A Cost-Effective Protection and Restoration Mechanism for Ethernet-Based Networks: an Experiment Report

Yi Lei Chung-Horng Lung
Department of Systems and Computer Engineering
Carleton University, Ottawa, Canada

Anand Srinivasan
EION Inc., Ottawa, Canada

Outline

- Motivation
- Solution
  - Link protection
  - Node protection
  - 1:n protection
- Experiment Environment
  - Open IP Environment
  - Auto-negotiation on the Ethernet
- Performance Analysis
  - Model
  - Results
- Conclusion

Motivation

- Network failure happens → Failure protection is required
- What others have done?
  - Protection in network layer is slow
  - Protection at physical layer requires dedicated hardware (expensive)
  - protection in the MPLS layer is a trade off between the performance and cost.
- Why Ethernet based MPLS protection?
  - Ethernet networks are growing rapidly
  - Is there enough published research for MPLS layer protection?

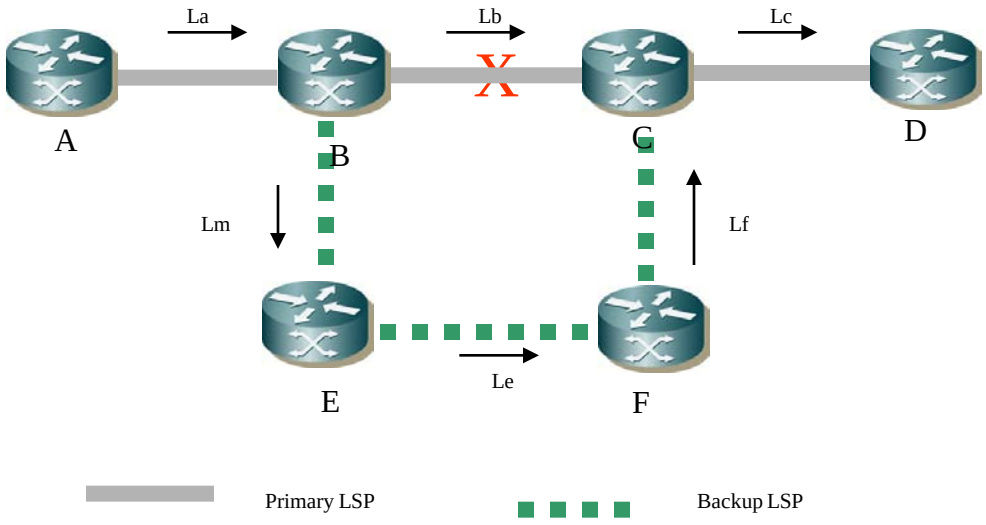
Objectives

- Provide a simple failure detection mechanism in an Ethernet-based network without the overhead of complex signaling protocols?
- Achieve a fast and reliable MPLS protection?

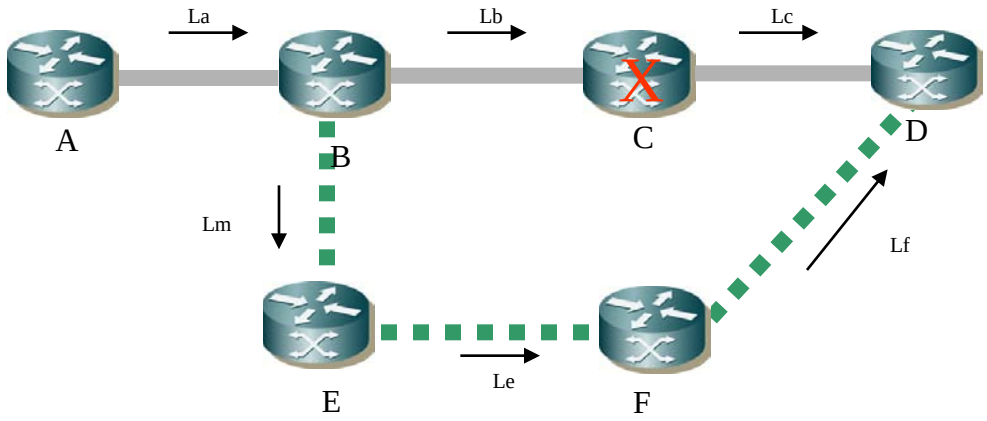
Approach

- Introduce an MPLS protection for Ethernet-based networks
- Adopt fast reroute approach
  - local repair based on pre-established LSPs
    - Link Protection
    - Node Protection
    - 1:n Protection

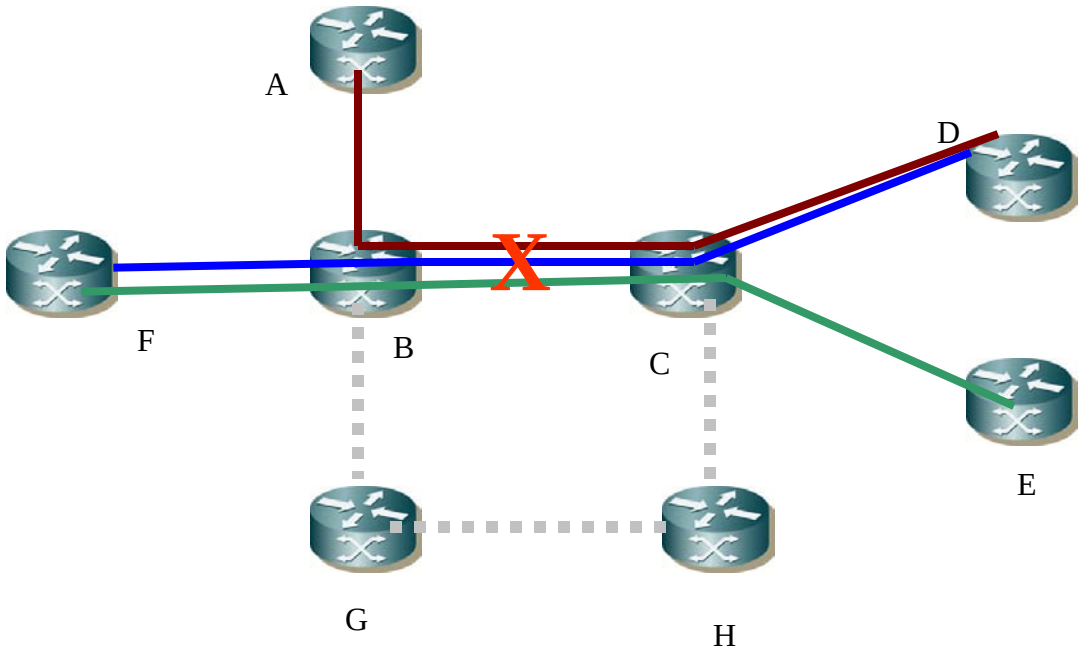
Link Protection



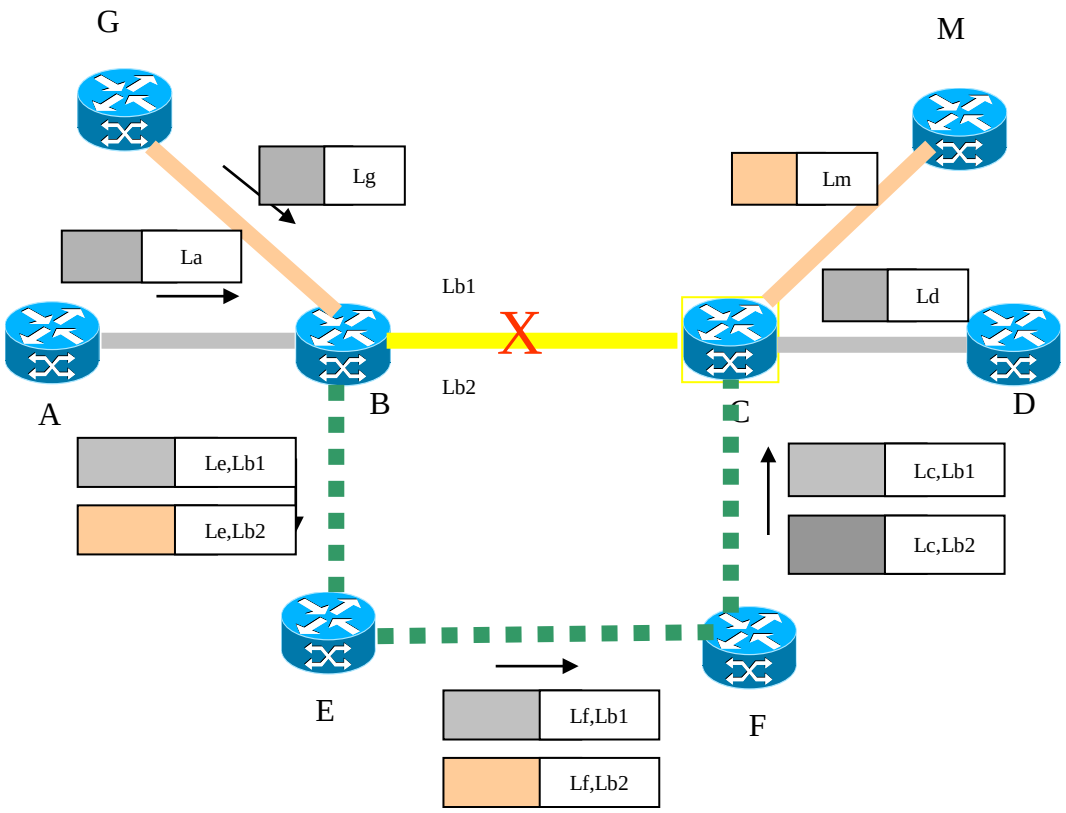
Node Protection



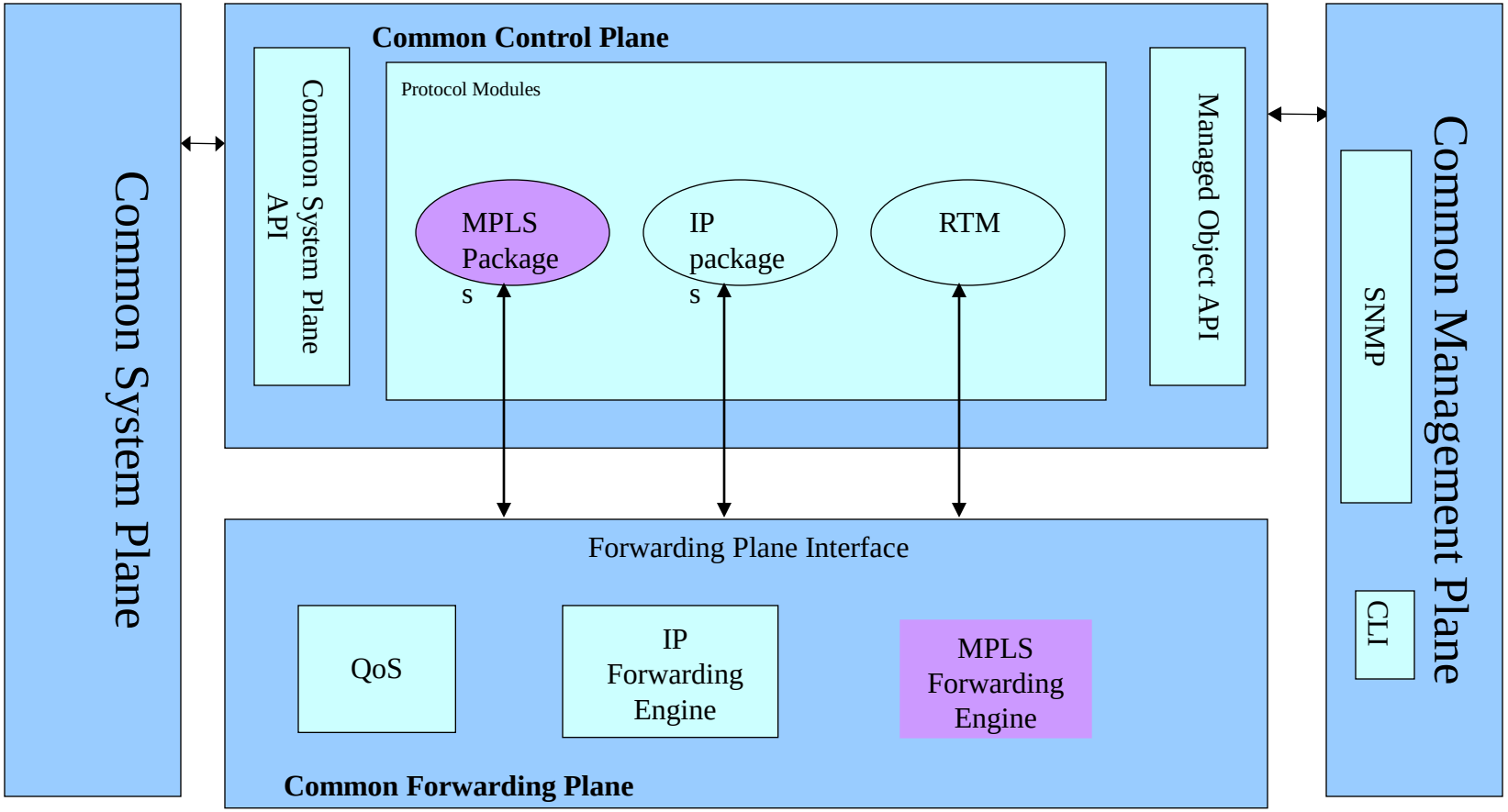
1:n Protection



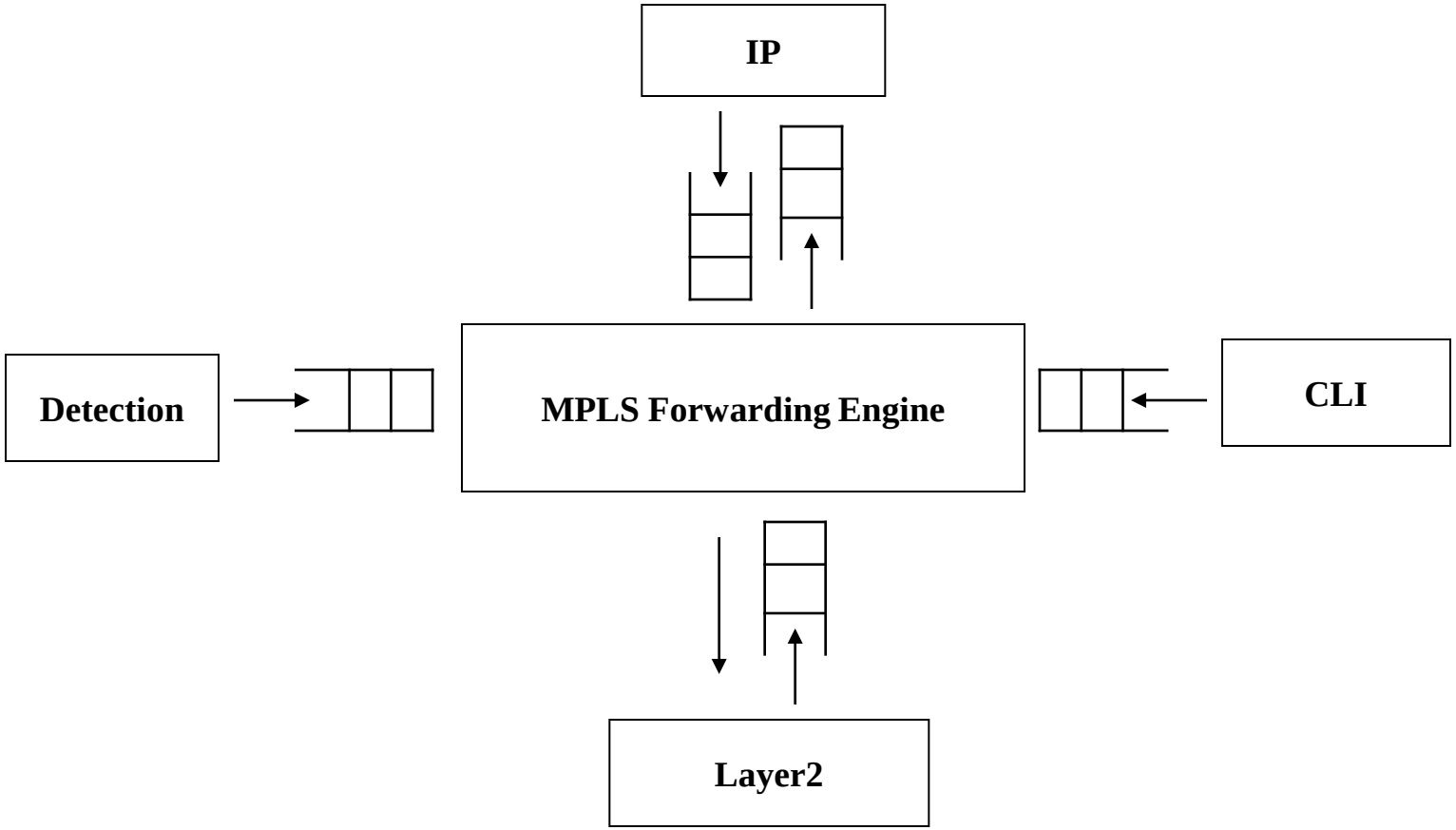
Two Levels of Label Stack



Open IP Environment



MPLS Forwarding Engine Message Flows



Design Components

- Label Space
- FEC
- Label Merging
- Data Structures
  - ip2Fec Table
  - NHLFE Table
  - FIB Table
- Message Distinction
- Command Line Interface

Failure Detection

- Signaling Protocol
- Physical Layer
- Ethernet
  - Polling Mechanism & Auto-Negotiation

Failure Recovery Steps

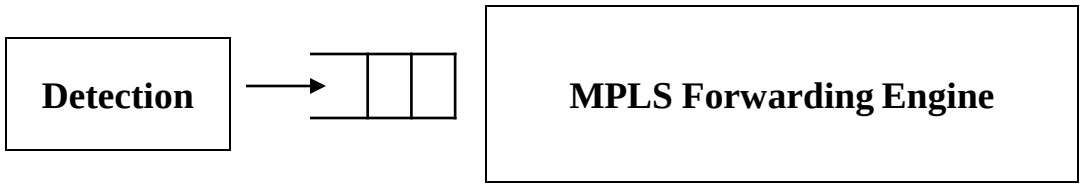
- Failure Detection
- Failure Notification
- Switchover

Auto-Negotiation

- Ethernet specific
- Monitor the interface
- Interoperable between IEEE 802.3 LANs
- Supports 10Base-T, 10Base-T Full Duplex, 100Base-TX, 100Base-TX Full Duplex, and 100Base-T4.
- Uses Fast Link Pulse (FLP) signals.
  - FLP bursts occur at the interval of 16.8 ms with a duration of 2ms.

Failure Notification

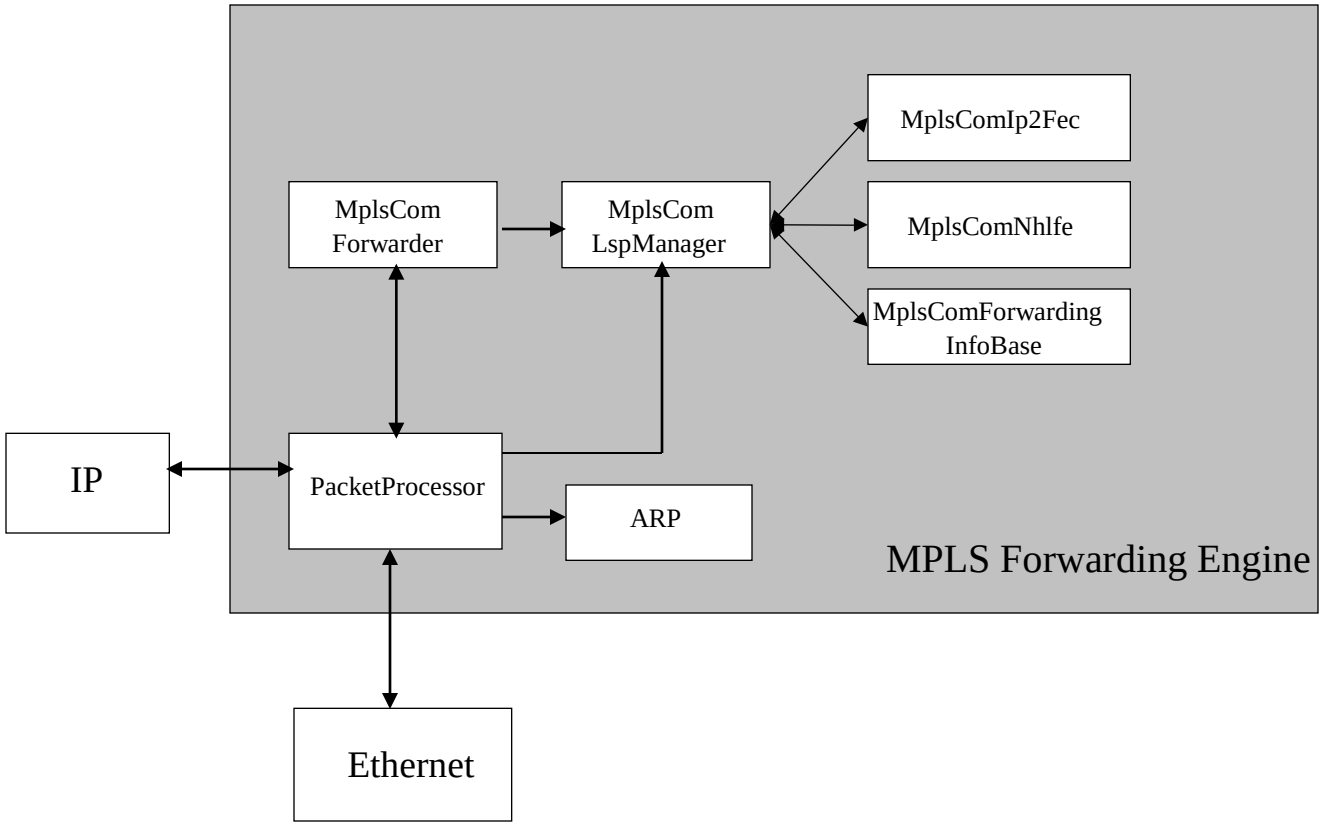
- Constructing a Failure Notification Message
- Delegating the message to the forwarding Engine



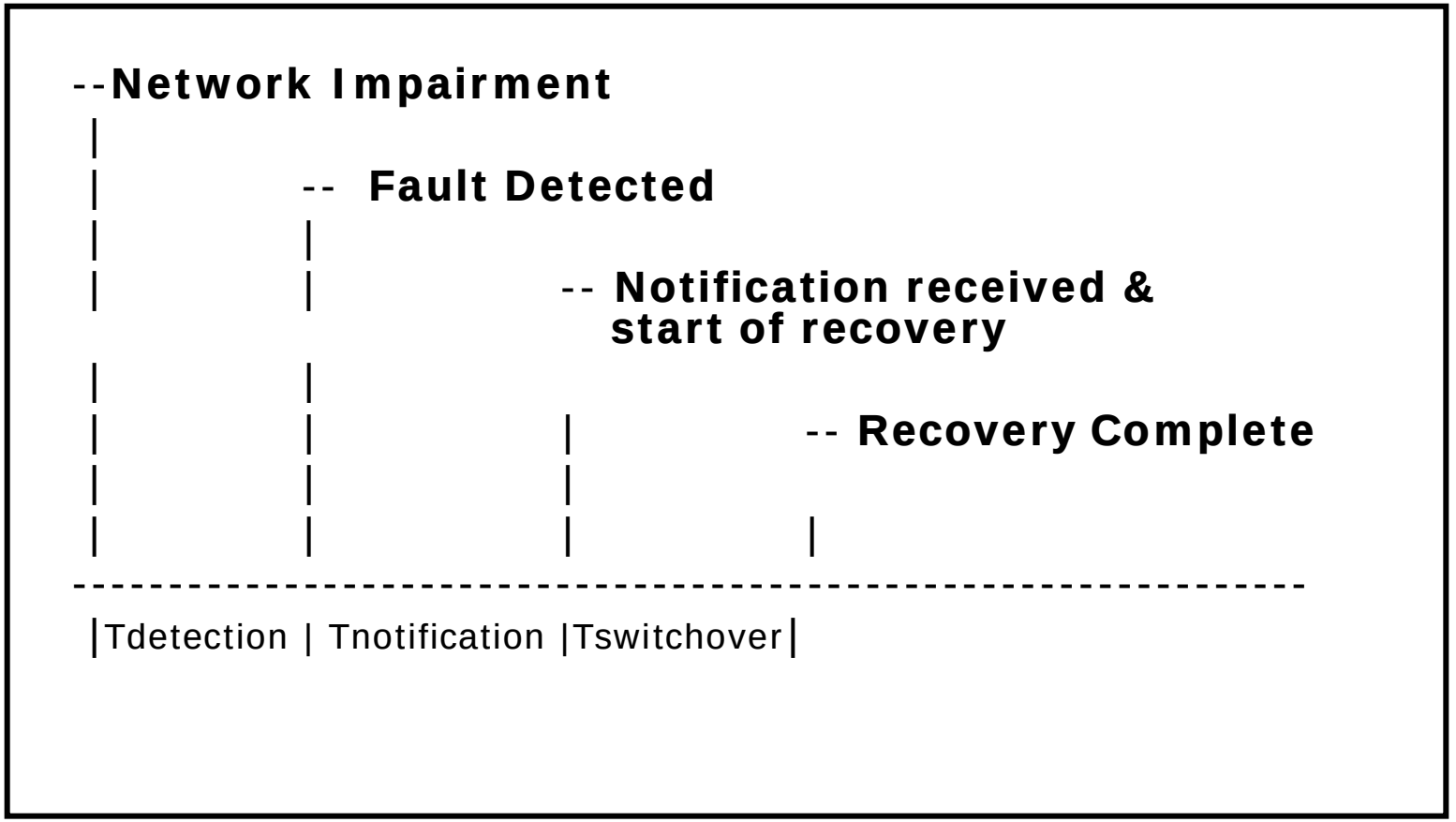
Switching over Process

- Extracting information from the Failure Notification Message
- Recognizing the failed primary LSPs
- Recognizing the associated backup LSP
- Switching over from the failed Primary LSP to its backup LSP

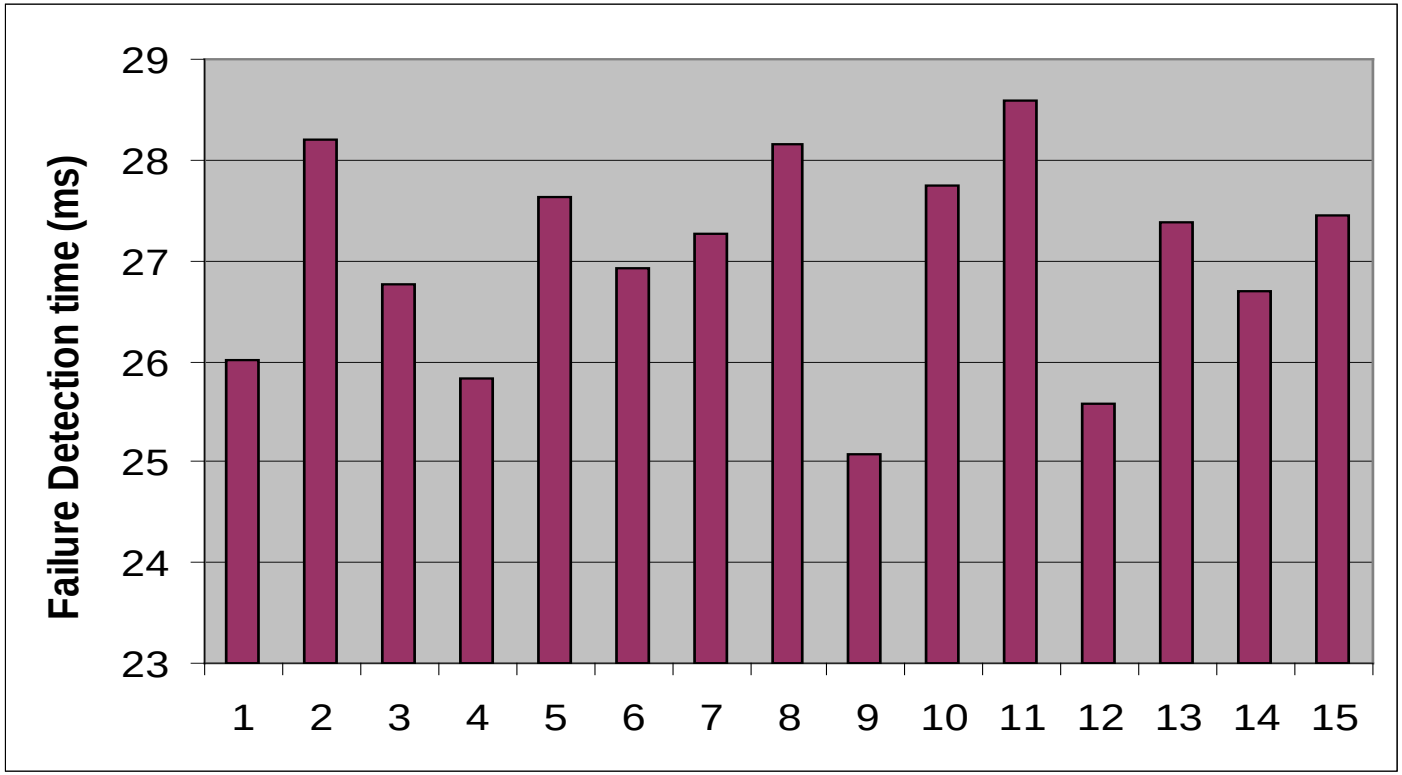
Forwarding Engine & Components



Performance Analysis Model

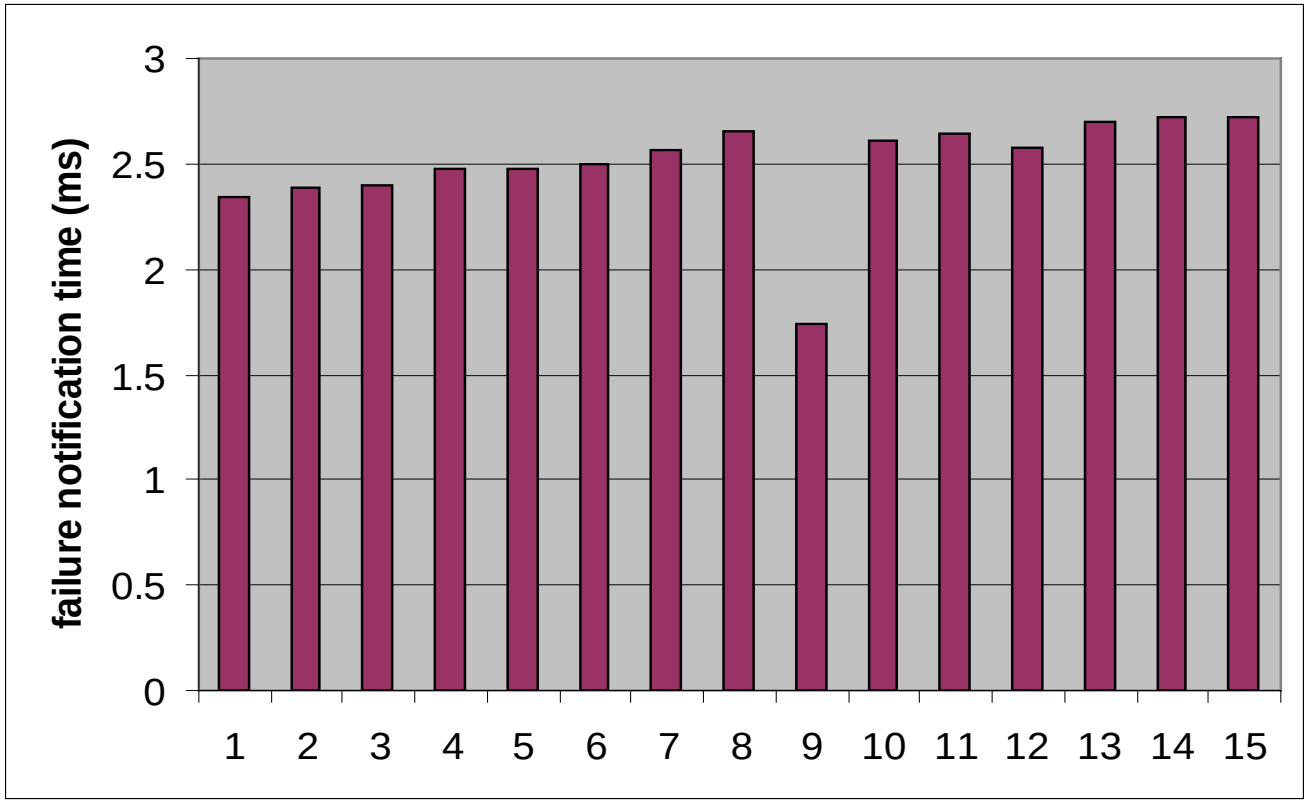


Performance Analysis Result: Failure Detection Time



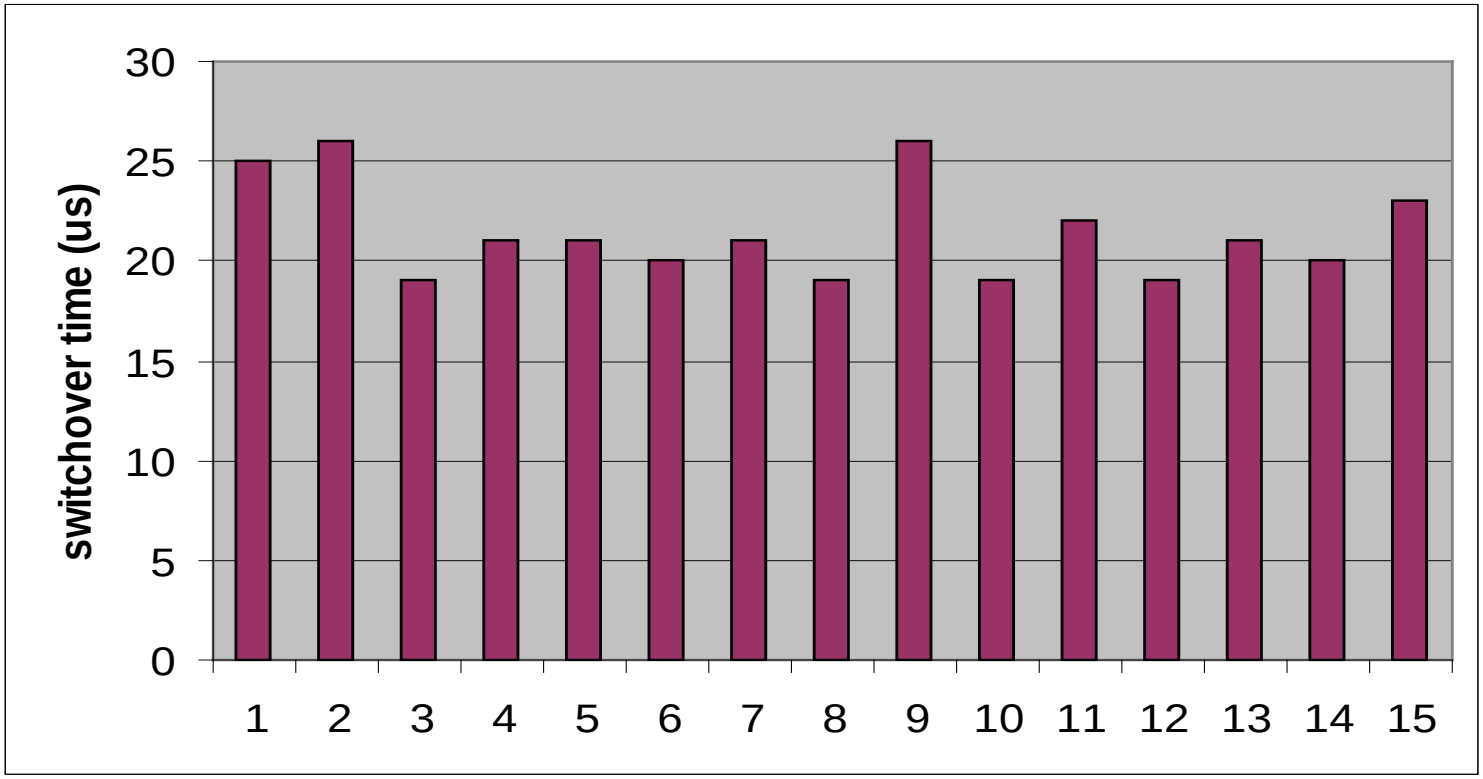
Average Failure Detection Time: 27.021ms.
Confidence Interval: 27.02±0.522.

Performance Analysis Result: Failure Notification Time



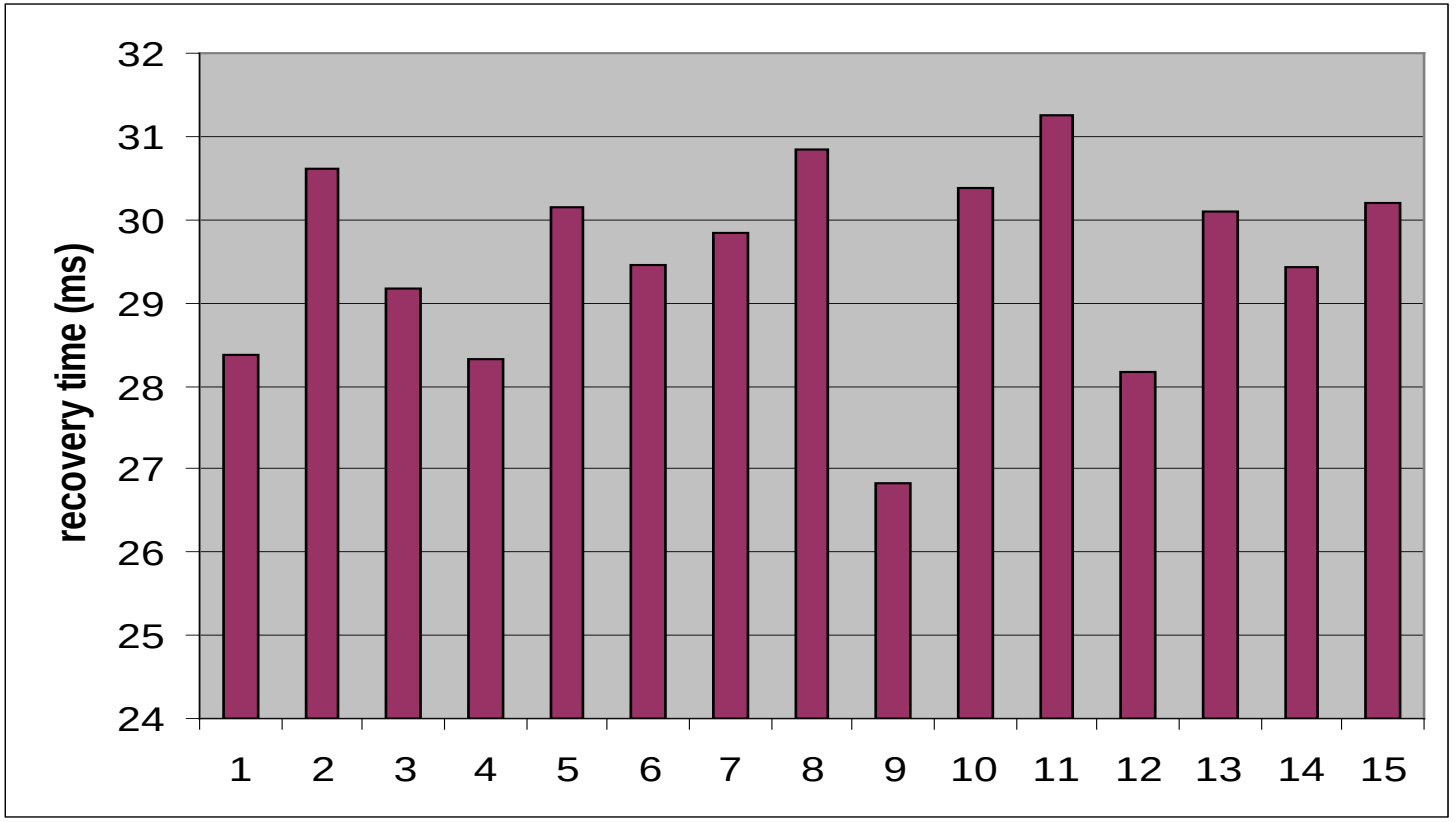
Average Failure Notification Time: 2.501ms.
Confidence Interval: 2.501±0.124

Performance Analysis Result: Swicthover Time



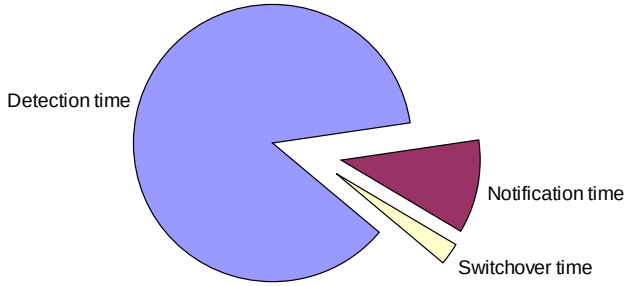
Average Switchover Time: 21.467us.
Confidence Interval: 21.47±1.252.

Performance Analysis Result: Total Recover Time



Average Recovery Time:29.542ms.
Confidence Interval: 29.54±0.602.

Performance Analysis Result: Total Recover Time Percentage



	Average time	Weight
Detection time	27.02ms	91.47%
Notification time	2.5ms	8.45%
Switchover time	0.0215ms	0.08%
Recovery time	29.54ms	100%

Conclusion

- Developed a methodology to support three types of protection (Link/Node) for Ethernet-based networks.
- Link/Node protection is achieved by integrating a simple efficient failure detection method with MPLS.
- The failure detection mechanism is protocol independent and easy to deploy.
- The analysis shows that the entire failure recovery process requires an average of 29.54 ms.

Future Research

- Failure detection enhancement could be the focus of a future research, e.g., by employing more efficient mechanisms rather than the polling mechanism.
- Build it into the kernel space and increase the priority.