Provisioning Mission-Critical Telerobotic Control Systems over Internet Backbone Networks with Essentially-Perfect QoS

Ted H. Szymanski, *Member, IEEE,* and Dave Gilbert, *Member, IEEE*

***Abstract*—Over the next decades, the Internet will evolve to support increasingly complex mission-critical services such as telerobotically controlled surgery. The world's first telerobotic surgery over the public Internet was performed in 2003, and since then several hundred more have been performed. Three critical requirements of these services include: (i) essentially 100% restoration capability, (ii) small and bounded end-to-end queuing delays (ie $<$ 250 millsec), and (iii) very low-jitter communications (ie $<$ 10 millisec). In this paper, algorithms to provision mission-critical services over the Internet with essentially 100% restoration capability and essentially-perfect QoS are proposed, building upon two theoretical foundations. Mission-critical traffic is routed using the theory of shared backup protection paths or p-cycles, while background traffic is routed using multiple edge-disjoint paths. Mission-critical traffic is scheduled using the theory of recursive stochastic matrix decomposition to achieve two constraints: (i) near-minimal end-to-end queuing delay and jitter and (ii) essentially-perfect QoS. Designs of the Application-Specific Token-Bucket Traffic Shaper Queues (ASTSQs) and the Application-Specific Playback Queues (ASPQs) for telerobotic services are provided. To test the theory, extensive simulations of a saturated Internet backbone network supporting telerobotic services along with competing background traffic (ie VOIP, IPTV) are reported. It is shown that all mission-critical traffic can be delivered while meeting the three critical requirements, even in fully saturated backbone IP networks.**

***Index Terms*—telerobotic control, telerobotic surgery, quality of service, fault tolerance, reliability, availability, router, scheduling, low-jitter, stochastic matrix decomposition**

I. Introduction

THE CURRENT Internet network is one of the milestone achievements of the 20-th century, and it illustrates many technical challenges and opportunities. Many current IP routers use 'Input-Queued' (IQ) switches which require unity speedup. However, the provisioning of services with essentially-perfect Quality of Service (QoS) in a network of IQ routers is an unsolved problem. According to the US *National Science Foundation (NSF)*, the Internet also faces challenges of security and seemless integration of wireless technologies [1], problems which have *'plagued the Internet'* since its inception [2]. To compound the problems, the *US Federal Communication Commission (FCC)* has required that all television broadcasts be converted to digital format in 2009, and the increasing importance of IPTV traffic over the Internet is also causing major technical problems [3]. To solve these problems, the NSF recently initiated the *'Global Environment for Network Innovations'* (GENI) program which is open to a complete *'clean-state'* redesign of the Internet if necessary [1][2]. In summary, new approaches to handle QoS, security, reliability and wireless technologies within the Internet are needed.

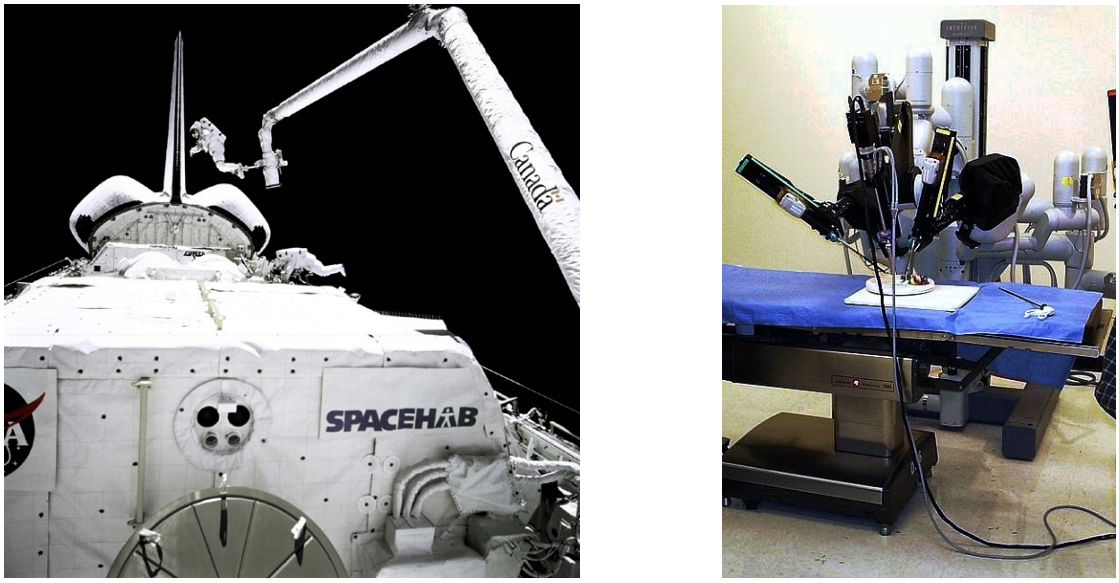
Further adding to the challenges, the use of mission-critical telerobotic services over the Internet has been growing [4-9]. In recent years robotic control systems have progressed from large complex systems such as the Space Shuttle Robotic Arm shown in Fig. 1a, to the DaVinci *'minimally invasive surgery'* robotic system shown in Fig. 1b ($en.wikipedia.org$). The DaVinci system allows a surgeon to operate on a patient within the same room. Several precisely-controlled robotic arms are inserted into a patient using small openings. The surgeon controls the procedure at a nearby *'surgeons console'*, which provides a high-resolution 2D or 3D visual environment along with robotic controllers and sensors. By 2007, there were 795 shipments of the DaVinci robot worldwide, and the unit was used in approx. 50,000 radical prostatectomies in the USA, a 50% growth over 2006 [5]. The possibility of telerobotically controlled surgery, where the expert surgeon is in one location while the patient and medical support team are perhaps thousands of miles away, is an appealing concept which can potentially improve the delivery of medical services world-wide [4][5].

The world's first trans-Altantic minimally-invasive telerobotic surgery on a human, named Operation Lindberg, was performed in 2001 between Strasbourg, France and New York city, using a constant-bit-rate service over a trans-atlantic fiber channel [9]. The world's first telerobotic surgery over the public Internet was performed in 2003 by Dr. Anvari of McMaster University and St. Josephs Hospital in Hamilton, using a VPN configured over an IP/MPLS backbone network of Cisco routers managed by Bell Canada [5]. Since then, several hundred more operations have been performed. NASA has initiated experiments with telerobotic surgey, for potential use in its space program and future MARS missions. A test facility named Aquarius has been constructed on the ocean floor to simulate the space station, and several missions called the *'NASA Extreme Environment Mission Operations'*(NEEMO) have been performed. Experiments with surgical telecontrolled robots from SRI [12] where performed in the NEEMO 12 mission ($http://www.nasa.gov/mission\_pages/NEEMO$).

Manuscript received 1 April 2009; revised 1 November 2009.

T.H. Szymanski and D. Gilbert are with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, L8S 4K1 Canada e-mail: teds@mcmaster.ca.

Digital Object Identifier 10.1109/JSAC.2010.100602.



(a) (b)

Fig. 1. (a) Space-Station robotic arm (www.nasa.org). (b) Da Vinci Medical Robot (www.intuitivesurguical.com).

Telerobotic control systems have stringent real-time QoS constraints. Unbounded delay and jitter can lead to control loop instability and mission failure. Without compression, a high-resolution medical video stream requires 270 Mbit/sec [10-11]. Reference [10] recommends MPEG 2 video encoding using only the I-frames in a Group of Pictures (GOP) structure, for a bit-rate of between 10 and 40 Mbps and a low video encoding delay. The use of 2 video streams for 3D vision would require 20 - 80 Mbps. The robotic control stream requires $<$ 50 Kbps. For telerobotic surgery, doctors have reported that maximum delays of up to 250 millisec may be tolerated. However, the jitter must relatively small, of the order 10s of milliseconds, and hence traditional TCP flow-control cannot be used.

According to Dr. Dave Williams, a medical doctor and former NASA astronaut, and currently Director of the McMaster University Centre for Medical Robotics which is leading research into medical robotics and space medicine, the *'operating room of the future will look significantly different from the operating room of today'*. There may be much more reliance on medical robots and telerobotic control systems [14]. Given the current challenged state of the Internet and the growing use of telerobotic systems, there is a significant divide to be conquered. In this paper, algorithms to statically provision end-to-end mission-critical traffic flows in an IP/MPLS network of IQ switches with (i) essentially 100% restoration capability and (ii) essentially-perfect QoS are described.

Consider the problems of the underlying Internet infrastructure. An IQ crossbar switch is shown in Fig. 2a. Each input port has N *Virtual Output Queues (VOQs)*. Each $VOQ(j,k)$ stores the cells arriving at input port j to be forwarded to output port k. The VOQs remove the Head-of- Line blocking found in traditional IQ switches. A scheduling algorithm is used to schedule the transmission of conflict-free sets (permutations) of $\leq$ N cells from the VOQs to the output ports in every time-slot. A switch with a combination of IQs and Crosspoint-Queues (CIXQ) also places FIFO queues at each crosspoint. Our proposed algorithms will apply to IP/MPLS networks using both IQ and CIXQ switches and switches with Combined Input and Output Queueing (CIOQ). Scheduling traffic within such networks to meet QoS guarantees is a difficult problem which is summarized in section 3.

Currently services are provisioned over the Internet using *'Traffic Specifications' (Tspecs)*, which specify the average bit rate, maximum (burst) bit rate, burst size, average delay, maximum delay and packet loss rate [15]. A typical end-to-end cell delay variation (CDV) over the Internet is shown in Fig. 2b, where alpha is the maximum allowable probability a packet is lost due to excessive delay. This plot illustrates one problem with the current Internet, the statistical delay performance and its inability to achieve rigorous provably small delay bounds.

Consider any backbone IP/MPLS network of IQ switches operating at capacities $\leq$ 100% with unity speedup. Assume that variable size IP packets are segmented into fixed size cells at the ingress router of an MPLS domain, and reconstructed into IP packets at the egress router of the domain. The same results hold for variable-size packets, when adjusted for packet-size. It has been recently established in theory that given any admissible routing of shaped traffic flows within this network, every competing end-to-end traffic flow can be delivered with essentially-perfect end-to-end QoS, where the number of cells queued per flow per switch is near-minimal and bounded, and where the end-to-end normalized queueing delay, jitter and service lead/lag are near-minimal and bounded [16-21]. By using a small playback buffer with finite size, all network-introduced delay jitter can be provably removed, ie all traffic flows can be delivered with zero network-introduced

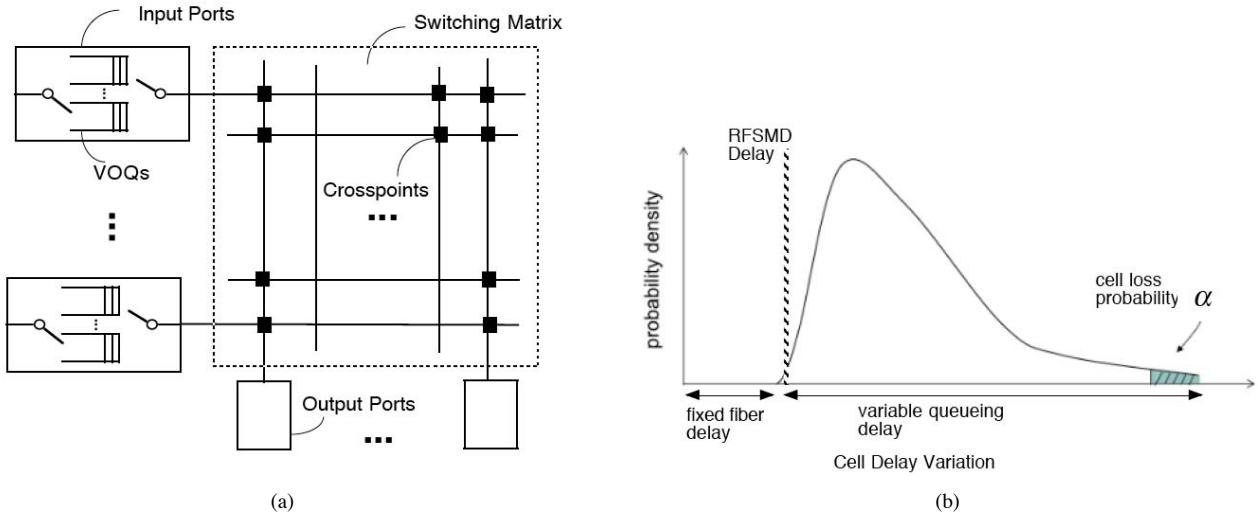


Fig. 2. (a) IQ Crossbar switch. (b) Tspec - End-to-End Cell Delay Variation.

delay jitter. These theories apply to an IP/MPLS network of IQ, CIXQ or CIOQ switches, and explain experimental results reported in [17][18][19][20].

In this paper, we apply the theory in [16-21] to the provisioning of mission-critical telerobotic control systems over the COST266 pan-European backbone network shown in Fig. 3 ($http : alabamamaps.ua.edu$). Telerobotic services are provisioned between every pair of cities, representing a significant demand on the network. At each ingress point to the network, each bursty telerobotic traffic stream is shaped by an Application-Specific token bucket traffic Shaper Queue (ASSQ) to limit burstiness. At each destination node, the traffic is received with very low network-introduced delay jitter. Zero-jitter bursty video frames are reconstructed using an Application-Specific Playback Queue (ASPQ). The designs of these important components are presented.

This paper: (a) briefly summarizes the state of the art in mission-critical telerobotic surgery systems, (b) characterizes the traffic requirements for such systems, (c) addresses the issue of 100% restoration capability, (d) applies the theory of recursive fair stochastic matrix decomposition to the scheduling of mission critical traffic with essentially-perfect QoS, and (e) presents extensive simulations over the COST266 European backbone network to corroborate the theory. The simulator is custom-written and contains over 20,000 lines of code. It is shown that mission-critical systems which achieve essentially 100% restoration along with essentially-perfect QoS over any backbone Internet network are achievable, consistent with theory. While the COST266 topology is used as an illustrative example herein, the same theories apply to scheduling mission-critical traffic in infrastructure Wireless Mesh networks [18].

Section 2 reviews reliable routing and low-jitter scheduling in the Internet network. Section 3 describes the RFSMD scheduling algorithm. Section 4 describes the telerobotic traffic model. Section 5 introduces the multipath routing algorithm. Section 6 presents experimental results. Section 7 presents a comparison with the state-of-the-art methods.

II. The Internet Network Model

1) Reliable Routing: Assume an IP/MPLS network model, where each node has an optical crossconnect (OXC) and a label-switched MPLS router (LSR), the same model as in [22][23]. The two most common failures modes in this model include: (1) failure in the WDM layer of lightwave spans arising from fiber cuts or component failures, and (2) failures in the IP/MPLS layer of routers, arising from component failures or software crashes [23].

One approach to achieve reliable end-to-end communications in an IP/MPLS network is the *'Shared Backup Path Protection' (SBPP)* scheme [23]. In this scheme, each working path has a pre-arranged end-to-end backup path which is node-disjoint from the working path. Referring to Fig. 4, two node-disjoint paths between routers 1 and 4 are shown, the working and backup paths. However, due to combinatorics many more pairs of node-disjoint paths between 1 and 4 exist. The SBPP scheme can be applied to each end-to-end connection, at either the WDM or the IP/MPLS layers [23]. One shared backup path can be used by multiple disjoint working paths, to reduce the bandwidth consumed for protection. When a failure is detected on a working path, a restoration process is invoked, where the traffic flows are diverted to the backup path. Pre-planned forwarding tables in the MPLS layer are invoked to realize the diversion of traffic.

An alternative approach involving multi-layer optimization of an IP/MPLS network was addressed in [23]. Three schemes were proposed and evaluated for their ability to provide multilayer restoration: (i) the intrinsic p-Cycle protection, (ii) 'Node Encircling p-Cycles' (NEPC) protection, and (iii) 'Node-Flow p-Cycles' (NFPC) protection. When a failure is detected, a restoration process is invoked where the traffic flows on the failed component are diverted over a suitable p-cycle. The over-subscription of traffic on any link is bounded during the planning process. The schemes were evaluated on their ability to provide 100% restoration, while minimizing the oversubscription of traffic on links as a result of failures. Oversubscription occurs when a link in the network must carry its originally allocated traffic, plus traffic re-routed in response

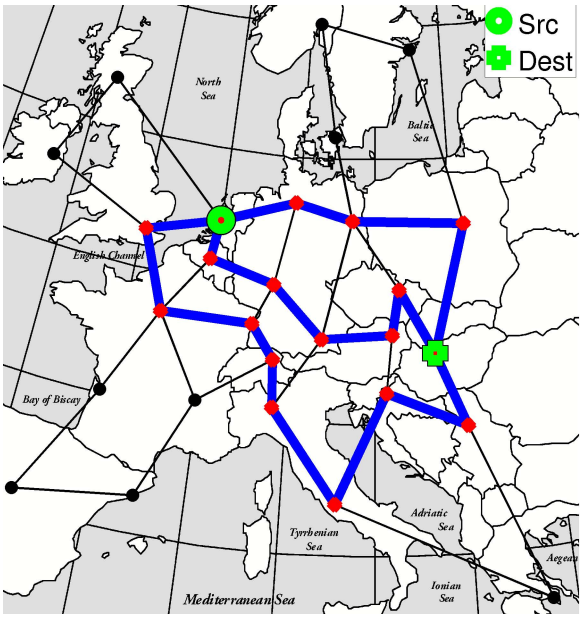


Fig. 3. COST266 Pan European Network with 3 node-disjoint paths from Amsterdam to Budapest.

to the failure, which increases the statistical delay and reduces QoS.

In both schemes (SBPP and p-cycles), each solution is designed specifically for a given global traffic demand matrix for a given topology. The authors described some insights into the practical application of these strategies [23]. At the WDM layer, given a traffic demand matrix and network topology, the SBPP or p-cycles are designed offline. Once defined, they operate in a 'trip-wire' manner, where they are invoked when a lightwave span failure is detected [23]. At the IP/MPLS layer, a central network control site receives information from neighboring nodes about a node failure, and a pre-planned restoration process is invoked, where neighbors of the failed node are reconfigured to restore the affected traffic flows. Each neighbor is reconfigured either by using a pre-planned set of LSP routing tables, or by downloading a routing table in real-time from a central control site. In summary, the restoration processes can provide fast response to lightwave span failures, of the order of 80 milliseconds, and can provide slower response to node failures, of the order of a few seconds [23]. Our approach to achieve 100% restoration capability for mission-critical traffic can use either the SBPP or the p-cycle schemes.

2) Scheduling Traffic for Essentially-Perfect QoS: The problems of scheduling traffic in a network to achieve maximum throughput along with simultaneous QoS bounds has a long history. A model for *constrained queueing systems* such as networks of IQ switches was proposed in [24]. They considered dynamic scheduling policies where the selection of active queue servers for each time-slot are based on a *Maximum Weight Matching (MWM)* of a bipartite graph for each time-slot. They established that the dynamic MWM scheduling algorithm can achieve bounded queue sizes (stability) within the Capacity Region of the network, and can reach essentially 100% of the achievable capacity. However, the solution of a

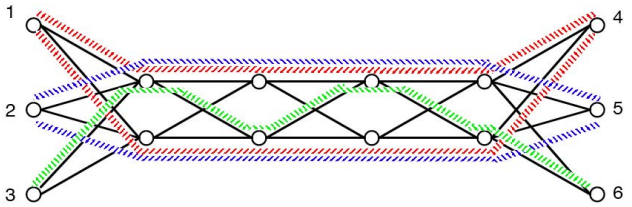


Fig. 4. Multipath Routing.

MWM over N servers has complexity $O(N^3)$, which renders the algorithm intractable for realistic networks. Furthermore, the MWM algorithm can result in large queue sizes before stability is reached; in [25] the queuing delay is $O(1/(1-\rho))$ and approaches infinity as $\rho \rightarrow 1$. In practice, heuristic scheduling algorithms such as the PIM and iSLIP algorithms are used in commercial routers and switches [26]. Heuristic schedulers are implemented in hardware and attempt to find sub-optimal *Maximal Matchings (MM)* of a bipartite graph in every time-slot. Given a 40 Gbps link and a 64-byte cell, the time-slot duration equals 12.8 nanoseconds. Due to the stringent time-constraints and inherent sub-optimality of all MM algorithms, heuristic schedulers cannot achieve 100% throughput, bounded queue sizes or rigorous QoS guarantees.

The problem of achieving small or bounded jitter in one IQ switch is equally difficult. The problem of scheduling traffic to minimize jitter in one IQ switch with unity speedup can be formulated as an NP-HARD integer programming problem [27,28]. The traffic rates to be supported by one NxN crossbar switch can be specified in an NxN traffic rate matrix Γ :

$$\Lambda = \begin{pmatrix} \lambda_{0,0} & \lambda_{0,1} & \cdots & \lambda_{0,N-1} \\ \lambda_{1,0} & \lambda_{1,1} & \cdots & \lambda_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ \lambda_{N-1,0} & \lambda_{N-1,1} & \cdots & \lambda_{N-1,N-1} \end{pmatrix},$$

where $\sum_{i=0}^{N-1} \lambda_{i,j} \leq 1$ and $\sum_{j=0}^{N-1} \lambda_{i,j} \leq 1$.

Each element $\lambda_{i,j}$ represents the fraction of the link rate reserved for the traffic rate between the IO pair (i,j) over the $VOQ(i,j)$. Let X^k for $k = 1, 2, ..., K$ be matrices with elements $x_{i,j}$ in a set of K permutation matrices. Given traffic rate matrix Λ for one switch with elements $\lambda_{i,j}$, the objective is to decompose the matrix Λ into a set of constituent permutations matrices X^k along with associated weights θ_k, as in the following integer programming problem (ILJD problem [27]):

$$D = \min \sum_{k=1}^{K} \phi_k$$

subject to

$$\sum_k \phi_k x_{ij}^k \geq \lambda_{ij} \qquad \forall i,j \qquad (1)$$

$$\sum_k x_{ij}^k = 1 \qquad \forall i,j \qquad (2)$$

$$\sum_i x_{ij}^k \leq 1 \qquad \forall j,k \qquad (3)$$

$$\sum_j x_{ij}^k \leq 1 \qquad \forall i,k \qquad (4)$$

$$x_{ij}^k \in \{0,1\} \qquad \forall i,j,k \qquad (5)$$

Constraints (3)-(5) imply that X^k is a partial permutation matrix. Constraint (1) implies that the weighted sum of the permutation matrices X^k is greater than the traffic rate matrix Λ being decomposed. Constraint (2) implies that each element in the rate matrix belongs to exactly one element in the set of matrices in the decomposition. (The above formulation requires that matrices X^k are partial permutation matrices.) The bandwidth requirement of the decomposition is therefore $\sum_{i=1}^k \theta_k$. It is established in [27] that the problem is NP_Hard.

A tractable *Greedy Low-Jitter Decomposition (GLJD)* with complexity $O(N^3)$ was also proposed by Bell Labs. [27,28]. However, it requires a worst-case speedup of $O(logN)$ and can achieve relatively low jitter schedules for loads $\leq$ about 80%. A frameless mathematical scheduling algorithm called the *Birkoff von Neumman (BVN)* decomposition was proposed by NTU [29]. Under the constraint of unity speedup, the jitter and service lag are bounded by $O(N^2)$ time-slots [29].

In the BVN decomposition, given a doubly-substochastic or stochastic traffic rate matrix Λ for an NxN crossbar switch which meets the following 2 conditions, $\sum_{i=0}^{N-1} \lambda_{i,j} \leq 1, \forall j$, and $\sum_{j=0}^{N-1} \lambda_{i,j} \leq 1, \forall i$, then the original matrix Λ can be decomposed into a set of positive numbers θ_k and permutation matrices $X^k, k = 1, ..., K$ for some $K \leq N^2 - 2N + 2$ that satisfies the following two equations: $\Lambda \leq \sum_{k=1}^K \theta_k X^k$ and $\sum_{k=1}^K \theta_k = 1$. However, the complexity of the decomposition is $O(N^{4.5})$. The BVN decomposition is equivalent to the Bell Lab's ILJD optimization problem, where constraint (3) has been relaxed. Furthermore, the permutation matrices must be scheduled, which introduces additional computational complexity.

Another frame-based low-jitter scheduling scheme was developed at MIT [30]. With speedup $S = 1 + sN$ between 1 and 2, the maximum *'Service Lag'* over all IO pairs was shown to be bounded by $O((N/4)(S/(S-1)))$ time-slots. It was shown that for a $S = 2$ the service lag is $O(N)$ time-slots, whereas it can be $O(N^2)$ time-slots for unity speedup. For a 256x256 IQ switch with a speedup $S = 2$, the service lag bound is 128 time-slots, whereas with unity speedup it can be 65,000 time-slots [30]. Another greedy stochastic matrix decomposition algorithm was proposed by UCR in [31]. This algorithm also decomposes a traffic rate matrix for one switch into a convex set of permutation matrices and associated weights which must be independently scheduled, as in the prior ILJD and BVN methods. The algorithm is relatively quick but it cannot guarantee short-term fairness or 100% throughput. The authors establish a jitter bound which grows as the switch size N increases, and identify an open problem: *"to determine the minimum speedup required to provide hard guarantees, and whether such guarantees are possible at all"* [31]. In summary, tractable scheduling algorithms which achieve stability within the capacity region and which achieve bounded delays, jitter or service lags under the constraint of unity speedup, in one IP/MPLS router or a network of IP/MPLS routers, are unknown. These issues are revisited in section 6 after our contributions have been presented.

III. Low-Jitter Scheduling Using Stochastic Matrix Decomposition

In a Guaranteed-Rate (GR) scheduling algorithm with unity speedup, all the traffic in the matrix Λ must be scheduled in a scheduling frame consisting of F time-slots. Define a new quantized traffic rate matrix R where each traffic rate $R(j,k)$ is expressed as an integer number of time-slot reservations for IO pair (j,k) per scheduling frame with F time-slots:

$$R = \begin{pmatrix} R_{0,0} & R_{0,1} & \cdots & R_{0,N-1} \\ R_{1,0} & R_{1,1} & \cdots & R_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ R_{N-1,0} & R_{N-1,1} & \cdots & R_{N-1,N-1} \end{pmatrix}$$

where $\sum_{i=0}^{N-1} R_{i,j} \leq F$ and $\sum_{j=0}^{N-1} R_{i,j} \leq F$.

To be admissible, the total number of time-slot reservations for traffic leaving the router over output port(i) $\leq F$, and the total number of time-slot reservations for traffic arriving at the router over input port(j)$\leq F$.

A *Low-Jitter Guaranteed-Rate* scheduling algorithm with unity speedup based on *Recursive Fair Stochastic Matrix Decomposition (RFSMD)* was proposed in [16-21]. A doubly substochastic or stochastic traffic rate matrix is quantized to have integer values, and then is recursively decomposed in a fair manner. Let $P(M,F)$ denote the problem of scheduling an admissible quantized traffic rate matrix M into a scheduling frame of length F time-slots. The problem $P(M,F)$ is recursively decomposed into 2 problems $P(M_1, F/2)$ and $P(M_2, F/2)$, such that matrices $M_1 + M_2 = M$, where M_1 and M_2 are admissible traffic rate matrices, and for all j and k where $0 \leq j < N$ and $0 \leq k < N$, then $M_1(j,k) \leq M_2(j,k) + c$ and $M_2(j,k) \leq M_1(j,k) + c$ for $c = 1$. This step is a combinatorial problem and relies upon the theory of routing permutations in a rearrangeably nonblocking switching network [17]. One step in the decomposition for a 4x4 matrix operating at 99.2% load with unity speedup is shown in Eq. 6 [18]. Given an NxN switch and a fixed scheduling frame length F, the RFSMD matrix decomposition algorithm [16-21] bounds the service lead and service lag (formally defined ahead) for the aggregated traffic leaving any node to $\leq K$ IIDT time-slots for constant K, where IIDT represents the *'Ideal Inter- Departure Time'* for cells belonging to the aggregated traffic leaving an edge. Furthermore, the bound applies to all individual competing traffic flows traversing each edge, provided that cells are selected for service within each VOQ according to a GPS scheduling algorithm [17].

$$\begin{bmatrix} 106 & 222 & 326 & 345 \\ 177 & 216 & 303 & 326 \\ 459 & 232 & 183 & 147 \\ 282 & 352 & 211 & 178 \end{bmatrix} = \begin{bmatrix} 53 & 111 & 163 & 172 \\ 88 & 108 & 152 & 163 \\ 230 & 116 & 91 & 74 \\ 141 & 176 & 105 & 89 \end{bmatrix} + \begin{bmatrix} 53 & 111 & 163 & 173 \\ 89 & 108 & 151 & 163 \\ 229 & 116 & 92 & 73 \\ 141 & 176 & 106 & 89 \end{bmatrix} \quad (6)$$

In section 6, exhaustive simulations of several hundred traffic specifications provisioned over the COST266 European optical backbone network are presented and several technical parameters will be observed and plotted. The following technical definitions are necessary to interpret the graphs presented in section 6. Similar definitions are presented in [16-18][30].

Definition: A 'traffic flow' in a network specifies a guaranteed traffic-rate to be achieved between a source-destination pair (a, z). This traffic flow must be routed along an end-to-end path of routers $(a, b, \ldots, y, z)$, in which appropriate buffer space is reserved in each router and in which the required bandwidth is reserved in each link in the path, ie $(a, b), \ldots, (y, z)$.

Definition: A Frame transmission schedule for one router is a sequence of F partial or full permutation matrices which define the crossbar switch configurations for F time-slots within a scheduling frame, ie $P \equiv \{P(t)\}, 0 \leq t \leq F-1$, where $P_{j,k}(t) = 1$ if VOQ(j,k) has a scheduled service opportunity in time-slot t. Each permutation matrix identifies up to N conflict-free VOQs for service. Given a line-rate L, the frame length F is determined by the desired minimum quota of reservable bandwidth = L/F. To achieve $L/F \leq 1\%$ of L, set $F \geq 100$, ie $F = 128$.

Definition: A Flow transmission schedule for one router is a sequence of F matrices which define the flow to be serviced in each VOQ for the F time-slots within a scheduling frame, given a frame transmission schedule which identifies the VOQs to be serviced, ie $\mathbf{Z} \equiv \{Z(t)\}, 0 \leq t \leq F-1$, where $Z_{jk}(t) = f$ if flow f within VOQ(j,k) has a scheduled service opportunity in time-slot t. A flow transmission schedule can be computed from the frame transmission schedule; When VOQ(j,k) receives a service opportunity, a flow f traversing VOQ(j,k) is selected for a service opportunity using the GPS algorithm.

Definition: Let $s(f, c)$ denote the service time of cell c in flow f. The Inter-Departure Time (IDT) of cell c in a GR flow f is defined as $s(f, c) - s(f, c-1)$ for $c \geq 2$. The Ideal Inter-Departure Time (IIDT) of cells in a GR flow f with quantized GR of $\theta(f)$ time-slot reservations per frame is given by IIDT(f) = $F/\theta(f)$ time-slots of duration (C/L) sec.

Definition: Given the set of all flows where each flow f traverses a VOQ(j,k), the cumulative service is a sequence of F vectors $\mathbf{S} \equiv \{S_f(t)\}, 0 \leq t < F$, where $S_f(t)$ equals the number of service opportunities for flow f in VOQ(j,k) in the interval $[0, t]$. The cumulative arrivals is a sequence of vectors $\mathbf{A} \equiv \{A_f(t)\}, 0 \leq t < F$, where $A_f(t)$ equals the number of cells arriving for flow f in VOQ(j,k) in the interval [0,t]. The cumulative departures is a sequence of vectors $\mathbf{D} \equiv \{D_f(t)\}$, $0 \leq k < F$, where $D_f(t)$ equals the number of cells which depart for flow f in VOQ(j,k) in the interval $[0, t]$. The Q backlog is a sequence of F vectors $\mathbf{Q} \equiv \{Q_f(t)\}$ where $Q_f(t) = \lfloor A_f(t) - D_f(t) \rfloor$ for $0 \leq t < F$, where $Q_f(t)$ equals the positive part of the cumulative arrivals - cumulative departures for flow f in VOQ(j,k) at time t. The Service lead/lag is a sequence of F vectors $LL \equiv \{LL_f(t)\}, 0 \leq t < F$, where at time-slot t $LL_f(t) = S_f(t) - (t/F)\theta(f)$. Intuitively, a positive Service Lag represents how many time-slots behind service the flow has fallen, relative to an ideal service schedule. A negative Service Lag is called a Service Lead, and represents how many time-slots ahead of service the flow has jumped. The *Normalized Service Lead/Lag* of a flow is defined as the Service Lead/Lag of the flow expressed in cells or packets. A positive normalized service lag represents how many cells behind service the flow has fallen, relative to an ideal service schedule. A negative normalized service lag represents how many cells ahead of service the flow has jumped. The concept of the normalized service lead/lag is critical to establishing the 4 theorems to follow shortly.

Consider a discrete-time queueing model, where time is normalized for all flows and is expressed in terms of the IIDT for each flow. The following notations presented in [18] are used. The cumulative arrival curve of a traffic flow f is said to conform to $T(\lambda, \beta, \delta)$, denoted $A_f \sim T(\lambda, \beta, \delta)$, if the average cell arrival rate is λ cells/sec, the burst arrival rate is $\leq \beta$ cells/sec, and the maximum normalized service lead/lag is δ. A similar notation is used for cumulative departures and cumulative service. In any router, the cumulative departure curve for f is said to 'track' the cumulative service curve for f when cell departures are constrained by the scheduled service opportunities. This situation occurs when flow f has queued cells at VOQ(j,k).

The following four theorems were established in [18]. Assume each traffic flow is admitted to an IP/MPLS network subject to an *Application-Specific Token-Bucket Traffic Shaper Queue (ASSQ)*, and has a maximum normalized service lead/lag of K cells. The traffic rate matrix for each router is updated by a resource reservation protocol such as RSVP or DiffServ [27,28]. Each router is scheduled using the proposed RFSMD algorithm with a maximum normalized service lead/lag of K cells [17]. Assume fixed size cells, with any reasonable cell size. Similar bounds apply for variable-size IP packets.

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

Theorem 2: When all queues in all intermediate nodes have reached steady-state, the maximum end-to-end queueing delay of a GR flow traversing H routers is $O(KH)$ IIDT time-slots.

Theorem 3: In the steady-state, the departures of traffic flow f at any IQ router along an end-to-end path of H routers are constrained by the scheduling opportunities, and will exhibit a maximum normalized service lead/lag of K, ie $Sf \sim T(\theta(f), \beta, K)$. The normalized service lead/lag of a flow is not cumulative when traversing multiple routers.

Theorem 4: A traffic flow which traverses H IQ routers along an end-to-end path can be delivered to the end-user with essentially-zero network-introduced delay jitter, when

a playback buffer of size 4K cells is employed, ie $S_f \sim T(\theta(f), \beta, K)$.

The above 4 theorems will be applied to simplify the routing problem in IP networks in section 5.

IV. Telerobotic Traffic Model

According to [10,11], an uncompressed video stream for a telerobotic unit can utilize 270 Mbps, while with MPEG-2 coding of I-frames the bit-rate can be reduced to 10-40 Mbps. A 3D vision system would require 2 video streams for a bitrate of 20-80 Mbps. Let a single *'telerobotic session'* equal 2 video streams at 20 Mbps each, one telerobotic control stream at 64 Kbps, and an additional 128 Kbps for digitized voice, for a bandwidth of 40 Mbps. An average of 4 telerobotic sessions will be aggregated and provisioned between every directed pair of cities (a,b) in the COST266 backbone network shown in Fig. 3. Each session will have an *'excess bandwidth'* of 40 Mbps (for reasons explained ahead), for a total provisioned bit-rate of 80 Mbps. This represents a significant load of guaranteed-rate telerobotic traffic.

The following network parameters are used. Each link in the COST266 network has a bandwidth of 40 Gbps. A scheduling frame of size F=4K time-slots is used in every router. By the earlier definitions, each time-slot reservation in a scheduling frame represents a bandwidth reservation of 40 Gbps/4K = 10 Mbps. To reserve the 80 Mbps for a directed telerobotic session, 8 time-slot reservations per scheduling frame are required.

The excess bandwidth will be used to accommodate some burstiness in the video traffic within the network, and (perhaps surprisingly) it will be shown in section 7 to determine the end-to-end delay. Each bursty video stream is shaped by an *Application-Specific token bucket traffic Shaper Queue (ASSQ)* at the source. Incoming video frames are fragmented into fixed size cells, which are buffered in the ASSQ. The ASSQ will lower the burstiness by limiting the injection rate of cells into the network to at most 80 Mbps, and will have the capacity to buffer several video frames of cells while they await transmission. The ASSQ will introduce an application-specific delay and jitter into the video stream at the source, which depends upon the burstiness of the incoming video stream. Each destination node has an *Application-Specific Playback Queue (ASPQ)*, which is used to filter out any network-introduced jitter, and reconstruct the 2 original bursty video streams at the surgeon's console with delay $\leq$ 250 milliseconds and zero video frame jitter.

To find the burstiness of a video stream encoded with an MPEG-2 codec using only I-frames, we processed the 'Terminator-2' video stream available at the University of Arizona [32]. The video trace uses a GOP format of (IBBPBBPBBPBB). The I-frames have (minimum, average and maximum) sizes of (14K, 25K, 50K) bytes respectively. We normalized the I-frame statistics to retain the same relative burstiness at a data-rate of 20 Mbps, to represent one telerobotic video stream.

V. Multipath Routing in the Internet Backbone

This section summarizes four key concepts: (1) the generation of the traffic specifications to be routed in the COST266 network, (b) two routing problem formulations for backbone networks, the single path and multipath routing problems, (c) a simple greedy solution to the multipath routing problem, and (d) a description of how the traffic rate matrices used for scheduling are computed from the routing information.

Each *'traffic specification'* consists of a set of mission-critical telerobotic sessions to be provisioned in the network (along with backup paths), where each session specifies the source, destination and a guaranteed data rate to be supported. All the sessions in a traffic specification must be routed through the network such that no constraints are violated, and they must be scheduled to provide the guaranteed bandwidth. In addition to mission-critical traffic, we add competing background traffic to essentially saturate the IP network.

The following multipath routing problem formulation is based upon [33]. Let F be the set of all traffic flows in a traffic specification, denoted as (source,destination) pairs. Each flow f has a stationary unidirectional traffic rate r_f from the source to the destination. Let P_f be set of all directed paths from the source node to the destination, available to carry the traffic required by flow f. Let p denote an individual path within the set P_f. Let x_p be the traffic rate in bits/second assigned along a path p. Let $\mathbf{x}$ be the vector of all path flow rates $\{x_p | f \in F, p \in P_f\}$. For an admissible routing, the rate vector must satisfy the two constraints: (a) for every flow f, the sum of the traffic rates over all paths in P_f must equal the specified traffic rate for the flow f, and (b) the traffic rate along any path must equal or exceed 0 (ie negative traffic rates are not allowed).

In our backbone network, assume every unidirectional edge (i, j) has a constraint on the sum of traffic it can carry, denoted $C(i, j)$. The summation all traffic leaving node i over edge (i, j) over all paths and flows, denoted $\lambda_{i,j}$, must be $\leq C(i, j)$, ie

$$\lambda(i,j) = \sum_{f \in F, p \in P_f, (i,j) \in p} x_p \leq C(i,j)$$

One classical QoS routing problem formulation is a *Mixed Integer Non-Linear Delay-Constrained* optimization problem. Assume a traffic flow cannot be split over multiple paths, ie the traffic demand of one flow is assigned to exactly 1 path in P_f. Let each flow f have a vector of binary decision variables B_f. Let $P_f(j)$ denote the j-th path in P_f, and let $B_f(j)$ denote the j-th decision variable. The decision variable is asserted if the flow selects the corresponding path. A QoS optimization problem which minimizes the sum of delays in the network can be stated as follows:

$$\text{minimize} \sum_{(i,j) \in E} D(i,j)$$

subject to

$$\sum_{p \in P_f} x_p = r_f \qquad \forall f \in F \tag{7}$$

$$x_p \geq 0 \qquad \forall p \in P_f, f \in F \tag{8}$$

$$\lambda_{i,j} \leq C_{i,j} \qquad \forall i,j \in V \tag{9}$$

$$\sum_{j=1}^{|P_f|} B_f(j) \cdot x_p(j) = r_f \qquad \forall f \in F \tag{10}$$

$$\sum_{j=1}^{|P_f|} B_f(j) \leq 1 \qquad \forall f \in F \tag{11}$$

Constraints 7 and 10 assert that every flow achieves it guaranteed rate. Constraint 7 is implied within constraint 10 and is not necessary but adds clarity. Constraint 11 asserts that the data associated with one traffic flow traverses at most 1 path. The determination of an optimal routing given nonsplittable traffic flows is computationally difficult. In the above optimization problem, there are potentially exponentially many paths to be considered for each flow to be routed (in set P_f), and the problem of selecting one optimal path for each traffic flow is NP-Hard in the general case [34].

The use of the low-jitter RFSMD scheduling algorithm results in 2 significant simplifications for the above network routing problem: (1) Theorems 2 and 3 in section 3 guarantee that the normalized delay and jitter along any end-to-end path through any network are near-minimal (when the conditions are met). (2) Given the near-minimal delays along every path, the use of a nonlinear delay objective function to meet QoS guarantees in the first QoS optimization problem can be eliminated. *Given the bounded normalized delay and jitter guarantees provided by the RFSMD algorithm, the sole criteria for finding a routing which achieves near-minimal end-to-end normalized delay and jitter and near-perfect QoS can be reduced to that of finding any admissible routing*. By introducing multiple paths and allowing a traffic flow to be split over multiple paths, the routing problem is simplified considerably such that routings with near-perfect QoS can be easily found. (A similar observation was made for routing in wireless mesh networks in [35].)

The above simplifications lead to a second simplified multipath routing problem formulation. Given a set of paths P_f available for each traffic flow to be provisioned, such that the traffic rate can be split arbitrarily over all the paths in P_f, a constrained multipath optimization problem which minimizes the unrouted traffic in the backbone network can be stated as follows

$$\text{minimize} \sum_{f \in F} r_f - \sum_{f \in F} \sum_{p \in P_f} x_p$$

subject to

$$\sum_{p \in P_f} x_p = r_f \qquad \forall f \in F$$

$$x_p \geq 0 \qquad \forall p \in P_f, f \in F$$

$$\lambda_i \leq C_i \qquad \forall i \in V$$

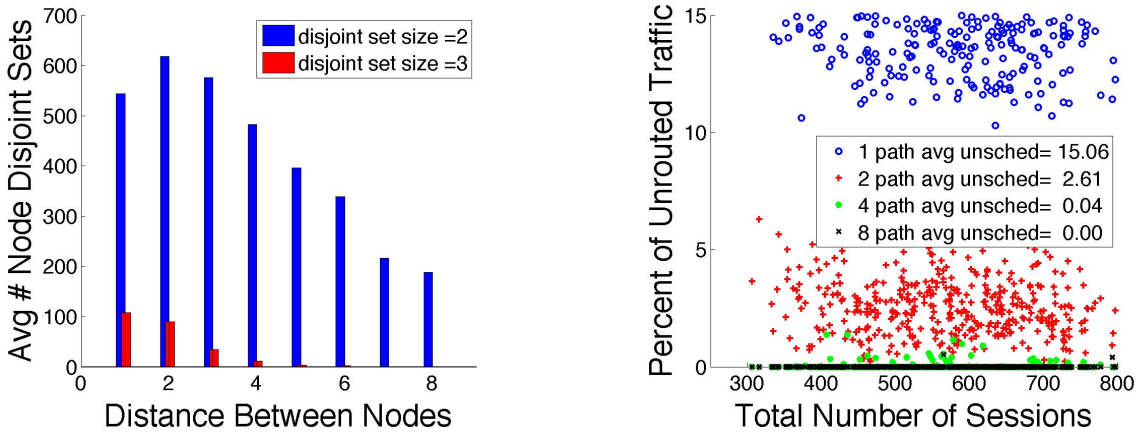
Any admissible multipath routing is guaranteed to achieve near-minimal normalized delay and jitter with essentially-perfect QoS for every flow, as stated by Theorems 1-4 in section 3. Once any admissible routing for the traffic specification has been found, the traffic rate matrix $R(j)$ for each IQ switch j in the network is updated by an RSVP of Diffserv resource reservation protocol. The matrices can be computed as shown in Eq. (12), where $M^{v(i)}$ denotes the traffic rate matrix for router $v(i)$, where $EI_{v(i)}$ is the input port of router $v(i)$ reserved for new traffic injected into the network, where $EO_{v(i)}$ is the output port of router $v(i)$ reserved for traffic which leaves the network, where $I_{v(i),v(j)}$ is the input port of router $v(j)$ connected to router $v(i)$, and where $O_{v(i),v(j)}$ is the output port of router $v(i)$ connected to router $v(j)$. Once the traffic rate matrices $M^{v(i)}$ for all routers $v(i)$ in the network are specified, each matrix can be decomposed and scheduled by at each router using the RFSMD algorithm, and the resulting frame transmission schedule yields the conflict-free permutations to be realized by each router for F time-slots within a scheduling frame:

$$\begin{aligned}
&for\ f \in F \\
&\quad for\ p = (v_1, v_2, ..., v_L) \in P_f \\
&\qquad M^{v_1}(EI_{v_1}, O_{v_1,v_2}) = M^{v_1}(EI_{v_1}, O_{v_1,v_2}) + r_f \\
&\qquad M^{v_L}(I_{v_{L-1},v_L}, EO_{v_L}) = M^{v_L}(I_{v_{L-1},v_L}, EO_{v_L}) + r_f \\
&\qquad for\ i = 2, ..., L-1 \\
&\qquad\quad M^{v_i}(I_{v_{i-1},v_i}, O_{v_i,v_{i+1}}) = \\
&\qquad\qquad M^{v_i}(I_{v_{i-1},v_i}, O_{v_i,v_{i+1}}) + r_f
\end{aligned} \tag{12}$$

To solve the second multipath optimization problem, a set of K paths P_f must be specified for each traffic flow to be routed. In general there may be exponentially many paths between a (source,destination) pair or (s,d) pair in a large network. Fig. 5a illustrates the average number of node-disjoint paths between node pairs of various distances in the COST266 network. Node pairs separated by ≤ 5 hops have > 400 node-disjoint path pairs. For mission-critical traffic, we select K pairs of node-disjoint paths, for K working paths and K disjoint backup paths, which are routed separately. For background traffic flows, we select the K paths pseudorandomly for inclusion into sets P_f. Let X denote the shortest distance (in hops) between the (s,d) pair for flow f. Consider the set of paths Z with distances $X+3$. Select any set of K paths randomly from Z, with a bias towards shorter paths. In section 6, each traffic specification is routed with 4 choices for K, ie with $K = (1, 2, 4, 8)$ paths per traffic flow. Once the K paths are selected, linear programming (LP) is be used to assign the traffic rates along each path. The results of this multipath routing algorithm are presented in section 6.

VI. Example: Provisioning Telerobotic Control Traffic

To test the proposed routing and scheduling algorithms, exhaustive simulations were performed over the COST266 European optical backbone network. Four hundred admissible traffic specifications where generated. Each traffic specification consists of mission-critical telerobotic control traffic between every pair of cities, along with competing background



(a) (b)

Fig. 5. (a) Node disjoint path sets in COST266 topology. (b) Multipath routing results.

traffic. Each of the traffic specifications saturates the network, so that every link operates at 100% load given non-splittable flows. Each traffic specification represents an extremal point in the Capacity Region of the network where every link is saturated given non-splittable flows. These traffic specifications will generally be the most difficult to provision while guaranteeing near-perfect QoS for all traffic flows.

An iterative computer program was created to find these 400 admissible traffic specifications. For each traffic specification, first the telerobotic sessions would be routed between all pairs of cities, with bandwidth provisioned for restoration capability along the backup paths. These routings were fixed and recorded. Secondly, background traffic flows between randomly-selected pairs of nodes would be iteratively added and routed until no more flows could be added, to utilize the remaining bandwidth. Background traffic is allowed to use the SBPPs of mission-critical traffic, while the primary paths are operational. Therefore, the bandwidth of the backup paths becomes available for background traffic to use, on a lower priority basis. It becomes increasingly difficult to add a new background traffic flow between a pair of nodes along a multi-hop path as the network load increases, due to the difficulting in finding an end-to-end path with the available capacity. In the last iteration, one-hop traffic flows were added to saturate every link, resulting in an average link load of 100%. The current traffic specification, consisting of all the (s, d)-pairs and rates, would be recorded. However, the routing information for the background traffic was not recorded.

The background traffic in all 400 traffic specifications was then routed using the multipath optimization problem in section 5. Fig 5b. illustrates the results for the routing of all traffic specifications. The background traffic in each of the 400 traffic specifications was routed with 4 different options for K, ie with $K = (1, 2, 4, 8)$ paths per traffic flow. There are 1,600 points in Fig. 5b, corresponding to the 400 traffic specifications and the 4 routings per specification. When K=1, an average of 15.06% of the traffic remained unrouted after solving the second optimization problem with linear programming. When K=2, an average of 2.61% of the traffic remained unrouted. When K=4, an average of 0.04% of the traffic remained unrouted, and with K=8 an average of 0% of the traffic remained unrouted. Fig. 5b illustrates that multipath routing with linear programming is an excellent choice for routing, where essentially 100% of all traffic flows can be routed for K $\geq$ 8, even in essentially saturated backbone networks. (Similar observations were reported for multipath routing in wireless mesh networks in [35].)

Once each traffic specification was routed, the traffic rate matrices for all switches in the network were computed and scheduled using the RFSMD algorithm. The scheduling results for all traffic specifications where virtually identical, so the details of one specific traffic specification are described in detail next. The selected traffic specification consists of 1,628 distinct traffic flows to be routed through the COST266 network. Each node on average was the source (or destination) for about 53 competing traffic flows. The simulations were conducted on a custom-written simulator with over 20,000 lines of code, which allows us to retrieve conditional probability distributions for any parameters of interest. (The simulator was developed by four graduate students funded as part-time research associates over a period of 2 years.)

Fig. 6 illustrates the normalized service lead/lag curves for all 1,628 traffic flows over all routers in the network. There are 5,275 individual red curves shown in Fig. 6, since each of the 1,628 flows traverses 3.24 routers on average, and each flow generates a normalized service lead/lag curve at each router. The ideal normalized service for each flow is represented by the main diagonal. The dashed lines above and below the main diagonal illustrate a normalized service lead or service lag of 4 cells. According to Fig. 6, the normalized service received by all 1,628 flows in all nodes is *essentially perfect*, as each service curve deviates only slightly from the ideal curve. Every cell arrives at a *near-perfect arrival time*, and every cell departs at a *near-perfect departure time*.

Fig. 7a illustrates the end-to-end (E2E) normalized delay observed for all 1,628 traffic flows. The minimum E2E delay is approx. 0 IIDT, while the maximum E2E delay is 22 IIDT. Fig. 7b plots the deviation in the end-to-end normalized delay for every flow from its mean value, ie the end-to-end delay jitter per traffic flow. There are 1,628 curves in Fig. 7b, one

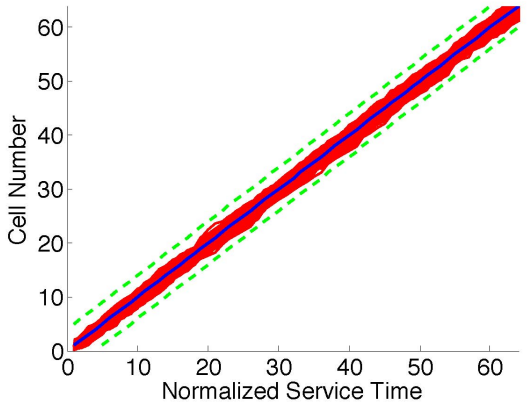


Fig. 6. Normalized Service Lead/Lag, all 1,628 flows.

for each end-to-end flow. The normalized delay jitter for every flow is less than 4 IIDTs, which is small and bounded as predicted by Theorem 3. Fig. 7b again illustrates that every flow receives *essentially-perfect QoS*, ie cells arrive with *near-perfect arrival times*, cells depart with *near-perfect departure times*, and over all 1,628 end-to-end flows the delay jitter is limited to approx. $\pm$ 4 cell times, consistent with theorem 3.

Fig. 8a illustrates the normalized *Inter-Arrival Time (IAT) PDF* for all cells in all flows arriving at all routers, another indication of the delay jitter. Fig. 8a illustrates that the average normalized IAT in every flow is 1 IIDT. In other words, *the expected IAT between cells equals one perfect IIDT*. Since the arriving stream at one router equals the departing stream from a previous router, the Inter-Departure Time (IDT) PDF for all flows must equal the IAT PDF. Therefore, *the expected IDT between cells equals one perfect IIDT*. In Fig. 8a, the IATs vary from a low of about 0 IIDT to a maximum of 4.5 IIDTs, indicating a relatively small network-introduced jitter for every flow in the network. Recall that the network is essentially fully-saturated. Even at very high loads, all provisioned traffic flows are delivered with very low network-introduced jitter.

Fig. 8b illustrates the PDF for the number of queued cells per flow per router. Fig. 8b is based on observations from all 1,628 flows over all routers. There are 5,275 curves in Fig. 8b, since each of the 1,628 flows traverses 3.24 routers on average. The average number of queued cells per flow per router is 1.75, which is consistent with Theorem 1, which establishes a bound of 16 queued cells per flow per router. According to Fig. 8b, in this traffic specification the maximum number of queues cells per flow per router is 6, which is well below the theoretical maximum of 16 cells established in Theorem 1. These results are several orders of magnitude better than those from existing scheduling algorithms, as the discussion in section 7 will establish.

To dimension the ASSQ and the ASPQ, a discrete-time batch $D^x/D^y/1$ queueing model was utilized. Batches arrive/depart at deterministic times (1/30 of a second) with batch size distributions x/y respectively. The arriving batch-size distribution x is determined from the video I-frame traffic statistics. The service batch-size distribution y is determined from the provisioned guaranteed-rate. Fig. 9 illustrates the average delay of the ASSQ+ASPQ, for a telerobotic session with 2 video streams with excess bandwidth = 40Mbps. The average delay is approx. 100 millsec (not including the fiber delays). Fig. 10 illustrates the worst-case end-to-end ASSQ+ASPQ delay, versus the excess bandwidth. An excess bandwidth of 85% or higher for each telerobotic session with 2 video streams, results in ASSQ+ASPQ queueing delays 200 millisec. Recall each destination router reconstructs the two original bursty video streams and makes zero-jitter video streams available at the surgeon's console. Fig. 10 illustrates that all network-introduced delay jitter from the Internet backbone has been removed from consideration, and the video streams are delivered well within the 250 millisec deadline.

Fig. 11 illustrates that the mission-critical traffic has exceptionally high availability. The failure probability (1- availability) is plotted for all (s, d) pairs with a given distance in hops on one curve, assuming only 2 node-disjoint paths are utilized, and a link/node failure prob. of 10^{-6}. (Many (s, d) pairs have far more disjoint paths.) The provisioned working path failure probability is typically less than 10^{-6}.

VII. Comparison

In this section, a comparison between the proposed techniques to achieve end-to-end QoS and the current state-of-the-art is presented. The proposed techniques are a combination including low-jitter scheduling, traffic shaping at the ingress and egress points, and multipath routing. In this section we focus on the scheduling problem, and revisit some of the state-of-the-art scheduling methods summarized in section 2.2.

The scheduling problem in an IP router consists of finding a sequence of matchings in a bipartite graph which provide guarantees on the delay, jitter and service lag. There are two main approaches to the scheduling problem; (i) *mathematical methods* which provide rigorous guarantees, and (ii) *heuristic methods* which provide probabilistic guarantees.

As stated in section 2.2, it is well established in theory that scheduling based upon the solution of a *Maximum Weight Matching (MWM)* problem in each time-slot can achieve 100% throughput and bounded queue sizes, delay and jitter. Unfortunately, the bounds can be very large, ie thousands of packets per router. Furthermore, the MWM algorithm has a complexity of $O(N^3)$ per time-slot. As line-rates increase from 40 Gbps to 160 Gbps, IP routers are required to compute schedules at the rates of about 76 and 300 Million matchings per second respectively. The MWM algorithm is intractable at these rates. Mathematical methods to compute matchings using stochastic matrix decompositions have also been well studied, as summarized in section 2.2. These include the BVN decomposition [29], ILJD and GLJD decompositions [27], the MIT decomposition [30] and the UCR decomposition [31]. However, these mathematical algorithms are also intractable at these rates. Furthermore, all of these mathematical algorithms except for the BVN method require a speedup of greater than 1, which limits their practical applicability.

Currently, many IP routers use heuristic scheduling algorithms such as Parallel Iterative Matching (PIM) and iSLIP [26]. These algorithms can be implemented in parallel hardware and can be pipelined to yield high computation rates. For example, the iSLIP scheduling algorithm can use $O(N)$

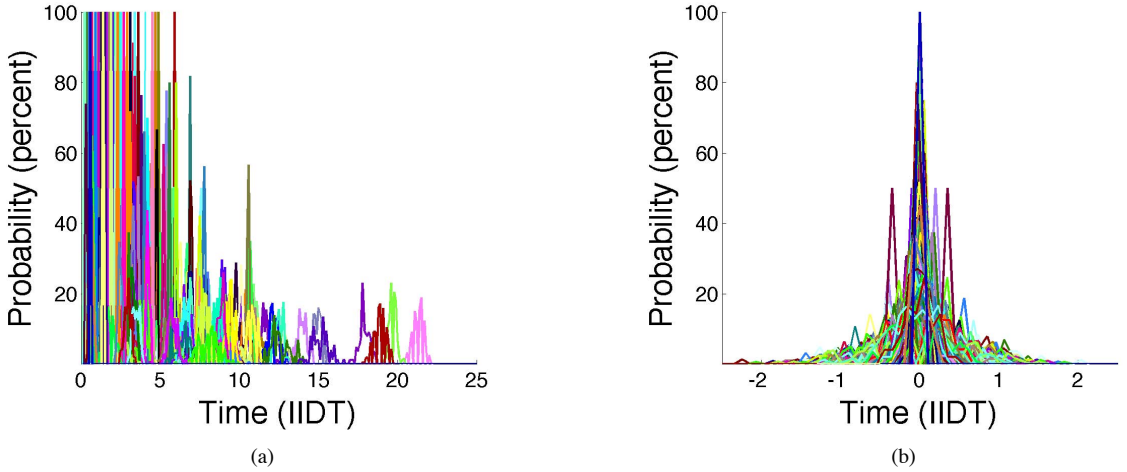


Fig. 7. (a) End-to-End delay distribution. (b) E2E delay deviation from mean.

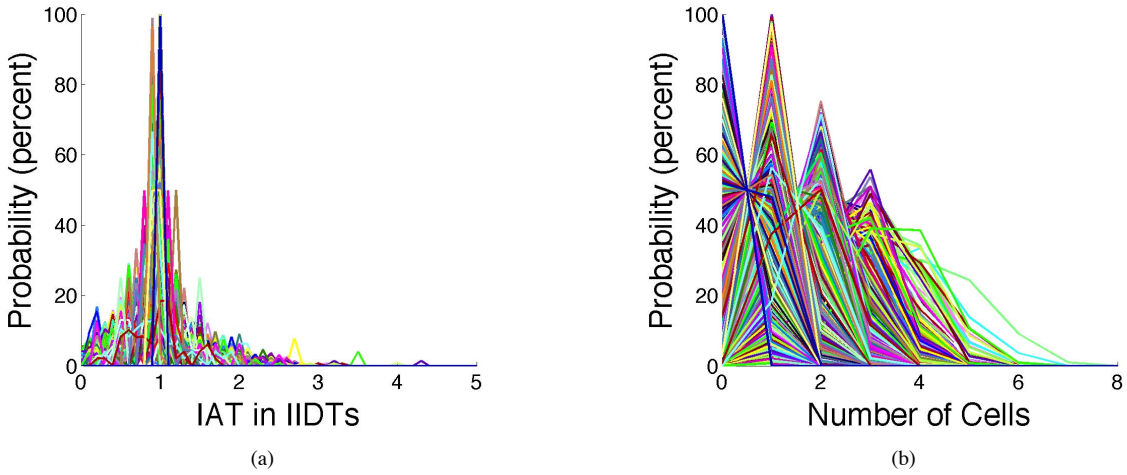


Fig. 8. (a) Inter-Arrival time PDF. (b) PDF, Queued cells per flow per router.

arbiters and $O(N^2)$ wires to realize one maximal matching per time-slot in $O(logN)$ iterations. Unfortunately, heuristic algorithms cannot provide rigorous and small guarantees on the queue sizes, delay or jitter. Therefore, existing IP routers typically saturate at high loads, where the number of queued packets, the delay and jitter can become unbounded. As a result, existing IP routers currently require large packet buffers to avoid queue starvation and to maintain high throughput. It is estimated that router manufacturers spend hundreds of millions of dollars a year just on memory for buffers [36].

The *ACM Computer Communications Review* has recently hosted a debate on buffer sizing in IP routers through a series of articles [37]. A well-established design rule called the *'classical buffer rule'* states that each link in each IP router requires a buffer of $B = O(C \cdot T)$ bits, where C is the link capacity and T is the round-trip time of the flows traversing the link [38]. According to data in [38], a 40 Gbps link handling TCP flows with a round-trip time of 250 millisec requires a buffer size B about five million IP packets. Each IP packet may contain up to 1,500 bytes or equivalently 24 cells, and a buffer may require up to 5 Gigabytes of memory per link. A *'small buffer rule'* was proposed in [38], where $B = O(CT/N^{1/2})$, and where N is the number of long-lived TCP flows traversing the router. With the same parameters reported above, the buffer size B is reduced to about fifty thousand IP packets [38]. More recently, [41] proposed a *'tiny buffer rule'* where $B = O(logW)$, where W is the maximum TCP congestion window size. With the same parameters, it was postulated that average buffer sizes of between 20-50 IP packets or equivalently up to about one thousand cells may suffice if (a) the jitter of incoming traffic at the source node is sufficiently small, (b) the IP routers introduce a sufficiently small jitter, and (c) 10-15% of the throughput is sacrificed. However, [42][43] have argued that small buffers may cause significant losses, instability or performance degradation at the application layer.

A recent 2009 journal article has addressed the buffer-sizing issue [37] and stated: *the basic question - how much buffering do we need at a given router interface? - has received hugely different answers in the last 15 to 20 years, such as a few dozens of packets, a bandwidth-delay product, or a multiple of the number of large TCP flows in that link. It cannot be*

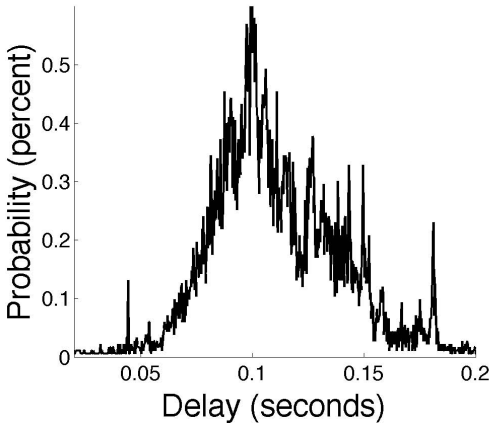


Fig. 9. Delay PDF, ASSQ+ASPQ

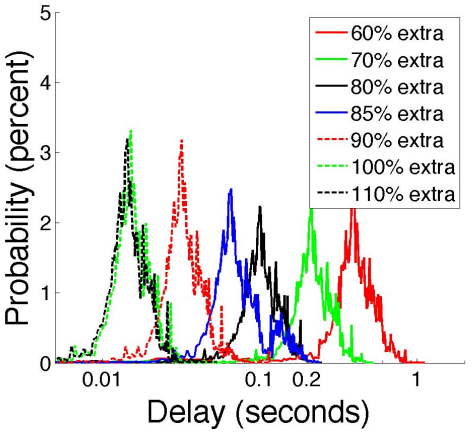


Fig. 10. ASSQ+ASPQ Delay vs. excess bandwidth, for 1 telerobotic session

that all these answers are right. It is clear that we are still missing a crucial piece of understanding despite the apparent simplicity of the previous question. The paper presents further insights on buffer requirements, by relating the buffer size to the ratio of output-to-input bandwidths in routers.

The previous discussions illustrate that the inter-related issues of buffer sizing, packet loss rates, delay, jitter and QoS are the subject of considerable interest and debate. In summary, there are significant problems when using existing heuristic schedulers and high-jitter TCP flow control protocols in IP networks. The prospect of realizing mission-critical telerobotic control systems such a life-critical telerobotic surgery over the current Internet while meeting rigorous QoS guarantees seems challenging. Indeed, the current problems of the Internet have motivated the NSF GENI program which is open to new ideas to address these challenges.

A necessary condition to guarantee bounded queue sizes in IP networks, for every flow regardless of its mean rate, is the concept of the *normalized service lead/lag* described in section 3 and first presented in [17][18]. Theorems 1-4 in section 3 require that the incoming traffic and the service schedule at each queue achieve a bounded normalized service lead/lag for every flow, regardless of its rate. All of the mathematical scheduling algorithms described in section 2.2 have reported service lead/lag bounds of at best $O(N^2)$ time-slots under the constraint of unity speedup, and will exhibit normalized service lead/lag bounds of at best $O(N)$ under the constraint of unity speedup. Therefore, theorems 1-4 indicate that the buffer sizes in any router using any of these prior mathematical scheduling algorithms will be at least $O(N)$ and may be unbounded. In contrast, the RFSMD algorithm has a normalized service lead/lag bound of K for every flow, where K is a small integer provided that the node degree N and scheduling frame length F are bounded [17]. To date, the RFSMD algorithm is the only known algorithm to achieve 100% throughput for all admissible traffic matrices with a bounded normalized service lead/lag under the constraint of unity speedup.

The RFSMD algorithm examined in this paper addresses the buffer-sizing issue by presenting a tractable mathematical scheduling algorithm which provides rigorous end-to-end

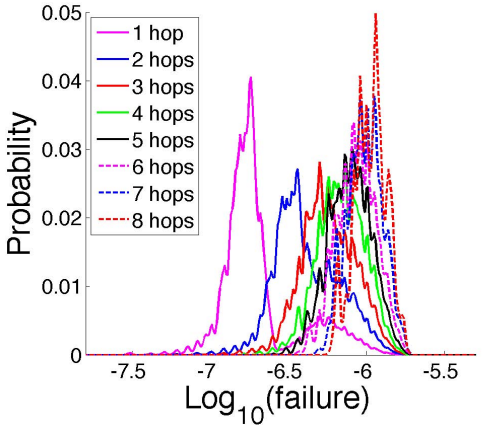


Fig. 11. Failure probability (1-Availability) for all combinations of 2 node-disjoint paths.

QoS guarantees, as stated in theorems 1-4. Our exhaustive simulations in section 6 indicate that network queueing delays are several orders of magnitude smaller than those in current IP routers, consistent with the theory. Each IP router needs to buffer on average about 2 cells (128 bytes) per flow per router to guarantee 100% throughput and essentially-perfect end-to-end QoS. In comparison, existing IP routers using the combination of heuristic schedulers, TCP flow control and the bandwidth-delay buffer-sizing rule require buffers of about 5 million IP packets per link at 40 Gbps to achieve high throughput without QoS guarantees [38]. The RFSMD algorithm and theorems 1-4 allow for reductions in buffer sizes by several orders of magnitude compared to the existing technology, while simultaneously meeting QoS constraints.

The RFSMD scheduling algorithm has a serial complexity of $O(NFlog(NF))$ to compute F matchings, equivalent to $O(Nlog(NF))$ per matching. This complexity figure is considerably better than those reported for all previous mathematical scheduling methods, and is better than those reported for many heuristic schedulers. Given an 8x8 IP router, a link rate of 40 Gbps and a scheduling frame size of $F = 1K$ time-slots, the schedules are required at the rate of 76.3 KHz, equivalent to about 76 million matchings per second. The RFSMD

scheduling algorithm has a similar structure and comparable execution time to the well-known *Fast-Fourier Transform* algorithm. Intel projects the performance of its multi-core processors to be 100,000 MIPS by 2010 (*www.intel.com*). With this performance, we estimate that a 2010 multi-core CPU chip can compute new schedules in software at rates up to 100 KHz for an 8x8 router with $F = 1K$, equivalent to about 100 million matchings per second. A multiple chip implementation or a hardware-based FPGA implementation should be able to compute schedules considerably faster. Furthermore, when using a resource reservation protocol such as RSVP or Diffserv, the traffic rate matrix in each router will change much more slowly, corresponding to the rate at which new traffic flows are added or removed from a router. This point was observed by researchers at Bell Labs [27][28] as well. In this case, we estimate that schedules can be recomputed in each router only about 100 to 1000 times/sec, and the RFSMD algorithm should easily be able to compute router schedules in software. The computational complexity of the multipath routing algorithm in section 4 is effectively polynomial. The selection of K paths per traffic flow involves solving a shortest path algorithm such as Dykstra's algorithm. The solution of the linear program (LP) in section 5 is very quick, as current LP solvers can easily handle tens of thousands of variables.

In summary, we believe that our theory and techniques address several important state-of-the-art questions debated in the literature, and address problems which current technologies cannot adequately solve.

VIII. Conclusion

Algorithms to provision mission-critical telerobotic control traffic in a backbone IP/MPLS network with essentially-perfect restoration capability and QoS have been presented. Mission-critical traffic is routed either by using SBPP schemes or by using the theory of p-cycles. A multipath routing algorithm is proposed to route competing background traffic flows over multiple paths, thereby exploiting path diversity. The multipath routing algorithms is efficiently solved using linear programming. The routings can achieve essentially 95-100% of the network capacity region for path multiplicities of 8 or higher. The traffic rate matrix in each router is updated by a DiffServ or RSVP protocol and is scheduled using the low-jitter RFSMD stochastic matrix decomposition algorithm. Exhaustive simulations were performed to test the RFSMD theory. The proposed algorithms achieve near-minimal normalized delay and jitter and essentially-perfect QoS for all traffic flows routed through the IP/MPLS network, including all mission-critical and all competing backgound traffic as predicted by theory. The destination routers deliver bursty zero-jitter telerobotic video streams to their surgeon's consoles. Every video source application must have an Application-Specific Token-Bucket traffic Shaper Queue at the ingress point to the network, and an Application-Specific Playback Queue at the destination router. Practical designs for these systems were presented. In summary, schemes which deliver mission-critical telerobotic control traffic in a backbone IP/MPLS network with 100% restoration capability and essentially-perfect QoS are achievable

IX. Acknowledgements

The careful reviews of the Guest Editors are appreciated. Funding to test these theories on a 160-node supercomputer at McMaster University is acknowledged from the Ontario Centers of Excellence (OCE) program. Funding to assist in the commercialization of these technologies is acknowledged from the Ontario Centers of Excellence (OCE) program.

References

[1] M. Baard, 'NSF Preps New Improved Internet', Wired, Aug 2005.
[2] BBN Technologies - GENI Project Office, "Global Environment for Network Innovations - GENI System Overview", Dec. 2007 ($http://genie.net/$).
[3] L. Hardesty, "Internet Gridlock: Video is clogging the Internet. How we choose to unclog it will have far-reaching implications", MIT Technology Review, July/August 2008
[4] A. Pirisi, "Telerobotics Brings Surgical Skills to Remote Communities", The Lancet, Vol. 361 May 2003.
[5] Cisco Systems New Release, Bell Canada Utilizes Cisco Systems Technology to Help Deliver Surgical Grade Network to Power Historic Telerobotics Assisted Surgery, 2003 [Online]. Available $http://newsroom.cisco.com/dlls/prod\_030403.html$
[6] S. Wexner, R Bergamaschi, A. Lacy, J. Udo, H. Brolmann, R. Kennedy, H. John "The Current status of robotic pelvic surgery: results of a multinational interdisciplinary consensus conference", Sugical Endoscopy, Vol 23, pp. 438-443, 2009.
[7] P. Kazanzides, G. Fichtinger, G. Hager, A. Okamura, L WhitComb, R. Taylor "Surgical and Interventional Robotics: Core Concepts, Technology and Design", IEEE Robot. Automat. Mag., June 2008.
[8] B Harnett, C Doarn, J. Rsen, B. Hannaford, T. Broderick "Evaluation of Unmanned Airborne Vehicles and Mobile Robotic Telesurgery in an Extreme Environment", Telemedicine and e-Health, July/August 2008.
[9] Computermotion "Operation Lindbergh: A World First in TeleSurgery", France Telecom, Press Conference, Sept 19, 2001.
[10] S. Naegel-Jackson, P. Holleczek, T. Rabenstein, J. Maiss, E. Hahn, M. Sackmann, Influence of Compression and Network Impairments on the Picture Quality of Video Transmissions in Tele- Medicine Proc. 35th Hawaii International Conference on System Sciences, 2002.
[11] Rabenstein t., Maiss J., Naegele-Jackson S., Liebl K., Hengstenberg T., Rapespiel-Troger M., Holleczek P., Hahn E., Sackmann M. Tele-Endoscopy: Influence of Data Compression, Bandwidth and Simulated Impairments on the Usability of Real-Time Digital Video Endoscopy Transmissions for Medical Diagnoses Endoscopy, pp 703-710, vol. 34, 2002.
[12] M. Lum, D. Friedman, J. Rosen, G. Sankaranarayanan, H. King, K. Fodero, R. Leuschke, M. Sinanan, B. Hannaford, 'The RAVEN - Design and Validation of a Telesurgery System", International Journal of Robotics Research, January 2009.
[13] S. Butner, M. Ghodoussi Transforming a Surgical Robot for Human Telesurgery, IEEE Trans. Robot. Autom., Vol. 19, No. 5, Oct 2003.
[14] K. Cleary, Medical Robotics and the Operating Room of the Future, Proc. 2005 IEEE Engineering in Medicine and Biology 27th Annual conference, Sept 1-4, 2005.
[15] A. Lombardo, G. Schembra, and G. Morabito, 'Traffic Specifications for the Transmission of Stored MPEG Video on the Internet', IEEE Trans. Multimedia, VOL. 3, NO. 1, MARCH 2001.
[16] T.H. Szymanski, "QoS Switch Scheduling using Recursive Fair Stochastic Matrix Decomposition", IEEE HPSR, 2006, pp. 417-424.
[17] T.H. Szymanski, "A Low-Jitter Guaranteed-Rate Scheduling Algorithm for Packet-Switched IP Routers", IEEE Trans. Commun., Vol. 57, No. 11, Nov. 2009.
[18] T.H. Szymanski, "Bounds on the End-to-End Delay and Jitter in Input-Buffered and Internally Buffered IP Networks", IEEE Sarnoff Symposium, Princeton, NJ, March/April, 2009.
[19] T.H. Szymanski and D. Gilbert, "Low-Jitter Guaranteed-Rate Communications for Cluster Computing Systems", (Invited) Int. Journal of Computer Networks and Dist. Systems, pp. 140-160, 2008.
[20] T.H. Szymanski and D. Gilbert, "Internet Multicasting of IPTV with Essentially-Zero Delay Jitter", IEEE Trans. Broadcast., Vol. 55, No. 1, March 2009, pp. 20-30.
[21] T.H. Szymanski, "Method and Apparatus to Schedule Packets through a Crossbar Switch with Delay Guarantees", US Patent Application, 2007.
[22] G. Shen, W. Grover "Extending the p-Cycle concept to Path Segment Protection for Span and Node Failure Recovery", IEEE J. Sel. Areas Commun., Vol. 21, No. 8, October 2003.

[23] J. Doucette, W. Grover "Physical-Layer p-cycles Adapted for Router-Level Node Protection: A Multi-Layer Design and Operation Strategy" IEEE J. Sel. Areas Commun., Vol. 25, No. 5, 2007.

[24] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks", IEEE Trans. Autom. Control, Vol. 27, Dec 1992, pp. 1936-1948.

[25] V.Tabatabaee and L. Tassiulas, "MNCN: A Critical Node Matching Approach to Scheduling for Input-Buffered Switches with No Speedup", IEEE Trans. Netw., Vol. 17, Feb 2009, pp. 294- 304.

[26] N. McKeown, "The iSLIP Scheduling Algorithm for Input Queued Switches", IEEE Trans. Netw., Vol. 7, No. 2, April 1999, pp. 188-201

[27] I. Keslassy, M. Kodialam, T.V. Lakshamn and D. Stilliadis, "On Guaranteed Smooth Scheduling for Input-Queued Switches", IEEE/ACM Trans. Netw., Vol. 13, No. 6, Dec. 2005

[28] M.S. Kodialam, T.V. Lakshman and D. Stilladis, "Scheduling of Guaranteed-bandwidth low-jitter traffic in input-buffered switches", US Patent Application No. 20030227901

[29] W.J. Chen, C-S. Chang. and H-Y. Huang, "Birkhoff-von Neumann Input Buffered Crossbar Switches", IEEE Trans. Commun., Vol. 49, No. 7, July 2001, pp. 1145-1147.

[30] C.E Koksal, R.G. Gallager, C.E. Rohrs, "Rate Quantization and Service Quality over Single Crossbar Switches", IEEE Infocom 2004

[31] S.R. Mohanty and L.N. Bhuyan, "Guaranteed Smooth Switch Scheduling with Low Complexity", IEEE Globecom, 2005, pp. 626- 630

[32] P. Seelingm, M. Reisslein, and B. Kulapa, Network performance evaluation using frame size and quality traces of single layer and two layer video: A tutorial, IEEE Comm. Surveys, vol. 6, no. 3, pp. 5878, 3rd Q 2004.

[33] D. Bertsekas and R. Gallager, 'Data Networks', Prentice Hall, 1992.

[34] R.M. Karp "On the Computational Complexity of Combinatorial Problems", Networks, Vol. 5, pp45-68, 1975.

[35] T.H. Szymanski and D. Gilbert, Video Distribution over Multihop Wireless Mesh Networks with Near-Minimal Delay and Jitter, submitted.

[36] S. Iyer, RR. Kompella, N. Mckeown, "Designing Packet Buffers for Router Linecards", IEEE Trans. Netw., Vol. 16, No. 3, June 2008, pp. 705-717

[37] R.S. Prasas, C. Dovrolis, M. Thottan, "Router Buffer Sizing for TCP Traffic and the Role of the Output/Input Capacity Ratio", IEEE Trans. Netw., To Appear 2009.

[38] Y. Ganjali, N, McKeown, "Update on Buffer Sizing in Internet Routers", ACM Sigcomm Comp. Comm. Rev., vol. 36, no. 5, pp. 67- 70, Oct 2006.

[39] G. Appenzeller, I. Keslassy and N. McKeown, "Sizing router buffers", ACM Sigcomm Comp. Comm. Rev., USA, pp. 281-292, 2004.

[40] G. Raina and D. Wishick, "Buffer sizes for large multiplexers: TCP queueing theory and instability analysis", EuroNGI, Rome, Italy, April 2005.

[41] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, T. Roughgarden, "Routers with very small buffers", IEEE Infocom, Spain, April 2006

[42] A. Dhamdhere and C. Dovrolis, "Open Issues in Router Buffer Sizing", ACM/SIGCOMM Comp. Comm. Rev., vol. 36, no. 1, pp. 87-92, Jan 2006.

[43] G. Vu-Brugier, R.S. Stanojevic, D.J. Leith, and R.N. Shorten, "A Critique of recently proposed buffer sizing strategies", ACM/SIGCOMM Comp. Comm. Rev., vol. 37, no. 1, pp. 43-47, May 2007.

Ted H. Szymanski (M87) completed a BaSc. in Engineering Science and the MaSc. and PhD degrees in Electrical Engineering at the University of Toronto. He has held faculty positions at Columbia University in New York, where he was affiliated with the Center for Telecommunications Research (CTR), and McGill University in Montreal, where he was affiliated with the Canadian Institute for Telecommunications Research (CITR). From 1993 to 2003, he was a principle architect in a national research program on Photonic Systems funded by the Canadian Networks of Centers of Excellence (NCE) program. The program brought together significant industrial and academic collaborators, including Nortel Networks, Newbridge Networks (now Alcatel), Lockheed-Martin/Sanders, Lucent Technologies and McGill, McMaster, Toronto and Heriot-Watt Universities. The program demonstrated a free-space "intelligent optical backplane" exploiting emerging optoelectronic technologies with 1,024 micro-optic laser channels per square centimeter of bisection area, for which he holds two patents. Since 2001 he has served as the Red Wilson / Bell Canada Chair in Data Communications at McMaster University. His current interests include switching, scheduling, and network QoS for emerging telemedicine and telerobotic control systems, as well as optical and wireless networks. He has served on the program committees of several international conferences. During his sabbatic leaves he has been a visiting professor at FORTH Greece, the University of Toronto, and the University of Victoria. He has consulted for several companies and he has several patents issued or pending. He has also served as the Associate Chair (undergraduate) and the undergraduate student advisor in the ECE Department at McMaster University.

Dave Gilbert (M97) completed his Ph.D. in the Dept. of Electrical and Computer Engineering (ECE) at McMaster University in 2007, in the area of nuclear reactor modeling. He is currently a Post-Doctoral Fellow in the Department of ECE. His research interests include nuclear reactor modeling, software-based problem-solving environments, and network performance modeling.