0 indicates pure threshold based assembly. Fig. 6 depicts the new object. We included this modified object in the RSVP protocol and were able to signal service class and assembly parameterization along the path. The assembly parameter of the size threshold is implicitly given by the constant frame size of Ethernet jumbo frames, i. e., 9 KByte.

Besides the extension of the signaling protocol RSVP, we also extended the UNI to include CoS information and the parameterization of the assembly timeout. These extensions include mainly the modification of the command line interface of the DRAGON UNI as well as the interface to the signaling protocol. As both do not affect standardized protocols, this paper skips the details here.

Control channel interface. The DRAGON VLSR implements the SNMP protocol at the interfaces to the switches and the FAU. Section 3 already introduced that the FAU does not support the high level language SNMP. Instead, it provides a lower level language to access registers and the memory of the FAU. This discrepancy requires a protocol gateway, which translates high level SNMP messages into low level UMP messages. Fig. 7 depicts schematically the gateway, i. e., the Virtual Frame Assembly Unit (VFAU).
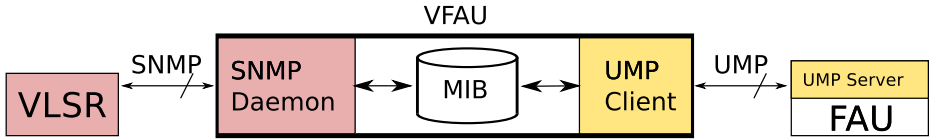
Fig. 7. Virtual FAU (VFAU)

On the interface towards the VLSR, the VFAU emulates a simple two port switch. For communication compatibility, the VFAU implements a SNMP daemon. For VLAN application in Ethernet networks, the VFAU implements the required Management Information Database (MIB, [2]) as the other switches in this network, too. On the interface towards the FAU, the VFAU implements a UMP Client. The UMP Client serves as a backend of the MIB. SNMP Messages, which operate on MIB entries (cf. Section 4.1), automatically trigger the corresponding UMP messages to put this in action on the FAU. The UMP messages include the necessary configuration data on the FAU, cf. Section 3.

Besides these configuration messages, the gateway also manages the resource state of the FAU. Each FAU is able to handle a limited number of FEC per direction, i. e., in our implementation we support 7 per direction. If the number of available resources is exhausted, the gateway signals this information to the VLSR and prohibits additional path setups until paths terminate. The information on the resource occupation state is configured in the FAU and stored and updated in the gateway.

5 Measurement Results

Our testbed in Fig. 5 and its realization in Fig. 8 not only demonstrate the feasibility of frame switching networks, but also enable studies on the impact of packet assembly. For validation of the functionality of the testbed and for preliminary results regarding packet assembly, we setup a measurement environment. Our testbed consists of three AENs (Fig. 5 shows only two AENs because of space limitations) and one core switch representing the FS core network.

The inherent burstification of the transported traffic is one of the major sources of skepticism by practitioners and network operators. The frame assembly may delay packets until the end of the assembly process (time/size threshold reached) at the assembly edge node. This waiting time follows a load dependent distribution, which leads to an additional jitter. These changes in traffic characteristics may cause problems on higher layers. Nonetheless, our preliminary results indicate that the additional delay introduced by the frame assembly is well controlled and remains small enough compared to other delay sources, e. g., queuing or propagation delay.

For our study, we used the setup of Fig. 5. We study the impact of frame assembly on a test flow in the presence of a background flow. The background flow represents the aggregated traffic of several users. It originates at B1 and
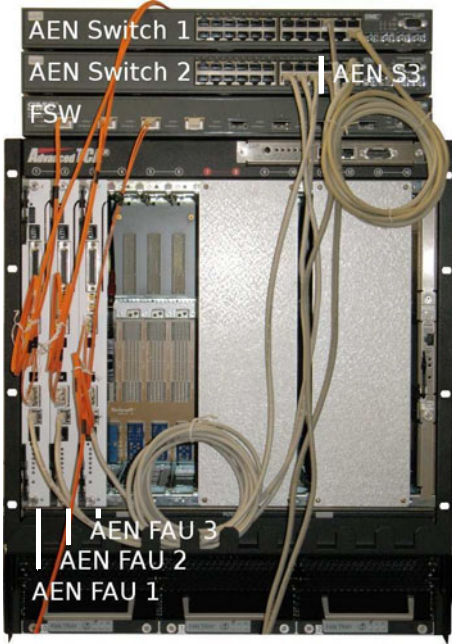
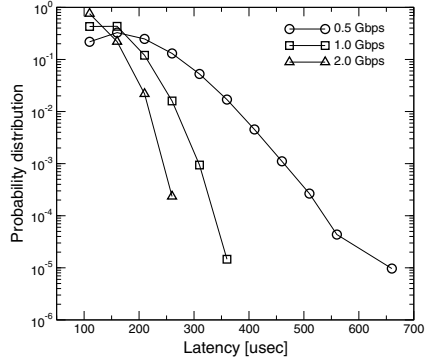**Fig. 8.** Frame Switching Testbed

Fig. 9. Packet latency

terminates at B2. The test flow represents the traffic of one user or application. It originates at T1 and terminates at T2. Both flows share the same FEC and thus share the same resources in both FAUs. The target of our study is the experienced latency of the test flow with respect to a changing background flow.

The background traffic has an average rate of 0.5, 1, or 2 Gbps, respectively. We compose the background traffic by an overlay of randomly arriving 10 Mbps application streams and use the same traffic model as in [17]. The average rate of the test flow is 10 Mbps. It is composed by 500 Byte packets showing a constant inter-arrival time. At T2, we record the latency of the packets after traversal of the testbed. We use only threshold-based assembly with fixed size (9 KByte) jumbo frames. The timer-based assembly is switched off.

Fig. 9 shows the empirical probability distribution of the latency of the test flow packets with respect to different background load levels. All three curves show a constant offset of about 110 μsec propagation delay.

The maximum of each distribution shifts reciprocally with the traffic load, i. e., the distribution moves to the right with decreasing traffic load. These findings correspond to a load dependent waiting time during frame assembly. In low load situations, the latency shows a rather long tail, because only threshold based assembly scheme is used. However, any timer-based assembly would limit the maximum additional delay and cut the distribution.

Finally, we point out the small absolute values of the measured jitter. With our realistic traffic conditions, we observed jitter values far below 1 ms. As each packet receives this jitter only once at the network ingress. So the jitter does not accumulate with the network size. Consequently, we do not expect these jitter values to significantly influence the higher layers.

6   Conclusion

The contribution of our paper is twofold. We show the feasibility of frame switching networks and provide a testbed, which enables comprehensive studies on the impact of packet assembly on the traffic characteristics.

For transparent operation and easy integration in existing networks, our testbed resides on connection-oriented Ethernet and operates on 10 Gbps. The two major achievements of our testbed are the assembly edge nodes and a GMPLS control plane. The assembly edge node consists of a switching device and the frame assembly unit, which assembles packets to larger containers, named frames. The GMPLS control plane is responsible for path management.

Our frame assembly unit supports timer and threshold based assembly, fixed and variable size frames, packet segmentation for a 100 % frame fill ratio and the ITU-T Generic Framing Procedure for packet delineation. The architecture is modular to ease design adaptation to any packet oriented transport technology. We implemented the design in VHDL to show its feasibility and validated its functionality within our testbed.

Our control plane for Frame Switching networks applies the GMPLS control plane implementation of the DRAGON project. We extended the signaling protocols and the UNI interface to adapt to the special requirements of frame assembly at the network edge, i. e., QoS classes and assembly scheme parameters. The resulting control plane implementation is capable to setup and tear down end-to-end connections of different service classes between access networks. Each connection request includes the parameterization of the frame assembly units as well as of the intermediate switches.

For preliminary results, we setup a network scenario with realistic traffic conditions and measured the impact of the packet assembly scheme in terms of packet latency. We found that the absolute delay and jitter introduced by threshold based assembly is far below 1 ms. However, any timer-based assembly further bounds this delay. Consequently, we do not expect these delays to significantly influence the higher layers.

In our ongoing studies, we extend the measurement scenario towards additional traffic characteristics and load scenarios. Additionally, tests to measure the impact on transport and application layer protocols are planed.

Acknowledgement

References

1. Allman, M., Paxson, V., Stevens, W.: TCP Congestion Control. RFC 2581, IETF (April 1999)
2. Bell, E., Smith, A., Langille, P., Rijhsinghani, A., McCloghrie, K.: Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions. RFC 2674, IETF (August 1999)
3. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., W.Weiss.: An Architecture for Differentiated Service. RFC 2475, IETF (December 1998)
4. Case, J.D., Fedor, M., Schoffstall, M.L., Davin, J.: Simple Network Management Protocol (SNMP). RFC 1098, IETF (April 1989)
5. Farrel, A., Vasseur, J.-P., Ash, J.: A Path Computation Element (PCE)-Based Architecture. RFC 4655, IETF (August 2006)
6. IEEE Computer Society. 802.1D: IEEE Standard for Local and Metropolitan Area Networks–Media Access Control (MAC) Bridges (2004)
7. IEEE Computer Society. 802.1Q: IEEE Standard for Local and Metropolitan Area Networks–Virtual Bridged Local Area Networks (2005)
8. IEEE Computer Society. 802.3: IEEE Standard for Local and Metropolitan Area Networks–Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications (2005)
9. IEEE Computer Society. 802.1ad: IEEE Standard for Local and Metropolitan Area Networks–Virtual Bridged Local Area Networks, Amendment 4: Provider Bridges (May 2006)
10. IEEE Computer Society. 802.1ah: Draft Standard for Local and Metropolitan Area Networks–Virtual Bridged Local Area Networks, Amendment 6: Provider Backbone Bridges (November 2007)
11. IEEE Computer Society. P802.3ba: IEEE Standard for Local and Metropolitan Area Networks–Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Amendment: Media Access Control Parameters, Physical Layers and Management Parameters for40 Gb/s and 100 Gb/s Operation (2008)
12. ITU-T. Framework of Optical Transport Network Recommendations. Rec.G.871/Y.1301, ITU-T (October 2000)
13. ITU-T. Functional architecture of connectionless layer networks. Rec. G.809, ITU-T (March 2003)
14. ITU-T. Generic framing procedure (GFP). Rec. G.7041/Y.1303, ITU-T (August 2005)
15. Kornaros, G., Lautenschlaeger, W., Sund, M., Leligou, H.-C.: Architecture and implementation of a frame aggregation unit for optical frame-based switching. In: International Conference on Field Programmable Logic and Applications 2008, vol. 642, pp. 639–642 (September 2008)
16. Kögel, J., Hauger, S., Junghans, S., Köhn, M., Necker, M.C., Stanchina, S.: Designand Evaluation of a Burst Assembly Unit for Optical Burst Switching on a Network Processor. In: EUNICE 2005: Networks and Applications Towards a Ubiquitously Connected World (July 2006)
17. Lautenschläger, W., Frohberg, W.: Bandwidth Dimensioning in Packet-based Aggregation Networks. In: 13th International Telecommunications Network Strategy and Planning Symposium, Networks2008, Budapest (2008)
18. Lehman, T., Sobieski, J., Jabbari, B.: Dragon: a framework for service provisioning in heterogeneous grid networks. IEEE Communications Magazine 44(3), 84–90 (2006)

19. Leligou, H.C., et al.: Hybrid burst/packet switching architectures from IP NOBEL. In: Dingel, B.B., et al. (eds.) Optical Transmission Systems and Equipment for Networking, vol. 6388, p. 63880C. SPIE, San Jose(2006)
20. Malis, A.G., Hsiao, T.: Protocol Extension for Support of Asynchronous Transfer Mode (ATM) Service Class-aware Multiprotocol Label Switching (MPLS) Traffic Engineering. RFC 3496, IETF (March 2003)
21. Masetti, F., et al.: Design and implementation of a multi-terabit optical-burst/packet router prototype. In: Optical Fiber Communication Conference and Exhibit, pp. FD1-1–FD1-3 (March 17-22, 2002)
22. Munoz, R., Pinart, C., Martinez, R., Sorribes, J., Junyent, G., Amrani, A.: The ADRENALINE testbed: integrating GMPLS, XML, and SNMP in transparent DWDM networks. IEEE Communications Magazine 43(8), s40–s48 (2005)
23. Mutter, A., Köhn, M., Sund, M.: A generic 10 Gbps assemblyedge node and testbed for frame switching networks. In: Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, TridentCom 2009 (2009) (page accepted for publication)
24. Nejabati, R., Klonidis, D., Simeonidou, D., O'Mahony, M.: Demonstration of user-controlled network interface for subwavelength bandwidth-on-demand services. In: Optical Fiber Communication Conference, vol. 3 (March 6-11, 2005)
25. Optical Internetworking Forum (OIF). User network interface (uni) 1.0 signaling specification (oif-uni-01.0) (October 2001)
26. Pedroso, P., et al.: An interoperable GMPLS/OBS control plane: RSVP and OSPF extensions proposal. In: CNSDSP 2008, pp. 418–422 (July 2008)
27. Plummer, D.: Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware. RFC 826, IETF (November 1982)
28. Postel, J.: Internet Protocol. RFC 791, IETF (September 1981)
29. Rajaduray, R., Ovadia, S., Blumenthal, D.J.: Analysis of an edge router for span-constrained optical burst switched (obs) networks. Journal of Lightwave Technology 22(11), 26–93 (2004)
30. Generalized Multi-Protocol Label Switching (GMPLS) Architecture. RFC 3945, IETF (October 2004)
31. OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS). RFC 4203, IETF (October 2005)
32. Sun, Y., Hashiguchi, T., Minh, V.Q., Wang, X., Morikawa, H., Aoyama, T.: Design and implementation of an optical burst-switched network testbed. IEEE Communications Magazine 43(11), S48–S55 (2005)
33. Swallow, G., Drake, J., Ishimatsu, H., Rekhter, Y.: Generalized Multiprotocol-Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model. RFC 4208, IETF (October 2005)
34. Yang, X.: Network Aware Resource Broker (NARB) Design and User Manual. Technical Report, University of Southern California (USC), Information Sciences Institute (ISI) (June 2006)