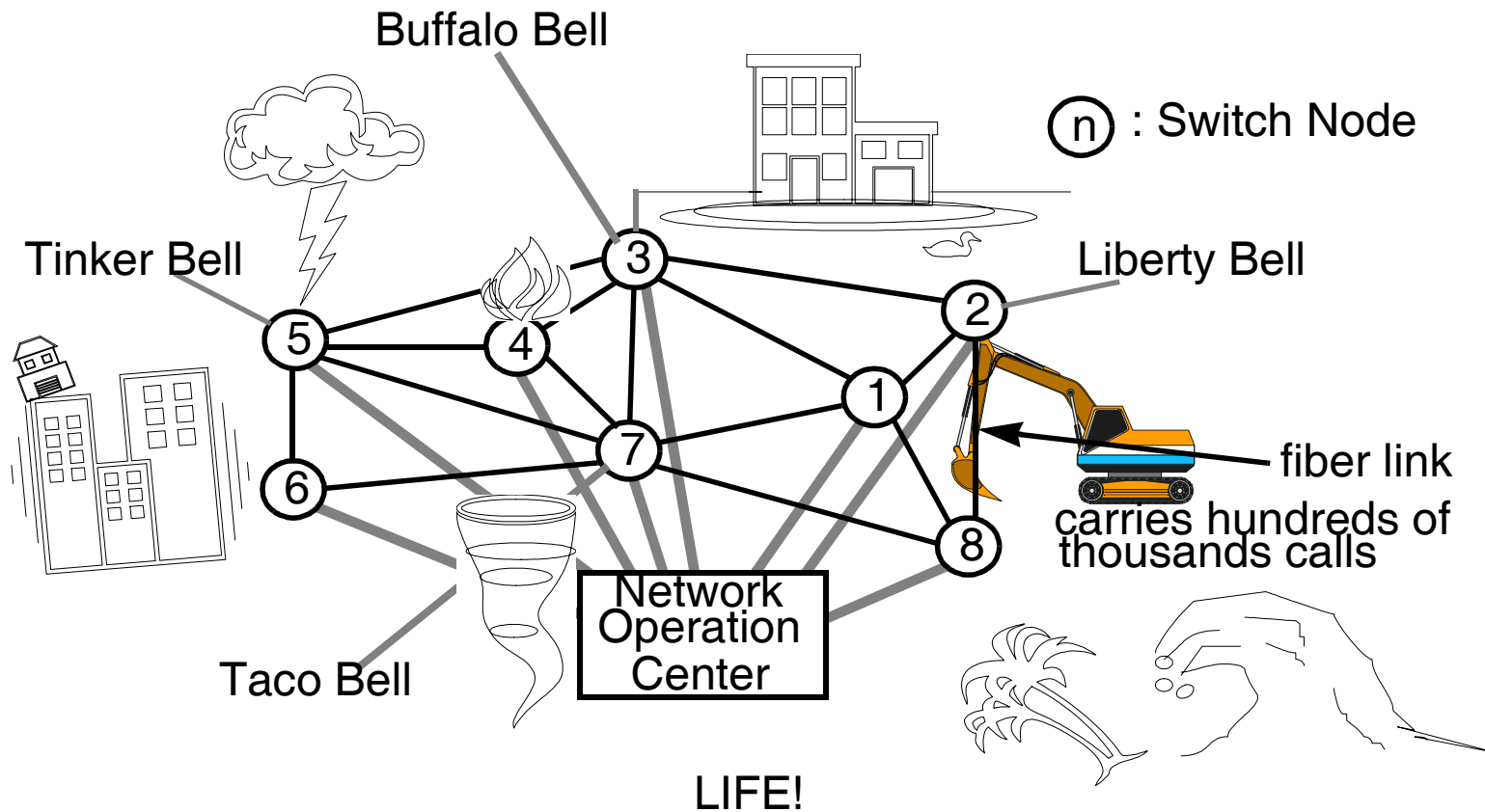




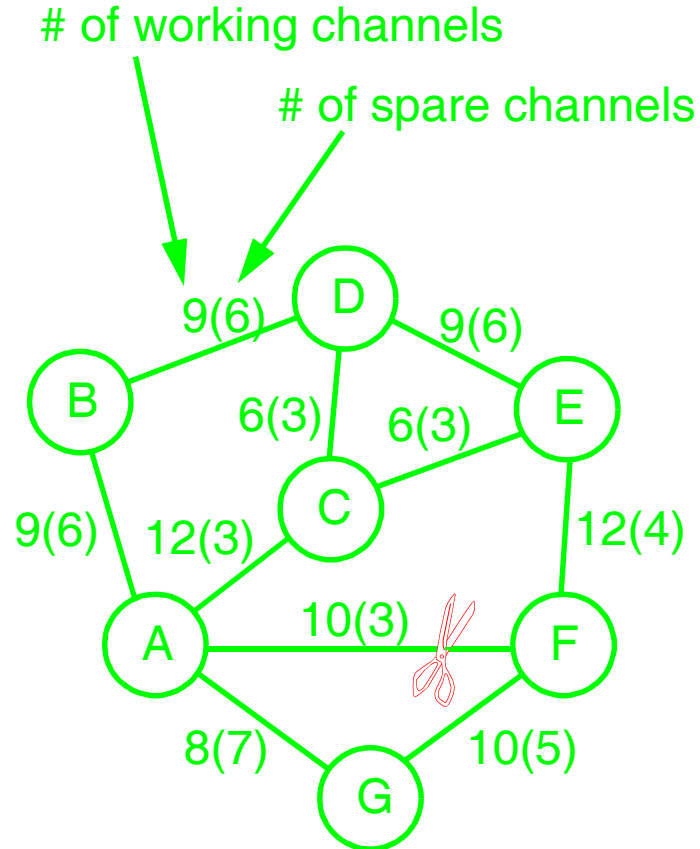
What is the Network Restoration Problem?



- 1 Types of Failures: Channel, Link, Node, Area, and Transient Failures.
- 1 Network Restoration: Process of recovering from network failures.
- 1 Types of Restoration Approaches: Centralized, Distributed, Hierarchical.



Network Model and Performance Metrics



Performance metrics for evaluating network restoration algorithms:

- 1 Time to restoration
- 1 Restoration level
- 1 Spare channel usage

For path-based approach, this is defined to be the no. of spares used in the restoration paths minus the no. of working channels in the disrupted paths that were released

- 1 Message Volume

Transmission delay \leftarrow Message length / Transmission speed

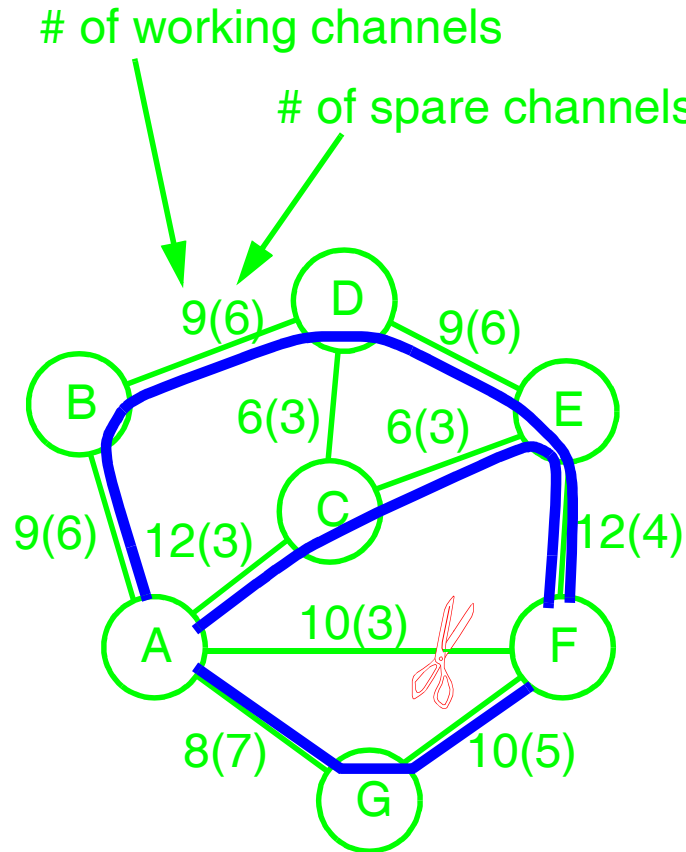
Propagation delay \leftarrow Node distance / Signal propagation speed

Queueing delay \leftarrow Messages in front * Msg processing time

DCS connection delay \leftarrow DCS connection time * # of channels to be connected



Restoration Paths and Spare Channel Usage



For Link A-F failure,
three restoration paths can be found to
restore 9 of the 10 disrupted channels.

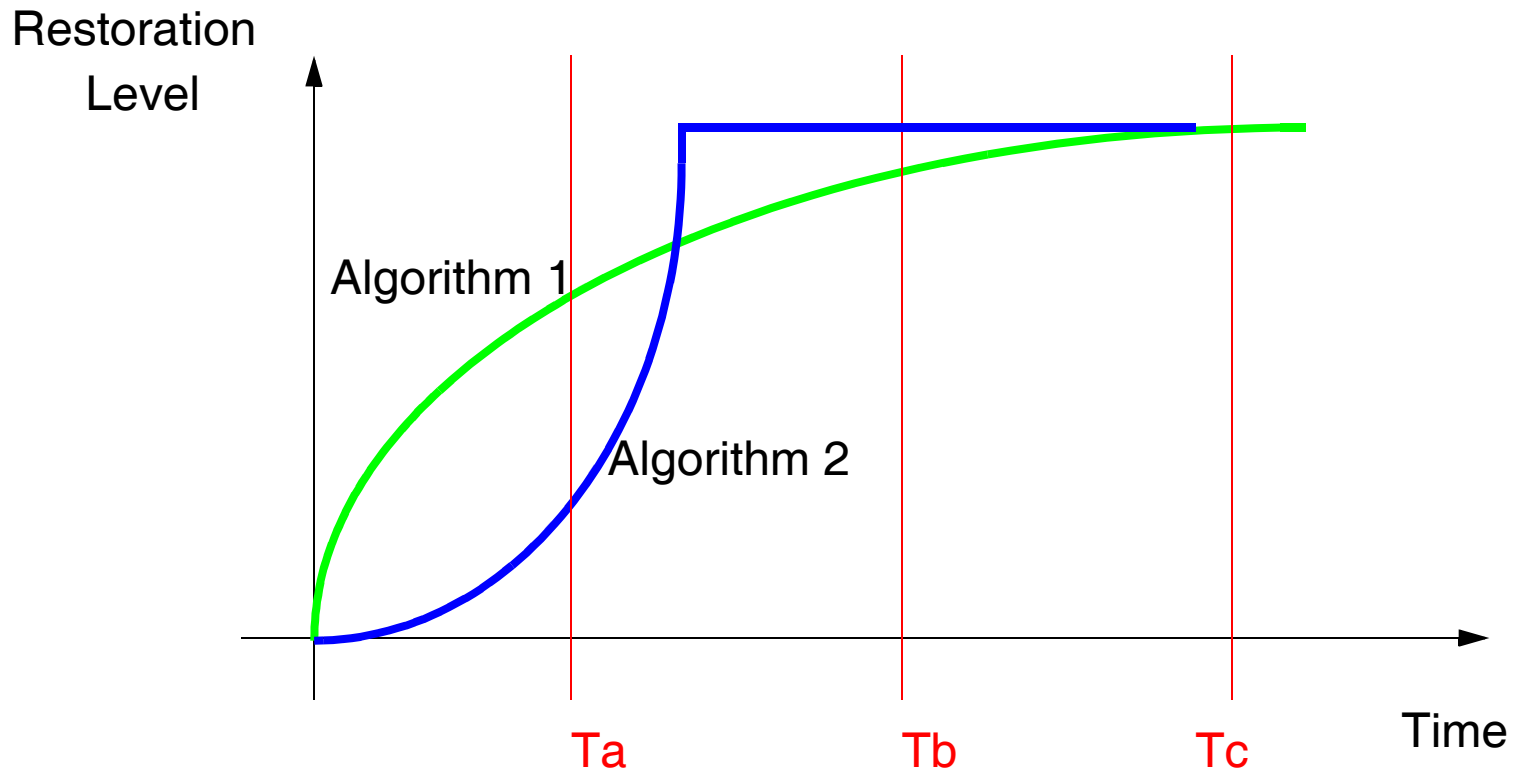
- 1 Restoration Level-90%
- 1 Spare channel usage-23 spares used

An optimal algorithm based on RELAX-IIIT
algorithm (courtesy of Prof. Bersekas of MIT)
was implemented to obtain the optimal spare
channel usage for a given link failure.



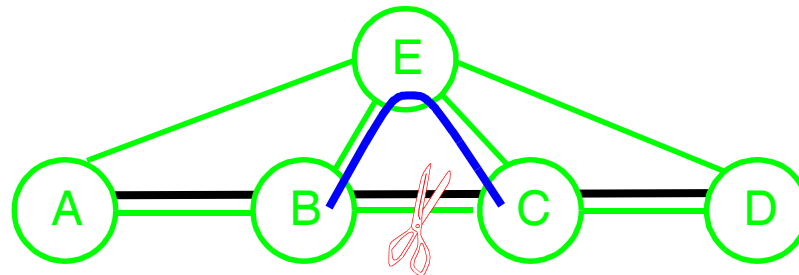
Restoration Level and Time to Restore

The selection of network restoration algorithms depends on the restoration curves of the algorithms, the real-time requirements, and the existence of priority traffic.

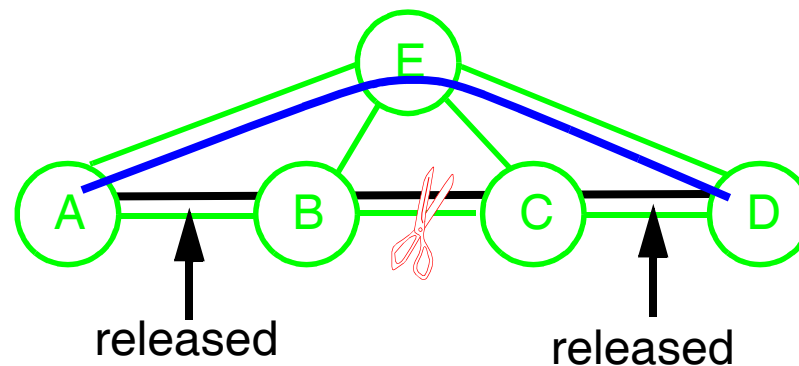




Link Based vs. Path Based Network Restoration Approaches



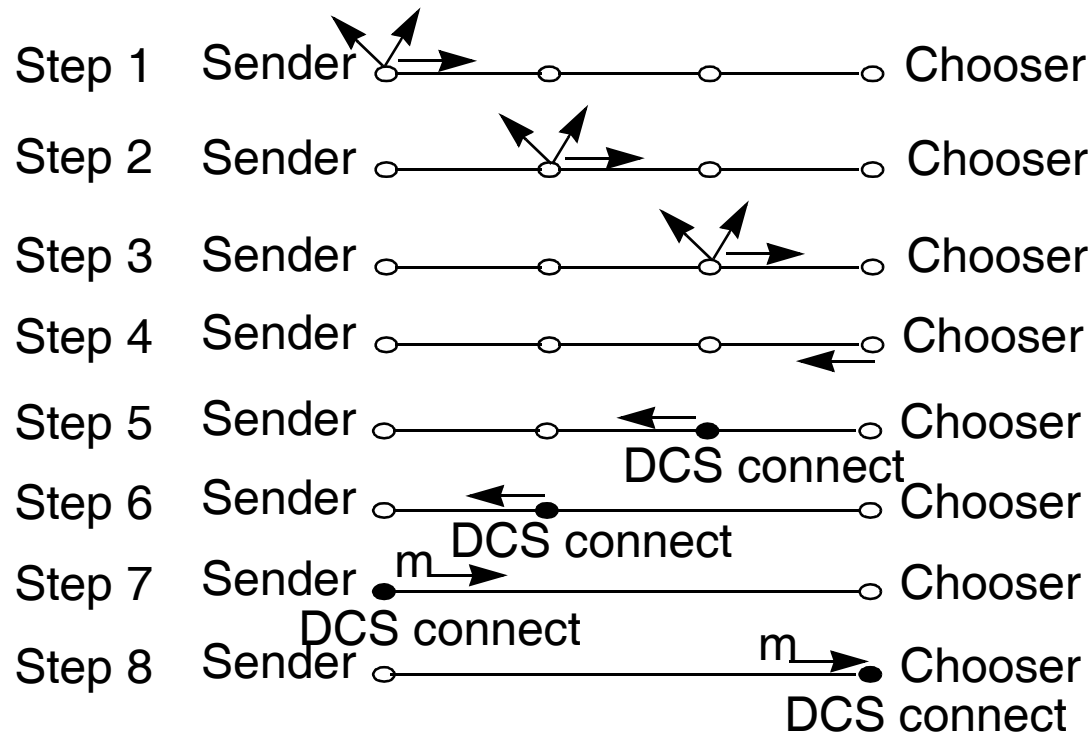
Link based restoration approaches try to find paths around the disrupted areas and keep other working channels intact.



Path based restoration approaches release the working channels of the disrupted paths and have original sources re-establish connections.



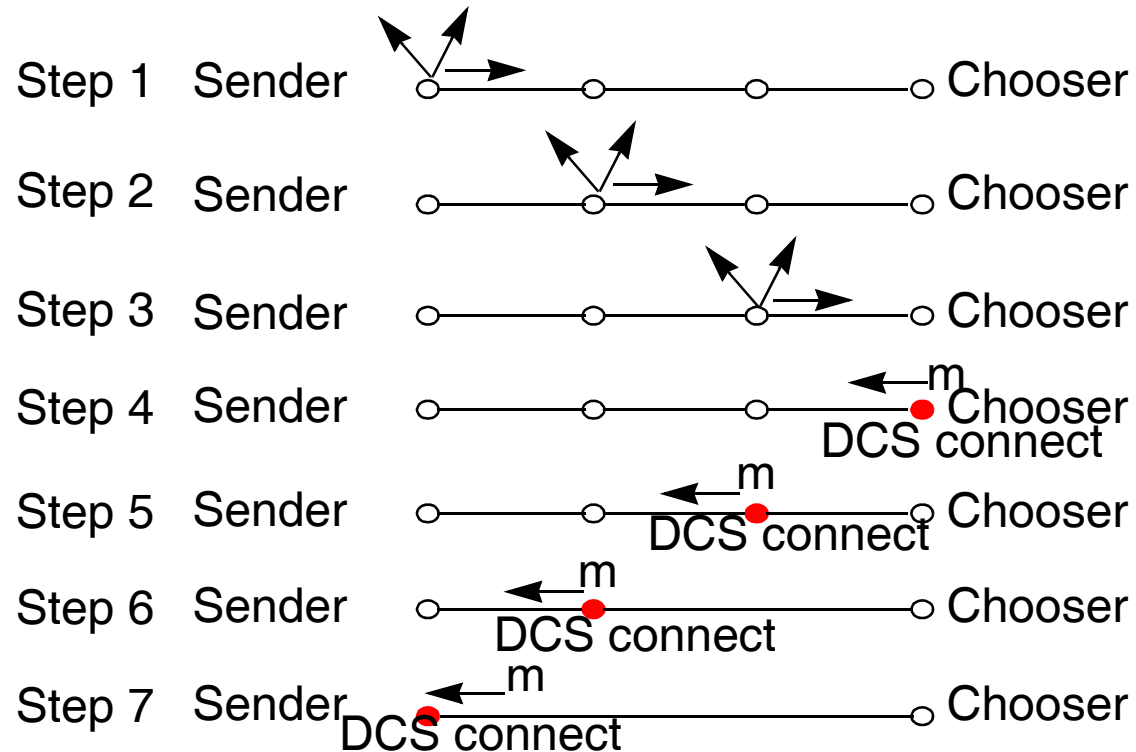
Grover's Self Healing Network [Grover87,89]



- 1 Use Sender Chooser Paradigm.
- 1 Restore disrupted connects on channel by channel basis, one msg per spare
- 1 Perform good on small networks or low spare networks.



Bellcore's FITNESS [Yang88]



- 1 Use Sender-Chooser Paradigm
- 1 Request Bandwidth on aggregated basis, one request on one outgoing link.
- 1 Use Time-out and wave mechanisms to choose largest bandwidth in a wave.

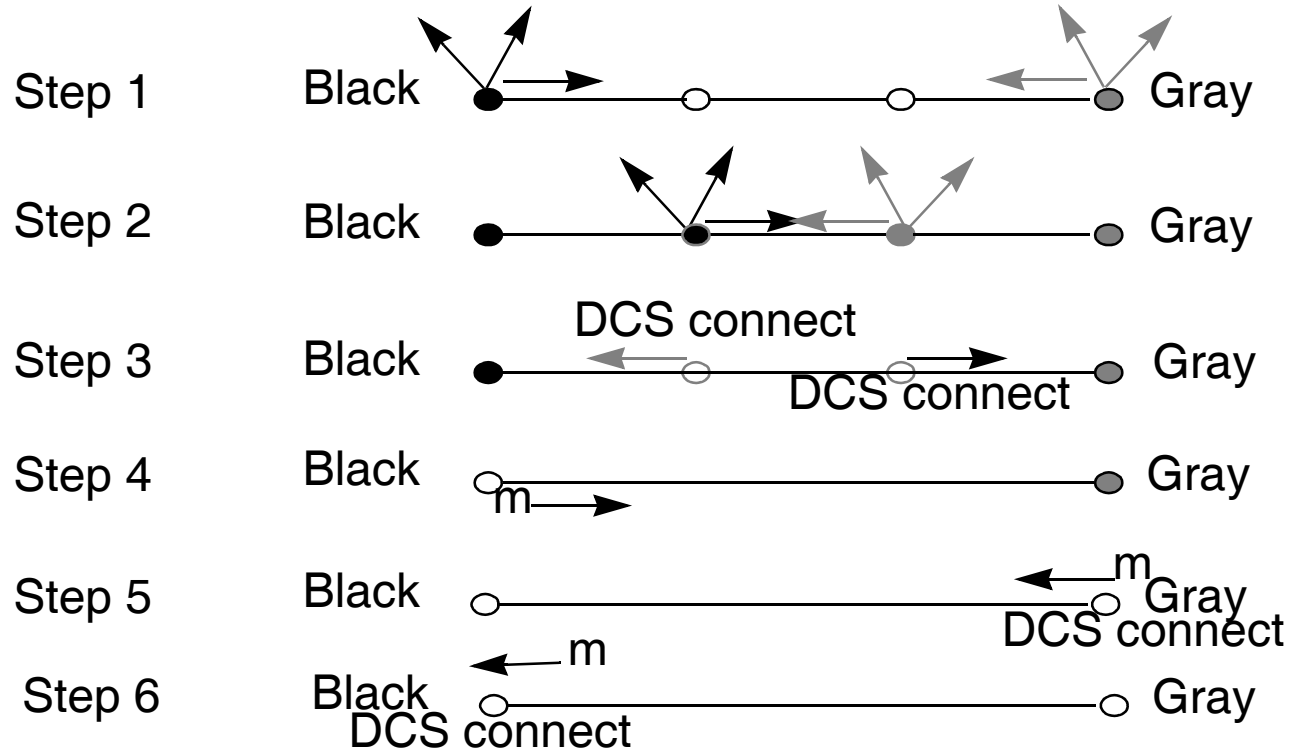


UCCSÕ RREACT[Chow93a]

- 1 Use Sender-Chooser Control Paradigm
- 1 Use aggregate request similar to FITNESS.
- 1 Attach traverse path info in the request messages.
- 1 Request messages explore all possible paths between Sender and Chooser.
- 1 Chooser builds "smart" current network topology based on these path info.
- 1 Chooser allocate bandwidth on FIFS basis.
- 1 Consistently find paths with low spare usage.
- 1 Very reliable.
- 1 Drawback: Message volume increases exponentially with network size.



UCCSÕ Two Prong [Chow93b]



- 1 Two disrupted end nodes broadcast request simultaneously.
- 1 Tandem nodes on receiving msgs from both ends start connections.
- 1 Request messages forward (not broadcast) to other nodes after mid-way
- 1 Fast concurrent connections and low message volume.



Path-based Two Prong

- 1 In this CASI project, we designed a path-based Two Prong algorithm based on link-based Two Prong.
- 1 The network model needs to include path information.
- 1 The protocol handles the release of working capacity in the disrupted paths.
- 1 The protocol resolves the spare contention problem among requests of different disrupted paths. Use priority scheme based on Path ID.
- 1 Retry and backtrack mechanisms implemented to increase
- 1 It can handle both link and node failure cases.
- 1 Message volume increases dramatically (2~8 times) compared with link-based approaches.
- 1 Restoration levels are close to those achieved by link-based approaches.
- 1 Restoration is slower (2~10 times on NJ test network) compared with link-based Two Prong.



Path-based Two Prong: Detecting/Flooding Phase

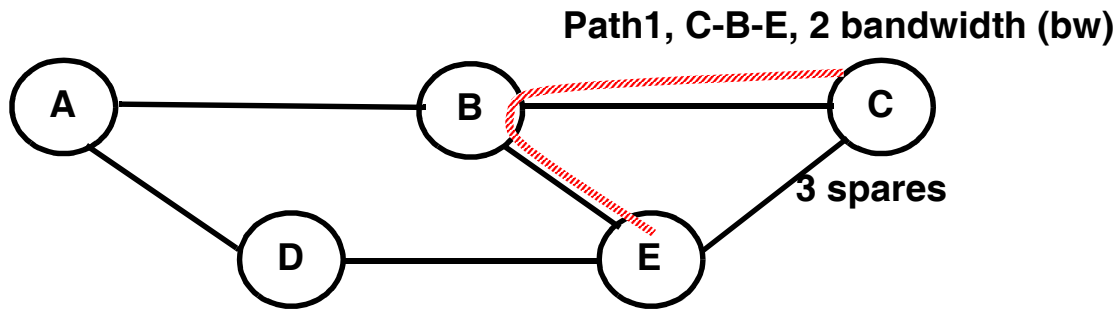


Figure 3.1a. Prior to Node B failure.

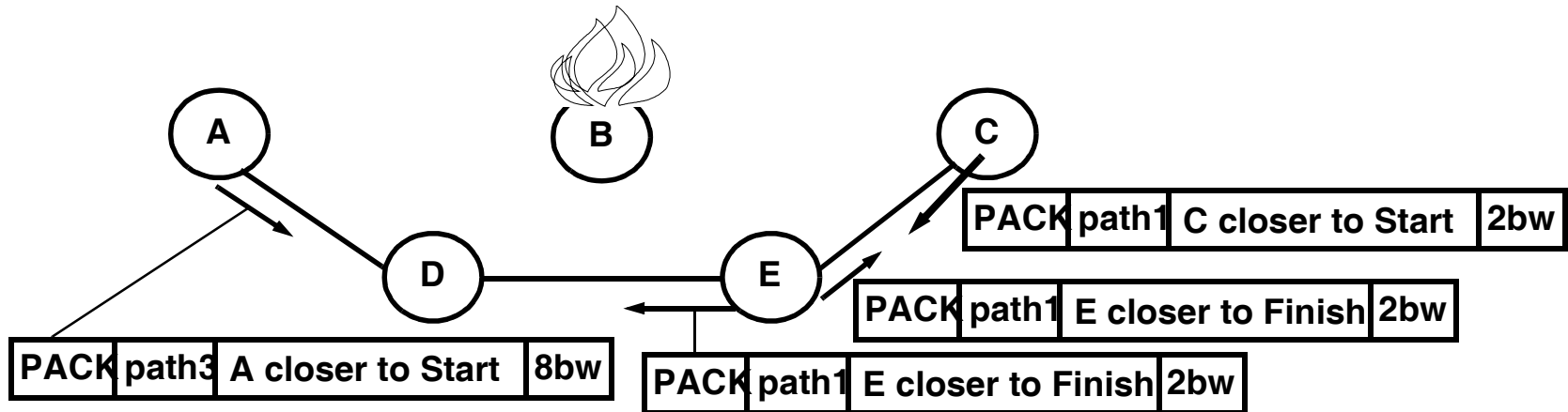


Figure 3.1b. Node B fails.



Path-base Two-Prong: Request/Carving Phase

Path1, C-B-E, 2 bandwidth (bw)

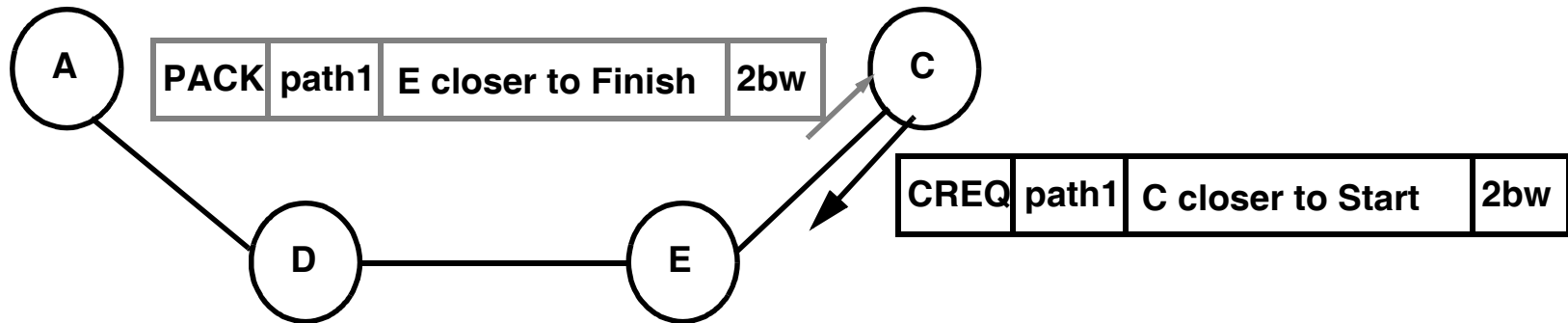


Figure 3.1c. Requesting a connection.

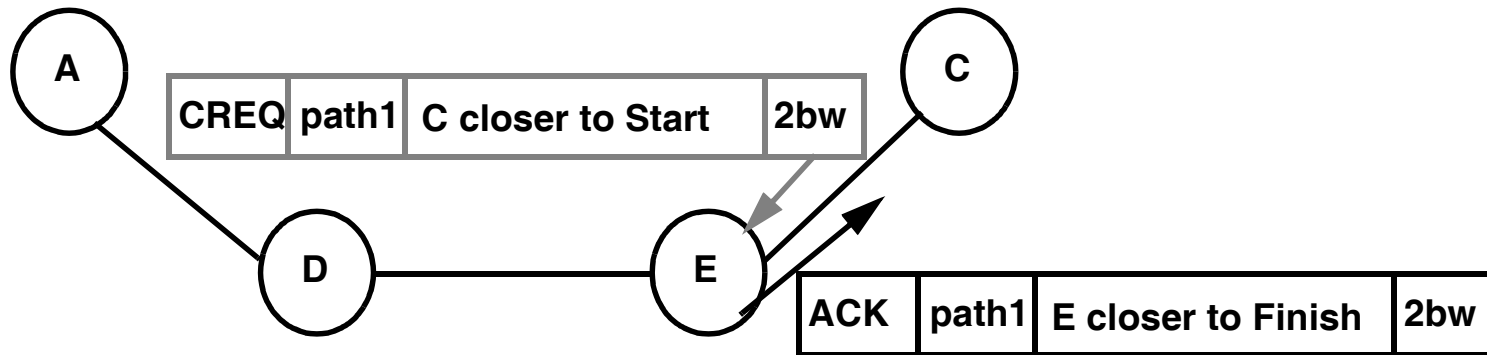


Figure 3.1d. Exploring the restoration path.



Path-base Two-Prong: Confirmation Phase

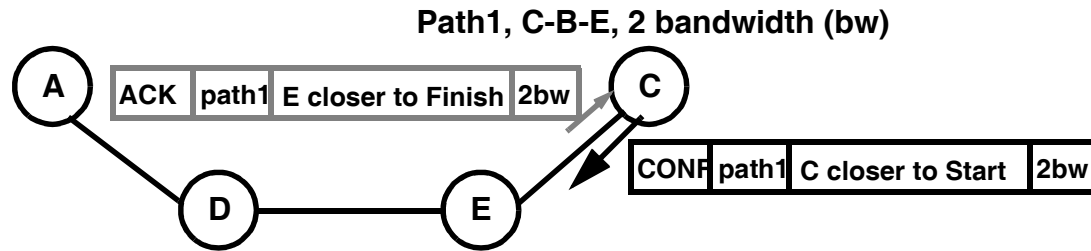


Figure 3.1e. Starting confirmation phase.

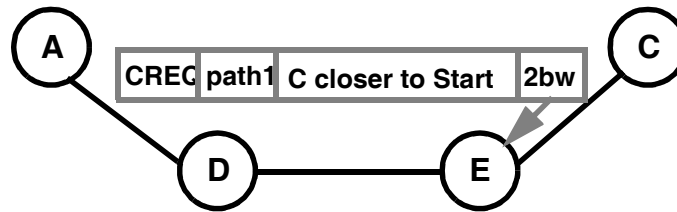


Figure 3.1f. Ignoring the connection request.

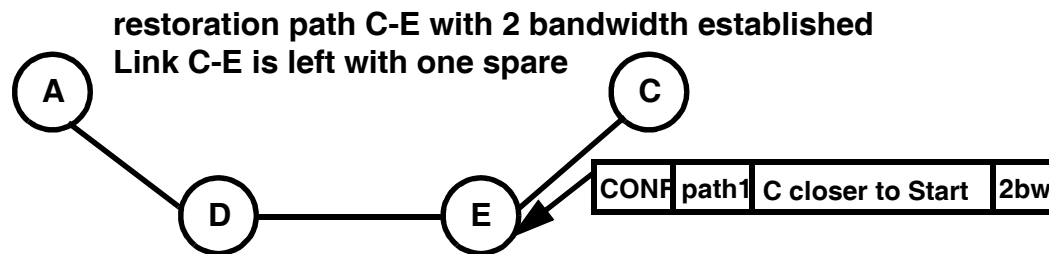
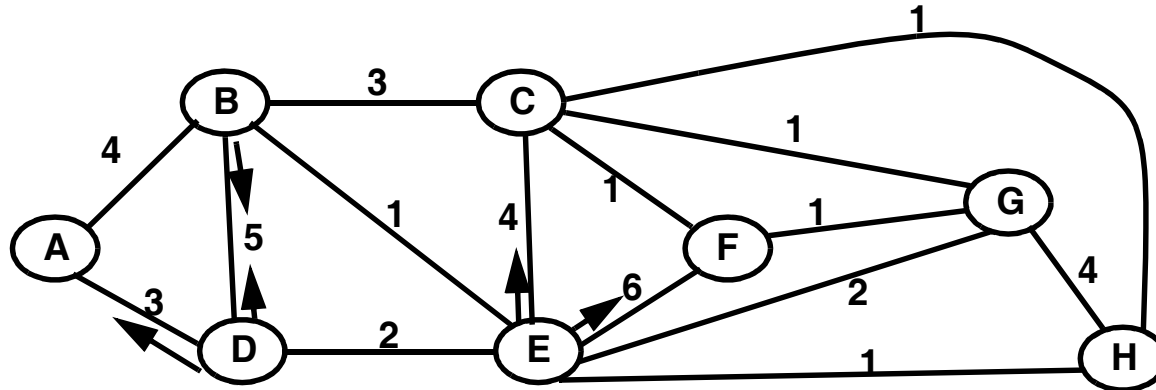


Figure 3.1g. Restoration path completed.



Problems Encountered in Path-based Two-Prong

- 1 Linger of old reservation requests after the path is restored.



Disrupted path: A-H with 6 bw.

Restoration paths: A-B-C-H, 1bw
found
A-B-C-G-H, 1bw
A-B-C-F-G-H, 1bw
A-B-E-G-H, 1bw
A-D-E-H, 1bw
A-D-E-G-H, 1bw

Figure 3.2. Nodes B, D, and E still explore restoration paths.

Sometimes the end nodes can detect the path is **fully/maximally restored**.

- 1 Detecting the request cycle. (use trail fields)

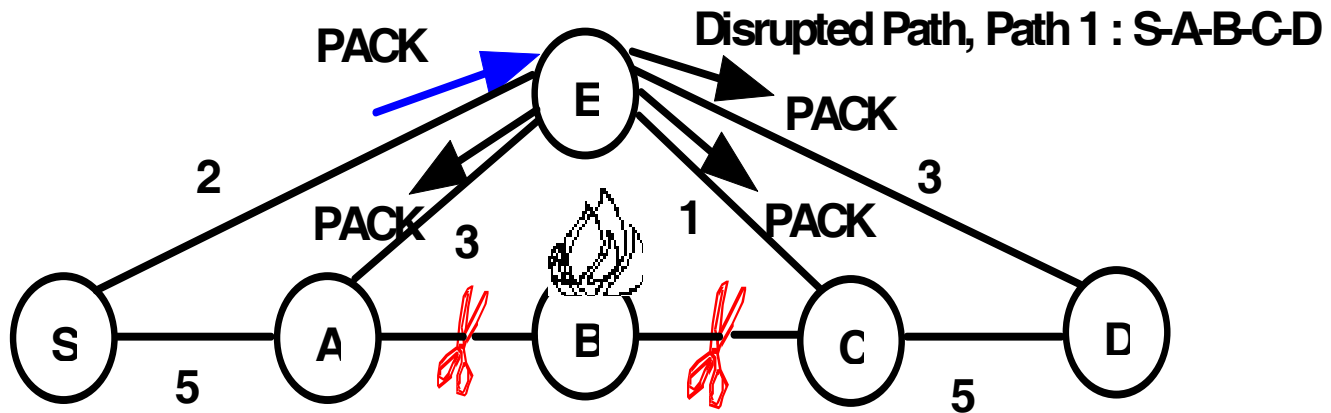
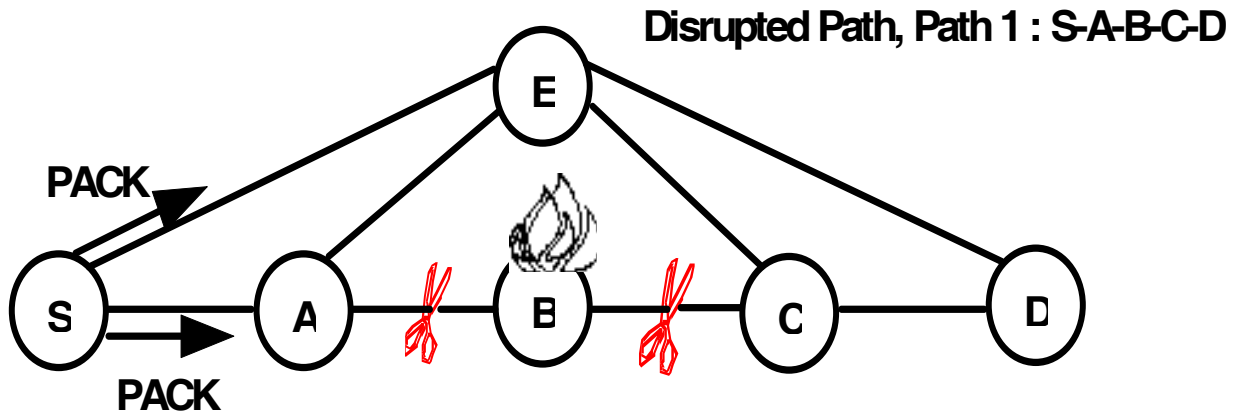


Path based One Prong

- 1 Use Sender Chooser control paradigm. (Multiple sender-chooser pairs.)
- 1 Simplify Two Prong logic.
- 1 Preliminary implementation is operational.
- 1 On NJ test network, the restoration level is very close to Two Prong.
- 1 The restoration time seems to be shorter.
- 1 Message volume almost cut in half.
- 1 There are bugs in the simulation program for some of the link cut, possibly due to the overflow on the message buffer.
- 1 The additional simulation control parameters need to be incorporated and simulation results verified.



Adaptive One Prong: Broadcast Phase





Selective Message Forwarding

When a node receives a duplicate copy of the PACK message from another neighboring node, it will only forward the PACK message to the sending node of the first PACK message.

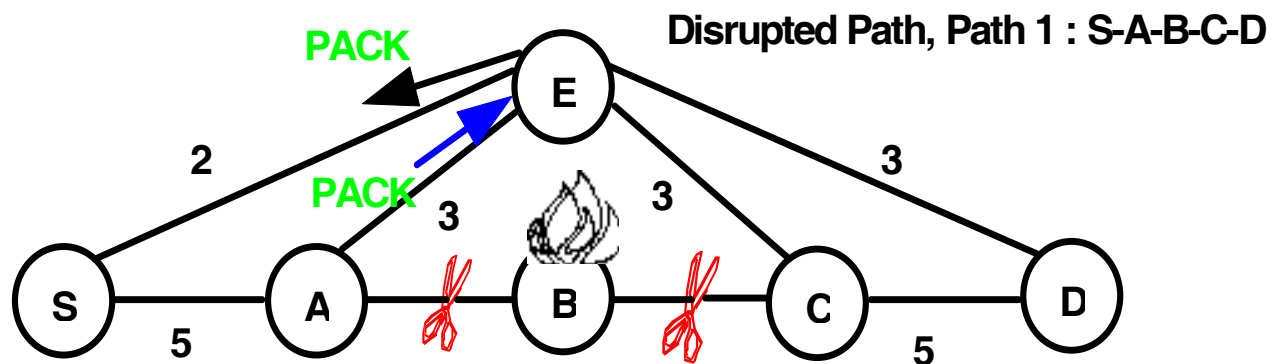
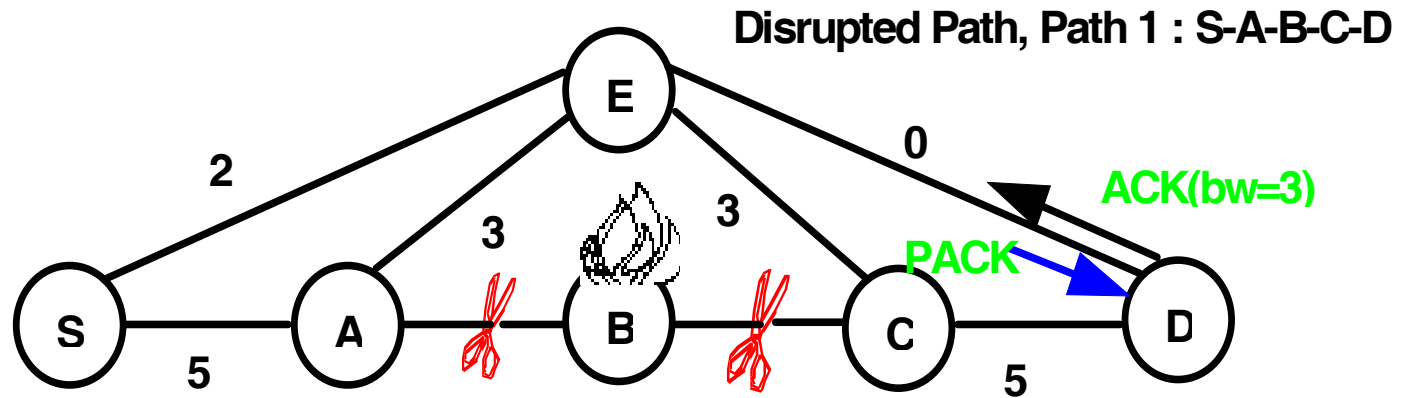


Table 1: Restoration Table in Node E

Path ID	UP	HOP	BW	DOWN	ACKHOP	Confirmed	Part	Msg_Send	Msg_Recv
1	S	1	5			0	M	0	0
1	A	2	5			0	M	0	0



Path Trace out Phase





Path Trace-out Phase

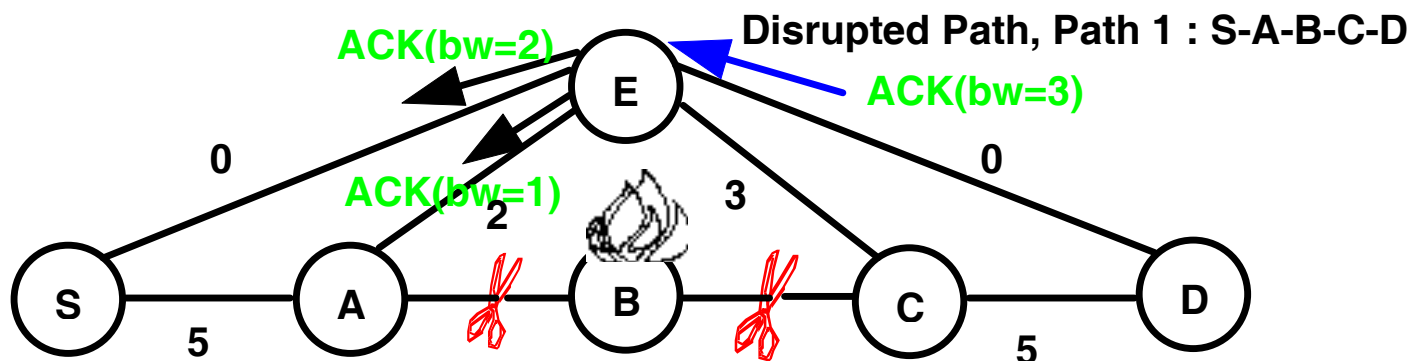


Table 3: Restoration Table in Node E after sending ACK(1bw) to A.

Path ID	UP	HOP	BW	DOWN	ACKHOP	Confirmed	Part	Msg_Send	Msg_Recv
1	S	1	2	D	1	ACKED	M	1	1
1	S	1	3			0	M	0	0
1	A	2	1	D	1	ACKED	M	2	1
1	A	2	4			0	M	0	0



Path Trace-out Phase

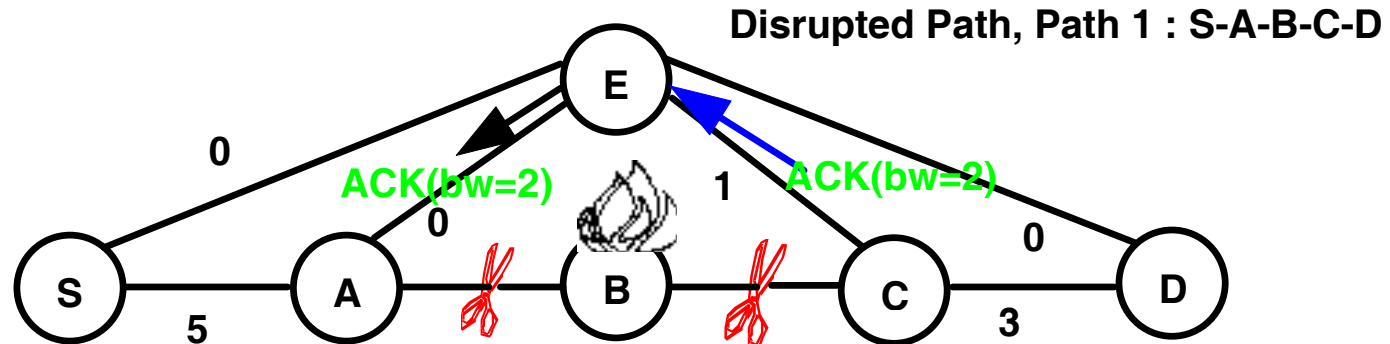


Table 4: Restoration Table in Node E after sending ACK(2bw) to A.

Path ID	UP	HOP	BW	DOWN	ACKHOP	Confirmed	Part	Msg_Send	Msg_Recv
1	S	1	2	D	1	ACKED	M	1	1
1	S	1	3			0	M	0	0
1	A	2	1	D	1	ACKED	M	2	1
1	A	2	2	C	2	ACKED	M	3	2
1	A	2	2			0	M	0	0



Handle partially satisfied request

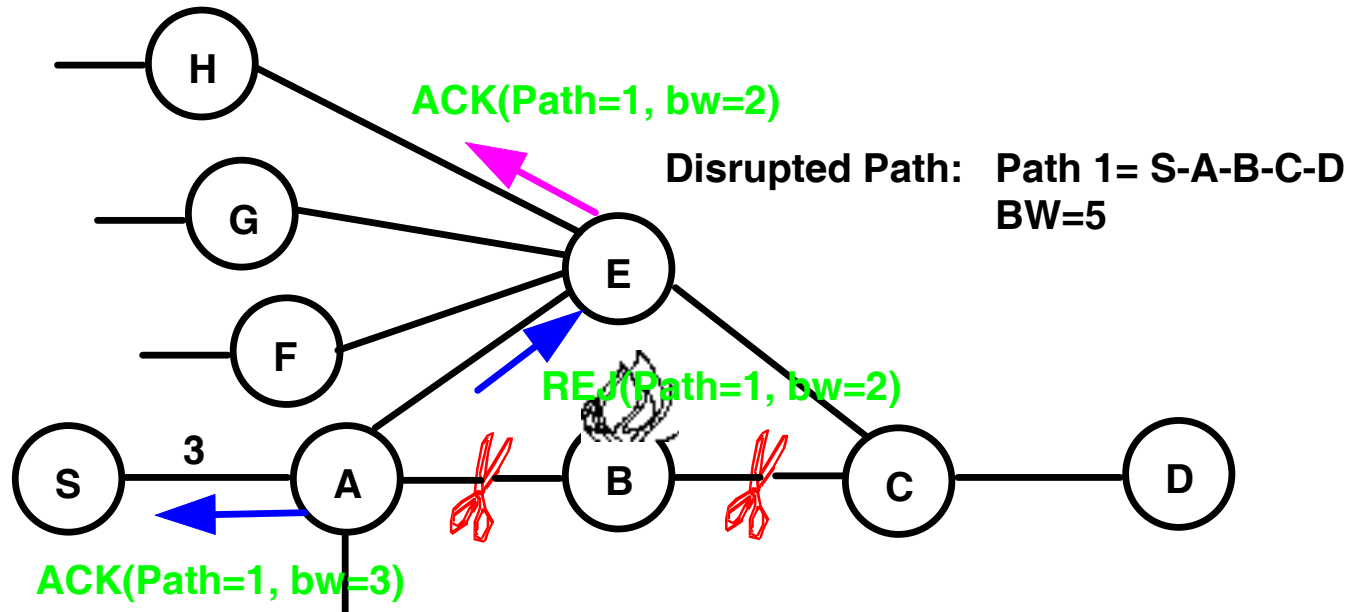


Table 10: Restoration Table in Node E after receiving REJ from A.

Path	UP	Retry	HOP	BW	DOWN	ACKHOP	Confirmed	Part	Msg_Send	Msg_Recv
1	A	1	2	3	D	1	ACKED	M	1	1
1	H	0	3	5	D	1	ACKED	M	2	1
1	H	0	3	5						
1	F	0	5	5						
1	A	1	2	2						



Confirmation Phase

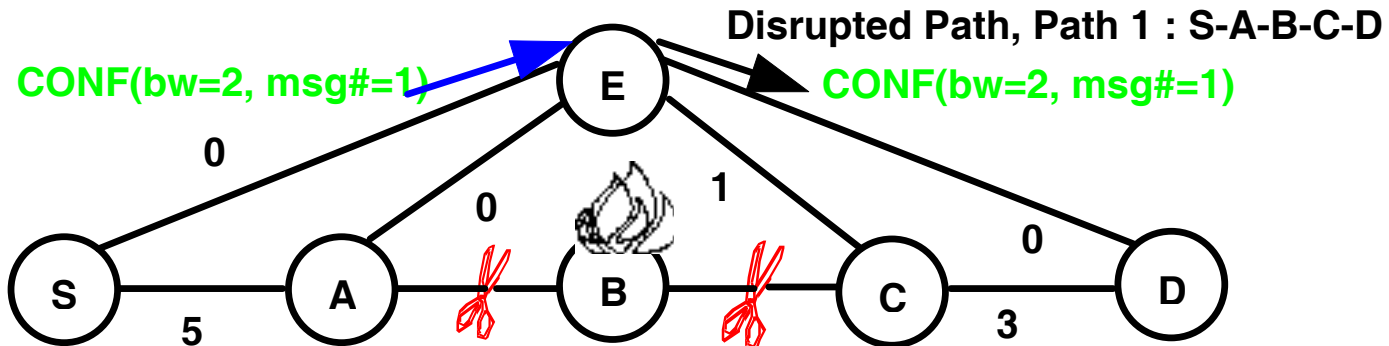
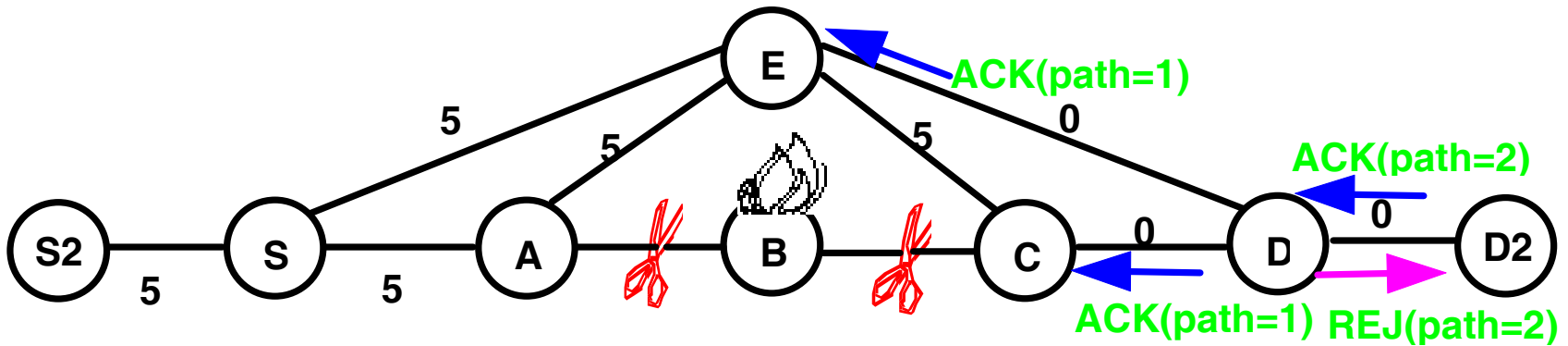
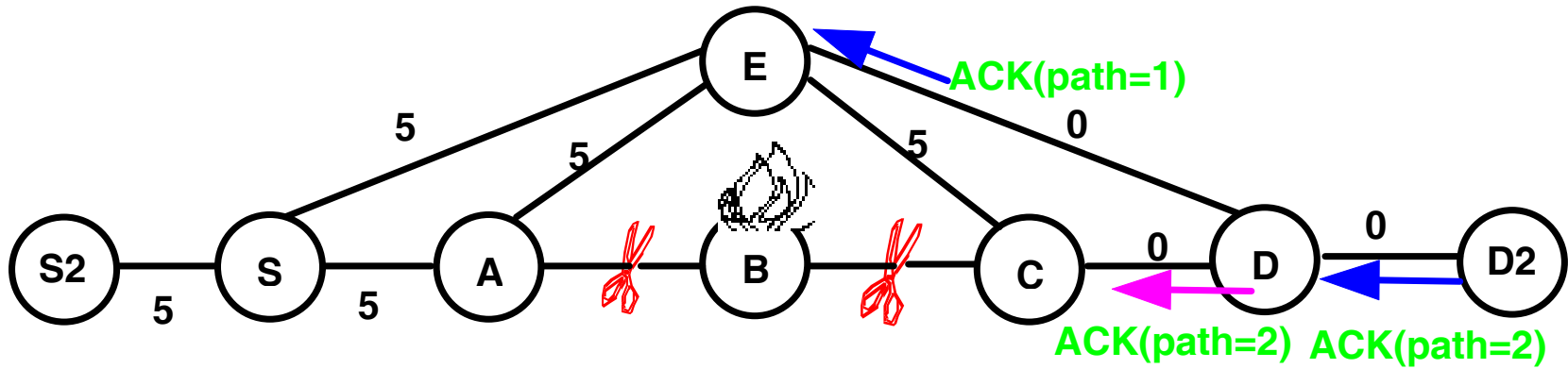


Table 5: Restoration Table in Node E after sending the first CONF to Node D.

Path ID	UP	HOP	BW	DOWN	ACKHOP	Confirmed	Part	Msg_Send	Msg_Recv
1	S	1	2	D	1	CONF	M	1	1
1	S	1	3			0	M	0	0
1	A	2	1	D	1	ACKED	M	2	1
1	A	2	2	C	2	ACKED	M	3	2
1	A	2	2			0	M	0	0



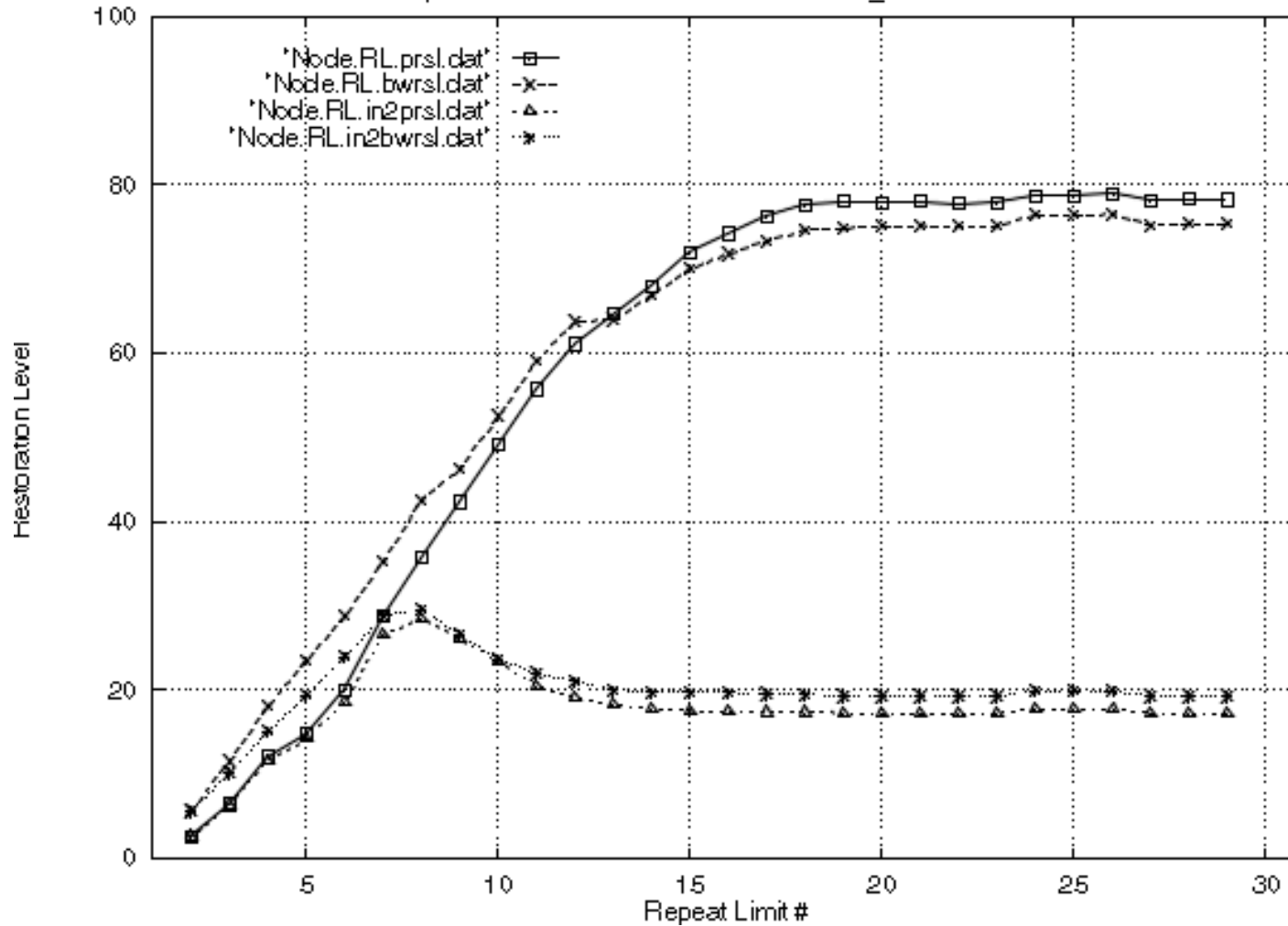
Conservative vs. Aggressive Bandwidth Request





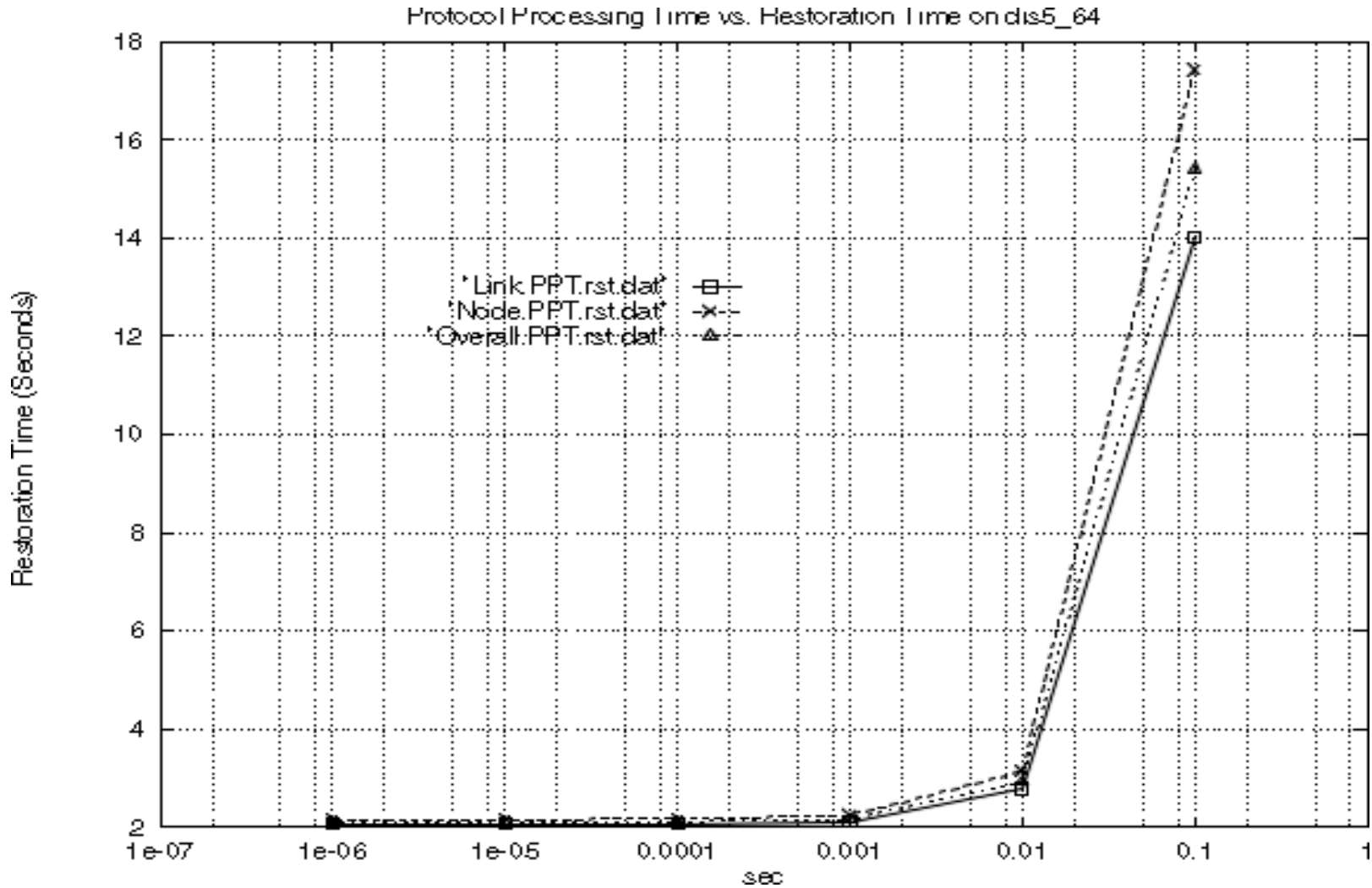
Impact of Repeat limit on Restoration level

Repeat Limit vs. Restoration Level on dis5_64: Node case



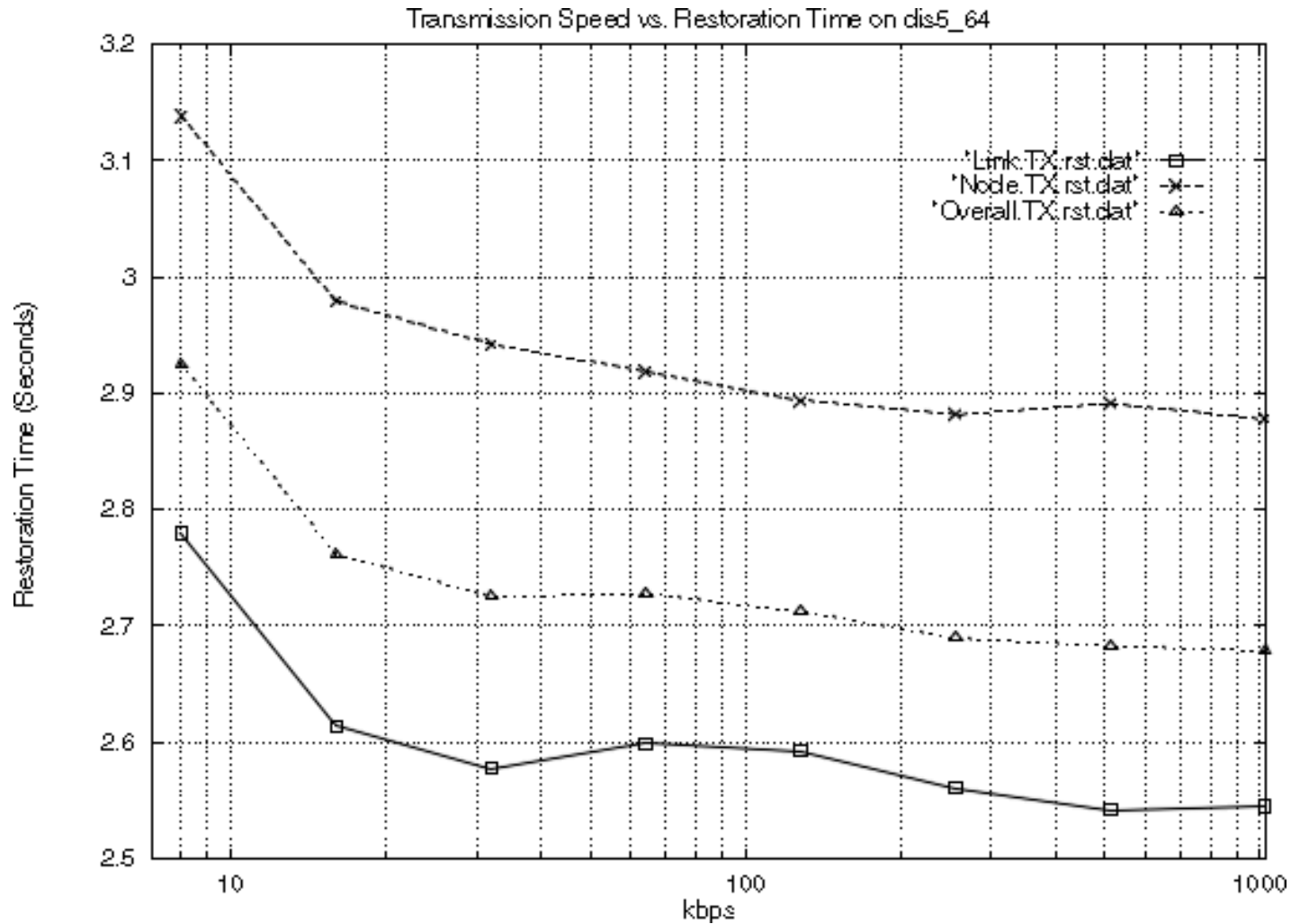


Impact of Protocol Processing time on Performance





Impact of Transmission Speed





Comparison of Network Restoration Algorithms Link Failure Cases

Scenario	Perf. Metric	Centralized Link-based	Two Prong Link/Path based	One Prong Path Based	Grover's SHN	FITNESS	RREACT
New Jersey Single Link Failure N01 - N02	Time msec	257	482/2959	1287	3126	1096	582
	Level	100%	100%	74%	100%	100%	100%
	Spares Used	312	318/160	60	312	343	252
	# of Msgs	12	126/674	308	4269	136	78
New Jersey Single Link Failure N04- N05	Time msec	127	107/874	862	3173	1151	275
	Level	100%	100%	100%	100%	100%	100%
	Spares Used	237	204/196	290	212	269	204
	# of Msgs	12	89/222	157	2753	30	107
New Jersey Single Link Failure N08 - N11	Time msec	205	475/4836	1363	2387	1827	739
	Level	100%	100%/98%	100%	93.75%	100%	100%
	Spares Used	301	215/175	160	192	239	233
	# of Msgs	13	133/914	413	3177	197	114

Restoration Time: centralized methods faster, link-based Two prong close.

Number of Messages: centralized methods have 8-10 times fewer messages.

SHN restoration slows down due to heavy message volume.



Comparison of Network Restoration Algorithms Node Failure Cases

Scenario	Perf. Metric	Centraliz ed Path based	Centraliz ed Link based	Centralized Combined	Komine*	Two Prong Path based	One Prong Path Based
New Jersey Single Node Failure N01	Time msec Level # of Msgs Spare Usage	467 100% 16 -	396 81.9% 13 -	430 81.9% 15 -	1445 57% 1413 -	35889 46% 863 3 (r168)	1526 46% 454 -53 (r168)
New Jersey Single Node Failure N04	Time msec Level # of Msgs Spare Usage	191 100% 33 -	153 100% 28 -	167 100% 11 -	2025 100% 1647 -	3175 100% 596 61(r61)	996 100% 209 63(r61)
New Jersey Single Node Failure N05	Time msec Level # of Msgs Spare Usage	416 100% 18 -	370 90.5% 18 -	426 100% 18 -	2337 91% 1633 -	18954 69% 5246 72(r102)	1940 80% 442 61(r102)

* These are the simulation results of our implementation of Fujitsu's KOMINE network restoration algorithm.



NSTOOL 1.2

Built a graphical user interface to facilitate the survivable network design.

The screenshot displays the NSTOOL 1.2 graphical user interface. At the top, a status bar shows the system name 'SANLUIS', the user 'LOCAL', and the terminal 'PIGLET'. The date and time are 'Wed Oct 5 12:52:28 1994'. The main window title is 'nstoool : NETWORK SIMULATION TOOL'. Below the title bar is a menu bar with options: Simulation, Network, Zoom, Comment, Simulation Control, Status, Quit, and Print. The interface is divided into several panes:

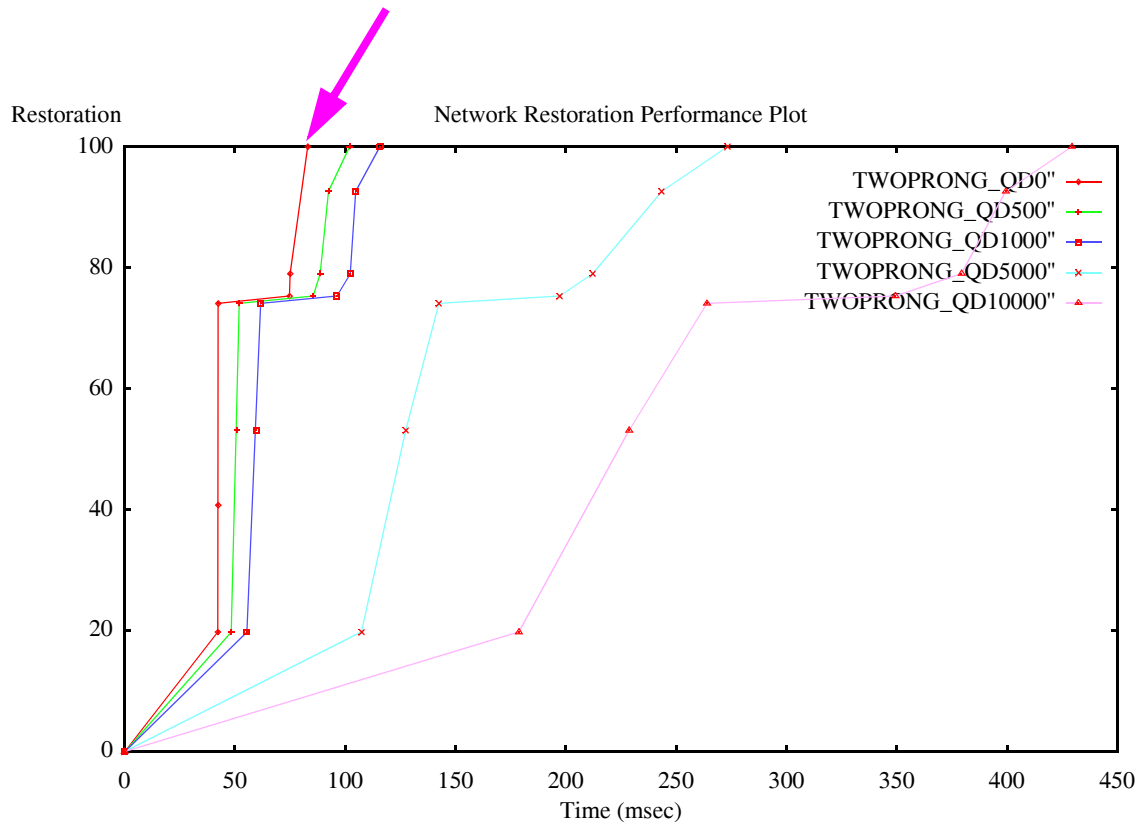
- Simulation Control:** Includes buttons for 'create Network', 'load Network', 'save Network', 'place node', 'move node', 'remove node', 'change node label', 'change initial node', 'place link', 'move link', 'remove link', 'change link label', 'move network', and 'redisplay'.
- Zoom:** Includes buttons for '1', '1/2', '1/4', '1/8', and '1/16'.
- Comment:** Includes buttons for 'enter comment', 'change comment', and 'remove comment'.
- Simulation:** Includes buttons for 'start', 'pause', and 'resume'.
- Network Statistics:** Includes buttons for 'Show_all_nodes', 'Show_node', 'Show_all_links', 'Show_link', 'Plot_graph', and 'Show message table'.
- Output Messages:** A scrollable text area showing simulation results, such as:

```
run=10 bw=20 rstrd=58.02 time=751.29
path=(N05-N11-N09-N04-N08)
run=10 bw=16 rstrd=77.78 time=1055.90 path=(N05-N07-N08)
run=10 bw=14 rstrd=95.06 time=1382.73 path=(N05-N11-N08)
run=10 bw= 3 rstrd=98.77 time=1862.58
path=(N05-N03-N01-N04-N09-N11-N08)
run=10 bw= 1 rstrd=100.00 time=2223.85
path=(N05-N03-N01-N06-N08)
run=40 bw=16 rstrd=19.75 time=178.69 path=(N08-N07-N05)
run=40 bw=27 rstrd=53.09 time=228.69 path=(N08-N06-N05)
```
- Simulation Parameter Specification Frame:** A form for configuring simulation parameters:
 - Algorithm(s) to be simulated: FITNESS RREACT SHN TWOPRONG PATHRES LINKRES COMBRES
 - NETWORK FILENAME: nj.net
 - Link break at node1: N05, node2: N08
 - For CNR, NOC: N05
 - event file: event_file
 - path file: path_file
 - Transmission speed(s) to be simulated: 8 64 128 172 256 320 384 448 512 576
 - DCS connect time (in sec): 0.01
 - Queueing delay per message (in sec): 0.01
 - Parallel DCS operating mode (0 for connecting all in one, 1 for individual, n for n in one): 0

On the right side, a network graph titled 'NJ.NET' is displayed, showing nodes N01 through N11 connected by links with associated bandwidth and delay values.



Exploring the theoretical limitation of survivable network algorithms



Generated by GNUPLOT