CS 622  
Distributed Networks  
Multi-speed Access Network Design

Dr. Xiaobo Zhou  
Department of Computer Science

Review: One-speed One-center CMST Problem

° CMST problem: Given a central node $N_0$ and a set of other nodes ($N_1, ..., N_n$), a set of weights ($w_1, ..., w_n$) for each node, the capacity of a link, $W$, and a cost matrix $Cost(i, j)$, find a set of trees $T_1, ..., T_k$ such that each $N_i$ belongs to exactly one $T_j$ and each $T_j$ contains $N_0$, and

$$\sum_{i \in T_j, i > 0} w_i < W$$

$$\min \sum_{\text{Trees}} \sum_{i \in \text{Links}} Cost(\text{end1}_i, \text{end2}_i)$$

° Esau-Williams algorithm vs. Sharma algorithm
Review: Multi-speed Local Access Algorithm (MSLA)

- Assign each node the smallest link \( l \) possible to connect it to the center. For each node \( n \), compute \( \text{spare capacity}(n) = W_l - w_n \), and set \( \text{pred}(n) = 0 \) (center).
  - In Esau-Williams’ algorithm, spare capacity is fixed as \( W - 1 \).

- Create trade-off values (in heap structure) for \( n \), similar to Esau-Williams. The trade-offs represent the saving from linking site \( n \) to site \( i \) rather than directly linking it to the center.

**Difference to Esau-Williams:** Allow upgrading links to carry additional traffic (if no spare capacity for traffic aggregation)

\[
\text{Tradeoff}(n,i) = c(n,i, l) + \text{Upgrade}(i,w_n) - c(n,0,l)
\]

\( \text{Upgrade}(i,w_n) \) is the cost of adding \( w_n \) units to the links that connect \( i \) and \( 0 \) (center) by following back the predecessors.

- Add the edges as long as the tradeoffs are less than or equal to 0. Terminate when the tree is built.

Multiple-speed Links in Access Networks

- It is more natural that an access network is constructed by the use of various links which have different speed.

- In general, the cost of links increases roughly as the square root of capacity.

<table>
<thead>
<tr>
<th>Link type</th>
<th>Fixed cost</th>
<th>DIST_Cost1</th>
<th>DIST_Cost2</th>
<th>DIST1</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.6 Kbps</td>
<td>$200</td>
<td>$2.00</td>
<td>$1.40</td>
<td>300</td>
</tr>
<tr>
<td>56 kbps</td>
<td>$500</td>
<td>$5.00</td>
<td>$3.00</td>
<td>250</td>
</tr>
<tr>
<td>128 kbps</td>
<td>$750</td>
<td>$8.00</td>
<td>$4.40</td>
<td>350</td>
</tr>
</tbody>
</table>

Three typical link types used in local-access design.

- **Problem:** we need an efficient algorithm that builds a tree with links of different capacity
  - Intuitively, the tree should have small-capacity links at the ends and become “fatter” as moving toward the center.
Predecessor Function in Trees

**Definition 6.1**
A tree $T$ rooted at a node $Root$ can be represented uniquely by a predecessor function $\text{pred} : V \rightarrow V$ on the set of vertices. The predecessor function moves one step closer to the $Root$.

$\text{pred}(\text{Root}) = \text{Root}; \text{pred}(N) \neq N; \text{pred}^n(N) = \text{Root}, n > 0$

![Diagram of a tree with nodes and predecessor function values]

Ancestor Function in Trees

**Definition 6.2**
Given a tree $T$ and associated predecessor function, the ancestors of $N$ are all the nodes $N'$ such that $\text{pred}^n(N') = N$ for some $n > 0$

![Diagram of a tree with ancestors highlighted]

- Ancestors of $N_0$ are $N_1, N_2, \ldots, N_8$
- Ancestors of $N_1$ are $N_3, N_4$ and $N_6$
- Ancestors of $N_5$ are $N_7$ and $N_8$

*In a capacitated MST, the weight of all the nodes of branch (all ancestors of a son node of the Center) is limited by a fixed quantity $W$*
Multi-speed One-center CMST Problem

- Given the following
  - A set of nodes N0 (center), N1, N2, ..., Nn
  - A set of weights (w1, w2, ..., wn) for each node
  - A set of link types L1, L2, ..., Lm with capacities W1, W2, ..., Wm
  - A cost matrix C(i, j, k) giving the cost of a link of Lk b.w. Ni and Nj

- Find the tree rooted at N0, such that

\[ \sum_{N \cup \text{Ancestors}(N)} w(i) < W_{\text{Link}(N, \text{pred}(N))} \]

and

\[ \sum_{l \in \text{Links}} c(\text{end1}_l, \text{end2}_l, \text{type}_l) \] is minimum.

- Here \( \text{pred}(N) \) is a predecessor function that leads towards the root
- Ancestor of \( N \) are nodes, \( N' \), such that \( \text{pred}(n')(N) = N \) for some \( n > 0 \)
- If \( m = 1 \), the problem is reduced to CMST

Multi-speed Local Access Algorithm (MSLA)

- Assign each node the smallest link \( l \) possible to connect it to the center. For each node \( n \), compute \( \text{spare} \_\text{capacity}(n) = W_l - w_n \), and set \( \text{pred}(n) = 0 \) (center)
  - In Esau-Williams’ algorithm, spare capacity is fixed as \( W - 1 \).

- Create trade-off values (in heap structure) for \( n \), similar to Esau-Williams. The trade-offs represent the saving from linking site \( n \) to site \( i \) rather than directly linking it to the center.

**Difference to Esau-Williams:** Allow upgrading links to carry additional traffic (if no spare capacity for traffic aggregation)

\[ \text{Tradeoff}(n, i, l) = c(n, i, l) + \text{Upgrade}(i, w_n) - c(n, 0, l) \]

\( \text{Upgrade}(i, w_n) \) is the cost of adding \( w_n \) units to the links that connect \( i \) and 0 (center) by following back the predecessors.

- Add the edges as long as the tradeoffs are less than or equal to 0. Terminate when the tree is built.
An MSLA Example

- Three types of links (9.6 Kbps, 56 Kbps, 128 Kbps), 50% utilization
  - Link weights: $W_0 = 4800$, $W_1 = 28000$, $W_2 = 64000$

- Initial state
  - $\text{spare}_{\text{capacity}}(1) = 28000 - 20000 = 8000$ (link type 1, 56 Kbps)
  - $\text{spare}_{\text{capacity}}(2) = 4800 - 2400 = 2400$ (type 0, 9.6 Kbps)
  - $\text{spare}_{\text{capacity}}(3) = 28000 - 9600 = 18400$ (link type 1, 56 Kbps)
  - $\text{spare}_{\text{capacity}}(4) = 4800 - 4800 = 0$ (link type 0, 9.6 Kbps)

- Cost matrix $c(i, j, k)$ is NOT given in details

MSLA (2)

- $N_2$ is furthest away from $N_0$, and it is closest to $N_4$.

- For $N_2$ (traffic weight 2400) to go through $N_4$, it requires link (4,0) to upgrade from 9.6 Kbps to 56 Kbps. The upgrade cost is calculated as
  - $\text{upgrade}(4, 2400) = c(4,0,1) - c(4,0,0)$
  - Tradeoff $\tau(4) = c(2,4,0) + (c(4,0,1)-c(4,0,0)) - c(2,0,0) > 0$, Not pick.

- Next, $N_4$ is furthest away from $N_0$, and it is closest to $N_3$
  - $N_4$ goes through $N_3$, no upgrade is needed, since $\text{spare}_{\text{capacity}}(3) > 4800$
  - It is the best tradeoff
Next, the most attractive tradeoff is to route N2 through N3
• no upgrade is needed, since spare_capacity (3) = 28000 − 9600 − 4800 = 13,600 > W2 (2400)

Finally, connect N3 to N1 and upgrade (1,0) to 128 Kbps link
Esau-Williams: 20 Nodes with 9.6 Kbps Links

- The design costs $26,963, little use of traffic aggregation by sharing links (10 sites share links, all other star into the center N0)
  - Link capacity is not enough for supporting traffic aggregation

Esau-Williams: 20 Nodes with 56 Kbps Links

- More expensive design, costing $30,160, though more traffic aggregation by sharing links
  - Too much link capacity wasted on the peripheral of the network
MSLA: 20 nodes with Multi-speed links

- A center D56 tree involving (N0, N2, N5, N10, N17) and a peripheral D96 tree connecting other nodes; Cost: $22,760