University of Wuerzburg
Department of Distributed Systems
Prof. Dr.-Ing. P. Tran-Gia

A Planning Tool to Improve the Resilience of MPLS Networks

Detection and Avoidance of Link Overload Due to Network Failures

Jens Milbrandt, Michael Menth, Korbinian Humm, Frank Lehrieder, and Stephan Seufert

Link Overload – A Problem for Internet Service Providers

Routing in MPLS-based backbone networks

- Following shortest paths
- Determined by standard routing protocols (e.g. OSPF, IS-IS)
- Setup of label-switched paths (LSPs) along shortest paths

Routing in case of network failures

- Fast and reliable rerouting required for service continuity
- Rerouting on shortest paths
  - → Slow routing protocol convergence
  - → **Possible link overload** along the traffic detour
- Rerouting using MPLS technology
  - → Fast rerouting on explicit backup LSPs
  - → **Avoidance of link overload** by optimization of explicit backup LSPs

Example for link overload

→ Link overload on shortest backup path

→ Avoidance of link overload by explicit backup LSP

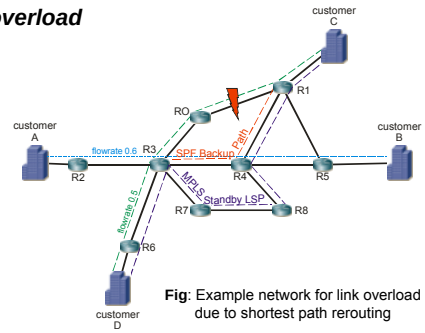


Fig: Example network for link overload due to shortest path rerouting

TES – A Tool for Traffic Engineering and Network Optimization

Tool characteristics

- JAVA-based application
- XML-based network description supporting NDL and SNDlib formats
- Flexible plugin-based architecture for easy functional extension
- Rich visualization and reporting system

Tool functions

- Miscellaneous routing calculations
- Network resilience analysis
- Explicit backup path optimization
- Comfortable editors
  - → Traffic matrix
  - → Link capacities
  - → Link metrics

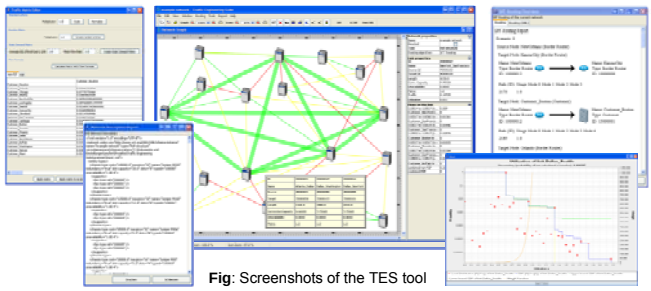


Fig: Screenshots of the TES tool

Network resilience analysis

- Calculation of link overload probabilities
- Calculation of customer disconnection probabilities
- Consideration of all probable multi-failure scenarios

Backup path optimization

- Calculation of explicitly routed backup LSPs
- Performance measures
  - → Number of rerouted aggregates
  - → Link utilization relative to shortest path rerouting

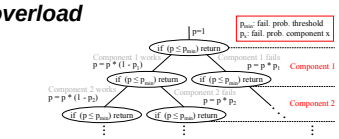
Detection of Link Overload

Genesis of link overload

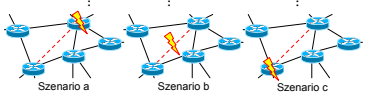
- Restoration or protection switching mechanisms triggered by network failures
- Traffic redirected over backup paths
- Normal and backup traffic on the links → link overload

A priori detection of link overload

- Determination of relevant (probable) failure scenarios



- Merging of failure scenarios in effective topologies



- Complementary distribution function of the link utilization

$$p(U(l) > u|\mathcal{S}) = \frac{1}{p(\mathcal{T})} \cdot \sum_{\{(T,c(T,l)) \in \mathcal{L}(l):\, U(T,l) > u\}} p(T)$$

- Weak spot identification → link ranking

link id	$R_{n_c}(l)$, $n_c = 0.3$	link id	$R_{n_c}(l)$, $n_c = 0.6$	link id	$R_{n_c}(l)$, $n_c = 0.9$
SaF-Sea	0.0734	LoA-SaF	0.0329	LoA-SaF	0.0324
LoA-SaF	0.0600	Den-SaF	0.0318	SaF-Sea	0.0285
Den-SaF	0.0540	SaF-Sea	0.0291	Den-SaF	0.0252

- Recommendations for infrastructure upgrades

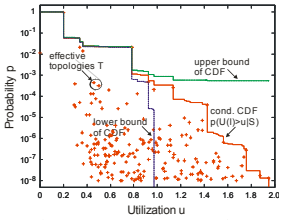


Fig: Complementary distribution function of the link utilization for link Dal→Was

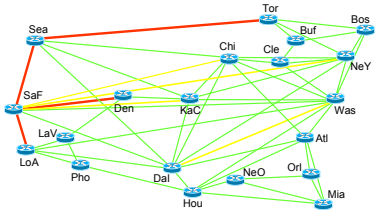


Fig: Network with indicated link overload probabilities

Avoidance of Link Overload

Heuristic to compute explicit backup LSPs for traffic flows

REPEAT

1. Determine links with potential overload
2. Sort list of reroutable border-to-border traffic aggregates according to
   - → Their bandwidth in descending order (**BW MAX**)
   - → Their bandwidth in ascending order (**BW MIN**)
   - → Their occurrence on overloaded links in descending order (**OCC**)
3. For each aggregate a in the list, create a virtual network topology
   - → Excluding all links of the primary path of aggregate a
   - → Including only links with sufficient capacity
4. Compute backup LSP for aggregate a

UNTIL (no more overloaded links exist **OR** no more aggregates can be rerouted)

Preliminary results

- Random networks with 20 nodes and increasing node degree
- Random traffic matrices
- Calculated link capacities → link utilization of 20% in failure-free case
- Number of rerouted aggregates influenced by aggregate sorting methods (BW MAX, BW MIN, OCC)
- Maximum link utilization improved relative to OSPF rerouting

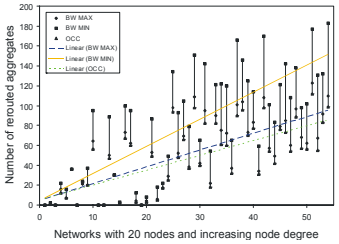


Fig: Number of rerouted aggregates depending on aggregate sorting methods

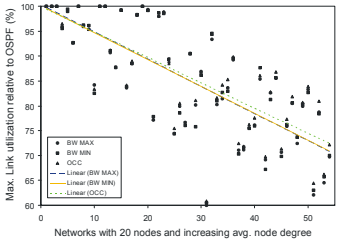


Fig: Maximum link utilization relative to OSPF