Lecture 13:
Traffic Engineering

CSE 222A: Computer Communication Networks
Alex C. Snoeren

Thanks: Mike Freedman, Nick Feamster
Lecture 13 Overview

- Evolution of routing in the ARPAnet
- Today's TE: Adjusting edge weights
- Dealing with multiple paths
- Multihoming
Do IP Networks Manage Themselves?

- In some sense, yes:
  - TCP senders send less traffic during congestion
  - Routing protocols adapt to topology changes
- But, does the network run *efficiently*?
  - Congested link when idle paths exist?
  - High-delay path when a low-delay path exists?
- How should routing adapt to the traffic?
  - Avoiding congested links in the network
  - Satisfying application requirements (e.g., delay)
- ... essential questions of traffic engineering
ARPAnet Routing (1969)

- **Routing**
  - Shortest-path routing based on link metrics; distance vector

- **Metrics**
  - *Instantaneous* queue length plus a constant
  - Each node updates distance computation periodically
Problems With the Algorithm

- Instantaneous queue length
  - Poor indicator of expected delay
  - Fluctuates widely, even at low traffic levels
  - Leading to routing oscillations

- Distance-vector routing
  - Transient loops during (slow) convergence
  - Triggered by link weight changes, not just failures

- Protocol overhead
  - Frequent dissemination of link metric changes
  - Leading to high overhead in larger topologies
New ARPAnet Routing (1979)

- Averaging of the link metric over time
  - Old: Instantaneous delay fluctuates a lot
  - New: Averaging reduces the fluctuations

- Link-state protocol
  - Old: Distance-vector path computation leads to loops
  - New: Link-state protocol where each router computes shortest paths based on the complete topology

- Reduce frequency of updates
  - Old: Sending updates on each change is too much
  - New: Send updates if change passes a threshold
Performance of New Algorithm

- **Light load**
  - Delay dominated by the constant part (transmission delay and propagation delay)

- **Medium load**
  - Queuing delay is no longer negligible on all links
  - Moderate traffic shifts to avoid congestion

- **Heavy load**
  - *Very* high metrics on congested links
  - Busy links look bad to *all* of the routers
  - All routers avoid the busy links
  - Routers may send packets on longer paths
Over-Reacting to Congestion

- Routers make decisions based on old information
  - Propagation delay in flooding link metrics
  - Thresholds applied to limit number of updates
- Old information leads to bad decisions
  - All routers avoid the congested links
  - ... leading to congestion on other links
  - ... and the whole thing repeats

“SigAlert on the 5” on radio triggers back-up in Del Mar
Newer ARPAnet Metric (1987)

- Prevent over-reacting
  - Shed traffic from a congested link gradually
  - Starting with alternate paths that are just slightly longer
  - Through weighted average in computing the metric, and limits on the change from one period to the next

- Limit path length
  - Bound the value of the link metric
  - “This link is busy enough to go two extra hops”

- New algorithm
  - New way of computing the link weights
  - No change to link-state routing or shortest-path algorithm
Today: “Static” Link Weights

- Routers flood information to learn topology
  - Determine “next hop” to reach other routers…
  - Compute shortest paths based on link weights

- Link weights configured by network operator
Setting the Link Weights

- How to set the weights
  - Inversely proportional to link capacity?
  - Proportional to propagation delay?
  - Network-wide optimization based on traffic?
Measure, Model, and Control

Network-wide “what if” model

Topology/Configuration

Offered traffic

Changes to the network

Operational network

CSE 222A – Lecture 13: Traffic Engineering
Key Ingredients

● Measurement
  - Topology: monitoring of the routing protocols
  - Traffic matrix: passive traffic measurement

● Network-wide models
  - Representations of topology and traffic
  - “What-if” models of shortest-path routing

● Network optimization
  - Efficient algorithms to find good configurations
  - Operational experience to identify constraints
Optimization Problem

- **Input:** graph $G(R,L)$
  - $R$ is the set of routers
  - $L$ is the set of unidirectional links
  - $c_l$ is the capacity of link $l$

- **Input:** traffic matrix
  - $M_{i,j}$ is traffic load from router $i$ to $j$

- **Output:** setting of the link weights
  - $w_l$ is weight on unidirectional link $l$
  - $P_{i,j,l}$ is fraction of traffic from $i$ to $j$ traversing link $l$
Equal-Cost Multipath (ECMP)

Values of $P_{i,j,l}$
Objective Function

- Computing the link utilization
  - Link load: \( u_l = \sum_{i,j} M_{i,j} P_{i,j,l} \)
  - Utilization: \( u_l / c_l \)

- Objective functions
  - \( \min( \max_l (u_l/c_l)) \)
  - \( \min( \sum F(u_l/c_l)) \)
Complexity of Optimization

- NP-complete optimization problem
  - No efficient algorithm to find the link weights
  - Even for simple objective functions
  - Have to resort to searching through weight settings

- Clearly suboptimal, but effective in practice
  - Fast computation of the link weights
  - Good performance, compared to “optimal” solution
Operational Realities

- Minimize number of changes to the network
  - Changing just 1 or 2 link weights is often enough

- Tolerate failure of network equipment
  - Weights settings usually remain good after failure
  - … or can be fixed by changing one or two weights

- Limit dependence on measurement accuracy
  - Good weights remain good, despite random noise

- Limit frequency of changes to the weights
  - Joint optimization for day & night traffic matrices
Need for Inter-domain TE

- Avoid congested edge links
  - Links between domains are common points of congestion in the Internet

- Exploit upgraded link capacity
  - Operators frequently install higher-bandwidth links
  - Aim to exploit the additional capacity

- Comply with terms of peering agreement
  - For example, enforce ratio of in/out traffic on peering link
BGP Traffic Engineering

- Predict effects of changes to import policies
  - Inputs: routing, traffic, and configuration data
  - Outputs: flow of traffic through the network

![Diagram showing BGP routing model and its inputs and outputs.]

- Topology
- BGP policy configuration
- BGP routes
- Offered traffic

Flow of traffic through the network
Goals for Interdomain TE

- Predictable traffic flow changes
- Limiting the influence of neighboring domains
  - Check for consistent advertisements
  - Use BGP policies that limit the influence of neighbors
- Reduce the overhead of routing changes
  - Focus on small number of prefixes
Predictable Traffic Changes

- Avoid globally visible changes
  - Don’t do things that would result in changes to routing decisions from a neighboring AS
  - For example, make adjustments for prefixes that are only advertised via one neighbor AS
Reduce Overhead of Changes

- Group related prefixes
  - Don’t explore all combinations of prefixes
  - Simplify configuration changes
  - **Routing choices**: groups routes that have the same AS paths (lots of different granularities to choose from)

- Focus on the (small) fraction of prefixes that carry the majority of the traffic
  - E.g., top 10% of origin ASes are responsible for about 82% of outbound traffic
Multiple Paths

- Establish multiple paths in advance
  - Good use of bandwidth, withstand failures
- Disseminate link-congestion information
  - Flood thru network, piggyback on data, direct to controller
Adjust Traffic Splitting

- Source router adjusts the traffic
  - Changing traffic fraction, e.g. based on congestion
  - Often use hash-based splitting to prevent packet reordering within a flow

CSE 222A – Lecture 13: Traffic Engineering
Multi-Homing

- Reliability
  - Reduced fate sharing
  - Survive ISP failure
- Performance
  - Multiple paths
  - Select the best
- Financial
  - Leverage through competition
  - Game 95th-percentile billing model

Stub: an Autonomous System that does not provide transit
Outbound Traffic: Pick a BGP Route

- Easier to control than inbound traffic
  - IP routing is destination based
  - Sender determines where the packets go
- Control only by selecting the next hop
  - Border router can pick the next-hop AS
  - Cannot control selection of the entire path
Inbound Traffic: Influencing Others

- Harder to control than outbound traffic
  - Sender determines where the packets go
- Control only by influencing others’ decisions
  - Static configuration of the providers
  - BGP route attributes sent by the stub
  - Selective advertising of destination prefixes
Inbound Traffic: Multiple Addresses

- Multiple external addresses for a service
  - One IP address for each entry point
- Use DNS to adjust mapping of name to address
  - Give different answers to different clients
  - Adjust over time to performance, cost, traffic, etc.

Provider 1

12.34.1.0/24

5.6.7.0/24

Provider 2

12.34.1.2

5.6.7.8

12.34.1.0/24
For Next Class…

- Read and review XCP paper
- Project checkpoint due Thursday
  - Email 1-2 page summary to Bhanu
Limitations of Shortest-Path Routing

- Sub-optimal traffic engineering
  - Restricted to paths expressible as link weights
- Limited use of multiple paths
  - Only equal-cost multi-path, with even splitting
- Disruptions when changing the link weights
  - Transient packet loss and delay, and out-of-order
- Slow adaptation to congestion
  - Network-wide re-optimization and configuration
- Overhead of the management system
  - Collecting measurements and performing optimization
Explicit End-to-End Paths

- Establish end-to-end path in advance
  - Learn the topology (as in link-state routing)
  - End host or router computes and signals a path
- Routers supports virtual circuits
  - Signaling: install entry for each circuit at each hop
  - Forwarding: look up the circuit id in the table

Used in MPLS with RSVP
Label Swapping

- Problem: using VC ID along the whole path
  - Each virtual circuit consumes a unique ID
  - Starts to use up all of the ID space in the network

- Label swapping
  - Map the VC ID to a new value at each hop
    - Table has old ID, next link, and new ID
  - Allows reuse of the IDs at different links
Multi-Protocol Label Switching

- Multi-Protocol
  - Encapsulate a data packet
    - Could be IP, or some other protocol (e.g., IPX)
  - Put an MPLS header in front of the packet
    - Actually, can even build a stack of labels...

- Label Switching
  - MPLS header includes a label
  - Label switching between MPLS-capable routers
Pushing, Popping, and Swapping

- **Pushing**: add the initial “in” label
- **Swapping**: map “in” label to “out” label
- **Popping**: remove the “out” label
Constrained Shortest Path First

- Run a link-state routing protocol
  - Configurable link weights
  - Plus other metrics like available bandwidth

- Constrained shortest-path computation
  - Prune unwanted links (e.g., not enough bandwidth)
  - Compute shortest path on the remaining graph

- Signal along the path
  - Source router sends a message to pin the path to destination
  - Revisit decisions periodically, in case better options exist
Choosing Outbound Traffic

- Primary and Backup
  - Single policy for all prefixes
    » High local-pref for session to primary provider
    » Low local-pref for session to backup provider
  - Outcome of BGP decision process
    » Choose the primary provider whenever possible
    » Use the backup provider when necessary

- Load Balancing: Selectively use each provider
  - Assign local-pref across destination prefixes
  - Change the local-pref assignments over time
Inbound Traffic: Primary and Backup

- Ask your provider to be a backup
  - Provider violates “prefer customer” policy
  - … by assigning lower local-pref to customer
  - Backup link is only used if the primary link fails

**Diagram**

- Provider 1
  - 12.34.158.0/24
  - local-pref=90

- Provider 2
  - local-pref=100

- Traffic

- 12.34.158.0/24
Inbound Traffic: AS Prepending

- Make one path look longer
  - Advertise short path one way
  - ... and longer path another
  - In the hope of influencing choices
  - But, how much prepending to do?

```
12.34.158.024: (3)
```

```
12.34.158.024: (3, 3, 3)
```

CSE 222A – Lecture 13: Traffic Engineering
Inbound Traffic: Selective Advertising

- When you don’t have enough prefixes…
  - Advertise one subnet to each provider
  - Advertise the supernet to both providers
  - Traffic splits due to the longest-prefix match
  - Supernet ensures backup connectivity after failure

Causes unwanted increase in global routing tables