Link Failure Recovery for MPLS Networks with Multicasting

Master’s Thesis Presentation – August 12th, 2002

Yvan Pointurier
Advisor: Jörg Liebeherr

Multimedia Networks Group
Department of Computer Science
University of Virginia
Outline

- Introduction: MPLS, multicasting, tree repair problem
- A solution to the tree repair problem using graph theory
- MPLS multicast Fast Reroute mechanism
- Protocol design and implementation
- Experimental results
- Conclusion
Switching technology

- Circuit Switching
  - PSTN
- Datagram Packet Switching
  - IP
- Virtual Circuit Packet Switching
  - ATM and MPLS
Recovery from link failures

- **Traditional approaches to link failure recovery**
  - Physical layer
    - *Example: SONET ring wrapping after a failure*
    - *Pro: fast (50 ms)*
    - *Con: hardware solution ⇒ high cost*
  - Network layer
    - *Example: rerouting after IP tables convergence*
    - *Pro: software solution done by routing algorithms ⇒ no additional cost*
    - *Con: slow convergence (several minutes)*

- **MPLS goals**
  - Recovery at the network layer
  - Same recovery time as SONET using backup virtual circuits
MPLS
(Multiprotocol Label Switching)

- Provides Virtual Circuit Switching to IP networks
  - Design started in 1997 by IETF, RFC 3031 was released in 2001.

- Label
  - Identifier (20 bits) for a Virtual Circuit
  - With Ethernet and IP: a label is contained in a “shim header”
  - A label can be pushed, popped, swapped by MPLS routers
MPLS: Signaling protocol

- Protocol responsible for virtual circuits management
- Two signaling protocols have been defined in MPLS: LDP/CR-LDP and RSVP-TE
- **Example:** establishing a label “binding” between two MPLS routers

![Diagram of MPLS signaling protocol]

- Upstream node for flow “a”
- Downstream node for flow “a”
- Direction for flow “a”
- Request for binding for “a”
- Label binding for “a”
MPLS Fast Reroute

- Name given to a rerouting technique with MPLS
  - Pre-planned techniques are faster
  - MPLS Fast Reroute provides a trade-off between cost and speed
  - Objective: 50 ms rerouting time
MPLS Fast Reroute

6 steps:

1. Link failure detection
2. Link failure notification
3. Path switchover
4. Link recovery detection
5. Link recovery notification
6. Path switchback
Multicasting

- **Goal:** provide group communication

- Shared multicast routing trees

- Multicast and MPLS
  - Still in early draft stage
The tree repair problem

- Pre-planned multicast tree protection against link failure

- **Research Problem:** Where should backup paths be placed?

- **Objective:** find an algorithm to place “good” backup paths in a multicast routing tree

[Diagram of network nodes A, B, C, D showing unicast-style protection and cheaper protection with the same effect]
In this Thesis, we…

- Define a metric (resilience) which measures the degree to which a backup path protects a multicast tree
- Specify and analyze a Graph Algorithm for multicast tree protection against single link failure
- Propose protocol mechanisms (“MPLS multicast Fast Reroute”) for repair of multicast trees
- Design and implement the protocol mechanisms as extensions to MPLS signaling protocols
- Measure the performance of the implementation
Outline

- Introduction: MPLS, multicasting, tree repair problem
- A solution to the tree repair problem using graph theory
- MPLS multicast Fast Reroute mechanism
- Protocol design and implementation
- Experimental results
- Conclusion
A tree repair algorithm

- **Given:** Network, multicast routing tree and a resilience metric
- **Find:** Single backup path that maximizes the resilience of the multicast tree for single link failures

- Algorithm is presented as a graph algorithm:
  - MPLS network ⇒ graph
  - Multicast routing tree ⇒ tree embedded in the graph
Modeling process (1/2)
Modeling process (2/2)
Link failure and backup path in a tree
Formalization

- $w_l$: number of receivers downstream of a link $l$
- $f_l$: Link failure rate of link $l$
- $tdrop_l$: number of receivers dropped when a link $l$ fails

$$tdrop_l = \sum_{k \in \text{subtree}_l} w_k$$
Defining “Resilience for a tree with a backup path”

- Resilience $R(A,B)$ of a tree protected by a backup path between nodes A and B:

$$R(A, B) = R_t - R_d(A, B)$$

$$R_t = \sum_{k \in \text{tree}} (f_k \cdot tdrop_k)$$

$$R_d(A, B) = \sum_{k \in \text{tree, } k \notin \text{primary path(A,B)}} (f_k \cdot tdrop_k)$$

- $R_t$: number of receivers dropped on a link failure (no backup path)
- $R_d(A, B)$: number of receivers dropped on a link failure if a backup path is set between A and B
- $R(A, B)$: number of receivers not dropped on a single link failure if a backup path is set between A and B
Resilience of a tree protected by a backup path

- Large resilience $R(A, B) \Leftrightarrow$ more hosts are in the rearranged multicast tree on a single link failure

- We want to maximize the resilience
  - This is equivalent to minimizing $R_d$
  - Find the two nodes of the tree $A, B$ such that

$$R_d (A, B) = \min_{Y, Z \text{ nodes of the tree}} R_d (Y, Z)$$
Algorithm

1. Compute $R_d$ for all pairs of nodes
   - It can be proven that one node of the pair that maximizes the resilience is a leaf
   - $R_d$ can be computed in time $\log(n)$ with $n=$ number of nodes in the tree
2. Choose the pair of nodes that minimizes $R_d$
3. Compute the shortest path between the two nodes

- **Time complexity:** $O(n^2 \log(n) + (N-n) \log(N-n)) = O(nN \log(N))$
  - $n = $ number of nodes in the tree
  - $N = $ number of nodes in the network
- Incremental version
Outline

- Introduction: MPLS, multicasting, tree repair problem
- A solution to the tree repair problem using graph theory
- **MPLS multicast Fast Reroute mechanism**
- Protocol design and implementation
- Experimental results
- Conclusion
MPLS multicast Fast Reroute (1/3)

- A multicast virtual circuit
- The mechanism consists of 6 steps

1. Link failure detection
   - Use probes
   - Several missing probes \( \Leftrightarrow \) failure

2. Link failure notification
   - Propagate notification messages on the tree
MPLS multicast Fast Reroute (2/3)

3. Switchover
- Traffic flows on the backup path
- The tree is repaired, no node is dropped
4. Link recovery detection
   • Faster than link failure detection

5. Link recovery notification
   • Same as link failure notification

6. Switchback
   • Same topology as before the failure occurred
Outline

- Introduction: MPLS, multicasting, tree repair problem
- A solution to the tree repair problem using graph theory
- MPLS multicast Fast Reroute mechanism
- Protocol design and implementation
- Experimental results
- Conclusion
Design and implementation

- **MPLS multicast**
  - MPLS multicast is not standardized
  - Extension to MPLS-Linux unicast
  - Duplicate packets inside the Linux kernel and forward the copies on different interfaces: “mswap”
  - We provide a C API that enables users to create forwarding rules

- **MulTreeLDP signaling protocol**
  - A label distribution protocol for MPLS multicast
  - Support for Explicit Tree Routing
  - Message formats are similar to those in LDP
  - Implements MPLS multicast Fast Reroute
Hardware used for the experiments

- The Indra testbed
  - 6 PC routers, PII-450, 128MB RAM, 4x100 Mbps Ethernet, Linux
- Fast enough to duplicate packets that arrive at full speed
  - We achieved rates of 93Mbps even when each packet gets duplicated twice

1 sender

3 receivers
Testbed setup
Experimental results

- **Link failure detection**
  - Two consecutive missing probes are interpreted as a link failure
  - Distributed between 10 ms and 40 ms, average: 25.4 ms

- **Link recovery detection**
  - First probe received after link failure is interpreted as link recovery
  - Distributed between 0 ms and 10 ms, average: 5.5 ms

- **Notification**
  - Measured 1.2 ms per hop

- **Service interruption time**
  - Measured at PC5: 29.4 ms (average)
1. Experimental distribution of the service interruption times

Mean: 29.4 ms
Max: 49.6 ms
Stdev: 7.1 ms
2. Traffic received by PC5

![Graph showing traffic received over time](image)

- Traffic received (MBytes)
- Time (s)

- Traffic increased by 5 MBytes every 21 ms.
- Traffic increased by 10 MBytes every 10 ms.

---

31/33
Conclusions

Contributions
- Algorithm that selects a backup path
- MPLS multicast implementation for Linux
- Explicit Tree routing
- MPLS multicast Fast Reroute
- Design, implementation and evaluation

Future work
- Algorithm: multiple link/node failure - testing
- Standardize MulTreeLDP
- Avoid packet duplicates on switchback
- Implementation in larger networks
Questions and Answers

Thanks for your attention!

- After the Questions and Answers session, I will show a demo of MPLS multicast Fast Reroute in the MNG lab