Routing vs. Forwarding

- Inspired by the the Locator/ID split work
  - See Locator/ID Separation Protocol (LISP)
- Let’s make a distinction between infrastructure nodes $V$ and destinations $D$.
- Assume $V \cap D = \{\}$
- $M$ is a $V \times D$ mapping matrix
  - $M(v, d) \neq \infty$ means that destination (identifier) $d$ is somehow attached to node (locator) $v$
Simple example of forwarding = routing + mapping

Simple example of forwarding = routing + mapping

simple example of forwarding = routing + mapping

More Interesting Example: “Hot-Potato” Idiom — find attachment that is closest

More Interesting Example: “Hot-Potato” Idiom — find attachment that is closest

More Interesting Example: “Hot-Potato” Idiom — find attachment that is closest

More Interesting Example: “Hot-Potato” Idiom — find attachment that is closest

More Interesting Example: “Hot-Potato” Idiom — find attachment that is closest
General Case

Give $G = (V, E)$

A $|V| \times |V|$ (left) routing matrix $L$ solves an equation of the form

$$L = (A \otimes L) \oplus I,$$

over semiring $S$.

$D$ is the set of destinations.

$A |V| \times |D|$ forwarding matrix is defined as

$$F = L \triangleright M,$$

over some structure $(N, \Box, \triangleright)$, where $\triangleright \in S \rightarrow (N \rightarrow N)$.

forwading = path finding + mapping

Does this make sense?

$$F(i, d) = (L \triangleright M)(i, d) = \Box_{q \in V} L(i, q) \triangleright M(q, d).$$

- Once again we are leaving paths implicit in the construction.
- Routing paths are best paths to egress nodes, selected with respect to $\Box$-minimality.
- $\Box$-minimality can be very different from selection involved in path finding.
When we are lucky ...

<table>
<thead>
<tr>
<th>matrix</th>
<th>solves</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A^*$</td>
<td>$L = (A \otimes L) \oplus I$</td>
</tr>
<tr>
<td>$A^* \triangleright M$</td>
<td>$F = (A \triangleright F) \Box M$</td>
</tr>
</tbody>
</table>

When does this happen?

When $(N, \square, \triangleright)$ is a (left) semi-module over the semiring $S^a$.

---

A model of Internet routing using semi-modules. John N. Billings and Timothy G. Griffin. RelMiCS11/AKA6 2009

---

(Left) Semi-modules

- $(S, \oplus, \otimes, 0, 1)$ is a semiring.

A (left) semi-module over $S$

Is a structure $(N, \square, \triangleright, 0_N)$, where

- $(N, \square, 0_N)$ is a commutative monoid
- $\triangleright$ is a function $\triangleright : S \rightarrow (N \rightarrow N)$
- $(a \otimes b) \triangleright m = a \triangleright (b \triangleright m)$
- $0 \triangleright m = 0_N$
- $s \triangleright 0_N = 0_N$
- $1 \triangleright m = m$

and distributivity holds,

- **SMLD**: $s \triangleright (m \square n) = (s \triangleright m) \square (s \triangleright n)$
- **SMRD**: $(s \oplus t) \triangleright m = (s \triangleright m) \square (t \triangleright m)$
Example: Hot-Potato

S idempotent and selective

\[
S = (S, \oplus_S, \otimes_S) \\
T = (T, \oplus_T, \otimes_T) \\
\triangleright_{fst} \in S \to (S \times T) \to (S \times T) \\
s_1 \triangleright_{fst} (s_2, t) = (s_1 \otimes_S s_2, t)
\]

Hot(\(S, T\)) = (\(S \times T, \vec{\times}, \triangleright_{fst}\)),

where \(\vec{\times}\) is the (left-to-right) lexicographic product of \(\oplus_S\) and \(\oplus_T\).

Define \(\triangleright_{hp}\) on matrices

\[
(L \triangleright_{hp} M)(i, d) = \vec{\times}_{q \in V} L(i, q) \triangleright_{fst} M(q, d)
\]

Example of hot-potato routing

![Diagram of hot-potato routing]

Matrix that solves

\[
A^* \triangleright_{hp} M = L = (A \otimes L) \oplus I \\
F = (A \triangleright_{hp} F) \times M
\]

Routing matrix

\[
M = \begin{bmatrix}
  d_1 & d_2 \\
  \infty & \infty \\
  (0, 3) & \infty \\
  \infty & \infty \\
  \infty & (0, 1) \\
  (0, 2) & (0, 3)
\end{bmatrix}
\]

Mapping matrix

\[
F = \begin{bmatrix}
  d_1 & d_2 \\
  (2, 3) & (4, 3) \\
  (0, 3) & (4, 3) \\
  (3, 2) & (3, 3) \\
  (7, 2) & (0, 1) \\
  (0, 2) & (0, 3)
\end{bmatrix}
\]
Example: Cold-Potato

\[ S = (S, \oplus_S, \otimes_S) \]
\[ T = (T, \oplus_T, \otimes_T) \]
\[ \triangleright_{\text{fst}} \in S \rightarrow (S \times T) \rightarrow (S \times T) \]
\[ s_1 \triangleright_{\text{fst}} (s_2, t) = (s_1 \otimes_S s_2, t) \]

Cold(S, T) = (S \times T, \ltimes, \triangleright_{\text{fst}}),
where \ltimes is the (right-to-left) lexicographic product of \oplus_S and \oplus_T.

Define \triangleright_{\text{cp}} on matrices

\[(L \triangleright_{\text{cp}} M)(i, d) = \ltimes_{q \in V} L(i, q) \triangleright_{\text{fst}} M(q, d)\]

Example of cold-potato routing

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 4 \\
6 & 5 & 4 & 3 \\
1 & 6 & 4 & 3 \\
\end{array}
\]

\[
\begin{bmatrix}
1 & \infty & \infty \\
2 & (0, 3) & \infty \\
3 & \infty & \infty \\
4 & \infty & (0, 1) \\
5 & (0, 2) & (0, 3) \\
\end{bmatrix}
\]

Mapping matrix

\[
\begin{bmatrix}
1 & (4, 2) & (5, 1) \\
2 & (4, 2) & (9, 1) \\
3 & (3, 2) & (4, 1) \\
4 & (7, 2) & (0, 1) \\
5 & (0, 2) & (7, 1) \\
\end{bmatrix}
\]

Routing matrix

\[
\begin{bmatrix}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\end{bmatrix}
\]