Network Congestion Control

EL 933, Class10
Yong Liu
11/22/2005

Motivation

- Network bandwidth shared by all users
- Given routing, how to allocate bandwidth
  - efficiency
  - fairness
  - stability
- Challenges
  - distributed/selfish/uncooperative end systems
  - remote/dumb/oblivious routers

Outline

- Basics on TCP/RED

Congestion Control

- Network is a distributed resource sharing system
- Overload means low throughput
- Individual user depends on feedback from system to regulate its load

Fig. 2: A control system model of n users sharing a network.
Criteria for Congestion Control

- **Efficiency**: the closeness of the total usage to the optimal level.
- **Fairness**: the fairness index is defined as
  \[ F(x) = \frac{\sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n} x_i} \]
  and the **Max-Min Ratio** is
  \[ M = \min_{i,j} \left( \frac{x_i}{x_j} \right) \]
- **Convergence**: how fast does the system converge to the desired state.

AIMD

- **Additive Increase Multiplicative Decrease**
  \[ x_i(t) = \begin{cases} 
  a_i + x_i(t - 1) & \text{if no congestion}, \\
  b_D x_i(t - 1) & \text{if congestion}. 
  \end{cases} \]
- **AI**: aggressive in grabbing available bandwidth
- **MD**: conservative when network is congested
- **trade-off**
  - efficiency & fairness

Internet Congestion Control

- **Transport Control Protocol (TCP) at edge**
- **Active Queue Management (AQM) in core**

TCP Congestion Control

- **adapt sending rate to network congestion**
- **window-based rate control**
  - send out a window of W packets each round
  - increase W by 1 every RTT if no packet loss
TCP Congestion Control

- adapt sending rate to network congestion
- window based rate control
  - send out a window W of packets each round
  - increase W by 1 every RTT if no packet loss
  - decrease W by half upon packet loss

Queue Management

- natural solution: DropTail -- drop new packets if buffer full
  - synchronization among flows (?)
  - response too late
  - large buffer → large delay
- Active Queue Management (AQM)
  - random drop: break synchronization
  - proactively drop: earlier reaction
  - new approaches: RED, BLUE, PI, AVQ, REM ......
Optimal Rate Control

- **Demand v.s. supply**
  - demand: user rate $x_r, r \in R$
  - supply: link bandwidth $C_j, i \in J$
  - consumption: routing matrix $A_{jr} = 1$ if $j \in r$

- **Quantification**
  - utility function: $U_r(x_r)$
  - link cost: $C_j(y_j), y_j = \sum_{r: j \in r} x_r$

- **Optimal routing**
  - maximize aggregate utility under resource constraint
  - convex optimization solvable

$$\text{SYSTEM}(U, A, C) : \max_{r \in R} \sum U_r(x_r) \text{ subject to: } AX \leq C$$

Centralized v.s. Distributed

- **Centralized solution**
  - network determines rate for all users
  - require knowledge of routing and utility functions of all users

- **Distributed algorithm more realistic**
  - user determines its own sending rate
  - network uses "pricing" to regulate users
  - minimum info. exchange between users and network
  - right price $\rightarrow$ selfish optimum $=$ global optimum
  - analogy from Market Economics

Outline

- optimal rate control
- decomposition
- primal/dual algorithms
- user adaptation
- extensions
Decomposition: distributed optimization

- **User Optimization**
  - given a price, how much to send?

  \[
  \text{USER}_r(U_r; \lambda_r) : \quad \max_{w_r} U_r(\frac{w_r}{\lambda_r}) - w_r \\
  \text{subject to} \quad w_r \geq 0
  \]

- **Network Optimization**
  - given user payment, how to allocate?

  \[
  \text{NETWORK}(A, C; w) : \quad \max_{x \in \mathbb{R}^n} \sum_{r \in R} w_r \log x_r \\
  \text{subject to} \quad Ax \leq C, x \geq 0
  \]

Decomposition: efficiency & fairness

- **Network Opt.: primal-dual formulation**

  \[
  \text{primal problem: optimal rates for users} \quad \max_{x \in \mathbb{R}^n} \sum_{r \in R} w_r \log x_r \\
  \text{subject to} \quad Ax \leq C, x \geq 0
  \]

  \[
  \text{dual problem: optimal prices on links} \quad \max_{\mu \in \mathbb{R}^m} \sum_{r \in R} w_r \log(\sum_{j \in J} \mu_j) - \sum_{j \in J} \mu_j C_j \\
  \text{subject to} \quad \mu \geq 0
  \]

  duality theorem: dual solution matches primal solution

  \[
  x_r^* = \frac{w_r}{\sum_{j \in J} \mu_j^*}, \quad \forall r \in R
  \]
Network Opt.: primal algorithm

- link penalty function
  \[ C_j(y_j) = \int_0^{y_j} p_j(y)dy, \quad p_j(y) \to \infty \text{ as } y \to C_j \]

- unconstrained optimization
  \[ UC(x) : \max_x \sum_{r \in R} w_r \log x_r - \sum_{j \in J} C_j(\sum_{s : j \in S} x_s) \]

- dynamic algorithm: user rate adjustment
  \[ \frac{d}{dt} x_r(t) = k(w_r - x_r(t) \sum_{j \in J} p_j(\sum_{s : j \in S} x_s)) \]

- convergence
  - \( x(t) \to \bar{x}^* = \arg\max UC(x), \text{ as } t \to \infty \)
  - \( \bar{x}^* \) arbitrarily close to \( x^* \) of \( NETWORK(A,C;w) \)

Network Opt.: dual algorithm

- dynamic algorithm: link price adjustment
  \[ \frac{d}{dt} \mu_j(t) = k(\sum_{r : j \in R} x_r(t) - q_j(\mu_j(t))) \]
  \[ \text{where} \]
  \[ x_r(t) = \frac{w_r}{\sum_{k \in R} \mu_k(t)} \quad q_j(y) = p_j^{-1}(y) \]

- convergence
  - \( \mu(t) \to \bar{\mu}^*, \text{ as } t \to \infty \)
  - \( \bar{\mu}^* \) arbitrarily close to \( \mu^* \) of \( DUAL(A,C;w) \)

User Adaptation

- network optimization: \( NETWORK(A,C;w) \):
  - approximation:
    \[ \max_x \sum_{r \in R} w_r \log x_r - \sum_{j \in J} \int_0^{\bar{x}_j \in S} p_j(y)dy \]
  - user scheme:
    \[ \frac{d}{dt} x_r(t) = k(w_r - x_r(t) \sum_{j \in J} p_j(\sum_{s : j \in S} x_s)) \]

- system optimization: \( SYSTEM(U,A,C) \):
  - approximation:
    \[ \max_x \sum_{r \in R} U_r(x_r) - \sum_{j \in J} \int_0^{\bar{x}_j \in S} p_j(y)dy \]
  - user scheme:
    \[ \frac{d}{dt} x_r(t) = k(x_rU'_r(x_r) - x_r \sum_{j \in J} p_j(\sum_{s : j \in S} x_s)) \]

Other Properties

- speed of convergence to the optimum
- robustness against
  - information accuracy
  - information delay
- Utility function of TCP
  - reno: \( U_r^{reno}(x_r) = \frac{\sqrt{2}}{\tau_r} \tan^{-1}(\frac{x_r\tau_r}{2}) \)
  - vega: \( U_r^{vegas}(x_r) = \alpha_r d_r \log x_r \)
Extensions

- One user employs multiple paths
  - $0 - 1$ Matrix $H$: $H_{sr} = 1$ if user $s$ uses path $r$
  - $x_s = \sum_{r: H_{sr} = 1} y_r$

- Joint rate control and routing

\[
\text{SYSTEM}(U, H, A, C) : \\
\max \sum_{s \in S} U_s(x_s) - \sum_{j \in J} C_j(\sum_{r: j \in r} y_r) \\
\text{subject to: } Hy = x, \, x, y \geq 0
\]

- System converges to optimum
  - User only uses paths with minimum price
  - Traffic rates on selected paths equal

Summary

- Network optimal rate allocation decomposed into distributed optimization
  - Users maximize net gain
  - Links minimize cost
  - Dynamically approach optimum
    - Right link price
    - Optimal rate allocation
    - Maximized aggregate utility

- TCP Works!!
  - Maximize efficiency
  - Maintain some sort of fairness
  - Stable and robust

Overview

- Motivation
- Key idea
- Modeling details
- Validation with ns
- Analysis sheds insights into RED
- Conclusions
Motivation

- current simulation technology, e.g. ns
  - appropriate for small networks: 10s - 100s of network nodes, 100s - 1000s of IP flows
  - inflexible packet-level granularity
- current analysis techniques
  - UDP flows over small networks
  - TCP flows over single link

Challenge

Explore large scale systems
- 100s - 1000s network elements
- 10,000s - 100,000s of flows (TCP, UDP, NG)

Our Belief

Fluid-based techniques that abstract out protocol details are key to scalable network simulation/understanding

Contribution of Paper

First differential equation based fluid model to enable transient analysis of TCP/AQM networks developed

Key Idea

- model traffic as fluid
- describe behavior of flows and queues using Stochastic Differential Equations
- obtain Ordinary Differential Equations by taking expectations of SDEs
- solve resultant coupled ODEs numerically

Differential equation abstraction: computationally highly efficient

Loss Model

\[ \lambda(t) = B(t-\tau)p(t-\tau) \]

Round Trip Delay (\(\tau\))

Loss Rate as seen by Sender: \(\lambda(t) = B(t-\tau)p(t-\tau)\)
Start Simple: A Single Congested Router

- One bottlenecked AQM router
  - capacity \( C \) (packets/sec)
    - queue length \( q(t) \)
    - drop prob. \( p(t) \)
- \( N \) TCP flows
  - window sizes \( W_i(t) \)
  - round trip time \( R_i(t) = A_i + q(t)/C \)
  - throughputs \( B_i(t) = W_i(t)/R_i(t) \)

System of Differential Equations

All quantities are average values. Timeouts and slow start ignored

Window Size:
\[
\frac{dW_i}{dt} = \frac{1}{\hat{R}_i(q(t))} - \frac{\hat{W}_i(t-\tau)}{2} \frac{\hat{W}_i(t-\tau)}{\hat{R}_i(q(t-\tau))}
\]

Queue length:
\[
\frac{dq}{dt} = -1_{[q(t) > 0]} C + \sum \frac{\hat{W}_i(t)}{\hat{R}_i(q(t))}
\]

Average queue length:
\[
\frac{dx}{dt} = \frac{\ln(1-\alpha)}{\delta} \hat{x}(t) - \frac{\ln(1-\alpha)}{\delta} \hat{q}(t)
\]

Where
- \( \alpha \) = averaging parameter of RED \( w_q \)
- \( \delta \) = sampling interval \( \sim 1/C \)

Loss probability:
\[
\frac{dp}{dt} = \frac{dp}{dx} \frac{dx}{dt}
\]

Where \( dp \) is obtained from the marking profile

Stepping back: Where are we?

\( W=\)Window size, \( R=\)RTT, \( q=\)queue length, \( p=\)marking probability

\[
\frac{dW_i}{dt} = f_1(p,R_i) \quad i = 1..N
\]

\[
\frac{dp}{dt} = f_3(q) \quad \frac{dq}{dt} = f_2(W_i)
\]

\( N+2 \) coupled equations solved numerically using MATLAB
Extension to Network

Networked case: $N$ AQM routers

- queuing delay = aggregate delay
  $$q(t) = \sum_N q_N(t)$$
- loss probability = cumulative loss probability
  $$p(t) = 1 - \Pi_N (1 - p_N(t))$$

Other extensions to the model

- Timeouts: Leveraged work done in [PFTK Sigcomm98] to model timeouts
- Aggregation of flows: Represent flows sharing the same route by a single equation

Timeouts: Leveraged work done in [PFTK Sigcomm98] to model timeouts
Aggregation of flows: Represent flows sharing the same route by a single equation

Performance of SDE method

- queue capacities 5 Mb/s
- load variation at $t=75$ and $t=150$ seconds
- 200 flows simulated
- DE solver captures transient performance
- time taken for DE solver ~ 5 seconds on P450

Observations on RED

- “Tuning” of RED is difficult
  - sensitive to packet sizes, load levels, round trip delay, etc.
- discontinuity of drop function contributes to, but is not the only reason for oscillations.
- RED uses variable sampling interval $\delta$. This could cause oscillations.

Further Work

Conclusions

- differential equation based model for evaluating TCP/AQM networks
- computation cost of DE method a fraction of the discrete event simulation cost
- formal representation and analysis yields better understanding of RED/AQM

Future Directions

- model short lived and non-responsive flows
- demonstrate applicability to large networks
  "Fluid Models and Solutions for Large-Scale IP Networks", (Sigmetrics 2003)
- analyze theoretical model to rectify RED shortcomings
  "On Designing Improved Controllers for AQM Routers Supporting TCP Flows", (INFOCOM 2001)
- apply techniques to
  - other protocols, e.g. TCP-friendly protocols
  - other AQM mechanisms like Diffserv

New Directions...

- high speed networks
  - linear increase not efficient enough: HTCP, FAST
  - more feedback from routers: XCP
- wireless networks
  - corruption loss: Explicit Congestion Notification
  - link coupling: cross layer design
- overlay/p2p networks
  - application layer control: parallel/relay
  - revisit: efficiency/fairness/stability