Network Design: Notations & Illustrations

Hongwei Zhang
http://www.cs.wayne.edu/~hzhang

Acknowledgment: the slides are based on those from Drs. Yong Liu, Deep Medhi, and Michał Pióro.
Outline

- Link-Path Formulation
- Node-Link Formulation
- Notions and Notations

- Dimensioning Problems
- Shortest-Path Routing
- Fair Networks
- Topological Design
- Restoration Design
Network Flow Example
in Link-Path Formulation

- node: generic name for routing and switching devices
- link: communication channel between nodes, directed/undirected
- path: sequence of links
- demand: source-destination pair

- demand path-flow variables: amount of flow traffic on each path

![Diagram of network flow example](image)

**Figure 2.1** (a) Three-Node Network Example. (b) All Possible Paths for the Three-Node Example
Constraints on Demand Path-Flow Variables

- Legitimate flow variables
- Demand Constraints (equalities)
- Link Capacity Constraints (inequalities)

=> set of feasible solutions
Objective Function

- Objective function:
  - design goal expressed through a function of design variables

- Objective can be in terms of
  - routing cost
    - unit routing cost of unit flow on each link
  - congestion delay
  - delay on the most congested link
  ...

Put it Together

Linear programming problem; Multi-commodity network flow problem

Optimal solution/optimal cost

$$\hat{x}_{12}^* = 5, \hat{x}_{13}^* = 7, \hat{x}_{23}^* = 8, F^* = 20 \text{ (uniqueness?)}$$
Outline

- Link-Path Formulation
- Node-Link Formulation
- Notions and Notations

- Dimensioning Problems
- Shortest-Path Routing
- Fair Networks
- Topological Design
- Restoration Design
Node-Link Formulation

- *link flow*: traffic of one demand on each link
- flow conservation
  - Source
  - Destination
  - Transit node

*Figure 2.2*  Flow View for Demand between Nodes 1 and 2
Optimization in Node-Link Formulation

\[
\begin{align*}
\text{minimize} & \quad F = \bar{x}_{12,12} + \bar{x}_{13,12} + \bar{x}_{32,12} + \bar{x}_{12,13} + \bar{x}_{13,13} + \bar{x}_{23,13} + \bar{x}_{21,23} + \bar{x}_{13,23} + \bar{x}_{23,23} \\
\text{subject to} & \quad \bar{x}_{12,12} + \bar{x}_{13,12} - \bar{x}_{13,12} + \bar{x}_{32,12} - \bar{x}_{12,12} - \bar{x}_{32,12} = \hat{h}_{12} \\
& \quad \bar{x}_{12,13} + \bar{x}_{13,13} - \bar{x}_{13,13} + \bar{x}_{23,13} - \bar{x}_{12,13} - \bar{x}_{23,13} = \hat{h}_{13} \\
& \quad \bar{x}_{21,23} + \bar{x}_{13,23} - \bar{x}_{21,23} - \bar{x}_{13,23} = \hat{h}_{23} \\
& \quad \bar{x}_{13,12} + \bar{x}_{13,13} + \bar{x}_{23,13} + \bar{x}_{21,23} + \bar{x}_{13,23} + \bar{x}_{23,23} = \hat{c}_{12} + \hat{c}_{13} + \hat{c}_{23} + \hat{c}_{32} \\
& \quad \bar{x}_{32,12} \quad \text{all } \bar{x} \text{ non-negative.}
\end{align*}
\]
Outline

- Link-Path Formulation
- Node-Link Formulation
- Notions and Notations
- Dimensioning Problems
- Shortest-Path Routing
- Fair Networks
- Topological Design
- Restoration Design
Notions and Notations

- **demand:**
  - source, destination
  - *pair label*

- **link:**
  - head, tail
  - *link label*

- **path:**
  - node-identifier-based notation
  - link-demand-path-identifier-based notation:
    - *labeled paths for each demand: “demand””path”*

<table>
<thead>
<tr>
<th>node-identifier-based</th>
<th>path identifier</th>
<th>path</th>
<th>link-demand-path-identifier-based</th>
<th>path identifier</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>132</td>
<td>1-3-2</td>
<td></td>
<td>12</td>
<td>{2,3}</td>
</tr>
<tr>
<td></td>
<td>213</td>
<td>2-1-3</td>
<td></td>
<td>32</td>
<td>{1,2}</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>2-3</td>
<td></td>
<td>31</td>
<td>{3}</td>
</tr>
</tbody>
</table>
\[ \text{minimize} \]
\[ F = x_{11} + 2x_{12} + x_{21} + 2x_{22} + x_{31} + 2x_{32} \]

\[ \text{subject to} \]
\[
\begin{align*}
    x_{11} + x_{12} & = h_1 \\
    x_{21} + x_{22} & = h_2 \\
    x_{31} + x_{32} & = h_3 \\
    x_{11} + x_{22} & \leq c_1 \\
    x_{12} + x_{21} & \leq c_2 \\
    x_{12} + x_{22} + x_{31} & \leq c_3
\end{align*}
\]
\[ (2.3.1) \]
\[ x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32} \geq 0. \]
Outline

- Link-Path Formulation
- Node-Link Formulation
- Notions and Notations
- Dimensioning Problems
- Shortest-Path Routing
- Fair Networks
- Topological Design
- Restoration Design
Dimension Problems (DP)

- DP: minimizing the cost of network links with given demand volume between node pairs which can be routed over different paths.

- Illustrative Example

**Figure 2.3** Four-Node Network Example
DP (cont.d)

- link cost

- list of candidate paths for each demand

**Figure 2.4** Four-Node Network Example: Demand Volume and Link Cost
Link-Path Incidence Relation

\[ \delta_{edp} = 1 \text{ if link } e \text{ is on path } p \text{ of demand } d, \ 0 \text{ otherwise} \]

<table>
<thead>
<tr>
<th></th>
<th>( P_{dp} )</th>
<th>( P_{11} = {2, 4} )</th>
<th>( P_{21} = {5} )</th>
<th>( P_{22} = {3, 4} )</th>
<th>( P_{31} = {1} )</th>
<th>( P_{32} = {2, 3} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
DP Formulation

\[ \text{minimize} \]
\[ F = 2y_1 + y_2 + y_3 + 3y_4 + y_5 \]

subject to
\[
\begin{align*}
  x_{11} + x_{21} + x_{22} + x_{31} + x_{32} &= 15 \\
  x_{31} + x_{32} &= 20 \\
  y_1 + y_2 + y_3 + y_4 + y_5 &= 10 \\
  x_{11} + x_{22} + x_{32} &= 15 \\
  x_{21} + x_{31} &= 20 \\
  x_{11}, x_{21}, x_{22}, x_{31}, x_{32} &\geq 0, \quad y_1, y_2, y_3, y_4, y_5 \geq 0.
\end{align*}
\]
General DP Formulation

\[
\begin{align*}
\text{minimize} & \\
\text{objective/cost function (2.4.9):} & \quad F = \sum_{e} \xi_{e} y_{e} \\
\text{subject to} & \\
\text{demand constraints (2.4.3):} & \quad \sum_{p} x_{dp} = h_{d}, \quad d = 1, 2, \ldots, D \\
\text{capacity constraints (2.4.7):} & \quad \sum_{d} \sum_{p} \delta_{e dp} x_{dp} \leq y_{e}, \quad e = 1, 2, \ldots, E \\
\text{constraints on variables:} & \quad x \geq 0, y \geq 0. \\
\end{align*}
\]

- Shortest-Path Allocation Rule for DP
  - For each demand, allocate its entire demand volume to its shortest path, with respect to links’ unit costs and candidate path.
  - If there is more than one shortest path for a demand then the demand volume can be split among the shortest paths in an arbitrary way.
Variations of DP

- Non-bifurcated (unsplittable) flows: each demand only takes single path
  - Pro.s? simple to implement
  - Con.s? higher complexity in solving the problem

- Modular link capacity: link capacity only takes discrete modular values
  - combined with single-path requirement $\Rightarrow$ NP-completeness

- Uncapacitated vs. Capacitated
Outline

- Link-Path Formulation
- Node-Link Formulation
- Notions and Notations

- Dimensioning Problems
- **Shortest-Path Routing**
- Fair Networks
- Topological Design
- Restoration Design
Shortest-Path Routing

- Link State/Distance Vector Algorithms
  - given a set of link weights, find the shortest path from one node to another
  - *how to set up link weights*

- Single Shortest-path allocation problem
  - For given link capacities and demand volumes, find a link weight setting such that the resulting shortest paths are unique and the resulting flow allocation is feasible
  - Very complex problem!
Sources of complexity

- non-bifurcated flow may not be feasible
- difficult to identify single path solution
- difficult and may be infeasible to find weight setting to induce the single path solution

**Figure 2.6** Infeasible Unique Shortest-Path Case
Shortest-path Routing with Equal Splitting

- ECMP (equal-cost multi-path) used in OSPF:
  - For a fixed destination, equally split outgoing traffic from a node among all its outgoing links that belong to the shortest paths to that destination

Different splitting for 6→7 vs. 7→6
Outline

- Link-Path Formulation
- Node-Link Formulation
- Notions and Notations
- Dimensioning Problems
- Shortest-Path Routing
- Fair Networks
- Topological Design
- Restoration Design
Fair Networks

- Fairness: how to allocate available b.w. among network users.

**Figure 2.8** Two-Link Network
Max-Min Fairness

Definition 1: A feasible rate vector $R^*$ is max-min fair if no rate $R_i$ can be increased without decreasing some $R_k$ s.t. $R_k < R_i$

Definition 2: A feasible rate vector $R^*$ is an optimal solution to the MaxMin problem iff for every feasible rate vector $\hat{R}$ with $\hat{R}_i > R^*_i$, for some user $i$, then there exists a user $k$ such that $\hat{R}_k < R^*_k$ and $\hat{R}_k < \hat{R}_i$

Fairness v.s. Efficiency (e.g., overall throughput)
Other Fairness Measures

**Proportional fairness** [Kelly, Maulloo & Tan, ’98]
- A feasible rate vector $x$ is proportionally fair if for every other feasible rate vector $y$
  \[ \sum w_i \frac{(y_i - x_i)}{x_i} \leq 0 \]
- Proposed decentralized algorithm, proved properties

**Generalized notions of fairness** [Mo & Walrand, 2000]
- $\alpha$-fairness: A feasible rate vector $x$ is $\alpha$-fair if for any other feasible rate vector $y$
  \[ \sum w_i \frac{(y_i - x_i)}{x_i^\alpha} \leq 0 \]
- Special cases: $\alpha = 1$ : proportional fairness
  $\alpha \rightarrow \infty$ : max-min fairness

Revenue objective: to max. natural logarithms of demand volumes
Outline

- Link-Path Formulation
- Node-Link Formulation
- Notions and Notations
- Dimensioning Problems
- Shortest-Path Routing
- Fair Networks
- Topological Design
- Restoration Design
Topological Design

- link cost: capacity-dependent cost + installation cost

- network cost function: 
  \[ F = \sum_e \xi_e y_e + \sum_e \kappa_e u_e \]

- additional constraint: 
  \[ y_e \leq \Delta u_e \]

- mixed-integer programming
Outline

- Link-Path Formulation
- Node-Link Formulation
- Notions and Notations
- Dimensioning Problems
- Shortest-Path Routing
- Fair Networks
- Topological Design
- Restoration Design
Restoration Design

- design for the ability to recover from link/node failures: replicate the formulation for different states

\[ s = 0 \quad s = 1 \quad s = 2 \quad s = 3 \]

\[ \text{demand } k_{10} = 3 \quad \text{demand } k_{11} = 3 \quad \text{demand } k_{12} = 3 \quad \text{demand } k_{13} = 3 \]

**Figure 2.10** A Bifurcated Solution
Restoration Design

FIGURE 2.11 A Non-Bifurcated Solution
Summary

- Link-Path Formulation
- Node-Link Formulation
- Notions and Notations

- Dimensioning Problems
- Shortest-Path Routing
- Fair Networks
- Topological Design
- Restoration Design
Exercise #0 (chapter 2)

- Find another feasible solutions (other than the one already discussed) that satisfy the set of equations and inequalities (2.1.1a) of the textbook.

TinyExam #0
Example TinyExam Questions

- Give a link-path formulation of the topological design problem

- Does the shortest-path-allocation rule apply to capacities networking dimensioning problem (i.e., network dimensioning problem where there is a finite upper bound on link capacities)?