Computer Networks

Lecture 20

Bandwidth Allocation Framework;
Intro to Network/Routing

11/12/2013

Outline

- Admin and recap
- Bandwidth allocation framework
- Network overview
- Routing overview
Admin.

- Exam 1
  - High - 75
  - Low - 44
  - Mean - 66.97
  - Std. dev - 7.3
  - Please discuss any question with Namratha

- Assignment four
  - Questions?

Admin: Web Server Benchmarking

Apache

Apache: 4.925 Gbps
Fastest student server: 7.530 Gbps
6 students receive 10 points for comparable perf to Apache
Recap: TCP/Reno and TCP/Vegas

<table>
<thead>
<tr>
<th>Congestion signal</th>
<th>TCP/Reno</th>
<th>TCP/Vegas</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>loss rate $p$</td>
<td>queueing delay $T_{queueing}$</td>
</tr>
</tbody>
</table>

Dynamics ($x'$)

- TCP/Reno: $\dot{x} = \frac{1}{RTT} - \frac{1}{2} p x^2$
- TCP/Vegas: $\dot{x} = \frac{1}{RTT} (RTT_{min} + \frac{a}{x} - RTT)$

Equilibrium

- TCP/Reno: $X_{reno} = \frac{\alpha_{reno}}{RTT \sqrt{p}}$
- TCP/Vegas: $X_{vegas} = \frac{\alpha_{vegas}}{T_{queueing}}$

Recap: Interpreting Congestion Measure

$$p_f = \sum_{l \text{ uses } f} q_l$$

TCP/Reno: $\dot{x} = \frac{1}{RTT} - \frac{1}{2} p x^2 = \frac{1}{2} x^2 \left( \frac{2}{RTT^2 x^2} - p \right)$

TCP/Vegas: $\dot{x} = \frac{1}{RTT} (RTT_{min} + \frac{a}{x} - RTT) = \frac{1}{RTT} \left( \frac{a}{x} - T_{queueing} \right)$
Recap: Allocation Examples

<table>
<thead>
<tr>
<th>Objective</th>
<th>Allocation (x1, x2, x3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/Reno</td>
<td>0.26 0.74 0.74</td>
</tr>
<tr>
<td>TCP/Vegas</td>
<td>1/3 2/3 2/3</td>
</tr>
<tr>
<td>Max Throughput</td>
<td>0 1 1</td>
</tr>
<tr>
<td>Max-min</td>
<td>1/2 1/2 1/2</td>
</tr>
<tr>
<td>Max sum log(x)</td>
<td>1/3 2/3 2/3</td>
</tr>
<tr>
<td>Max sum of -1/(RTT^2 x)</td>
<td>0.26 0.74 0.74</td>
</tr>
</tbody>
</table>

Resource Allocation Frameworks

- **Forward (design) engineering:**
  - how to determine objective functions
  - given objective, how to design effective alg

\[
\begin{array}{ccc}
\text{max} & \sum_{f \in F} U_f(x_f) \\
\text{subject to} & \sum_{f \text{ link } l} x_f & \leq c_l \text{ for any link } l \\
& x & \geq 0
\end{array}
\]

- **Reverse (understand) engineering:**
  - understand current protocols (what are the objectives of TCP/Reno, TCP/Vegas?)
Network Bandwidth Allocation Using Nash Bargain Solution (NBS)

- **High level picture**
  - Given the feasible set of bandwidth allocation, we want to pick an allocation point that is efficient and fair.

- **The determination of the allocation point should be based on “first principles” (axioms)**

### Nash Bargain Solution (NBS)

- Assume a finite, convex feasible set in the first quadrant.

- **Axioms**
  - Pareto optimality
    - Impossibility of increasing the rate of one user without decreasing the rate of another.
  - Symmetry
    - A symmetric feasible set yields a symmetric outcome.
  - Invariance of linear transformation
    - The allocation must be invariant to linear transformations of users' rates.
  - Independence of irrelevant alternatives
    - Assume s is an allocation when feasible set is R, s ∈ T ⊆ R, then s is also an allocation when the feasible set is T.
Nash Bargain Solution (NBS)

- Surprising result by John Nash (1951)
  - the rate allocation point is the feasible point which maximizes
    \[ x_1 x_2 \cdots x_F \]
- This is equivalent to maximize
  \[ \sum_f \log(x_f) \]
- In other words, assume each flow has utility function \( \log(x_f) \)
- I will give a proof for \( F = 2 \)
  - think about \( F > 2 \)

Nash Bargain Solution

- Assume \( s \) is the feasible point which maximizes \( x_1 x_2 \)
- Scale the feasible set so that \( s \) is at \((1, 1)\)
- Question: after the transformation, is \( s \) still the point maximizing \( x_1 x_2 \)?:

...
Nash Bargain Solution

Question: after the transformation, is there any feasible point with $x_1 + x_2 > 2$?

Nash Bargain Solution

- Consider the symmetric rectangle $U$ containing the original feasible set.
- According to symmetry and Pareto, $s$ is the allocation when feasible set is $U$.
- According to independence of irrelevant alternatives, the allocation of $R$ is $s$ as well.
NBS ⇔ Proportional Fairness

- Allocation is proportionally fair if for any other allocation, aggregate of proportional changes is non-positive, e.g. if $x_f$ is a proportional-fair allocation, and $y_f$ is any other feasible allocation, then require

$$\sum_f \frac{y_f - x_f}{x_f} \leq 0$$

Summary: Allocation Schemes

- Max throughput
- Max-min
- Proportional fair
  - NBS
Roadmap: Resource Allocation Frameworks

- Forward (design) engineering:
  - how to determine objective functions
  - given objective, how to design effective alg

- Reverse (understand) engineering:
  - understand current protocols (what are the objectives of TCP/Reno, TCP/Vegas?)

A Two-Slide Summary of Constrained Convex Optimization Theory

\[
\begin{align*}
\text{max} & \quad \sum_{j=1}^{m} U_j(x_j) \\
\text{subject to} & \quad Ax \leq C \\
& \quad x \geq 0
\end{align*}
\]

- Map each \( x \) in \( S \), to \([g(x), f(x)]\)
- For each slope \( q \geq 0 \), computes \( f(x) - q g(x) \) of all mapped \([f(x), g(x)]\)

\[
D(q) = \max_{x\in S} (f(x) - qg(x))
\]
A Two-Slide Summary of Constrained Convex Optimization Theory

\[
\begin{align*}
\max & \quad f(x) \\
\text{subject to} & \quad g(x) \leq 0 \\
\text{over} & \quad x \in S
\end{align*}
\]

\[D(q) = \max_{x \in S} (f(x) - qg(x))\]

- \(D(q)\) is called the dual; \(q \geq 0\) are called prices in economics.
- \(D(q)\) provides an upper bound on obj.
- According to optimization theory: when \(D(q)\) achieves minimum over all \(q \geq 0\), then the optimization objective is achieved.

Primal

\[
\begin{align*}
\max & \quad \sum_{f \in F} U_f(x_f) \\
\text{subject to} & \quad \sum_{f : f \text{ uses link } l} x_f \leq c_l \text{ for any link } l \\
\text{over} & \quad x \geq 0
\end{align*}
\]

Q: Why is it hard to solve the primal problem?
### Dual of the Primal

\[
\begin{align*}
\text{max} & \quad \sum_{f \in F} U_f(x_f) \\
\text{subject to} & \quad \sum_{f \in F} x_f \leq c_l \text{ for any link } l \\
\text{over} & \quad x \geq 0
\end{align*}
\]

\[
D(q) = \max_{x_f \geq 0} \left( \sum_{f \in F} U_f(x_f) - \sum_{l:f \text{ uses } l} q_l \left( \sum_{f \in F} x_f - c_l \right) \right)
\]

### Decomposition

- Assume each link \( l \) has non-negative congestion signal \( q_l \), consider the dual \( D(q) \)

\[
D(q) = \max_{x_f \geq 0} \left( \sum_{f \in F} U_f(x_f) - \sum_{l:f \text{ uses } l} q_l \left( \sum_{f \in F} x_f - c_l \right) \right)
\]

\[
= \max_{x_f \geq 0} \sum_{f \in F} \left( U_f(x_f) - x_f \sum_{l:f \text{ uses } l} q_l \right) + \sum_{l} q_l c_l
\]

\[
= \sum_{f} \max_{x_f \geq 0} \left( U_f(x_f) - x_f \sum_{l:f \text{ uses } l} q_l \right) + \sum_{l} q_l c_l
\]
Distributed Optimization: User Problem

Given price signal per unit rate $p_f$ (=sum of $q_i$ along the path) flow $f$ chooses rate $x_f$ to maximize:

$$\max_{x_f} U_f(x_f) - x_f p_f$$
over $x_f \geq 0$

Using the price signals, the optimization problem of each user is decoupled: independent of each other!

How should flow $f$ adjust $x_f$ locally?

$$\Delta x_f \propto U'_f(x_f) - p_f$$

At equilibrium (i.e., at optimal), $x_f$ satisfies:

$$U'_f(x_f) - p_f = 0$$
Interpreting Congestion Measure

\[ p_f = \sum_{f \text{ uses } l} q_l \]

\[ \Delta x_f \propto U_f'(x_f) - p_f \]

Distributed Optimization: Network Problem

\[ D(q) = \sum_{x_f, x_l} \max_{x_f} \left( U_f(x_f) - x_f \sum_{l, f \text{ uses } l} q_l \right) + \sum q_i \]

The network (i.e., link l) adjusts the link signals \( q_l \) (assume after all flows have picked their optimal rates given congestion signal)

\[ \min_{q \geq 0} \tilde{D}(q) = \sum_{l} q_l (c_l - \sum_{f: l \text{ uses } l} x_f) \]
Distributed Optimization: Network Problem

\[ \min_{q \geq 0} D(q) = \sum_l q_l (c_l - \sum_{f: l \text{ uses } l} x_f) \]

how should link \( l \) adjust \( q_l \) locally?

\[ \Delta q_l \propto -\frac{\partial D(q)}{q_l} \]

\[ \frac{\partial}{\partial q_l} D(q) = c_l - \sum_{f: l \text{ uses } l} x_f \]

\[ \Delta q_l \propto \sum_{f: l \text{ uses } l} x_f - c_l \]

Decomposition

- **SYSTEM(U):**
  \[ \max \sum_{f \in F} U_f(x_f) \]
  \[ \text{subject to } \sum_{f: f \text{ uses } l} x_f \leq c_l \text{ for any link } l \]
  \[ \text{over } x \geq 0 \]

- **USER\( f \):**
  \[ \max_{x_f} U_f(x_f) - x_f p_f \]
  \[ \text{over } x_f \geq 0 \]

- **NETWORK:**
  \[ \min_{q \geq 0} \tilde{D}(q) = \sum_l q_l (c_l - \sum_{f: l \text{ uses } l} x_f) \]
Outline

❖ Admin and recap
❖ Bandwidth allocation framework
   ❖ framework
   ❖ Nash Bargaining Solution (NBS)
   ❖ distributed computation
   ❖ TCP/Reno, TCP/Vegas revisited

TCP/Reno Dynamics

\[ \Delta x_f \propto U'_f(x_f) - p_f \]

\[ \dot{x} = \frac{x^2}{2} \left( \frac{2}{RTT^2 x^2} - p \right) \]

\[ U'(x_f) - p_f \]

\[ \Rightarrow U_f(x_f) = \left( \frac{\sqrt{2}}{x_f RTT} \right)^2 \Rightarrow U_f(x_f) = -\frac{2}{RTT^2 x_f} \]
**TCP/Vegas Dynamics**

\[ \Delta x_f \propto U'_f(x_f) - p_f \]

\[ \dot{x}_f = \frac{x}{RTT^2} \left( \frac{\log(x_f)}{x_f} - (RTT - RTT_{\text{min}}) \right) \]

\[ U'_f(x_f) - p_f \]

\[ \Rightarrow U'_f(x_f) = \frac{\alpha}{x_f} \quad \Rightarrow U_f(x_f) = \alpha \log(x_f) \]

---

**Summary: TCP/Vegas and TCP/Reno**

- Pricing signal is queueing delay \( T_{\text{queueing}} \)
- Pricing signal is loss rate \( p \)

\[ x_f = \frac{\alpha}{T_{\text{queueing}}} \]

\[ x_f = \frac{\alpha}{RTT \sqrt{p}} \]

\[ U'_f(x_f) = T_{\text{queueing}} \]

\[ U'_f(x_f) = p \]

\[ \Rightarrow U'_f(x_f) = \frac{\alpha}{x_f} \]

\[ \Rightarrow U'_f(x_f) = \left( \frac{\alpha}{x_fRTT} \right)^2 \]

\[ \Rightarrow U_f(x_f) = \alpha \log(x_f) \]

\[ \Rightarrow U_f(x_f) = -\frac{\alpha'}{RTT^2 x_f} \]
Summary: Resource Allocation Frameworks

- **Forward (design) engineering:**
  - how to determine objective functions
  - given objective, how to design effective alg

- **Reverse (understand) engineering:**
  - understand current protocols (what are the objectives of TCP/Reno, TCP/Vegas?)

- **Additional pointers:**
  - [http://www.statslab.cam.ac.uk/~frank/pf/](http://www.statslab.cam.ac.uk/~frank/pf/)

Outline

- Admin and recap
- Bandwidth allocation framework
  - Network overview
Network Layer

- Transport packet from source to dest.
- Network layer in every host, router

Most basic functions:
- Control plane: path determination and call setup
  - determine routes taken by packets from sources to destinations
- Data plane: forwarding
  - move packets from router's input port to router output port

Internet Network Layer: Protocols

Network layer functions:

- **Routing protocols**
  - path selection
  - e.g., RIP, OSPF, BGP

- **Control protocols**
  - error reporting
  - e.g., ICMP

- **Control protocols**
  - router “signaling”
  - e.g., RSVP

- Network layer protocol (e.g., IP)
  - addressing conventions
  - packet format
  - packet handling conventions

Transport layer

Link layer

physical layer
Data Plane: Forwarding

Routing and call setup

Local forwarding table
- header value
- output link
  - 0100: 3
  - 0101: 2
  - 0111: 2
  - 1001: 1

Value in arriving packet’s header

Control Plane: Routing

Routing
Goal: determine “good” paths (sequences of routers) thru network from source to dest.

Graph abstraction for the routing problem:
- graph nodes are routers
- graph edges are physical links
  - links have properties: delay, capacity, $ cost, policy
Routing Design Space

Routing has a large design space

- who decides routing?
  - source routing: end hosts make decision
  - network routing: networks make decision

- how many paths from source s to destination d?
  - multi-path routing
  - single path routing

- does the route(s) provide QoS?
  - QoS
  - best effort

- will routing adapt to network traffic demand?
  - adaptive routing
  - static routing

Routing Design Space:
User-based, Multipath, Adaptive

Routing has a large design space

- who decides routing?
  - source routing: end hosts make decision
  - network routing: networks make decision

- how many paths from source s to destination d?
  - multi-path routing
  - single path routing

- does the route(s) provide QoS?
  - QoS
  - best effort

- will routing adapt to network traffic demand?
  - adaptive routing
  - static routing

...
**User Optimal, Multipath, Adaptive**

- User optimal: users pick the shortest routes (selfish routing)

![Diagram](image)

- Braess’s paradox

---

**Bound on Price of Anarchy**

For a network with linear latency functions

\[ \text{total latency of user (selfish) routing for given traffic demand} \leq 4/3 \]

\[ \text{total latency of network optimal routing for the traffic demand} \]
Bound on Price of Anarchy

- For any network with continuous, non-decreasing latency functions →

  total latency of user (selfish) routing for given traffic demand ≤

  total latency of network optimal routing for twice traffic demand

Routing Design Space:

Adaptive Routing

- Routing has a large design space
  - who decides routing?
    - source routing: end hosts make decision
    - network routing: networks make decision
  - how many paths from source s to destination d?
    - multi-path routing
    - single path routing
  - does the route(s) provide QoS?
    - QoS
    - best effort
  - will routing adapt to network traffic demand?
    - adaptive routing
    - static routing
Routing: Single-Path, Adaptive Routing

- Assume link costs reflect current traffic

Solution: Link costs are a combination of current traffic intensity (dynamic) and topology (static). To improve stability, the static topology part should be large. Thus less sensitive to traffic; thus non-adaptive.