Traffic Engineering for ISP Networks

Jennifer Rexford

Internet and Networking Systems
AT&T Labs - Research; Florham Park, NJ
http://www.research.att.com/~jrex

Joint work with Anja Feldmann, Albert Greenberg, Carsten Lund, Nick Reingold, and Fred True, and AT&T IP Services

Outline

◆ Background
  – Internet architecture and routing protocols
  – Internet service provider backbone

◆ Traffic engineering
  – Optimizing network configuration to prevailing traffic
  – Requirements for topology, routing, and traffic info

◆ Traffic demands
  – Volume of load between edges of the network
  – Ideal measurement methodology (what we wanted...)
  – Adapted measurement methodology (...what we had)
  – Analysis of the demands on AT&T's IP Backbone
Internet Architecture

◆ Divided into Autonomous Systems
  – Distinct regions of administrative control (~8000)
  – Set of routers and links managed by a single institution
  – Service provider, company, university, …

◆ Hierarchy of Autonomous Systems
  – Large, tier-1 provider with a nationwide backbone
  – Medium-sized regional provider with smaller backbone
  – Small network run by a single company or university

◆ Interaction between Autonomous Systems
  – Internal topology is not shared between ASes
  – … but, neighboring ASes interact to coordinate routing

Autonomous Systems (ASes)

Path: 6, 5, 4, 3, 2, 1

Client → Web server
Interdomain Routing (Between ASes)

- ASes exchange info about who they can reach
- Local policies for path selection (which to use?)
- Local policies for route propagation (who to tell?)
- Policies configured by the AS’s network operator

Internet Service Provider Backbone

- neighboring providers
- modem banks, business customers, web/e-mail servers

How should traffic be routed through the ISP backbone?
**Intradomain Routing (Within an AS)**

- Routers exchange information to learn the topology
- Routers determine “next hop” to reach other routers
- Path selection based on link weights (shortest path)
- Link weights configured by AS’s network operator
- … to engineer the flow of traffic

---

**Traffic Engineering in an ISP Backbone**

- **Topology of the ISP backbone**
  - Connectivity and capacity of routers and links
- **Traffic demands**
  - Expected/offered load between points in the network
- **Routing configuration**
  - Tunable rules for selecting a path for each traffic flow
- **Performance objective**
  - Balanced load, low latency, service level agreements …

**Question:** Given the topology and traffic demands in an IP network, which routes should be used?
State-of-the-Art in IP Networks

- **Missing input information**
  - The topology and traffic demands are often unknown
  - Traffic fluctuates over time (user behavior, new appls)
  - Topology changes over time (failures, growth, reconfig)
- **Primitive control over routing**
  - The network does not adapt the routes to the load
  - The static routes are not optimized to the traffic
  - Routing parameters are changed manually by operators
  (But, other than that, everything is under control…)

---

Example: Congested Link

- **Detecting that a link is congested**
  - Utilization statistics reported every five minutes
  - Sample probe traffic suffers degraded performance
  - Customers complain (via the telephone network?)
- **Reasons why the link might be congested**
  - Increase in demand between some set of src-dest pairs
  - Failed router/link within the AS causes routing change
  - Failure/reconfiguration in another AS changes routes
- **How to determine why the link is congested??**
  - Need to know the *cause*, not just the *manifestations*!
- **How to alleviate the congestion on the link??**
**Requirements for Traffic Engineering**

- **Models**
  - Traffic demands
  - Network topology/configuration
  - Internet routing algorithms

- **Techniques for populating the models**
  - Measuring/computing the traffic demands
  - Determining the network topology/configuration
  - Optimizing the routing parameters

- **Analysis of the traffic demands**
  - Knowing how the demands fluctuates over time
  - Understanding the traffic engineering implications

---

**Modeling Traffic Demands**

- **Volume of traffic** $V(s,d,t)$
  - From a particular source $s$
  - To a particular destination $d$
  - Over a particular time period $t$

- **Time period**
  - Performance debugging -- minutes or tens of minutes
  - Time-of-day traffic engineering -- hours
  - Network design -- days to weeks

- **Sources and destinations**
  - Individual hosts -- interesting, but huge!
  - Individual prefixes -- still big; not seen by any one AS!
  - Individual edge links in an ISP backbone -- hmmm…. 
Traffic Matrix

Traffic matrix: \( V(\text{ingress}, \text{egress}, t) \) for all pairs (ingress, egress)

Problem: Multiple Exit Points

- ISP backbone is in the *middle* of the Internet
  - Multiple connections to other autonomous systems
  - Destination is reachable through multiple exit points
  - Selection of exit point depends on intradomain routes

- Problem with traditional point-to-point models
  - Want to predict impact of changing intradomain routing
  - But, a change in routing may change the exit point!
### Traffic Demand

- **Definition: \( V(\text{ingress}, \{\text{egress}\}, t) \)**
  - Entry link (ingress)
  - Set of possible exit links (\{egress\})
  - Time period (t)
  - Volume of traffic (\(V(\text{ingress},\{\text{egress}\}, t)\))

- **Avoids “coupling” problem of point-to-point model**

### Ideal Measurement Methodology

- **Measure traffic where it enters the network**
  - Input link, destination address, # bytes, and time
  - Flow-level measurement (Cisco NetFlow)

- **Determine where traffic can leave the network**
  - Set of egress links associated with each network address
  - Router forwarding tables (IOS command “show ip cef”)

- **Compute traffic demands**
  - Associate each measurement with a set of egress links
  - Aggregate all traffic with same ingress, \{egress\}, and t
Measuring Flows Rather Than Packets

- **IP flow abstraction**
  - Set of packets with “same” src and dest IP addresses
  - Packets that are “close” together in time (a few seconds)

- **Cisco NetFlow**
  - Router maintains a cache of statistics about active flows
  - Router exports a measurement record for each flow

NetFlow Data

- **Source and destination information**
  - Source and *destination* IP addresses (hosts)
  - Source and destination port numbers (application)
  - Source and destination Autonomous System numbers

- **Routing information**
  - Source and destination IP prefix (network address)
  - *Input* and output links at this router

- **Traffic information**
  - *Start* and *finish* time of flow (in seconds)
  - Total number of *bytes* and packets in the flow
Identifying Where the Traffic Can Leave

- **Traffic flows**
  - Each flow has a dest IP address (e.g., 12.34.156.5)
  - Each address belongs to a prefix (e.g., 12.34.156.0/24)

- **Forwarding tables**
  - Each router has a table to forward a packet to “next hop”
  - Forwarding table maps a prefix to a “next hop” link

- **Process**
  - Dump the forwarding table from each router
  - Identify entries where the “next hop” is an egress link
  - Identify set of egress links associated with each prefix
  - Associate flow’s dest address with the set of egress links

Locating the Set of Exit Links for Prefix \(d\)

- Prefix \(d\): exit links \(\{i, k\}\)
- Table entry: \((d, i)\)
- Table entry: \((d, k)\)
Adapted Measurement Methodology

- Measuring only at peering links
  - Measurement support directly in the interface cards
  - Small number of routers (lower management overhead)
  - Less frequent changes/additions to the network
  - Smaller amount of measurement data (~100 GB/day)

- Sufficiency of measuring at peering links
  - Large majority of traffic is interdomain
  - Measurement enabled in both directions (in and out)
  - Inference of ingress links for traffic from customers

Inbound and Outbound Flows on Peering Links

Outbound (adapted methodology)

Inbound (ideal methodology)
Inferring Ingress Links for Outbound Traffic

Outbound traffic flow measured at peering link

Identify candidate ingress links based on source IP address of traffic flow and customer IP address assignments.

Use routing simulation to trace back to the ingress links!
Computing the Traffic Demands

- **Operational data**
  - Large, diverse, lossy
  - Collected at different time intervals, across the network
  - Subject to network and operational dynamics
- **Algorithms, details, and anecdotes in paper!**

Experience with Populating the Model

- **Largely successful**
  - 98% of all traffic (bytes) associated with a set of egress links
  - 95-99% of traffic consistent with an OSPF simulator
- **Disambiguating outbound traffic**
  - 67% of traffic associated with a single ingress link
  - 33% of traffic split across multiple ingress (typically, same city!)
- **Inbound and transit traffic (ingress measurement)**
  - Results are good, since we can apply the ideal methodology
- **Outbound traffic (ingress disambiguation)**
  - Results are pretty good for traffic engineering applications
  - May want to measure at selected or sampled customer links
Proportion of Traffic in Top Demands (Log Scale)

Time-of-Day Effects (San Francisco)
### Traffic-Engineering Implications

- **Small number of demands contribute most traffic**
  - Small number of heavy demands (Zipf’s Law!)
  - Optimize routing based on the heavy demands
  - Measure a small fraction of the traffic (sample)
  - Watch out for changes in load and egress links
- **Time-of-day fluctuations in traffic volumes**
  - U.S. business, U.S. residential, & International traffic
  - Depends on the time-of-day for human end-point(s)
  - Reoptimize the routes a few times a day (three?)
- **Traffic stability? Yes and no...**

### Conclusions

- **Internet traffic engineering is hard**
  - Decentralized (over 8000 Autonomous Systems)
  - Connectionless (traffic sent as individual packets)
  - Changing (topological changes, traffic fluctuations)
- **Traffic engineering requires knowing the demands**
  - Interdomain traffic has multiple possible exit points
  - Demand as the load from entry to set of exit points
  - Not available from traditional measurement techniques
- **Measurement of traffic demands**
  - Derivable from flow-level measurements at *entry* points
  - … and “next hop” forwarding info from *exit* points
### Ongoing Work

- **Detailed analysis of traffic demands**
  - Statistical properties (how to study stability?)
  - Implications for traffic engineering

- **Online computation of traffic demands**
  - Distributed flow-measurement infrastructure
  - Real-time view of topology and reachability data
  - Online aggregation of flow data into demands

- **Network operations (“operations” research?)**
  - Efficiently detecting sudden changes in traffic or routing
  - Optimizing routes based on topology and demands
  - Getting the network to run itself… 😊

### To Learn More...

- **Traffic demands**
  - “Deriving traffic demands for operational IP networks: Methodology and experiences”
    (http://www.research.att.com/~jrex/papers/sigcomm00.ps)

- **Topology/configuration**
  - “IP network configuration for traffic engineering”
    (http://www.research.att.com/~jrex/papers/netdb.tm.ps)

- **Routing model**
  - “Traffic engineering for IP networks”
    (http://www.research.att.com/~jrex/papers/ieeenet00.ps)

- **Route optimization**
  - “Internet traffic engineering by optimizing OSPF weights”