Traffic Restoration Algorithms in Communication Networks

Vladica Stanisic and Michael Devetsikiotis
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695-7914

***Abstract*— E-commerce and mission-critical Internet services require a maximum of availability of the network and a minimum of network outage times. Due to the increasing demands to carry mission critical traffic, real-time traffic, and other high priority traffic over the Internet network survivability represents a requirement for the future networks. In case of failure a large number of connections would require a simultaneous retransmission of lost packets. This work proposes the use of prioritized resource allocation and preemption for reestablishing connections that have been disrupted by a failure. In this paper we propose a restoration priority scheme based on types of applications which can be mapped to the QoS architecture of wireless networks and DiffServ standards. We investigate the order in which the connections are being restored and the order of restoring the connections and routing within each application type in the presence of dynamic bandwidth allocation.**

I. Introduction and Motivation

E-commerce and mission-critical Internet services require a maximum of availability of the network and a minimum of network outage times. The current Internet has a built-in degree of survivability due to the connectionless IP protocol. Dynamic routing protocols react to faults by changing routes when routers learn about topology changes via routing information updates (e.g., link status advertisements). Loss of QoS has not been an issue because current Internet traffic is best-effort. The new connection-oriented, real-time interactive services that are already being offered on the Internet (or are currently emerging) have increased resilience requirements. Quality of Service assurance is becoming a necessity in the Internet of future. Traffic-engineering methods that allow the provisioning of network resilience are a clear requirement for current and future Internet networks.

Network survivability refers to the capability of a network to maintain service continuity in the presence of faults, by promptly recovering from network impairments and maintaining the required QoS for existing services after recovery. Due to the increasing demands to carry mission critical traffic, real-time traffic, and other high priority traffic over the Internet network survivability represents a requirement for the future networks [1].

Many methods have been proposed for QoS assurance in case of network failures or load fluctuations. Packet switching and TCP/IP as the technological foundations of the Internet, do not guarantee survivable communications in the event of an attack at a node or a link. In case of failure a large number of connections would require a simultaneous retransmission of lost packets. Retransmitted packets create a backlog at the traffic source that combined with the collective attempt to reestablish connections may cause undesirable transients and congestion in the network, which is the dominant factor on network performance immediately after a failure.

One way of improving network dependability is to prepare *redundant network resources* in order to cope with failures or load fluctuations. When failures occur, this pre-assigned spare capacity is reconfigured in distributed manner and used to restore the failed connections. Another approach toward improving network dependability involves the use of *reconfigurable networks* and *self-sizing networks*. Reconfigurable network is a network where the effective topology and capacities can be dynamically adapted to changes in traffic requirements or to changes in the physical network due to failures. Self-sizing network operation is a traffic engineering and operation concept developed for ATM networks which allows networks to be rapidly operated and re-dimensioned, based on measurement of traffic flow and demand. Reconfigurable networks and self-sizing network restore connections after failures or load fluctuations using centralized reallocation of resources from working connections to degraded connections, which are not suitable for restoration after failures. It is difficult to

achieve rapid restoration using a centralized approach, especially in a large networks. For those reasons, an algorithm for resource reallocation must be decentralized [2].

This work proposes the use of prioritized resource allocation and preemption for reestablishing connections that have been disrupted by a failure. No network can be economically designed for extreme overloads. Loads on public networks reach up to five times normal during an emergency causing that important traffic receives equally poor access to resources as low priority traffic. The better alternative for sudden load increases is to assign the priorities of various types of traffic, and divert the lowest-priority communications to other network providers that may have excess capacity, or to other types of networks [3]. In this paper we propose a restoration priority scheme based on types of applications which can be mapped to the QoS architecture of wireless networks and DiffServ standards [4], [5]. A failure typically results in a large number of disrupted connections of each application type, which must be restored simultaneously. In this paper we investigate the order in which the connections are being restored and the order of restoring the connections and routing within each application type in the presence of dynamic bandwidth allocation.

II. Utility-based Priority Assignment

Utility based resource allocation has recently received attention both in the wired Internet [6], [7], [8], [9] and in wireless networks [10], [11], [12]. Most of the previous work has investigated rate control algorithms based on the utilities of the users while being fair, in order to achieve the system optimal rates in the sense of maximizing aggregate utility. This paper differs from previous work in the respect, that we investigate how to relate priority levels to QoS requirements of the request through its utility function.

We focus on *four* types of applications which can be mapped to the QoS architecture of wireless networks as well as DiffServ standards. The first type of applications that we consider are called *hard real-time applications*, which need their data to arrive within a given delay bound. Examples are disaster recovery and emergency traffic or some important advanced applications, such as remote surgery or remote instrument control. The second type are *delay-adaptive applications* which are more tolerant of occasional delay bound violations and dropped packets. The third type are the *rate-adaptive applications* which can adjust their transmission rate in response to network congestion. The fourth type are *elastic applications*, which have more relaxed or lower quality of service requirements [13].

For each type of application, we define a specific utility function u which represents the "level of guarantee" provided to a user by the network or "insurance" in case of transient overloads or network faults. The utility function is used in order to relate to the connection's quality of service (QoS) requirements, such as reliability desired, bandwidth requirement, real-time delivery constraints, and desired blocking probability.

A request r_k can be defined as a flow of information from a source to a destination involving a certain amount of "bandwidth". Bandwidth can be represented by the peak rate of the request R_k and the minimum rate of the connection m_k, duration, a priority level p_k, and a utility function u_k that is based on the bandwidth received and type of application. The allocated bandwidth, $\hat{c}_k$, for the request r_k can then be calculated as the *effective bandwidth* according to the model introduced in [14], [15].[1] Therefore, a utility function describes how the performance of an application changes with the amount of effective bandwidth it receives.

The utility of a request is an arbitrary function of the bandwidth received during its session, depending upon the application generating the request. Applications that are designated to be transmitted at a fixed rate with no interruptions may generate a request that has a *step* utility function as shown in Figure 1(a). In case of a step utility function, the received bandwidth is equal to the peak rate of the request. Applications that are designated to *adapt* to transmissions and transmit at a variable rate, delay adaptive and rate adaptive applications, may generate a request that has a utility function, as shown in Figures 1(b) and 1(c), respectively. Elastic applications generate requests with a utility function such as shown in Figure 1(d) [13].

Priority-based bandwidth allocation algorithms, with low time complexity, can be used to optimize the blocking probability and increase network utilization, by offering resources in a *more dynamic way* and providing preferential treatment for some services. The importance or "value" of a connection can be expressed by a priority level. The priority levels can be preassigned by the end-system or by the network administrator based on the type of the user. Alternatively, this can be done by using various factors, in order to relate to the connections Quality of Service (QoS) requirements, such as reliability of packet delivery, pricing, bandwidth requirement, timeliness, desired blocking probability, duration of the connection, geographical distance, or nature of the traffic

[1]Without loss of generality, we utilize statistical effective bandwidths in our generalized framework. However, several alternative quantities of similar use, for example deterministic effective bandwidths or traffic envelopes can be used instead [16], [17], [18].

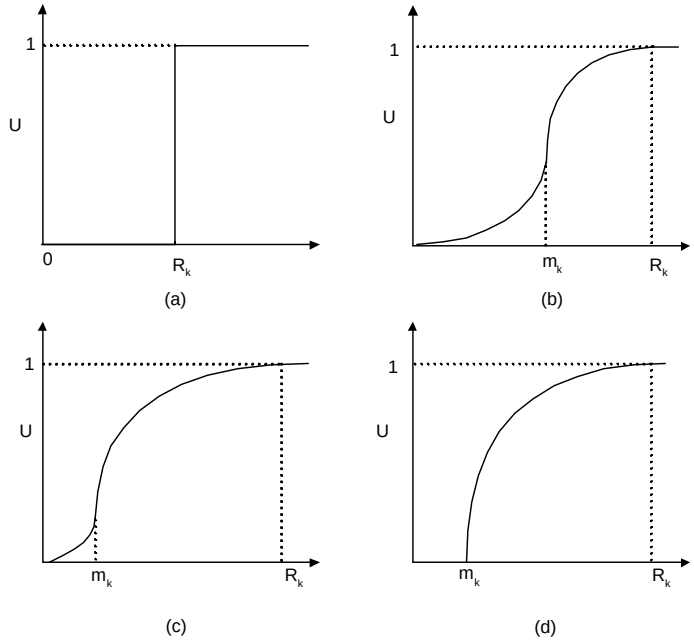

Fig. 1. Utility (performance) of hard real-time application (a), delay adaptive application (b), rate-adaptive application (c) and elastic application (d) as a function of bandwidth.

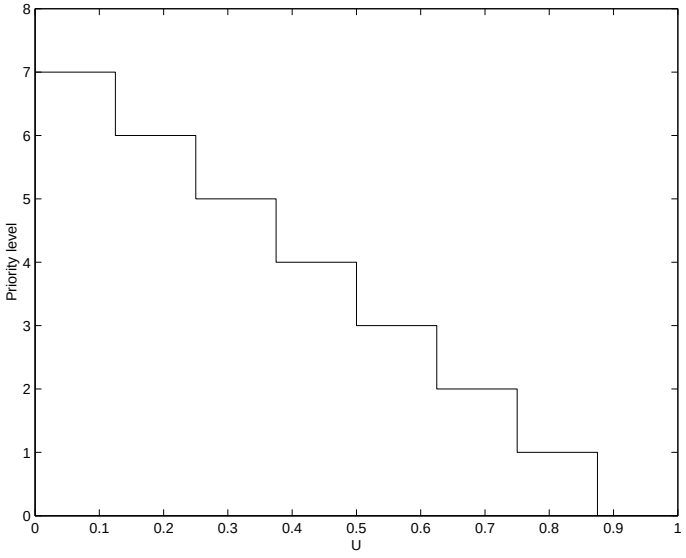

Fig. 2. Quantizing function for assigning priorities based on the weighted utility of the request.

such as voice, video, or WWW traffic.

In our approach, the priority level is assigned by quantizing the *weighted utility* of the request into N levels, where N cannot be too large or too small. In the MPLS model [1], [19] priorities can take values in the range from zero (0) to seven (7), with the value zero (0) being the priority assigned to the most important path. Therefore, the range of priority values is divided (without loss of generality) into eight levels using the quantizing function on Figure 2.

III. Rerouting algorithms

Once a request for a new connection arrives, the routers on the path to be established by the new request need to check for bandwidth availability on all links that compose the path. For the links in which not enough bandwidth is available, an algorithm has to decide which ongoing connections of lower priority to reroute in order to establish the high-priority connection. The algorithm is run *locally* in each link in order to guarantee the end-to-end bandwidth reservation for the new request. This is a decentralized approach, in which every node on the path would be responsible to independently determine which connections would be rerouted in order to accept the new request. For these reasons, a decentralized approach, although easier to be integrated in the current Internet environment, may not lead to a strictly optimal solution.

The two parameters describing a connection are, namely, *bandwidth* and *priority*. Which connections would be rerouted, if high priority users need resources but there is no room to accommodate them, can be determined by optimizing an objective function over these two parameters of the connections, and the number of connections to be rerouted. The objective could be any or a combination of the following:

1) Reroute the least amount of bandwidth. Network bandwidth is better utilized and there is minimum disruption of user traffic.
2) Reroute the connections that have the least priority. There is less disturbance of high-priority connections and the QoS of higher priority users is better satisfied.
3) Reroute the least number of connections. A minimum number of connections have to be rerouted.
4) Reroute the traffic according to the a performance criteria for each traffic type. Traffic should be rerouted as much when necessary, not whenever possible.

Much recent work has been accomplished to formulate optimal and approximate on-line algorithms for finding the best combination of which connections would be rerouted, if high priority users need resources but there is no room to accommodate them. One of the crucial performance objective of these algorithms is that they can be deployed over the Internet without significant modification within the network. In [20] the authors proposed two algorithms they named *Min_BW* and *Min_Conn* that optimize the criteria above in a certain

order of importance. The algorithm *Min_BW* optimizes the criteria of (i) the amount of bandwidth to be rerouted, (ii) the priority of connections to be rerouted, and (iii) the number of connections to be rerouted, in that order. The algorithm *Min_Conn* optimizes the criteria of (i) the number of connections to be rerouted, (ii) the bandwidth to be rerouted, and (iii) the priority of connections to be rerouted, in that particular order.

These algorithms are globally and strictly optimal with respect to their objective functions because they perform an *exhaustive* search to select a solution based on the criteria. In [21] the authors proposed an objective function that can be adjusted by the service provider in order to stress the desired criteria for optimization and derive a heuristic which approximates the optimal result. In [22], [23] we proposed and analyzed a *random selection* of connections to be rerouted from the set of connections with lower priorities and concluded that random selection algorithms could provide a high quality of service to higher-priority network connections, while utilizing network bandwidth efficiently.

IV. Restoration mechanisms

In this paper we analyze the prioritized resource allocation for reestablishing connections that have been disrupted by a failure. One aspect of fault recovery of the connection affected by the failure is the procedure for their re-acceptance into the network. Since the failure typically results in several nodes being sources for affected connections, in each of those nodes there will be many connections to simultaneously restore. Restoration algorithm needs to determine which connections are to be restored for finding the best combination of which connections would be rerouted from the set of connections affected by failure. The objectives of the restoration mechanisms are following:

- Maximize the bandwidth amount of restored traffic in the network.
- Maximize the number of restored connections in the network
- Maximize the total utility of restored requests
- Decrease the impact of the failure on higher priority connections.

Objectives of the restoration algorithms can be seen as the inverse of the objectives of the rerouting algorithms. Therefore, following approaches can be implemented for determining the order in which connection are to be restored:

1) Random ordering of connections.
2) Restore the connections by optimizing the criteria of (i) maximize the amount of bandwidth to be restored, (ii) the priority of connections to be restored, and (iii) the number of connections to be restored,in that particular order, which is the inverse of *Min_BW* algorithm.
3) Restore the connections by maximizing the criteria of (i) the number of connections to be restored, (ii) the bandwidth to be restored, and (iii) the priority of connections to be rerouted, in that particular order, which is the inverse of *Min_Conn* algorithm..
4) Prioritized per application type and process each connection in decreased ordering according to its utility value.
5) Prioritized over all application and process each connection in decreased ordering according to its utility value.

An algorithm for traffic restoration must be decentralized. If the system has information about the utility functions of the users, the optimization problem of maximizing the sum of the user utilities may be mathematically tractable. However, in practice not only is the system not likely to know the information about all users in the network, but also it is impractical for a single centralized system to compute and allocate the users' rates, due to the computational intractability of the problem for large networks. The increasing complexity and size of the Internet make centralized bandwidth allocation policies impractical. Distributed algorithms aim how one might *enhance*, if not optimize, average user-perceived performance and describe how it can be implemented in a real network.

We consider a network with a set of nodes N, set of resources or links J where link j has a capacity C_j and set of users K. Each user has a fixed route J_k, which is a nonempty subset of J. Let K_j represent a set of requests who share the link $j \in J$. Let S represent the set of satisfied requests that arrive in the network over a certain period of time. The problem that we are trying to solve is to determine which connections to restore from the set of connections affected by failure under the following objective and constraints. The total utility of all requests in S which we want to maximize is the sum of all utilities:

$$\max \sum_{k \in S} u_k \tag{1}$$

assuming that the utilities are additive, which is a reasonable assumption since the network users are independent, subject to the constraints:

$$\sum_{k \in K_j} c_k \leq C_j, \quad \forall j \in J \tag{2}$$

indicating that the total rate of the sessions using a link cannot exceed the capacity of the link.

We study the feasibility of achieving the maximum total utility of the users in a distributed environment, using only the information available at the *end hosts*.

V. Results

We have written a simulation program in *C* to study the first approach of randomly restoring the connections with the most general case of *static* resource allocation, called a *complete sharing* (CS) admission policy and dynamic bandwidth allocation policy with rerouting.

In the simulations, the two-tiered network topology shown in Figure 3 from [24] was used. We used a capacity corresponding to OC-1 for all links, without loss of generality.

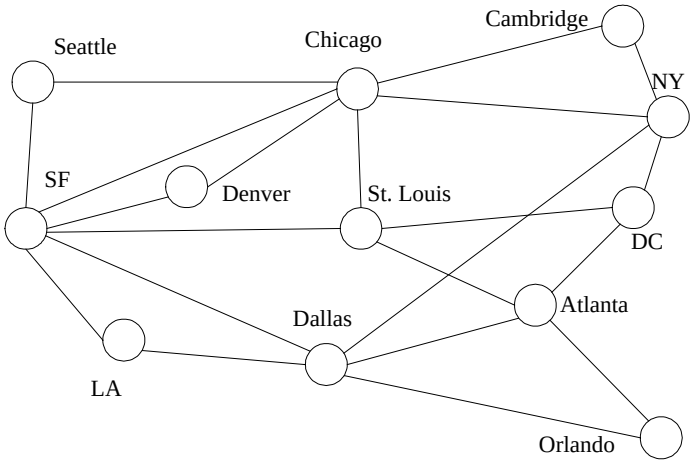

Fig. 3. Baseline network topology for simulation.

Since it is hard to obtain the traffic mix from real MPLS networks we chose to generate bandwidth values of connections randomly. The bandwidth range for mean rate of connections is taken to be between 64 Kbps and 4,000 Kbps with uniform distribution as in the paper by Peyravian and Kshemkalyani [20], where the maximum value of the required bandwidth corresponds to 8-10% of the OC-1 link capacity. The source and destination for the connections were selected randomly with a uniform distribution such that the load in the network is uniformly distributed.

A number of connections with different source-destination node pairs which are affected due to the link failure depends on the number of failed links and their location in the network. In our experiments the location of the failed links is chosen based on the degree of the node. From Figure 3, we can see that the minimum node degree is 2 and the maximum is 6. We increase the number of the failed link from single link failure to complete node failure.

The links are made to fail around 2000 seconds into the simulation, once the network has reached a stable level. We assume that the load is uniformly distributed in the network. Obviously the impact of the failure(s) on the traffic in the network and thus the network performance will depend on the node degree where link or multiple links failed. We vary the number of links from one to complete node failure for three nodes with node degrees 2, 4 and 6. The network performance metrics are represented in Figures 4 and 5.

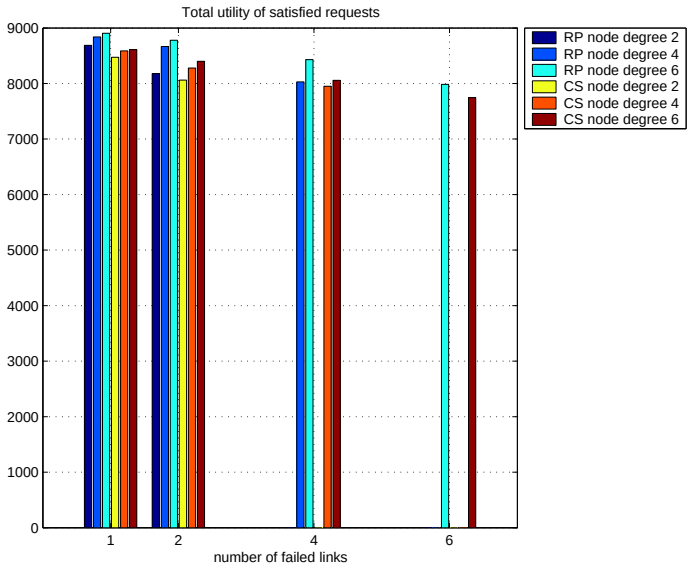

Fig. 4. The total utility of satisfied requests as a function of number of failed links for different node degrees for dynamic (RP) and static (CS) bandwidth allocation.

Figure 4 shows the *total utility of satisfied requests* as a function of number of failed links for different node degrees, and for dynamic (RP) and static (CS) bandwidth allocation. When the number of failed links is low, the impact of the link failure is much larger on the *smaller degree* nodes, as expected with the total utility of satisfied request decreasing as more links fail. Figure 5 and show the distribution of unsatisfied requests for the case of two failed links at a node with degree 4.

VI. Conclusions

In this paper, we presented a priority schemes for restoration of connections in communication networks. We 5 different approaches for ordering of the restored connections. Each of these algorithms tries to determine the best combination of which connections would be rerouted from the set of connections affected by failure over criteria defined in the paper. We presented initial

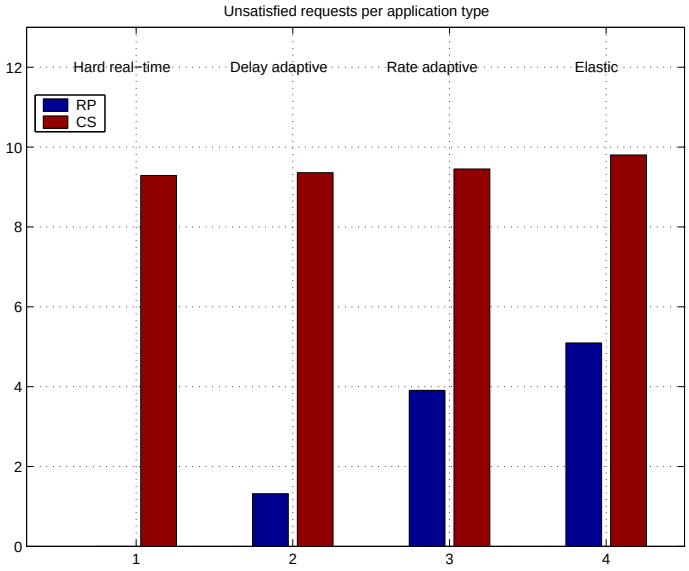

Fig. 5. The number of unsatisfied requests per application type for the case of two failed links at node with degree 4, for dynamic (RP) and static (CS) bandwidth allocation.

results for the first restoration approach of randomly selecting the connections affected by failure in presence of complete sharing and dynamic bandwidth allocation. Using a dedicated priority scheme, the network performance after a failure is improved and the impact of failure on higher priority connections is decreased. Work will continue in an effort to give a complete picture of the restoration mechanisms by investigating other optimal approaches for determining the order in which connection are to be restored.

REFERENCES

[1] D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, *Overview and Principles of Internet Traffic Engineering*, IETF, RFC 3272, May 2002.

[2] T. E. et al., "Qos restoration for dependable networks," Network Operations and Management Symposium, NOMS'98, February 1998, new Orleans, Louisiana.

[3] E. M. Noam, "Testing the communications network," *The New York Times*, September 24 2001.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *An Architecture for Differentiated Services*, IETF, RFC 2475, December 1998.

[5] S. Maniatis, E. G. Nikolouzou, and I. S. Venieris, "QoS Issues in the Converged 3G Wireless and Wired Network," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 44–53, August 2002.

[6] F. Kelly, "Charging and Accounting for Bursty Connections," in *Internet Economics*, J. Surdej, D. Fraipont-Caro, E. Gosset, S. Refsdal, and M. Remy, Eds., 1997, pp. 253–278.

[7] P. Dharwadkar, H. J. Siegel, and E. K. P. Chong, "A Heuristic for Dynamic Bandwidth Allocation with Preemption and Degradation for Prioritized Requests," in *Proceedings of the 21st International Conference on Distributed Computing Systems*, Phoenix, Arizona, April 16–19 2001, pp. 547–556.

[8] L. DaSilva, D. Petr, and N. Akar, "Equilibrium Pricing in Multiservice Priority Based Networks," in *Globecom 97*, Phoenix, AZ, November 1997.

[9] R. J. La and V. Anantharam, "Utility-Based Rate Control in the Internet for Elastic Traffic," *IEEE/ACM Transactions on Networking (TON)*, vol. 10, no. 2, 2002.

[10] Y. Cao and V. O. K. Li, "Utility-oriented Adaptive QoS and Bandwidth Allocation in Wireless Networks," in *ICC 2002*, New York City, April 2002.

[11] L. Song and N. B. Mandayam, "Hierarchical SIR and Rate Control on the Forward Link for CDMA Data Users under Delay and Error Constraints," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 1871–1882, 2001.

[12] C. Saraydar, N. Mandayam, and D. Goodman, "Efficient Power Control Via Pricing in Wireless Data Networks," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 291–303, 2002.

[13] S. Shenker, "Fundamental Design Issues for the Future Internet," *IEEE Journal on Selected Areas in Communication*, vol. 13, no. 7, pp. 1176–1188, September 1995.

[14] L. Gun and R. Guerin, "Bandwidth Management and Congestion Control Framework of the Broadband Network Architecture," *Computer Networks and ISDN Syst.*, vol. 26, no. 1, pp. 61–78, 1993.

[15] H. Ahmadi, J. Chen, and R. Guerin, "Dynamic Routing and Call Control in High-speed Networks," in *Teletraffic and Data Traffic: Socio-Economic Aspects*, June 1991, pp. 397–403, proc. 13th International Teletraffic Congress (ITC-13), Copenhagen, Denmark.

[16] J. Qiu and E. W. Knightly, "Measurement-Based Admission Control with Aggregate Traffic Envelopes," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 199–210, 2001. [Online]. Available: citeseer.nj.nec.com/qiu01measurementbased.html

[17] N. Fonseca and R. Facanha, "Statistical Multiplexing of Self-Similar Sources," in *Proceedings of IEEE Globecom 00*, vol. 3, San Francisco, Nov.-1 Dec. 2000 2000, pp. 1334 –1338.

[18] A. I. Elwalid and D. Mitra, "Effective Bandwidth of General Markvian Traffic Sources and Admission Control of High Speed Networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 1, no. 3, pp. 329–343, 1993.

[19] F. L. Faucheur and W. Lai, *Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering*, IETF, RFC 3564, July 2003.

[20] M. Peyravian and A. D. Kshemkalyani, "Decentralized Network Connection Preemption Algorithms," *Computer Networks and ISDN Syst.*, vol. 30, no. 11, pp. 1029–1043, June 1998.

[21] J. de Oliveira, C. Scoglio, I. F. Akyildiz, and G. Uhl, "A New Preemption Policy for DiffServ-Aware Traffic Engineering to Minimize Rerouting," in *INFOCOM 2002*, New York City, June 2002.

[22] V. Stanisic and M. Devetsikiotis, "A Dynamic Study of Providing Quality of Service Using Preemption Policies with Random Selection," in *IEEE ICC 2003*, Anchorage, Alaska, May 2003.

[23] ——, "Dynamic Utility-based Bandwidth Allocation Policies: The Case of Overloaded Network," September 2003, to be presented at ICC 2004.

[24] N. G. Duffield, P. Goyal, A. G. Greenberg, P. P. Mishra, K. K. Ramakrishnan, and J. E. van der Merive, "A Flexible Model for Resource Management in Virtual Private Networks," in *SIGCOMM*, 1999, pp. 95–108.