Routing

Problem: Given more than one path from source to destination, which one to take?

Features:

- Architecture
- Algorithms
- Implementation
- Performance
Architecture

Hierarchical routing:

\[ \rightarrow \text{Internet: intra-domain vs. inter-domain routing} \]

\[ \rightarrow \text{separate decision making} \]
Granularity

- Router
- Domain: autonomous system (AS)
  \[\rightarrow 16\text{ bit identifier}\]

Network representation

- Router graph
- AS graph

\[\text{Stub AS} \quad \text{Transit AS} \quad \text{Transit AS} \quad \text{Transit AS} \quad \text{Stub AS} \]

\[\text{Customer} \quad \text{Provider}\]
Route or path: criteria of goodness

- Hop count
- Delay
  \[ \rightarrow \text{composed of three parts} \]
- Bandwidth
  \[ \rightarrow \text{available bandwidth} \]
- Loss rate

Composition of goodness metric:

\[ \rightarrow \text{quality of end-to-end path} \]

- Additive: hop count, delay
- Min: bandwidth
- Multiplicative: loss rate
Goodness of routing:

→ assume \( N \) users or sessions

→ suppose path metric is delay

• System optimal routing

→ choose paths to minimize \( \sum_{i=1}^{N} D_i \)

• User optimal routing

→ each user \( i \) chooses path to minimize \( D_i \)

→ selfish actions
Pros/cons:

- System optimal routing:
  - Good: minimizes delay for the system as a whole
  - Bad: complex and difficult to scale

- User optimal routing:
  - Good: simple
  - Bad: may not make efficient use of resources
    → utilization

Some pitfalls of user optimal routing:

→ stemming from selfishness

- Fluttering or ping pong effect
- Braess paradox
Braess paradox example:

- 6 users sending 1 Mbps traffic
- Delay on shared link increases with traffic volume $x$
- Users make routing decisions one after the other

$\begin{align*}
\text{User 1} & \rightarrow 5x + 1 \rightarrow \text{B} \\
\text{User 2} & \rightarrow \text{A} \\
\text{User 3} & \rightarrow \text{C} \\
\text{User 4} & \rightarrow \text{A} \\
\text{User 5} & \rightarrow \text{C} \\
\text{User 6} & \rightarrow \text{A}
\end{align*}$

- 3 users will take $A \rightarrow B \rightarrow D$
- 3 users will take $A \rightarrow C \rightarrow D$
- total delay per user: $(5 \cdot 3 + 1) + (3 + 25) = 44$
Resource provisioning:

\[ \text{high bandwidth link is added between } B \text{ and } C \]

- User 1: \( A \to B \to C \to D \) (13)
- User 2: \( A \to B \to C \to D \) (23)
- User 3: \( A \to B \to C \to D \) (33)
- User 4: \( A \to B \to C \to D \) (43)
- User 5: \( A \to B \to D \) (52)
- User 6: \( A \to C \to D \) (52)
Adding extra link should improve things, but has the opposite effect

\[\rightarrow\] paradox possible due to selfishness

\[\rightarrow\] D. Braess (1969)

\[\rightarrow\] cannot arise in system optimal routing

\[\rightarrow\] i.e., cooperative routing

Adam Smith: let the “invisible hand” do its work

\[\rightarrow\] doesn’t always lead to best outcome

\[\rightarrow\] capitalism vs. communism

Modus operandi of the Internet: user optimal routing

\[\rightarrow\] simplicity wins the day
Algorithms

Find short, in particular, shortest paths from source to destination.

Key observation on shortest paths:

- Assume \( p \) is a shortest path from \( S \) to \( D \)
  \[ \rightarrow S \xrightarrow{p} D \]

- Pick any intermediate node \( X \) on the path

- Consider the two segments \( p_1 \) and \( p_2 \)
  \[ \rightarrow S \xrightarrow{p_1} X \xrightarrow{p_2} D \]

- The path \( p_1 \) from \( S \) to \( X \) is a shortest path, and so is the path \( p_2 \) from \( X \) to \( D \)
Illustration:

\[ S \overset{p}{\longrightarrow} D \]

\begin{itemize}
  \item reverse implication need not hold
  \item suggests algorithm for finding shortest path
\end{itemize}
Procedure: Grow a routing tree $T$ rooted at source $S$

\[ \text{initially } T \text{ only contains } S \]

1. Find a node $X$ with shortest path from $S$.
\[ \text{there may be more than one such node} \]
\[ \rightarrow \text{add } X \text{ (and path } S \xrightarrow{p} X \text{) to routing tree } T \]

2. Find node $Y$ with shortest path from $T$.
\[ \text{update existing paths if going through } Y \text{ is shorter} \]
\[ \rightarrow \text{uses shortest path decomposition property} \]

3. Repeat step two until no more nodes left to add.

Observations:

\[ \rightarrow \text{ once node is added, it’s final (no backtracking)} \]
\[ \rightarrow \text{ builds minimum spanning tree routed at } S \]
\[ \rightarrow \text{ Dijkstra’s algorithm} \]
Remarks:

- Running time: $O(n^2)$ time complexity
  
  $\rightarrow n$: number of nodes

- Can also be run “backwards”
  
  $\rightarrow$ start from destination $D$ and go to all sources
  
  $\rightarrow$ single-destination/all-source shortest path

- Source $S$ requires global link distance knowledge
  
  $\rightarrow$ centralized algorithm (center: source $S$)
  
  $\rightarrow$ every router runs Dijkstra with itself as source
• Internet protocol implementation
  → OSPF (Open Shortest Path First)
  → link state algorithm

• Minimum spanning tree routed at $S$:
  → multicasting: multicast tree
  → standardized but not implemented on Internet
Distributed/decentralized shortest path algorithm:

\[ \rightarrow \text{ Bellman-Ford algorithm} \]

\[ \rightarrow \text{ based on shortest path decomposition property} \]

Key procedure:

- Each node \( X \) maintains current shortest distance to all other nodes
  \[ \rightarrow \text{ a distance vector} \]

- Each node advertises to neighbors its current best distance estimates

- A node \( X \), upon receiving an update from neighbor \( Y \), performs update: for all \( Z \)
  \[
  d(X, Z) \leftarrow \min\{ d(X, Z), \ d(Y, Z) + \ell(X, Y) \}
  \]

\( \ldots \) same criterion as Dijkstra’s algorithm
Remarks:

• Running time: $O(n^3)$

• Each source or router only talks to neighbors
  → local interaction

  → no need to send update if no change

  → if change, entire distance vector must be sent

• Knows shortest distance, but not path
  → just the next hop is known

• Elegant but additional issues compared to Dijkstra’s algorithm
  → e.g., stability

• Internet protocol implementation
  → RIP (Routing Information Protocol)