MPLS Performance Engineering & Linux based Emulator

Abhay Karandikar
Associate Professor
Department of Electrical Engineering
IIT Bombay, Mumbai
karandi@ee.iitb.ac.in

Agenda of Talk

- MPLS Traffic Engineering
- QoS and MPLS
- Architecture for MPLS- TE and DiffServ QoS
- Towards a MPLS-TE Server
  - MPLS Emulator
    - Architecture of Linux based emulator
  - Network provisioning Tool
  - MPLS Protocol Development Environment

Next Generation Internet Access

- Last Mile
  - Ethernet
- Metro
  - Next Generation SONET/SDH
  - Optical Ethernet
  - MPLS as transport mechanism
- Core
  - DWDM based Intelligent optical network
    - MPLS as control plane

Performance Requirements of Customer

- Specified through SLA and TCA
- Typical Service Level Specifications
  - Mean Time between service outages
  - Mean Time to Repair outage
  - Maximum/Mean duration of outage
  - Minimum and Sustained Bandwidth
  - Packet Level Performance
    - Packet Delay
    - Jitter
    - Packet Loss
    - Out of Profile Traffic treatment

Network Level Performance

- Load Balancing
- Link Utilization and Congestion
- Path Protection and Restoration Capability
- Rapid Dynamic Provisioning Capability
- Throughput and Fairness

MPLS Traffic Engineering Positioning

- Can operate at multiple time scales
  - Longer Time Scales
    - Global Network Optimization
  - Shorter Time Scales
    - Dynamic Resource Management

MPLS Traffic Engineering

- Explicit Route Computation
  - Constrained based Routing
    - TE Constraints and Network State
- Signaling mechanism to establish TE Trunks/LSP
  - RSVP-TE
  - CR-LDP
- MPLS-TE Traffic Trunk Attributes
  - Bandwidth
  - Resource class affinity
  - Path Selection Policy
  - Priority/Preemption
  - Resilience

QoS in Internet- Differentiated Services

- Traffic flows aggregated into few flows
- Edge router classifies the packets into DiffServ classes (Behavior Aggregates)
- Classes are encoded in DSCP
- DSCP identifies Per Hop Behavior (PHB)
- DiffServ PHB
  - Expedited Forwarding
    - Low Latency, Low Class
  - Assured Forwarding
    - Four classes and three drop precedences

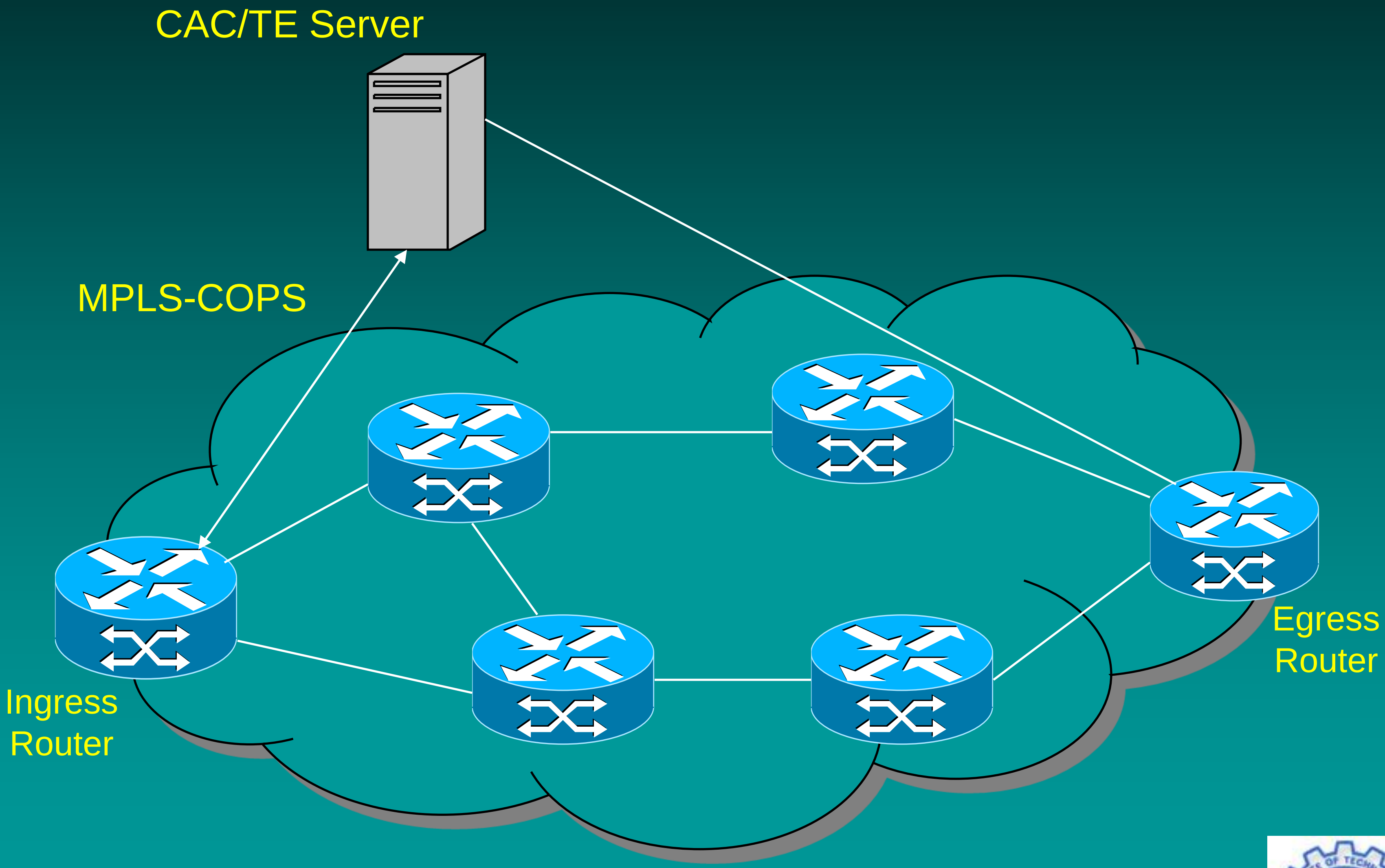
MPLS and DiffServ

- E-LSP
  - PHB determined from 3 EXP bits
  - Up to eight PHB per LSP
- L-LSP
  - Packets of different PHB but of same PHB scheduling class (PSC) mapped to a LSP
  - PSC is is signaled at LSP setup
  - PHB determined from Label and EXP bits

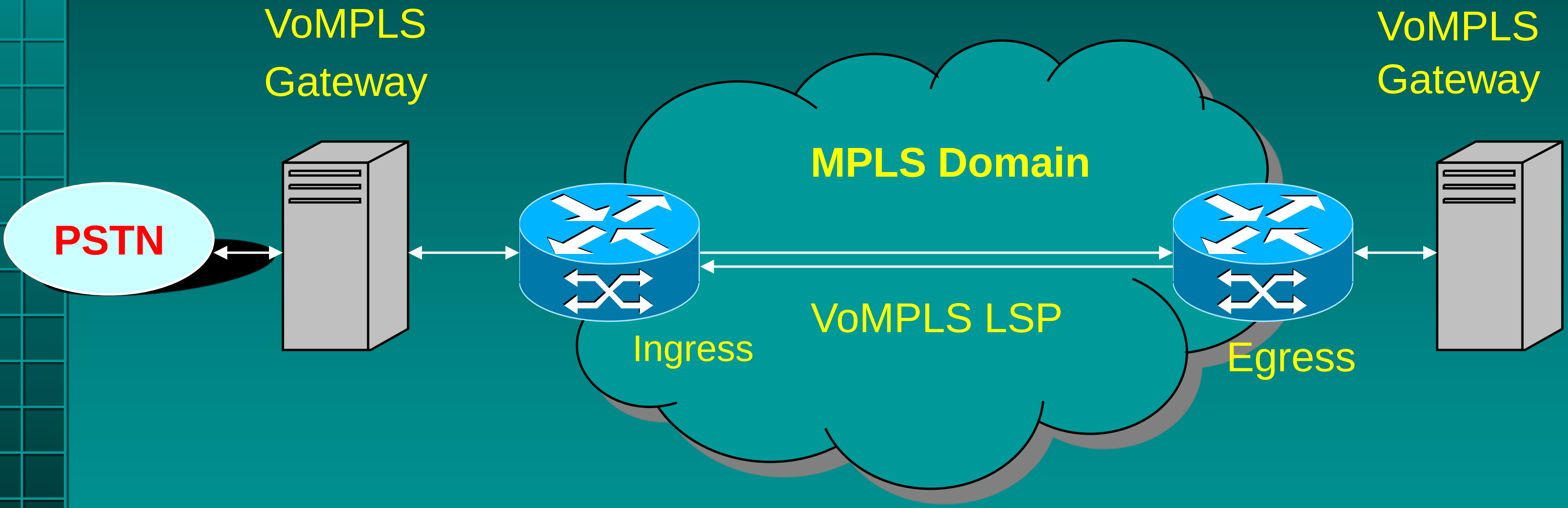
An Architecture for Providing QoS/Traffic Engineering

- Centralized Traffic Engineering Server
  - Bandwidth Broker, Traffic Engineering Engine, Policy Server
- SLS enforced at Ingress Router through MPLS COPS
  - MPLS management and Device Configuration
- Network Performance Monitoring
  - SNMP

Architecture for MPLS-TE/Qos



Voice over MPLS



Functional Model of a Centralized TE Server

- Traffic Engineering QoS Data Base
  - Network topology
  - Link state information
  - Traffic demands
  - Queuing policy at nodes
- Network State Determination Interface
  - Active network monitoring
  - Extensions to OSPF/IS-IS

TE Server…..

- Traffic Forecasting Engine
  - Traffic demands through SLA
  - Traffic Prediction Model
    - Time series model for traffic forecasting
- Route Compute Engine
  - Offline
    - Network Planning Time scale
  - Online
    - Dynamic Route Optimization

MPLS Performance Monitoring

- SLA Verification, accounting and billing
- Testing integrity of LSP
- Fault detection and fault isolation
- Packet level performance measurements
  - bit error rate
  - packet loss, delay and jitter
  - bandwidth

Towards a Traffic Engineering Server

- The objective is to develop a Linux based Network Provisioning and Traffic Engineering Tool
- The initial development has been in the area of Linux based MPLS Emulator (LiME)
  - Extended to Offline Traffic Engineering Tool
  - Finally, Traffic Engineering Server
  - Leverages on the Multithreaded implementation of LDP and CR-LDP and Switching engine in Linux kernel developed at IIT, Bombay
  - MPLS protocol development environment (MPDE)

LiME Features

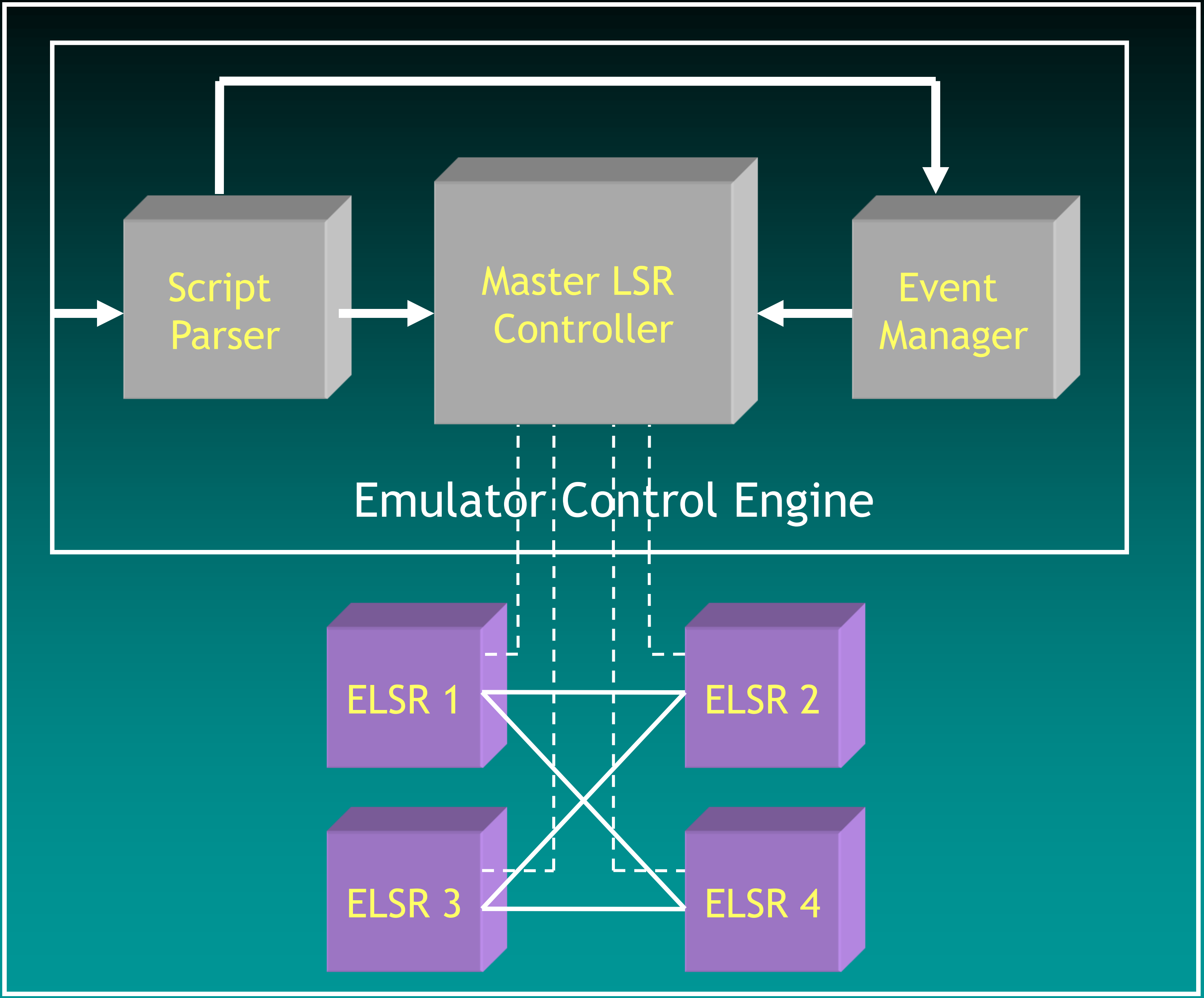
- Creates user defined network topologies and scenarios
- Add/change/configure nodes/links characteristics
- Test the effects of different events in the network
  - Fault insertion
- Performance monitoring and statistics gathering abilities
  - LSP statistics
  - Route convergence time
  - LSP trace
  - Performance of traffic engineering
- Graphical User Interface

Functional Modules of LiME

- Emulation of Network
  - Network Topology and Link characteristics
  - Routing behavior
- Emulation of LSR/LER
  - Packet classification Module
  - Control Protocol Module
  - Forwarding Information Base
  - Label switching module

Components of LiME

- Emulator Control Engine
  - User Interface
  - Master LSR Controller
  - Event Manager
- Emulated LSR
  - Control Path
  - Data Path
- MPLS Device
  - An abstract device to emulate network interface



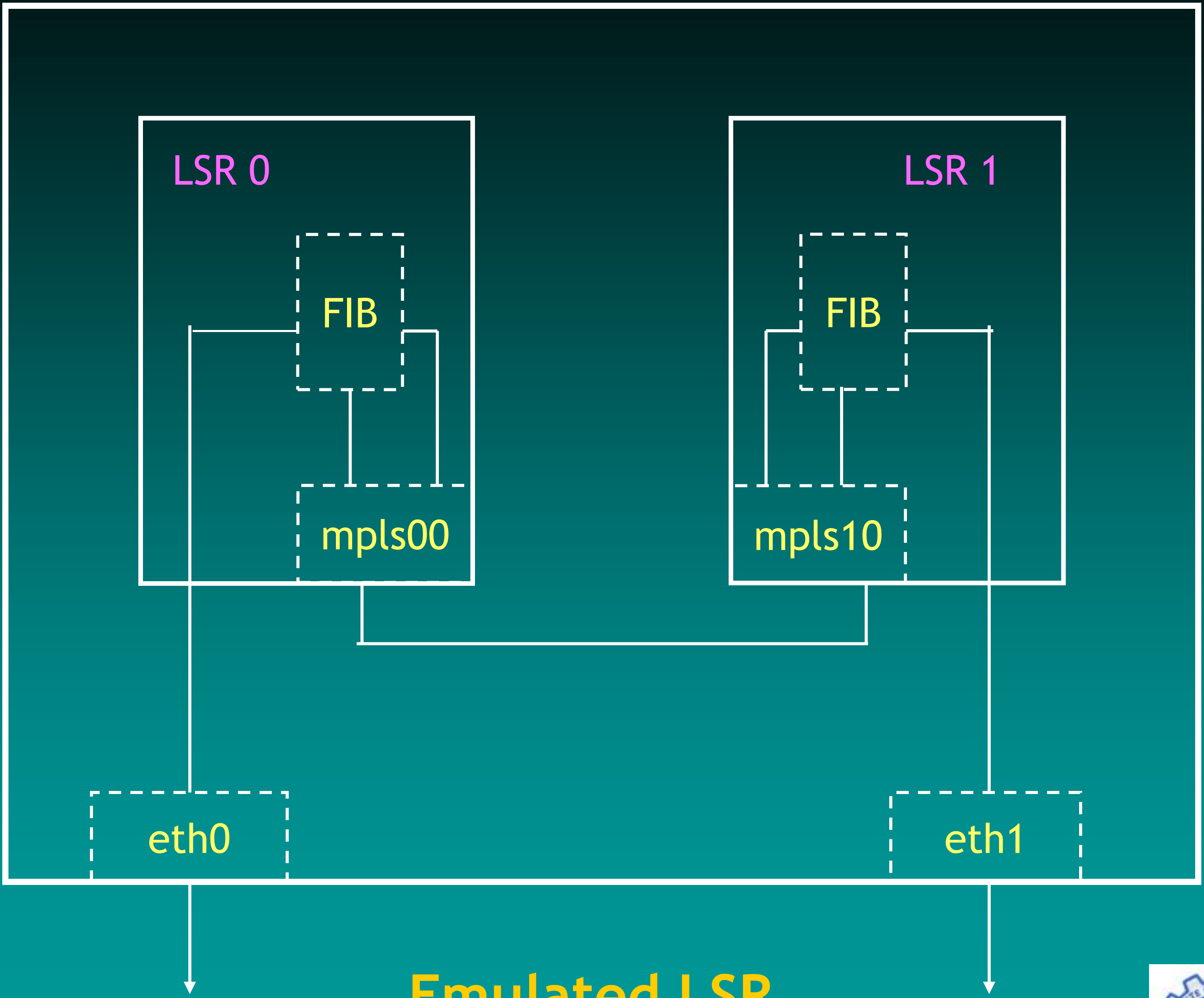
Top Level Design of LiME

Emulator Control Engine

- User Interface
  - The network topology supplied through user interface
  - Edit network scenarios
- Master LSR Controller
  - Parses the topology information
  - Controls the behavior of emulated LSR
  - Maintains overall state and timing information
- Event Manager
  - Registers the network events like link and node failures

Emulated LSR

- Control Path
  - LDP/CR-LDP/RSVP-TE daemon (multithreaded)
    - State maintenance thread
    - I/O thread
    - Timer management thread
- Data Path
  - Switching engine implemented in Linux kernel
  - Multiple LSRs are emulated by making the kernel code re-entrant



Emulated LSR

MPLS Device

- Emulates network interfaces and required for communications between two emulated LSRs
- Implemented as loadable kernel module
- Configurable from user-space
- Compatible with implementation of hardware devices like eth
- For every emulated LSR, one or more MPLS devices may be present

Issues in MPLS/IP forwarding in LiME

- Same instance of MPLS/IP stack is used for every emulated LSR
- The code is made re-entrant
- MPLS Device as point of entry into MPLS stack
- LSR, which is supposed to be processing packet at a given instance is inferred from MPLS device on which the packet is received.
- A separate copy of FIB is maintained for every emulated LSR.
  - Multiple Routing Tables
  - Rules in FIB

Modifications to IP FIB in Linux

- Exploits multiple routing tables of Linux
- The ‘main’ routing table in Linux kernel not used.
- Additional route type of `RTN_LIME_LOCAL`. An entry for IP address of every MPLS device, is maintained in ‘local’ routing table with this type.
- Lookup in the ‘local’ routing table is disabled and instead an entry with type `RTN_LOCAL` is maintained in the routing table associated with LSR.
- ‘local’ routing table is consulted only when source address for outgoing packet is to be assigned.

MPLS Forwarding Engine in LiME

- The network layer interface contains FEC table
- FEC table is now an array indexed by LSR identifier
- Incoming Label Map (ILM) is two dimensional array
- Similar to IP FIB, we define MPLS rules that are implemented to determine `lsrid` when a packet is received on a MPLS device.

MPLS Device – Implementation details

- A network device structure associated with every device known to the kernel.
  - Device name
  - Device hardware information
  - Network protocol specific device information
  - Device call back functions
  - Device flags, MTU etc.
  - Device Private Data
- Functional Description
  - Module maintenance functions
  - Device call-back functions

MPLS Subnet

- Emulates multiple access network like Ethernet in LiME.
- Achieves delivery of broadcast and multicast packets in the LiME environment.
- A List of subnets in the emulated topology can be configured from the user-space.

LiME Configuration

- ‘ip’ utility is used as interface with the IP FIB in the kernel to manage routing tables and FIB rules.
- The interface with the MPLS forwarding engine is based on Netlink sockets.
- User defined toplogy is entered through Tcl/Tk script

Future Scope

- Network provisioning tool
- Traffic Engineering tool
  - Interface with network monitoring tool
  - Traffic prediction algorithms
  - Compute TE constrained LSP using offline computations
- Traffic Engineering server
  - Ingress device configuration
  - Policy management

LiME Release

LiME release available from

www.ee.iitb.ac.in/uma/~mpls/

Primary Contributors

Abhijit Gadgil

Praveen Kumar

Acknowledgements

- Abhijit Gadgil
- Abhishek Jain
- Shruti Mahajan