Siemens – LUMS
Resilient Multi-layer Networks Project
Project Closing Document and Deliverables

Project Description

Ethernet is a low-cost, flexible and high-speed transport technology, which has traditionally seen success in local area networks and is rapidly gaining popularity in metro networks. However, its control plane, primarily based on spanning tree protocol, is not well-suited for the metro and core networks. For such networks, the Internet Engineering Task Force (IETF) is evaluating a proposed framework called GELS which uses GMPLS as the control plane for Ethernet data plane.

An important consideration in the search for new control planes for Ethernet is that it should offer efficient network utilization, resilience, traffic engineering and QoS support, as compared to the decades old native Ethernet control plane. GMPLS inherently offers all these capabilities. However, tradeoffs in using GMPLS for Ethernet are less well-known to academia and industry. Siemens, Germany and LUMS, Pakistan initiated the collaboration to study such tradeoffs in February 2006.

The major goal of the collaborative project was to compare and contrast two technologies for connecting end-users in metropolitan networks. One technology (RSTP) has existed for more than two decades and a new proposed technology (GELS) is under consideration by standardization bodies that for Internet technologies.

Individuals involved in this cutting edge research from Siemens' High Speed Networks group in Germany were:

- Dr. Hubertus Dewitz (Liaison and Advisory role)
- Dr. Johannes Riedl (Technical role)
- Dr. Martin Nathensen (Technical role)
- Dr. Kulkarni (Technical role)

The LUMS Networks and Communications Research group comprised of

- Dr. Khurram Afridi (Advisory role)
- Dr. Zartash Afzal Uzmi (Technical and Supervisory role)
- Fahad Rafique Dogar (Research Associate)
- Muhammad Saqib Ilyas (Student Researcher)
- Fawaz Saleem Bokhari (Student Researcher)
- Atif Nazir (Student Researcher)

The group from LUMS conducted extensive simulation experiments to evaluate GMPLS as a control plane for metro and core networks consisting of point-to-point Ethernet links and compared it with that of native Ethernet control plane under both normal and single element failure scenarios. The LUMS team delivered a series of PowerPoint presentations to Siemens group in two phases. These presentations summarized progressive research results which were discussed in detailed teleconferences with a focus on current research direction and future aspects of Siemens-LUMS project.

Simulation Testbed and Tools (Developed and Used during the collaboration)

- STP Simulator developed using C language by the team at LUMS for computing the RSTP/STP tree in a bridged topology
- C program developed by LUMS team to repeatedly run RSTP simulations under various failure scenarios using the open source bridgesim simulator
- Unix shell scripts developed to compute convergence time from RSTP simulation results reported in bridgesim trace files
- C program developed to generate random traffic matrices for simulations performed on COST239 and COST266 network topologies. This program has the flexibility to generate either partial meshed or fully meshed network topologies.
- TOTEM (Toolbox for Traffic Engineering Methods) simulator was used to carry out simulation results
- A modification to CSPF algorithm, Compromised CSPF was developed in Java and integrated with TOTEM
- Java code to convert from TOTEM XML topology into STP simulator compatible topology files was developed
- Java code to import STP simulator’s output active topology into TOTEM was developed
- Java hooks for TOTEM and STP simulator interoperation were developed

Development Methodology

- Modified version of CSPF (Constrained Shortest Path First) algorithm called Compromised CSPF was used in order to fairly compare the performance of GELS and native Ethernet control plane. This new Compromised CSPF algorithm is integrated with the TOTEM simulator.
- Performance evaluation of both the native Ethernet and GELS control planes under normal network operations was done using STP pruned reference topologies and un-pruned reference topologies respectively. For calculating the three metrics under failure conditions, we used RSTP in native Ethernet control plane for fast convergence and LSP protection and restoration schemes in GELS control plane. Both of the above mentioned modules are integrated with TOTEM simulator.

Links to download the documents and software:

A webpage which contains all the following links is available at:
http://suraj.lums.edu.pk/~zartash/siemens-lums/index.html

Links to individual components are available at the following locations:

1. Paper summarizing research output
   http://suraj.lums.edu.pk/~zartash/siemens-lums/GELS-Siemens.pdf
2. TOTEM simulator (http://totem.run.montefiore.ulg.ac.be/) with the provision of Ethernet and GELS control plane simulation options under normal and single element failure scenarios
   http://suraj.lums.edu.pk/~zartash/siemens-lums/totem.tar.gz
3. STP Simulator developed in LUMS for evaluation of native Ethernet control plane under normal network conditions
   http://suraj.lums.edu.pk/~zartash/siemens-lums/stpsim.tar.gz
4. RSTP simulator (http://www.cs.cmu.edu/~acm/bridgesim/index.html) “bridgesim” used for native Ethernet control plane under single element failure scenarios
   http://suraj.lums.edu.pk/~zartash/siemens-lums/bridgesim.tar.gz
5. C/C++ programs to repeatedly launch bridgesim with various topologies and scenarios
   http://suraj.lums.edu.pk/~zartash/siemens-lums/c-code.zip
6. Unix shell scripts to calculate the convergence time for bridgesim simulations
   http://suraj.lums.edu.pk/~zartash/siemens-lums/shellscripts.zip
7. TOTEM XML topology files
   http://suraj.lums.edu.pk/~zartash/siemens-lums/topologies.tar.gz
8. TOTEM XML traffic matrices
   http://suraj.lums.edu.pk/~zartash/siemens-lums/traffic-matrices.tar.gz
9. C toolset for topology and traffic matrix generation
   http://suraj.lums.edu.pk/~zartash/siemens-lums/topgen.tar

How to run the software:

1. Download and build the STP simulator (building instructions below!)
2. Download and build bridgesim simulator
3. Copy the shell scripts into /netsim directory
4. Download and build TOTEM simulator
5. Download the TOTEM XML topology files
6. Download the TOTEM traffic matrices

Building and running STP simulator

1. Create a directory called netsim in the root directory by changing to the root directory and issuing the command “mkdir netsim”
2. Copy the stpsim.tar.gz to the /netsim directory
3. Extract the contents of the STP simulator tar ball using the command “tar –xzvf stpsim.tar.gz”
4. Change the directory to where the simulator has been extracted and type “gcc network_simulator.c bridge.c –o bridge”
5. Launch the simulator using the command “./bridge topology.in scenario.in” to make sure that the simulator is running properly
6. The simulator should be placed in /netsim directory to be able to work properly

Building and running RSTP simulator

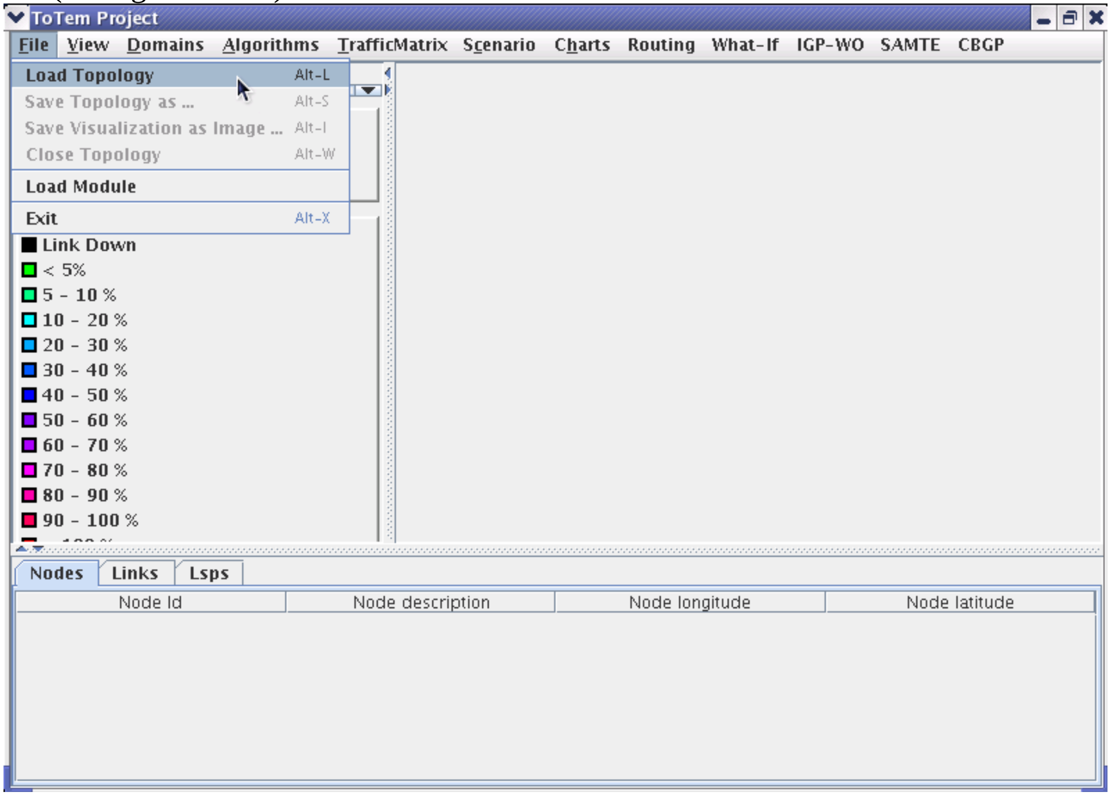
1. Copy the bridgesim tar ball to the /usr/src directory
2. Extract the contents of the bridgesim tar ball by using the command “tar –xzvf bridgesim.tar.gz”
3. Build the simulator using the command “make”
4. For a list of command line arguments and to ensure that the simulator is working properly type the command “./bs”
5. Make sure that the simulator is placed in the /netsim directory for the testbed to work properly

Building and running TOTEM:

1. Copy the TOTEM tar ball into the /usr/src directory
2. Extract the tar ball by running the following command:

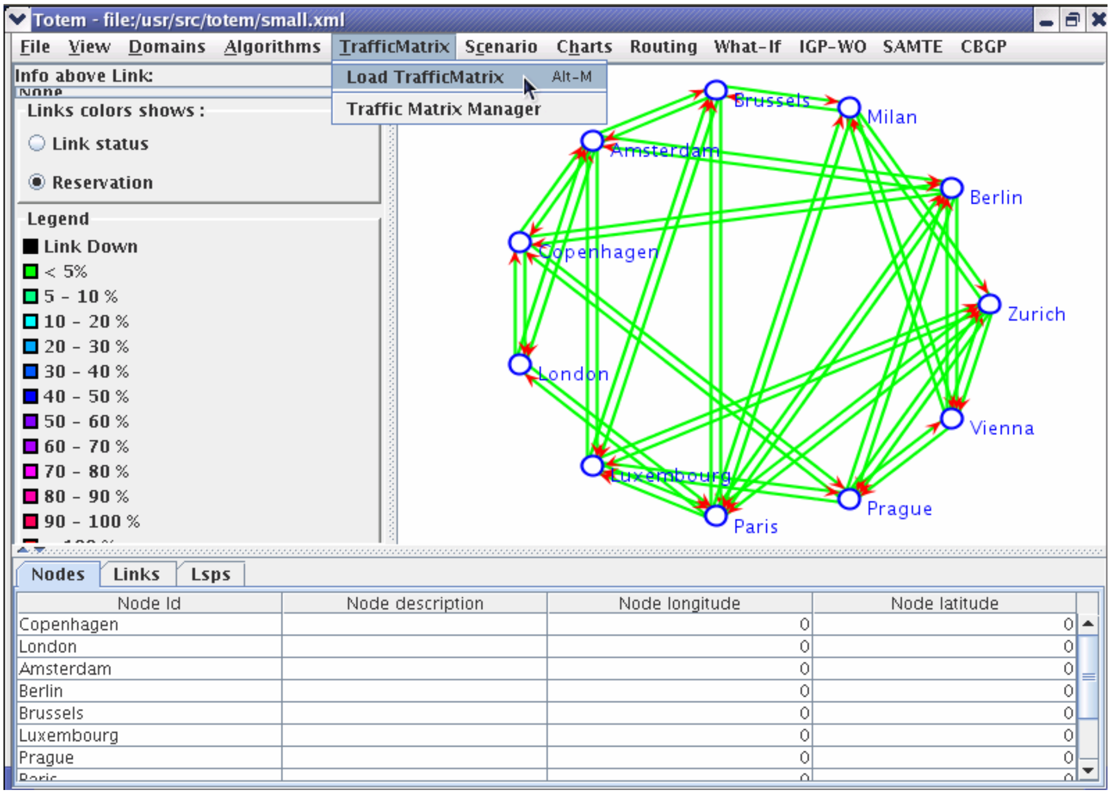
   ```
   tar –xzvf totem.feb.07.tar.tar.gz
   ```
3. Change to the directory in which TOTEM is extracted
4. Follow the instruction in README file located in this directory to build TOTEM
5. Copy the traffic matrix and topology XML files from the respective tar balls by copying them to the /usr/src/totem folder and using the “tar –xzvf topologies.tar.gz” and “tar –xzvf traffic-matrices.tar.gz” commands
6. Start X-Windows if not already done
7. Run the command ./totem.sh to launch TOTEM simulator

8. Load the topology file by going to “File” menu and selecting “Load topology” (see figure below)



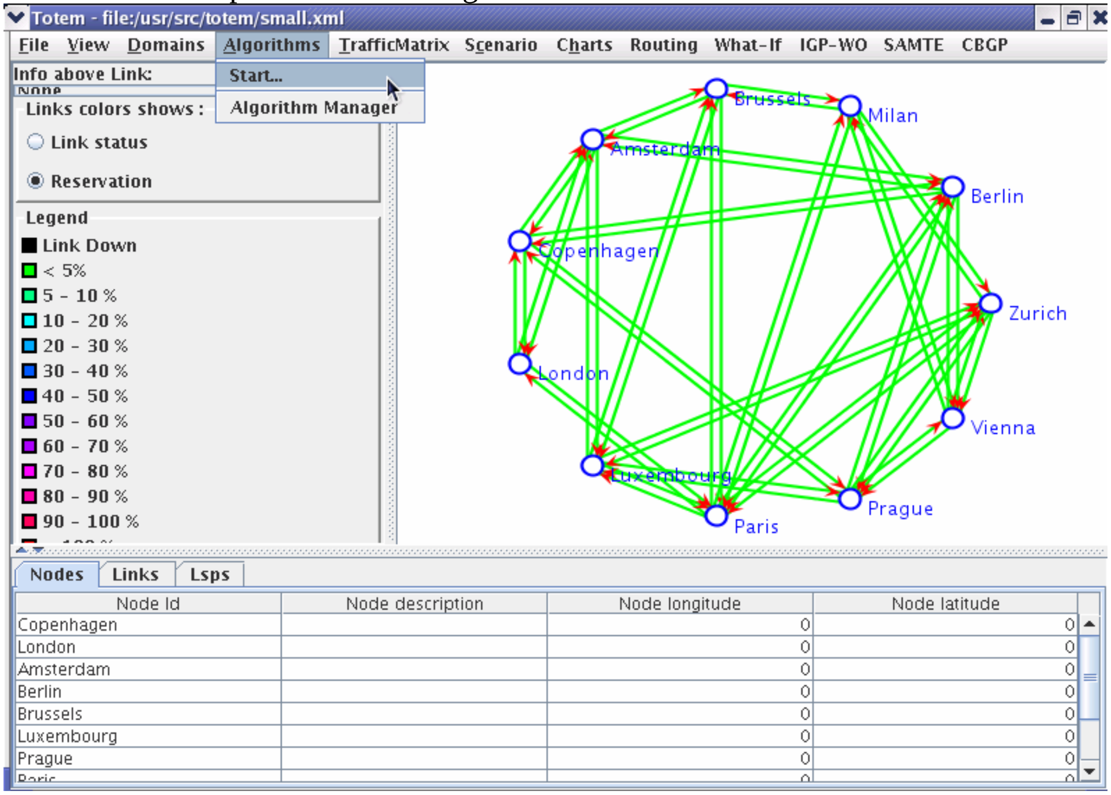
Node Id	Node description	Node longitude	Node latitude

9. Load a traffic matrix



Node Id	Node description	Node longitude	Node latitude
Copenhagen		0	0
London		0	0
Amsterdam		0	0
Berlin		0	0
Brussels		0	0
Luxembourg		0	0
Prague		0	0
Paris		0	0

10. Start the Compromised CSPF algorithm



Info above Link: None

Links colors shows :

○ Link status

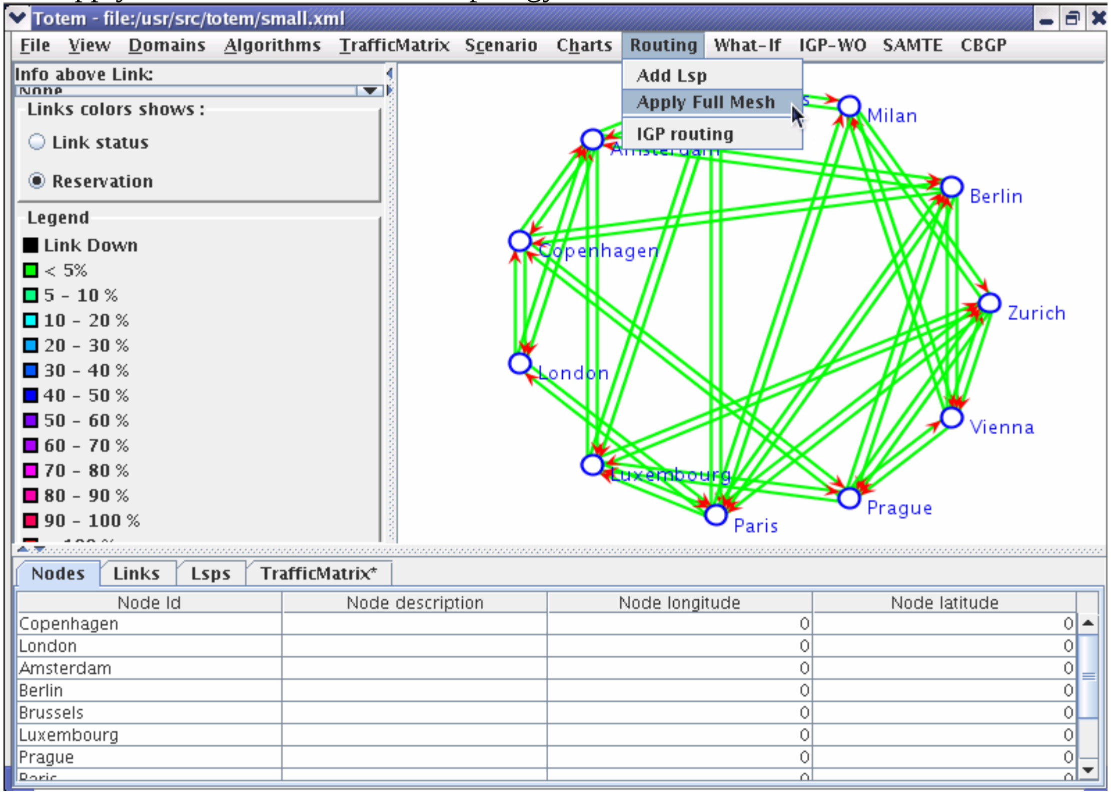
● Reservation

Legend

- ■ Link Down
- 🟩 < 5%
- 🟩 5 - 10 %
- 🟦 10 - 20 %
- 🟦 20 - 30 %
- 🟦 30 - 40 %
- 🟦 40 - 50 %
- 🟪 50 - 60 %
- 🟪 60 - 70 %
- 🟪 70 - 80 %
- 🟥 80 - 90 %
- 🟥 90 - 100 %

Node Id	Node description	Node longitude	Node latitude
Copenhagen		0	0
London		0	0
Amsterdam		0	0
Berlin		0	0
Brussels		0	0
Luxembourg		0	0
Prague		0	0
Paris		0	0

11. Apply the traffic matrix to the topology



Info above Link: None

Links colors shows :

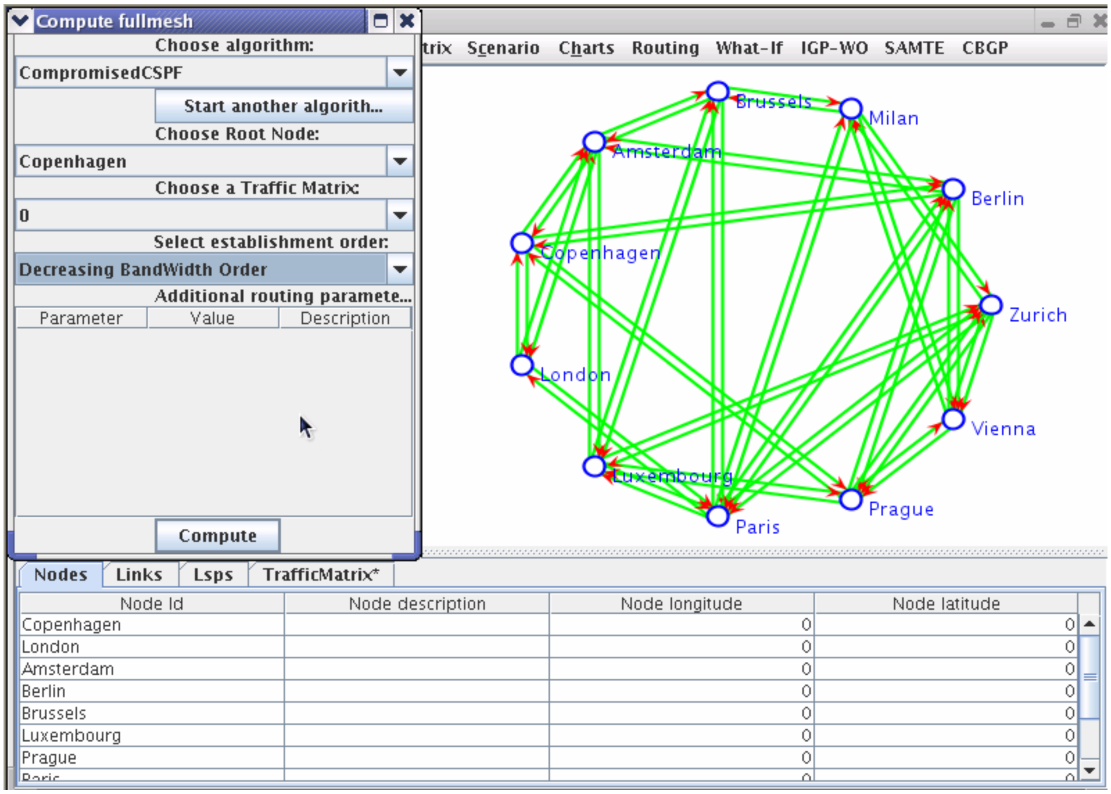
○ Link status

● Reservation

Legend

- ■ Link Down
- 🟩 < 5%
- 🟩 5 - 10 %
- 🟦 10 - 20 %
- 🟦 20 - 30 %
- 🟦 30 - 40 %
- 🟦 40 - 50 %
- 🟪 50 - 60 %
- 🟪 60 - 70 %
- 🟪 70 - 80 %
- 🟥 80 - 90 %
- 🟥 90 - 100 %

Node Id	Node description	Node longitude	Node latitude
Copenhagen		0	0
London		0	0
Amsterdam		0	0
Berlin		0	0
Brussels		0	0
Luxembourg		0	0
Prague		0	0
Paris		0	0



Choose algorithm: CompromisedCSPF

Choose Root Node: Copenhagen

Choose a Traffic Matrix: 0

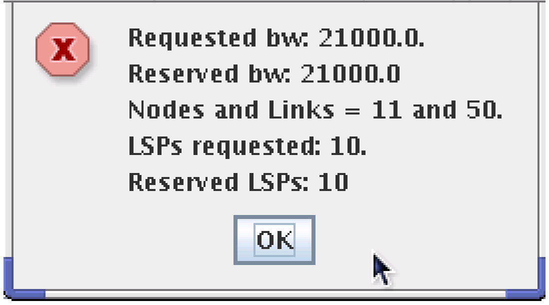
Select establishment order: Decreasing BandWidth Order

Additional routing paramete...

Parameter	Value	Description

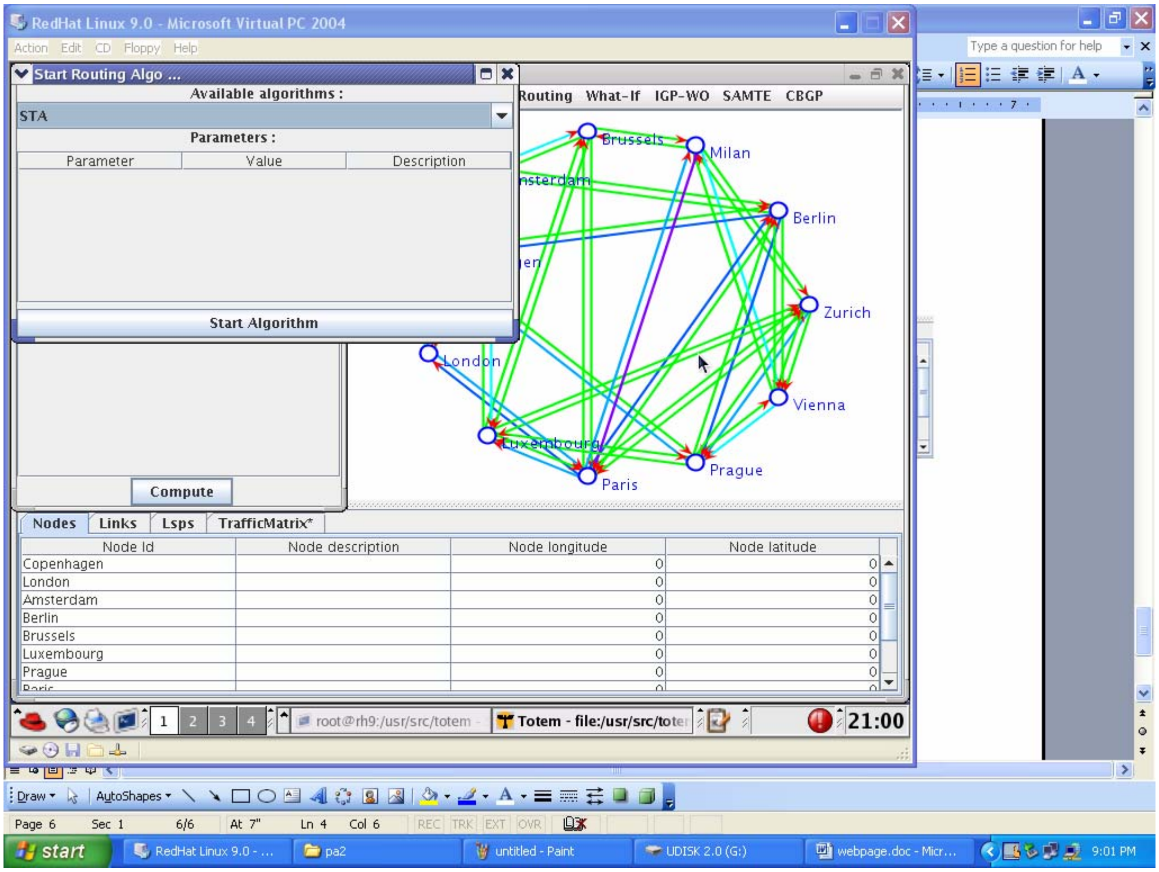
Node Id	Node description	Node longitude	Node latitude
Copenhagen		0	0
London		0	0
Amsterdam		0	0
Berlin		0	0
Brussels		0	0
Luxembourg		0	0
Prague		0	0
Paris		0	0

12. A dialog box will display the aggregated simulation results



Requested bw: 21000.0.
Reserved bw: 21000.0
Nodes and Links = 11 and 50.
LSPs requested: 10.
Reserved LSPs: 10

13. To run STP simulations, start the STA algorithm and select the bridge to be used as root bridge



Start Routing Algo ...

Available algorithms :

STA

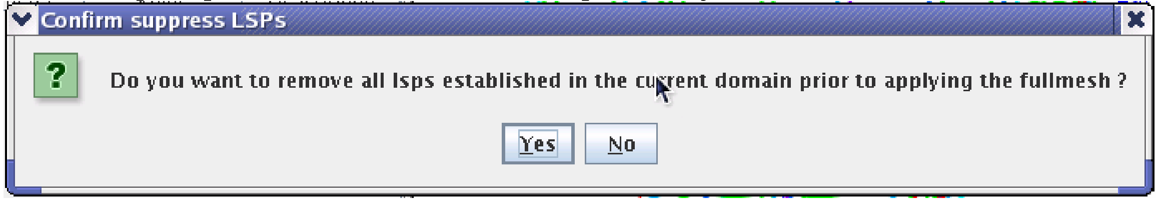
Parameters :

Parameter	Value	Description

Start Algorithm

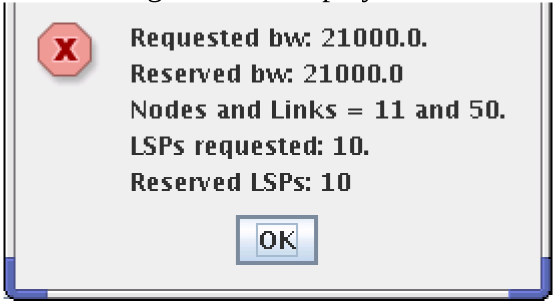
Node Id	Node description	Node longitude	Node latitude
Copenhagen		0	0
London		0	0
Amsterdam		0	0
Berlin		0	0
Brussels		0	0
Luxembourg		0	0
Prague		0	0
Paris		0	0

14. When prompted, click “Yes” to clear the previously established LSPs

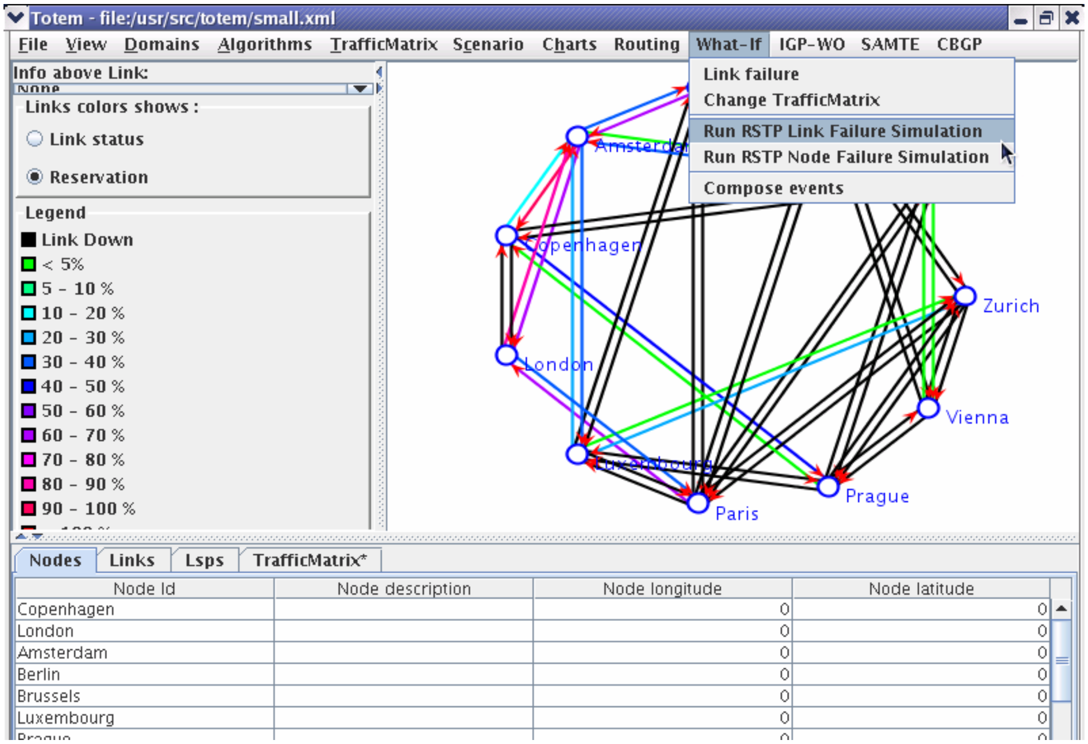

Confirm suppress LSPs

Do you want to remove all lsps established in the current domain prior to applying the fullmesh ?

15. A dialog box will display simulation results


Requested bw: 21000.0.
Reserved bw: 21000.0
Nodes and Links = 11 and 50.
LSPs requested: 10.
Reserved LSPs: 10

16. To run RSTP simulation, select RSTP link failure simulation


Totem - file:/usr/src/totem/small.xml
File View Domains Algorithms TrafficMatrix Scenario Charts Routing What-If IGP-WO SAMTE CBGP
Link failure
Change TrafficMatrix
Run RSTP Link Failure Simulation
Run RSTP Node Failure Simulation
Compose events
Amsterdam
Copenhagen
London
Luxembourg
Paris
Prague
Vienna
Zurich
Nodes Links Lsps TrafficMatrix*

Info above Link:
None

Links colors shows :

○ Link status

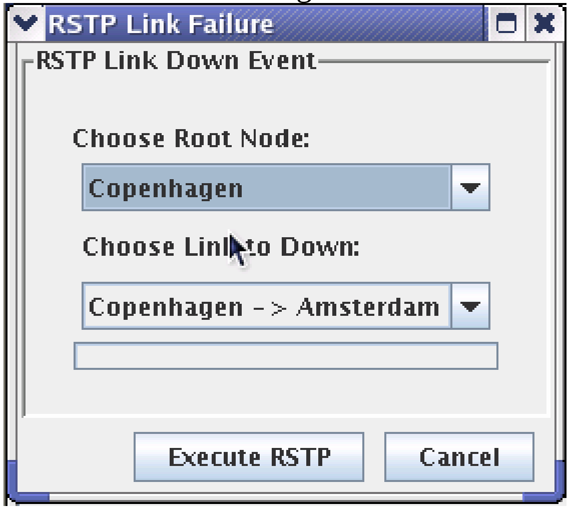
● Reservation

Legend

■ Link Down
■ < 5%
■ 5 - 10 %
■ 10 - 20 %
■ 20 - 30 %
■ 30 - 40 %
■ 40 - 50 %
■ 50 - 60 %
■ 60 - 70 %
■ 70 - 80 %
■ 80 - 90 %
■ 90 - 100 %

Node Id	Node description	Node longitude	Node latitude
Copenhagen		0	0
London		0	0
Amsterdam		0	0
Berlin		0	0
Brussels		0	0
Luxembourg		0	0
Prague		0	0

17. Select the root bridge and link whose failure needs to be simulated

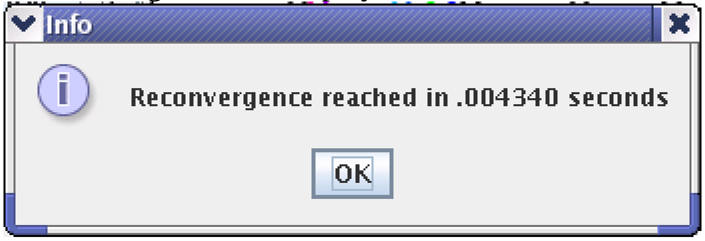


RSTP Link Down Event

Choose Root Node: Copenhagen

Choose Link to Down: Copenhagen -> Amsterdam

18. A dialog box will display simulation results



Reconvergence reached in .004340 seconds

19. Similarly, you may run RSTP node failure simulations

Building and running the topology and traffic matrix generators

Based on a plain text topology file such as those provided by Siemens and included in the tar ball, this toolset generates TOTEM compatible XML topology files and corresponding traffic matrices

1. Change to /usr/src and run the command “mkdir topgen”
2. Copy the topgen.tar tar ball to /usr/src/topgen directory
3. Extract the topgen.tar archive using “tar –xvf topgen.tar”
4. Build the topology file generator using “gcc –o ex ex.c”
5. Change to the trafficgen directory and run the command “gcc –o trafficgen trafficgen.c”
6. Run these programs and follow the self-explanatory instructions