Dualtiy Model of TCP/AQM

Steven Low

CS & EE, Caltech
netlab.caltech.edu

2002
Acknowledgments

- S. Athuraliya, D. Lapsley, V. Li, Q. Yin (UMelb)
- S. Adlakha (UCLA), D. Choe (Postech/Caltech), J. Doyle (Caltech), K. Kim (SNU/Caltech), L. Li (Caltech), F. Paganini (UCLA), J. Wang (Caltech)
- L. Peterson, L. Wang (Princeton)
Network protocols.

HTTP

TCP

IP

Routers

Files

packets

from J. Doyle
Web traffic

Is streamed out on the net.

Web servers

Creating internet traffic

from J. Doyle
Congestion Control

Heavy tail → Mice-elephants

efficient & fair sharing

small delay

queueing + propagation

Congestion control

CDN
TCP

\[ x_i(t) \]
Example congestion measure $p_l(t)$

- Loss (Reno)
- Queueing delay (Vegas)
Protocol (Reno, Vegas, RED, REM/PI...)

\[
\begin{align*}
    x(t+1) &= F(p(t), x(t)) \\
    p(t+1) &= G(p(t), x(t))
\end{align*}
\]

**Equilibrium**
- Performance
  - Throughput, loss, delay
- Fairness
- Utility

**Dynamics**
- Local stability
- Cost of stabilization
Duality theory $\rightarrow$ equilibrium

- Source rates $x_i(t)$ are primal variables
- Congestion measures $p_i(t)$ are dual variables
- Flow control is optimization process
TCP & AQM

Control theory → stability

- Internet as a feedback system
- Distributed & delayed
Outline

- TCP/AQM
  - Reno/RED
- Equilibrium
  - Duality model
- Stability & optimal control
  - Linearized model
- A scalable control
TCP & AQM

TCP:
- Reno
- Vegas

AQM:
- DropTail
- RED
- REM/PI
- AVQ
TCP & AQM

TCP:
- Reno
- Vegas

AQM:
- DropTail
- RED
- REM/PI
- AVQ
TCP/AQM

- Tahoe (Jacobson 1988)
  - Slow Start
  - Congestion Avoidance
  - Fast Retransmit
- Reno (Jacobson 1990)
  - Fast Recovery
- Vegas (Brakmo & Peterson 1994)
  - New Congestion Avoidance
- RED (Floyd & Jacobson 1993)
- REM/PI (Athuraliya et al 2000, Hollot et al 2001)
- AVQ (Kunniyur & Srikant 2001)
Model structure

Multi-link multi-source network

\[ F_1 \xrightarrow{x} R_f(s) \xrightarrow{y} G_1 \]

\[ F_N \xrightarrow{q} R_b'(s) \xrightarrow{p} G_L \]

TCP  Network  AQM

from F. Paganini
TCP Reno (Jacobson 1990)

- **SS**: Slow Start
- **CA**: Congestion Avoidance

- Fast retransmission/fast recovery
TCP Vegas (Brakmo & Peterson 1994)

- Converges, no retransmission
- ... provided buffer is large enough
Vegas model

for every RTT

\{ 
\text{if } \frac{W}{RTT_{\text{min}}} - \frac{W}{RTT} < \alpha \text{ then } W ++ \\
\text{if } \frac{W}{RTT_{\text{min}}} - \frac{W}{RTT} > \alpha \text{ then } W -- \\
\}

for every loss

\[ W := \frac{W}{2} \]

\[ F_i: \quad x_i(t+1) = \begin{cases} 
\frac{1}{D_i} & \text{if } x_i(t)q_i(t) < \alpha_id_i \\
\end{cases} \]

\[ x_i(t+1) = \begin{cases} 
\frac{1}{D_i} & \text{if } x_i(t)q_i(t) > \alpha_id_i \\
\end{cases} \]

\[ x_s(t+1) = x_s(t) \quad \text{else} \]

\[ G_l: \quad p_l(t+1) = \left[ p_l(t) + \frac{y_l(t)}{c_l} - 1 \right]^+ \]
Vegas model

\[ F_i = x_i(t) + \frac{1}{(d_i + q_i(t))^2} \text{sgn}(\alpha_i d_i - x_i(t)q_i(t)) \]

\[ G_l = \left( p_l(t) + \frac{y_l(t)}{c_i} - 1 \right)^+ \]
Overview

Protocol (Reno, Vegas, RED, REM/PI...)

\[ x(t+1) = F(p(t), x(t)) \]
\[ p(t+1) = G(p(t), x(t)) \]

**Equilibrium**
- Performance
  - Throughput, loss, delay
- Fairness
- Utility

**Dynamics**
- Local stability
- Cost of stabilization
Outline

- TCP/AQM
  - Reno/RED
- Equilibrium
  - Duality model
- Stability
  - Linearized model
- A scalable control
Flow control

- Interaction of source rates $x_s(t)$ and congestion measures $p_l(t)$
- Duality theory
  - They are primal and dual variables
  - Flow control is optimization process
- Example congestion measure
  - Loss (Reno)
  - Queueing delay (Vegas)
Model

- Network
  - Links $l$ of capacities $c_l$

- Sources $i$
  - $L(s)$ - links used by source $i$
  - $U_i(x_i)$ - utility if source rate = $x_i$

\[
x_1 + x_2 \leq c_1 \\
x_1 + x_3 \leq c_2
\]
Primal problem

\[
\max_{x_i \geq 0} \quad \sum_i U_i(x_i)
\]
subject to \[ y_l \leq c_l, \quad \forall l \in L \]

Assumptions
- Strictly concave increasing \( U_i \)
- Unique optimal rates \( x_i \) exist
- Direct solution impractical
Related Work

- Formulation
  - Kelly 1997
- Penalty function approach
  - Kelly, Maulloo & Tan 1998
  - Kunniyur & Srikant 2000
- Duality approach
  - Low & Lapsley 1999
  - Athuraliya & Low 2000
- Extensions
  - Mo & Walrand 2000
  - La & Anantharam 2000
- Dynamics
  - Johari & Tan 2000, Massoulie 2000, Vinnicombe 2000, ...
  - Hollot, Misra, Towsley & Gong 2001
  - Paganini 2000, Paganini, Doyle, Low 2001, ...
Duality Approach

Primal: \[ \max_{x_s \geq 0} \sum_s U_s(x_s) \quad \text{subject to} \quad x^l \leq c_l, \quad \forall l \in L \]

Dual: \[ \min_{p \geq 0} D(p) = \left( \max_{x_s \geq 0} \sum_s U_s(x_s) + \sum_l p_l (c_l - x^l) \right) \]

Primal-dual algorithm:

\[ x(t+1) = F(p(t), x(t)) \]

\[ p(t+1) = G(p(t), x(t)) \]
Duality Model of TCP

- Source algorithm iterates on rates
- Link algorithm iterates on prices
- With different utility functions

Primal-dual algorithm:

\[x(t+1) = F(p(t), x(t))\]  \(\quad\) Reno, Vegas
\[p(t+1) = G(p(t), x(t))\]  \(\quad\) DropTail, RED, REM
Duality Model of TCP

\((x^*, p^*)\) primal-dual optimal if and only if

\[ y_i^* \leq c_i \quad \text{with equality if } p_l^* > 0 \]

Primal-dual algorithm:

\[ x(t+1) = F(p(t), x(t)) \quad \text{(Reno, Vegas)} \]
\[ p(t+1) = G(p(t), x(t)) \quad \text{(DropTail, RED, REM)} \]
Duality Model of TCP

Any link algorithm that stabilizes queue
- generates Lagrange multipliers
- solves dual problem

Primal-dual algorithm:

\[
\begin{align*}
x(t+1) &= F(p(t), x(t)) \\
p(t+1) &= G(p(t), x(t))
\end{align*}
\]

\begin{itemize}
  \item Reno, Vegas
  \item DropTail, RED, REM
\end{itemize}
Gradient algorithm

- Gradient algorithm

  source: \( x_i(t+1) = U_i^{-1}(q_i(t)) \)

  link: \( p_i(t+1) = [p_i(t) + \gamma_i(y_i(t) - c_i)]^+ \)

**Theorem** (Low & Lapsley ’99)

Converge to optimal rates in distributed asynchronous environment
Gradient algorithm

- Gradient algorithm

  source: \[ x_i(t+1) = U_i^{t-1}(q_i(t)) \]
  link: \[ p_l(t+1) = [p_l(t) + \gamma_l(y_l(t) - c_l)]^+ \]

- Vegas: approximate gradient algorithm

\[
F_i = x_i(t) + \frac{1}{(d_i + q_i(t))^2} \text{ sgn}\left(\bar{x}_i(t) - x_i(t)\right)
\]

\[ U_i^{t-1}(q_i(t)) \]
Summary: equilibrium

- **Flow control problem**
  \[
  \max_{x_s \geq 0} \sum_s U_s(x_s)
  \]
  subject to \(x^l \leq c_l, \quad \forall l \in L\)

- **Primal-dual algorithm**
  \[
  x(t+1) = F(p(t), x(t))
  \]
  \[
  p(t+1) = G(p(t), x(t))
  \]

- **TCP/AQM**
  - Maximize aggregate source utility
  - With **different** utility functions

---

Redo: summary

- **Flow control problem**
  \[
  \max_{x_s \geq 0} \sum_s U_s(x_s)
  \]
  subject to \(x^l \leq c_l, \quad \forall l \in L\)

- **Primal-dual algorithm**
  \[
  x(t+1) = F(p(t), x(t))
  \]
  \[
  p(t+1) = G(p(t), x(t))
  \]

- **TCP/AQM**
  - Maximize aggregate source utility
  - With **different** utility functions
Implications

- Performance
  - Rate, delay, queue, loss
- Fairness
  - Utility function
- Persistent congestion
Performance

- Delay
  - Congestion measures: end to end *queueing* delay
  - Sets rate \( x_s(t) = \alpha_s \frac{d_s}{q_s(t)} \)
  - Equilibrium condition: Little’s Law

- Loss
  - No loss if converge (with sufficient buffer)
  - Otherwise: revert to Reno (loss unavoidable)
Vegas Utility

Equilibrium \((x, p) = (F, G)\)

\[ U_i(x_i) = \alpha_i d_i \log x_i \]

Proportional fairness
## Validation  
(L. Wang, Princeton)

<table>
<thead>
<tr>
<th></th>
<th>Source 1</th>
<th>Source 3</th>
<th>Source 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTT (ms)</td>
<td>17.1 (17)</td>
<td>21.9 (22)</td>
<td>41.9 (42)</td>
</tr>
<tr>
<td>Rate (pkts/s)</td>
<td>1205 (1200)</td>
<td>1228 (1200)</td>
<td>1161 (1200)</td>
</tr>
<tr>
<td>Window (pkts)</td>
<td>20.5 (20.4)</td>
<td>27 (26.4)</td>
<td>49.8 (50.4)</td>
</tr>
<tr>
<td>Avg backlog (pkts)</td>
<td>9.8 (10)</td>
<td>27 (26.4)</td>
<td>49.8 (50.4)</td>
</tr>
</tbody>
</table>

- NS-2 simulation, single link, capacity = 6 pkts/ms
- 5 sources with different propagation delays, $\alpha_s = 2$ pkts/RTT
Persistent congestion

- Vegas exploits buffer process to compute prices (queueing delays)
- Persistent congestion due to
  - Coupling of buffer & price
  - Error in propagation delay estimation
- Consequences
  - Excessive backlog
  - Unfairness to older sources

**Theorem** (Low, Peterson, Wang ’02)

A relative error of $\varepsilon_s$ in propagation delay estimation distorts the utility function to

$$\hat{U}_s(x_s) = (1 + \varepsilon_s) \alpha_s d_s \log x_s + \varepsilon_s d_s x_s$$
Validation (L. Wang, Princeton)

- Single link, capacity = 6 pkt/ms, $\alpha_s = 2$ pkts/ms, $d_s = 10$ ms
- With finite buffer: Vegas reverts to Reno

Without estimation error

With estimation error

Buffer Usage at Router 1 (alpha 2pkts/ms)

buffer occupancy

0 10 20 30 40 50 60 70 80 90 100

0 50 100 150 200 250 300 350 400 450 500

time (second)
## Validation

(L. Wang, Princeton)

<table>
<thead>
<tr>
<th>#</th>
<th>src1</th>
<th>src2</th>
<th>src3</th>
<th>src4</th>
<th>src5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.98 (6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.05 (2)</td>
<td>3.92 (4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.96 (0.94)</td>
<td>1.46 (1.49)</td>
<td>3.54 (3.57)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.51 (0.50)</td>
<td>0.72 (0.73)</td>
<td>1.34 (1.35)</td>
<td>3.38 (3.39)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.29 (0.29)</td>
<td>0.40 (0.40)</td>
<td>0.68 (0.67)</td>
<td>1.30 (1.30)</td>
<td>3.28 (3.34)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>queue (pkts)</th>
<th>baseRTT (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19.8 (20)</td>
<td>10.18 (10.18)</td>
</tr>
<tr>
<td>2</td>
<td>59.0 (60)</td>
<td>13.36 (13.51)</td>
</tr>
<tr>
<td>3</td>
<td>127.3 (127)</td>
<td>20.17 (20.28)</td>
</tr>
<tr>
<td>4</td>
<td>237.5 (238)</td>
<td>31.50 (31.50)</td>
</tr>
<tr>
<td>5</td>
<td>416.3 (416)</td>
<td>49.86 (49.80)</td>
</tr>
</tbody>
</table>
Outline

Protocol (Reno, Vegas, RED, REM/PI...)

\[
x(t+1) = F(p(t), x(t)) \]
\[
p(t+1) = G(p(t), x(t))
\]

Equilibrium
- Performance
  - Throughput, loss, delay
- Fairness
- Utility

Dynamics
- Local stability
- Cost of stabilization
Vegas model

\[ F_i = x_i(t) + \frac{1}{(d_i + q_i(t))^2} \text{sgn}(\alpha_i d_i - x_i(t)q_i(t)) \]

\[ G_l = \left( p_l(t) + \frac{y_l(t)}{c_l} - 1 \right)^+ \]
Vegas model

\[ [R_f]_{li} = e^{-s\tilde{\tau}_{li}} \quad \text{if source } i \text{ uses link } l \]

\[ [R_b]_{li} = e^{-s\tilde{\tau}_{li}} \quad \text{if source } i \text{ uses link } l \]
Approximate model

\[ \dot{x}_i(t) = \frac{1}{T(t)^2} \text{sgn}\left(1 - \frac{x_i(t)q_i(t)}{\alpha_i d_i}\right) \]
Approximate model

\[
\dot{x}_i(t) = \frac{1}{T(t)^2} \, \text{sgn} \left( 1 - \frac{x_i(t)q_i(t)}{\alpha_i d_i} \right)
\]

\[
\dot{x}_i(t) = \frac{2}{\pi} \frac{1}{T(t)^2} \, \tan^{-1} \eta \left( 1 - \frac{x_i(t)q_i(t)}{\alpha_i d_i} \right)
\]
Linearized model

\[ \dot{x}_i = - \frac{x_i}{q_i s T_i + a_i} \partial q_i \]

\[ a_i = - \frac{2\eta}{\pi} \frac{1}{x_i T_i} \]

\[ \partial p_l = - \frac{\gamma}{c_l} \partial y_l \]

\( \gamma \) controls equilibrium delay
Stability

**Theorem** (Choe & Low, '02)

Locally asymptotically stable if

\[
\frac{\text{link queueing delay}}{\text{round trip time}} > \frac{a_i \sin \omega_c}{\min a_i \omega_c} > 0.63
\]

Cannot be satisfied with >1 bottleneck link!
\[
\dot{x}_i(t) = \frac{2}{\pi} \frac{1}{T(t)^2} \tan^{-1}(1 - \frac{x_i(t)q_i(t)}{\alpha_i d_i})
\]

\[
\dot{x}_i(t) = \frac{2}{\pi} \frac{1}{T(t)^2} \tan^{-1}(1 - \frac{x_i(t)q_i(t)}{\alpha_i d_i} - \kappa_i \dot{q}_i(t))
\]
\[ \dot{x}_i = -\frac{x_i}{q_i} \frac{a_i}{sT_i + a_i} \partial q_i \]

\[ \partial \dot{p}_l = -\frac{\gamma}{c_l} \partial y_l \]

\( \gamma \) controls equilibrium delay
Linearized model

\[ \partial x_i = -b_i \frac{s + a_i}{s + \mu_i a_i} \partial q_i \]

choose \( a_i = a, \alpha_i = \mu \)

\[ \partial \dot{p}_l = -\frac{\gamma}{c_l} \partial y_l \]

\( \gamma \) controls equilibrium delay
Stability

**Theorem** (Choe & Low, '02)

Locally asymptotically stable if

\[ M \frac{\text{round trip time}}{\text{round trip queueing}} < \sigma(a, \mu) \]

**Example**

- \( LHS < 10 \times 10 = 100 \)
- \( a = 0.1, \mu = 0.015 \Rightarrow \sigma(a, \mu) = 120 \)
Stability

**Theorem** (Choe & Low, '02)

Locally asymptotically stable if

\[ M \frac{\text{round trip time}}{\text{round trip queueing}} < \sigma(a, \mu) \]

**Application**

- Stabilized TCP with current routers
- Queueing delay as congestion measure has the right scaling
- Incremental deployment with ECN
Vertical decomposition

Utility maximization

Primal: \[ \max_R \max_{x_i \geq 0} \sum_i U_i(x_i) \]
subject to \[ y_l \leq c_l, \quad \forall l \in L \]

Dual: \[ \min_{p \geq 0} \sum_i \max_{x_i \geq 0} \left( U_i(x_i) - x_i \min_{R_i} \sum_l R_{li} p_l \right) + \sum_l p_l c_l \]

Shortest path routing!
Vertical decomposition

Utility maximization

Primal: \( \max_R \max_{x_i \geq 0} \sum_i U_i(x_i) \)
subject to \( y_l \leq c_l, \quad \forall l \in L \)

Dual: \( \min_{p \geq 0} \sum_i \max_{x_i \geq 0} \left( U_i(x_i) - x_i \min_{R_i} \sum R_{li} p_l \right) + \sum l p_l c_l \)

Can shortest-path routing (IP) and TCP/AQM maximize utility?
Vertical decomposition

**Theorem** (Wang, Li, Low, Doyle ’02)

Primal problem is NP-hard

- Cannot be solved by shortest-path routing and TCP/AQM
- Shortest path routing based on prices can be unstable
- Even when stable, there can be duality gap
- How well does TCP/AQM/IP solve it approximately?
Papers

- A duality model of TCP flow controls (ITC, Sept 2000)
- Optimization flow control, I: basic algorithm & convergence (ToN, 7(6), Dec 1999)
- Understanding Vegas: a duality model (J. ACM, 2002)
- Scalable laws for stable network congestion control (CDC, 2001)
- REM: active queue management (Network, May/June 2001)

netlab.caltech.edu