GMPLS networks

Malathi Veeraraghavan
Professor
Charles L. Brown Dept. of Electrical & Computer Engineering
University of Virginia
mvee@virginia.edu

Tutorial at IEEE Globecom 2007

Acknowledgment to co-authors

• Postdoc:
  - Tao Li: LMP and OTN

• Graduate students:
  - Helali Bhuiyan: OSPF-TE
  - Xiuduan Fang: GFP, VCAT, LCAS
  - Mark McGinley: MPLS, PWE
  - Xiangfei Zhu: RSVP, Cheetah
Outline

• Principles
  - Different types of connection-oriented networks
• Technologies
  - Single network
  - Internetworking
• Usage
  - Commercial networks
  - Research & Education Networks (REN)

Principles

• Packet-switched vs. circuit switched networks
• Connection-oriented vs. connectionless modes of bandwidth sharing
• Analytical models
A model for a single network

- Hosts represent data sources and sinks
- A switch moves data units from one link to another
  - enable sharing of a link’s bandwidth
- My definition of "switch:" the multiplexing scheme is the same on all the links (e.g., a network of SONET TDM switches or a network of Ethernet packet switches)

Types of switched networks

<table>
<thead>
<tr>
<th>Switching type (Networking type)</th>
<th>Circuit-switched (CS)</th>
<th>Packet-switched (PS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectionless (CL)</td>
<td>Not an option</td>
<td>e.g., switched Ethernet networks</td>
</tr>
<tr>
<td>Connection-oriented (CO)</td>
<td>e.g., telephone network, SONET networks</td>
<td>e.g., MultiProtocol Label Switching (MPLS)</td>
</tr>
</tbody>
</table>
Circuit switch vs. packet switch

- Depends upon multiplexing technique used on interfaces
  - Position based multiplexing (circuit switch)
    • "Position" means time or frequency (wavelength)
  - Packet based multiplexing (packet switch)
    • Header-field information

Connectionless vs. connection-oriented networks

<table>
<thead>
<tr>
<th>Support function Network type</th>
<th>Addressing (in data or control plane?)</th>
<th>Routing</th>
<th>Signaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectionless (CL)</td>
<td>Data plane</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Connection-oriented Circuit Switched (CS)</td>
<td>Control plane</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Connection-oriented Packet Switched (CO PS)</td>
<td>Control plane</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Addressing

• Where are host interface addresses used:
  - In connectionless packet-switched networks, destination addresses are carried in packet headers
    • hence we place this function as being in the data plane
  - In connection-oriented (circuit/VC) networks, these addresses are used in signaling messages (needed for call setup)
    • hence we place this function as being executed in the control plane

Goal of routing algorithms

• Goal is to allow for the calls or packets to be routed on the "shortest" path, where the shortest path is determined by some metric, e.g.:
  - minimum weight path (add link weights)
  - minimum end-to-end delay
  - path with the most available bandwidth
• The algorithm should adapt to changes in
  - topology (includes administrator-set link weights)
  - reachability
  - loading conditions
Distributed routing:  
Routing protocols

- Two types:
  - Distance vector protocols
    - Each switch maintains a distance table (in addition to the routing table). The distance table shows the distances to all nodes in the network through each of its neighbors. The shortest path is then computed and the outgoing port information is stored in the routing table.
  - Link state protocols
    - The whole topology of the network is kept at each switch. Shortest path algorithms such as Dijkstra’s are then run to determine the routing tables.

Examples of routing protocols

- In Ethernet networks
  - Address learning and the spanning tree algorithm
- In the Internet:
  - Link-state routing protocols, such as Open Path Shortest First (OSPF)
  - Distance-vector based routing protocols, such as Border Gateway Protocol (BGP)
- In telephone networks:
  - Real-Time Network Routing (RTNR)
Purpose of signaling
(needed only in CO networks)

- Functions:
  - Call setup:
    - route selection
    - bandwidth reservation on each link of end-to-end connection
    - switch fabric configuration of each switch
  - Call release
    - release bandwidth for use by others

Examples of signaling protocols

- ISDN User Part of the SS7 (Signaling System No. 7) protocol stack
  - to set up and release DS0 (64kbps) circuits in a telephone (circuit-switched) network
- Resource reSerVation Protocol with Traffic Engineering (RSVP-TE)
  - used in CO PS networks such as MPLS/ATM
  - used in CS networks such as SONET/SDH and WDM
Sub-section outline

- Operation of three types of networks
  - Connectionless (CL)
  - Circuit-switched (CS)
  - Connection-oriented Packet-switched (CO-PS)

Connectionless packet-switched networks

Phase 1: Routing protocol exchanges
+ routing table precomputation

Routing table (will have other entries)

- I, II, III, IV, V: switches
- Link weights are shown next to links
- Host interface addresses are derived from switch addresses (e.g. I-A is connected to switch I)
- Example routing table entries shown at switches I, III, IV
- III-*: summarized address for all hosts connected to switch III
Connectionless (CL) packet-switched networks
Phase 2: User (data)-plane packet forwarding

- Packet header carries destination host interface address (unchanged as it passes hop by hop)
- Each CL packet switch does a route lookup to determine the outgoing next hop node or port

Sub-section outline

- Operation of three types of networks
  - Connectionless (CL)
  - Circuit-switched (CS)
  - Connection-oriented Packet-switched (CO-PS)
Circuit-switched networks
Phase 1: Routing protocol exchanges + routing table precomputation

- Same as the Phase 1 routing protocol exchanges described for connectionless (CL) packet-switched networks
- More emphasis on exchanging loading information

Circuit-switched networks
Phase 2: Signaling for call setup

Connection setup
(Dest: III-B;
BW: OC1;
Timeslot: a, 1)

Connection setup actions at each switch on the path:
1. Parse message to extract parameter values
2. Lookup routing table for next hop to reach destination
3. Read and update CAC (Connection Admission Control) table
4. Select timeslots on output port
5. Configure switch fabric: write entry into timeslot mapping table
6. Construct setup message to send to next hop
Circuit-switched networks
Phase 2: Signaling for call setup

Connection setup
(Dest: III-B;
BW: OC1;
Timeslot: a, 1)

Connection setup actions at each switch on the path:
1. Parse message to extract parameter values
2. Lookup routing table for next hop to reach destination
3. Read and update CAC (Connection Admission Control) table
4. Select timeslots on output port
5. Configure switch fabric: write entry into timeslot mapping table
6. Construct setup message to send to next hop

Routing table

CAC table

Timeslot mapping table

Update to remove timeslot 1 from available list

Perform same set of 6 connection setup steps at switch IV
write timeslot mapping table entry, update CAC table and
send connection setup message to the next hop
Circuit-switched networks
Phase 2: Signaling for call setup

Perform same set of 6 connection setup steps at switch III

Reverse setup-confirmation messages typically sent from destination through switches to source host

Circuit-switched networks
Phase 3: User-data flow

• Bits arriving at switch I on time slot 1 at port a are switched to time slot 1 of port c
Release procedure

- When a communication session ends, there is a hop-by-hop release procedure (similar to the setup procedure) to release timeslots/wavelengths for the next call

Sub-section outline

- Operation of three types of networks
  - Connectionless (CL)
  - Circuit-switched (CS)
  - Connection-oriented Packet-switched (CO-PS)
CO packet-switched (VC) networks

Phase 1: Routing protocol exchanges
+ routing table precomputation

- Same as the Phase 1 routing protocol exchanges described for connectionless (CL) packet-switched networks
- More emphasis on exchanging loading information

CO packet-switched (VC) networks

Plane 2: Signaling

Connection setup actions at each switch on the path:
1. Message parsing to extract parameter values
2. Route lookup for next hop to reach destination
3. CAC (Connection Admission Control) for BW and buffer
4. Label selection
5. Switch fabric configuration
6. Message construction to send to next hop
CO packet-switched (VC) networks
Plane 3: User-data flow

- Packets sent by host I-A with the label field in the packet header set to 1 are switched according to entries in the switch configuration tables at each switch following the path of the established virtual circuit.

Let us not confuse addresses with labels

- Addresses:
  - numbers assigned to end hosts or end host interfaces
  - globally unique
- Labels:
  - assigned to identify a virtual circuit on a link
  - unique just to the link (like seat assignments on a flight; same seat numbers can be assigned on different flights)
- Scope for confusing the two:
  - When the action performed by a packet switch is examined,
    - a connectionless switch forwards packets based on addresses
    - while a connection-oriented switch forwards packets based on labels
Rationale for VC networks

• Combine
  - QoS-guaranteed service of circuit-switched networks
  - Ability of packet-switched networks to handle bursty traffic

"Best" of both worlds

• Service guarantees to users
• High utilization: beneficial to service providers
"Worst" of both worlds

- Complexity
  - Control plane: Switch controllers need to implement signaling protocols and handle setup/release requests for bandwidth
    - Inherits complexity of circuit switch controllers
  - Data plane: Line cards need packet based demultiplexing, space switch needs to be reconfigured on a packet-by-packet basis, need buffering
    - Inherits complexity of packet switches

Principles

- Packet-switched vs. circuit switched networks
- Connection-oriented vs. connectionless modes of bandwidth sharing
  - Analytical models
Bandwidth sharing

- The very purpose for the existence for networks is to enable bandwidth sharing
- The purpose of a communication link is to move data bits from one point to another
- But the purpose of a network of links interconnected by switches is to enable the sharing of bandwidth on these links

How is bandwidth shared on a connectionless packet-switched network?

- Pre-1988 IP network:
  - Just send data without reservations or any mechanism to adjust rates
- Van Jacobson's 1988 contribution:
  - Added congestion control to TCP
  - TCP software at the sending end host adjusts its sending rate based on estimates of congestion in the router buffers
TCP throughput

\[ B = \frac{1}{RTT \sqrt{\frac{2bp}{3}} + T_0 \min(1, \frac{3bp}{8}) p(1 + 32p^2)} \]

- B: Throughput, RTT: Round-trip time
- b: an ACK is sent every b segments (b is typically 2)
- p: packet loss rate on path
- \( T_0 \): initial retransmission time out in a sequence of retries
- Interesting observation: throughput is independent of bottleneck link rate
  - congestion-avoidance algorithm model
  - for low packet loss rate, it does matter, when file size is large
- Padhye, Firouzi, Towsley, Kurose, ACM Sigcomm 98 paper

<table>
<thead>
<tr>
<th>Case</th>
<th>Input parameters</th>
<th>Mean transfer delay for a 1GB file (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Packet loss rate</td>
<td>Bottleneck link rate</td>
</tr>
<tr>
<td>1</td>
<td>0.0001</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>2</td>
<td>0.0001</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>3</td>
<td>0.0001</td>
<td>500 Mbps</td>
</tr>
<tr>
<td>4</td>
<td>0.0001</td>
<td>1Gbps</td>
</tr>
<tr>
<td>5</td>
<td>0.0001</td>
<td>500 Mbps</td>
</tr>
<tr>
<td>6</td>
<td>0.0001</td>
<td>500 Mbps</td>
</tr>
<tr>
<td>7</td>
<td>0.001</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>8</td>
<td>0.001</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>9</td>
<td>0.001</td>
<td>500 Mbps</td>
</tr>
<tr>
<td>10</td>
<td>0.001</td>
<td>1Gbps</td>
</tr>
<tr>
<td>11</td>
<td>0.001</td>
<td>500 Mbps</td>
</tr>
<tr>
<td>12</td>
<td>0.001</td>
<td>500 Mbps</td>
</tr>
<tr>
<td>13</td>
<td>0.01</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>14</td>
<td>0.01</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>15</td>
<td>0.01</td>
<td>500 Mbps</td>
</tr>
<tr>
<td>16</td>
<td>0.01</td>
<td>1Gbps</td>
</tr>
<tr>
<td>17</td>
<td>0.01</td>
<td>500 Mbps</td>
</tr>
<tr>
<td>18</td>
<td>0.01</td>
<td>500 Mbps</td>
</tr>
</tbody>
</table>

~2 Mbps

~21 Mbps
How is bandwidth shared on a circuit-switched network?

- The signaling procedure described is for immediate-request calls
- Example: telephone networks
- Send a call setup request:
  - if requested bandwidth is available, it is allocated to the call
  - if not, the call is blocked (rejected)
- \( M/G/m/m \) model:
  - \( m \): number of circuits

**ErlangB formula**

\[
\rho = \frac{\lambda}{\mu} \quad P_b = \frac{\rho^m}{m!} \sum_{k=0}^{m} \frac{\rho^k}{k!} \quad u_b = \frac{(1 - P_b) \times \rho}{m}
\]

\( \rho \): offered traffic load in Erlangs
\( \lambda \): call arrival rate
\( 1/\mu \): mean call holding time
\( m \): number of circuits
\( P_b \): call blocking probability
\( u_b \): utilization

For a 1% call blocking probability, i.e., \( P_b = 0.01 \)

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>( m )</th>
<th>( u_b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>24.8%</td>
</tr>
<tr>
<td>10</td>
<td>17</td>
<td>58.2%</td>
</tr>
<tr>
<td>100</td>
<td>117</td>
<td>84.6%</td>
</tr>
</tbody>
</table>
Delay model - to compare with TCP approach

- What happens after the call is blocked?
- If user waits and tries again, then the call does not simply go away
- A better model would be an $M/M/m/\infty$ queueing system
  - approximate, since "queueing" is distributed at the end hosts, which have no idea when to try again
  - probability of an arriving call finding all $m$ circuits busy is much higher than in call blocking model since calls linger

Impact of increasing $m$ at different values of link utilization $U_d$

- Prob. of arriving job finding all $m$ circuits busy
- Offered load: call arrival rate/call departure rate
- Link capacity expressed in channels
- High-rate per-call circuits
- Low-rate per-call circuits

$P_q=41\%$
Impact of mean call holding time, $1/\mu$

\[ E[W_d] = \frac{1}{m\mu(1-U_d)} \]

- $\lambda'$: per host call-generation rate
- $U_d$: 90%

BW sharing modes in circuit/VC networks

- Large $m$: Moderate throughput
  - Immediate-request with call blocking + retries
    - Video, gaming
  - Short calls
    - Bank teller
  - Book-ahead
  - High throughput
- Small $m$: High throughput
  - Long calls
    - Doctor's office
    - Immediate-request with delayed-start times
      - Video, gaming
      - File transfers
- Mean waiting time is proportional to mean call holding time
- Can afford to have a queueing based solution when $m$ is small if calls are short.
How is bandwidth shared on a virtual-circuit network?

- In connection-oriented packet-switched networks,
  - bandwidth allocation to a virtual circuit is independent of label selection
- In circuit-switched networks,
  - when "labels" are selected (e.g., timeslots are selected on a SONET link), it means bandwidth allocation to the circuit is immediately fixed

Savings in bandwidth allocation over circuit-switched networks

- peak bandwidth assignment $C_L = NR_p$
- QoS specified bandwidth assignment
- average bandwidth assignment

Mischa Schwartz's 1996 textbook on broadband networks
Bandwidth allocation for virtual circuits

- How is bandwidth allocated to a virtual circuit?
  - Call setup request carries
    - traffic descriptor parameters
    - desired quality-of-service parameters
  - Call admission control algorithm is executed at the switch controller to determine
    - bandwidth allocation for the virtual circuit
    - buffer space allocation for the virtual circuit

Traffic descriptors

- Peak rate
- Sustained rate (average)
- Mean Burst Size
QoS measures

- Packet Loss Ratio
- Packet Transfer Delay
- Packet Delay Variance

Traffic source model

- On-off Markov model to characterize the traffic source: fluid flow model

\[ \text{OFF} \quad \alpha \quad \text{ON} \]
\[ \beta \]

\[ \text{OFF} \quad \text{ON} \quad \text{OFF} \quad \text{ON} \quad \text{OFF} \]

time

mean: \( 1/\beta \)
mean: \( 1/\alpha \)

probability that the source is the ON state: \( p = \frac{\alpha}{\alpha + \beta} \)
Traffic descriptors values for ON-OFF model

- Peak rate = $R_p$
- Sustained rate (average) = $pR_p$
- Mean Burst Size = $R_p/\beta$

N sources instead of one source

- To compute bandwidth allocation, we set up the problem assuming N homogeneous independent sources, each of which can be represented by the same ON-OFF model (with the same parameter values)

Buffer length = $x$
"Equivalent bandwidth"

\[ C = \min(C_s, C_f) \]

- Two approximations (both conservative):
  - Stationary approximation (buffer is ignored)
  - Flow approximation (statistical multiplexing is ignored)
- Seminal paper by Guerin, Ahmadi and Naghshineh, JSAC 1991

Flow approximation

\[ C_f = \frac{C_L}{N} = R_p \left( \frac{1-k}{2} + \sqrt{\left(\frac{1-k}{2}\right)^2 + kp} \right) \]

\[ k \equiv \frac{\beta x}{R_p (1-p) \ln(1/P_L)} \]

- \( x \): buffer size
- \( P_L \): packet loss ratio
- \( R_p \): peak rate
- \( p \): probability of source being in ON state
- \( N \): number of sources
- \( 1/\beta \): mean ON-state duration
Stationary approximation

peak bandwidth assignment: \( C = NR_p \)

QoS-specified allocation: \( C_S \geq mR_p \) more than average

\[
m = pN
\]

\[
\sigma^2 = Np(1 - p)
\]

\[
p = \frac{\alpha}{(\alpha + \beta)}
\]

\[
C_S = (m + K\sigma)R_p
\]

\[
C_S = \left( m + \sigma\sqrt{-\ln(2\pi) - 2\ln\varepsilon} \right) R_p
\]

- \( m \): average number of ON sources
- \( \sigma^2 \): variance of the number of ON sources
- \( \varepsilon \): probability of being in the overload region
- Use binomial distribution to find \( m \) and \( \sigma^2 \)

\[
\varepsilon \approx P_L
\]

Example

- \( x \): buffer size = 3 Mbits
- \( P_L \): cell loss ratio = 10^{-5}
- \( R_p \): peak rate = 4 Mbps
- \( p \): ON-state probability = 0.35
- \( C_l \): Link capacity = 400Mbps
- \( 1/\beta \): mean ON-state duration = 100msec
- \( k = 0.65/(1-p) = 1 \)

\[
C_f = \frac{C_l}{N} = 0.59R_p = 2.36 \text{ Mbps}
\]

Therefore number of calls that can be admitted is:

\[
N = \left| \frac{C_l}{2.36} \right| = 169 \text{ instead of 100 (peak-rate allocation)}
\]
Need data-plane algorithms to achieve QoS guarantees

Outline

• Principles
  - Different types of connection-oriented networks

➢ Technologies
  - Single network
  - Internetworking

• Usage
  - Commercial networks
  - Research & Education Networks (REN)
Technologies

- **Connection-oriented (CO) networks**
  - Data-(user-) plane protocols
    - packet-switched: MPLS, VLAN Ethernet, Intserv IP
    - circuit-switched: SONET/SDH, WDM, SDM
  - Control-plane protocols:
    - RSVP-TE
    - OSPF-TE
    - LMP
- **Internetworking**
  - GFP, VCAT, LCAS for SONET/SDH
  - PWE3 for MPLS networks
  - Digital wrapper for OTN

### MPLS Architecture

**Label Switched Path (LSP):** Term used for virtual circuits in MPLS networks

- **Label Ingress Router (LIR):** Entry point into MPLS network: isolating packets to map to LSP
- **Label Egress Router (LER):** Exit point, removes label and routes based on native format
- **Label Switched Router (LSR):** Routers along the path that examine the top label in stack and forward accordingly
MPLS header

<table>
<thead>
<tr>
<th>Label Value</th>
<th>CoS</th>
<th>S</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Bits</td>
<td>3</td>
<td>1</td>
<td>8</td>
</tr>
</tbody>
</table>

- **Label Value**
  - (20 bits) Label used to identify the virtual circuit
- **Class of Service (CoS)**
  - (3 bits) Experimental field, Used for QoS support
- **S**
  - (1 bit) Identifies the bottom of the label stack
- **TTL**
  - (8 bits) Time-To-Live value

MPLS label stacking

- MPLS labels can be stacked
- What does this mean?
  - Create one virtual circuit (VC) on a link
  - Say we allocate 100Mbps to this VC
  - We can create another VC within this VC and allocate a portion of this 100Mbps
- Why is label stacking required?
  - Expected to be required originally for scalability
  - Most vendors support at least 4 levels (Malis’ paper)
  - Currently, this has become a useful feature for pseudo-wire services (point-to-point services) and VPNs (multipoint)

MPLS Label Stacking:
Hierarchical packet forwarding

IEEE 802.1Q Ethernet VLAN

<table>
<thead>
<tr>
<th>Dest. MAC Address</th>
<th>Source MAC Address</th>
<th>TPID</th>
<th>TCI</th>
<th>Type/Len</th>
<th>Data</th>
<th>FCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN Tag</td>
<td>802.1Q Tag Type</td>
<td>User Priority</td>
<td>CFI</td>
<td>VLAN ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Bytes</td>
<td>3 Bytes</td>
<td>1 Bit</td>
<td>12 Bits</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
VLAN Tag Fields

- Tag Protocol Identifier (TPID)
  - (2 bytes) 802.1Q Tag Protocol Type - set to 0x8100 to identify the frame as a tagged frame

- Tag Control Information (TCI)
  - User Priority
    - (3 bits) As defined in 802.1p, 3 bits represent eight priority levels
  - CFI
    - (1 bit) Canonical Format Indicator, set to indicate the presence of an Embedded-RIF
  - VLAN ID
    - (12 bits) VID uniquely identifies the frame’s VLAN

Integrated services (Intserv)
IP network

- "Label" on which switch performs its forwarding function:
  - Destination IP address
  - Source IP address
  - Protocol field in IP header: TCP or UDP
  - Destination TCP or UDP port number
  - Source TCP or UDP port number
SONET STS Frame

- SONET streams carry two types of overhead
- **Path overhead (POH):**
  - inserted & removed at the ends
  - *Synchronous Payload Envelope (SPE)* consisting of Data + POH traverses network as a single unit
- **Transport Overhead (TOH):**
  - processed at every SONET node
  - TOH occupies a portion of each SONET frame
  - TOH carries management & link integrity information

Special OH octets:
- A1, A2: Frame Sync
- B1: Parity on Previous Frame (BER monitoring)
- J0: Section trace (Connection Alive?)
- H1, H2, H3: Pointer Action
- K1, K2: Automatic Protection Switching

**STS-1 Frame**

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>J0</th>
<th>J1</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>E1</td>
<td>F1</td>
<td>B3</td>
</tr>
<tr>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>C2</td>
</tr>
<tr>
<td>H1</td>
<td>H2</td>
<td>H3</td>
<td>G1</td>
</tr>
<tr>
<td>B2</td>
<td>K1</td>
<td>K2</td>
<td>F2</td>
</tr>
<tr>
<td>D4</td>
<td>D5</td>
<td>D6</td>
<td>H4</td>
</tr>
<tr>
<td>D7</td>
<td>D8</td>
<td>D9</td>
<td>Z3</td>
</tr>
<tr>
<td>D10</td>
<td>D11</td>
<td>D12</td>
<td>Z4</td>
</tr>
<tr>
<td>S1</td>
<td>M0/1</td>
<td>E2</td>
<td>N1</td>
</tr>
</tbody>
</table>

1 column of Path OH + 8 data columns
3 Columns of Transport OH
Synchronous Payload Envelope (SPE)

Order of transmission
1. 9 rows
2. 90 columns

810 Octets per frame @ 8000 frames/sec
810x64kbps=51.84 Mbps
125 µs

Courtesy: Leon-Garcia and Widjaja's textbook
SONET/SDH rates
(number is the multiplier)

<table>
<thead>
<tr>
<th>SONET</th>
<th>SDH</th>
<th>Data rate (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Optical</td>
<td>Gross</td>
</tr>
<tr>
<td>STS-1</td>
<td>OC-1</td>
<td>51.84</td>
</tr>
<tr>
<td>STS-3</td>
<td>OC-3</td>
<td>155.52</td>
</tr>
<tr>
<td>STS-9</td>
<td>OC-9</td>
<td>466.56</td>
</tr>
<tr>
<td>STS-12</td>
<td>OC-12</td>
<td>622.08</td>
</tr>
<tr>
<td>STS-18</td>
<td>OC-18</td>
<td>933.12</td>
</tr>
<tr>
<td>STS-24</td>
<td>OC-24</td>
<td>1244.16</td>
</tr>
<tr>
<td>STS-36</td>
<td>OC-36</td>
<td>1866.24</td>
</tr>
<tr>
<td>STS-48</td>
<td>OC-48</td>
<td>2488.32</td>
</tr>
<tr>
<td>STS-192</td>
<td>OC-192</td>
<td>9953.28</td>
</tr>
</tbody>
</table>

Example: An OC48 frame has 48 x 90 columns in 125 µs

Optical transport networks

- ITU-T G.872 specifies an optical transport network (OTN) architecture, which defines two interface classes
  - Inter-domain interface (IrDI): interface between operators/vendors; defined with 3R processing (retiming, reshaping, and regeneration)
  - Intra-domain interface (IaDI): interface within an operator/vendor domain
- ITU-T G.709 is about the information transferred across IrDI and IaDI interfaces
  - Defines several layers in the OTN hierarchy
**Objective and features**

- Need to support the transmission needs of today’s diverse digital services on optical links
- Need to equip DWDM equipment with operational, administration, and maintenance functionalities, similar to those seen in SONET/SDH
- Advantages relative to SONET/SDH
  - Management of optical signals in the optical domain
    - without O/E/O conversion
  - Transparent transport of client signals
  - Stronger Forward Error Correction (FEC)
- G. 872 layers
  - OTS: Optical Transmission Section
  - OMS: Optical Multiplex Section
  - OCh: Optical Channel

---

**Layers within an OTN**

- Optical line amplifier (OTS termination)
- Optical cross connect/drop/terminal mux (OMS termination)
- 3-R regeneration (OCh, OTU termination)
- Client access (ODU termination)

Figure 12 OTN Network Layers

Courtesy: T. Walker's tutorial
**OTN Hierarchy**

- **Electrical domain:**
  - OTU: Optical Channel Transport Unit
  - ODU: Optical Channel Data Unit
  - OPU: Optical Channel Payload Unit

- **Optical channel (OCh) overhead:** support operations, administration, and maintenance functions

- **OCh payload:** can be STM-N, ATM, IP, Ethernet, GFP frames, OTN ODUk, etc.

- **FEC:** Reed-Solomon RS(255, 239) code recommended; roughly introduces a 6.7% overhead

- **Frame size:** 4 rows of 4080 bytes

- **Frame period:**
  - OTU1 - 48.971 µs (payload data rate: roughly 2.488 Gbps)
  - OTU2 - 12.191 µs (payload data rate: roughly 9.995 Gbps)
  - OTU3 - 3.035 µs (payload data rate: roughly 40.15 Gbps)
References for OTN

- ITU-T G. 872 and G.709/Y.1331 Specifications

Technologies

- Connection-oriented (CO) networks
  - Data-(user-) plane protocols
    - packet-switched: MPLS, VLAN Ethernet, Intserv IP
    - circuit-switched: SONET/SDH, WDM, SDM
  - Control-plane protocols:
    - RSVP-TE: signaling protocol
    - OSPF-TE: routing protocol
    - LMP
- Internetworking
  - PWE for MPLS networks
  - GFP, VCAT, LCAS for SONET/SDH
  - Digital wrapper for OTN
The evolution of Resource reSerVation Protocol (RSVP)

- RSVP (RFC2205, 1997)
- RSVP-TE (RFC 3209, 2001)
- RSVP-TE GMPLS Extension (RFC 3471, 3473, 2003)

RSVP-RFC 2205

- Designed to support integrated services on the Internet
- Reserve resources to meet required QoS measures for a data flow
- Seven messages:
  - Path, Resv, PathErr, ResvErr, PathTear, ResvTear, and ResvConf (triggered by an optional object, RESV_CONFIRM, in Resv messages)
- All messages begin with a common header, followed by a body consisting of a variable number of “objects”
- Common header format

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vers</td>
<td>Protocol version</td>
</tr>
<tr>
<td>Flags</td>
<td>Message type</td>
</tr>
<tr>
<td>Msg Type</td>
<td>Message type</td>
</tr>
<tr>
<td>RSVP Checksum</td>
<td>Protocol checksum</td>
</tr>
<tr>
<td>Send_TTL</td>
<td>Protocol length</td>
</tr>
<tr>
<td>(Reserved)</td>
<td>(Reserved)</td>
</tr>
<tr>
<td>RSVP Length</td>
<td>Protocol length</td>
</tr>
</tbody>
</table>
Path message

• Three mandatory objects
  - SESSION
    • Carries the destination address of an LSP
  - RSVP_HOP
    • Used to identify the GMPLS neighbor node (sender/receiver of signaling message)
  - TIME_VALUES
    • Set refresh timer

• Optional objects
  - SENDER_TEMPLATE
    • Carries the source address of an LSP
  - SENDER_TSPEC
    • Carries traffic descriptor parameters (IntServ Tspec)

Key objects: destination and label

• Session object:
  - Carries the destination IP address
  - IP protocol type field (TCP or UDP)
  - Destination TCP or UDP port number

• Sender-template object
  - Carries the source IP address
  - Source TCP or UDP port number
Key objects: traffic descriptor and QoS metrics

- Sender Tspec: Traffic descriptor
- AdSpec: QoS metrics

IntServ Tspec (RFC 2210)

Object format:

```
<table>
<thead>
<tr>
<th>31</th>
<th>24 23</th>
<th>16 15</th>
<th>8  7</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 [a]</td>
<td>reserved</td>
<td>7 [b]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1 [c]</td>
<td>0 [reserved]</td>
<td>6 [d]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>127 [e]</td>
<td>0 [f]</td>
<td>5 [g]</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Token Bucket Rate [z] (32-bit IEEE floating point number)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Token Bucket Size [b] (32-bit IEEE floating point number)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Peak Data Rate [p] (32-bit IEEE floating point number)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Minimum Policed Unit [m] (32-bit integer)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Maximum Packet Size [M] (32-bit integer)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
RSVP-TE (RFC 3209)

- RSVP extensions to support MPLS
- What is new?
  - A new message, "Hello", for node failure detection
  - Change of the Path message
    - Updates of some objects
      - SESSION
      - SENDER_TSPEC
      - ...
    - A new mandatory object
      - LABEL_REQUEST
    - Two optional objects become mandatory
      - SENDER_TEMPLATE
      - SENDER_TSPEC
    - A new optional object
      - EXPLICIT_ROUTE object (ERO)

SESSION object

- Original SESSION object (RFC 2205)
  - IPv4 DestAddress (4 bytes)
  - Protocol Id | Flags | DestPort

- New SESSION object (RFC 3209)
  - IPv4 tunnel end point address: IP address of the egress node for the tunnel
  - Tunnel ID: An ID that remains constant over the life of the tunnel
  - Extended Tunnel ID: Can be set to the IP address of the ingress node to narrow the scope of the session to the ingress-egress pair
**SENDER_TEMPLATE object**

- Original format (RFC 2205)

```
+---------------------------------+-------------------+
| IPv4 SrcAddress (4 bytes)       | SrcPort           |
+---------------------------------+-------------------+
```

- New format (RFC 3209)

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----------------------------------------------+
| IPv4 tunnel sender address                    |
+-----------------------------------------------+
| MUST be zero                                  |
| LSP ID                                        |
+-----------------------------------------------+
```

**LABEL_REQUEST object**

- Three types
  - Without label range

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----------------------------------------------+
| Reserved                                      |
| LSPID                                         |
```

  - With an ATM label range

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----------------------------------------------+
| Reserved                                      |
| LSPID                                         |
| M | Res | Minimum VPI | Minimum VCI |
|   | Res | Maximum VPI | Maximum VCI |
```

  - With a Frame Relay label range

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----------------------------------------------+
| Reserved                                      |
| LSPID                                         |
| Reserved                                      |
| [GL] | Minimum DLCI |
| Reserved                                      |
| Maximum DLCI                                 |
```

85

86
Explicit Route Object (ERO)

- A list of groups of nodes along the explicit route (generically called "source route")
- Thinking: source routing is better for calls than hop-by-hop routing (which is used for packet forwarding) as it can take into account loading conditions
- Constrained shortest path first (CSPF) algorithm executed at the first node to compute end-to-end route, which is included in the ERO

Source routing

- Many papers describe the call setup procedure as ingress node performing CSPF or Routing and Timeslot Allocation (RTA), and then sending an RSVP message with an ERO
  - contrast with the hop-by-hop approach described in the Principles section
Source routing

Routing updates from Chicago to NYC about its link to SF

- NYC trusts the "OC3 left" message from Chicago and routes the call to Chicago to reach SF
- This works if call arrival rate is low; if this rate is high and Chicago to SF calls could have used up the OC3 before the next update and the signaling request arrives in between, this will not work.

RSVP-TE extension for GMPLS (RFC 3471, 3473)

- RSVP-TE extension for Generalized MPLS
- What is new?
  - A new message, "Notify", for supporting fast failure notification
  - Update of objects
    - Generalized LABEL_REQUEST
    - Generalized LABEL
      - Support labels to identify timeslots, wavelengths, etc
      - The label "class" is implicit in the multiplexing capability of the link
  - Interface ID field added to RSVP-HOP object, ERO
  - A new object
    - UPSTREAM_LABEL - support bidirectional setup
**Generalized LABEL_REQUEST object**

<table>
<thead>
<tr>
<th>LSP Enc. Type</th>
<th>Switching Type</th>
<th>G-PID</th>
</tr>
</thead>
</table>
| LSP Encoding Type: encoding of the LSP being requested  
  • e.g.: Ethernet, SONET, Digital Wrapper… - lowest layer |
| Switching Type: type of multiplexing  
  • e.g.: TDM, LSC, FSC… |
| Generalized PID: identify the payload of the LSP - what is carried on the LSP  
  • e.g.: SONET/SDH for Lambda encoding  
  DS1/DS3 for SONET encoding |

**Need for Interface ID**

- Separation of control plane from data plane in GMPLS networks - out-of-band
Need for Interface ID

- Control plane separation:
  - Requires upstream switch to identify on which data-plane interface the virtual circuit should be routed
  - Interface ID field defined in the tag-length-value format
    - Identifier types:
      - IPv4 or IPv6 address ("numbered" link)
      - Interface index ("unnumbered" link)
        - Saves on IP addresses
        - Little need to allocate a separate address to each interface of a SONET switch
    - Embedded within the RSVP-HOP object

Unnumbered Links (RFC 3477)

- Unnumbered links: links that are not assigned IP addresses
- Two issues:
  - How to carry TE information about unnumbered links in IGP TE extensions (covered by GMPLS-ISIS and GMPLS OSPF)?
  - How to specify unnumbered links in GMPLS signaling?
    - An unnumbered link has to be point-to-point
    - The switch at each end assigns a 32-bit ID to the link
    - Unnumbered interface IDs (IF_IDS) are supported in RSVP_HOP object and ERO, etc.
Unidirectional vs. Bidirectional

- In RFC 3209, to set up a bidirectional LSP, two unidirectional paths must be established independently.
- UPSTREAM_LABEL object
  - Indicates the request of a bidirectional circuit
  - Same format as the LABEL object
- Why do we need this?
  - Reduce setup delay and control overhead
  - Avoid race conditions in resource assignment
  - Bidirectional optical LSPs are often required in optical networking services (many vendors only support bidirectional setup)

RSVP-TE GMPLS extension for SONET/SDH (RFC 4606)

- Label and bandwidth parameters changed
  - A new LABEL format for SONET/SDH - SUKLM
  - A new Tspec format for SONET/SDH Traffic
### SONET_/SDH_Tspec

<table>
<thead>
<tr>
<th>Signal Type</th>
<th>RCC</th>
<th>NCC</th>
<th>NVC</th>
<th>Multiplier (MT)</th>
<th>Transparency (T)</th>
<th>Profile (P)</th>
</tr>
</thead>
</table>

- **Signaling type:** the type of elementary signal
  - Eg: VT1.5, STS-1, STS-12...
- **RCC:** requested contiguous concatenation
- **NCC:** number of contiguous components
- **NVC:** number of virtual components
- **MT:** number of identical signals requested

### SONET/SDH LABEL - SUKLM

- **Five parameters but only some of them are significant for different multiplexing schemes**

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **S=1->N:** the index of a particular STS-3 inside an STS-N multiplexed signal
- **U=1→3:** the index of a particular STS-1_SPE within an STS-3
- **K=1→3:** for SDH only
- **L=1→7:** the index of a particular VT_Group within an STS-1_SPE
- **M:** the index of a particular VT1.5/2/3_SPE
RSVP-TE signaling procedures

- Distribute bandwidth management functionality to each switch for its own interfaces
- 5 steps of circuit setup processing at each switch:
  - Message parsing
  - Route determination
  - Connection admission control
  - Date-plane configuration
  - Message construction

RSVP-TE signaling procedures

- Data tables maintained at each switch:
  - Routing table
    - Simplest: next-hop node to reach destination
    - Precomputed after routing information is collected by OSPF-TE
  - Connectivity table
    - Data-plane interfaces and interface IDs
    - Control-plane address correlation
  - CAC table
    - Available bandwidth for each data-plane interface
  - State table
    - Information about each live circuit or VC
Path message processing:
main step

- **SESSION**
  - Search State table to check if session exist
    - Yes (Refresh)
    - No
  - Search Routing table for next hop
  - Route found
    - Yes
    - No **PathErr**
  - From **SENDER_TSPEC**
    - Allocate bandwidth on data-plane interface outgoing to the next hop (CAC)
    - Yes
    - No **PathErr**
    - Allocation successful
    - Yes
    - No **PathErr**
    - Update CAC table
    - DONE

Processing of Resv message:
main step

- **LABEL**
  - Outgoing_label(s) <- Label(s)
  - From **SESSION & FILTER_SPEC**
  - Outgoing_Label(s) in accordance with Outgoing_assigned_Timeslots
    - No **ResvErr**
    - Yes
  - Update Outgoing CAC table if necessary
  - From **SESSION & FILTER_SPEC**
  - Program switch fabric with Incoming/outgoing physical interface ID and Incoming/outgoing labels
  - DONE
Technologies

- **Connection-oriented (CO) networks**
  - Data-(user-) plane protocols
    - packet-switched: MPLS, VLAN Ethernet, Intserv IP
    - circuit-switched: SONET/SDH, WDM, SDM
  - Control-plane protocols:
    - RSVP-TE
    - OSPF-TE
    - LMP

- **Internetworking**
  - GFP, VCAT, LCAS for SONET/SDH
  - PWE3 for MPLS networks
  - Digital wrapper for OTN

---

OSPF-TE

- OSPF-TE adds more attributes to links in OSPF link state advertisements (LSA).
- These LSAs are distributed in a given OSPF area.
- Routers build an extended link database based on these LSAs that can be used to
  - Monitor the link attributes.
  - Perform local constraint-based source routing.
  - Global traffic engineering.
Purpose of OSPF-TE

- To advertise loading conditions
- RFC 3630 - for MPLS networks

OSPF-TE LSA

<table>
<thead>
<tr>
<th>Link-state age</th>
<th>Options</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link-state ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advertising Router</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link-state sequence number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link-state checksum</td>
<td>Length</td>
<td></td>
</tr>
<tr>
<td>LSA Payload (variable)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Common LSA header 0-31
**TE-LSA Header**

- **Link-state age**: time since this LSA generation.
- **Options**: optional functionality supported by the router.
- **Type**: OSPF opaque LSA (=10) with area flooding scope.
- **Link-state ID**: 1 in the first octet followed by an Instance field in the remaining 3 octets.
- **Advertising router**: router ID of the router generating this LSA.
- **Link-state sequence number**: Identifies a unique LSA to detect losses and duplicates.
- **Checksum**: Covering all except the age field.
- **Length**: in bytes of the LSA including the LSA header.

---

**TLV**

- The TE-LSA payload carries one or more nested Type/Length/Value (TLV) triplets

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Value</td>
</tr>
<tr>
<td></td>
<td>(variable)</td>
</tr>
</tbody>
</table>

+-------------------+------------+------------------+
| TLV format        | 0          | 31               |
+-------------------+------------+------------------+

- **Type**: either Router Address TLV (=1) or Link TLV (=2)
- **Length**: length of the value field in octets
TLV

- **Router Address TLV**
  - Router ID of the advertising router
  - Router ID is a loopback address that can be reached via any interface (typically used in routing protocols instead of a specific interface IP address to avoid loss of reachability to the router if the interface fails)
  - The value field contains this IP address.
  - It must appear in exactly one TE-LSA from a router
  - Purpose: assume it is to identify that the router is a TE-capable router

TLV

- **Link TLV**
  - It describes attributes of a single link.
  - It is composed of a set of sub-TLVs.
  - Each TE-LSA carries only one link TLV.
Sub-TLVs

- Contained in the value field of a Link TLV
- Multiple types of sub-TLVs are defined. Some of them are
  - Link type: Point-to-point or Multi-access.
  - Link id: identifies the other end of the link.
  - Local interface IP address
  - Remote interface IP address
  - Traffic engineering metric: typically assigned by the administrator and could be different from the OSPF link metric
  - Maximum bandwidth: maximum bandwidth that can be used.
  - Maximum reservable bandwidth: can be greater than the maximum bandwidth to support oversubscription
  - Unreserved bandwidth
  - Administrative group (4-byte mask: 1 bit per admin group for link)
- Bandwidth fields (in bytes) are expressed in IEEE floating point format

Link type

- Link type: point-to-point or multi-access
- Link ID: identifies the other end of the link as in a Router LSA
  - point-to-point links: Router ID of the neighbor
  - multi-access links: interface address of the designated router
OSPF-TE extensions for GMPLS (RFC 4202 and 4203)

- New sub-TLVs for the Link TLV
  - Link Local/Remote Identifiers
  - Link Protection Type
  - Shared Risk Link Group
  - Interface Switching Capability Descriptor (ISCD)
    - main extension since GMPLS allows multiple types of switching techniques

New sub-TLVs

- Link Local/Remote Identifiers
  - Since GMPLS added interface IDs for unnumbered links (i.e., links that are not assigned IP addresses), this sub-TLV carries those identifiers
- Link protection type: Extra, Shared, dedicated 1:1, dedicated 1+1, unprotected, enhanced
Shared risk link group (SRLG)

- SRLG: set of links that share a resource whose failure may affect all links in the set.
  - Example, two fibers in the same conduit would be in the same SRLG.
- SRLG sub-TLV for a link is an unordered list of SRLGs that the link belongs to. This could be more than 1.
- SRLG is identified by a 32 bit number that is unique within an IGP domain.

Interface Switching Capability Descriptor (ISCD)

<table>
<thead>
<tr>
<th>Switching cap</th>
<th>Encoding</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max LSP Bandwidth at priority 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max LSP Bandwidth at priority 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max LSP Bandwidth at priority 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switching Capability specific information (variable)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ISCD format 0-31
Interface Switching Capability Descriptor (ISCD)

• It describes the switching capability of the link
• Switching capability can be
  - Packet switch capable (PSC)
  - Layer-2 switch capable (L2SC)
  - Time-division-multiplex switch capable (TDM)
  - Lambda-switch capable (LSC)
  - Fiber-switch capable (FSC)
• Encoding: Same as LSP encoding in Generalized label request object of RSVP-TE - see RFC 3471

Interface Switching Capability Descriptor (ISCD)

• The maximum LSP bandwidth at priority $p$: the smaller of the unreserved bandwidth at priority $p$ and a "Maximum LSP Size" parameter which is locally configured on the link, and whose default value is equal to the max link bandwidth.
### ISCD Specific Information

- No ISCD specific information for L2SC, and LSC.
- When the switching capability is PSC, the following fields are generated:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum LSP bandwidth</td>
<td></td>
</tr>
<tr>
<td>Interface MTU</td>
<td></td>
</tr>
<tr>
<td>Padding</td>
<td></td>
</tr>
</tbody>
</table>

- Padding is used to make the ISCD 32-bits aligned.

### ISCD Specific Information

- For TDM switching capability, the following fields are generated:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum LSP bandwidth</td>
<td></td>
</tr>
<tr>
<td>Indication</td>
<td></td>
</tr>
<tr>
<td>Padding</td>
<td></td>
</tr>
</tbody>
</table>

- Minimum LSP Bandwidth example: OC1 on a SONET interface if the switch demultiplexes down to OC1 level
- The indication field takes a binary value stating whether the interface supports standard or arbitrary SONET/SDH
- Optionally, how many time-slots are free on a TDM link can be incorporated in the ISCD specific information field
  - 32 bit tuple: `<signal_type(8 bits), number of unallocated timeslots(24 bits)>`
References for OSPF-TE


Difference between labels in MPLS and circuit-switched GMPLS

- In circuit-switched GMPLS networks, labels are not carried in the data plane
  - Labels in circuit-switched networks identify "position" of data for the circuit - time or wavelength
- In circuit-switched GMPLS networks, cannot assign labels without associated bandwidth reservation
  - In usage section, we will see the value of this feature in MPLS networks
  - See two applications: traffic engineering, VPLS (addressing benefits)
Technologies

- **Connection-oriented (CO) networks**
  - Data-(user-) plane protocols
    - packet-switched: MPLS, VLAN Ethernet, Intserv IP
    - circuit-switched: SONET/SDH, WDM, SDM
  - Control-plane protocols:
    - RSVP-TE
    - OSPF-TE
    - LMP

- **Internetworking**
  - PWE3 for MPLS networks
  - GFP, VCAT, LCAS for SONET/SDH
  - Digital wrapper for OTN

LMP procedures

- **Control channel management**
  - Set up and maintain control channels between adjacent nodes

- **Link property correlation**
  - Aggregate multiple data links into a TE link
  - Synchronize TE link properties at both ends

- **Link connectivity verification (optional)**
  - Data plane discovery; If_Id exchange; physical connectivity verification

- **Fault management (optional)**
  - Fault notification and localization

Reference: IETF RFC 4204
What is a control channel?

• A control channel is a pair of mutually reachable interfaces that are used to enable communication between nodes for routing, signaling, and link management.

• Obvious question: bootstrap issue
  - how do you exchange messages on a control channel to create a control channel?

Types of control channels

• LMP does not specify the exact implementation of the control channel

• Examples of control channels:
  - a separate wavelength or fiber
  - an Ethernet link
  - an IP tunnel through a separate management network (e.g., Internet)
  - the overhead bytes of a data link (e.g., DCC)
Control channel identifier

(\textit{CC\_Id})

- A number from the space in which unnumbered interface IDs are assigned by a node
  - a 32-bit integer unique to the node
- Assign IP addresses to control channel ends
  - Because LMP runs over UDP/IP (UDP port number: 701)
  - remote end IP address: manually configured or automatically discovered

Automatic discovery

- How does a node automatically discover the IP address assigned to remote end of one of its control channels:
  - \textit{Config} message sent:
    - source IP address: unicast address
    - destination IP address: multicast 224.0.0.1
    - \textit{Config\_ACK} message returned with destination IP address
  - Used when control channel is a DCC channel within a data link
Control channel management

- **Config, ConfigAck, ConfigNack messages**
  - Specify
    - Control_Channel_ID
    - Node_ID (Router ID used in routing protocols)
    - Hello protocol parameters (hello interval and dead interval)
    - Message_ID - just for ARQ support for these LMP message exchanges
      - process used in RSVP too because RSVP runs on IP
- **Hello messages - a lightweight keep-alive mechanism**
  - Used to maintain control channel connectivity and detect control channel failure
- **Multiple control channels allowed**
  - Useful in case of control channel failure

Link property correlation

- **Message LinkSummary**
  - Summarizes TE link information (data-plane interfaces);
    Indicates support for fault management and link verification procedures
- **Message LinkSummaryAck**
  - Signals agreement on message LinkSummary
- **Message LinkSummaryNack**
  - Indicates disagreement; may suggest alternative values for negotiable parameters
  - Example: if one end of a TE-link is assigned an IPv4 address and the other end is assigned an IPv6 or unnumbered interface ID, there is a problem
Link connectivity verification (optional)

- **Obj**: Verify physical connectivity of data links and dynamically learn the TE link and interface ID associations.
  - A node must be able to send message over any data link
- **Procedure**
  - Exchange of a pair of `BeginVerify` and `BeginVerifyACK` message over a control channel
  - Upstream node sends `Test` messages with local If_Id on a data link
  - Downstream node replies with `TestStatusSuccess` or `TestStatusFailure` accordingly over the control channel
  - If `TestStatusSuccess`, upstream node records the mapping of local If_Id and remote If_Id, marks the link as "Up", and then follows up with a `TestStatusAck` message for acknowledgement. If `TestStatusFailure`, marks the link as "Failed".
  - Use `EndVerify` message to complete the procedure when all data links are tested.

Fault management (optional)

- For failure notification and localization only
  - Assume fault detection done at lower layer, e.g., loss of light observed at physical layer
- **Purpose of procedure**:
  - "To avoid multiple alarms stemming from the same failure, LMP provides failure notification through the ChannelStatus message"

Reference: IETF RFC 4204
Fault management procedure

A failure occurs between Nodes 2 and 3:

a. Node 3 (downstream node) will detect the failure and send a ChannelStatus message to node 2 indicating the failure.

b. Node 2 will immediately acknowledge this message by returning a ChannelStatusAck message.

c. Node 2 will then correlate the message to see if the failure is also detected locally

d. If there is no problem on the input side to Node 2 and within Node 2, it means the failure is localized

e. Node 2 then sends a ChannelStatus message to node 3 indicating that the failure has been localized and that the link is either failed or OK
   • Presumably, if there was a protection path, Node 2 could quickly restore the channel and send an OK status.

Control-plane security

• Need authentication and integrity for all control-plane exchanges

• Since RSVP, OSPF, LMP run over IP, IPsec is a possible solution
Technologies

• Connection-oriented (CO) networks
  - Data-(user-) plane protocols
    • packet-switched: MPLS, VLAN Ethernet, Intserv IP
    • circuit-switched: SONET/SDH, WDM, SDM
  - Control-plane protocols:
    • RSVP-TE
    • OSPF-TE
    • LMP

[Internetworking]
  - GFP, VCAT, LCAS for SONET/SDH
  - PWE3 for MPLS networks
  - Digital wrapper for OTN

Why internetworking?

• GMPLS networks do not exist as standalone entities
• Instead they are part of the Internet:
  - Obvious usage: to interconnect IP routers
  - Newer uses:
    • Commercial: interconnect Ethernet switches in geographically distributed LANs via point-to-point links or VPNs
    • Research & Education networks: connect GbE and 10GbE cards on cluster computers and storage devices to GMPLS networks
Obvious usage

- Router-to-router circuits and virtual circuits

Router-to-router usage

- OSPF-enabled usage
  - simply treat MPLS virtual circuit or GMPLS circuit as a link between routers
  - allow routing protocol to include these in routing table computations

- Data-plane
  - IP over MPLS
  - IP over PPP over SONET
    - Packet-over-SONET (PoS)
IP over MPLS

Label Switched Path (LSP) from DC to Sunnyvale, CA

Newer uses

- Ethernet over MPLS/GMPLS
  VC/circuits:
  - port mapped
  - VLAN mapped
**Ethernet port mapped over MPLS**

- Send all Ethernet frames received on ports I and II on to the MPLS virtual circuit
- MPLS virtual circuit: Pseudo-wire
- Enterprise can allocate IP addresses from one subnet: Virtual private LAN
- Explains one use for MPLS virtual circuits with no bandwidth allocation

**Gateway**: interfaces have different MUX schemes unlike switch ("my definition")

**SDM**:
- Space Division Multiplexing

**Ethernet VLAN mapped over MPLS**

- Extract frames carrying a specific VLAN ID tag on Ethernet ports I and II and map only these frames on to the MPLS virtual circuit
Ethernet port or VLAN mapped over GMPLS circuits

- Send all frames or frames matching a given VLAN ID tag from Ethernet ports I and II on to the SONET/SDH/WDM circuit
- SONET/SDH/WDM switches now have Fast Ethernet/GbE/10GbE interfaces in addition to SONET/SDM or WDM interfaces

Commercial services

- EPL: Ethernet private line: map an Ethernet port to a SONET/SDH circuit
- Fractional-EPL: Map a GbE port to a lower-rate SONET circuit
  - Pause frames received from switch to client node on the other side of the GbE
- V-EPL: Lower-rate VLAN mapped to an equivalent rate SONET circuit
**RENE application**

- Cluster computers, disk arrays, visualization clusters have GbE/10GbE interfaces
- Network: SONET/SDH/WDM or MPLS, for rate-guaranteed service

**Technology**

- So what technologies are required for this type of internetworking:
  - mapping Ethernet frames on to MPLS/GMPLS virtual circuit/circuit mapping?
Technologies

• Connection-oriented (CO) networks
  - Data-(user-) plane protocols
    • packet-switched: MPLS, VLAN Ethernet, Intserv IP
    • circuit-switched: SONET/SDH, WDM, SDM
  - Control-plane protocols:
    • RSVP-TE
    • OSPF-TE
    • LMP

• Internetworking
  • GFP, VCAT, LCAS for SONET/SDH
  • PWE for MPLS networks
  • Digital wrapper for OTN

Reference

• IEEE Communications Magazine, May 2002, Special issue on "Generic Framing Procedure (GFP) and Data over SONET/SDH and OTN," Guest Editors, Tim Armstrong and Steven S. Gorshe

• 6 excellent papers
What is GFP?

• Generic Framing Procedure (GFP) is a mechanism to transport packet-based data streams or block-oriented data streams over a synchronous communications channel, such as SONET/SDH
• My classification: It is a data-link layer protocol

Protocol stacks for various data transport applications
Why do we need GFP?

- Why do we need yet another data-link layer protocol?
  - More specifically, to transport data packets over synchronous links?

Main reason

- The framing techniques used in other data-link layer protocols have problems
- For example, IP packets are carried over SONET using PPP/HDLC frames (called PoS)
  - HDLC inserts idle frames because SONET is synchronous it needs a constant flow of frames to avoid losing synchronization
- But, there is a problem:
  - HDLC uses flags for frame delineation. The issue with this framing technique is that if the flag pattern occurs in the payload, an escape byte has to be inserted
  - This causes an increase in the required bandwidth
  - The amount of increase is payload-dependent
Other framing techniques

- HEC - Header Error Control
  - this is the CRC framing technique used in ATM
  - "A header CRC hunting mechanism is employed by the receiver to extract the ATM cells from the bit/byte synchronous stream. The HEC location is fixed and ATM cell length is fixed. Starting from the assumed cell boundary, the ATM receiver compares its computed HEC value for the assumed ATM cell header against the HEC value indicated by the assumed HEC field. Cell stream delineation is declared after positive validations of the incoming HEC fields of a few consecutive ATM cells."
- ATM cells are fixed in length, but Ethernet frames are variable-length
- Therefore, we need a length field in order to implement this HEC-based frame delineation mechanism

Main features of the GFP protocol

- Common aspects:
  - HEC + Length based delineation
    - Core header has payload length and HEC
  - Error control: error detection
    - Payload type HEC, payload Frame Check Sequence (CRC-32)
  - Multiplexing: linear and ring extension headers
  - Idle frames are sent to maintain synchronization as in HDLC
  - Scrambling as in ATM:
    - core header + payload scrambling
  - Client management - client fail signal
- Client-dependent aspects:
  - Client-specific encapsulation techniques
GFP frame types

- GFP frames
  - Client frames
    - Client data frames (CDFs)
    - Client management frames (CMFs)
  - Control frames
    - Idle frames
    - OA&M frames

  - CDFs: client data.
  - CMFs: information associated with the management of the client signal or GFP connection.
  - Idle frames: 4-byte GFP control frames.
  - OA&M frames: operations, administration, and maintenance.

GFP frame structure

- Client data frames
  - Core header
    - Client data frames
    - Payload Area
      - Payload length MSB
      - Payload length LSB
      - Core HEC MSB
      - Core HEC LSB
  - Payload header
    - Payload information N [536,550] or variable length packets
    - Payload FCS
  - Payload type MSB
  - Payload type LSB
  - Type HEC MSB
  - Type HEC LSB
  - 0-60 bytes of extension headers (optional)
  - Payload FCS MSB
  - Payload FCS LSB
  - PTI| PFI | EXI | UPI
  - CID
  - Spare
  - Extension HEC MSB
  - Extension HEC LSB
  - Linear extension header shown (others may apply)
  - Client control frames
    - Bit transmission order
    - Byte or variable length packets
    - Payload FCS
  - Idle frame (scrambled)
**GFP core header**

- **Payload length indicator (PLI): 2 bytes**
  - the size of the payload area in bytes
  - allows GFP frame delineation independent of the content of higher-layer PDUs
- **Core HEC (cHEC): 2 bytes**
  - CRC16 to enable delineation

![Diagram showing Hunt, Sync, Presync, and their relationships with Correct/Incorrect cHEC for N and M frames.]

**GFP payload area**

- **Payload header: 4-64 bytes**
  - Payload type: mandatory field; 2 bytes; the content and format of the payload
    - Payload type identifier (PTI): 3 bits; the type of GFP client frames (CDF or CMF)
    - Payload FCS Indicator (PFI): 1 bit; the presence of the payload FCS field
    - Extension Header Identifier (EXI): 4 bits; the type of extension header GFP (e.g., linear extension header)
    - User Payload Identifier (UPI): 8 bits; the type of payload
  - Type Hec (tHEC): 2 bytes; CRC-16 to protect the payload type field
GFP payload area

- Payload header:
  - Extension headers: 0-60 bytes (optional)
    - Null Extension Header: 0 bytes; by default
    - Linear Extension Header: 2 bytes; multi-access link
      - Channel ID (CID): 1 byte; like MPLS label or VLAN ID for multiplexing
      - Spare field: 1 byte
    - Ring Extension Header: sharing of the GFP payload across multiple clients in a ring configuration
  - Extension HEC (eHEC): mandatory; 2 byte; CRC-16 to protect the extension header

GFP payload area

- Payload information field: 0 to (65535 - X) bytes, where X is the length of payload header and payload FCS;
- Payload Frame Check Sequence (FCS): optional (indicated by PFI); 4 bytes; CRC-32 to protect payload information field
**GFP's location in protocol stack**

<table>
<thead>
<tr>
<th>Ethernet</th>
<th>IP/PPP</th>
<th>Other client signals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>GFP-Client specific aspects</strong> (payload-dependent)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>GFP-Common aspects</strong> (payload-independent)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SONET/SDH path</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>OTN OCh path</strong></td>
</tr>
</tbody>
</table>

**Main features of the GFP protocol revisited**

- **Common aspects:**
  - HEC + Length based delineation
    - Core header has payload length and HEC
  - Error control: error detection
    - Payload type HEC, payload Frame Check Sequence (CRC-32)
  - Multiplexing: linear and ring extension headers
  - Idle frames are sent to maintain synchronization as in HDLC
    - Scrambling as in ATM:
      - core header + payload scrambling
  - Client management - client fail signal
- **Client-dependent aspects:**
  - Client-specific encapsulation techniques

Page 68 of reference
Need for scrambling

- Line coding used in SONET/SDH and OTN optical communication links is NRZ (Non-Return to Zero)
  - Laser is turned ON if bit is 1, and OFF if bit is 0
- Advantages of NRZ: simplicity and bandwidth-efficiency
- Disadvantage: loss of synchronization possible at the receiver by the clock and data recovery circuits if there are many consecutive 0 bits in the data stream
  - could be caused by a malicious user sending such a payload

Scrambling solution

- Self-synchronous payload scrambler
  - Use a polynomial of $x^{43}+1$: XOR bit with scrambler output bit that preceeded it by 43 bits
  - Drawback: error multiplication

- Solution to error multiplication
  - Select a CRC generator polynomial with triple error detection capability and have no common factor with scrambler

$$x^{16} + x^{15} + x^{12} + x^{10} + x^4 + x^3 + x^2 + x + 1$$
GFP client-based aspects

- Frame-mapped GFP (GFP-F)
  - 1-to-1 mapping: one client frame is mapped into one GFP frame
  - Applicable to most packet data types, e.g., Ethernet MAC frames, IP packets

- Transparent-mapped GFP (GFP-T)
  - Many-to-1 mapping: a fixed # of client characters are mapped into a GFP frame of predetermined length
  - Applicable to 8B/10B block-coded client signals such as fiber channel, GbE (1Gb/s Ethernet)

GFP-F frame

<table>
<thead>
<tr>
<th>PLI</th>
<th>cHEC</th>
<th>Payload header</th>
<th>Client PDU (PPP, IP, Ethernet, RPR, etc)</th>
<th>FCS (optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 bytes</td>
<td>2 bytes</td>
<td>4 bytes</td>
<td>0-65,531 bytes</td>
<td>4 bytes</td>
</tr>
</tbody>
</table>

- GFP header
- GFP payload
- GFP FCS
### GFP-T Frame

<table>
<thead>
<tr>
<th>PLI</th>
<th>eHEC</th>
<th>Payload header</th>
<th>#1</th>
<th>8x64B/65B + 16 superblock bits</th>
<th>#N</th>
<th>FCS (optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 bytes</td>
<td>2 bytes</td>
<td>4 bytes</td>
<td></td>
<td></td>
<td></td>
<td>4 bytes</td>
</tr>
</tbody>
</table>

- **GFP header**: GFP payload
- **PLI**: 2 bytes
- **eHEC**: 2 bytes
- **Payload header**: 4 bytes
- **#1**: 8x64B/65B + 16 superblock bits
- **#N**: (optional)
- **FCS**: 4 bytes

#### GFP-T Encoding Steps

1. **Decode 8B/10B code words into original 8-bit values**
2. **Map eight decoded characters into a 64B/65B block code and set a flag bit to indicate if the block contains only data characters (DCI)**
3. **Create a superblock**
   1. Group 8 64B/65B blocks
   2. Rearrange leading bits at end
   3. Generate and append CRC-16 check bits to form a superblock
4. **Repeat creating at least N such superblocks**
   - **N**: minimum # of superblocks per GFP frame (e.g., 95 for GbE)
5. **Prepend with GFP core and payload headers**
6. **Scramble payload header and payload with \(x^{43}+1\)**

---

LCC: last control character  
CCI: control code indicator  
CCL: control code locator  
DCI: data character identifier
Comparing performance of GFP-F & GFP-T

• GFP-F
  - Efficient bandwidth utilization:
    • only delivers client data frames (idle frames are removed)
    • if client signal is lightly loaded, GFP-F can map this signal to a lower-rate circuit or GFP multiplex with other signals
  - Higher latency: associated with buffering an entire client data frame at the ingress to the GFP mapper

• GFP-T
  - Advantage: transparent transport of 8B/10B control characters as well as data characters
    • minimum protocol awareness
    • a single hardware implementation can handle many types of client signals (all that use 8B/10B coding)
  - Lower bandwidth utilization: if client signal contains idle frames, these are transported through transparently
  - Lower latency: only a few bytes of mapper/demapper latency
Virtual Concatenation (VCAT)

- Allows for SONET/SDH rates in-between the rigid rates of the original hierarchy
  - VT1.5-7v: means 7 virtually concatenated VT1.5 signals
- VCAT as an inverse multiplexing scheme
  - It allows for individual components of the virtually concatenated signal to be routed along different paths before recombining them into a contiguous-bandwidth signal at the far endpoint
  - Need to compensate for delays differences on the various paths used for the individual components
- Bandwidth partitioning
  - It allows for a SONET/SDH link to be partitioned into arbitrary units of bandwidth

VCAT increased bandwidth efficiency

<table>
<thead>
<tr>
<th>Data signal</th>
<th>SONET/SDH payload mapping and bandwidth efficiency</th>
<th>SONET/SDH with VCAT payload mapping and bandwidth efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet (10 Mb/s)</td>
<td>STS-1/VC-3 – 21%</td>
<td>VT1.5-7v/VC-11-7v – 89%</td>
</tr>
<tr>
<td>Fast Ethernet (100 Mb/s)</td>
<td>STS-3c/VC-4 – 67%</td>
<td>VT1.5-64v/VC-11-64v – 98%</td>
</tr>
<tr>
<td>Gigabit Ethernet (1000 Mb/s)</td>
<td>STS-48c/VC-4-16c – 42%</td>
<td>STS-3c-7v/VC-4-7v – 95% STS-1-21v/VC-3-21V – 98%</td>
</tr>
</tbody>
</table>
Inverse multiplexing in VCAT

Implementation of VCAT is only required at select nodes (i.e., the edge nodes); not all multiplexers need to support VCAT

Bandwidth partitioning with VCAT
Link Capacity Adjustment Scheme (LCAS)

- LCAS is a mechanism to allow for automatic bandwidth tuning of a virtually concatenated signal
  - The VCAT group of circuits should already be established using a
    - centralized NMS/EMS based procedure, or
    - by a distributed RSVP-TE based procedure
  - Note that bandwidth cannot be increased beyond the aggregate value of the VCAT signal without a GMPLS RSVP or NMS/EMS procedure of circuit setup

Interaction between GMPLS RSVP and LCAS

![Diagram](image-url)
Link Capacity Adjustment Scheme (LCAS)

• LCAS is basically a synchronization procedure between the two ends of a VCAT signal
  - Unlike GMPLS RSVP, it is NOT a bandwidth reservation and circuit setup or release procedure
• LCAS procedures (triggered by GMPLS or NMS/EMS):
  - add or remove a member of a VCAT group
  - renumber the members in a VCAT group
• Messages are exchanged between the originating and terminating SONET/SDH nodes to execute these LCAS procedures
  - Add member (ChID, GID)
  - Remove member (ChID, GID)
  - Member status
• Messages are sent in the H4 byte for high-order VCAT

Hitless change

• Hitless capacity adjustment
  - Without causing an errors during the process
  - "Two ends of the link must agree precisely when the VCAT group transitions to a new payload in which new members have been added or some previous members removed"
  - "Needs hardware-level synchronization as to when the SONET/SDH mappers should begin/stop inserting/extracting a payload from a VCAT group member"
• The link capacity adjustment does not impact user traffic flow (what if that is the bottleneck link for a TCP session?)
Applications of LCAS

• Adjusting bandwidth requirements on a time-of-day basis
  - A GbE signal may only require on average a 200-300Mbps SONET circuit
  - Establish an STS-1-7v (388.688Mbps) VCAT circuit
  - Then add/delete members as load increases or decreases
  - Need buffering and PAUSE signals to handle bursts
  - Can map two different GbE signals to one VCAT group with different sets of members?
• Rerouting of traffic after failures

Data over SONET/SDH (DoS)

• Using GFP, VCAT, & LCAS, DoS provides a set of mechanisms for efficient transport of data packets on SONET/SDH circuits
  - GFP: an efficient and standard data link layer protocol
  - VCAT: flexible bandwidth assignment scheme requiring no modification to intermediate nodes
  - LCAS: dynamic bandwidth adjustment of VCAT signal
Technologies

- **Connection-oriented (CO) networks**
  - Data-(user-) plane protocols
    - packet-switched: MPLS, VLAN Ethernet, Intserv IP
    - circuit-switched: SONET/SDH, WDM, SDM
  - Control-plane protocols:
    - RSVP-TE
    - OSPF-TE
    - LMP
- **Internetworking**
  - GFP, VCAT, LCAS for SONET/SDH
  - PWE3 for MPLS networks
  - Digital wrapper for OTN

Pseudo Wire Emulation

- **Pseudo Wire Emulation Edge-to-Edge (PWE3)** is a mechanism for emulating certain services across a packet-switched network:
  - Services: Frame-relay, ATM, Ethernet, TDM services, such as SONET/SDH
  - Packet-switched network:
    - IP
    - MPLS
Example of a PWE3 service: Ethernet over MPLS

- PW control word:
  - status
  - sequencing
  - timing - Real-time transport protocol (RTP)
- PW label and tunnel label:
  - MPLS label, L2TP session id, UDP port number


Example: NY to Chicago link is a point-to-point Ethernet link
- LSP encoding: Ethernet
- Switching type: PSC
- GPID: Ethernet
Digital wrapper

- ITU-T G. 709 provides a method to carry Ethernet frames, ATM cells, IP datagrams directly on a WDM lightpath

Outline

- Principles
  - Different types of connection-oriented networks
- Technologies
  - Single network
  - Internetworking
- Usage
  - Commercial networks
  - Research & Education Networks (REN)
Commercial uses

- Semi-permanent MPLS virtual circuits
  - Traffic engineering
  - Voice over IP
    - QoS concerns: telephony has a 150ms one-way delay requirement (with echo cancellers)
  - Business or service provider interconnect
    - interconnecting geographically distributed campuses of an enterprise
    - interconnecting wide-area routers of an ISP service provider

Traffic engineering (TE)

- Since BGP and OSPF routing protocols mainly spread reachability information, routing tables are such that some links become heavily congested while others are lightly loaded
- MPLS virtual circuits are used to alleviate this problem
  - e.g., NY to SF traffic could be directed to take an MPLS virtual circuit on a lightly loaded route avoiding all paths on which more local traffic may compete
- This is an application of MPLS VCs without bandwidth allocation
Goals of Traffic Engineering (TE)

- Monitor network resources and control traffic to maximize performance objectives
  - Goal of TE is to achieve efficient network operation with optimized resource utilization in an Autonomous System

- Goals of TE can be:
  - Traffic oriented
    - Enhance the QoS of traffic streams
    - Minimization of loss and delay
    - Maximization of throughput
  - Resource oriented
    - Load balancing
    - Minimize maximum congestion or minimize maximum resource utilization
    - Output - decreased packet loss and delay, increased throughput

Business or service provider interconnect

- Multiple options:
  - TDM circuits (traditional private line, T1, T3, OC3, OC12, etc.)
  - Ethernet private line
    - point-to-point (PWE3)
    - VPNs (called Virtual private LAN service)
  - MPLS VPNs
  - WDM lightpaths
  - Dark fiber
First option: buy OC192 between routers

Example: Internet2 purchased OC192s from Qwest

Second option: buy Ethernet point-to-point private lines

Example: NLR Framenet service; also Pacificwave
Third option: buy multipoint Ethernet VPN

VPLS: Virtual Private LAN service: an Ethernet private LAN created over a wide-area network

- SF PoP
- DC PoP
- Houston PoP

Multipoint Ethernet VLAN (VPN)

Can place all three ports in one VLAN

Dynamic circuits/VCs (GMPLS control-plane)

- Commercial:
  - fast restoration
    - circuit/VC setup delay significant
  - rapid provisioning
    - similar to scheduled (book-ahead reservations) of REN (research & education networks)
Industry usage of dynamic capability of GMPLS control-plane protocols

- Highly limited
- OIF interoperability testing focused on routers sending SONET setup messages to SONET switches
  - OIF UNI 1.0R2 and ENNI support only SONET circuits
- In 2005:
  - UNI 2.0 testing: to support GbE interfaces
  - But signaling/routing support for GbE-SONET-GbE circuits includes proprietary INNI solutions and no ENNI solution
  - GbE-SONET hybrid circuits important for REN applications

Compare "wire" services

- Disadvantages of Ethernet based solutions:
  - Spanning tree:
    - convergence slow
    - 7-hop limit
  - Flat addressing:
    - no summarization of MAC addresses
  - VLAN tag:
    - only 12 bits (only 4096 LANs)
    - No VLAN ID swapping (unlike MPLS labels)
      - contiguous requirement like lambdas in a WDM network
  - Few diagnostic tools to trace problems

Compare "wire" services

- WDM networks:
  - Low power consumption
- SONET/SDH networks:
  - Good error monitoring features
  - Higher-rate interfaces are cheaper than on IP routers

Research & Education

(G)MPLS networks

- NSF-funded CHEETAH
- NSF-funded DRAGON
- DOE's Ultra Science Network (USN)
- DOE's ESnet - Science Data Network
- Next-generation Internet2
- etc.
CHEETAH network - data plane links
GbEthernet and SONET

Sycamore SN16000
SONET switch with GbE/10GbE interfaces

CHEETAH network - control plane links
Design goal: scalable GMPLS network
Networking software

- Sycamore switch comes with built-in GMPLS control-plane protocols:
  - RSVP-TE and OSPF-TE
- We developed CHEETAH software for Linux end hosts:
  - circuit-requestor
    - allows users and applications to issue RSVP-TE call setup and release messages asking for dedicated circuits to remote end hosts
  - CircuitTCP (CTCP) code

Network service

- On-demand circuit-switched service for 1Gb/s dedicated host-to-host circuits
- Call setup delay: 1.5sec
  - Sycamore implemented a proprietary build for hybrid GbE-SONET-GbE circuits
  - No standard yet for such hybrid circuits
  - Sets up 7 STS-3c and VCATs them to carry a GbE signal
- In contrast, their GMPLS standards implementation for pure-SONET circuits incurs a call setup delay of 166ms (2-hop)
Applications

• eScience: Terascale Supernova Initiative
  - File transfers
  - Ensight remote visualization

• general-purpose:
  - file transfers between CDN servers, web mirrors
  - web caching
  - video applications

Interesting design considerations in the CHEETAH project

• Addressing: assignment of IP addresses to the end host and switches in the network
• Enabling OSPF-TE automatic neighbor discovery
• Security
Addressing

• Public vs. private? static vs. dynamic?
  - Shortage of IPv4 addresses
  - Enterprises often use private and/or dynamic IP addresses (NAT, DHCP, etc)
  - We assign **static public** IP addresses for both data-plane and control-plane IP addresses, why?
    • Data-plane
      - Static: an end hosts need to be "called" by other hosts
      - Public: the address need to be globally unique (Private IP addresses sufficient if goal for CHEETAH is to create a small eScience network)
    • Control-plane
      - Static: the control-plane IP addresses are configured in local Traffic-Engineering link configuration
      - Public: same global uniqueness reason for border switches

Address assignment example

TN SN16000  GA SN16000  NC SN16000

Data-plane links

Control-plane links

zelda1, zelda4, wukong: hosts
Impact of this addressing

- After dedicated circuit is setup:
  - far end NIC has an IP address from a different subnet
    - e.g.: zelda4 and wukong in the address assignment example
  - Default setting of IP routing table entries will indicate that such an address is only reachable through the default gateway
- Our solution:
  - Automatically update the routing table and ARP table when circuit is set up as part of signaling code
    - comparable to switch fabric programming in the switch
    - ARP table is also automatically updated to avoid extra round-trip propagation delay and potential broadcast storms caused by ARP
  - But how does the host find the remote MAC address?

Using DNS TXT resource record

- Add a TXT record for the DNS entry of each CHEETAH end host in the local DNS server
  - Indicate that the host is in the CHEETAH network
  - Record the MAC address of the host’s second NIC
- During circuit setup
  - The two CHEETAH hosts execute DNS lookup to retrieve the remote MAC address
  - At the end of a CHEETAH circuit setup, the two CHEETAH hosts
    - Add a host-specific entry for the far-end second NIC’s IP address into the IP routing table.
    - Add an entry into the ARP table to map the far-end second NIC’s IP address to its MAC address.
  - When the CHEETAH circuit is released, these entries are removed.
Enabling OSPF-TE automatic neighbor discovery

- Automatic neighbor discovery of OSPF-TE
  - Based on "Hello" messages
  - Hello messages will not be forwarded by IP routers
  - If two switches are data-plane neighbors, we need to ensure they are control-plane neighbors as well

Solution:
- IP-in-IP tunnels
  - Outer datagram header carries the Ethernet control port IP addresses
  - Inner datagram header carries the Router ID and the broadcast IP address as source and destination

Control-plane security

- Importance: a malicious user could tie up circuits
- Cannot use SSH, SSL, etc., because RSVP-TE and OSPF-TE use raw IP
- Our solution - IPsec tunnels
  - Use external security device (Juniper NS-5XT) for switches
  - Use open-source software (openswan) on Linux end hosts
  - Establish IPsec tunnels between adjacent switches and end hosts
- Firewalls
  - recall our static public IP address assignments
  - Use Juniper NS-5XT for switches and iptables for Linux hosts
- Limitation: host-based instead of user-based
  - Any user of the end host can request circuits after IPsec tunnel is established
  - Future plan: use the RSVP-TE INTEGRITY object
CHEETAH architecture

Applications
- DNS client
- RSVP-TE module
- TCP/IP
- C-TCP/IP

End Host
- NIC 1
- NIC 2

Internet
- SONET circuit-switched network
- Circuit Gateway

CHEETAH end-host software

- DNS lookup – to support our scalability goal
- Circuit-request setup
  - Message parsing
    - RSVP
  - CAC for UNI link
  - Date-plane configuration
    - Routing/ARP table update
  - Message construction
    - RSVP

Integrate CD API into web servers, FTP servers, etc., so that "elephant" flows are automatically handled via a dynamically created dedicated circuit/VC
End-to-end signaling delay measurements

• Signaling delays incurred in setting up a circuit between zelda1 (in Atlanta, GA) and wuneng (in Raleigh, NC) across the CHEETAH network.

<table>
<thead>
<tr>
<th>Circuit type</th>
<th>End-end circuit setup delay (s)</th>
<th>Processing delay for Path message at the NC SN16000 (s)</th>
<th>Processing delay for Resv message at the NC SN16000 (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC-1</td>
<td>0.166103</td>
<td>0.091119</td>
<td>0.008689</td>
</tr>
<tr>
<td>OC-3</td>
<td>0.165450</td>
<td>0.090852</td>
<td>0.008650</td>
</tr>
<tr>
<td>16Gb/s EoS</td>
<td>1.645673</td>
<td>1.566932</td>
<td>0.008697</td>
</tr>
</tbody>
</table>

Round-trip signaling message propagation plus emission delay between GA SN16000 and NC SN16000: 0.025s

• Observations:
  - Delays for setting up SONET circuits for rates in the original SONET hierarchy are very small (166ms)
  - Delays for other rates are much higher (1.6s) (vendor implementation)
  - Signaling message processing delay dominate the end-to-end circuit setup delay

Other R&E networks

• DRAGON:
  - GbE and WDM (Movaz)
  - VLSR code: external implementation of RSVP-TE and OSPF-TE: popular
  - per-domain route computation unit called NARB

• ESnet and Science data network
  - OSCARS: an advance-reservation system
  - MPLS network

• UltraScience Network
  - Research network for DoE labs
  - GbE and SONET (Ciena)
  - Centralized scheduler for advance-reservation calls
How advance-reservation systems work?

1. Maintains bandwidth availability over a time horizon for all links in the domain
2. A new protocol (BW requested + time)
3. When request for an advance reservation arrives, try different routes and find one with required bandwidth (centralized CAC)
4. Answer
5. Third-party Path message with ERO (just before scheduled time)
6. Program switch fabric
7. Path message

GMPLS RSVP-TE signaling used for “rapid provisioning”

Advantages of GMPLS control-plane sacrificed

- RSVP-TE engines at switch controllers are supposed to manage bandwidth for the interfaces of the switch
  - distributed bandwidth management
- Route computations are supposed to be distributed to each switch
  - distributed routing protocols
- Both these steps are centralized in a domain scheduler because RSVP-TE and OSPF-TE do not support parameters for advance-reservation calls
Wide-area REN

- **HOPI (Hybrid Optical Packet Infrastructure)**
  - Uses Ethernet switches to provide VLAN based virtual circuit service
  - Cheetah control-plane tested on HOPI

- **Next-generation Internet2**
  - Offers a dynamic circuit service (DCS)
  - Wide-scale deployment of Ciena CD-CIs

---

Internet2's new Dynamic Circuit Services (DCS) network

Yellow nodes: Ciena CD-CI SONET switches
Blue nodes: Juniper T640 IP routers

Courtesy: Rick Summerhill (2006)
References for REN projects

- IEEE Communication Magazine special issue, March 2006
  - DRAGON, USN, CHEETAH, several other projects
- CHEETAH web site:
  - http://www.ece.virginia.edu/cheetah/

Summary

- Principles
  - Different types of connection-oriented networks
- Technologies
  - Single network: MPLS, SONET, OTN
  - Internetworking: GFP, PWE3, G.709
- Usage
  - Commercial networks
  - Research & Education Networks (REN)