ALTO extensions for Network Information Bandwidth Constraints & Enhanced Filtering

<draft-bernstein-alto-large-bandwidth-cases-02.txt>

Greg Bernstein, Grotto Networking

Young Lee, Huawei

84th IETF – Vancouver DC, July 29 - August 3, 2012

Outline

- Bandwidth Constraints in Optimization Problems
- Technologies and Path Choices
- Bandwidth Constraint Representation
  - Goal: Reduce amount of information shared while promoting optimization
  - Abstract paths with abstract shared bottlenecks
  - Abstract cost-constraint graphs

Bandwidth Constraints

- Individual BW demands small compared to link capacity
  - Don't need explicit bandwidth constraints, other methods such as changing path costs over time, e.g., [P4P], may be used.
- ***Large bandwidth case***, individual BW demands significant compared to link capacity
  - Optimizations must enforce link capacity constraints:

$$\sum_{(s,d)\in R} \sum_{(i,j)\in Links} q_{sd} x_{ij,sd} \leq b_{ij}$$

Where

q_{sd} The amount of bandwidth required between source and destination, ***known by application***

b_{ij} The amount of bandwidth available on link (*i, j*), ***known by network***

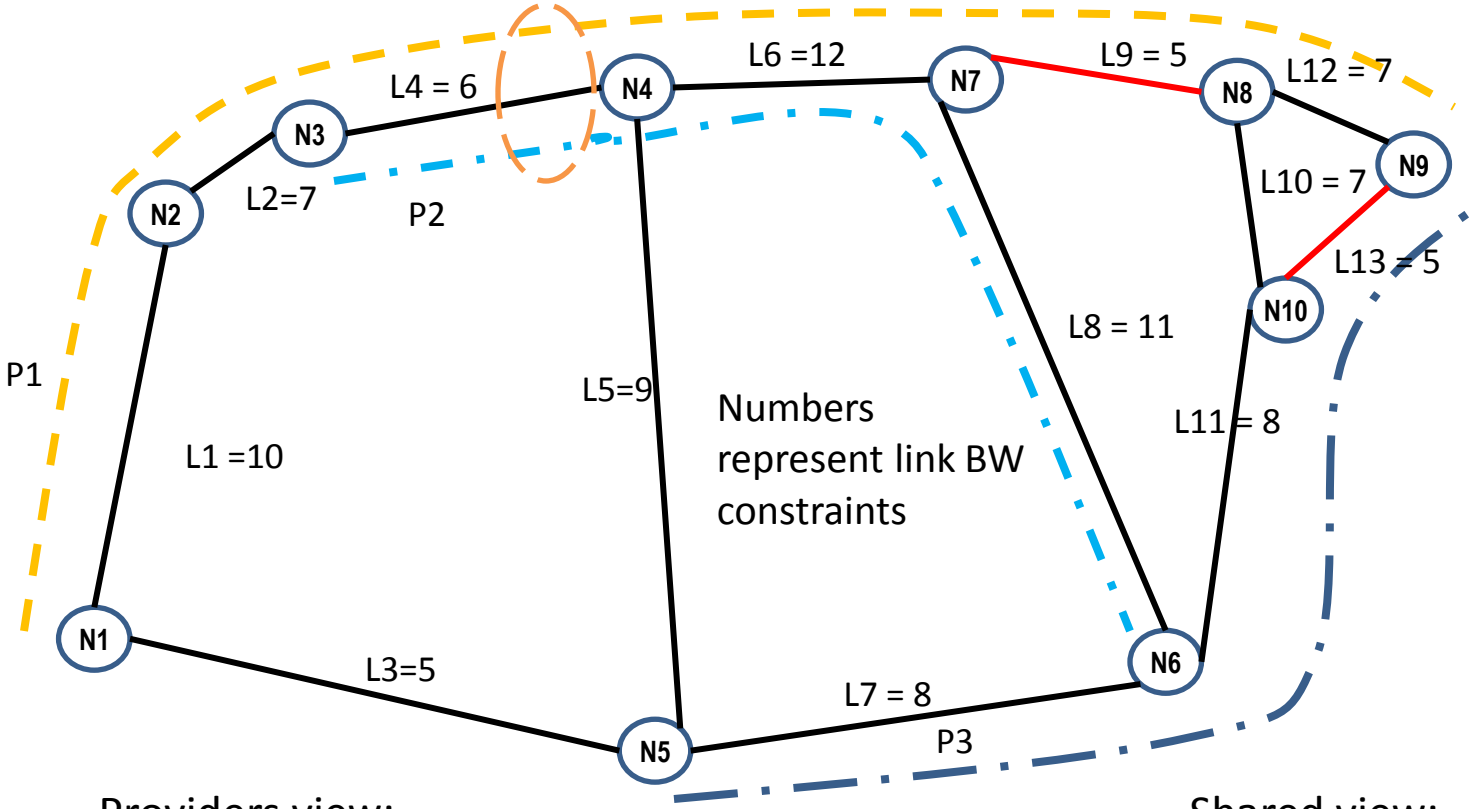
$x_{ij,sd}$ The amount of bandwidth {0,1} or [0-1] used on link (*i, j*) between source and destination, *solved for during optimization*

Path Choices and Technologies

General Categories

- Arbitrary Path Choices (*Graph representation*)
  - Connection Oriented Technologies: WDM, TDM, MPLS, InfiniBand (CO service), OpenFlow
- Limited Path Choices (*Path representation*)
  - Single path: OSPF, BGP, Ethernet etc…
  - Multiple paths: Multi-Topology Routing (OSPF), MSTP-Ethernet, WDM networks with impairments
- Limited Choices derivable from Graph (*either*)
  - OSPF, Ethernet, MSTP-Ethernet, MT routing

Paths & shared bottlenecks



Providers view:

Path	Links
P1	L1, L2, L4, L6, L9, L12
P2	L4, L6, L8
P3	L7, L11, L13

Shared view:

Path	Bottleneck Links
P1	L4, L9
P2	L4
P3	L13

Abstract Path & BW constraints

- Tentative JSON Representation
  - Named paths with their costs, constraints, and identification of **shared links**
  - Shared links with their constraints

```
object {
   PIDName source;
   PIDName dest;
   JSONNumber wt;
   JSONNumber delay;
   JSONNumber bw;
   LIDName mutual-links<1..*>;
} PathData;
```

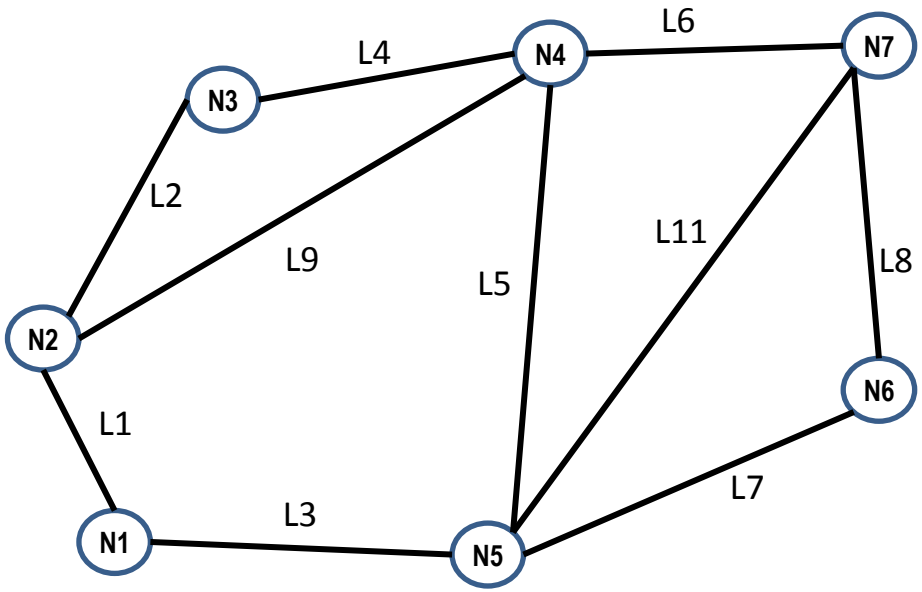
```
object {
   JSONNumber bw;
} SharedAbstractLink;
```

```
object {
   PathData [pathname]<0..*>;
   SharedAbstractLink [linkname]<0..*>;
} NetworkPathData;
```

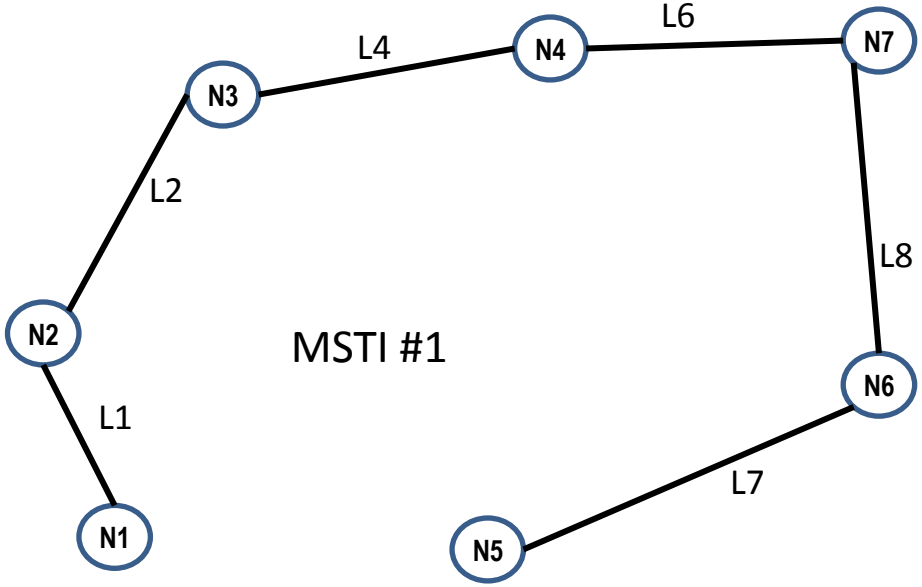
Paths Derivable from Graph Example

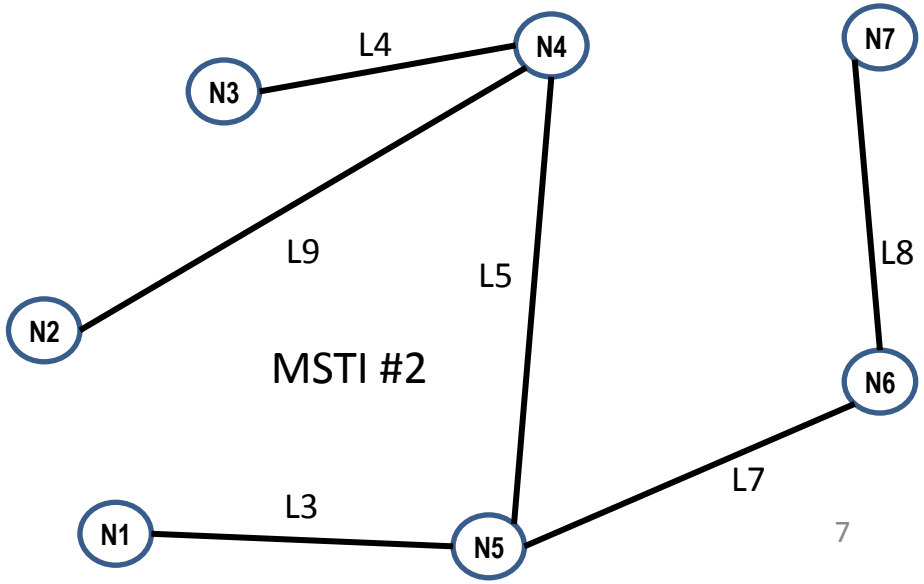
Only interested in Source-Destination nodes: N1, N3, N5, N6, N7

- MSPT-Ethernet
  - Original Graph
  - Spanning tree instances MSTI #1 and MSTI #2



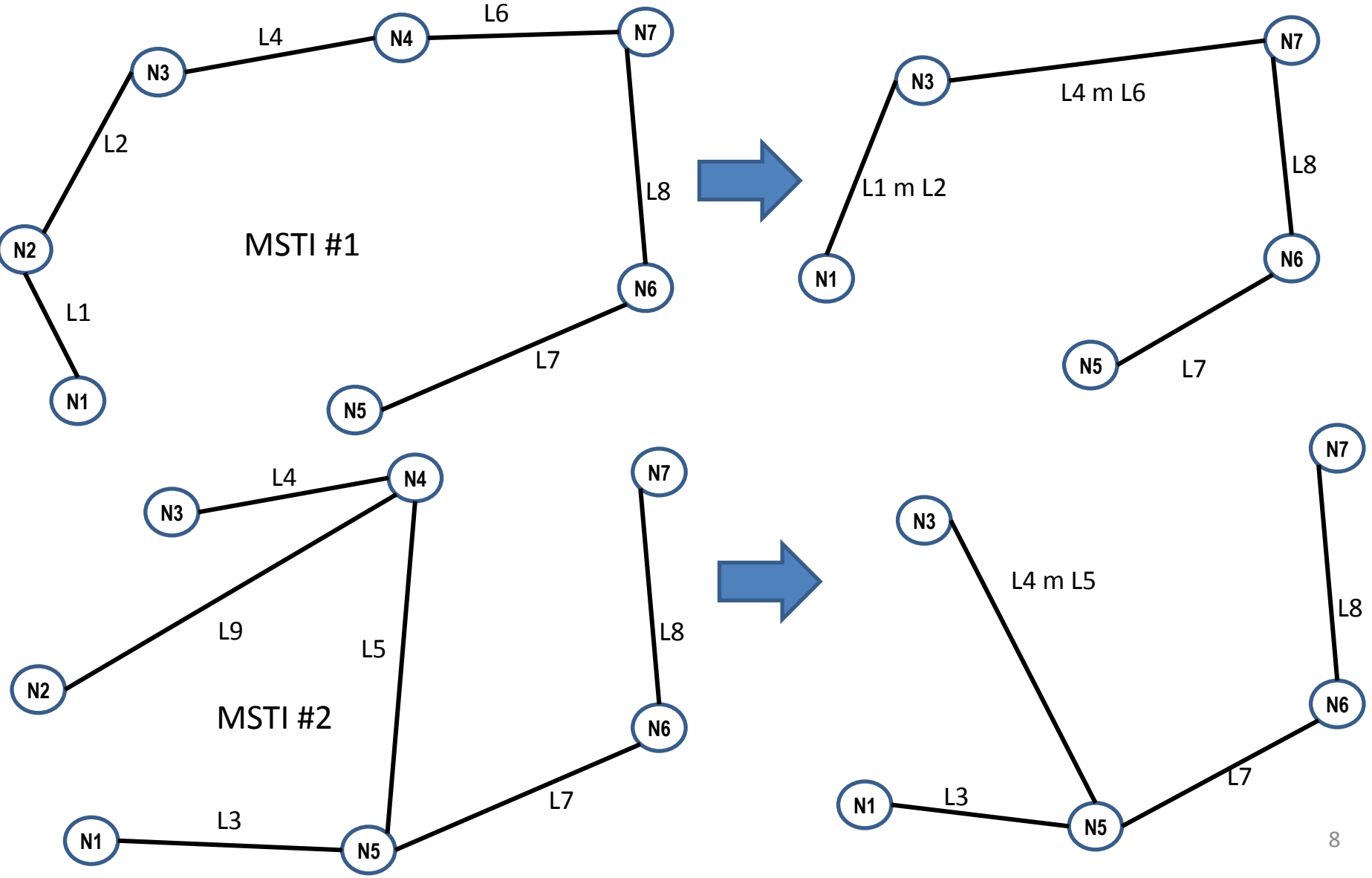
Fictitious Ethernet Network Graph





Graph Reduction (abstraction) Example

Only interested in Source-Destination nodes: N1, N3, N5, N6, N7



Abstract Graphs

- Enhanced Tentative JSON Representation
  - Link Data, Graph Data, Multiple-Graph data

```
object {
   NIDName aend;
   NIDName zend;
   JSONNumber wt;
   JSONNumber delay;
   JSONNumber bw;
  // Other costs could be added
  // use a multi-cost mechanism?
} LinkData
```

```
object {
   LinkData [lidname]<0..*>; // Link id (LID)
} NetworkGraphData;
```

```
object {
   VersionTag     map-vtag;
  NetworkGraphData [graphname]<1..*>;
   // other information such as graph choice
   // restrictions or routing restrictions.
} InfoResourceNetwork;
```

Enhanced Filtering of paths choices or graph extent

- Cost Limits
  - routing cost, total delay, delay variation, etc…
  - Can reduce the number of paths or extent of graph returned by network
- User demand limits
  - Previous reductions are based on topology and link constraints. Sharing user demands or limits on them can allow further path/graph reduction

$$\sum_{(s,d)\in R}\ \sum_{(i,j)\in Links} q_{sd}\, x_{ij,sd} \le b_{ij}$$

Application demands

Summary and Next Steps

- This draft demonstrated usefulness of network topology abstraction and its encoding
  - Abstract paths with abstract shared bottlenecks
  - Abstract cost-constraint graphs
- This allows information hiding (from network's point of view) without compromising optimization efficiency (joint APP-NET)
- Application demand and location information can further reduce the amount of processing and data transfer from network to application.