Routing algorithms and traffic engineering
Routing algorithms and traffic engineering

- MPLS and OSPF
- Traffic engineering
- Linear programming
Network layer functions

Three important functions:

- **path determination**: route taken by packets from source to destination
- **forwarding**: move packets from router’s input to appropriate router output
- **call setup**: some network architectures require router call setup along path before data flows
Network layer functions

Three important functions:

- **path determination**: route taken by packets from source to destination
- forwarding: move packets from router’s input to appropriate router output
- call setup: some network architectures require router call setup along path before data flows
Routing

Routing protocol

Goal: determine “good” path (sequence of routers) thru network from source to dest.

Graph abstraction for routing algorithms:

- graph nodes are routers
- graph edges are physical links
  - link cost: delay, $ cost, or congestion level

“good” path:
  - typically means minimum cost path
  - other def’s possible
Routing algorithm classification

Global or decentralized information?

Global:
- all routers have complete topology, link cost info
- “link state” algorithms (Dijkstra)

Decentralized:
- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- “distance vector” algorithms (Bellman Ford)
OSPF (Open Shortest Path First)

- link state protocol
  - link costs between 0 and 65,535
  - Cisco recommendation - link cost $\propto 1/(\text{link capacity})$
  - rapid, loop-free convergence, scales well
  - topology map at each node, route computation using Dijkstra’s algorithm
  - OSPF advertisement carries one entry per neighbor router, advertisements flooded to entire Autonomous System

- multiple equal-cost paths allowed: flow equally split on all outgoing links belonging to shortest paths

- IS - IS (intermediate system-intermediate system) similar
  - Cisco standard
Routing vs switching

- **routing**: based on address lookup, max prefix match
  - search operation
  - complexity $O(\log n) - O(n)$

- **switching**: based on circuit numbers
  - indexing operation
  - complexity $O(1)$
  - scalable to large networks

$\Rightarrow$ **MPLS**

Multiprotocol Label Switching
History: Ipsilon’s IP switching

- developed by Ipsilon (1996)
- routing software in every ATM switch
- initially, packets reassembled by routing software, forwarded to next hop
- long term flows transferred to separate VCs
Ipsilon’s IP switching

ATM VCs set up when new IP “flows” seen, i.e., “data-driven” VC setup

- flow oriented traffic: ftp, ssh, http, multimedia
- short-lived traffic: DNS, SMTP, NTP
- Ipsilon claimed 90% of bytes flow-oriented
- runs as added software on ATM switch (12,000 lines of code)
Alphabet Soup!

- CSR Cell Switched Router
- ISR Integrated Switch and Router
- LSR Label Switching Router
- TSR Tag Switching Router
- Multi layer switches
- Direct IP
- FastIP
- PowerIP

MPLS - IETF standard
MPLS concept: route at edge, switch in core
MPLS

- flows assigned labels, routing along Label Switched Path
- finer granularity for routing
  - can allow uneven traffic split
- not tied to any route computation algorithm
Traffic engineering

Goal:
- set up long term routes to achieve some optimization objective
- knowledge of topology
- traffic matrix
  - $K$ - set of origin destination flows
  - $k \in K$, $d_k$ - demand, $s_k$ - source, $t_k$ - destination
- optimization criteria
  - minimize maximum utilization
  - keep utilizations below 60%
How does one set link weights?
Digression – linear programming
Linear program (standard form)
(D.G. Lunenberger, Linear and Nonlinear Programming)

\[
\text{minimize } \sum_{i=1}^{n} c_i x_i \\
\text{subject to } \sum_{j=1}^{n} a_{i,j} x_j = b_i, \quad i = 1, \ldots, m \\
x_j \geq 0, \quad j = 1, \ldots, n
\]

- polynomial time solution in \(n, m\)
Surplus variables
Slack variables
Free variables
Example: computing optimal routes

- topology \( G = (V, E) \)
- \( K \) - set of origin destination flows
  - \( k \in K, s_k \) - source, \( t_k \) - destination
- set of given link weights \( \{w_{ij}: (i,j) \in E\} \)
- \( X_{ij}^k \) fraction of flow \( k \) going over \( (i,j) \in E \)
Example

- decomposes into separate problems per flow $k \in K$

$$\min \sum_{(i,j) \in E} w_{ij} X_{ij}^k$$

s.t. $$\sum_{j:(i,j) \in E} X_{ij}^k - \sum_{j:(j,i) \in E} X_{ji}^k = 0, \quad i \neq s_k, t_k$$

$$\sum_{j:(s_k,j) \in E} X_{s_k,j}^k - \sum_{j:(j,s_k) \in E} X_{j,s_k}^k = 1,$$

$$\sum_{j:(t_k,j) \in E} X_{t_k,j}^k - \sum_{j:(j,t_k) \in E} X_{j,t_k}^k = -1,$$

$$X_{ij}^k \geq 0$$
Interpretation

- Let \( \{ X_{ij}^k \} \) be optimal solutions.

- If \( \{ X_{ij}^k \} \) take values 0 and 1, corresponds to shortest paths.

- If \( \{ X_{ij}^k \} \) take other values, there exist multiple shortest paths.
Linear program

minimize \( \sum_i c_i x_j \)
subject to \( \sum_j a_{ij} x_j = b_i, i = 1, \ldots, m \) \( \rightarrow \) subject to \( A x = b \)

\( x_j \geq 0, \quad j = 1, \ldots, n \)
\( x \geq 0, \)

\( x^0 \) is feasible if \( A x^0 = b \) and \( x^0 \geq 0 \)
Basic solutions
Theorem of LP

Given a linear program

1. if \( \exists \) a feasible solution, there exists a basic feasible solution
2. if \( \exists \) an optimal solution, there exists a basic optimal solution

Simplex algorithm searches basic solutions
Dual problem (symmetric form)

**Primal**

minimize $c^T x$

subject to $Ax \geq b$

$x \geq 0,$

**Dual**

maximize $y^T b$

subject to $y^T A \leq c^T$

$y \geq 0$
Dual problem (asymmetric form)

**Primal**

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0,
\end{align*}
\]

**Dual**

\[
\begin{align*}
\text{maximize} & \quad y^T b \\
\text{subject to} & \quad y^T A \leq c^T
\end{align*}
\]
Dual problem: properties.

- if $x, y$ feasible, then $c^T x \geq y^T b$

- if $x, y$ feasible and if $c^T x = y^T b$, then $x$ and $y$ are optimal

- if either problem has finite solution, so does other, if either has unbounded solution, the other has no feasible solution.
Complementary slackness
(asymmetric form)

Let $x$ and $y$ be feasible solutions. A necessary and sufficient condition for them to be optimal is that for all $i$

1. $x_i > 0 \implies y^T A_i = c_i$
2. $x_i = 0 \iff y^T A_i < c_i$

Here $A_i$ is $i$-th column of $A$
Example: primal (P-SP)

\[
\begin{align*}
\text{min } & \sum_{(i,j) \in E} w_{ij} X_{ij}^k \\
\text{s.t. } & \sum_{j: (i,j) \in E} X_{ij}^k - \sum_{j: (j,i) \in E} X_{ji}^k = 0, \quad i \neq s_k, t_k \\
& \sum_{j: (s_k,j) \in E} X_{ij}^k - \sum_{j: (j,s_k) \in E} X_{ji}^k = 1, \\
& \sum_{j: (t_k,j) \in E} X_{ij}^k - \sum_{j: (j,t_k) \in E} X_{ji}^k = -1, \\
& X_{ij}^k \geq 0
\end{align*}
\]
Dual problem

- introduce dual variables $y^k_i, i \in V$
- dual problem, for all $k \in K$

$$\max y^k_{s_k} - y^k_{t_k}$$

s.t. $y^k_i - y^k_j \leq w_{ij}, (i, j) \in E$
Example

\[
\begin{align*}
\max & \quad U^k_{t_k} \\
\text{s.t.} & \quad U^k_j - U^k_i \leq w_{ij}, (i, j) \in E \\
& \quad U^k_{s_k} = 0
\end{align*}
\]

- \{\bar{U}^k_i\} \text{ optimal solution to dual problem}
- \bar{X}^k_{ij} > 0 \Rightarrow \bar{U}^k_j - \bar{U}^k_i = w_{ij}, \bar{U}^k_j \text{ length of shortest path from } s_k \text{ to } j
- \bar{U}^k_{t_k} \text{ length of shortest path from } s_k \text{ to } t_k
How does one set link weights for OSPF?
Traffic engineering problem: minimize maximum link utilization

- topology $G = (V, E)$
- $c_{ij}$ - capacity of link $(i, j) \in E$
- $K$ - set of origin destination flows
  - $k \in K$, $d_k$ - demand, $s_k$ - source, $t_k$ - destination
- $\alpha$ - maximum link utilization
LP formulation
LP formulation

minimize $\alpha + r \sum_{k \in K} \sum_{(i,j) \in E} X_{ij}^k$

s.t. $\sum_{j:(i,j) \in E} X_{ij}^k - \sum_{j:(j,i) \in E} X_{ji}^k = \begin{cases} 0, & i \neq s_k, t_k, k \in K \\ d_k, & i = s_k, k \in K \\ -d_k, & i = t_k, k \in K \end{cases}$

$-c_{ij}\alpha + \sum_{k \in K} X_{ij}^k \leq 0,$

$X_{ij}^k \geq 0, \quad \alpha \geq 0$
**LP formulation**

- can be many solutions with same $\alpha$
- in case of tie, want solution with short paths

$$\Rightarrow \text{add term } r \sum_{k \in K} \sum_{(i,j) \in E} X_{ij}^k$$

with small $r$ to cost

- use standard LP algorithms (Simplex) to solve

**Q:** can we find link weights so that solution comes from shortest path problem?
Duality revisited

Primal

minimize $c^T x$
subject to $Ax = b_1$
$A'x \geq b_2$
$x \geq 0,$

Dual

maximize $y_1^T b_1 + y_2^T b_2$
subject to $y_1^T A + y_2^T A' \leq c^T$
$y_2 \geq 0$

- equality constraints in primal $\Rightarrow$ free variables in dual
LP formulation

minimize $\alpha + r \sum_{k \in K} \sum_{(i, j) \in E} X_{ij}^k$

s.t. $\sum_{j: (i, j) \in E} X_{ij}^k - \sum_{j: (j, i) \in E} X_{ji}^k = \begin{cases} 0, & i \neq s_k, t_k, k \in K \\ d_k, & i = s_k, k \in K \\ -d_k, & i = t_k, k \in K \end{cases}$

$c_{ij} \alpha - \sum_{k \in K} X_{ij}^k \geq 0,$

$X_{ij}^k \geq 0, \quad \alpha \geq 0$
Dual formulation

- Decision variables: \( \{ U_i^k \}, \{ W_{ij} \} \)

\[
\begin{align*}
\text{max} & \quad \sum_{k \in K} d_k U_{tk}^k \\
\text{s.t.} & \quad U_j^k - U_i^k \leq W_{ij} + r, \quad k \in K, (i, j) \in E \\
& \quad \sum_{(i, j) \in E} c_{ij} W_{ij} = 1, \\
& \quad W_{ij} \geq 0, U_{sk}^k = 0
\end{align*}
\]
Properties of primal-dual solutions

- optimal solution to primal problem \( \{\bar{X}_{ij}^k\} \), \( \bar{\alpha} \)
  - dual problem \( \{\bar{U}_i^k\}, \{\bar{W}_{ij}\} \),
- if \( \bar{X}_{ij}^k > 0 \), then \( \bar{U}_j^k - \bar{U}_i^k = \bar{W}_{ij} + r \)
- can think of \( \bar{U}_j^k \) as shortest path distance from \( s_k \) to \( j \) when link weights are \( \{\bar{W}_{ij} + r\} \)

Therefore: solution to TE problem is also solution to shortest path problem with

\[
w_{ij} = \bar{W}_{ij} + r
\]
Link weight assignment

- works for rich set of cost functions
- example:

$$\Phi = \sum_{(i,j) \in E} \Phi_{ij} \left( \sum_{k \in K} d_k X_{ij}^k \right)$$

- where $\Phi_{ij}$ are piecewise linear
Issues

- solutions are flow specific - need destination specific solutions
  - not a big deal, can reformulate to account for this
- solutions may not support equal split rule of OSPF
  - accounting for this yields NP-hard problem
  - modify IP routing
One approach to overcome “splitting problem” (Sridharan, et al ’05)

- current routing tables have thousands of routing prefixes
- instead of routing each prefix on all equal cost paths, selectively assign next hops to (each) prefix
  - i.e., remove some equal cost next hops assigned to prefixes
- goal: to approximate optimal link load
Example:

EQUAL-SUBSET-SPLIT

Prefix A: Hops k,l
Prefix B: Hops k,l
Prefix C: Hops j,l
Prefix D: Hops j,l

Prefixes: A B C D
2.5 + 0.5 + 4 + 5 = 12

Prefixes: A
2.5 + 0.5 = 3

Prefixes: B
4 + 5 = 9

Prefixes: C D

Prefix A: 5
Prefix B: 1
Prefix C: 8
Prefix D: 10

Prefixes: A B C D
Prefixes: C D
Prefixes: A B
Advantages

- simple greedy “max-load” heuristic
- requires no change in data path
- can leverage existing routing protocols
- current routers have 10,000s of routes in routing tables
  - provides large degree of flexibility in next hop allocation to match optimal allocation
Examples

Synthetic topology

Sprint topology

Max load heuristic exhibits good performance
Summary

- can use OSPF/ISIS to support traffic engineering objectives
- performance objectives are computing link weights
- equal splitting rule complicates problem
  - heuristics provide good performance
  - small changes to IP routing provide in better performance
- MPLS suffers none of these problems
Traffic engineering

Goal:
- set up long term routes to achieve some optimization objective
- knowledge of topology
- traffic matrix
  - $K$ - set of origin destination flows
  - $k \in K$, $d_k$ - demand, $s_k$ - source, $t_k$ - destination
- optimization criteria
  - minimize maximum utilization
  - keep utilizations below 60%

$T = [t_{ij}]$

\[ i, \text{ source, } j, \text{ dest.} \]

\[ \text{traffic load between } i, j. \]
Linear program (standard form)

minimize $\sum_{i=1}^{n} c_i x_i$

subject to $\sum_{j=1}^{n} a_{i,j} x_j = b_i$, $i = 1, \ldots, m$

$x_j \geq 0$, $j = 1, \ldots, n$

- polynomial time solution in $n$, $m$
  \[ n \geq m \]

\[ \exists \text{decision variables} \quad \text{all else known} \]

linearly independent.
Surplus variables

\[
\begin{align*}
\min \sum_{i} C_i x_i \\
\{x_i\} \\
\sum_{j} a_{ij} x_j &\geq b_i, \quad i = 1, \ldots, m \\
x_j &\geq 0
\end{align*}
\]

\[
\begin{align*}
\min \sum_{i} C_i x_i \\
\sum_{j} a_{ij} x_j - y_i &= b_i \\
x_j &\geq 0, \quad y_i \geq 0
\end{align*}
\]
Slack variables

\[\begin{align*}
\min & \sum_{i} c_i x_i \\
\sum_{j} a_{ij} x_j & \leq b_i, \quad i = 1, \ldots, m \\
x_j & \geq 0
\end{align*}\]

\[\begin{align*}
\min & \sum_{i} c_i x_i \\
\sum_{j} a_{ij} x_j + y_i & = b_i \\
x_j & \geq 0, \quad y_i \geq 0
\end{align*}\]
Free variables

\[
\begin{align*}
\min & \quad \sum_i C_i x_i \\
\text{s.t.} & \quad \sum_j a_{ij} x_j = b_i \\
& \quad x_j \geq 0, \quad j = 2, 3, \ldots, n
\end{align*}
\]

0. \( x_1 = u_i - v_i \); \( u_i \geq 0, \, v_i \geq 0 \)

\[
x_1 = b_1 - \sum_{j \neq 1} a_{ij} x_j / a_{11}
\]
Example: optimal routes

- topology $G = (V,E)$
- $K$ - set of origin destination flows
  - $k \in K$, $s_k$ - source, $t_k$ - destination
- set of given link weights $\{w_{ij} : (i,j) \in E\}$
- $X^k_{ij}$ fraction of flow $k$ going over $(i,j) \in E$

$$\min \sum \sum \sum_{(i,j) \in E} w_{ij} X^k_{ij}$$

$|K|$ problems

$$\min \sum_{(i,j) \in E} w_{ij} X^k_{ij} \quad \forall k$$
\[
\min \sum_{i,j} w_{ij} x_{ij}^{k}
\]
\[\text{subject to:}\]
\[\sum_{j: (i,j)} x_{ij}^{k} - \sum_{l: (l,i)} x_{el}^{k} = 0 \quad \text{for} \quad i \neq s_k,\]

\[\sum_{j: (s_k,j)} x_{sj}^{k} - \sum_{l: (l,s_k)} x_{el}^{s_k} = 1 \quad \text{for} \quad s_k \]

\[-\sum_{j: (j,t_k)} x_{jk}^{k} + \sum_{l: (t_k,l)} x_{el}^{k} = -1 \quad \text{for} \quad t_k \]

\[x_{ij} \geq 0\]
Basic solutions

\[ \begin{align*}
& a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_m + a_{1m+1}x_{m+1} + \cdots + a_{1n}x_n = b_1 \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mm}x_m + \cdots + a_{mn}x_n = b_m \\
& \text{subject to} \quad A \mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq 0
\end{align*} \]

\[ B \mathbf{x}_b = \mathbf{b}, \quad \mathbf{x}_b \text{ - basic vector} \]

\[ A \mathbf{x}^0 = \mathbf{b}, \quad \mathbf{x}^0 \geq 0, \quad \text{if } \mathbf{x}_0 \text{ is a basic soln} \]

\[ \text{then } (n-m \text{ variables } = 0) \]
Theorem of LP

Given a Linear Program
if \( \exists \) feasible soln, then \( \exists \) basic feasible soln
if \( \exists \) an optimal soln, \( \exists \) a basic optimal soln.

Simplex algorithm searches basic solns.