Journal of Network and Systems Management, Vol. 14, No. 4, December 2006 (© 2006)
DOI: 10.1007/s10922-006-9045-6

A Bandwidth Monitoring Mechanism Enhancing SNMP to Record Timed Resource Reservations

K. Manousakis,[1] V. Sourlas,[1] K. Christodoulopoulos,[1] E. Varvarigos,[1] and K. Vlachos[1,2]

The efficient use of resources and the lossless transfer of data bursts in future optical networks requires the accurate knowledge of the available bandwidth for each network link. Such information is important in monitoring congestions and can be used by appropriate load balancing and congestion avoidance mechanisms. In this paper we propose a mechanism for monitoring and subsequently managing bandwidth resources, using the Simple Network Management Protocol (SNMP). In the proposed mechanism, link bandwidth availability is not a scalar parameter, but a function of time that records the future utilization of the link. For every output port, each agent-node maintains a simple data structure in the form of a table that records the *utilization profile* of that outgoing link. With the addition of new objects in the Management Information Base (MIB) of each agent-node and proper synchronization, SNMP can be used to update and retrieve the reservations made on the links in order to obtain an instant picture of the network traffic situation.

KEY WORDS: network management; link utilization profile; congestion avoidance; timed reservations.

1. INTRODUCTION

In the era of multi-gigabit networking, which is already becoming a reality, network delays are dominated by link propagation delays [1]. The high propagation latencies (relative to the burst/session transmission times) differentiate high-speed from conventional networks, and have an impact not only on the performance of such networks, but also on the protocols designed to manage them and the applications that use them. Particularly the bandwidth-delay product, being large,

---

[1] Computer Engineering and Informatics Department and Research Academic Computer Technology Institute, University of Patras, Rio, Greece.
[2] To whom correspondence should be addressed at Computer Engineering and Informatics Department and Research Academic Computer Technology Institute, University of Patras, GR26500, Rio, Greece. E-mail: kvlachos@ceid.upatras.gr.

can result in the discarding of substantial amounts of data if lossless transfer is not provided. Further, for high-speed file-transfer and grid applications, long session/burst transmissions can easily overload the network if adequate capacity is not reserved in advance. Therefore, the connection establishment and flow control protocols need to be re-examined to meet the characteristics of this new communication paradigm.

Using a conventional protocol for reserving capacity, such as RSVP-TE [2] (Resource Reservation—Traffic Extension) and CR-LDP [3] (Constrained—based Label Distribution Protocol), for a burst or a session is inefficient, since such protocols do not convey temporal information, while the capacity at a node is actually used at least one round-trip delay later than the arrival of the setup message making the reservation at the node. Over long transmission distances, the round-trip propagation delay may be comparable to, or even larger than, the holding time of the transmission, leading to low efficiency in the use of resources. Recording information on the bandwidth availability as a function of time is important for monitoring the congestion in the network and it can be used by appropriate load balancing and congestion avoidance mechanisms. For example in Multi-Protocol-Label-Switched (MPLS) networks, the bandwidth availability information can be used in the process of selecting the routes over which the Label-Switched-Paths (LSPs) are established.

In this paper, we present a mechanism for monitoring and subsequently managing network resources, in which the structures that are employed to record bandwidth utilization information as a function of time can be accessed by and transferred with a network management protocol. More specifically, for this study we have assumed that the protocol used for network management is the Simple Network Management Protocol (SNMP). By keeping these structures at the SNMP's Management Information Base (MIB) (database) we maintain a *picture* of the network resources availability as a function of time. In order to serve a connection establishment request, a sophisticated route and path selection algorithm may use this information (retrieve bandwidth information from the Network Management System—NMS), in order to avoid congested nodes and further to balance the load in the core network. We present a routing paradigm that uses the proposed monitoring mechanism and evaluate its performance in a full network simulation scenario. More specifically, for every transmission request a simple congestion avoidance algorithm takes into account the utilization profiles of the links over the shortest path and schedules the burst/session accordingly. The resource reservation/utilization update process is performed by sending messages that contain temporal information (timed/delayed reservations) to all the nodes that comprise the path.

The idea of monitoring the resources and particularly monitoring the utilization of links is definitely not new. Several techniques have been proposed in the literature, but have been limited in estimations for the bandwidth usage. In this case, the information about the link utilization is only coarsely estimated or can be outdated by the time it is received. One such approach that uses SNMP variations is studied in [4] and [5]. An extension of the link utilization estimation is presented in [6, 7]. More specifically, [6] presents a QoS-aware path selection algorithm for flows requiring bandwidth guarantees based on an estimation algorithm for the available bandwidth on the links of the network. The estimation algorithm predicts the available bandwidth and also provides the duration for which the estimate is valid with a high degree of confidence. [7] presents a method for monitoring i) the bandwidth usage for a given set of links or packet flows, and ii) the path latencies for a given set of paths, while minimizing the overhead imposed by the management tools on the underlying production network.

The major differences between our proposed scheme and the MPLS-TE (Traffic Engineering) that is widely adopted in today's networks are:

- In MPLS-TE, link state information floods the network in order to reach the appropriate nodes, resulting in a large network control plane overhead and consequently in a large processing overhead. In our approach the link utilization information is maintained at each node and accessed/retrieved by the Network-Management-System through the underlying SNMP protocol. In order to demonstrate the efficiency of this mechanism, we propose and evaluate a simple congestion avoidance mechanism that upon a transmission request obtains the bandwidth utilization profiles of the links that comprise the shortest path between the source node and the specified destination and then calculates the exact time after which adequate capacity is available across that path.
- In our approach *resources are reserved only for the time duration they are actually used,* and thus potential gaps (voids) in capacity usage can be monitored and exploited in order to serve other requests. Signalling protocols, used in MPLS i.e. RSVP-TE [2] and CR-LDP [3], do not convey temporal information and thus these schemes cannot be used for accurately monitoring bandwidth availability. In our approach, the nodes are able to calculate and use the correct information regarding the utilization of the links by keeping track of the utilization profile of a link as a function of time and also by taking into account the network propagation delays.

The remainder of the paper is organized as follows. Section 2, describes the node structure and the features needed to enhance SNMP with the monitor capability, while Section 3 describes a mechanism that can be used for updating the node utilization profiles and exchanging bandwidth link state information. Section 4 presents the routing and timed reservation problem as an application example that can benefit from the proposed monitoring mechanism. Section 5 presents performance evaluation results.

2. NODE STRUCTURE AND BANDWIDTH MONITORING MECHANISM

In this section we propose a mechanism for monitoring bandwidth reservations and future link utilization, and describe the data structures that are required at every node. More specifically, in order to monitor network resources, we enhance the Management Information Base—MIB- of each node-agent and define a new table structure that is commissioned to contain the links state information. Additional implementation issues related to node synchronization and reservation issues are discussed in Section III.

2.1. Node Data Structures

We assume that each session or burst reserves a constant amount of bandwidth for a given time duration. The utilization profile [8, 9] $U_l(t)$ of a link $l = (i, j)$ is then a stepwise function with discontinuities at the points at which reservations begin or end (Fig. 1), and is updated dynamically with the admission of each new session or burst. We define the *capacity availability profile* of link $l = (i, j)$ as

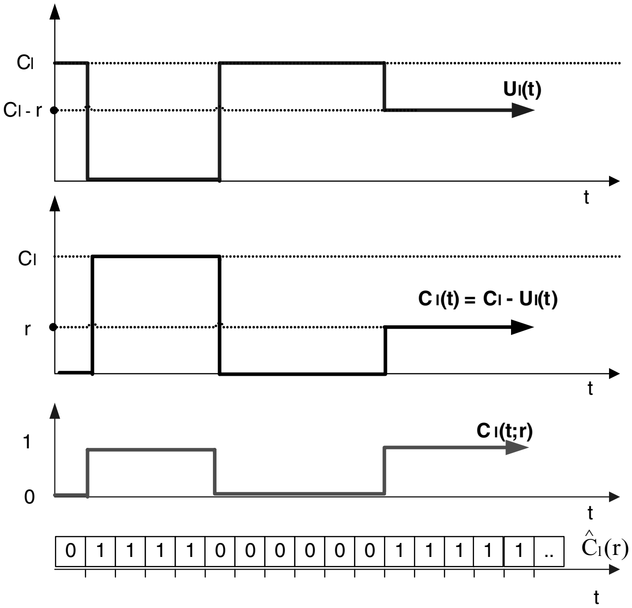


Fig. 1. Illustrates the utilization profile $U_l(t)$, the capacity availability profile $C_l(t)$, the r-capacity availability profile $C_l(t; r)$, and the binary r-capacity availability vector $\hat{C}_l(r)$ of a given link $l = (i, j)$ of capacity C_l when the rate requested by the burst is r and the time discretization step is τ.

$C_l(t) = C_l - U_l(\mathrm{t})$ where C_l is the capacity of link l. Since a source that tries to route a burst of rate r is only interested in time periods at which $C_l(t) > r$, we also define the *r-capacity-availability profile* $C_l(t;\, r)$ of link $l = (i, j)$ as the binary function:

$$C_1(t;r) = \begin{cases} 1 \text{ if } C_1(t) \geq r \\ 0 \quad \text{otherwise} \end{cases}$$

Generally, the duration of the bursts that pass through a link and the link propagation delays are arbitrary, and thus the time intervals at which bandwidth r is available and their starting/ending points are also arbitrary. In order to obtain from the r-capacity availability profile $C_l(t;\, r)$ of link l a data structure that is easier to store and manipulate, we discretize the time in steps of duration τ and obtain a *binary r-capacity availability vector* $\hat{C}_l(r)$**,** abbreviated BAV. Following this process, BAV is a vector whose k-th entry is:

$$\{\hat{C}_1(r)\}_k = \left\{ \begin{array}{ll} 1, & \text{if} C_1(t;r) = 1, \text{ for all} (k-1)\tau \leq t \leq k\tau \\ 0, & \text{otherwise} \end{array} \right\}, \; k = 1, 2, \ldots, d$$

The discretization of the horizontal axis (time) in time steps of duration τ results in some loss of information (unless all burst durations and propagation delays are multiples of τ). Assuming that D is the maximum allowable delay of a burst we wish to route, we are only interested in values of $C_l(t;\, r)$ before time D from the present time, or equivalently we can assume that the binary r-capacity availability vector $\hat{C}_l(t;\, r)$ of link $l = (i, j)$ has $d = [\mathrm{D}/\tau]$ entries. If we choose a small value for τ then we will have better accuracy in recording the availability profiles of the links, but the binary availability vector will have a rather large dimension d and may be difficult to store, transfer and manipulate. In contrast, if the discretization step τ is chosen to be large, the binary availability vectors will have an acceptable for manipulations size with a degrading accuracy as the predominant trade off. In any case, the choice of the discretization step τ depends on the requirements of the application and the trade off between the required accuracy/efficiency and the size of the binary utilization vectors that we are willing to maintain and process.

For the rest of the paper, when no confusion can arise, we will denote the profiles $C_l(t;\, r)$ and $\hat{C}_l(r)$ of a link $l = (i, j)$ by $C(t)$ and $\hat{C}$, respectively, suppressing the dependence on $l = (i, j)$ and r, in order to simplify notation.

2.2. Enhancing Simple Network Management Protocol

The binary r-capacity availability vector $\hat{C}$ (BAV) presented in the previous paragraph must be defined in the MIBs of the nodes-agents. The information in a management system is organized as a set of objects, each of which constitutes

Table I. Object type and description

Object-type	Description	Object-ID
Monitor Link	The id of a link of the node; if a node has k links then a number from 1 to k	Monitor 1
Monitor Entry	The binary r-capacity availability vector $\hat{C}$ (BAV) of the corresponding link	Monitor 2
Link Delay	Propagation delay of the link	Monitor 3
Creation Time	The UTC time when the monitor Table was created	Monitor 4

an abstraction or a representation of a network element. In SNMP, there is a mechanism that maintains the relation between the elements of the network and the objects that represent them. For that reason we define in the MIB an object that contains the number (id) of the particular link, and a BAV.

Using Abstract Syntax Notation One (ASN.1) we can define the vector that corresponds to the utilization profile at a node, as summarized in Table I. Figure 2 depicts these objects in a tree form under the mib-2 sub-tree.

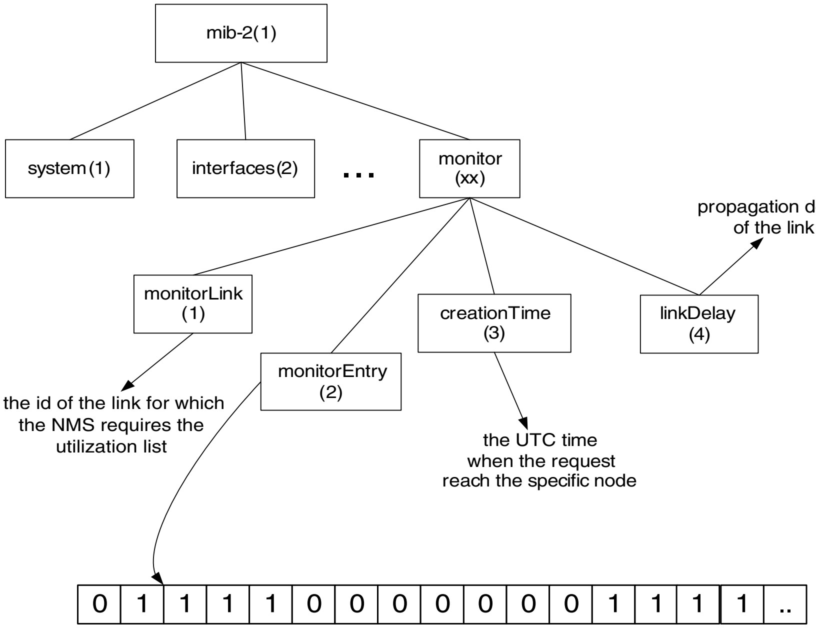


Fig. 2. Monitor sub-tree and all its associated objects.

Additional important objects that are defined under the *monitor* branch and should be transported via the SNMP are the *propagation delay* of the examined link and the *creation time*, which is the time when the request reaches the specific node. When another node receives the table, it can use the *creation time* in order to discard the entries in the table that have expired (due to the transmission and propagation delay and generic synchronization aspects).

A source node wishing to retrieve these tables first contacts the NMS by sending a TRAP-PDU. After the reception of a TRAP request, the NMS retrieves the link state information by sending GET or GET-NEXT requests to the appropriate network nodes (for example, the nodes that constitute the shortest path). The node-agent sends responses until the entire table has been retrieved. For the retrieval of the table, the NMS can also use the GET-BULK operation, provided by SNMPv2 and SNMPv3, which allows the NMS to retrieve an entire table at once. We can organise the information retrieval process to be performed either on demand or automatically. In the former case the NMS starts sending GET requests to the appropriate nodes only when a TRAP-PDU arrives, while in the latter case the NMS periodically requests the BAV of the links and updates these data in its database. Clearly, both approaches have advantages and disadvantages. The first approach may result in a higher delay (to gather the information), which may not be tolerable for some sessions/bursts. In the second approach, the information is already at the NMS and the delay is smaller, but it has the disadvantage that the information may not be up-to-date and that not all the transmitted information is used (periodic refreshments waste network resources).

2.3. Traffic Monitoring and Route Selection Mechanisms

When a new burst/session request arrives at a node, an open issue is who will make the decision about the path to be followed: the source node or the NMS. As presented in the previous paragraph, the NMS is more likely to be aware of—or can easily obtain the current network traffic state. Thus, it is more efficient to forward the address of the end-destination to the NMS in which to apply a routing algorithm. In this case, the algorithm can take into account the links state information and choose the path with the smallest propagation delay, the minimum number of hops, or the largest available bandwidth, etc. Figure 3 shows the full operation of the proposed traffic-monitoring and the route selection scheme assuming that the NMS retrieves network state information upon the reception of a TRAP (transmission) request from the source node.

A source wishing to make a reservation for a flow with specific QoS, requirements first sends a trap to the NMS with the *Address* of the destination. The NMS first analyzes the request, computes candidate paths, requests the tables' information from the corresponding nodes and finally sends back to the source a path and (optionally) a time offset after which the transmission should start. The algorithm

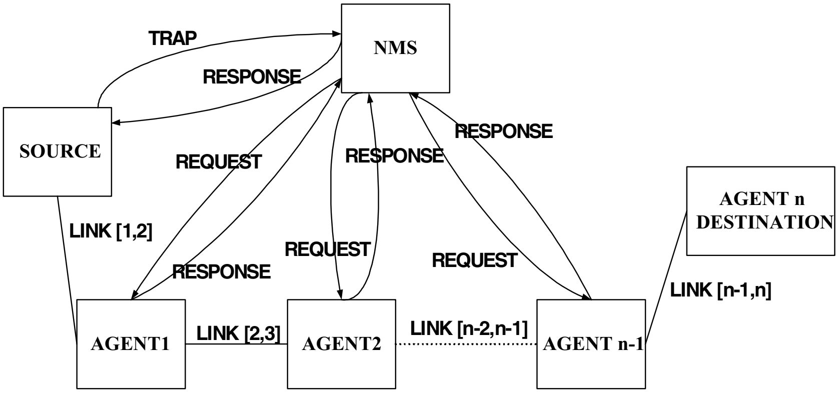


Fig. 3. Diagram of the traffic monitoring and route selection mechanisms.

that decides the path to be followed takes into account the QoS requirements of the flow and the utilization profiles of the links. In order to provide an example of the function of the proposed routing paradigm (and for simplicity purposes) in this study we have experimented with the Dijkstra's shortest path algorithm. Thus for the following section we assume that the NMS always chooses the path with the smallest propagation delay.

2.4. Node Synchronization

For the implementation of the proposed traffic monitoring mechanism, accurate and reliable timing is necessary to record the *creation time* of each request. In our case, timing information is not used just for the synchronization of nodes but is treated as a *timestamp* for the received *picture* of the resources availability. Thus, this information must be sent along with the other information retrieved from each node-agent. For such a purpose, NTP (Network Time Protocol) can provide the necessary time accuracy (order of msec).

3. A MECHANISM FOR UPDATING AND EXCHANGING BANDWIDTH AVAILABILITY INFORMATION

Conventional schemes for flooding link state information between the nodes result in a large number of messages exchanged over the signalling channel and increase the processing overhead. To overcome these deficiencies we consider a centralized approach, in which we can enforce a management policy with respect to the message exchange process. According to the routing paradigm presented in the previous section, the centralized NMS is responsible for retrieving the links' state information. The NMS then computes the path (always shortest path for this study) and the time instances that the data of the session/burst will pass from the links. Since this information is available, we extend the role of the NMS so that it also informs the nodes of the selected path for the timed reservation. This is performed by sending to each node of the chosen path a SNMP TPAP packet.

The parameters that the TRAP packet needs to convey to enable an agent-node to maintain/update a *utilization profile* are:

- The link identifier (*Lid*)
- The start time, denoted as ST_i, that specifies the time at which the reservation of capacity of the specific outgoing link (*Lid*) should begin.
- The requested bandwidth r.
- The information field I that specifies the amount of information that will be transmitted during the holding time of the session.

An intermediate node i on the path, upon reception of a TRAP packet carrying these parameters, finds the utilization profile that corresponds to *Lid* and (if this is feasible) adds a jump of size r at time ST_i for a duration that corresponds to the transition of I bits (time equal to I/r).

4. APPLICATION: TIMED RESERVATION

One application that can benefit from the proposed traffic-monitoring mechanism is the timed reservation mechanism. A bandwidth reservation request (with a requested rate r, an estimate of the amount of the data to be transmitted I, and a maximum delay tolerance D), is forwarded from source node S to the NMS. The NMS retrieves from the nodes-agents the objects described in section II (BAVs, link propagation delays and timestamps). Thus, by subtracting the time that the information reached the NMS from the time the agent i received the GET request for the corresponding link l (*creation time*), the NMS calculates the propagation delay $d_{i,NMS}$. By the time the BAV $\hat{C}_l$ of link l reaches the NMS, its first $d_{i,NMS}$ entries will correspond to time intervals that have already expired. So in order to have a present picture (present traffic profile), the NMS should delete the first $d_{i,NMS}$ entries of the vector $\hat{C}_l$. Moreover, the NMS should discard the first $d_{S,i}$ of the resulting $\hat{C}_l$ to take into account the propagation delay that any data sent from S will suffer in order to reach node N. Since in this study we consider only the shortest path routing algorithm, this process is performed for the links that constitute the shortest path between the source S to the specified destination. Taking a different approach, if a routing selection algorithm was also applied, the NMS would retrieve and manipulate BAVs from other links in order to investigate alternative paths. In either case, with the state information of the links comprising a path in hand, the NMS is capable of finding the time offset TO (in terms of time discretization steps τ) after which the transmission of data should start in order to

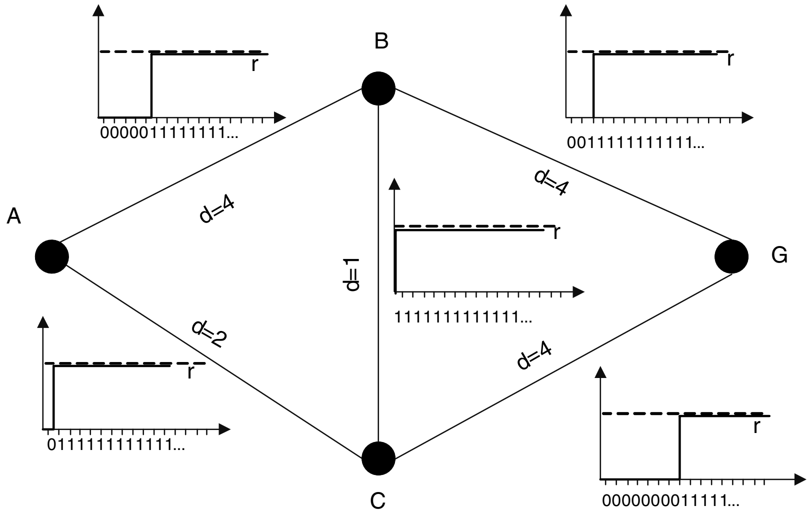
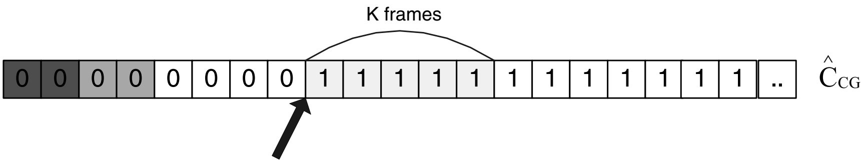


Fig. 4. A typical network topology and the corresponding link utilization profiles. A request for capacity *r* to transmit *I* data occurs at node *A* destined to node *G*.

find available capacity at all intermediate links of the path. It is worth noting that the availability of the capacity in this path is guaranteed if no (other) reservations are performed in the meanwhile at the intermediate nodes. Figures 4 and 5 present an example in order to illustrate this process.

Figure 4 shows a network of 4 core nodes. Node *A* requests to reserve capacity *r* for the transmission of *I* data to destination node *G*. For simplicity we assume that the NMS is located at node *A* and that the Dijkstra's algorithm is used for the selection of the path. Thus the path ACG is chosen. For each of the links that constitute the shortest path (links *AC* and *CG*) an utilization profile is kept at the



$\hat{C}_{CG}$ is shifted left 4 frames: 2 corresponds to time intervals that have already expired (red color) and 2 for the forward propagation delay(orange color)

Fig. 5. By subtracting the relevant propagation times we compute the earlier availability (supposing that the data transmission time is equal to $K = I/(r \cdot \tau)$ frames long). The first available time is shown by the arrow.

corresponding node agents (nodes A and C respectively). The insets of Fig. 4 show these profiles and the requested capacity r (marked with a dotted line). In order to calculate the time offset TO after which the transmission should begin, the NMS searches for the first time instance where the quantity available capacity in all the links down the path becomes positive for duration equal to the data holding time defined by I/r (equals to $K = 5$ in this example). Figure 5 shows the BAV manipulation steps for the link CG. More specifically, the time $d_{C,NMS} = d_{C,A}$ has been discarded (by shifting the vector components) due to the fact that this information corresponds to time intervals that have already expired by the time $\hat{C}_{CG}$ reaches the NMS. Moreover, to take into account the forward propagation delay to reach agent C we subsequently discard from the remaining vector the first $d_{A,C}$ elements. In Fig. 5 the arrow depicts the first time instance at which the requested capacity is available at link CG. After combining this information for all the links, the NMS calculates the time offset (TO) after which the transmission should start. In the sequence, the NMS pass this to the source A and also sends SNMP TRAP messages to inform the intermediate nodes for the timed reservation (only to node-agent C in the example). It is worth noting that we have assumed that the source node A has enough buffer capacity to store the data of the session until it starts transmission (i.e. after time TO).

5. PERFORMANCE RESULTS

In order to asses the applicability of the proposed monitoring mechanism we performed full network simulation experiments. We have extended the ns-2 platform [10] and tested the following two routing algorithms:

- *The Dijkstra shortest path algorithm.* Bursts/sessions are routed on the shortest propagation delay path taking into account reservations made only at the first hop. If at an intermediate node there is no adequate capacity the transmission is considered blocked.
- *The Dijkstra shortest path algorithm with contention avoidance* (*Dijkstra/CA*). This algorithm implements the paradigm presented in the previous section. The shortest paths are computed at the beginning of the simulation for every source-destination pair. Upon a transmission request the source node sends a TRAP request to the NMS that collects and manipulates the utilization profiles of the links of the specific (shortest) path. The NMS sends a reply to the source with a time offset after which the transmission should begin and also informs the intermediate nodes for the timed reservation (via a TRAP update-message as described in section III). The TRAP message may not find adequate capacity at an intermediate node, due to the fact that the utilization profile at that specific link may have changed by the time the TRAP message arrives. In this case the transmission is considered blocked.

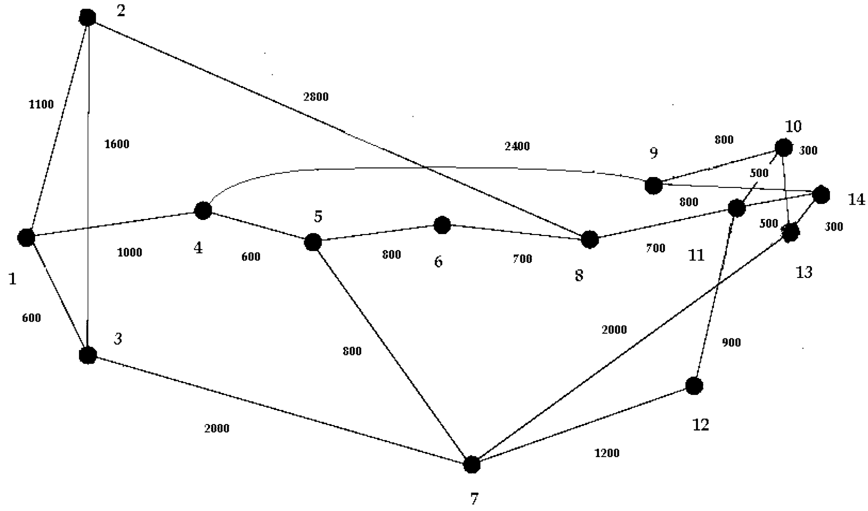


Fig. 6. The NSF network topology.

The experiments were conducted assuming the NSF network topology (Fig. 6 - [11]). In the NSFNET, all links were assumed to be bi-directional with a single wavelength per direction of bandwidth $C = 1\ Gb/s$. Propagation delays were taken to be proportional to the physical distance between the nodes (5 μsec/Km), and the SNMP packet processing delay was set to 0.05 μsec. At each edge node, burst/session transmission requests arrive following a Poisson process with rate λ requests per second and burst destinations were uniformly distributed over all nodes. We have assumed that a burst/session requests to reserve the full capacity of a wavelength and thus $r = C$. Typical mean burst transmission durations (I/C) were considered between $0.1 - 0.8$ msec that are two order of magnitude less than the mean propagation delay of the network ($\sim$ 11 msec). We have considered the transmission blocking probability and the average end-to-end delay as the main metrics for our simulation.

Figure 7(a) shows the transmission blocking probability and the corresponding average end-to-end delay of both algorithms as a function of the transmission arrival rate λ for two average transmission sizes ($I = 200$ KBytes and $I = 400$ KBytes), while Fig. 7(b) presents the same metrics as a function of size I, for two arrival rates ($\lambda = 50$ and $\lambda = 100$ requests per second). As expected, for both cases the blocking and the average end-to-end delay performances deteriorate as the offered load increases (increase of λ or I in Figs 7(a) and (b) respectively). Throughout all the experiments, the Dijkstra/CA algorithm is shown to outperform the simple Dijkstra algorithm. We can observe that the Dijkstra/CA scheme can ensure a low transmission block probability with a small penalty in the end-to-end

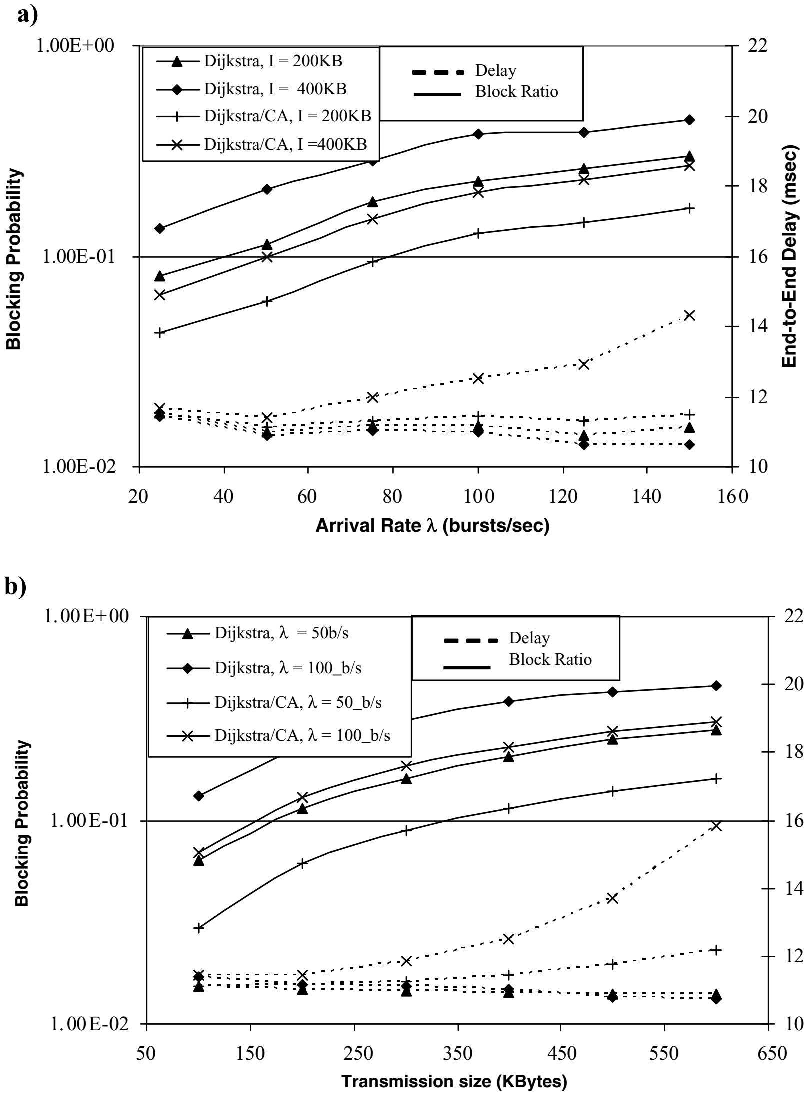


Fig. 7. (a) Blocking brobability and average end-to-end delay as function of Poisson arrival rate (λ), for 2 average transmission size values. (b) Loss ratio and average end-to-end delay as function of average transmission size (*I*) for two different λ values.

delay, a penalty that increases as the network load increases. The penalty in the average end-to-end delay is the accumulation of the time offsets (TOs) issued in order to efficiently schedule the requests and thus can be considered as the tradeoff for the low blocking probability at subsequent nodes.

We have demonstrated with a simple routing paradigm that we can use the links' state information in order to obtain significant improvements in the acceptance of transmission requests. Therefore, we can conclude that the proposed monitoring mechanism that records the link bandwidth availability as a function of time can facilitate the efficient utilization of the network resources and a low blocking probability for bandwidth reservation requests. Further studies are required in order to identify sophisticated routing and scheduling algorithms that would further exploit this information.

6. ACKNOWLEDGEMENTS

This work has been supported by the European Commission through the IST-LASAGNE and *e*-Photon/One + projects, and by GSRT through the Pythagoras project.

REFERENCES

1. L. Kleinrock, The latency/bandwidth trade off in gigabit networks, *IEEE Commun Mag*, Vol. 30, No. 4, pp. 36–41, 1992.
2. D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, *RSVP-TE: Extensions to RSVP for LSP tunnels*, IETF RFC 3209, Dec. 2001.
3. P. Ashwood-Smith and L. Berger, *Generalized Multi-Protocol Label Switching (GMPLS) Signaling Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions*, RFC 3472, Jan. 2003.
4. S. Amarnath, A. Kumar, A new technique for link utilization estimation in packet data networks using SNMP variables, *Proceeding of IEEE Global Telecommunications Conference*, Vol. 1, pp. 212–216, 1997.
5. J. Strauss, D. Katabi, and F. Kaashoek *A Measurement Study of Available Bandwidth Estimation Tools*, Proceedings of the ACM SIGCOMM Internet Measurement Conference pp. 39–44, Oct. 2003.
6. T. Anjali, C. Scoglio, J. Oliveira, L. Chen, I. Akyildiz, and J. Smith, *A New Path Selection Algorithm for MPLS Networks Based on Available Bandwidth Estimation*, Proceedings of Quality of future Internet Services conference, pp. 205–214, Oct. 2002.
7. Y. Breitbart, C. Y. Chan, M. Garofalakis, R. Rastogi, and A. Silberschatz, *Efficiently Monitoring Bandwidth and Latency in IP Networks*, Proceedings of IEEE International Conference on Computer Communication, pp. 933–942, Apr. 2001.
8. E. A. Varvarigos and V. Sharma, An efficient reservation connection control protocol for gigabit networks, *Computer Networks and ISDN Systems*, Vol. 30, No.12, pp. 1135–1156, 1998.
9. J.T. Vitramo, A model of reservation systems, *IEEE Trans. Communications*, Vol. 40, No. 1, pp. 109–118, 1992.
10. The Network Simulator—ns-2: http://www.isi.udu/nsnam/ns
11. NSF network: http://moat.nlanr.net/INFRA/NSFNET.html

Konstantinos Manousakis received the diploma degree from the Computer Engineering and Informatics Department, University of Patras, Greece, in 2004. Currently, he is pursuing his MSc in Computer Science and Engineering of the Computer Engineering and Informatics Department. His research interests include routing and wavelength assignment, all-optical networks and management protocols.

Vasileios Sourlas received the diploma degree from the Computer Engineering and Informatics Department, University of Patras, Greece, in 2004. Currently, he is pursuing his MSc in Computer Science and Engineering of the Computer Engineering and Informatics Department. His current research interests include routing and scheduling protocols in optical networks.

Konstantinos Christodoulopoulos received the Diploma of Electrical and Computer Engineering from the National Technical University of Athens, Greece, in 2002 and the M.Sc. degree in Advanced Computing from Imperial College London, U.K., in 2004. He is currently working toward the Ph.D. degree at the Department of Computer Engineering and Informatics of the University of Patras, Greece. His research interests are in the areas of protocols and algorithms for optical networks and grid computing.

Emmanouel (Manos) Varvarigos received a Diploma in Electrical and Computer Engineering from the National Technical University of Athens in 1988, and the M.S. and Ph.D. degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology in 1990 and 1992, respectively. He has held faculty positions at the University of California, Santa Barbara and Delft University of Technology, the Netherlands. In 2000 he became a Professor at the department of Computer Engineering and Informatics at the University of Patras, Greece, where he heads the Communication Networks Lab. He is also the Director of the Network Technologies Sector at the Research Academic Computer Technology Institute. His research activities are in the areas of high-speed networks, protocols, network architectures, distributed computation and grid computing.

Kyriakos G. Vlachos received the Ph.D. degree from the National Technical University of Athens, Greece, in 2001. From 2001 to 2003, he was a Member of the Technical Staff of Bell Laboratories, Lucent Technologies, while from 2003 to 2005 he was a research associate at Computer Engineering Laboratory, Delft University of Technology. From 2004 to 2005, he served as a telecom advisor for the National Regulatory Authority of Greece and since 2005 he has been a faculty member in the Computer Engineering and Informatics Department of the University of Patras, Greece. He has participated in various European and National research and development programs. He holds 5 international patents and is the author of more than 65 journal or conference publications.