An Ultra-Low Latency Guaranteed-Rate Internet for Cloud Services

T.H. Szymanski, Dept. ECE, McMaster University, Canada, teds@mcmaster.ca

Abstract

A *Deterministic-Internet* network which provides ultra-low latency *'Guaranteed-Rate' (GR)* communications for Cloud Services is proposed. The network supports 2 traffic classes, the *Smooth* and *Best-Effort* classes. *Smooth* traffic flows receive low-jitter GR service over *Virtual-Circuit-Switched (VCS)* connections with negligible buffering and queueing delays, up to 100% link utilizations, deterministic end-to-end QoS guarantees, and improved energy-efficiency. End-to-end delays are effectively reduced to the fiber 'time-of-flight', i.e., the speed of light in fiber. A new router scheduling problem called the *'Bounded-Normalized-Jitter'* integer-programming problem is formulated. A fast polynomial-time approximate solution is presented, allowing TDM-based router schedules to be computed in microseconds. We establish that all admissible traffic demands in any packet-switched network can be simultaneously satisfied with deterministic GR-VCS connections, with minimal buffering. Each router can use 2 periodic TDM-based schedules to support GR-VCS connections, which are updated automatically when the router's traffic rate matrix changes. The design of a *Silicon-Photonics* all-optical packet-switch with minimal buffering is presented. The *Deterministic-Internet* can: (a) reduce router buffer requirements by factors of $\geq$ 1,000, (b) increase the Internet's aggregate capacity, (c) lower the Internet's capital and operating costs, and (d) lower greenhouse gas emissions through improved energy-efficiency. *

Keywords - future internet; deterministic internet; industrial internet of things; tactile internet of things; IoT; IIoT; ultra low latency; buffer size; SDN; control plane; power saving; energy efficiency; cloud; green cloud computing; data centers; low latency, QoS; communication; network; guaranteed rate; DiffServ; routing; scheduling

I. INTRODUCTION

*This paper is a slighted editted version of a paper with the same title published in the IEEE/ACM Trans. on Networking in February 2016. The name *'Enhanced Internet'* has been replaced with *'Deterministic Internet'*. A link to the original IEEE paper is provided in the header of this paper.

The *Best-Effort* (BE) Internet is a universal platform for delivering digital services. However, it relies on significant over-provisioning of bandwidth to achieve relatively poor *Quality of Service* (QoS) guarantees, which results in poor utilization, throughput and energy-efficiency [1-10]. The Internet's poor energy-efficiency leads to excess energy costs annually and contributes noticeably to *Global Warming* and greenhouse gas emissions [1]. The reliance on significant over-provisioning also leads to excessive capital expenditures of several $Billion per year, to deploy a fundamentally inefficient Internet architecture. As a result, governments worldwide are exploring *'Future Internet* architectures, with both *evolutionary* or *revolutionary* changes to the Internet architecture.

The *Cloud* offers a new paradigm for service delivery. *Cloud Service Providers* (CSPs) can deploy new services on a global scale by leasing cloud storage and computing facilities from cloud infrastructure providers. Cloud services include large scale video distribution (i.e., Netflix and YouTube), and *High Performance Computing* (HPC) systems in the cloud. In 2014, Cisco announced a $1 Billion investment to create a global *'Intercloud'*, linking public and private clouds to create an unprecedented global services infrastructure. Unfortunately, the BE-Internet faces challenges as a service-oriented infrastructure, due to its inherently poor utilization and energy-efficiency, and lack of QoS guarantees for time-critical services. In 2011, the *Greentouch* consortium stated a goal to achieve a significant reduction in energy-consumption per bit, attempting to limit the Internet's rapid growth in energy costs (www.greentouch.org).

In this paper, a simple change to the basic *Best-Effort Internet* is proposed, resulting in a packet-switching network that supports *Guaranteed-Rate Virtual Circuit Switched* (GR-VCS) connections called

the *Deterministic-Internet*. It supports the existing bursty BE traffic class, a new *Smooth (Guaranteed-Rate)* class, and optionally one or more *Quasi-Smooth (Guaranteed-Rate)* traffic classes. *Smooth* traffic flows receive congestion-free Guaranteed-Rate service with deterministic QoS guarantees, over TDM-based GR-VCS connections on the packet-switched Internet. The research community and IETF have long recognized the benefits of guaranteed-rate service [12], but there has never been a feasible technology to support such a service (see section II).

The use of GR-VCS lowers router buffer sizes and queueing delays by factors of $\geq$ 1,000. The topic of *'Ultra-Low-Latency'* networking for machine-to-machine communications has received considerable attention lately. A 1 millisec delay can cost a financial firm performing automated stock trading $100 Million/year [10]. Cloud services such as cloud computing are also sensitive to network latency [11]. The BE-Internet has average delays of $\approx$ 100 millisec, representing a significant challenge for time-sensitive traffic. The *Deterministic-Internet* provides *Smooth* traffic flows with low latencies, deterministic QoS guarantees and improved energy-efficiency.

The *Deterministic-Internet* requires a control-plane to establish the GR-VCS connections. Several technologies can be used: (1) The existing RSVP-TE control plane used in MPLS-TE networks; (2) A new IETF DiffServ model [13,14,15] which allows logical connections to be established using software commands; and (3) *Software Defined Networking* (SDN) and *Open-Flow* to provide a user-programmable control plane [16,17]. (None of these prior methods address how to achieve GR-VCS connections in a packet-switched network.) It is shown that when 2 technologies are combined; (i) a control-plane with a resource-reservation signalling protocol, and (ii) a QoS-aware low-jitter scheduling algorithm for the routers, then all admissible traffic demands can be satisfied with TDM-based GR-VCS connections, and achieve (i) reduced router buffer sizes by factors of $\geq$ 1,000, (ii) reduced router queuing delays to a negligible value relative to the fiber delays, (iii) deterministic and essentially-perfect throughput, resource-utilization and QoS guarantees, and (iv) significantly improved energy-efficiency, for all link loads $\leq$ 100%. (The concept of *'Essentially-Perfect' QoS* is defined in section IV.)

A new QoS-aware router scheduling problem with 100% throughput is formulated, called the *Bounded-Normalized-Jitter* integer-programming problem. A fast polynomial time approximate solution is presented, which allows periodic TDM-based schedules for routers to be computed in microseconds. A router can use 2 periodic TDM-based schedules to support the GR-VCS connections, called the *'Queue'* and the *'Flow'* schedules. These schedules can be updated when the router's traffic rate matrix is updated by the control-plane. The *Queue-schedule* defines the conflict-free matchings between the IO ports of a router. This schedule activates a set of conflict-free *'Virtual Output Queues'* (VOQs) in each time-slot of a scheduling frame. In a core router, each VOQ may support thousands of competing traffic flows, and the *Flow-schedule* identifies the flow for service within an activated VOQ.

Three flow scheduling algorithms are defined and analysed in this paper: (i) the *'Static-GPS'* algorithm, (ii) the *'Dynamic-GPS'* algorithm, and (iii) the *'Random'* algorithm. These algorithms can reduce the router buffer sizes to $\approx$ 1/2 packet per smooth traffic flow per router. In comparison, existing BE routers using TCP flow control can buffer up to one million IP packets per link (at 100 Gbps), and they cannot achieve 100% throughput or any deterministic QoS guarantees. Extensive simulations indicate that the buffer sizes for *Smooth* traffic flows can be reduced by factors of $\geq$ 1,000 compared to current BE routers using TCP flow control. These results are important for future all-optical routers, where optical buffering is very limited.

For *Smooth* traffic flows, the majority of buffering can be removed from the core routers and can occur at the cloud data centers. Existing routers typically have programmable token-bucket based *Traffic Shaper Queues (TSQs)*, which are often un-utilized for Best-Effort traffic. In the *Deterministic-Internet*, *TSQs* can be enabled at each cloud data center to aggregate bursty traffic streams and generate a low-jitter stream of IP packets for transmission. A *Traffic Playback Queue (TPQ)* can be used at each destination (i.e., a cloud distribution center), to demultiplex the aggregated stream into the original bursty traffic streams. The TSQ and TPQ queues are external to the core routers in the Internet, and can reside in the access routers. Note that the access routers at cloud data centers already have extensive buffering capacity and

TSQs/TPQs, so no new buffering capacity is needed at the data centers.

The US Department of Energy (DOE) has recently concluded that the cloud computing is not cost effective and is only suitable for applications with minimal communication requirements, due to the excessive delays and poor energy-efficiency of the Best-Effort Internet [11]. The costs of energy inefficiencies in global data-centers and the Internet can be estimated at several Billion $US/year (see section VII-C). By enabling large-scale scientific cloud computing, the *Deterministic-Internet* can recover much of these costs, offering a significant return-on-investment. It can: (i) increase the utilization and energy-efficiency of data-centers, and (ii) increase the utilization and energy-efficiency of the Internet. The *Deterministic-Internet* requires only relatively minor hardware changes to the existing BE-Internet, i.e., the addition of an FPGA per linecard to manage the schedules. (The cost of FPGAs is very small compared to the costs of an inefficient Internet, as shown in section VII.)

This paper extends preliminary results presented in [19,20]. The new results include a stronger theoretical framework; (i) the formulation of a new integer-programming problem called the *'Bounded Normalized Jitter'* scheduling problem; (ii) theorems which establish that all admissible traffic demands in any packet-switched network $G(V, E)$ can be simultaneously satisfied using TDM-based GR-VCS connections with minimal buffer sizes and queueing delays; (iii) a comparison to conventional TCP and MPLS-TE congestion control; and (iv) the design of a single-chip *Silicon-Photonics* all-optical packet switched router.

The paper is organized as follows. Section II reviews prior work. Section III describes the *Deterministic-Internet*. Section IV presents several new integer-programming QoS-aware low-jitter scheduling problems. Section V presents several flow-scheduling algorithms. Section VI presents the end-to-end theorems. Section VII discusses TCP flow control and presents an all-optical packet switch design, and section VIII concludes the paper.

II. Prior Work

A. A Review of ATM and MPLS-TE:

The ATM standard supports (i) *'Permanent Virtual Circuits' (PVCs)* which are established for months/years, and (ii) *'Switched Virtual Circuits' (SVC)* which are established and released dynamically. ATM supports 5 traffic classes, including the *'Constant Bit Rate' (CBR)* class. The *'Cell Delay Variation' (CDV)* represents the end-to-end cell delay distribution. According to the ATM Forum 'Traffic Management Specification V4.1', *"QoS Commitments are probabilistic in nature, and are intended to be only a first order approximation of the performance the network expects to offer"*. The ATM Forum recognizes that the ATM CBR is not a true GR service in its specification.

According to the IETF RFC 2381, the ATM CBR traffic class has a nominal *'Peak Cell Rate' (PCR)* and a nominal *jitter tolerance* called the *'Cell Delay Variation Tolerance' (CDVT)*. According to the Mariam-Webster dictionary, the phrase nominal means *"Existing as something in name only; not actual or real"*. The IETF also recognizes that the ATM CBR is not a true GR service in RFC 2381, by recognizing that the PCR and jitter are nominal quantities.

Cisco further clarifies the problem in its document 10422: *"Ideally, an ATM router schedules cells of a given VC at an even inter-cell gap. This ideal time may be affected by ... cells carrying the physical layer framing, or cells from other VCs configured in the same interface and competing for the same timeslot"*. Clearly, the root of the problem to achieving a true Guaranteed CBR service is a scheduling problem at the routers/switches, i.e., the need to schedule all the cells from all the competing CBR flows through a switch or router, while simultaneously ensuring that the spacing between cells in each CBR flow is ideal. (This ideal scheduling problem is addressed in Section IV-A).

According to Cisco document 10422, the buffers for CBR traffic flows can become heavily loaded, and cells belonging to CBR traffic flows will be dropped: *"The hardware must have reassembly buffers large enough to accommodate the largest CDV present on a VC to prevent underflow and overflow, yet not so large as to induce excessive overall delay It is important to emphasize that this value should optimize the jitter versus absolute delay tradeoff."* According to Cisco, there is a fundamental tradeoff

between jitter and absolute delay and a single VC cannot simultaneously have both an exceptionally-low jitter and an exceptionally-low absolute delay. Furthermore, *"The number of intervening switches, their queue management, and line speeds have a significant impact on the distribution of the CDV that must be handled."* Cisco discusses how the CDV accumulates across multiple nodes in its documentation, and is therefore unbounded.

The ATM standard was developed by many people over many years. Ultimately, ATM became a complex standard that could not provide deterministic QoS guarantees, and it was largely abandoned in favour of a much simpler Best-Effort Internet. Nevertheless, ATM had some good ideas which were absorbed into the newer MPLS-TE standard. Unfortunately, the MPLS-TE standard suffers from the same drawbacks of the ATM standard, i.e., it is complex, it also cannot provide deterministic QoS guarantees, and it also has been largely bypassed in favour of the simpler BE-Internet. Unfortunately, the poor QoS and energy-efficiency of the simple BE-Internet are now taking their toll. New real-time services such as scientific cloud-computing are not well supported. The capital and energy costs due to data-centers and Internet inefficiencies are measured in the tens of $Billions (see Section VII-C), and are growing exponentially [1]. A simple method to achieve improved QoS and energy-efficiency in the BE-Internet may help and is proposed in this paper.

B. A Review of Scheduling:

The terms *Capacity Region, Throughput Region, Stability Region, and Schedulability Region* has been defined to describe the concept of maximum achievable capacity of a network [21,22,23]. A network $G(V, E)$ with N nodes can support $N \times (N-1)$ traffic flows for the different source-destination or (s,d) pairs, subject to constraints on edge capacities. Each vector of $N \times (N-1)$ achievable traffic flow rates for all (s, d) pairs defines a point in $N \times (N-1)$ dimensional space. The set of all achievable points defines a polytope in $N \times (N-1)$-dimensional space, and the convex hull of the polytope defines the *Capacity Region* of the network [21,22].

References [21,22,23] have shown that a *Maximum Weight Matching* (MWM) scheduling algorithm can achieve bounded buffer sizes and stability within one Input-Queued (IQ) router. However, the MWM algorithm has complexity $O(N^3)$, which renders it intractable for realistic networks. The problem of minimizing jitter in one IQ router is shown to be NP-HARD in [24]. A tractable polynomial-time *Greedy Low-Jitter Decomposition* (GLJD) was also proposed in [24]. However, it requires a worst-case speedup of O(logN), rendering it inefficient. The *Birkoff von Neumann* (BVN) algorithm proposed in [25] can schedule traffic through an IQ router with low jitter and with complexity $O(N^{4.5})$, which is considered intractable. A scheduling scheme developed at MIT [26] requires a speedup of ≈ 2 to achieve high throughput and low jitter. All these prior low-jitter algorithms have speedup requirements or very high computation complexities (typically $O(N^{4.5})$ complexity), rendering them intractable. In summary, tractable low-jitter scheduling algorithms which achieve stability within the *Capacity Region*, i.e., bounded buffer sizes and queueing delays under the constraint of unity speedup, are unknown.

In practice, heuristic *Best-Effort* (BE) schedulers are implemented in BE-Internet routers, ATM and MPLS-TE switches. BE-schedulers attempt to find *Maximal Matchings* (MM) in each time-slot in each router/switch. The iSLIP algorithm is one example [28]. However, heuristic BE-schedulers cannot achieve 100% throughput, a weakness in BE-Internet, ATM and MPLS-TE networks. Heuristic router scheduling algorithms such as iSLIP have a peak throughput of about 75% for non-uniform traffic, and they exhibit excessive queueing delays (i.e., several thousand packets) at high loads. In practice, IP routers are often over-provisioned and operate at light loads, typically $\leq 33\%$ [2,3,4]. As a result of bursty traffic and over-provisioning, IP networks often operate at reduced utilizations, often forfeiting up to 66% of the network capacity to over-provisioning. It is often said that *'A chain is as strong as its weakest link'*, and one weak link in network QoS and energy-efficiency is the *Best-Effort* nature of routing and scheduling algorithms in the BE-Internet, ATM and MPLS-TE networks.

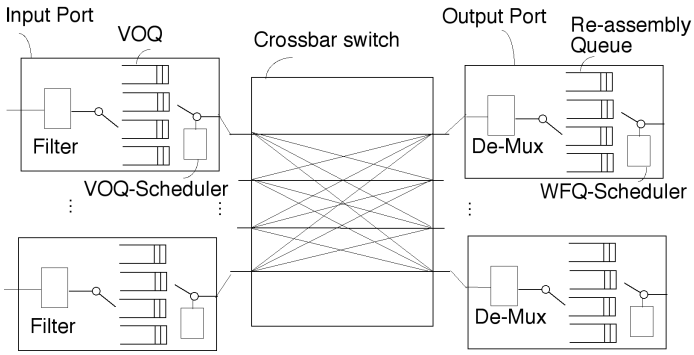


Fig. 1. Best-Effort Internet router design with VOQs.

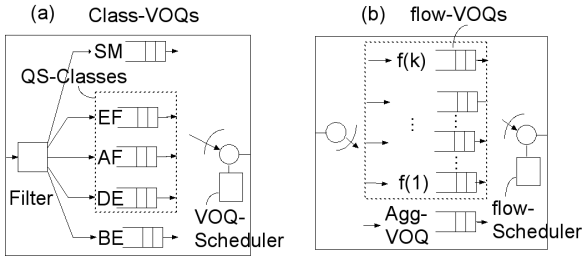


Fig. 2. (a) An Input Port with class-VOQs. (b) A class-VOQ partitioned into flow-VOQs and one Aggregate-VOQ.

III. Router Designs

A *Best-Effort* Input-Queued (IQ) Internet router is shown in Fig. 1. A router of size $M \times M$ consists of M input and output ports $IP(i)$ and $OP(i)$, for $1 \leq i \leq N$. Each input port has M *Virtual Output Queues* (VOQs), where VOQ(i,j) stores packets at input port i destined for output port j. Each input port contains an Input-Filter to filter, police and classify incoming packets, and a demultiplexer to forward packets to the appropriate VOQ. Variable-size packets are typically segmented into fixed sized cells at the input side, and are reconstructed at the output side. The *VOQ-scheduler* selects a VOQ for service in each time-slot. A *Best-Effort* Internet router will use a *Best-Effort* scheduler as the VOQ-scheduler. The *WFQ-scheduler* selects a reconstructed packet for transmission at the output side.

A. Deterministic-Internet Router Design

The proposed *Deterministic Internet* routers supports the existing *Best-Effort* class, and the new *Smooth* traffic class. It can also support one or more optional *Quasi-Smooth* traffic classes.

In Fig. 2a, each VOQ from Fig. 1 is partitioned into several new classes of *VOQs*; the *Smooth* VOQ, the *Best-Effort* (BE) VOQ, and three optional *Quasi-Smooth* VOQs called the (EF, AF, DE) VOQs, The new *Smooth* VOQ contains smooth (low-jitter) traffic flows which can achieve significantly improved energy-efficiency and end-to-end QoS guarantees. The new *Smooth* traffic class will handle Guaranteed-Rate smooth traffic flows which have been established using a resource-reservation signalling protocol, as described ahead. No new buffers are required, as existing *Best-Effort* Internet routers already have very large buffers, typically over 1 million IP packet buffers per Input Port.

The optional *Quasi-Smooth* VOQs are shown in Fig. 2a, to provide backward-compatibility with the existing legacy IETF Differentiated-Services (DiffServ) and the legacy MPLS-TE traffic types. The legacy Diffserv and MPLS-TE traffic types do not have significant constraints on their burstiness. Therefore, if these legacy traffic types are too be supported then they can be assigned to the optional *Quasi-Smooth* VOQs.

Traffic flows belonging to the *Smooth* class can be identified by their packet headers. *Smooth* traffic flows have a very low burstiness value, typically $\pm$ K maximum-size packets for relatively small K.

Assuming K = 4 maximum-sized packets of ≈ 1,500 bytes each, the *Maximum Burst Size* is ≈ 6K bytes. The optional *Quasi-Smooth* traffic flows may have larger burstiness values, typically 1%, 10% or 100% of the average bit rates (see section VII); In contrast, Best-Effort traffic flows including TCP traffic are bursty with unrestricted burstiness.

Default traffic is treated as BE traffic by the input filters in Fig. 2a, and is assigned to the BE VOQ. Several servers denoted by circles are shown in Fig. 2a and 2b. Each server can be controlled by a pre-computed periodic schedule, which can be stored in a *LookUp Table* (LUT), denoted by the boxes associated with each server in Fig. 2. In a *Deterministic-Internet* router, 2 LUTs can be used in Figs. 2a and 2b. The *VOQ-Scheduler* in Fig. 2 can use a *'Queue-schedule*, which will identify a class-VOQ for service between each input and output port in each time-slot.

In Fig. 2b, each class-VOQ can be sub-divided into an optional set of fine-grain *'flow-VOQs'*, and a course-grain *'Aggregate-VOQ'*. Each flow-VOQ handles one flow and receives GR service for that flow, in each router and in each scheduling frame. Each Aggregate-VOQ handles multiple flows and receives GR service in each router and in each scheduling frame, sufficient to meet the aggregate rate required by all the traffic flows in the Aggregate-VOQ. (For the purpose of scheduling, the Aggregate-VOQ can be treated as one flow-VOQ.) The *Flow-Scheduler* in Fig. 2b can use a *'Flow-schedule'*, which will identify at most one *flow-VOQ* for service in each time-slot.

The 2 schedules can be easily computed in software periodically using the scheduling algorithms proposed in this paper. The only new hardware required in a router to support the new traffic classes are the *LookUp Tables* (LUTs). These tables can easily fit on one small FPGA per input port. This change in router design supports the *Smooth* and the optional *Quasi-Smooth* traffic classes, which can co-exist with regular bursty *Best-Effort* traffic class. All bursty TCP/IP Internet applications developed over the last 40 years will continue to work without any changes over the *Deterministic-Internet*. New applications which require low latency with improved QoS and energy-efficiency can exploit the new *Smooth* traffic class.

B. A Flow-Based Control Plane

The IETF has presented a new *"Resource Management in DiffServ"* model to enhance the performance of the traditional DiffServ model, in RFCs 5865, 5974, and 5977 [13,14,15]. The new DiffServ control-plane allows applications to establish DiffServ connections with a nominal bit-rate, specified in a *Traffic Specification* (TSPEC). The DiffServ TSPECs typically specify an *Average Bit Rate* and the *Maximum Burst Size*. The new DiffServ model supports 2 options; (a) a *Fine-Grain* QoS model, where a TSPEC can be specified for individual end-to-end flows in each router along a path, and (b) a *Course-Grain* QoS model, where a TSPEC can be specified for an Aggregate-VOQ in a router. Referring to Fig. 3, let a DiffServ traffic flow f in the Smooth class with a TSPEC be established between routers R(Src) and R(Dst), using the Fine-Grain QoS model.

A suitable end-to-end path P between routers R(Src) and R(Dst) can first be precomputed using any constraint-based routing algorithm. In this paper, assume a *Multicommodity Maximum-Flow Minimum-Energy* routing algorithm is used [30,31]. The new DiffServ control-plane can then signal the relevant routers in the network as shown in Fig. 3, to establish the connection. In the *Deterministic-Internet*, a new flow-VOQ can be created for the flow in each router, and the router schedules can be updated to meet the new traffic demand.

Alternatively, the *Deterministic-Internet* can use a control-plane based upon *Software Defined Networking* (SDN) and *Open Flow* to implement a user-programmable control plane, where users can program the ability to establish flow-based connections in a network of SDN-compatible routers [16].

IV. QoS-Aware Router Scheduling Algorithms

Consider an Input-Queued (IQ) router as shown in Fig. 1. The crossbar switch can be unbuffered or it can use crosspoint buffering, it can be single stage or multi-stage (i.e., a Clos topology). Let the time-axis be divided into *Scheduling Frames*, each consisting of F time-slots, where each time-slot supports

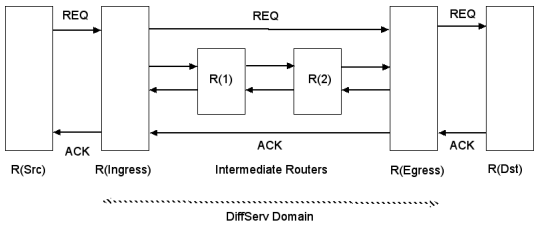


Fig. 3. Signalling within a NSIS Diffserv Domain

the delivery of a fixed-size packet from an input port to an output-port. For example, given a link with capacity 40 Gbps and a scheduling frame length of 2K time-slots, each recurring time-slot reservation in a frame represents a bandwidth of 20 Mbps.

Let $p(i, f)$ denote packet i in flow f (for $0 \leq i \leq F$). Let $R(f)$ denote the integer guaranteed rate for flow f, where $0 \leq R(f) \leq F$, equalling the number of time-slot reservations per scheduling frame needed to support the flow. Let $D(i, f)$ denote the *Actual Departure Time* of packet $p(i, f)$. Let $D^*(i, f)$ denote the *Ideal Departure Time* of packet $p(i, f)$, in a perfectly-scheduled zero-jitter flow. Let $\tau(i, f)$ denote the *Actual Inter-Departure Time* between packets i and $i-1$ of flow f, for $1 < i$. Let $\tau^*(f) = F/R(f)$ denote the *Ideal Inter-Departure Time* (IIDT) between adjacent packets in a perfectly-scheduled flow f with zero jitter. Therefore, $\tau^*(f) = F/R(f)$ = one IIDT.

The *Generalized Processor Sharing* (GPS) theory in [18] presents a theory for computing packet departure times, when multiple traffic flows contend for bandwidth of a single output link, under the assumption that any *excess bandwidth in the output link is shared* amongst backlogged flows. To support smooth traffic flows, we must disable the excess bandwidth sharing property of the GPS theory, to yield a *Smooth GPS* theory. According to this *Smooth GPS* theory, for packets i where $2 \leq i \leq R(f)$, let the ideal departure times on an output link for a perfectly-scheduled zero-jitter flow be given by

$$D^*(i, f) = D^*(i-1, f) + \tau^*(f),\ i > 1 \tag{1}$$

$$D^*(1, f) = rand(1, \tau^*(f)),\ i = 1 \tag{2}$$

where $rand()$ selects an integer between the bounds at random. Given a finite scheduling-frame length F of indivisible time-slots, the smoothest (i.e., lowest-jitter) departure times on an output link, for packets i where $2 \leq i \leq R(f)$, are given by

$$D^*(i, f) = round\Big(D^*(i-1, f) + \tau^*(f)\Big) \tag{3}$$

The *Jitter* of a traffic flow is often measured by electronic test equipment, and is often reported as the deviation of the inter-departure times $\tau(i, f)$ of two adjacent packets in a flow from the mean observed inter-departure time $\bar{\tau}(f)$ for flow f. The jitter of $p(i, f)$ is denoted $J(i, f)$ and can be defined as

$$J(i, f) = (D(i, f) - D(i-1, f)) - \bar{\tau}(f) \tag{4}$$

Unfortunately, a *Bounded Jitter* cannot be used in the theory of *Network Calculus* to provide any QoS guarantees. The arrival of a very long sequence of packets with a bounded jitter allows the traffic flow to fall behind a perfectly-scheduled traffic flow without bounds. To provide mathematically-provable QoS guarantees using the theory of *Network Calculus*, the concept of *Service Lead/Lag* (SLL) must be used.

Definition: Let a link of rate L bytes/sec service a fixed-sized packet with B bytes in a time-slot. The *Normalized Service Lead/Lag* NSLL(i, f) of packet $p(i, f)$ equals the difference in the number of packets serviced between the actual flow f at time $D(i, f)$ and the perfectly-scheduled flow at time $D^*(i, f)$. The time difference between the ideal and actual arrival times of the i-th packet is given by $D^*(i, f) - D(i, f)$,

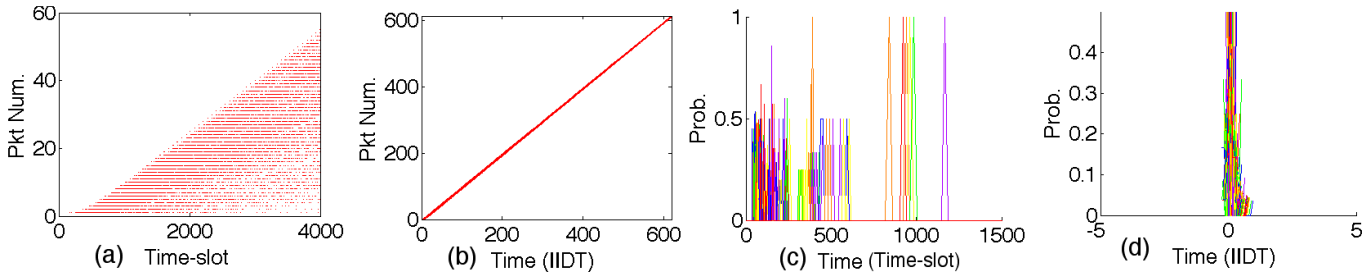


Fig. 4. Performance of the bounded normalized jitter scheduling problem.

which is multiplied by the GR to determine the lag:

$$NSLL(i,f) = (D(i,f) - D^*(i,f))(\frac{R(f)}{F}) \cdot L \tag{5}$$

Observe that the number of packets serviced at ideal time $D^*(i,f)$ cannot be measured by test equipment, since it does not necessarily correspond to the arrival or departure of any real packets. A positive NSLL represents how many packets behind service the flow has fallen, relative to a perfectly-scheduled flow. A negative NSLL represents how many packets ahead of service the flow has moved, relative to a perfectly-scheduled flow. The concept of the NSLL is critical to establishing the theorems using *Network Calculus* to follow shortly. The next 3 optimization problems use the term *Jitter* to mean a *NSLL*, since the term jitter is widely used in the community.

A. The Perfect-Minimum-Jitter Scheduling Problem

Define a *Perfect-Minimum-Jitter* schedule for a flow f as the case where Eq. (1) is satisfied for $1 < i \leq R(f)$. Observe that a sequence of perfectly-scheduled (zero-jitter) packet departure times for a flow on an output link can be circularly rotated within a scheduling frame, and remain perfectly-scheduled.

The *Perfect-Minimum-Jitter* QoS-scheduling problem can be stated as follows:

$$\text{Minimize: } J^* \tag{6}$$

Subject to:

$$\begin{array}{lll}
D(1,f) \geq 1 & \forall f \in F^k, k \in IP & (6.1)\\
D(1,f) \leq r(f) & \forall f \in F^k, k \in IP & (6.2)\\
D(i,f) \geq D^*(i,f) - 1 & \forall f \in F, k \in IP & (6.3)\\
D(i,f) \leq D^*(i,f,k) + 1 & \forall f \in F, k \in IP & (6.4)\\
\sum_{i=1}^{N} VOQ^t(i,j) \leq 1 & 1 \leq j \leq F & (6.5)\\
\sum_{j=1}^{N} VOQ^t(i,j) \leq 1 & 1 \leq i \leq F & (6.6)
\end{array}$$

$$J^* = \sum_{f \in F} \sum_{i=1}^{R(f)} (|D(i,f) - D(i-1,f)|)$$

---

where $D^*(i,f) = mod(round(D(i-1,f) + \tau^*(f)), F)$, where $mod()$ denotes modulo arithmetic. Let $VOQ^t(i,j) \in \{0,1\}$ be an indicator variable which denotes whether a VOQ(i,j) receives service in time-slot t of the scheduling frame. The problem requires that $VOQ^t(i,j) = 1$ for $t = D(i,f)$, where j is the output port required by flow f. Constraints 6.4 and 6.5 ensure that the active VOQs in each time-slot are conflict-free.

B. The Bounded-Low-Jitter Scheduling Problem

Researchers at Bell-Labs have shown that achieving a perfect minimum-jitter schedule in a single IQ-switch is NP-Hard [24]. Consider relaxing the previous scheduling problem, so that a finite bounded service lead/lag of K time-slots is acceptable. The *Bounded-Low-Jitter* QoS-scheduling problem can be stated using Eq (6), where constraints 6.3 and 6.4 are replaced by constraints 7.3 and 7.4 shown below, to allow a packet depart within K time-slots of its ideal departure time:

$$\begin{array}{lll} D(i,f) & \geq D^*(i,f) - K & \forall f \in F, k \in IP \quad (7.3) \\ D(i,f) & \leq D^*(i,f,k) + K & \forall f \in F, k \in IP \quad (7.4) \end{array}$$

Unfortunately, this problem is also restrictive and difficult to solve. For example, allowing a bounded jitter of $K = \{2, 4, 8\}$ time-slots does not significantly ease any constraints.

C. The Bounded-Normalized-Jitter Scheduling Problem

Consider relaxing the previous scheduling problem, so that a finite bounded service lead/lag of $K\tau^*(f)$ time-slots is acceptable. Constraints 7.3 and 7.4 are relaxed, to allow a packet depart within $K\tau^*(f)$ time-slots of its ideal departure time. The *Bounded-Normalized-Jitter* QoS-scheduling problem can be stated :

$$\text{Minimize: } J^* \quad (8)$$

Subject to:

$$\begin{array}{lll} D(1,f) & \geq 1 & \forall f \in F, k \in IP \quad (8.1) \\ D(1,f) & \leq r(f) & \forall f \in F, k \in IP \quad (8.2) \\ D(i,f) & \geq D^*(i,f) - K^* & \forall f \in F, k \in IP \quad (8.3) \\ D(i,f) & \leq D^*(i,f,k) + K^* & \forall f \in F, k \in IP \quad (8.4) \\ \sum_{i=1}^{N} VOQ^t(i,j) & \leq 1 & 1 \leq j \leq F \quad (8.5) \\ \sum_{j=1}^{N} VOQ^t(i,j) & \leq 1 & 1 \leq i \leq F \quad (8.6) \end{array}$$

$$J^* = \sum_{f \in F} \sum_{i=1}^{r(f)} (|D(i,f,k) - D(i-1,f,k)|)$$

where $K^* = K\tau^*(f)$. Consider a flow f with a low guaranteed rate of $R(f) = 3$ time-slot reservations per scheduling frame with F = 2,048 time-slots. The ideal inter-departure time $\tau^*(f)$ = F/3 = 682 time-slots. Constraints 8.3 and 8.4 allow the i-th packet to depart in any time-slot within the range $D^*(i,f) \pm K \cdot 682$, and still have a bounded NSLL. For large F, this problem significantly relaxes the constraints and significantly expands the feasible solution space. The scheduling algorithm for IQ routers in [29] solves this latter problem, in a fast recursive manner.

Definition: A *Smooth* traffic flow f with a rate requirement of $R(f)$ time-slot reservations per scheduling frame is said to receive *Essentially-Perfect* service with a NSLL = K packets when the scheduling algorithm in each router r satisfies the following condition:

$$(\alpha R(f) - K) \leq \sum_{t=1}^{\lceil \alpha F \rceil} S(t,f) \leq (\alpha R(f) + K) \quad (9)$$

where $S(t,f) \in \{0,1\}$ denotes whether flow f receives service in time-slot t of the scheduling frame, for any fraction $0 \leq \alpha \leq 1$ and small constant K. Eq. (9) requires that at any fraction of time $0 \leq \alpha \leq 1$ in a scheduling frame with F time-slots, the service flow f receives is equal to $\alpha R(f)$ packets plus or minus K packets, i.e., the flow has a bounded NSLL. (Given an network with a small finite packet error rate ϵ, the provisioned bandwidth for a flow should have sufficient bandwidth to account for potential packet retransmissions.)

D. A Polynomial-Time Approximation Algorithm

The *Low-Jitter Guaranteed-Rate* scheduling algorithm in [29] will accept a doubly-substochastic or stochastic traffic rate matrix T and recursively decompose it in a fair manner. Let $T(i,j)$ denote the number of time-slot reservations required between an input port i and an output port j of an IQ router in a scheduling frame with F time-slots. Let $P(T,F)$ denote the problem of scheduling an admissible integer traffic rate matrix T for a router into a scheduling frame of length F time-slots. The problem $P(T,F)$ is recursively decomposed into 2 smaller problems $P(T_1, F/2)$ and $P(T_2, F/2)$, as follows:

$$\text{Minimize: } J^* \tag{10}$$

Subject to:

$$\begin{array}{lll} T_1(i,j) & \geq \lfloor T(i,j)/2 \rfloor & 0 \leq i,j \leq N \quad (10.1) \\ T_2(i,j) & \geq \lfloor T(i,j)/2 \rfloor & 0 \leq i,j \leq N \quad (10.2) \\ T(i,j) & = T_1(i,j) + T_2(i,j) & 0 \leq i,j \leq N \quad (10.3) \end{array}$$

$$J^* = \sum_{i=1}^{N} \sum_{j=1}^{N} (|T_1(i,j) - T_2(i,j)|)$$

The above problem is an NP-Hard integer-programming problem. The polynomial-time algorithm in [29] transforms the matrix partitioning problem in Eq. (10) into another polynomial-time approximation problem of routing permutations in a rearrangeably nonblocking switching network. The algorithm in [29] applies Eq. (10) recursively $O(logF)$ times, and results in F permutation matrices of size $N \times N$, which form a *Queue-schedule*. The following theorem from [29] is stated for completeness.

Theorem on Recursive Fair Scheduling [29]: Given an admissible $N \times N$ traffic demand matrix T, where element $T(i,j)$ denotes the guaranteed rate demanded between input port i and output port j, and a scheduling frame with F time slots, any recursive scheduling algorithm which recursively partitions the scheduling problem into 2 smaller scheduling problems fairly in the time-domain as shown in Eq. (10), such that the amount of traffic allocated to each smaller scheduling problem differs by at most 1 or 2 fixed-size packets, will achieve a bounded NSLL for the traffic flowing between input port i and output port j.

Given an $N \times N$ switch and a fixed scheduling frame of length F, the application of Eq. (10) recursively will bound the NSLL for the traffic flowing between any pair of input-output ports in the router to $\leq K$ packets for constant $K = O(logF)$. Therefore, Eq. (10) can be applied recursively to compute the *Queue-schedule* defined next.

Definition: A *Queue-schedule* for one router is a sequence of F partial or full permutation matrices (or bipartite graph matchings) which define the crossbar switch configurations for F time-slots within a scheduling frame. Equivalently, the Queue-schedule defines the active VOQs for each time-slot in the scheduling frame, i.e., $VOQ \equiv \{VOQ_t\}, 1 \leq t \leq F$, where $VOQ_t(j,k) = 1$ if VOQ(j,k) has a scheduled service opportunity in time-slot t. Each permutation matrix $VOQ_t()$ identifies several conflict-free VOQs for service in time-slot t. Given a line-rate L, the frame length F determines the minimum quota of reservable bandwidth = L/F. For example, given a line-rate of L = 100 Gbps, to allocate link bandwidth in increments of $\leq 0.1\%$ of L = 100 Mbps, set $F \geq 1000$, i.e., $F = 1024$. To allocate link bandwidth in finer increments, the parameter F can be increased.

Definition: A *Flow-schedule* for one router is a sequence of F matrices Z_t which identify the *Smooth* flow to be serviced in each VOQ for the F time-slots within a scheduling frame, given a *Queue*-schedule which identifies the VOQs to be serviced in each time-slot. Equivalently, $\mathbf{Z} \equiv \{Z_t\}, 1 \leq t \leq F$, where $Z_t(j,k) = f$ if *Smooth* flow f within VOQ(j,k) has a scheduled service opportunity in time-slot t.

E. Experimental Results - Queue-Schedule

A linear chain of 10 routers of size 4x4 operating with 40 Gbps links was configured with scheduling frame size F=2048. Each time-slot reservation per scheduling frame reserves 20 Mbps. Several hundred

Smooth flows (514 flows) were routed from end-to-end using a backtracking routing algorithm, such that every router and every link are 100% saturated. Each flow reserved a virtual circuit-switched connection with a guaranteed-rate between 20...800 Mbps. (The routers used the 'Static-GPS' flow-schedule discussed ahead to ensure a bounded NSLL for each flow.) This network configuration represents an extremal point in the *Capacity Region* of the network; the network is operated at maximum load, where every edge and every router are fully saturated.

Fig. 4 illustrates the results of the recursive application of Eq. (10) to compute a *Queue-schedule*. Packets are inserted into the network at their ideal transmission time (Eq. 3), with a NSLL $\pm$ K=1 packet (using WFQ). Fig. 4a illustrates the exit time of the packets, expressed in real time (time-slots). Packets can never depart before they are inserted, so departures are illustrated with dots to the right side of the diagonal. In general, the packets depart at highly variable times ranging from 1...4000 time-slots. In particular, the first packet of every flow departs at a highly variable time ranging from 1...4000 time-slots. Consider a flow f with a guaranteed rate R(f) = 1 reservation per scheduling frame. Suppose the first service opportunity occurs at time-slot 1999, and the packet arrives at time-slot 2000. This packet will be serviced in the 2nd scheduling frame, and have an exit time $\approx$ 4000 time-slots. Fig. 4b illustrates the exit times of packets leaving the network, expressed in normalized time (IIDTs). Observe that all the packets of every flow depart at essentially perfect normalized departure times, i.e., the j-th packet typically departs after j IIDTs $\pm$ 1 IIDT. Fig. 4b illustrates that packets experience *essentially-perfect* service with very little queueing within the routers. Fig. 4c illustrates the jitter, i.e., the measured time between successive packet departures, expressed in real time (time-slots). Fig. 4d illustrates the normalized jitter, expressed in normalized time (IIDTs). The time between successive packet departures in a flow is typically $\pm$ 2 IIDT, indicating *essentially-perfect* service for all flows simultaneously.

V. FLOW SCHEDULING ALGORITHMS

Referring to Fig. 1 or 2, thousands or millions of competing traffic flows can share one class-VOQ in a backbone router. In Fig. 2b, in each time-slot a *'VOQ-Scheduler'* selects a class-VOQ for service. and a *Flow-Scheduler* selects a flow-VOQ within the class-VOQ for service. Whenever a class-VOQ is selected for service in a time-slot, there are thousands or millions of flows which are candidates for service. In an IQ switch, or a switch using a combination of Input Queueing and Crosspoint Queueing, the *Queue*-schedule can be computed using Eq. (10). However, a flow-scheduling algorithm with a bounded NSLL is needed, to provide fair service to the numerous flows sharing one class-VOQ.

The following algorithm called the *'Static-GPS'* algorithm will effectively partition a problem of scheduling a traffic flow into 2 smaller problems fairly, and will achieve a bounded NSLL for the flow. To support the *Fine-Grain* QoS model, within each VOQ a flow f may have its own virtual queue called the *flow-VOQ*. In practice, the packets belonging to a flow can exist in one shared VOQ and can be tracked using pointers. However, it is convenient to view each flow as having its own virtual flow-VOQ.

A. Static-GPS Flow Selection

Given a *Queue-schedule*, the Smooth-GPS theory presented earlier can be used to schedule flows for service within a VOQ. (Recall that the Smooth-GPS theory disables the sharing of excess bandwidth, to reduce jitter.) Initially, all flows associated with a VOQ with $R(f) > 0$ are assigned initial VFTs, as in Eq. (2). Given each service opportunity for the VOQ, the flow with the next smallest VFT is assigned to that service opportunity, and the VFT of the flow is updated as in Eq. (12). Let $p(k, f)$ denote packet k of flow f, and let $VFT(k, f)$ denote the *Virtual Finishing Time* of $p(k, f)$. The following equation can be used to compute the VFTs for all packets in all flows sharing a VOQ, which can be used to schedule the flow-server in Fig. 2b:

$$VFT(1, f) = rand(1, r(f)) \tag{11}$$

and for $k \leq R(f)$

$$VFT(k, f) = VFT(k-1, f) + \tau^*(f) \tag{12}$$

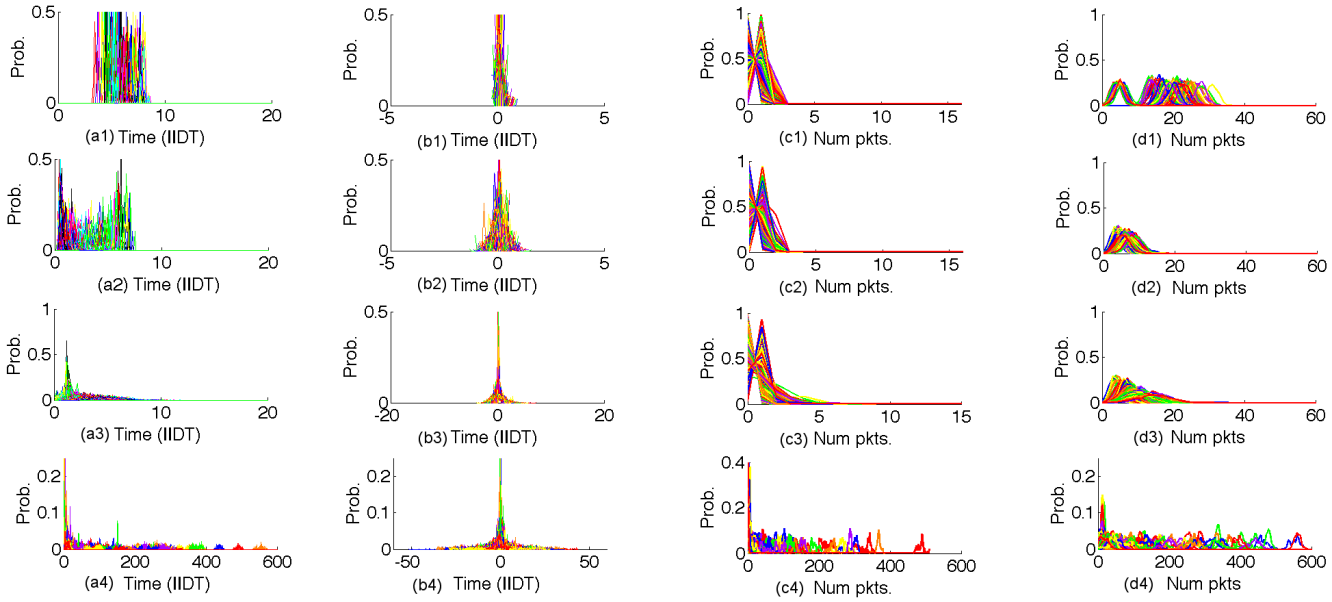


Fig. 5. Performance of Flow-Scheduling algorithms; Static-GPS, Dynamic-GPS, Random, Random (by rows).

For $k > R(f)$, $VFT(k, f) = \infty$. The VOQ-server in Fig. 2b can be scheduled using a *Queue-schedule*, which guarantees that each VOQ receives its requested service with a bounded NSLL. The above equations can then be used to compute a *Flow-schedule*, which is periodic. The flow schedule can be computed once, and reused for subsequent scheduling frames while the flow rates remain static.

B. Dynamic-GPS Flow Selection

The *Static-GPS* algorithm is not work-conserving. Consider the case when the VOQ has several non-empty flows, but the next flow selected for service is empty. The VOQ-server will remain idle even when the VOQ is non-empty, violating the definition of a work-conserving queuing system. To potentially improve the queuing performance, consider a *Dynamic-GPS* flow scheduling algorithm.

Let the dynamic arrival time of a packet at a VOQ determine its VFT and its departure order from the VOQ. When $p(k, f)$ arrives at a non-empty or empty VOQ, the following 2 equations determine the VFT.

$$VFT(k, f) = VFT(k-1, f) + B(k, f)/W(f) \tag{13}$$

$$VFT(k, f) = cVT + B(k, f)/W(f) \tag{14}$$

$B(k, f)$ is the number of bits in $p(k, f)$, and $W(f)$ is the weight of the flow (expressed as a number of bits served per GPS service cycle).

One property of the *Dynamic-GPS* algorithm is that any excess bandwidth on a link is shared equally amongst all non-empty flows. Therefore, a flow may receive more than its fair share of service over an interval of time. Therefore, the NSLL of the flow is not necessarily bounded by K packets. Once one flow loses its property NSLL $\leq K$, it may cause other flows to lose the same property, and the entire network may deteriorate to the case where all flows have lost the property that NSLL $\leq K$. It should be noted that provided each traffic flow is shaped at the source to have a NSLL $\leq K$, then the NSLL at any router using *Dynamic-GPS* is still bounded, but the bound is larger than K, i.e., it may be 10K or 100K. To avoid this potential deterioration, all work-conserving flow-scheduling policies should enable the traffic shapers or policers in each router, to ensure that all departing smooth flows have a bounded NSLL.

C. Random Flow Selection

Consider a random flow selection policy. When a VOQ receives service, any non-empty flow is selected at random to receive service. This server is clearly work-conserving. The *Random* flow-selection algorithm can lower the average number of queued cells per flow per router compared to the *Static-GPS* algorithm, since it is work-conserving. However, it also will increase the worst-case queue sizes, and it also cannot guarantee that every flow has a NSLL $\leq K$, just as the *Dynamic-GPS* scheduler.

D. Experimental Results - Flow-Schedules

The same experimental setup as in section IV was employed. When a VOQ is activated, 3 different flow-scheduling algorithms were used to select the flow for service from the VOQ. As before, 514 traffic flows enter the first router with an average of 128.5 flows per input port, and an average of $\approx$ 32 flows per VOQ. This network configuration represents an extremal point in the *Capacity Region* of the network.

Fig. 5 row 1 presents steady-state results for the *Static-GPS* flow-scheduler. Fig. 5 row 2 presents steady-state results for the *Dynamic-GPS* flow-scheduler. Fig. 5 rows 3 and 4 present steady-state results for the *Random* flow-scheduler. For the first 3 cases, at each traffic source the NSLL(f) = $\pm$ 1 packet, i.e., the flows are smooth when injected into the first router. For Fig. 5 row 4, at the traffic sources the NSLL was restricted to $\pm$ 200% of the reserved bandwidth, i.e., the flows are quite bursty when injected into the first router (see section VII for a discussion of burstiness.)

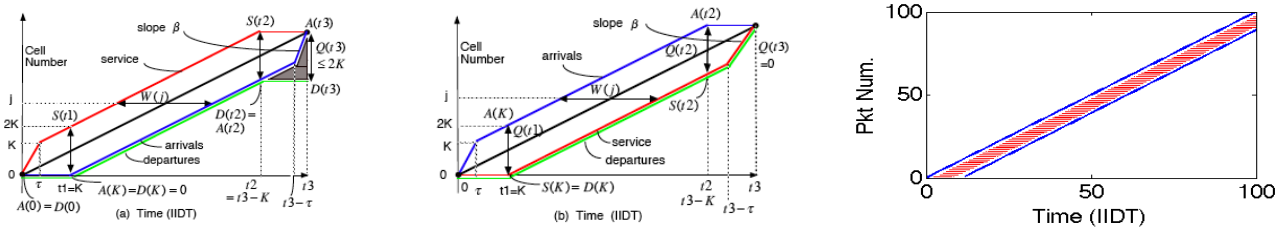
Fig. 5 column (a) illustrates the end-to-end delay for all 514 flows, expressed in normalized time. Fig. 5 column (b) illustrates the end-to-end age deviation from the mean, expressed in normalized time. Fig. 5 column (c) illustrates the flow-VOQ size distribution. There are 5,140 individual plots superimposed, representing 514 flow-VOQs in each of 10 routers. Fig. 7 column (d) illustrates the VOQ size distribution. There are 160 individual plots, representing 16 VOQs in each of 10 routers.

Referring to Fig. 5(d1), for the *Static-GPS* algorithm the mean size of each VOQ is $\approx$ 20 packets, when the NSLL at each source is constrained to be within $\pm$ 1 packet. Each VOQ contains $\approx$ 32 flow-VOQs on average, and each flow-VOQ buffers less than 1 packet per router on average. From Fig. 5(d2), for the *Dynamic-GPS* algorithm the mean size of each VOQ is $\approx$ 10 packets, when the NSLL at each source is constrained to be within $\pm$ 1 packet. From Fig. 5(d3), for the *Random* flow-selection algorithm the mean size of each VOQ is $\approx$ 12 packets, when the NSLL at every source $\leq \pm$ 1 packet. When all flows are *Smooth*, the random flow-selection algorithm offers excellent performance. The Random algorithm can be used to service traffic flows in an aggregated class-VOQ, without maintaining per-flow state information, to support the *Course-Grain* QoS model in RFC 5974.

Fig. 5(d4) represents the *Random* flow-selection algorithm, when the NSLL at each source is $\leq \pm$200% of the GR for each flow (i.e., a traffic flow with a GR of 200 Mbps can have a burst of 400 Mbits outstanding). The *Random* flow-selection algorithm does not guarantee smooth 'fine-grain' service to each flow individually. Even with a near-perfect *Queue-schedule*, the *Random* flow-scheduling algorithm results in large router buffer sizes and queueing delays, when the arriving traffic is bursty. Traffic flows with such large burstiness can be handled in the *Quasi-Smooth* traffic classes.

VI. Theorems for End-to-End QoS Guarantees

Definition: Given a flow f which traverses a queue Q, the *Cumulative Service* (cS) is a sequence of F vectors $\mathbf{c}S \equiv \{cS_t(f)\}, 1 \leq t \leq F$, where $S_t(f)$ equals the number of service opportunities for flow f in the Q in the interval of time $[1, t]$. The *Cumulative Arrivals* (cA) is a sequence of vectors $\mathbf{c}A \equiv \{cA_t(f)\}, 1 \leq t \leq F$, where $cA_t(f)$ equals the number of packets arriving for flow f in the Q in the time interval $[0, t]$. The *Cumulative Departures* (cD) is a sequence of vectors $\mathbf{c}D \equiv \{cD_t(f)\}, 1 \leq k \leq F$, where $D_t(f)$ equals the number of packets which depart for flow f in the Q in the time interval $[0, t]$. The Q backlog is a sequence of F vectors $\mathbf{Q} \equiv \{Q_t(f)\}$ where $Q_t(f) = \lfloor cA_t(f) - cD_t(f) \rfloor$ for $1 \leq t \leq F$, where $Q_t(f)$ equals the positive part of the $cA(f) - cD(f)$ for the Q at time t.



(a) (b) (c)

Fig. 6. (a) Proof case 1 (service leads arrivals). (b) Proof case 2 (arrivals lead service). (c) Transient Performance (before steady-state reached).

The following additional notations will be used. Given a discrete-time random process $x(t)$ defined over an interval [0,t], the minimum and maximum envelopes of $x(t)$ are defined as $\lfloor x(0,t) \rfloor$ and $\lceil x(0,t) \rceil$ The *Cumulative Arrival* curve of a traffic flow f is said to conform to $T(\lambda, \beta, \delta)$, denoted $Af \sim T(\lambda, \beta, \delta)$, if the average packet arrival rate is λ packets/sec, the burst arrival rate is $\leq \beta$ packets/sec, and the maximum NSLL is δ packets. A similar notation is used for the *Cumulative Departure* and *Cumulative Service* curves. In any router, the *Cumulative Departure* curve for f is said to *'track'* the *Cumulative Service* curve for f when packet departures are constrained by the scheduled service opportunities. This scenario occurs when the flow f is backlogged at VOQ(j,k), or if the packet arrivals for flow f occur at the same time-slots as the service opportunities for flow f.

Several theorems are now presented. Assume each traffic flow from a TSQ is injected into a network using a WFQ scheduler, and has a maximum NSLL of $\pm$ K packets. Let the traffic rate matrix for each router be updated using the flow-reservation signalling protocol. Each router uses a *Queue* transmission-schedule and the Static-GPS flow scheduling algorithm, where every smooth flow f has a maximum NSLL of K packets. Assume all IP packets have a fixed maximum size initially.

Theorem 1: Given a smooth flow f traversing VOQ(j,k) over an interval $t \in [0, \tau]$, with arrivals $Af \sim T(R(f), \beta, K)$, with service $Sf \sim T(R(f), \beta, K)$, and $Q(0) \leq 2K$, then $Q(t) \leq 4K = O(K)$.

The formal proof of theorem 1 is given in [19]. An outline of the proof is summarized here. The service diagrams used in the proof are shown in Fig. 6. The y-axis represents a number of packets, and the x-axis represents a time-slot in the scheduling frame. The upper curve in Fig. 6a represents a worst-case service schedule, which leads by at most K packets. The lower curve represents a worst-case arrival schedule, which lags by at most K packets. The vertical difference between these two curves represents the number of queued packets versus time. In [19], the worst-case number of queued packets is shown to be 4K packets.

Define a *steady-state* of a queue at time-slot t in a frame as the state (i.e., occupancy $Q(t)$) which remains constant given the same time-slot in successive frames, i.e., $Q(t) = Q(t+F)$ for $1 \leq t \leq F$.

Theorem 2: In the steady-state, the maximum end-to-end queueing delay of a guaranteed-rate smooth flow f with rate $R(f)$ traversing H routers is $4HK\tau^*(f)$ time-slots (where $\tau^*(f)$ is the IIDT for the flow).

Proof: The proof follows from Little's Law. ■

Theorem 3: In the steady-state, the departures of smooth traffic flow f at any router along an end-to-end path of H routers will exhibit a maximum NSLL of K, i.e., $Sf \sim T(R(f), \beta, K)$. Equivalently, the NSLL of a flow is not cumulative when traversing multiple routers.

Proof: Given a scheduling frame of length F, a scheduling algorithm with 100% throughput will guarantee the conservation of flow for a backlogged flow after F time-slots, i.e., $cA_f(1,F)$ = $cD_f(1,F)$ = $R(f)$. The scheduling algorithm in Eq. (10) and [29] achieves 100% utilization, where all demands are satisfied by the end of the scheduling frame. Theorem 1 ($Q_f(t) \leq 4K$ packets) and the previous conservation of flow condition guarantee that any arriving packets not serviced in one scheduling frame will be queued, and must be serviced in the next frame(s). Therefore, the departure curve will track the service curve and will inherit the parameters of the service curve; $S_f \approx T(R(f), \beta, K)$, and $D_f \approx T(R(f), \beta, K)$.

■

Theorem 3 ensures that the NSLL of a flow remains bounded after traversing any number of routers using the proposed scheduling algorithm, i.e., the jitter or NSLL is not cumulative. (Recall that the ATM and MPLS-TE networks cannot provide the same guarantee.)

Theorem 4: Given a message of size M bytes, a fixed packet size of P bytes, a line-rate of L=100 Gbps, a guaranteed-rate connection with rate R Gbps, the end-to-end delay for the message is upper bounded by

$$(\lceil M/P \rceil + 2)\tau_P^* + (4HK)\tau_P^* \tag{15}$$

where τ_P^* = $8P/R$ represents the IIDT for the packets at the Guaranteed-Rate.

Proof: The message is segmented into $\lceil M/P \rceil$ packets in a traffic shaper queue (TSQ), which are transmitted at the source with a bounded NSLL ± 1. By theorem 3, the maximum waiting time per router is $4K$ τ_P^* time. The worst-case end-to-end delay occurs when the first packet in the message incurs the maximum possible delay in each of the H routers, and to avoid a contradiction every packet following the first must arrive and be serviced within 1 τ_P^* thereafter. The packet is fully reassembled in the traffic playback queue at the destination after the last packet arrives. ■

Theorem 4 ensures that any bursty traffic flow can be shaped in the TSQ at a source node, and transmitted as sequential messages with mathematical end-to-end performance guarantees.

Example: Given L = 100 Gbps, R = 1 Gbps, M = 15,000 bytes, P = 1,500 bytes, and K=4, the packet IIDT τ_p^* = 12 μ sec, and the worst-case end-to-end queueing delay in the routers over a connection with 5 routers $\leq$ 1.1 millisec.

Corollary 1: Given (i) any directed network $G(V, E)$ of packet-switched IQ routers with unity speedup (where $|V| = N$), and (ii) any admissible traffic demand matrix $R = R(s, d)$ representing $N \times (N - 1)$ GR-VCS connections each denoted by f, each with source and destination nodes $s(f)$ and $d(f)$, and an GR demand of $R(f)$ time-slot reservations per scheduling frame, then 2 TDM-based periodic schedules can be computed for every IQ router $v \in V$; the *Queue-schedule*, and the *Flow-schedule*. The 2 schedules guarantee that every GR-VCS connection f specified in matrix R will receive its end-to-end guaranteed-rate $R(f)$ with a bounded NSLL of $\pm$ K packets at any router and at any point in time. Every admissible GR-VCS connection f can receive deterministic and essentially-perfect service simultaneously, with minimal router buffer sizes and minimal end-to-end queueing delays.

The corollary assumes fixed-size packets. Current IP packets have variable sizes, typically ranging from 64...1500 bytes. Variable-size IP packets are typically segmented into fixed-size cells at the input ports of a router, switched through the router, and re-assembed into variable size IP packets at the output ports. Once re-assembled, the GPS/WFQ algorithm can be used to schedule the transmission of re-assembed IP packets over the outgoing fibers. In the *Enhanced-Internet* using IP-v6, larger maximum-size packets can be used to improve transmission efficiency. The next theorem bounds the end-to-end delay, given variable-size IP packets with packet segmentation and re-assembly in each router.

Theorem 5: Given the network of Corollary 1, where: (i) incoming IP packets have a maximum size of B bytes, (i) the routers fragment incoming IP packets into fixed-size cells (i.e., 64 bytes) at the input side for transmission through the switch, and (ii) the routers reassemble maximum-size IP packets at the output side for transmission through the outgoing edge, then for every GR-VCS connection f the maximum end-to-end queueing delay of a packet over H routers is $4H(\tau_B^*(f) + K\tau_S^*(f))$ time-slots, where $\tau_B^*(f)$ denotes the IIDT of the maximum-size IP packets in a perfectly-scheduled *Smooth* flow with rate $R(f)$, and where $\tau_S^*(f)$ denotes the IIDT of the small cells in a perfectly-scheduled *Smooth* flow with rate $R(f)$.

Proof: To illustrate the worst-case scenario, let the maximum IP packet size be 6400 bytes, let each router segment IP packets into 100 cells with 64 bytes each, and let the line-rate L = 100 Gbps. Consider a traffic flow f with GR R(f) = L, where no excess-bandwidth is provisioned so that there are no idle periods. Each IP packet is segmented in 100 cells at the input-side of each router, and re-assembled at the output side of the router. Observe that $\tau_B^*(f) = 100\tau_S^*(f)$. Fig. 7 illustrates the worst-case queuing

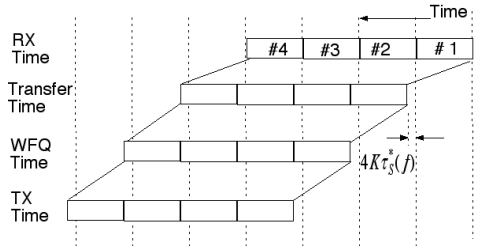


Fig. 7. Worst-case delay with big packets.

of big packets in a router. A big packet undergoes 4 phases: (i) During the *RX-Time*, the big packet is received at an input port. (ii) During the *Transfer-Time*, the big packet is transferred from the input to the output port as a sequence of cells. A worst-case delay of 4K $\tau_S^*(f)$ can be encountered, but not larger otherwise a contradiction occurs. (iii) During the *WFQ-Time*, the re-assembled big packet waits the maximum WFQ delay = $\tau_B^*(f)$ for access to the outgoing transmission line. (iv) During the *TX-Time*, the packet is transmitted. According to Fig. 7, each router queues at most 4 big packets, and the variable queueing delay is reduced to 4K $\tau_S^*(f)$ per router. Therefore the maximum end-to-end queueing delay over H routers is $4H(\tau_B^*(f) + K\tau_S^*(f))$ time-slots. ■

In summary, the use of variable-size IP packets with segmentation and re-assembly at each router has a relatively small impact on the end-to-end performance.

VII. Congestion Control, Buffer Sizes and QoS

The *Transmission Control Protocol* (TCP) carries most BE-Internet traffic [41]. TCP relies on the host-to-host principle, where senders do not receive explicit notification of congestion from the network. TCP varies the transmission rate in response to unacknowledged (dropped) packets. This principle allows routers to be relatively simple. There are many variants of TCP which use different heuristics to manage congestion [41]. The BE-Internet does not have any requirement on what TCP congestion-control algorithm should be used, nor does it have any enforcement on the transmission rate into the network. Denial-of-Service (DOS) attacks are common, where a large number of sources transmit high-rate traffic to a single destination, causing destination overload and potentially network-wide congestion.

All variants of TCP share several problems [41,42]; (1) TCP cannot provide any deterministic QoS guarantees to traffic flows; (2) TCP traffic flows typically receive highly-variable transmission rates, modulated by network congestion; (3) TCP has difficult providing *fairness* for competing flows with: (i) the same congestion control algorithm, (ii) different congestion control algorithms, and (iii) different 'Round Trip Times' (RTTs). It is well known that flows with small (RTTs) typically 'hog the bandwidth' relative to other flows.

TCP transmits traffic according to an *Additive Increase Multiplicative Decrease* (AIMD) rule to ensures stability [41]. When congestion is not detected, the transmission rate typically increases slowly. When congestion is detected through unacknowledged packets, the transmission rate is typically halved. In a fluid model, the transmission rate of a TCP flow can be viewed as a wave with a periodic sequence of crests and troughs. If several TCP flows crest at the same time at a router, the router will experience *'Transient Congestion'*, where the queues will overflow and drop packets. According to Cisco, *'Because network flows are additive, there is a high probability that when traffic exceeds the transmit queue length at all, it will vastly exceed the limit'* (www.cisco.com).

Existing Internet traffic can also be highly bursty, since the burstiness of Best-Effort TCP traffic flows is not constrained at the source. Commercial routers typically contain programmable traffic shapers which can constraint burstiness, although these are not often enabled for Best-Effort traffic. When traffic shapers are enabled, Cisco recommends that the *Maximum Burst Size* (MBS) of a TCP traffic flow is several seconds times the *Average Bit-Rate* (ABR) per second (Cisco Configuration Guide, V12.2). Otherwise, a TCP flow is unlikely to receive its desired ABR due to intense competition from other bursty TCP

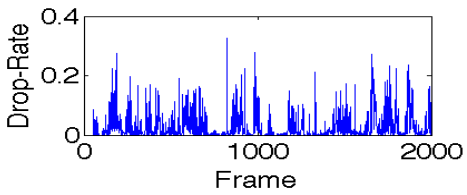


Fig. 8. Drop-rates of competing TCP traffic flows

flows. For example, according to Cisco a TCP flow with a ABR of 100 Mbps should allow for a MBS of 200-400 Mbits. We argue that the lack of burstiness controls at the traffic sources also contributes to *Transient Congestion* and high average queueing delays, since these bursts must be queued in the routers.

To minimize packet loss rates when congestion occurs, BE-routers typically use the classic *'Bandwidth-Delay-Product'* buffer sizing rule, where each link requires a buffer of $B = O(C \cdot T)$ bits, where T is the round-trip time on the network [35,36,37] and C is the link capacity. Commercial routers are typically built with 250 millisec buffers, to handle worst-case scenarios. A 100 Gbps link provisioned for a RTT of 250 millisec requires buffers for about 2 million maximum-size IP packets. A *'small buffer rule'* suggests that $B = O(CT/N^{1/2})$ where N is the number of long-lived TCP flows traversing the router, i.e., $B \approx$ fifty thousand IP packets [37]. Researchers have suggested a *'tiny buffer rule'* may apply, where $B = O(logW)$ and where W is the largest TCP congestion window size, under several simplifying assumptions [37]: (a) transient congestion does not occur, (b) the TCP flow rates are small relative to the link capacity, (c) the jitter of incoming traffic and IP routers is small, and (d) about 20% of the throughput of the BE-Internet is sacrificed. Unfortunately, these assumptions can be unrealistic with existing TCP practices.

To estimate the benchmark performance of the regular Best-Effort Internet carrying TCP traffic flows, consider the linear network of 10 routers using 40 Gbps links described in section V-D. Let there be 5 millisec of fiber following each router, for a 1-way fiber delay of 50 millisec and a RTT of $\geq$ 100 millisec. At 40 Gbps and full utilization, the number of outstanding (unacknowledged) TCP packets in the network is therefore $\geq$ 3.33 Million packets. The routers are configured to allow 50 millisec of queueing at each input port. All arriving packets which exceed this limit are dropped (called *Tail Dropping*). When congestion occurs, thousands of packets may be dropped before a TCP source detects any problem. We simulated the performance of this linear array of routers carrying 1,024 long-lived TCP traffic flows, which are all competing for link bandwidth. TCP sources were selected in a random order, and each source would transmit an average of $\leq$ 20 packets when selected. Each source maintains a TCP transmission window and implements the basic AIMD congestion-control scheme, using a minimum increment of 100 Kbytes when incrementing the window-size. Each router used the Best-Effort iSLIP scheduling algorithm [28]. Fig. 8 shows the instantaneous drop-rate, computed over each 'frame' of 100 time-slots. The average packet drop rate was 1.8%, and the peak instantaneous drop-rate was about 33%. The first few routers had very high buffer occupancies, with nearly 50 millisec of queueing delays each. These drop rates are quite high making it difficult for an *Internet Service Provider* (ISP) to sell this bandwidth, and the easiest solution for an ISP is to over-provision the network. This example illustrates some of the problems with conventional TCP congestion control, i.e., the high packet loss rates and lack of deterministic QoS guarantees, and provides the motivation for a better congestion control algorithm for data-center traffic.

The *Deterministic-Internet* achieves high throughput and very small buffers by exploiting GR-VCS connections on a packet-switched network, and constraining the traffic sources to transmit *Smooth* traffic at or below the provisioned guaranteed-rates. Using these techniques, the throughput of the BE-Internet can be increased, by removing (i) inefficient routing and scheduling, and (ii) the reliance on significant over-provisioning.

In our scheme, *Smooth* traffic flows have tight constraints on the allowable MBS, typically K = 4..8 maximum size packets. Consider a *Smooth* flow with an ABR of 1 Gbps and an MBS of 8 packets or 12,000 bytes. The ratio of the MBS to ABR is $\approx$ 0.01%. These burst sizes are much smaller than the

industry currently uses. Cisco recommends that a TCP flow uses a MBS equal to several seconds of the ABR (when the MBS is constrained). Otherwise, the TCP flow is not likely to get its desired bandwidth due to competition with other TCP flows. According to Cisco, the recommended ratio of the MBS to ABR is typically 2-4, up to 4000 times larger than the values advocated in this paper. According to Cisco's document *'MPLS Best Practices'*, even for MPLS-TE networks a wide range burst sizes can be used; examples of flows with an MBS of 1.3% and 150% of the ABR are shown. According to Cisco, congestion can occur in MPLS-TE networks, in which case the sources will typically halve their transmission rates, just as in TCP congestion-control.

The *Deterministic-Internet* supports the optional *Quasi-Smooth* (QS) traffic classes to handle legacy DiffServ and MPLS-TE traffic, which can be much burstier than *Smooth* traffic flows. The 3 QS traffic classes could allow for larger MBS values, for example 1%, 10% and 100% of the ABR. For these QS traffic classes, the router queuing delays will increase due to the larger burstiness, but service providers can control the over-provisioning, thereby controlling the average delays statistically.

The *Deterministic-Internet* represents an alternative to conventional TCP congestion-control. Sources can request a connection for a Smooth traffic flow (or traffic class) from the control plane. Given sufficient resources, this request is granted within a few RTTs, significantly faster than the time taken by TCP to perform one AIMD period for a high-rate traffic flow, which can take several hours [41,42]. Once established, a short-lived or long-lived TCP source can transmit packets without interruption, congestion or packet dropping, a level of deterministic QoS that even ATM or MPLS-TE networks cannot provide. Afterwards, the connection can be closed within a few RTTs, all before the time it takes for the conventional TCP to perform one AIMD period for a high-rate traffic flow (which can take several hours).

A. Cloud QoS

The Cloud has created an unprecedented opportunity to exploit aggregation and statistical multiplexing of independent TCP traffic flows, which the *Deterministic-Internet* exploits. Consider a large data-center providing 'Video-on-Demand' streams to 1 million customers located in 50 cities. A large data-center typically has 50,000-100,000 servers, where each server can transmit typically up to 1000 video streams. The real-time traffic leaving this data-center can therefore be modelled as the super-positioning of ≥ 5 million bursty TCP traffic flows. The data-center can provision 50 paths on the backbone network to the 50 cities, where all real-time traffic (i.e., voice and video) directed to one city can traverse one path. Each path can carry $\geq$ 20,000 independent TCP flows, as an aggregated smoothened stream with guaranteed QoS. Our research has shown that the aggregation of 10,000-100,000 independent bursty streams typically results in a single very low-jitter stream [31,32]. According to [31], to transmit 20,000 HD video streams (each with an ABR of 2 Mbps) with guaranteed QoS and low latencies between cities, each path can use a guaranteed-rate of 40 Gbps, with 1...5 % extra bandwidth to allow for some small burstiness. The *Deterministic-Internet* can deliver these aggregated flows with very low latencies, very high link utilizations (i.e., 95-100%), and deterministic QoS guarantees.

B. All-Optical Packet Switches

The proposed technologies can be used to realize a single-chip *Silicon-Photonics* all-optical packet switch as shown in Fig. 9. An electronic control plane can control the switch, using either the IETF or SDN technologies described earlier. Optical GR-VCS connections can be established in an all-optical network, potentially operating as a layer-2.5 all-optical network.

Packets arrive on incoming fibers, on multiple wavelengths. At each switch, the time-axis is divided into scheduling frames each consisting of F packet time-slots. Each incoming optical packet must be scheduled for transmission in one packet time-slot on an outgoing fiber and outgoing wavelength. The use of GR-VCS connections greatly simplifies the operation of the all-optical switch, as a result of the deterministic TDM-based periodic schedules: (i) packets arrive to each switch at deterministic times in a periodic schedule on each fiber, (ii) each packet will experience a deterministic queueing delay (using

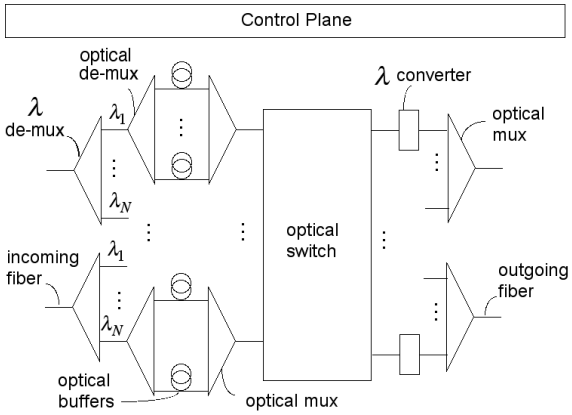


Fig. 9. A Deterministic All-Optical packet-switch with fiber-loop buffers

the Static-GPS flow-scheduling algorithm), and (iii) packets depart each switch at deterministic times in a periodic schedule on each fiber.

The control plane configures the optical-components to perform the switching: The optical demultiplexers are activated to forward packets into optical buffers at the correct time-slots. The optical multiplexers are activated to forward packets from optical buffers, through the optical switch to wavelength converters and onto the an outgoing fiber, in the correct time-slots. The optical switch is activated to perform the periodic *Queue-schedule* described in section V.

An integrated single-chip all-optical packet switch can be realized using the *Silicon-Photonics* technology [43]. This technology allows for the integration of CMOS logic along with optical waveguides, optical wavelength converters, and optical binary switches, all in the same integrated circuit. The optical packet buffers in Fig. 9 can use a small number of fiber delay loops, which are external to the *Silicon-Photonics* IC. It would be difficult to implement a regular Best-Effort Internet router using this technology, simply due to the vast amount of buffering required. However, by using the *Determinstic-Internet* router with deterministic TDM-based GR-VCS connections, the amount of buffering is reduced by several orders of magnitude, thus enabling the fabrication of single-chip optical packet switches. The packet buffers in Fig. 9 can also be implemented in CMOS, as the *Silicon-Photonics* technology integrates all-optical components and CMOS logic together.

C. The Energy Costs of Cloud Inefficiencies

Koomey et al. have estimated the energy costs of data-center inefficiencies [39]. Global data-centers used about 155 Billion KwHrs of energy in 2008, and operated at low utilizations of typically 20..30% [39]. Using industrial electricity rates of 7 cents per KwHr [40], the energy costs for global data-centers was estimated at $10.9 Billion US/year in 2008, of which 70-80% was wasted on low utilization [39,40]. Assuming the utilization of data-centers has risen to 50% by 2014 and that no new data-center capacity has been added, the energy-costs due to data-center inefficiencies are conservatively estimated to be $5.45 Billion US/year in 2014.

Using data in [38], the BE-Internet uses about 45 Billion KwHrs per year in 2010, and it also operates at low utilizations. Assuming the utilization of the Internet has risen to 50% by 2014 and that no new capacity has been added, the energy-costs due to Internet inefficiencies are conservatively estimated to be $1.6 Billion US/year in 2014. Given that Internet and data-center capacities are growing exponentially [1], these cost estimates are conservative.

These energy costs do not include the capital costs of low utilizations. According to its 2012 annual report, Cisco's sales of best-effort switching and routing equipment was $22 Billion US (and there are many other manufacturers). Assuming the utilization of the Internet has risen to 50% in 2014 (which is optimistic), the capital costs of Internet inefficiencies are estimated to be $11 Billion US/year. (The

capital costs of data-center inefficiencies are also very large, and are not considered here.) Together, the total costs of cloud inefficiencies exceed $18 Billion US/year in 2014. These costs provide the motivation to fix the *Best-Effort Internet*.

The *Deterministic-Internet* can operate links at very high (95...100%) utilizations, thereby increasing the aggregate capacity of the Internet, and reducing the costs of Internet inefficiencies. The US DOE has recently concluded that scientific cloud computing is infeasible, due primarily to the poor performance of the Internet [11]. The *Deterministic-Internet* can enable new time-critical services such as large-scale scientific cloud computing, thereby increasing the utilization of data-centers and reducing the costs of data-center inefficiencies. Finally, society can benefit from decreased greenhouse gas emissions, through improved energy efficiencies.

VIII. CONCLUSIONS

A *Deterministic-Internet* network that supports 2 traffic classes, the *Smooth* and *Best-Effort* classes, has been proposed. Traffic flows in the *Smooth* class are transported over *Guaranteed-Rate Virtual-Circuit-Switched* connections, and achieve exceptionally low end-to-end latencies, deterministic end-to-end QoS guarantees and significantly improved energy-efficiency. All *Best-Effort* TCP/IP Internet applications developed over the last 40 years will continue to run on the *Deterministic-Internet* without any changes. New cloud services can use the new and highly-efficient *Smooth* class to achieve significantly improved performance and energy-efficiency. The technologies can: (a) reduce router latencies and buffer requirements, (b) increase the Internet's aggregate capacity, (c) lower capital costs and operating costs of the Internet and data-centers, and (d) lower greenhouse gas emissions through improved energy-efficiency.

IX. ACKNOWLEDGEMENTS

The suggestions of the editors and reviewers are appreciated.

REFERENCES

[1] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures", IEEE Comm. Surveys and Tutorials, 2Q, 2011
[2] V. Jospeh and B. Chapman, "Deploying QoS for Cisco IP and Next-Generation Networks: The Definitive Guide", Elsevier/ Morgan-Kaufman Publishers, 2009.
[3] X. Xiao, L.M. Ni, "Internet QoS: A Big Picture", IEEE Network Magazine, March/April 1999.
[4] P. Gevros, J. Crowcroft, P. Kerstein and S. Bhatti, "Congestion Control Mechanisms and the Best-Effort Service Model", IEEE Network, May/June 2001
[5] A. Meddeb, "Internet QoS: Pieces of the Puzzle", IEEE Comm. Magazine, Jan. 2010
[6] L. Georgiadis, R. Guerin, V. Peris, K.N. Sivarajan, "Efficient Network QoS Provisioning Based on Per Node Traffic Shaping", IEEE/ACM Trans. Networking, Vol. 4, No. 4, Aug. 1996
[7] S. Shenker, "Fundamental Design Issues for the Future Internet", IEEE JSAC, Sept. 1995
[8] N. Shetty, G. Schwartz, J. Walrand, "Internet QoS and Regulations", IEEE/ACM Trans. Networking, Vol. 18, No. 6, Dec. 2010.
[9] V. Firoiu, JY Le Boudec, D. Towsley, and ZL Zhang, "Theories and Models for Internet QoS", Proc. IEEE, Vo. 90, No. 9, Sept. 2002
[10] Ciena Networks, "Ultra-Low-Latency Networking: Milliseconds can mean Millions", www.ciena.com, 2013
[11] K. Yelik, S. Coghlan, B. Draney, R.S. Canon, "Mallegan Report on Cloud Computing for Science", US Dept. of Energy, Office of Science, Dec. 2011,
[12] IETF RFC 2212, "Specification of Guaranteed Quality of Service", Sept. 1997
[13] IETF RFC 5865, "A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic", May 2010
[14] IETF RFC 5974, "NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling", Oct. 2010
[15] IETF RFC 5977, "RMD-QOSM: The Next Steps in Signalling (NSIS) Quality-of-Service Model for Resource Management in Diffserv", www.ietf.org, Oct. 2010
[16] H. Kim and N. Feamster, "Improving Network Management with Software Defined Networking", IEEE Comm. Mag., Vol. 5, no. 2, 2013
[17] N. McKeown, et al, "OpenFlow: Enabling Innovation in Campus Networks", ACM SIGCOMM Comp. Comm. Review 38.2, 2008
[18] A.K. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Service Networks: the Multiple Node Case", IEEE/ACM Trans. Networking, Vol. 2, No. 2, 1994,
[19] T.H. Szymanski, "Bounds on the End-to-End Delay and Jitter in Input-Buffered and Internally Buffered IP Networks", IEEE Sarnoff Symp., Princeton, NJ, March/April, 2009.
[20] T.H. Szymanski, "Memory Requirements in Future Internet Routers with Essentially-Perfect QoS Guarantees", IEEE Globecom Workshop CCNet, Miami, Florida, Dec. 2010.

[21] V. Anantharam, N. McKeown, A. Mekittikul and J. Walrand, "Achieving 100% Throughput in an Input Queued Switch", *Trans. Comm.*, vol. 47, no. 8, 1999.
[22] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks", IEEE Trans. Aut. Control, Vol. 27, Dec 1992
[23] P. Giaccone, E. Leonardi, D. Shah, "Throughput Region of Finite-Buffered Networks", IEEE Trans. Parallel Dist. Sys., Vol. 18, No. 2, Feb. 2007
[24] I. Keslassy, M. Kodialam, T.V. Lakshamn, and D. Stiliadis, "On Guaranteed Smooth Scheduling for Input-Queued Switches", *IEEE/ACM Trans. Networking*, Vol. 13, No. 6, Dec. 2005.
[25] W.J. Chen, C-S. Chang, and H-Y. Huang, "Birkhoff-von Neumann Input Buffered Crossbar Switches for Guaranteed-Rate Services, *IEEE Trans. Communications*, Vol. 49, No. 7, July 2001.
[26] C.E Koksal, R.G. Gallager, C.E. Rohrs, "Rate Quantization and Service Quality over Single Crossbar Switches", *IEEE Infocom*, 2004.
[27] S.R. Mohanty and L.N. Bhuyan, "Guaranteed Smooth Switch Scheduling with Low Complexity", IEEE Globecom, 2005
[28] N. McKeown, "The iSLIP Scheduling Algorithm for Input Queued Switches", IEEE Trans. Networking, Vol. 7, No. 2, April 1999
[29] T.H. Szymanski, "A Low-Jitter Guaranteed Rate Scheduling Algorithm for Packet-Switched IP Routers", *IEEE Trans. Comm.*, Vol. 57, No. 11, Nov. 2009
[30] T.H. Szymanski and D. Gilbert, "Provisioning Mission-Critical Telerobotic Control Systems over Internet Backbone Networks with Essentially-Perfect QoS", *IEEE JSAC,* Vol. 28, No. 5, June 2010
[31] T.H. Szymanski, "Max-Flow Min-Cost Routing in a Future Internet with Improved QoS Guarantees", IEEE Trans. Comm, Vol. 61, No. 4, April, 2013.
[32] T.H. Szymanski, "Maximum Flow Minimum Energy Routing in Exascale Cloud Computing Systems", IEEE PacRim Conf., Aug. 2013.
[33] G. Van der Auwera and M. Reisslein, "Implications of Smoothing on Statistical Multiplexing of H.264/AVC and SVC Video Streams', IEEE Trans,. Broadcasting, Vol. 55, No. 3, Sept. 2009
[34] T.H. Szymanski and D. Gilbert, "Internet Multicasting of IPTV with Essentially-Zero Delay Jitter", IEEE Trans. Broadcasting, Vol. 55, No. 1, March 2009
[35] R.S. Prasas, C. Dovrolis, M. Thottan, "Router Buffer Sizing for TCP Traffic and the Role of the Output/Input Capacity Ratio", IEEE/ACM Trans. Networking, 2009.
[36] S. Iyer, RR. Kompella, N. Mckeown, "Designing Packet Buffers for Router Linecards", IEEE Trans. Networking, Vol. 16, No. 3, June 2008
[37] Y. Ganjali, N, McKeown, "Update on Buffer Sizing in Internet Routers", ACM Sigcomm Comp. Comm. Rev., vol. 36, no. 5, Oct. 2006.
[38] B. Raghavan and J. Ma, "The Energy and Emergy of the Internet", Hotnets 2011
[39] J.G. Koomey, "Worldwide Electricity Used in Data-Centers", 2008 Environ. Res. Letters, IOP Science.
[40] E. Masanet, R.E. Brown, A. Shehabi, J.G. Koomey, and B. Nordman, "Estimating the Energy Use and Efficiency Potential of U.S. Data-Centers", Proc. IEEE, 2011
[41] A. Afanasyev, N. Tilly, P. Reiher and L. Kleinrock, "Host-to-Host Congestion Control for TCP", IEEE Comm. Surveys and Tutorials, 3Q, 2010.
[42] YT Li, D. Leith, R.N. Shorten, "Experimental Evaluation of TCP Protocols for High Speed Networks", IEEE Trans. Networking, Vol. 15, No. 5, Oct. 2007
[43] Y.A. Vlasov, "Silicon-CMOS Integrated Nano-Photonics for Computer and Data-Communications Beyond 100G", IEEE Comm. Mag., Feb. 2012

Ted H. Szymanski completed the PhD degree at the University of Toronto. He was affiliated with Columbia University and its *Center for Telecommunications Research*, and McGill University and the *Canadian Institute for Telecommunications Research*. From 1993 to 2003, he led the 'Optical Architectures' project within the Photonic Systems program within the *Networks of Centers of Excellence* of Canada. The program demonstrated a free-space 'intelligent optical backplane'. with about 1K microscopic laser channels. Contributors included Nortel Networks (now Ericsson), Newbridge Networks (now Alcatel), Lockheed-Martin/Sanders, and McGill, McMaster, Toronto and Heriot-Watt Universities. He holds two patents from this project, the first on intelligent optical networks, and the second on embedded FEC to improve throughput in parallel optical interconnects. His group also demonstrated the first FPGA with optical IO. From 2001-2011, he held the Bell Canada Chair in Data Communications at McMaster University. His research interests include QoS, energy-efficiency, the cloud and wireless networks.