Optimal Strategy For Graceful Network Upgrade

Ram Keralapura, UC-Davis
Chen-Nee Chuah, UC-Davis
Yueyue Fan, UC-Davis
Outline

- Introduction
- Graceful Network Upgrade Problem
- Two-Phase Framework
  - Phase-1: Finding Optimal End-State
  - Phase-2: Multistage Node Addition
- Numerical Example and Results
- Conclusions and Future Directions
Introduction

- Network Upgrade
  - Adding new nodes and links into an existing operational network
  - Does not refer to upgrading individual network components

- “Graceful” Network Upgrade
  - Network performance deterioration from the perspective of existing customers should be minimized during upgrade process
Steps for Graceful Network Upgrade

- Identifying set of potential locations where new nodes can be added
- Determining a subset of potential locations where nodes should be added and how they should be connected to the rest of the network
- Determining an ideal sequence for adding nodes and links into the network
Scenario

- Tier-1 ISP backbone networks
  - Nodes → Points of Presence (PoPs)
  - Links → IP layer virtual links

- Assumptions:
  - ISPs can reliably determine the future traffic demands when new nodes are added
  - ISPs can reliably estimate the revenue associated with the addition of every new node
  - Revenues resulting from different node additions are independent of each other
Network Performance: Customers’ View

- Link failures are the primary reason for performance deterioration in tier-1 ISP backbone networks

- Impact of link failures: Service Disruption (SD) Time and Traffic Disruption (TD)
  - Depends on the operational network conditions like shortest paths between nodes, exit points for prefixes, traffic demands, etc.

- Adding new nodes and links can change the above operational network conditions
Graceful Network Upgrade

- Given a budget constraint:
  - Find where to add new nodes and how they should be connected to the rest of the network
  - Determine optimal sequence for this upgrade

- Two-Phase Framework
  - Phase-1: Finding Optimal End State
    - Formulated as an optimization problem
  - Phase-2: Multistage Node Addition Strategy
    - Formulated as a multistage dynamic programming (DP) problem
Phase-1: Finding Optimal End State

- **Objective Function:** \( \max \{k^r\} \)

Subject to:
- **Budget constraint:**
  \[
  \frac{l_c}{2} \left\{ u_{n+g}^T (D' \otimes M') u_{n+g} - u_n^T (D \otimes M) u_n \right\} + (k')^T n_c \leq b 
  \]

- **Node degree constraint**
  \[
  \sum_{j=1}^{n+g} M'_{i,j} \geq 2k_i \quad \forall i = n+1, n+2, ... 
  \]
Phase-1: Finding Optimal End State

- SD Time constraint
  \[ \sum_{i \in L', q \in N'} S'_{i,j,q} \leq \sum_{i \in L, q \in N} S_{i,j,q} \quad \forall j \in N \]

- TD constraint
  \[ \sum_{i \in L', q \in N'} T'_{i,j,q} \leq \sum_{i \in L, q \in N} T_{i,j,q} \quad \forall j \in N \]

- Link utilization constraint
  \[ \max_{\forall i \in L'} LU_i \leq \alpha \]

- Can be solved by non-linear programming techniques – tabu search, genetic algorithms
Phase-2: Multistage Node Addition

- Assumptions:
  - Already have the solution from Phase-1
  - ISP wants to add one node and its associated links into the network at every stage
    - Hard to debug if all nodes are added simultaneously
    - Node/link construction delays
    - Sequential budget allocation
    - ...
  - Time period between every stage could range from several months to few years
Phase-2: Multistage Node Addition

- Multistage dynamic programming problem with costs for performance degradation
- Cost of node degree:

\[
c_{\text{degree}}^e(h)(i) = \frac{1}{(\kappa^e(h)(i))(\kappa^e(h)(i) - 0.5)}
\]

- Cost of link utilization:

\[
c_{\text{util}}^e(h) = \left\{ \frac{2\sigma^e(h) - \alpha}{\alpha} \right\}^{2\omega_{\text{util}}} \quad \text{where } \sigma^e(h) = \max_{\forall i \in L_e} LU_i
\]
Phase-2: Multistage Node Addition

- **Cost of SD Time:**

  \[ c_{SD}^{e,h}(i) = \left\{ \frac{2 \delta_{SD}^{e,h}(i) - \delta(i)}{\delta(i)} \right\}^{2\omega_{SD}(i)} \]

  where \( \delta^{e,h}(i) = \sum_{j \in L^e} \sum_{q \in N^e} S_{j,i,q}^e \)

- **Cost of TD:**

  \[ c_{TD}^{e,h}(i) = \left\{ \frac{2 \tau_{TD}^{e,h}(i) - \tau(i)}{\tau(i)} \right\}^{2\omega_{TD}(i)} \]

  where \( \tau^{e,h}(i) = \sum_{j \in L^e} \sum_{q \in N^e} T_{j,i,q}^e \)
Phase-2: Multistage Node Addition

- Functional equation:

\[
\Phi^e (v^e_n, h) = \max_{\forall h \in v^{e-1}_n} \left\{ -c^{e,h} + \gamma \Phi^{e+1} (v^{e+1}_n, h_1) \right\}
\]

\[
\Phi^{n'-n+1} (v^{n'-n+1}_n, h_2) = 0, \quad h_1 \in v^{e+1}_n, \quad h_2 \in 0, \quad n' = |N'|
\]
Numerical Example

revenue = \{45000, 25000, 38000, 50000\}
node cost = \{14000, 13000, 13800, 14000\}
linkcost = 1
b = 50000
Numerical Example – Phase-1 Solution

Solution from Phase-1
Numerical Example – Result

![Numerical Example Graph]

- **% increase in SD-Time**
- **% increase in TD**

**Link Utilization Threshold**

0.1, 0.2, 0.3, 0.4, 0.5
Numerical Example – Phase-2 Solution

Solution from Phase-1

- - - New Links
--- Original Links

New Nodes

Original Nodes

1 Seattle

A San Jose

B Phoenix

2 Dallas

3 Atlanta

4 Miami

Chicago

New York

D
Numerical Example – Phase-2 Solution

<table>
<thead>
<tr>
<th>Multistage Solution</th>
<th>With SD time and TD</th>
<th>Without SD time and TD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Node Added</td>
<td>Links Added</td>
</tr>
<tr>
<td>Stage-1</td>
<td>4</td>
<td>4 – C, 4 – D</td>
</tr>
<tr>
<td>Stage-2</td>
<td>2</td>
<td>2 – B, 2 – C</td>
</tr>
<tr>
<td>Stage-3</td>
<td>3</td>
<td>3 – 4, 3 – C</td>
</tr>
</tbody>
</table>

- Ignoring SD time and TD in Phase-2 results in significant performance deterioration for customers
Conclusions

- We address the problem of graceful network upgrade and propose a two phase framework.
- We showed the importance of considering performance from users’ perspective.
- Illustrated the feasibility of our approach using a simple numerical example.
- The approach is flexible and powerful, and can be easily adapted to include other constraints or scenario-specific requirements.
Future Work

- Need to relax simplifying assumptions
  - Revenue and traffic demands in the network after node additions could dynamically change because of endogenous and exogenous effects
    - Adaptive and stochastic dynamic programming
  - Revenue generated by adding one node is usually not independent of other node additions

- Applications of this framework in different environments
  - Wireless cellular networks