Layer 1 VPN Services in Next-Generation SONET/SDH Networks

Research Example

D. Benhaddou

University of Houston

TECH6100

Outline

- Research Cycle
- An example of research problem on Layer 1 VPN Overview
- Dynamic Shared L1 VPN Model
- Performance Evaluation
- Conclusions

Background and motivations

Importance of Research in Education

- Different learning style.
- Explore un-explored problems.
- Learn how to identify a problem.
- Pick up different skills that you would not learn in a regular classroom setting.
- Interact with faculty.
- Know how to define a problem and how to ask a research question.
- *But it is not easy*

Research phases

Define the problem

Solve the problem

Evaluate your solution

- **Analytical performance evaluation**
- **Numerical simulation**
- **Practical implementation**

Define research problem

The most important and tedious part.

It requires:

- Literature research
- Critical reading
- Discussion with advisor
- Re-reading
- Define the research topic

Solve the problem

A problem well define is a problem solved

Think about a solution

Use simulation tools

Use analytical tools

Implement the solution

Evaluate your solution

Develop the model and generate results.
Analyze the results
Modify the model for possible mistakes
Finalize the results

Research topic sample:
Layer 1 VPN Services in Next-Generation Networks

Motivations

Improve data services support

- Various improved mechanisms added:
  Efficient mappings, capacity assignment, control
- Re-use/maintain of legacy SONET:
  Ubiquity, framing/monitoring, grooming, resiliency

Key features and new developments

- *Generic framing procedure* (GFP, G.7041):
- *Virtual concatenation* (VCAT, G.707):
- *Link capacity adjustment scheme* (LCAS, G.7042):

Key Drivers & Motivations

"Infrastructure" virtualization

- Allows operators to better "amortize" infrastructures
- Allows clients to avoid costly *physical* build-outs

Virtualization is not a new concept

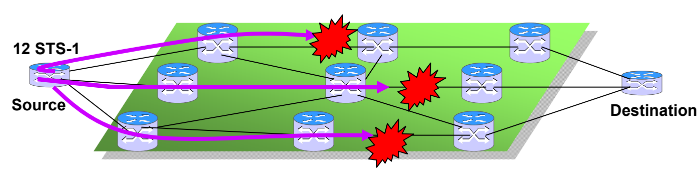
- Much work in Layer 2/3 VPN:
  Focus on resource management, security
- Primarily "packet-based" VPN solutions

"Layer 1 VPN" services

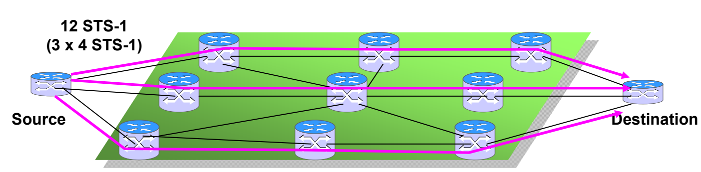
- Virtualization at the lowest (physical) layer
- Utilize some L2/L3 VPN designs concepts:
  Membership information, resource management.

Sample Illustration

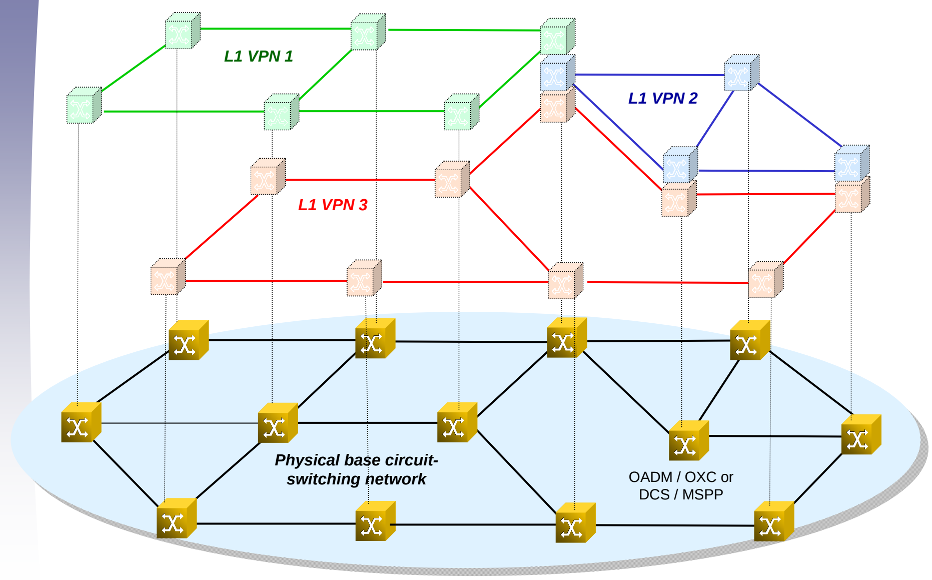
Regular Routing



Virtual Concatenation



Multiple Virtualized Infrastructures



Problem definition

Need of an algorithm that can enable dynamic provisioning of layer 1 VPN

Understand if L1-VPN is beneficial for transport networks

Dynamic L1 VPN Provisioning

Propose generic shared link model

- Much development in standards, few algorithmic studies
- Focus on *distributed* L1 VPN operation (dedicated, shared)
- Provide minimum guarantee, variable maximum

Key parameters

- Consider network graph $G(V,L)$ with N VPN's, $G_i(V_i,L_i)$
- Per-VPN i guarantee on link j is $Min\_VPN^i_j$
- Total sharable capacity on link j:

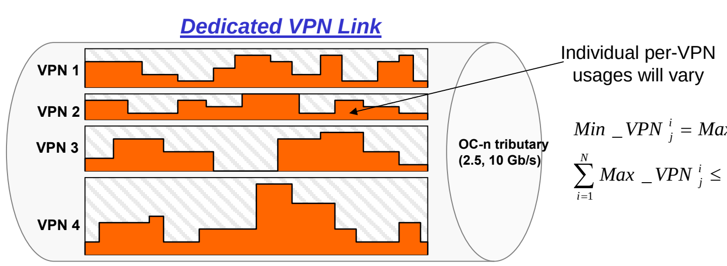
$$\Delta_j = C_j - \sum_{i=1}^{N} Min\_VPN^i_j$$

- Per-VPN maximum on link j:

$$Max\_VPN^i_j = Min\_VPN^i_j + \Delta_j$$

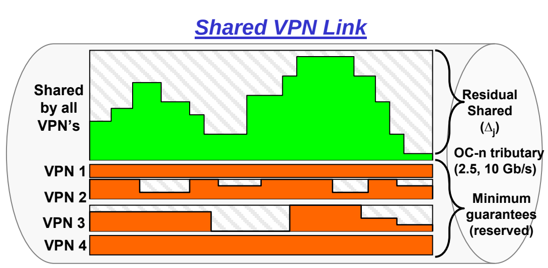
- Per-VPN usage is B^i_j, available is U^i_j

Generic Shared Link Model



$$Min\_VPN^i_j = Max\_VPN^i_j$$

$$\sum_{i=1}^{N} Max\_VPN^i_j \le C_j$$



$$Min\_VPN^i_j < Max\_VPN^i_j$$

$$\sum_{i=1}^{N} Min\_VPN^i_j$$

$$\Delta_j = C_j - \sum_{i=1}^{N} Min\_VPN^i_j$$

$$Max\_VPN^i_j = Min\_VPN^i_j + \Delta_j$$

Inter-VPN Resource Sharing

Need inter-VPN capacity management

- Notify/update available capacity values upon change
- Must be amenable to centralized and distributed operation

Capacity allocation

- Only update usage beyond minimum guarantee:

$$\varepsilon = x - \max((Min\_VPN_j^i - B_j^i),0)$$

- In distributed approach this requires LSA updates

Capacity de-allocation

- Only update reduction below minimum guarantee:

$$\varepsilon = \min((B_j^i - Min\_VPN_j^i), x)$$

- In distributed approach this requires LSA updates

Demand Resolution

"Split" a connection into "sub-connections"

- Several strategies possible, chose "even" approach
- Request for n STS-1 split into K sub-connections of size n_i

  where K is *inverse-multiplexing factor* and:

$$n_i = \begin{cases} d+1 & 1 \le i \le r \\ d & r < i \le K \end{cases}, \sum_i n_i = n$$

  and $d = \frac{n}{k}$ and $r = n - kd$

- Optimization of K is generally NP-complete, assume a-priori

Path computation

- Iterate over sub-connections, compute modified graph:
  - Prune links from $G_i(V_i,L_i)$ with free capacity < n_i
  - Remove any routed sub-connection capacity
- Run Dijskstra's shortest path on modified graph

Practical Considerations

Centralized approach

- OSS has centralized database, delayed updates
- Model using update delay parameter, Δ_C:
  Path computation on delayed graph, check on current

Distributed GMPLS-provisioning

- Per-VPN OPSF-TE and RSVP-TE instantiations (VPN id)
- Advertise available link capacity, PE's source route
- Use relative triggers, i.e., *significance change factors* (SCF)
- RSVP-TE signaling done in parallel for sub-connections:
  "Master" connection entities track setups

VPN Performance Evaluation

Overall approach

- Use *OPNET Modeler*^{TM} simulation tool
- Coded detailed OSPF-TE, RSVP-TE, CBR blocks

Key test parameters

- Use NSFNET base topology (16 nodes, 25 links)
- Exponential connection inter-arrival times
- Exponential connection holding time (600 sec mean)
- Model both legacy and emergent services

Assumptions

- No service protection (future study)

Tested Services

Legacy & Emergent Types

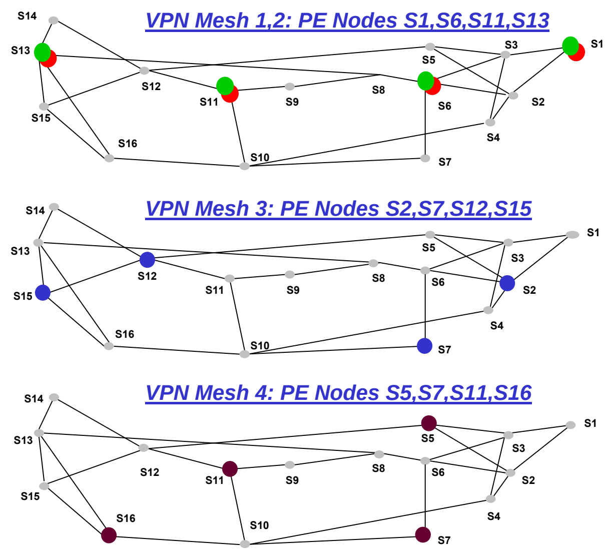
	Private Line	Bandwidth on Demand
Request Sizes	OC-3, OC-12, OC-48, GigE	Fractional GigE (1-21 STS-1 uniform)
Holding Times	Weeks-months	Hours-days
	➢All requests in STS-1 increments ➢Payloads mapped to TDM slots	

Simulation Parameters

Parameter	Value
Node message processing delay	0.01 ms
OSS controller processing delay	0.01 ms
Control channel rate	OC-12
Switching granularity	STS-1
Routing hold-off timer (OSPF)	5 sec (default)
Significance change factor (OSPF)	0.1 (default)
Mean holding time	600 sec (scaled)
Connection capacity range	OC-3/12/48, GigE

Note: All connections are bi-directional

Stacked Mesh VPN Topologies

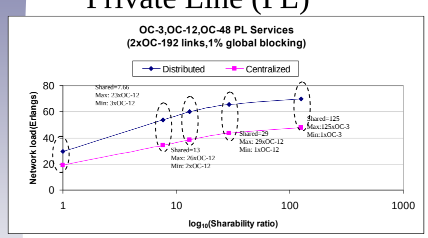


- Minimize effects of arbitrary topologies
- Maximize inter-VPN sharing/overlap

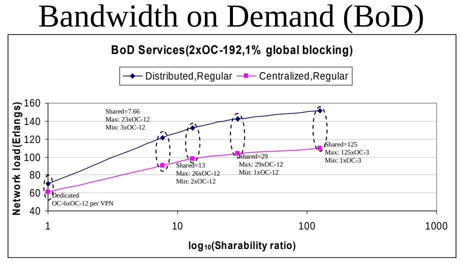
Dedicated and Shared Performance

1% Global Blocking

Private Line (PL)



Bandwidth on Demand (BoD)



$$SR = \max_{VPN,\ links}\left\{\frac{max\_VPN\_link\_capacity}{min\_VPN\_link\_capacity}\right\}$$

- Inter-VPN sharing yields much higher carried load (revenue)
- "Plateau effect" allows premium guarantees to be priced in

Overall Findings

Shared L1 VPN gives notable gains in carried load

- Averaging 50-200% for scenarios tested
- Sensitive to topological overlap, minimum guarantees

Distributed GMPLS control is better for L1 VPN

- Particularly for mesh networks using load-balancing:
  However sizeable increase in LSA routing load results
- Centralized OSS performs well for less connectivity (VPN rings)

VCAT gives selected gains

- Most effective for coarser demands, load-balancing routing
- Improved gains with higher fragmentation:
  Lower VCAT gains with higher inter-VPN sharing

Thank You!