Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels using Aggregated Link Usage Information

Murali Kodialam and T.V.Lakshman

IEEE Infocom 2001

Contents

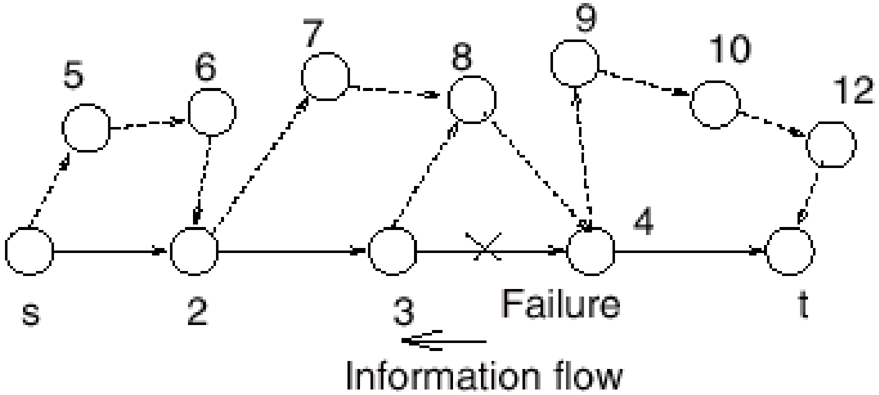
Introduction

- Path restoration
  - Allow backup paths to redirect traffic upon failure detection
  - Route both an active path and a backup path
  - Entail propagation delay of failure information to the source node
- Local restoration
  - A new algorithm is proposed to set up locally restorable bandwidth guaranteed tunnels

Restoration Options

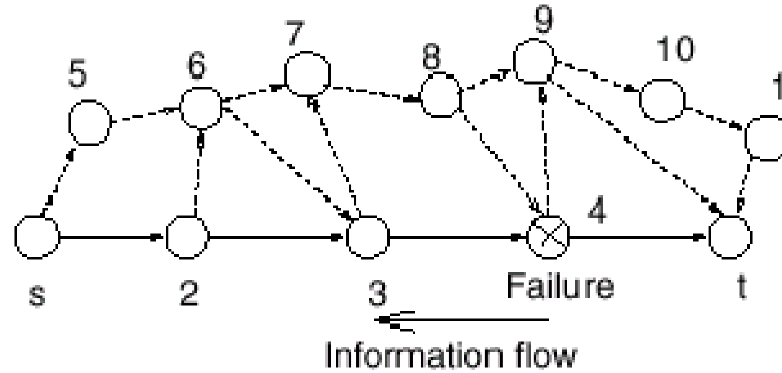
- Two failure modes are considered
  - Single link failures /single element failures
- Capacity in backup path can be shared
  - Inter-demand sharing /intra-demand sharing
- Three scenarios considered
  - No information case
  - Complete information case
  - Partial information case

Failure Modes



Primary Arc	Backup Path
s-2	s-5-6-2
2-3	2-7-8-4
3-4	3-8-4
4-t	4-9-10-12-t

Fig. 4. Forward and Backup Path for Single Link Failure



	Backup Path
Node 2	s-5-6-3
Node 3	2-6-7-8-4
Node 4	3-7-8-9-t
Link 4-t	4-9-10-12-t

Fig. 5. Forward and Backup Path for Single Element Failure

Problem Definition

- Notations
  - A network of n nodes and m links, all links are assumed directional
  - (o_k, t_k, b_k): request for tunnel k
  - A_{ij}: set of demands using link (i, j) in active paths
  - B_{ij}: set of demands using link (i, j) for backup
  - $F_{ij} = \sum_{k \in A_{ij}} b_k$, $G_{ij} = \sum_{k \in B_{ij}} b_k$, $R_{ij} = C_{ij} - F_{ij} - G_{ij}$
- Objective: Minimize sum of bandwidths that is used by the active and the backup paths

Key Design Ideas

- Cost of backup paths can be determined by solving shortest path problems, one for each link
- It is necessary to execute shortest path (Dijkstra) algorithm backwards starting at the sink
- Maintaining m-vector at each node that gives us the amount of bandwidth reserved for current demand can account for intra-demand sharing
- Modifying cost of links in computation of backup costs can account for node failures

Cost Function

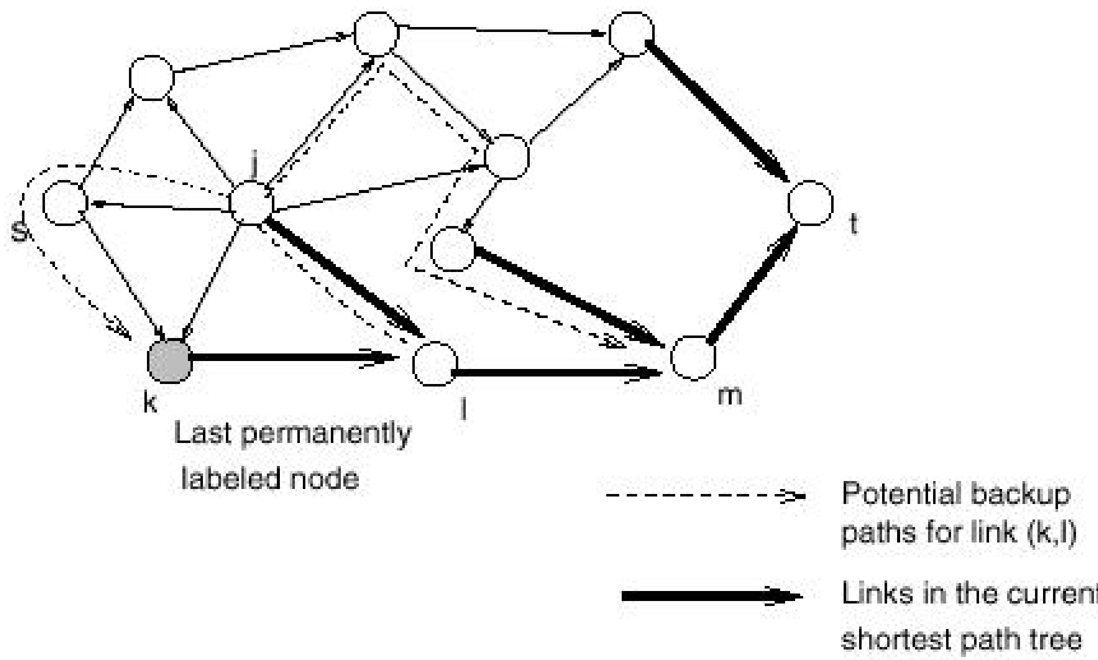
- Cost of using link (*u, v*) on the backup path if link (*i, j*) is used in the active path

complete information case

$$\theta_{ij}^{uv} = \begin{cases} 0 & \text{if } \delta_{ij}^{uv} + b \le G_{uv} \\ & \text{ and } (i,j) \ne (u,v) \\ \delta_{ij}^{uv} + b - G_{uv} & \text{if } \delta_{ij}^{uv} + b > G_{uv} \text{ and} \\ & R_{uv} \ge \delta_{ij}^{uv} + b - G_{uv} \\ & \text{and } (i,j) \ne (u,v) \\ \infty & \text{ Otherwise} \end{cases}$$

partial information case

$$\theta_{ij}^{uv} = \begin{cases} 0 & \text{if } F_{ij} + b \le G_{uv} \text{ and} \\ & (i,j) \ne (u,v) \\ F_{ij} + b - G_{uv} & \text{if } F_{ij} + b > G_{uv} \text{ and } R_{uv} \ge \\ & F_{ij} + b - G_{uv} \text{ and } (i,j) \ne (u,v) \\ \infty & \text{Otherwise} \end{cases}$$


j
s
t
k
l
m
Last permanently
labeled node
Potential backup
paths for link (k,l)
Links in the current
shortest path tree

λ_{uij} : amount of bandwidth reserved by current demand for all the backup paths for all the links leading from node *u* to the destination

$$\kappa_{mn} = F_{kj} + b - B_{mn} - \lambda^{j}_{mn}$$

$$l_{mn} = \begin{cases} 0 & \text{if } \kappa_{mn} \le 0 \\ \delta_{mn} & \text{if } 0 \le \kappa_{mn} \le b \text{ and } R_{mn} \ge \kappa_{mn} \text{ and} \\ & \ (m,n) \ne (k,j) \\ \infty & \text{Otherwise} \end{cases}$$

Modified Cost Function

- Consider single node failure case

$$\theta_{ij}^{uv} = \begin{cases} 0 & \text{if } \sum_{(j,k)\in E} \delta_{jk}^{uv} + b \leq \\ & G_{uv} \text{ and } (i,j) \neq (u,v) \\ \sum_{(j,k)\in E} \delta_{jk}^{uv} + b - G_{uv} & \text{if } \sum_{(j,k)\in E} \delta_{ij}^{uv} + b > G_{uv}, \\ & R_{uv} \geq \sum_{(j,k)\in E} \delta_{jk}^{uv} \\ & +b - G_{uv} \text{ and} \\ & (i,j) \neq (u,v) \\ \infty & \text{Otherwise} \end{cases}$$

$$\theta_{ij}^{uv} = \begin{cases} 0 & \text{if } \sum_{(j,k)\in E} F_{jk} + b \leq G_{uv} \\ & \text{and } (i,j) \neq (u,v) \\ \sum_{(j,k)\in E} F_{jk} + b - G_{uv} & \text{if } \sum_{(j,k)\in E} F_{jk} + b > G_{uv} \\ & \text{and } R_{uv} \geq F_{ij} + b - G_{uv} \\ & \text{and } (i,j) \neq (u,v) \\ \infty & \text{Otherwise} \end{cases}$$

Formal Algorithm Description

- Algorithm for single link failure case with partial information
  - Use repeated invocations of Dijkstra's algorithm to generate the path
  - Main routine: LOCAL_EDGE_DISJOINT()
  - ALT_PATH_COST(): determine the cost of providing a link with a local backup
  - SHORT_PRED_PATH():

LOCAL_EDGE_DISJOINT(s, t)

- INITIALIZATION

1: Reverse all links in the network.

2: $T = V,\ \ P = \emptyset,\ \ \phi_t = 0,\ \ \phi_j = \infty\ \ \forall j \neq t$

$\lambda^d_{mn} = 0\ \ \forall (m,n) \in E,\ \ Q(t) = \emptyset.$

- ITERATIVE STEP

2: $k = \text{Arg}\min_{j \in T} \phi_j$. If $k = s$ go to Step 6.

3: $T = T \setminus \{k\}$ and $P = P \cup \{k\}$.

4: For each $j \in T$, $(k, j) \in E$

$w_{kj} =$ ALT_PATH_COST(k, j)

if ($\phi_j \geq w_{kj} + \phi_k$)

$\phi_j = \phi_k + w_{kj}$

Q(j) = k

5: Go to Step 2.

- TERMINATION

6: Exit.

ALT_PATH_COST(k, j)

- INITIALIZATION

1: $u = k$, MIN $= \infty$.

- ITERATIVE STEP

2: If $u = \emptyset$ go to Step 6.

3: $\alpha =$ SHORT_PRED_PATH (k, u, j).

4: if $(\alpha \leq$ MIN)

MIN $= \alpha$

$\lambda^j_{mn} = \beta_{mn} \;\; \forall (m, n) \in E$

5: $u = Q(u)$ Go to Step 2.

- TERMINATION

6: Exit.

SHORT_PRED_PATH(k, u, j)

- INITIALIZATION

1: $\delta_{mn} = F_{kj} + b - B_{mn} - \lambda^{u}_{mn} \quad \forall (mn) \in E.$

2:

$$l_{mn} = \begin{cases} 0 & \text{if } \delta_{mn} \leq 0 \\ \delta_{mn} & \text{if } 0 \leq \delta_{mn} \leq b \text{ and } R_{mn} \geq \\ & \delta_{mn} \text{ and } (m,n) \neq (k,j) \\ \infty & \text{Otherwise} \end{cases}$$

3: $T' = V, \quad P' = \emptyset, \quad \gamma_u = 0, \quad \gamma_j = \infty \quad \forall j \neq u$

$\lambda^{d}_{mn} = 0 \quad \forall (m,n) \in E$

- ITERATIVE STEP

4: $w = \mathrm{Arg\,min}_{j \in T}\, \omega_j$. If $w = k$ go to Step 9.

5: $T' = T' \backslash \{w\}$ and $P' = P' \cup \{w\}$.

6: For each $i \in T'$, $(w, i) \in E$

if $(\gamma_i \geq l_{wi} + \gamma_w)$

$\gamma_i = \gamma_w + l_{wi}$

$Q'(i) = w$

7: Go to Step 2.

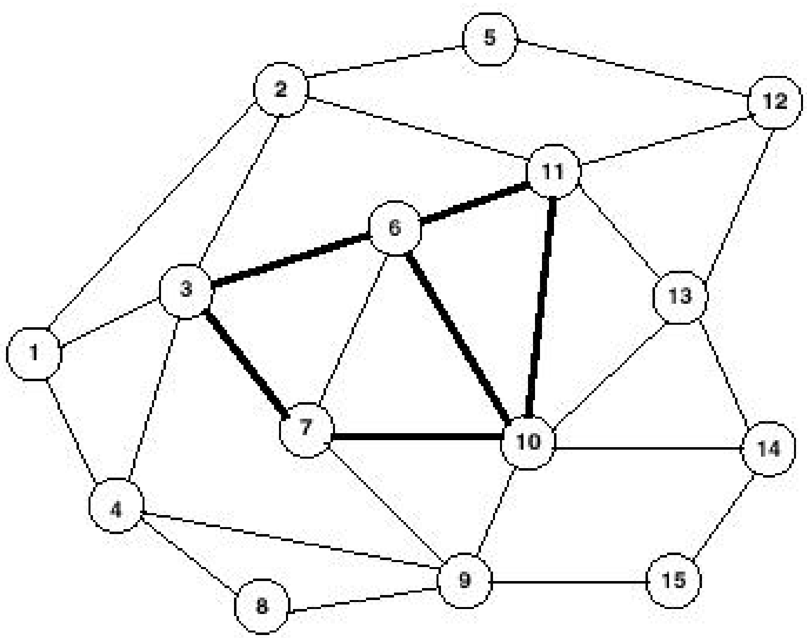
- TERMINATION

8: Set $\beta_{mn} = \lambda^{u}_{mn} \quad \forall (mn) \in E$, if arc (mn) is on the shortest path from u to j set $\beta_{mn} = \lambda^{u}_{mn} + l_{mn}$.

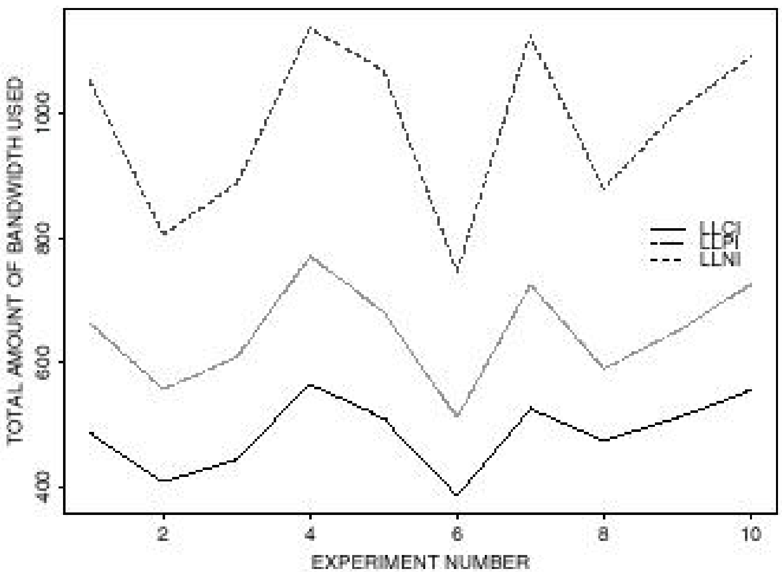
9: Exit.

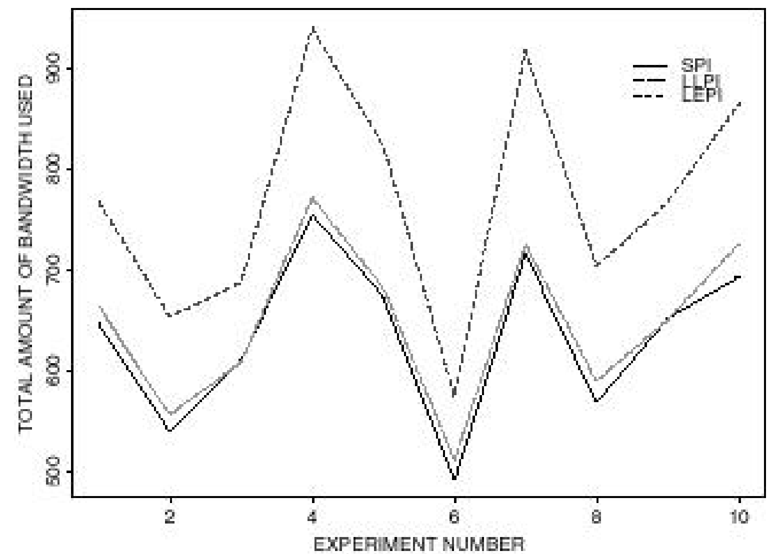
Experimental Results

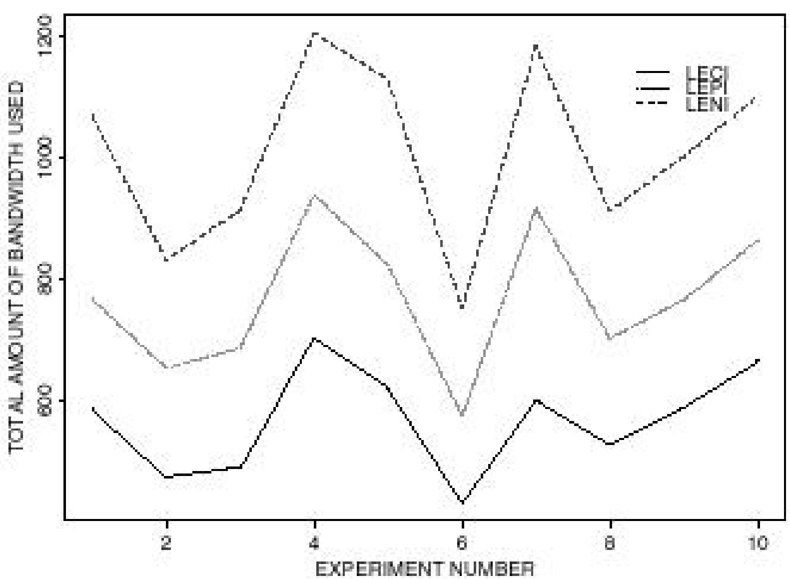
- Test network

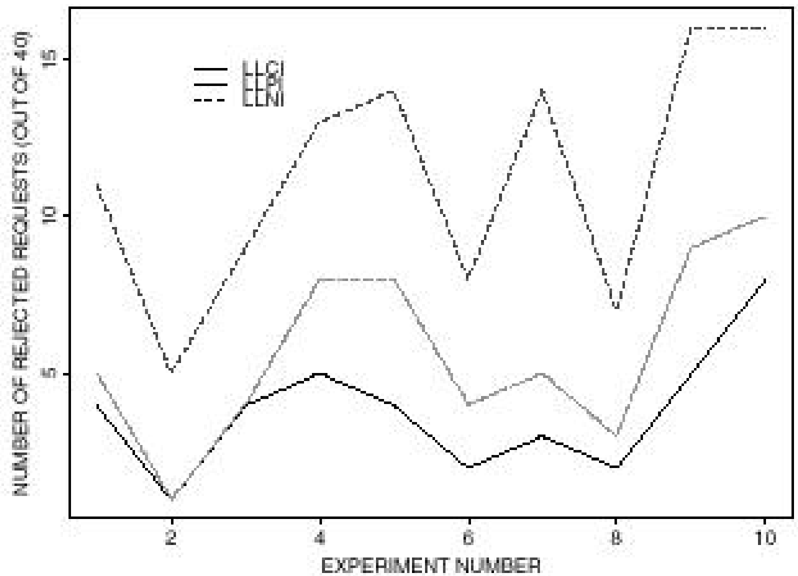


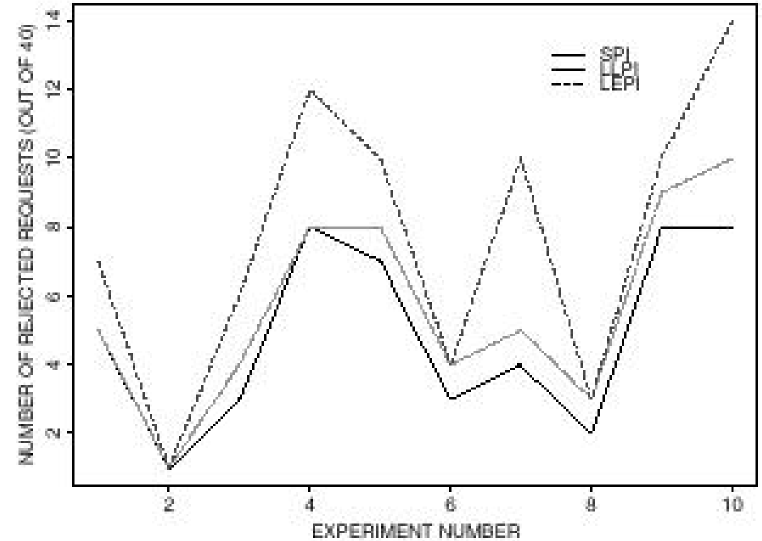
- Study scenarios
  1. SPI
  2. LLNI
  3. LLPI
  4. LLCI
  5. LENI
  6. LEPI
  7. LECI

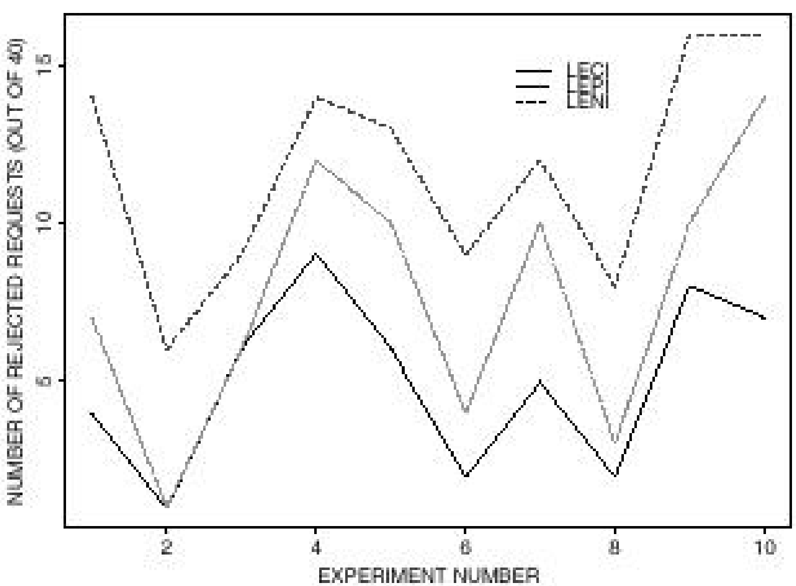

TOTAL AMOUNT OF BANDWIDTH USED
400
600
800
1000
2
4
6
8
10
EXPERIMENT NUMBER
LLCI
LLPI
LLNI


TOTAL AMOUNT OF BANDWIDTH USED
500
600
700
800
900
2
4
6
8
10
EXPERIMENT NUMBER
SPI
LLPI
LEPI


TOTAL AMOUNT OF BANDWIDTH USED
600
800
1000
1200
2
4
6
8
10
EXPERIMENT NUMBER
LECI
LEPI
LENI


LLCI
LLPI
LLNI
NUMBER OF REJECTED REQUESTS (OUT OF 40)
EXPERIMENT NUMBER


SPI
LLPI
LEPI
NUMBER OF REJECTED REQUESTS (OUT OF 40)
EXPERIMENT NUMBER


LECI
LEPI
LENI
NUMBER OF REJECTED REQUESTS (OUT OF 40)
EXPERIMENT NUMBER

Concluding Remarks

- A new algorithm is developed for the on-line routing of a locally restorable bandwidth guaranteed path.
- Bandwidth efficiency is achieved by exploiting the potential for inter-demand and intra-demand backup bandwidth sharing.
- The algorithm performs well in terms of the number of rejected requests and the total bandwidth used.