Optimization models for congestion control with multipath routing in TCP/IP networks

Roberto Cominetti
Cristóbal Guzmán

Departamento de Ingeniería Industrial
Universidad de Chile

Workshop on Optimization, Games and Dynamics
28-29 Novembre 2011, Institut Henri Poincaré, Paris
Overview

1. TCP/IP communication protocols
2. Congestion control and network utility maximization
3. Congestion control with Markovian multipath routing
TCP/IP – Single path routing

- Communication network $G = (N, A)$
- Each source $s \in S$ transmits packets from origin $o_s$ to destination $d_s$
- At which rate? Along which route?
Congestion measures: link delays / packet loss

Switch/Router

- Links have random delays \( \tilde{\lambda}_a = \lambda_a + \epsilon_a \) with \( E(\epsilon_a) = 0 \)

\[
\tilde{\lambda}_a = \text{Queuing} + \text{Transmission} + \text{Propagation}
\]

- Finite queuing buffers \( \Rightarrow \) packet loss probability \( p_a \)
TCP/IP – Current protocols

- **Route selection** (RIP/OSPF/IGRP/BGP/EGP)
  Dynamic adjustment of routing tables
  Slow timescale evolution (15-30 seconds)
  Network Layer 3

- **Rate control** (TCP Reno/Tahoe/Vegas)
  Dynamic adjustment of source rates – congestion window
  Fast timescale evolution (100-300 milliseconds)
  Transport Layer 4
TCP – Congestion window control

Packets →

TCP

ARE YOU GETTING ALL OF THIS?

NO, SLOW DOWN SO I CAN SAVE ACCURATELY!

← Acks
TCP – Congestion window control

\[ x_s = \text{source rate} \sim \frac{\text{congestion window}}{\text{round-trip time}} = \frac{W_s}{\tau_s} \]
TCP – Congestion control

Sources adjust transmission rates in response to congestion

Basic principle: higher congestion ⇔ smaller rates

\[ x_s : \text{source transmission rate [packets/sec]} \]
\[ \lambda_a : \text{link congestion measure (loss pbb, queuing delay)} \]

\[ y_a = \sum_{s \ni a} x_s \quad \text{(aggregate link loads)} \]
\[ q_s = \sum_{a \in s} \lambda_a \quad \text{(end-to-end congestion)} \]

Decentralized algorithms

\[ x_s^{t+1} = F_s(x_s^t, q_s^t) \quad \text{(TCP – source dynamics)} \]
\[ \lambda_a^{t+1} = G_a(\lambda_a^t, y_a^t) \quad \text{(AQM – link dynamics)} \]
Sources adjust transmission rates in response to congestion

Basic principle: higher congestion $\iff$ smaller rates

$\begin{align*}
    x_s & \quad : \quad \text{source transmission rate [packets/sec]} \\
    \lambda_a & \quad : \quad \text{link congestion measure (loss pbb, queuing delay)} \\
    y_a & = \sum_{s \ni a} x_s \quad \text{(aggregate link loads)} \\
    q_s & = \sum_{a \in s} \lambda_a \quad \text{(end-to-end congestion)}
\end{align*}$

Decentralized algorithms

$\begin{align*}
    x_s^{t+1} & = F_s(x_s^t, q_s^t) \quad \text{(TCP – source dynamics)} \\
    \lambda_a^{t+1} & = G_a(\lambda_a^t, y_a^t) \quad \text{(AQM – link dynamics)}
\end{align*}$
TCP – Congestion control

Sources adjust transmission rates in response to congestion

Basic principle: higher congestion ⇔ smaller rates

- $x_s$: source transmission rate [packets/sec]
- $\lambda_a$: link congestion measure (loss pbb, queuing delay)

$$y_a = \sum_{s \ni a} x_s$$ (aggregate link loads)
$$q_s = \sum_{a \in s} \lambda_a$$ (end-to-end congestion)

Decentralized algorithms

$$x_s^{t+1} = F_s(x_s^t, q_s^t)$$ (TCP – source dynamics)
$$\lambda_a^{t+1} = G_a(\lambda_a^t, y_a^t)$$ (AQM – link dynamics)
Example: TCP-Reno / packet loss probability

**AIMD control**

\[
W_{s}^{t+\tau_s} = \begin{cases} 
W_{s}^{t} + 1 & \text{if } W_{s}^{t} \text{ packets are successfully transmitted} \\
\lceil W_{s}^{t}/2 \rceil & \text{one or more packets are lost (duplicate ack's)}
\end{cases}
\]

\[
\pi^{t}_{s} = \Pi_{a\in s}(1-p^{t}_{a}) = \text{success probability (per packet)}
\]

**Additive congestion measure**

\[
q_{s}^{t} \triangleq -\ln(\pi^{t}_{s}) \\
\lambda^{t}_{a} \triangleq -\ln(1-p^{t}_{a})
\]

\[
\Rightarrow q_{s}^{t} = \sum_{a\in s} \lambda^{t}_{a}
\]

**Approximate model for rate dynamics**

\[
\mathbb{E}(W_{s}^{t+\tau_s}|W_{s}^{t}) \sim e^{-q_{s}^{t}W_{s}^{t}}(W_{s}^{t} + 1) + (1 - e^{-q_{s}^{t}W_{s}^{t}})\lceil W_{s}^{t}/2 \rceil
\]

\[
\Rightarrow x_{s}^{t+1} = x_{s}^{t} + \frac{1}{2\tau_s} \left[ e^{-\tau_s q_{s}^{t}x_{s}^{t}}(x_{s}^{t} + \frac{2}{\tau_s}) - x_{s}^{t} \right]
\]
Example: TCP-Reno / packet loss probability

**AIMD control**

\[ W_{s}^{t+\tau_s} = \begin{cases} W_s^t + 1 & \text{if } W_s^t \text{ packets are successfully transmitted} \\ \lceil W_s^t / 2 \rceil & \text{one or more packets are lost (duplicate ack's)} \end{cases} \]

\[ \pi_s^t = \prod_{a\in_s} (1 - p_a^t) = \text{success probability (per packet)} \]

**Additive congestion measure**

\[
\begin{align*}
q_s^t &\triangleq -\ln(\pi_s^t) \\
\lambda_a^t &\triangleq -\ln(1 - p_a^t)
\end{align*}
\]

\[ q_s^t = \sum_{a\in_s} \lambda_a^t \]

**Approximate model for rate dynamics**

\[ E(W_{s}^{t+\tau_s} | W_s^t) \sim e^{-q_s^t W_s^t} (W_s^t + 1) + (1 - e^{-q_s^t W_s^t}) \lceil W_s^t / 2 \rceil \]

\[ \Rightarrow x_s^{t+1} = x_s^t + \frac{1}{2\tau_s} \left[ e^{-\tau_s q_s^t} x_s^t (x_s^t + \frac{2}{\tau_s}) - x_s^t \right] \]
**Example: TCP-Reno / packet loss probability**

**AIMD control**

\[ W_{s}^{t+\tau_{s}} = \begin{cases} W_{s}^{t} + 1 & \text{if } W_{s}^{t} \text{ packets are successfully transmitted} \\ \lceil W_{s}^{t}/2 \rceil & \text{one or more packets are lost (duplicate ack's)} \end{cases} \]

\[ \pi_{s}^{t} = \prod_{a \in s}(1 - p_{a}^{t}) = \text{success probability (per packet)} \]

**Additive congestion measure**

\[ q_{s}^{t} \triangleq -\ln(\pi_{s}^{t}) \]
\[ \lambda_{a}^{t} \triangleq -\ln(1 - p_{a}^{t}) \]

\[ \Rightarrow q_{s}^{t} = \sum_{a \in s} \lambda_{a}^{t} \]

**Approximate model for rate dynamics**

\[ \mathbb{E}(W_{s}^{t+\tau_{s}} | W_{s}^{t}) \sim e^{-q_{s}^{t}W_{s}^{t}} (W_{s}^{t} + 1) + (1 - e^{-q_{s}^{t}W_{s}^{t}}) \lceil W_{s}^{t}/2 \rceil \]

\[ \Rightarrow x_{s}^{t+1} = x_{s}^{t} + \frac{1}{2\tau_{s}} \left[ e^{-\tau_{s}q_{s}^{t}}x_{s}^{t} (x_{s}^{t} + \frac{2}{\tau_{s}}) - x_{s}^{t} \right] \]
Example: AQM / Droptail → RED-REM

Marking probability on links controlled by AQM

\[ p_a^t = \varphi_a(r_a^t) \]

as a function of the link’s average queue length

\[ r_a^{t+1} = (1 - \alpha)r_a^t + \alpha L_a^t \]
Network Utility Maximization

- Low, Srikant, etc. (1999-2002) showed that current TCP/AQM control algorithms solve an implicit network optimization problem.
- During last decade, the model has been used and extended to study the performance of wired and wireless networks.
Steady state equations

\[ x_{s(t+1)} = F_s(x_s, q_s) \] (TCP – source dynamics)
\[ \lambda_{a(t+1)} = G_a(\lambda_a, y_a) \] (AQM – link dynamics)
Steady state equations

\[
\begin{align*}
x_s &= F_s(x_s, q_s) \quad \text{(TCP – source equilibrium)} \\
\lambda_a &= G_a(\lambda_a, y_a) \quad \text{(AQM – link equilibrium)}
\end{align*}
\]
Steady state equations

\[
\begin{align*}
x_s &= F_s(x_s, q_s) & \text{(TCP – source equilibrium)} \\
\lambda_a &= G_a(\lambda_a, y_a) & \text{(AQM – link equilibrium)}
\end{align*}
\]

\[
\begin{align*}
x_s &= f_s(q_s) & \text{(decreasing)} \\
\lambda_a &= \psi_a(y_a) & \text{(increasing)}
\end{align*}
\]

\[
\begin{align*}
q_s &= \sum_{a \in s} \lambda_a \\
y_a &= \sum_{s \ni a} x_s
\end{align*}
\]
Steady state equations

\[ \begin{align*}
    x_s &= F_s(x_s, q_s) \quad \text{(TCP – source equilibrium)} \\
    \lambda_a &= G_a(\lambda_a, y_a) \quad \text{(AQM – link equilibrium)}
\end{align*} \]

\[ x_s = f_s(q_s) \quad \text{(decreasing)} \]
\[ \lambda_a = \psi_a(y_a) \quad \text{(increasing)} \]
\[ q_s = \sum_{a \in s} \lambda_a \\
    y_a = \sum_{s \ni a} x_s \]

\[ x_s = f_s(\sum_{a \in s} \lambda_a) \\
    \lambda_a = \psi_a(\sum_{s \ni a} x_s) \]
Examples

**TCP-Reno** (loss probability)

\[ q_s = f_s^{-1}(x_s) \triangleq \frac{1}{\tau_s x_s} \ln(1 + \frac{2}{\tau_s x_s}) \]

\[ \lambda_a = \psi_a(y_a) \triangleq \frac{\delta y_a}{c_a - y_a} \]

**TCP-Vegas** (queueing delay)

\[ q_s = f_s^{-1}(x_s) \triangleq \frac{\alpha \tau_s}{x_s} \]

\[ \lambda_a = \psi_a(y_a) \triangleq \frac{y_a}{c_a - y_a} \]
Steady state – Primal optimality

\[ x_s = f_s(\sum_{a \in s} \lambda_a) \]
\[ \lambda_a = \psi_a(\sum_{s \ni a} x_s) \]

\[ f_s^{-1}(x_s) = \sum_{a \in s} \lambda_a = \sum_{a \in s} \psi_a(\sum_{u \ni a} x_u) \]

\( \equiv \) optimal solution of strictly convex program

\[
(P) \quad \min_x \sum_{s \in S} U_s(x_s) + \sum_{a \in A} \psi_a(\sum_{s \ni a} x_s)
\]

\[ U'_s(\cdot) = -f_s^{-1}(\cdot) \]
\[ \psi'_a(\cdot) = \psi_a(\cdot) \]
Steady state – Primal optimality

\[
x_s = f_s(\sum_{a \in s} \lambda_a)
\]
\[
\lambda_a = \psi_a(\sum_{s \ni a} x_s)
\]
\[
f_s^{-1}(x_s) = \sum_{a \in s} \lambda_a = \sum_{a \in s} \psi_a(\sum_{u \ni a} x_u)
\]
\[
\equiv \text{optimal solution of strictly convex program}
\]

\[
(P) \quad \min_x \sum_{s \in S} U_s(x_s) + \sum_{a \in A} \psi_a(\sum_{s \ni a} x_s)
\]
\[
U'_s(\cdot) = -f_s^{-1}(\cdot)
\]
\[
\psi'_a(\cdot) = \psi_a(\cdot)
\]
Steady state – Dual optimality

\[ x_s = f_s(\sum_{a \in s} \lambda_a) \]
\[ \lambda_a = \psi_a(\sum_{s \ni a} x_s) \]

\[ \psi_a^{-1}(\lambda_a) = \sum_{s \ni a} x_s = \sum_{s \ni a} f_s(\sum_{b \in s} \lambda_b) \]

\[ (D) \quad \min_{\lambda} \sum_{a \in A} \psi_a^*(\lambda_a) + \sum_{s \in S} U_s^*(\sum_{a \in s} \lambda_a) \]
Steady state – Dual optimality

\[ x_s = f_s(\sum_{a \in s} \lambda_a) \]
\[ \lambda_a = \psi_a(\sum_{s \ni a} x_s) \]

\[ \psi_a^{-1}(\lambda_a) = \sum_{s \ni a} x_s = \sum_{s \ni a} f_s(\sum_{b \in s} \lambda_b) \]

\( \equiv \) optimal solution of strictly convex program

\[ (D) \quad \min_{\lambda} \sum_{a \in A} \psi_a^*(\lambda_a) + \sum_{s \in S} U_s^*(\sum_{a \in s} \lambda_a) \]
Theorem (Low’2003)

\[ x_s = f_s\left(\sum_{a \in s} \lambda_a\right) \]
\[ \lambda_a = \psi_a\left(\sum_{s \ni a} x_s\right) \]

\( \Leftrightarrow \)

\( x \) and \( \lambda \) are optimal solutions for \((P)\) and \((D)\) respectively

Relevance:
- Reverse engineering of existing protocols / forward engineering \((f_s, \psi_a)\)
- Design distributed stable protocols to optimize prescribed utilities
- Flexible choice of congestion measure \(q_s\)

Limitations:
- Ignores delays in transmission of congestion signals
- Improper account of stochastic phenomena
- Single-path routing
Theorem (Low’2003)

\[
x_s = f_s(\sum_{a \in s} \lambda_a) \\
\lambda_a = \psi_a(\sum_{s \ni a} x_s)
\]

\[\Leftrightarrow\]

\(x\) and \(\lambda\) are optimal solutions for \((P)\) and \((D)\) respectively

Relevance:

- Reverse engineering of existing protocols / forward engineering \((f_s, \psi_a)\)
- Design distributed stable protocols to optimize prescribed utilities
- Flexible choice of congestion measure \(q_s\)

Limitations:

- Ignores delays in transmission of congestion signals
- Improper account of stochastic phenomena
- Single-path routing
Theorem (Low’2003)

\[ x_s = f_s \left( \sum_{a \in s} \lambda_a \right) \]
\[ \lambda_a = \psi_a \left( \sum_{s \ni a} x_s \right) \]

\[ \iff \quad x \text{ and } \lambda \text{ are optimal solutions for (P) and (D) respectively} \]

Relevance:
- Reverse engineering of existing protocols / forward engineering \((f_s, \psi_a)\)
- Design distributed stable protocols to optimize prescribed utilities
- Flexible choice of congestion measure \(q_s\)

Limitations:
- Ignores delays in transmission of congestion signals
- Improper account of stochastic phenomena
- Single-path routing
Markovian Network Utility Maximization (MNUM)

- Increase transmission rates: single path $\rightarrow$ multi-path
- Goal: design distributed TCP protocols with multi-path routing
- Packet-level protocol that is stable and satisfies optimality criteria
- Model based on the notion of Markovian traffic equilibrium
MNUM: integrated routing & rate control

- Cross-layer design: routing + rate control
- Based on a common congestion measure: delay
- Link random delays $\tilde{\lambda}_a = \lambda_a + \epsilon_a$ with $\mathbb{E}(\epsilon_a) = 0$

$$\tilde{\lambda}_a = \text{Queuing} + \text{Transmission} + \text{Propagation}$$
MNUM: Markovian multipath routing

At switch $i$, packets headed to destination $d$ are routed through the outgoing link $a \in A_i^+$ that minimizes the "observed" cost-to-go

$$\tilde{\tau}_i^d = \min_{a \in A_i^+} \tilde{\lambda}_a + \tilde{\tau}_{j_a}^d$$

Markov chain with transition matrix

$$P_{ij}^d = \begin{cases} \mathbb{P}(\tilde{Z}_a^d \leq \tilde{Z}_b^d, \forall b \in A_i^+) & \text{if } i = i_a, j = j_a \\ 0 & \text{otherwise} \end{cases}$$
MNUM: Markovian multipath routing

At switch $i$, packets headed to destination $d$ are routed through the outgoing link $a \in A_i^+$ that minimizes the “observed” cost-to-go

$$\tilde{\tau}_{i}^{d} = \min_{a \in A_i^+} \tilde{\lambda}_a + \tilde{\tau}_{j_a}^d$$

Markov chain with transition matrix

$$P_{ij}^d = \begin{cases} \mathbb{P}(\tilde{z}_a^d \leq \tilde{z}_b^d, \forall b \in A_i^+) & \text{if } i = i_a, j = j_a \\ 0 & \text{otherwise} \end{cases}$$
Expected flows (invariant measures)

The flow $\phi_i^d$ entering node $i$ and directed towards $d$

$$\phi_i^d = \sum_{o_s=i} x_s + \sum_{a \in A_i^-} v_a^d$$

splits among the outgoing links $a = (i, j)$ according to

$$v_a^d = \phi_i^d P_{ij}^d$$
Expected costs

Letting \( z_a^d = \mathbb{E}(\tilde{z}_a^d) \) and \( \tau_i^d = \mathbb{E}(\tilde{\tau}_i^d) \), we have

\[
\begin{align*}
  z_a^d &= \lambda_a + \tau_{j_a}^d \\
  \tau_i^d &= \varphi_i^d(z^d)
\end{align*}
\]

with

\[
\varphi_i^d(z^d) \triangleq \mathbb{E}(\min_{a \in A_i^+} [z_a^d + \epsilon_a^d])
\]

Moreover

\[
\mathbb{P}\left( \tilde{z}_a^d \leq \tilde{z}_b^d, \forall b \in A_i^+ \right) = \frac{\partial \varphi_i^d}{\partial z_a^d}(z^d)
\]
Expected costs

Letting $z_a^d = \mathbb{E}(\tilde{z}_a^d)$ and $\tau_i^d = \mathbb{E}(\tilde{\tau}_i^d)$, we have

$$z_a^d = \lambda_a + \tau_{ja}^d$$

$$\tau_i^d = \varphi_i^d(z^d)$$

with

$$\varphi_i^d(z^d) \triangleq \mathbb{E}(\min_{a \in A_i^+} [z_a^d + \epsilon_a^d])$$

Moreover

$$\mathbb{P} \left( \tilde{z}_a^d \leq \tilde{z}_b^d, \forall b \in A_i^+ \right) = \frac{\partial \varphi_i^d}{\partial z_a^d}(z^d)$$
Markovian NUM – Definition

\[ x_s = f_s(q_s) \quad \text{(source rate control)} \]
\[ \lambda_a = \psi_a(y_a) \quad \text{(link congestion)} \]
\[ y_a = \sum_d v_a^d \quad \text{(total link flows)} \]
\[ q_s = \tau_s - \tau_s^0 \quad \text{(end-to-end queuing delay)} \]

where \( \tau_s = \tau_{os}^d \) with expected costs given by

\[
(ZQ) \quad \left\{ \begin{array}{l}
    z_a^d = \lambda_a + \tau_{ja}^d \\
    \tau_i^d = \varphi_i^d(z^d)
\end{array} \right.
\]

and expected flows \( v^d \) satisfying

\[
(FC) \quad \left\{ \begin{array}{l}
    \phi_i^d = \sum_{os=i} x_s + \sum_{a \in A_i^-} v_a^d \quad \forall i \neq d \\
    v_a^d = \phi_i^d \frac{\partial \varphi_i^d}{\partial z_a^d}(z^d) \quad \forall a \in A_i^+ 
\end{array} \right.
\]
**MNUM Characterization: Dual problem**

- (ZQ) defines implicitly $z^d_a$ and $\tau^d_i$ as concave functions of $\lambda$
- $x_s = f_s(q_s)$ with $q_s = \tau^d_{0s}(\lambda) - \tau^d_{os}(\lambda^0)$ yields $x_s$ as a function of $\lambda$
- (FC) then defines $v^d_a$ as functions of $\lambda$

MNUM conditions $\iff \psi^{-1}_a(\lambda_a) = y_a = \sum_d v^d_a(\lambda)$

**Theorem**

**MNUM $\iff$ optimal solution of the strictly convex program**

$$(D) \min_{\lambda} \sum_{a \in A} \Psi^*_a(\lambda_a) + \sum_{s \in S} U^*_s(q_s(\lambda))$$
MNUM Characterization: Primal problem

Theorem

\[ \text{MNUM} \iff \text{optimal solution of} \]

\[ \min_{(x,y,v) \in P} \sum_{s \in S} U_s(x_s) + \sum_{a \in A} \psi_a(y_a) + \sum_{d \in D} \chi^d(v^d) \]

where

\[ \chi^d(v^d) = \sup_{z^d} \sum_{a \in A} (\varphi^d_{i_a}(z^d) - z^d_a)v^d_a \]

and \( P \) is the polyhedron defined by flow conservation constraints.
SUMMARY

- Described an optimization model for TCP/IP equilibrium rates
- Model extended to multipath routing & rate control (MNUM)
- Inspired from packet-level distributed protocols
- Implementable under current TCP/IP standards

FUTURE WORK

- Simulation and testing of MNUM-based protocols
- Investigate stochastic-stability of protocols
- Investigate delay-stability of protocols
- ECN mechanisms for congestion signals
SUMMARY

- Described an optimization model for TCP/IP equilibrium rates
- Model extended to multipath routing & rate control (MNUM)
- Inspired from packet-level distributed protocols
- Implementable under current TCP/IP standards

FUTURE WORK

- Simulation and testing of MNUM-based protocols
- Investigate stochastic-stability of protocols
- Investigate delay-stability of protocols
- ECN mechanisms for congestion signals
Some references


