Ultra Scalable UTC-based Pipeline Forwarding Switch for Streaming IP Traffic

D. Agrawal[◆], M. Baldi[•], M. Corrà[■], G. Fontana[◆], G. Marchetto[•], V. T. Nguyen[◆], Y. Ofek[◆], D. Severina[◆], T. H. Truong[◆], O. Zadedyurina[◆]

[◆]Università di Trento
www.unitn.it

[•]Politecnico di Torino
www.polito.it

[■]TRETEC S.r.l.
www.3tec.it

***Abstract*—As traffic on the Internet continues to grow exponentially, there is a real need to solve transmission and switching scalability. Moreover, future Internet traffic will be dominated by streaming media flows, such as video-telephony, video-conferencing, 3D video, virtual reality, and many more. Consequently, network solutions will need to offer quality of service and traffic engineering together with the abovementioned scalability — i.e., over-provisioning is not likely be a viable solution to accommodate streaming media traffic.**

This paper describes a testbed realizing and demonstrating the deployment of pipeline forwarding in order to: (*i*) construct ultra-scalable IP switches and (*ii*) provide quality of service for UDP-based streaming applications. Moreover, the testbed demonstrates the low complexity of pipeline forwarding implementation as the deployed network gear was realized from off-the-shelf components in only nine months through the design, implementation, and testing efforts of the authors.

***Index Terms*—quality of service, UDP-based streaming, testbed and experimentation, scalable IP networking, terabit IP switching, traffic engineering.**

I. Introduction

The Internet has been growing steadily in the past few years. However, services so far deployed over the Internet are nothing compared to the ones that can still be deployed – i.e., we are only at the very beginning. Thus, one may envision that service-wise and traffic-wise growth of the Internet applications is yet to come, one may say that "the best of the Internet is ahead of us".

Implementing UTC-based pipeline forwarding in a real testbed that is scalable to multi-terabit/s switching capacity has been a rewarding experience. The implementation success is a direct outcome of the simplicity of the pipeline forwarding method. The beauty is that the simplicity of this realization did not compromise two most desired performance properties for the future Internet: (1) switching scalability to 10 Tb/s in a single chassis and (2) predictable QoS performance for streaming media and large (content) file transfers.

Note that Cisco's top-of-the-line router, CRS-1, has 640 Gb/s per chassis [the announcement of 92 Tb/s should be divided by 2 (for counting input and output separately) and then by 72 chassis's], which represent a factor of 2 improvement after 5 years of development. So if the Internet traffic is doubling every, say, 18 months there is a real switching bottleneck on the horizon.

This work was supported in part by the European Union under the IP-FLOW contract No MC-EXC 002807 and the E-Next Project FP6-506869.

This paper presents a testbed demonstrating a method known as *pipeline forwarding* that is particularly suitable to carry streaming media applications over the Internet since it offers:

1. High scalability of network switches (multi-terabit/s in a single chassis),
2. Quality of service guarantees (deterministic delay and jitter, no loss) for (UDP-based) constant bit rate (CBR) and variable bit rate (VBR) streaming applications — as needed, while
3. Preserving the support of elastic, TCP-based traffic, i.e., existing applications based on "best-effort" services are not affected in any way.

In the presented testbed, shown in Figure 1, two streaming video flows are generated by a video server (to the left), transported, with *deterministic quality of service*, through a network of one router and two multi-terabit/s switches (all implementing pipeline forwarding) and delivered to two different video clients. IP packets carrying video samples are transported unchanged as a whole end-to-end. Namely, no change can be seen by observing packets flowing on any link of the testbed as only conventional IP packets encapsulated into Ethernet frames travel across the network testbed.

In essence what the testbed demonstrates is a novel technology and network architecture for supporting and engineering (UDP-based) streaming traffic, while elastic (TCP-based) traffic is kept unchanged and unaffected. Proper and efficient support of streaming UDP-based applications is getting increasingly important due to the fact that more and more streaming media traffic over the Internet is using UDP. Such applications need a minimum level of service quality in order to operate properly and current approaches to offer controlled quality based on over-provisioning do not scale. Conversely, the presented solution enables the realization of low complexity, highly scalable IP switches.

II. Underlying Principles and Technologies

Pipeline forwarding is a known optimal method that is widely used in computing and manufacturing. The necessary requirement for pipeline forwarding is having common time reference (CTR). In this design UTC (coordinated universal time) is used for CTR, consequently, the method used in the testbed is called *UTC-based pipeline forwarding*. An extensive and detailed description of UTC-based forwarding is outside the scope of this paper and is available in [1].

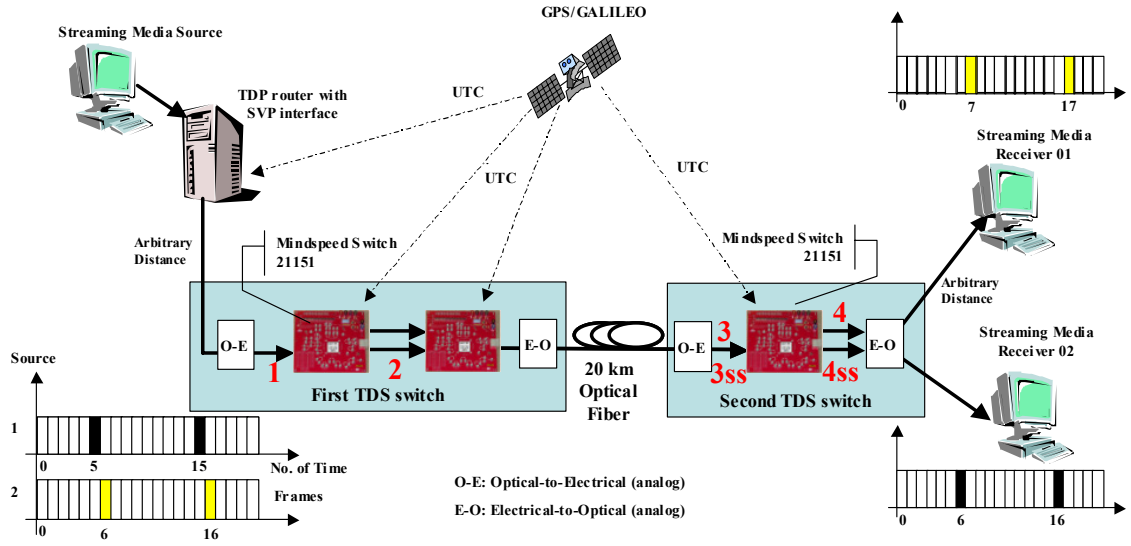

Figure 1. Testbed Functional Diagram

In UTC-based pipeline forwarding all packet switches are synchronized, while utilizing a basic time period called time frame (TF). The TF duration (T_f) may be derived, for example, as a fraction of the UTC second received from a time-distribution system such as the global positioning system (GPS) and, in the near future, Galileo. TFs are grouped into time cycles (TCs) and TCs are further grouped into super cycles, each super cycle lasts for one UTC second. TFs are partially or totally reserved for each flow during a resource reservation phase. The TC provides the periodicity of the reserved flow. This result in a periodic schedule for IP packets to be switched and forwarded, which is repeated every TC.

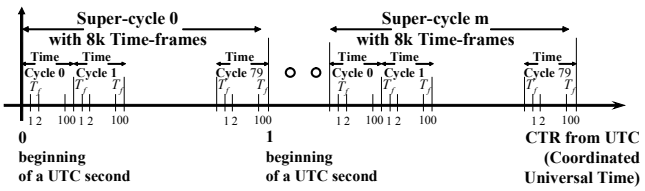

Figure 2. Common time reference structure

For example, in Figure 2, the 125-µs time frame T_f is obtained by dividing the UTC second by 8000; sequences of 100 TFs are grouped into one TC, and runs of 80 TCs are comprised in one super cycle (i.e., one UTC second).

The basic pipeline forwarding operation is regulated by two simple rules: (*i*) all packets that must be sent in TF *t* by a node must be in its output ports' buffers at the end of TF *t*-1, and (*ii*) a packet *p* transmitted in TF *t* by a node *n* must be transmitted in TF $t+d_p$ by node $n+1$, where d_p is an integer constant called *forwarding delay*, and TF *t* and TF $t+d_p$ are also referred to as the *forwarding TF* of packet *p* at node *n* and node $n+1$, respectively. The value of the forwarding delay is determined at resource-reservation time and must be large enough to satisfy (*i*). In pipeline forwarding, a synchronous virtual pipe (SVP) is a predefined schedule for forwarding a pre-allocated amount of bytes during one or more TFs along a path of subsequent UTC-based switches

UTC-based forwarding guarantees that reserved real-time traffic experiences: (*i*) bounded end-to-end delay, (*ii*) delay jitter lower than two TFs, and (*iii*) no congestion and resulting losses. Two implementations of the pipeline forwarding were proposed: Time-Driven Switching (TDS) and Time-Driven Priority (TDP).

Time-driven switching (TDS) was proposed to realize sub-lambda or fractional lambda switching (FλS) in highly scalable dynamic optical networking [2][3], which requires minimum optical buffers. In this context, TDS has the same general objectives as optical burst switching and optical packet switching: realizing all-optical networks with high wavelength utilization. TFs can be viewed as virtual containers for multiple IP packets that are switched at every TDS switch based on and coordinated by the UTC signal.

In TDS all packets in the same TF are switched the same way. Consequently, header processing is not required, which results in low complexity (hence high scalability) and enables optical implementation. The allocation granularity depends on the number of TFs per TC allocated to each flow. For example, with a 10 Gb/s optical channel and 1000 TFs in each TC, the minimum capacity (obtained by allocating one TF in every TC) is 10 Mb/s.

III. TESTBED

The network architecture for streaming media applications presented in the previous sections has been demonstrated by building the testbed for video distribution shown in Figure 1.

The main components of the testbed, whose functional diagram is shown in Figure 1, are:

4. Asynchronous streaming sources (audio, video and text) implemented by a video server;
5. TDP router that represents a TDP domain at the backbone edge; the TDP router acts both as edge router encompassing a SVP interface and as an interface to the TDS backbone;
6. 20 km single-mode optical fiber;
7. Two TDS switches that are constructed with Mindspeed M21151 switch boards, where the switching capacity of each M21151 is 400 Gb/s; an FPGA-based controller

controls each M21151.

8. FPGA-based controller for scheduling the operation of two M21151 (implementing a two stage Banyan interconnection network, which is scalable to 10Tb/s switching capacity); and
9. Streaming media receivers for separately playing the two video streams transmitted from the asynchronous streaming sources.

Time parameters (i.e., TF duration of 100μs and TC size of 100 TFs) are configured at the TDP router and the two TDS switches. Switching configurations can be configured in each switch controller by means of a graphical user interface (GUI). Multi mode fiber is used for the optical links between TDP router and first TDS switch and between second TDS switch and end-systems.

Two asynchronous video flows are generated by the streaming media source, as was shown in Figure 1, and transmitted to the SVP interface within the TDP router. The streaming video packets are then forwarded via an optical link by the TDP router through Gigabit Ethernet (GE) transceivers to the first TDS switch during different predefined TFs. Specifically, TF 5 and TF 15 belong to one SVP while TF 6 and TF 16 belong to another SVP. The optical signal entering the first TDS switch is converted to an electrical one, switched by the first stage and forwarded to the second switching stage through an electrical connection. Then the video streams are transmitted as an optical signal, through a single mode optical fiber link of 20 km, to the second TDS switch that routes each video stream to a different output. Then the separated video streams are forwarded to two receivers through optical links of an arbitrary length. Video flows are therefore multiplexed on the first and second link they traversed, but TDS switching ensures that video packets reach their corresponding destination with deterministic QoS, i.e., during TF 6 and TF 16 and during TF 7 and TF 17 respectively, as showed in Figure 1. Switching of all three switching boards and network interfaces are synchronized with the 1PPS (pulse per second) signal received from three different GPS receivers.

Figure 3 shows a block diagram and a photograph of the implemented TDS switch which has three major parts:

- GPS receiver (for UTC time signal): an EPSILON Board OEM II provides accurate and stable time and frequency signals for synchronization.
- Switch controller: an Opal Kelly XEM3001 module is used for implementing the FPGA (field programmable gate array) based switch controller. The XEM3001 consists of an EEPROM, a USB microcontroller, a PLL, and a 400,000-gate Xilinx Spartan-3 FPGA (XC3S400-4PQ208C) sub module.
- Switching fabric: it is a network of interconnected Mindspeed M21151 switching boards, which are low-power CMOS, high-speed 144 x 144 crosspoint switches (400 Gb/s switching capacity) with integrated Clock Data Recovery (CDRs), input equalization, and built-in system test and broadcasting features. A programmable input equalizer precedes each CDR.

The communication between the controller and the M21151 boards has three important signals: classification address, data path, and strobe signal as shown in Figure 3. The FPGA-based controller is connected to a PC via a USB 2.0 link. Control data, configured from GUI or retrieved from disk, are downloaded by the PC and stored (or updated) in a memory implemented in the FPGA prior to the actual operation of the switch. Multiple finite state machines (FSMs) are implemented in the FPGA. Each FSM coordinates one Mindspeed board. All FSMs are coordinated using UTC time received by the controller as 1PPS and 10 MHz signals.

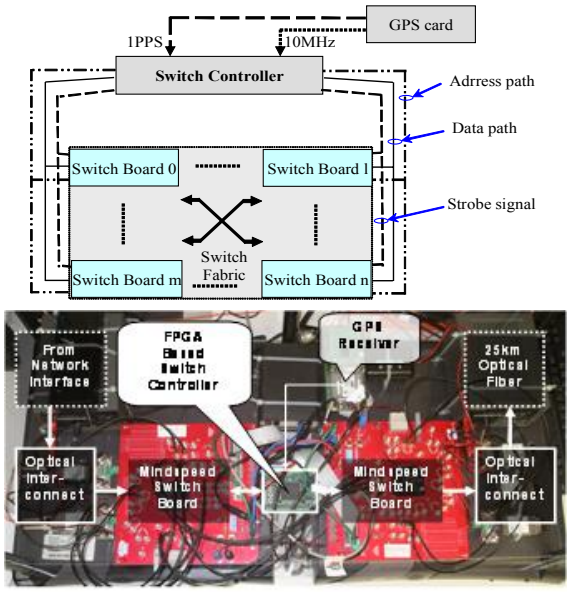

Figure 3. TDS switch: (top) block diagram, (bottom) photograph

IV. DEMONSTRATION

The demonstration of the above-described testbed is done through a combination of the following:

- Posters,
- Slide shows including detailed pictures of the testbed,
- Animations to intuitively illustrate the basic operating principles underlying pipeline forwarding,
- Video clips describing the testbed and its operation,
- A virtual live-demo: a network camera located at the Electronics Laboratory at the University of Trento, Italy, will enable a virtual visit to the testbed. Interaction with a remote presenter in Trento will allow visitors to receive interactive demonstrations directly on the running testbed.

REFERENCES

[1] C-S. Li, Y. Ofek, A. Segall and K. Sohraby, "Pseudo-isochronous cell forwarding," Computer Networks and ISDN Systems, 30:2359-2372, 1998.

[2] M. Baldi and Y. Ofek, "Fractional Lambda Switching - Principles of Operation and Performance Issues", *SIMULATION: Transactions of The Society for Modeling and Simulation International*, Vol. 80, No. 10, Oct. 2004, pp. 527-544.

[3] D. Grieco, A. Pattavina and Y. Ofek, "Fractional Lambda Switching for Flexible Bandwidth Provisioning in WDM Networks: Principles and Performance", *Photonic Network Communications*, Issue: Volume 9, Number 3, Date: May 2005, Pages: 281 – 296.

[4] M. Baldi, G. Marchetto, G. Galante, F. Risso, R. Scopigno,F. Stirano, "Time Driven Priority Router Implementation and First Experiments," *IEEE International Conference on Communications (ICC 2006), Symposium on Communications QoS, Reliability and Performance Modeling*, Istanbul (Turkey), June 2006.