

# A Distributed Scheme for Bandwidth Re-Allocation between Working and Recovery Paths

A.E. Krzesinski and K.E. Müller

**Abstract**—This paper presents a scheme for reallocating bandwidth in path-oriented transport networks based on, for example, MPLS, ATM, SDH or SONET. At specified time points, bandwidth can be switched to those routes that (possibly temporarily) value it most highly. This is done entirely according to local rules and without centralised control.

We present a simulation model of the bandwidth market and evaluate how capacity re-allocation deals with the failure of a single link. We consider a protection switching recovery model which works with pre-established reserve-on-demand recovery routes. A reserve-on-demand recovery route allocates required resources after a failure on the working path has been detected. The capacity re-allocation scheme is used to distribute capacity between the working routes before and after a failure, and between the working and recovery routes during the recovery phase.

To support the efficacy of such a system, we present a simulation model of a 12-node 14-link network. The simulation study reveals that bandwidth reallocation allows rapid recovery after the failure of a single link. Traffic is re-started on a recovery route within one round trip time and the recovery path attains the same Gos as its working counterpart within 1 call holding time.

## I. INTRODUCTION

In this paper we employ a distributed scheme for bandwidth reallocation [1] that can be used in the context of any path-oriented network in which relatively long-lived paths are used to provision resources for connections or flows, whose average holding times are much less than the path lifetimes. Possible environments in which such a model could be useful are Multi-Protocol Label Switching (MPLS) networks in which Label Switched Paths (LSPs) act as the long-lived paths or Asynchronous Transfer Mode (ATM) networks, in which Virtual Paths (VPs) act as the long-lived paths. Other examples include the Synchronous Digital Hierarchy (SDH) or the Synchronous Optical Network (SONET). Note that these concepts apply to both circuit-switching and packet-switching.

We shall use terminology that does not suggest any of the specific environments mentioned above. Thus, we shall call a long-lived path a *route*. Routes traverse one or more physical links. In this paper we restrict ourselves to connection-oriented traffic with admission controls. Hence we use the terms *connections* and *lost connections*. The number of connections that can be simultaneously carried on a route depends on the

amount of bandwidth allocated to the route. However, at any point in time, it is possible that the connections in service on one route are using only a small proportion of the bandwidth allocated to that route, while the bandwidth on another route is heavily utilised. In such a case, it makes sense to transfer bandwidth from the first route to the second one, if that is possible.

A systematic way of doing this is to employ a scheme in which each route places a value [2] on bandwidth, dependent on its current bandwidth and its current occupancy. Then bandwidth can be transferred from routes that place a low value on bandwidth to routes that place a high value on bandwidth. A broker referred to as *bandwidth manager* is assigned to each route. The manager uses knowledge of the route's current occupancy to calculate the value of an extra unit of bandwidth (the "buying price") and also the value that the route would lose should it give up a unit of bandwidth (the "selling price").

We shall view routes as being of two types: *direct* routes, which traverse just a single physical link, and *transit* routes, which traverse more than one physical link. We assume that each link supports a direct route. The direct routes on the links of a transit route are referred to as its *constituent* direct routes.

Bandwidth reallocation is driven by the managers of transit routes, and must take place between the transit routes and their constituent direct routes. In this way the managers are autonomous, act without centralized control from a system coordinator and behave entirely according to local rules.

Buying and selling prices are communicated via an in-band signalling mechanism. Specifically, signals or control packets are sent at certain intervals along each transit route, recording the buying and selling prices of the constituent direct routes. If the sum of the direct route buying prices is greater than the transit route selling price, then the transit route gives up a unit of bandwidth, which is taken up by each of the direct routes. Alternatively, if the sum of the direct route selling prices is less than the transit route buying price, then each of the direct routes gives up a unit of bandwidth, which is taken up by the transit route.

The bandwidth managers are thus aware of local resource demands and bandwidth prices, and reallocate bandwidth among themselves in order to maintain the performance of their routes. Such a scheme is distributed and scalable.

## II. THE NETWORK MODEL

In this Section we describe a model of a fictitious 12-node 14-link network based on the geography of South Africa. The

A.E. Krzesinski and K.E. Müller, Department of Computer Science, University of Stellenbosch, 7600 Stellenbosch, South Africa. Email: aek1@cs.sun.ac.za.

This work was supported by grant numbers 2054027 and 2677 from the South African National Research Foundation, Siemens Telecommunications and Telkom SA Limited.

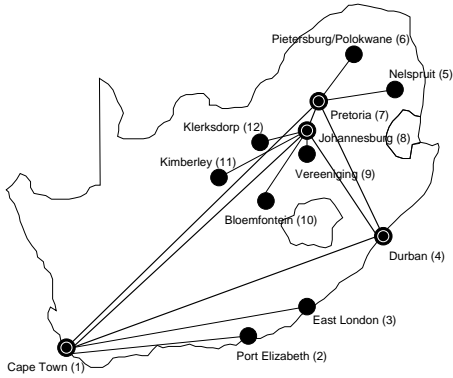


Fig. 1. The network model.

network is illustrated in Fig. 1.

### A. The network model

Each O-D pair is connected by a single shortest-path (minimum geometric distance) route. Each route  $r$  is initially configured with sufficient capacity  $C_r$  to carry its offered traffic to a GoS of 2%. The offered traffic is constructed as follows.

Consider subscribers in a city  $i \in \mathcal{N}$  where  $\mathcal{N}$  denotes the set of cities. Each city serves as a concentrator for access networks in the province in which the city is located. Let  $m_i$  and  $M_i$  denote respectively the population of the municipality and the province in which city  $i$  is located.

Let  $q_i = m_i M_i / \sum_{k \in \mathcal{N}} m_k$  denote the effective population of city  $i$ . The effective population of a city is thus proportional to the population of the province in which the city is located, weighted by a factor that depends on the relative size of the population of the municipality in which the city is located. The municipal weighting factor is meant to represent the traffic generating potential of the city. In addition, the provinces of Gauteng and the Western Cape are allocated a supplementary weighting of two: the remaining seven provinces are allocated a supplementary weighting of one.

Let  $Q_{i,j} = q_j / \sum_{k \in \{\mathcal{N}-i\}} q_k$  denote the fraction of the calls originating in city  $i$  that are destined for city  $j$  where  $j \neq i$ . We assume that a busy hour subscriber generates calls sampled from a normal distribution with mean  $a$  erlangs and standard deviation  $b$  erlangs. Calls between city  $i$  and city  $j$  are assumed to arrive in a Poisson process with a rate sampled from a normal distribution with mean  $a(M_i Q_{i,j} + M_j Q_{j,i})$  and variance  $b^2(M_i Q_{i,j} + M_j Q_{j,i})$ . We choose  $a = 0.01$  erlangs and  $b = 0.005$  erlangs.

Having estimated the offered loads to each route, the route capacities are initially chosen so that, without reconfiguration, each route has a loss probability of 0.02 (2%). The bandwidth  $B_\ell$  of link  $\ell$  is then given by

$$B_\ell = \sum_{r \in \mathcal{A}_\ell} C_r \quad (1)$$

where  $\mathcal{A}_\ell$  is the set of routes that use link  $\ell$ .

In order to compute multiple replications of a simulation in reasonable time on a single processor, the subscriber populations are scaled by a factor of 0.5 which reduces the connection arrival rates and the link capacities by a factor of 2. This scaled model allows  $10^7$  connection completions to be simulated in 3 minutes on an AMD Opteron 248 processor.

The above method of assigning values to the parameters yields a network model where the largest link is Pretoria–Johannesburg with 13118 units of bandwidth carrying 15 routes. The smallest link is Johannesburg–Klerksdorp with 277 units of bandwidth carrying 11 routes. The average link bandwidth is  $5138 \pm 4196$  bandwidth units. The links that carry the most routes are Cape Town–Johannesburg and Pretoria–Johannesburg with 15 routes each. The link Cape Town–Durban carries only 3 routes. Each link carries an average of  $10 \pm 4$  routes.

Table I lists the top ten links showing the link capacities and the number of routes carried by each link.

TABLE I  
TOP 10 LINKS FOR THE NETWORK MODEL.

Link	Capacity	Routes
Pretoria-Johannesburg	13118	15
Cape Town-Johannesburg	12017	15
Durban-Johannesburg	9577	5
Johannesburg-Vereeniging	8162	11
Cape Town-Pretoria	6199	9
Durban-Pretoria	4934	3
Cape Town-Port Elizabeth	4671	11
Cape Town-Durban	4515	3
Cape Town-East London	3314	11
Pietersburg-Pretoria	2020	11

### B. The reallocation parameters

In order for the capacity reallocation scheme to be efficient, the rate of reallocation attempts should be significantly lower than the rate of connection arrivals. If this were not the case, then it would be more efficient to use call-by-call signalling to set up a route with sufficient bandwidth to carry the connection. The signalling rate  $\eta_r$  on transit route  $r$  is therefore assumed to be a fraction of the connection arrival rate on route  $r$ . Thus

$$\eta_r = V \lambda_r$$

where  $V \in [0, 1]$  is the *signalling ratio*.

The calculation of a network-wide planning horizon would either involve the use of global data (which would not scale) or requires the route managers to agree on a common value. We therefore work with per-route planning horizons  $\tau_r$  that are assumed to be a multiple of the average reallocation interval  $1/\eta_r$ . Thus

$$\tau_r = P/\eta_r$$

where  $P > 0$  is the *planning ratio*.

If a small unit of bandwidth  $U$  is traded, then very frequent trading might be required to keep up with the changes in bandwidth occupancy. This situation should be avoided in

order to keep the reallocation overhead low. On the other hand, if a large unit of bandwidth is traded, bandwidth distribution will be coarser. This situation should also be avoided in order to keep the carried traffic high.

In the experiments presented in section IV below we have set the signalling ratio  $V = 0.1$ , the planning ratio  $P = 1$  and the reallocation unit  $U = 4$ . The size of each signalling packet is 104 bytes: a 40 byte header and a 64 byte signalling payload. The choice of these values is motivated in [1].

### III. LINK FAILURE

We will investigate the efficacy of the capacity reallocation scheme in recovering from a failure of a single link.

When a link fails, all the routes that used the failed link fail. Each failed route is replaced by a pre-established least cost (shortest euclidean distance) recovery route which need only be disjoint from its working counterpart at the failed link.

The resources required in the recovery process are bandwidth, buffers and router processing capacity. Our recovery model will only take the bandwidth resource into account.

The network is pre-configured with sufficient backup capacity (see section III-A) to construct equivalent recovery routes which replace the failed routes without degradation of service except for a short period of time while the bandwidth reallocation scheme is allocating bandwidth to the recovery route.

The recovery paths offer 1-to-1 protection: each working path has a dedicated recovery path. The recovery paths offer 1:1 (one for one) protection switching: the backup capacity can be used by the working routes before the failure occurs. The recovery mode is non-revertive: each failed route releases its resources to the network and traffic is not switched back to the working path when the working path is restored to service. When the working path is restored, it becomes the recovery path.

#### A. Configuring the direct routes with sufficient backup capacity to construct equivalent recovery routes

The direct routes must have sufficient capacity so that the recovery routes can acquire capacity from their constituent direct routes and meet their GoS constraints (2%) after the link failure.

Let  $x$  denote the link which is going to fail. Let  $\mathcal{A}_x$  denote the set of routes that use link  $x$ . For each route  $r \in \mathcal{A}_x$  a recovery route (which is the shortest route, in terms of euclidean distance, which does not use link  $x$ ) is pre-computed. Let  $\mathcal{B}_x$  denote the set of recovery routes. Let  $\mathcal{R}$  denote the set of routes used by the network. After the failure  $\mathcal{R} = (\mathcal{R} \setminus \mathcal{A}_x) \cup \mathcal{B}_x$ .

Let  $C_r$  denote the capacity needed by the recovery route  $r \in \mathcal{B}_x$  in order to meet its target GoS. The extra capacity needed on direct route  $i$  is thus

$$C_i^+ = \sum_{r \in \mathcal{B}_x: i \in \mathcal{D}_r} C_r.$$

When link  $x$  fails, the capacity used by each failed route  $r \in \mathcal{A}_x$  is returned to its constituent direct routes. The capacity returned to direct route  $i$  is thus

$$C_i^- = \sum_{r \in \mathcal{A}_x: i \in \mathcal{D}_r} C_r$$

The direct routes are configured with sufficient capacity to construct equivalent recovery routes by assigning a capacity

$$C_i + \max(0, C_i^+ - C_i^-)$$

to each direct route  $i \in \mathcal{D}$ .

#### B. Simulating the failure of a link

When a link fails, the capacity used by the failed routes is returned to their constituent direct routes. Each failed route is replaced by a pre-computed recovery route. The calls in progress on the failed routes are re-started on the recovery routes. The calls are either re-started immediately or after a random delay.

The signalling rate  $\eta_r$  and the planning horizon  $\tau_r$  for each recovery route are recomputed after a failure. The signalling packets on failed routes are discarded and new signalling packets are immediately scheduled on the recovery routes. Traffic starts flowing on a recovery route when the first reallocation signal returns to the PSL (path switch LSR) which will occur within one round trip time.

Initially no capacity is allocated to the recovery routes. The capacity re-allocation process determines the amount of capacity to allocate to each recovery route.

#### C. Quantifying the effect of a link failure

Let  $N_{i,r}$  and  $L_{i,r}$  denote respectively the total number of call arrivals and the total number of lost calls on route  $r$  in the period  $[0, t_i]$ . The loss probability  $P_{i,r}$  on route  $r$  during the interval  $[t_{i-1}, t_i]$  is given by

$$P_{i,r} = (L_{i,r} - L_{i-1,r}) / (L_{i,r} - N_{i-1,r}).$$

The average network-wide loss probability during the interval  $[t_{i-1}, t_i]$  is given by

$$\sum_{r \in \mathcal{R}} \lambda_r P_{i,r} / \sum_{r \in \mathcal{R}} \lambda_r \quad (2)$$

where  $\lambda_r$  denotes the average call arrival rate to route  $r$ .

The average loss probability on the recovery routes during the interval  $[t_{i-1}, t_i]$  is given by

$$\sum_{r \in \mathcal{B}_x} \lambda_r P_{i,r} / \sum_{r \in \mathcal{B}_x} \lambda_r. \quad (3)$$

The average loss probability on the surviving routes during the interval  $[t_{i-1}, t_i]$  is given by

$$\sum_{r \in \mathcal{R} \setminus \mathcal{B}_x} \lambda_r P_{i,r} / \sum_{r \in \mathcal{R} \setminus \mathcal{B}_x} \lambda_r. \quad (4)$$

The recovery time – the time taken for a recovery route to be activated and to have traffic flowing – is one round trip

time. The recovery cycle (the full restoration time) – the time until the recovery path can offer the same GoS as its working counterpart – is presented in the following section.

#### D. The recovery phase

Throughout the simulation run, a record is written to the simulation trace file after every  $\Delta$  call arrivals. Each trace record logs the simulation time  $t_i$  and the values of the loss probabilities (2), (3) and (4).

Let  $T_0$  denote the time at which link  $x$  fails. At time  $T_0$ , the simulator enters the recovery phase. During the recovery phase a record is written to simulation trace at times  $T_1, T_2, \dots, T_K$  where  $T_i - T_{i-1} \leq \Delta/10$ . The simulator exits the recovery phase at time  $T_K$  when the network target GoS has been restored. The calculation of the  $T_1, T_2, \dots, T_K$  is explained below.

Let  $N_0$  and  $L_0$  denote respectively the total number of call arrivals and the total number of lost calls during the period  $[0, T_0]$  prior to the link failure at time  $T_0$ .

The recovery phase starts at time  $T_0$  at which time a record is written to the simulation trace file. Let  $T > T_i$  denote an instant during the recovery phase. Let  $N_i$  and  $L_i$  denote respectively the total number of call arrivals and the total number of lost calls in the period  $[0, T_i]$ . Let  $N$  and  $L$  denote respectively the total number of call arrivals and the total number of lost calls during the period  $[0, T]$ . Let  $e_{i+1}$  denote the event

$$((N - N_i) > \Delta) \mid ((L - L_i) > \Delta G)$$

where  $G$  denotes the network target GoS (2%).

Define an indicator function

$$f(e_i) = \begin{cases} 1 & e_i \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

A record is written to the trace at time  $T$  when  $f(e_i) = 1$  and  $T_i$  is set equal to  $T$ . Let  $E_i$  denote the event

$$L_i - L_{i-1} \leq \Delta G.$$

Define an indicator function

$$F(E_i) = \begin{cases} 1 & E_i \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

The recovery phase starts at time  $T_0$  when the link fails and ends at time  $T_K$  when

$$\prod_{k=K-\kappa+1}^K F(E_k) = 1.$$

The recovery phase thus continues until  $\kappa$  consecutive trace records have been written. The  $k$ th such record where  $k = 0, \dots, \kappa - 1$  describes the interval  $[T_{K-k-1}, T_{K-k}]$ . During each of these intervals  $(N_{K-k-1} - N_{K-k}) \leq \Delta$  calls arrive and the network target GoS is met:  $(L_{K-k-1} - L_{K-k}) \leq \Delta G$  calls are lost.

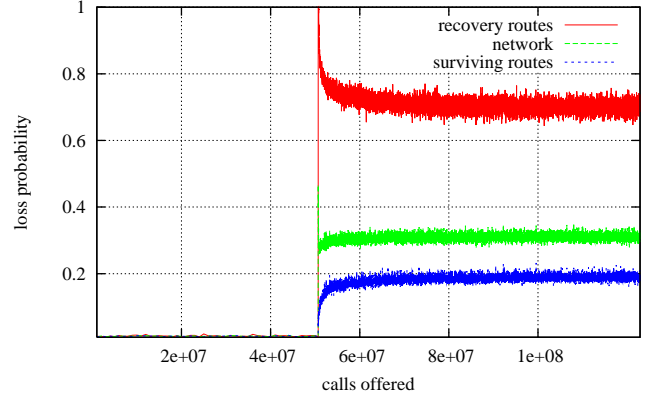


Fig. 2. The link from Pretoria to Johannesburg fails, no backup capacity

The recovery cycle is the time until the recovery path can offer the same GoS as its working counterpart. Let

$$\delta = \min_k \left( \prod_{i=k}^K F(E_i) = 1 \right).$$

Thus  $T_\delta$  is the earliest instant in the recovery phase when the network loss probability meets the target GoS for the remainder of the phase.

#### IV. SIMULATION RESULTS

Each simulation processed 100,000,000 call completions which takes some 3 hours on an AMD Opteron 248 processor. Records are written to the simulation trace after every  $\Delta = 10^6$  call arrivals. The link from Pretoria to Johannesburg carrying 15 routes is failed after 50,000,000 calls have completed. Some 55,500 calls are in progress when the link fails: about 13,000 of these calls are in progress on the failed routes and are re-started on the recovery routes. The following recovery scenarios are investigated

- 1) no backup capacity is available,
- 2) backup capacity is available,
- 3) the signalling ratio  $V$  is increased during the recovery phase,
- 4) the failed calls are re-started almost immediately, and
- 5) the failed calls are re-started after a random delay.

##### *Backup capacity is not available*

Fig. 2 presents the loss probabilities as calculated using Eqn. (2), (3) and (4) when no backup capacity is assigned to the direct routes. The network is clearly unable to recover after the link fails.

##### *Backup capacity is available*

The algorithm presented in section III-A is next used to assign sufficient backup capacity to the direct routes so that the network can meet its target GoS when the link from Pretoria to Johannesburg fails. The algorithm increased the capacities of 2 of the 14 direct routes namely Durban–Johannesburg and Durban–Pretoria, increasing the total network capacity by

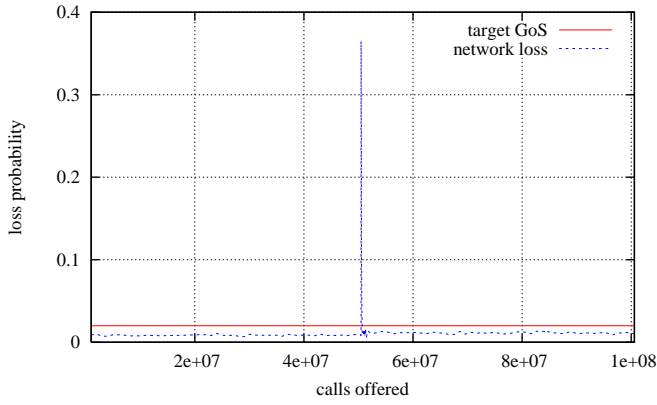


Fig. 3. The link from Pretoria to Johannesburg fails, backup capacity is available.

37%. The backup capacity is available for use both before and after the link failure occurs.

Fig. 3 presents the network loss probability (2) when the backup capacity is available. The network is now able to meet its target GoS after the failure and the network performance before and after the link failure is virtually identical. However, during the recovery cycle the network loss probability greatly exceeds the target GoS.

#### The signalling ratio is increased during the recovery phase

Fig. 4 investigates the impact of the reallocation parameters during the recovery phase. During the recovery phase, messages are written to the simulation trace at intervals not exceeding 100,000 call arrivals. The recovery phase ends after 10 successive records have been written to the trace where less than 2,000 calls were lost during each inter-record interval thus meeting the target GoS during each interval.

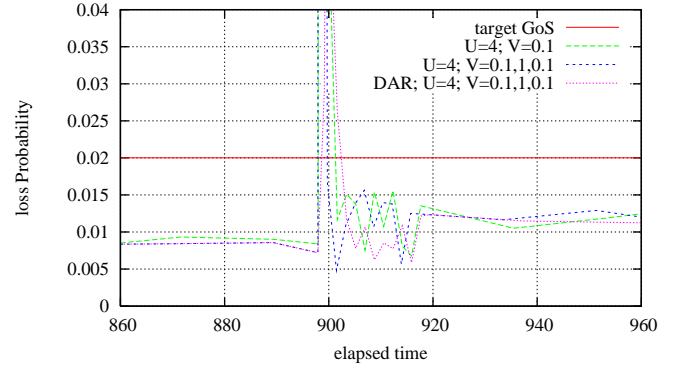
The recovery phase lasts about 5 call holding times. Fig. 4 shows that the impact of the link failure is severe if the signalling ratio  $V$  remains fixed at its standard value of 0.1 during the recovery phase (note that network loss probability Fig. 4(a) and the recovery route loss probability Fig. 4(b) go off-scale). If the signalling ratio  $V$  is increased to 1 during the recovery phase then the network loss probability and the loss probability on the surviving routes almost meet the target GoS.

#### Delayed re-start

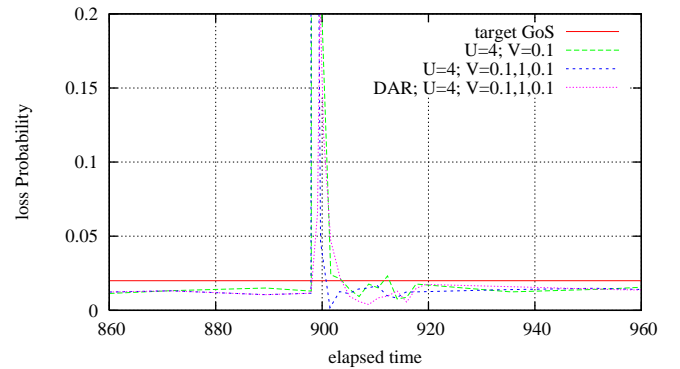
We finally investigate the impact of the re-started calls on the recovery cycle.

Events in the simulator occur at three timescales. Calls complete on route  $r$  at the instants of a Poisson process with parameter  $\mu_r$ . By convention,  $\mu_r = 1$ . Calls arrive to route  $r$  at the instants of a Poisson process with parameter  $\lambda_r$ . Reallocations are attempted on route  $r$  at the instants of a Poisson process with parameter  $\eta_r$ . The signalling rate  $\eta_r = 0.1\lambda_r$ .

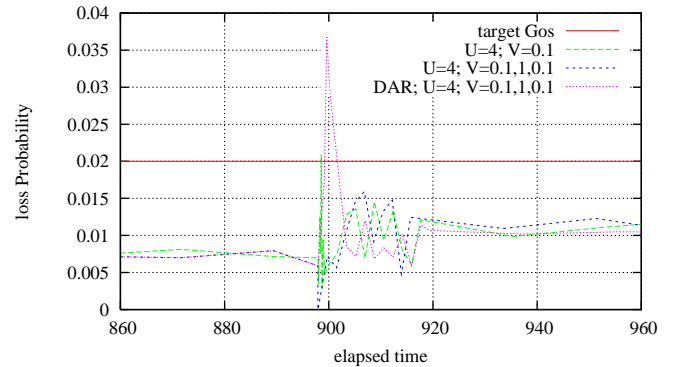
Table II and Fig 5 investigate the impact of the delay in re-starting the failed calls upon the performance of the simulated



(a) The network loss probability



(b) The loss probability on the recovery routes



(c) The loss probability on the surviving routes

Fig. 4. The recovery time when the link from Pretoria to Johannesburg fails.

TABLE II  
NETWORK PERFORMANCE DURING THE RECOVERY CYCLE

re-start delay	$\epsilon$ $T_\delta - T_0$	calls re-started	calls lost during $\epsilon$	loss prob during $\epsilon$
$\exp(\mu_r)$	3.0	12819	5424	1.9%
$\exp(\eta_r)$	1.2	12819	12813	11.0%
$\exp(\lambda_r)$	1.2	12819	13944	12.0%

network during the recovery cycle. Note that a delay in re-starting failed calls implies the existence of a call queuing mechanism at the ingress routers.

Three delay models are investigated: the failed calls are re-started on route  $r$  after a random delay sampled from an exponential distribution with parameter (1)  $\mu_r$ , (2)  $\eta_r$ , and (3)  $\lambda_r$ .

Model (1) imposes a substantial delay in re-starting the failed calls. This to an extent prevents the re-started calls and the calls which arrive during the recovery cycle from combining to overload the network. The disadvantage however is that the network requires 5 call holding times to return to its target GoS.

Models (2) and (3) re-start the failed calls almost immediately. Many calls are lost, but on the other hand the network returns to its target GoS within 2 call holding times. This illustrates a tradeoff between rapid recovery and the number of calls lost during the recovery cycle.

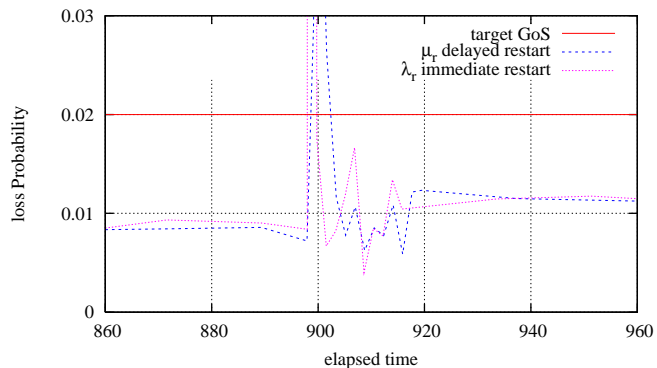
## V. CONCLUSIONS

This paper presents a scheme for bandwidth reallocation in a path-oriented transport network. A bandwidth manager is assigned to each route. The managers are autonomous, acting without centralized control from a system coordinator and behave entirely according to local rules. The managers are aware of local resource demands and bandwidth prices. The managers trade bandwidth among themselves in order to maintain the Quality of Service (QoS) of their routes.

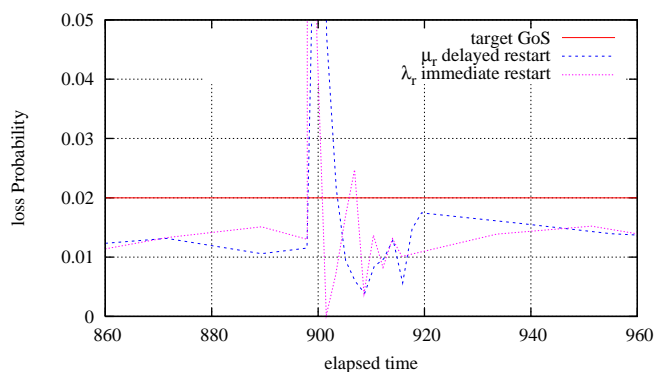
We present a simulation model of the bandwidth market. Initial studies of a small 12-node network model reveal that bandwidth reallocation can be used to efficiently and rapidly distribute the available resources between the working and recovery routes after a failure.

## REFERENCES

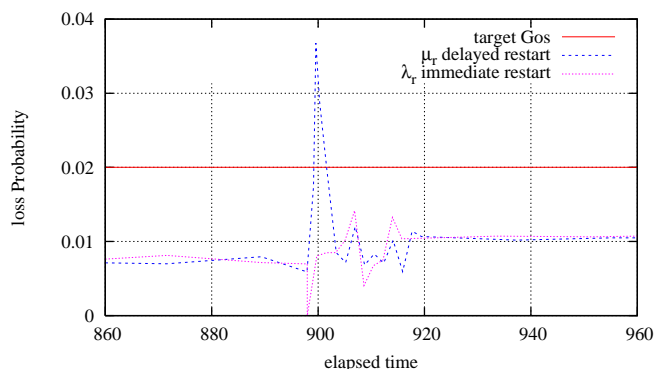
- [1] Å. Arvidsson, B.A. Chiera, A.E. Krzesinski and P.G. Taylor. "A Distributed Scheme for Value-Based Bandwidth Re-Configuration", submitted (2005).
- [2] B.A. Chiera and P.G. Taylor, "What is a Unit of Capacity Worth?", *Probability in the Engineering and Informational Sciences*, vol. 16, pp. 513–522, 2002.
- [3] V. Sharma and F. Hellstrand (Eds). "RFC 3469: Framework for Multi-Protocol Label Switching (MPLS)-based Recovery", February 2003.
- [4] P. Pan, G. Swallow, A. Atlas. "Fast Reroute Extensions to RSVP-TE for LSP Tunnels". Internet Draft draft-ietf-mpls-rsvp-lsp-fastreroute-07, Sept 2004.
- [5] R.J. Gibbens and F.P. Kelly, "Dynamic routing in fully connected networks", *IMA Journal of Mathematical Control and Information*, vol. 7, pp. 77–111, 1990.
- [6] J.M. Akinpelu, "The Overload Performance of Engineered Networks with Nonhierarchical and Hierarchical Routing", *AT&T Bell Labs Technical Journal*, vol. 63, pp. 1261–1281, 1984.



(a) The network loss probability



(b) The loss probability on the recovery routes



(c) The loss probability on the surviving routes

Fig. 5. The recovery time when the link from Pretoria to Johannesburg fails: delayed versus immediate re-start on the recovery routes.