Congestion Control for Internet Traffic

6.962 Presentation
May 17, 2001

Navid Sabbaghi

The different classes of Internet Traffic

- Elastic traffic (eg. telnet, ftp, dns, email,...) which is best-effort compatible and hence can handle delays and packet losses.
- Real-time traffic (eg. voice, video, ...) which are less tolerant to delay variations and packet losses.

But these types of traffic can grow and cause problems...

The congestion problem

- How does it create unfairness and lack of performance?
- How can we define the problem more precisely?
- How do we fix the problem?

The utility functions

- What are they functions of?
- What does it mean to reduce congestion? The concept of efficacy.
- What are two ways we can reduce congestion?
- How do rate-based and window-based congestion control fit into the picture?

Organization for this Presentation

- Background leading upto rate-based and window-based congestion control
- Rate-based congestion control vs Window-based congestion control
- TCP: a window-based approach
- A model of the network for comparative purposes
- Pricing: Kelly's rate-based congestion control algorithm
- Pricing: La's and Anantharam's rate-based congestion control algorithm
- Conclusions

Rate-based vs Window-based

- what are they?
- why window-based is considered better for controlling the stability of the network?
- TCP: a window based-approach

More about TCP

- Idea of packet conservation and efficiency of pipe
- Slow-start to reach equilibrium (time $R\log(W)$)
- Maintaining equilibrium – need for a round-trip time estimator
- TCP can be unfair (delay-bias) and high-packet loss
- Loss-based TCP vs Queue-based TCP

Network Model

- a set J of *resources*, C_j
- a *route* route $r \subset J$, set of routes R
- no routing– can be done but not necessary
- a $0-1$ matrix A (resource-route allocation matrix)

Rate-based control for elastic-traffic

- SYSTEM convex program:

$$SYSTEM(U, A, C):$$

$$max \sum_{r \in R} U_r(x_r)$$

subject to

$$Ax \leq C$$

over

$$x \geq 0.$$

- The break-up
- The NETWORK convex program encodes a notion of proportional fairness

The decomposition

$$USER_r(U_r; \lambda_r):$$

$$max U_r(\frac{p_r}{\lambda_r}) - p_r$$

over

$$p_r \geq 0.$$

$$NETWORK(A, C, p):$$

$$max \sum_{r \in R} p_r \log x_r$$

subject to

$$Ax \leq C$$

over

$$x \geq 0.$$

Fairness

- Max-min fairness - favors the small rates
- Proportionally fair rates

$$\sum_{r \in R} \frac{x_r^* - x_r}{x_r} \leq 0.$$

- Rates per unit charge are proportionally fair

$$\sum_{r \in R} p_r \frac{x_r^* - x_r}{x_r} \leq 0.$$

Dual: user picks rates, network picks price

- $$L(x,z;\mu) = \sum_{r\in R} p_r \log x_r + \mu^T (C - Ax - z)$$
  $$\frac{\partial L}{\partial x_r} = \frac{p_r}{x_r} - \sum_{j\in r} u_j$$

- $$DUAL(A,C;p):$$
  $$max \sum_{r\in R} p_r log(\sum_{j\in J} u_j) - \sum_{j\in J} u_j C_j$$
  over
  $$\mu \geq 0.$$

Decentralized algorithm

- primal:

$$\frac{d}{dt}x_r(t) = \kappa(p_r - x_r(t)\sum_{j\in r} u_j(t))$$

where

$$u_j(t) = p_j(\sum_{s:j\in s} x_s(t))$$

- note similarity to tcp
- dual:

$$\frac{d}{dt}u_j(t) = \kappa(\sum_{r:j\in r} x_r(t) - q_j(u_j(t))))$$

where

$$x_r(t) = \frac{p_r}{\sum_{k\in r} u_k(t)}$$

$\boxed{\textbf{Window-based congestion control}}$

- takes Walrand fair queue-based TCP and adds utillity
- fluid model:

$$Ax - C \leq 0$$

$$Q(Ax - C) = 0$$

$$X(A^T C' q + d) = w$$

rough overview of algorithm

- Walrand's work is used to implement NETWORK(A,C;p)
- $(p, 1)$-fairness vs (p, α)-fairness
- what about a pricing model? what do charge for? queuing delay
- USER convex program adjusts the window size

Conclusions

- tcp and how it is unfair
- two main pricing methodologies and how they promise fairness