Layering As Optimization Decomposition:
A Mathematical Theory of Network Architectures

Mung Chiang
Electrical Engineering Department, Princeton
Steven H. Low
Computer Science and Electrical Engineering Departments, Caltech
A. Robert Calderbank
Electrical Engineering and Mathematics Departments, Princeton

IEEE ISIT Tutorial
July 9, 2006

Schedule of the Tutorial

2:00 – 2:30pm Overview (Chiang)

2:30 – 3:00pm TCP: reverse and forward engineering (Low)

3:00 – 3:20pm MAC: reverse and forward engineering (Chiang)

3:20 – 3:35pm Decomposition theory and alternative decompositions (Chiang)

3:35 – 3:45pm Break

3:45 – 4:00pm Case 1: Joint congestion control and coding (Calderbank)

4:00 – 4:15pm Case 2: Joint congestion control, routing, and scheduling (Chiang)

4:15 – 4:30pm Case 3: TCP/IP interactions (Low)

4:30 – 4:50pm Future research challenges and Summary (30 open issues, 20 methodologies, 10 key messages) (Chiang)

4:50 – 5:00pm Question and Answers

Part I

Overview

Nature of the Tutorial

M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures" *Proceedings of IEEE*, December 2006.

- Give an overview of the topic. Details in various papers
- Not exhaustive survey. Highlight the key ideas and challenges
- Biased presentation. Focus on work by us

This is an appetizer. The beef in the papers

Outline

- Background: Holistic view on layered architecture
- Background: NUM and G.NUM

- Horizontal Decompositions
- Vertical Decompositions
- Alternative Decompositions

- Key Messages
- Key Methodologies
- Open Issues

Acknowledgement

Collaborators:

Stephen Boyd, Lijun Chen, Dah Ming Chiu, Neil Gershenfeld, Andrea Goldsmith, Prashanth Hande, Jiayue He, Sanjay Hegde, Maryam Fazel, Cheng Jin, Koushik Kar, Frank Kelly, P. R. Kumar, Jang-Won Lee, Ruby Lee, Lun Li, Ying Li, Xiaojun Lin, Jiaping Liu, Zhen Liu, Asuman Ozdaglar, Daniel Palomar, Pablo Parrilo, Jennifer Rexford, Devavrat Shah, Ness Shroff, R. Srikant, Chee Wei Tan, Ao Tang, Jiantao Wang, Xin Wang, David Wei, Bartek Wydrowski, Lin Xiao, Edmund Yeh, and Junshan Zhang

Funding Agencies:

NSF grants CCF-0440043, CCF-0448012, CNS-0417607, CNS-0427677, CNS-0430487, CNS-0519880, DARPA grant HR0011-06-1-0008, and Cisco grant GH072605

Layered Network Architecture

Application
Presentation
Session
Transport
Network
Link
Physical

Important foundation for data networking

Ad hoc design historically (within and across layers)

Rethinking Layering

How to, and how not to, layer? A question on architecture

Functionality allocation: who does what and how to connect them?

But want answers to be rigorous, quantitative, simple, and relevant

- How to modularize (and connect)?
- How to distribute (and connect)?
- How to search in the design space of alternative architectures?
- How to quantify the benefits of better codes/modulation/schedule/routes... for network applications?

A common language to rethink these issues?

The Goal

A Mathematical Theory of Network Architectures

- Particular focus on the architectures of layering and distributed control
- There are also boundaries to the use of mathematical approach to the economics, psychology, and engineering of network architectures
- But certainly provides rigorous approaches on why protocols work, when it will not work, and how to make it work better
- Also provides conceptually clear understanding on the opportunities and risks of cross layer design

Layering As Optimization Decomposition

The first unifying view and systematic approach

Network: Generalized NUM

Layering architecture: Decomposition scheme

Layers: Decomposed subproblems

Interfaces: Functions of primal or dual variables

Horizontal and vertical decompositions through

- implicit message passing (e.g., queuing delay, SIR)
- explicit message passing (local or global)

3 Steps: G.NUM $\Rightarrow$ A solution architecture $\Rightarrow$ Alternative architectures

Network Utility Maximization

Basic NUM (KellyMaulloTan98):

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s) \\ \text{subject to} & \mathbf{R}\mathbf{x} \preceq \mathbf{c} \\ & \mathbf{x} \succeq 0 \end{array}$$

Generalized NUM (one possibility shown here) (Chiang05a):

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s, P_{e,s}) + \sum_j V_j(w_j) \\ \text{subject to} & \mathbf{R}\mathbf{x} \preceq \mathbf{c}(\mathbf{w}, \mathbf{P}_e) \\ & \mathbf{x} \in \mathcal{C}_1(\mathbf{P}_e) \\ & \mathbf{x} \in \mathcal{C}_2(\mathbf{F}) \ \text{ or } \ \mathbf{x} \in \Pi \\ & \mathbf{R} \in \mathcal{R} \\ & \mathbf{F} \in \mathcal{F} \\ & \mathbf{w} \in \mathcal{W} \end{array}$$

GNUM

- Objective function: What the end-users and network provider care about (can be coupled, eg, one utility function for the whole network)
- Constraint set: Physical and economic limitations
- Variables: Under the control of this design
- Constants: Beyond the control of this design

Two Cornerstones for Conceptual Simplicity

Networks as optimizers

Reverse engineering mentality: give me the solution (an existing protocol), I'll find the underlying problem implicitly being solved

- Why care about the problem if there's already a solution?
- It leads to simple, rigorous understanding for systematic design

Layering as decomposition

1. Analytic foundation for network architecture
2. Common language for thinking and comparing
3. Methodologies, analytic tools

Layering as Optimization Decomposition

What's so unique about this particular framework for cross-layer design?

- Network as optimizer
- End-user application utilities as the driver
- Performance benchmark without any layering
- Unified approach to cross-layer design (it simplifies our understanding about network architecture)
- Separation theorem among modules
- Systematic exploration of architectural alternatives

Not every cross-layer paper is 'layering as optimization decomposition'

Utility

Which utility? (function of rate, useful information, delay, energy...)

1. Reverse engineering: TCP maximizes utilities

2a. Behavioral model: user satisfaction

2b. Traffic model: traffic elasticity

3a. Economics: resource allocation efficiency

3b. Economics: different utility functions lead to different fairness

Three choices: Weighted sum, Pareto optimality, Uncooperative game

- Goal: Distributed and modularized algorithm converging to globally and jointly optimum resource allocation
- Limitations to be discussed at the end

Layers

Restriction: we focus on resource allocation functionalities rather than semantics functionalities

- TCP: congestion control

Different meanings:

- Routing: RIP/OSPF, BGP, wireless routing, optical routing, dynamic/static, single-path/multi-path, multicommodity flow routing...
- MAC: scheduling or contention-based
- PHY: power control, coding, modulation, antenna signal processing...

Insights on both:

- What each layer can do (Optimization variables)
- What each layer can see (Constants, Other subproblems' variables)

Connections With Mathematics

- Convex and nonconvex optimization
- Decomposition and distributed algorithm
- Game theory, General market equilibrium theory

- Algebraic geometry (nonconvex formulations)
- Differential topology (heterogeneous protocols)

Adoption By Industry

Industry adoption of Layering As Optimization Decomposition:

- Internet resource allocation: TCP FAST (Caltech)
- Protocol stack design: Internet 0 (MIT)
- Broadband access: FAST Copper (Princeton, Stanford, Fraser)

This tutorial is mainly about the underlying common language and methodologies

Network Utility Maximization

BNUM (KellyMaulloTan98):

$$
\begin{array}{ll}
\text{maximize} & \sum_s U_s(x_s) \\
\text{subject to} & \mathbf{R}\mathbf{x} \preceq \mathbf{c} \\
& \mathbf{x} \succeq 0
\end{array}
$$

GNUM (one possibility shown here) (Chiang05a):

$$
\begin{array}{ll}
\text{maximize} & \sum_s U_s(x_s, P_{e,s}) + \sum_j V_j(w_j) \\
\text{subject to} & \mathbf{R}\mathbf{x} \preceq \mathbf{c}(\mathbf{w}, \mathbf{P}_e) \\
& \mathbf{x} \in \mathcal{C}_1(\mathbf{P}_e) \\
& \mathbf{x} \in \mathcal{C}_2(\mathbf{F}) \;\; \text{or} \;\; \mathbf{x} \in \Pi \\
& \mathbf{R} \in \mathcal{R} \\
& \mathbf{F} \in \mathcal{F} \\
& \mathbf{w} \in \mathcal{W}
\end{array}
$$

Dual-based Distributed Algorithm

BNUM with concave smooth utility functions:

Convex optimization (Monotropic Programming) with zero duality gap

Lagrangian decomposition:

$$
\begin{aligned}
L(\mathbf{x}, \boldsymbol{\lambda}) &= \sum_s U_s(x_s) + \sum_l \lambda_l \left(c_l - \sum_{s: l \in L(s)} x_s \right) \\
&= \sum_s \left[U_s(x_s) - \left(\sum_{l \in L(s)} \lambda_l \right) x_s \right] + \sum_l c_l \lambda_l \\
&= \sum_s L_s(x_s, \lambda^s) + \sum_l c_l \lambda_l
\end{aligned}
$$

Dual problem:

$$
\begin{aligned}
\text{minimize} \quad & g(\boldsymbol{\lambda}) = L(\mathbf{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}) \\
\text{subject to} \quad & \boldsymbol{\lambda} \succeq 0
\end{aligned}
$$

Dual-based Distributed Algorithm

Source algorithm:

$$x_s^*(\lambda^s) = \text{argmax}\left[U_s(x_s) - \lambda^s x_s\right], \quad \forall s$$

- Selfish net utility maximization locally at source s

Link algorithm (gradient or subgradient based):

$$\lambda_l(t+1) = \left[\lambda_l(t) - \alpha(t)\left(c_l - \sum_{s:l\in L(s)} x_s(\lambda^s(t))\right)\right]^+, \quad \forall l$$

- Balancing supply and demand through pricing

Certain choices of step sizes $\alpha(t)$ of distributed algorithm guarantee convergence to globally optimal $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$

Primal-Dual

Different meanings:

- Primal-dual interior-point algorithm
- Primal-dual solution
- Primal or dual driven control
- Primal or dual decomposition

Coupling in constraints (easy: flow constraint, hard: SIR feasibility)

Coupling in objective (easy: additive form, hard: min max operations)

Primal Decomposition

Simple example:

$$x + y + z + w \leq c$$

Decomposed into:

$$\begin{aligned} x + y &\leq \alpha \\ z + w &\leq c - \alpha \end{aligned}$$

New variable α updated by various methods

Interpretation: Direct resource allocation (not pricing-based control)

Engineering implications: Adaptive slicing (GENI)

Pricing feedback: dual decomposition

Adaptive slicing: primal decomposition

Horizontal Decompositions

Reverse engineering:

- Layer 4 TCP congestion control: Basic NUM (LowLapsley99, RobertsMassoulie99, MoWalrand00, YaicheMazumdarRosenberg00, KunniyurSrikant02, LaAnatharam02, LowPaganiniDoyle02, Low03, Srikant04...)
- Layer 4 TCP heterogeneous protocol: Nonconvex equilibrium problem (TangWangLowChiang05)
- Layer 3 IP inter-AS routing: Stable Paths Problem (GriffinSheperdWilfong02)
- Layer 2 MAC backoff contention resolution: Non-cooperative Game (LeeChiangCalderbank06a)

Forward engineering for horizontal decompositions also carried out recently

Vertical Decompositions

A partial list of work along this line:

- Jointly optimal congestion control and adaptive coding or power control (Chiang05a, LeeChiangCalderbank06b)
- Jointly optimal congestion and contention control (KarSarkarTassiulas04, ChenLowDoyle05, WangKar05, YuenMarbach05, ZhengZhang06, LeeChiangCalderbank06c)
- Jointly optimal congestion control and scheduling (ErilymazSrikant05)
- Jointly optimal routing and scheduling (KodialamNandagopal03)
- Jointly optimal routing and power control (XiaoJohanssonBoyd04, NeelyModianoRohrs05)
- Jointly optimal congestion control, routing, and scheduling (LinShroff05, ChenLowChiangDoyle06)

Vertical Decompositions

- Jointly optimal routing, scheduling, and power control (CruzSanthanam03, XiYeh06)

- Jointly optimal routing, resource allocation, and source coding (YuYuan05)

- TCP/IP interactions (WangLiLowDoyle05, HeChiangRexford06) and jointly optimal congestion control and routing (KellyVoice05, Hanetal05)

- Network lifetime maximization (NamaChiangMandayam06)

- Application adaptation and congestion control/resource allocation (ChangLiu04, HuangLiChiangKatsaggelos06)

Apology, Apology, Apology for any missing reference

Vertical Decompositions

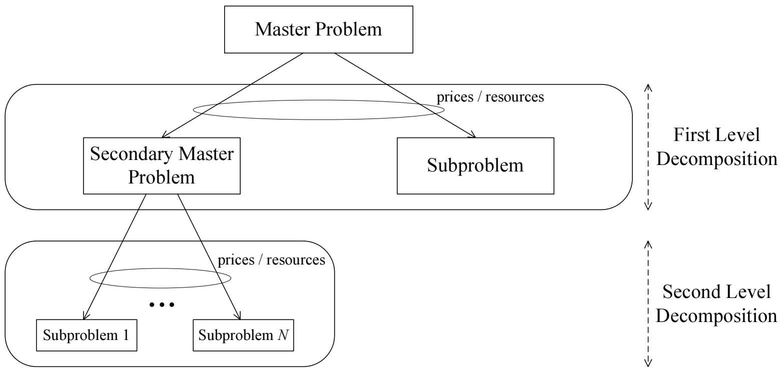
- Specific designs not important
- Common language and key messages methodologies important

Goal: Shrink, not grow knowledge tree on cross-layer design

Alternative Decompositions

Many ways to decompose:

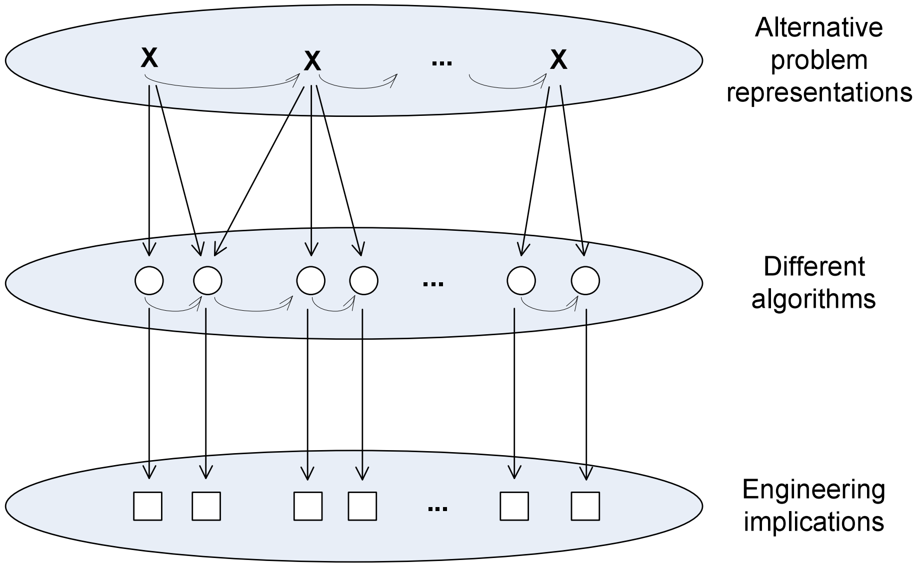
- Primal and dual decomposition
- Partial decomposition
- Multi-level decomposition



Lead to alternative architectures (PalomarChiang06) with different

- Communication overhead
- Computation distribution
- Convergence behavior

Alternative Decompositions



Systematically explore the space of alternative decompositions

Key Messages

- Protocols in layers 2,3,4 can be reverse engineered. Reverse engineering in turn leads to better design in a rigorous manner.

- There is a unifying approach to cross-layer design, illustrating both opportunities and risks.

- Loose coupling through "layering price" can be optimal and robust, and congestion price (or queuing delay, or buffer occupancy) is often the right "layering price" for stability and optimality, with important exceptions as well.

- User-generated pricing following end-to-end principle

- There are many alternatives in decompositions, leading to different divisions of tasks across layers and even different time-scales of interactions.

- Convexity of the generalized NUM is the key to devising a globally optimal solution.

- Decomposability of the generalized NUM is the key to devising a distributed solution.

Key Methodologies

- Dual decomposition for linear coupling constraints.
- Consistency pricing for coupled objective functions.
- Descent lemma for proof of convergence of dual-based distributed subgradient algorithm.
- Stability proof through Lyapunov function construction, singular perturbation theory, and passivity argument.
- Log change of variables to turn multiplicative coupling into linear coupling, and to turn nonconvex constraints to convex ones.
- Sufficient conditions on curvature of utility functions for it to remain concave after a log change of variables.
- Construction of conflict graph, contention matrix, and transmission modes in contention based MAC design.
- Maximum differential congestion pricing for node-based back-pressure scheduling (part of the connections between distributed convex optimization and stochastic control).

Future Research Issues

- Technical: Global stability under delay...
- Modeling: routing in ad hoc network, ARQ, MIMO...
- Time issues
- Why deterministic fluid model?

Shannon 1948: remove finite blocklength, Law of Large Numbers kicks in (later finite codewords come back...)

Kelly 1998: remove coupled queuing dynamics, optimization and decomposition view kicks in (later stochastics come back...)

- What if it's not convex optimization?

Rockafellar 1993: Convexity is the watershed between easy and hard (what if it's hard?)

- Is performance the only optimization objective?

Future Research: Time Issues

- Rate of convergence
- Timescale separation
- Transient behavior bounding
- Utility as a function of latency
- Utility as a function of transient rate allocations

Future Research: Stochastic Issues

Fill the table with 3 stars in all entries:

Union of Stochastic Network and Network Optimization

	Stability or Validation	Average Performance	Outage Performance	Fairness
Session Level	⋆⋆	⋆		⋆
Packet Level	⋆	⋆		
Channel Level	⋆⋆	⋆		
Topology Level				

Table 1: State-of-the-art in Stochastic Network Utility Maximization.

With a good layering architecture:

- Stochastic doesn't hurt
- Stochastic may help

Future Research: Nonconvexity Issues

- Nonconcave utility (eg, real-time applications)
- Nonconvex constraints (eg, power control in low SIR)
- Integer constraints (eg, single-path routing)
- Exponentially long description length (eg, scheduling)

Convexity not invariant under embedding in higher dimensional space or nonlinear change of variable

- Sum-of-squares method (Stengle73, Parrilo03)
- Geometric programming (DuffinPetersonZener67, Chiang05b)

From optimal/complicated to suboptimal/simple modules (LinShroff05)

Future Research: Network X-ities Issues

From Bit to Utility to Control and Management

Over-optimized? Optimizing for what?

- Evolvability
- Scalability
- Diagnosability

Pareto-optimal tradeoff between Performance and Network X-ities

From Forward Engineering to Reverse Engineering to

- Design for Optimizability

More later

See lists of

- 30 open issues
- 20 methodologies
- 10 key messages

at the end of the tutorial

New Mentalities

Layering As Optimization Decomposition, but move away from:

- One architecture fits all
- Deterministic fluids
- Asymptotic convergence
- Optimality
- Optimization

Think about “right” decomposition in the “right” way

Contacts

chiangm@princeton.edu
www.princeton.edu/∼chiangm
slow@caltech.edu
netlab.caltech.edu
calderbk@princeton.edu

Part II

TCP Congestion Control: Reverse and Forward Engineering

Some References

- F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of Operations Research Society,* vol. 49, no. 3, pp.237-252, March 1998

- S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Transactions on Networking,* vol. 11, no. 4, pp. 525-536, August 2003

- R. Srikant, *The Mathematics of Internet Congestion Control*, Birkhauser, 2004

- C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: Motivation, architecture, algorithms, and performance", *Proc. IEEE INFOCOM*, March 2004

- A. Tang, J. Wang, S. H. Low, and M. Chiang, "Network equilibrium of heterogeneous congestion control protocols," *Proc. IEEE INFOCOM*, March 2005

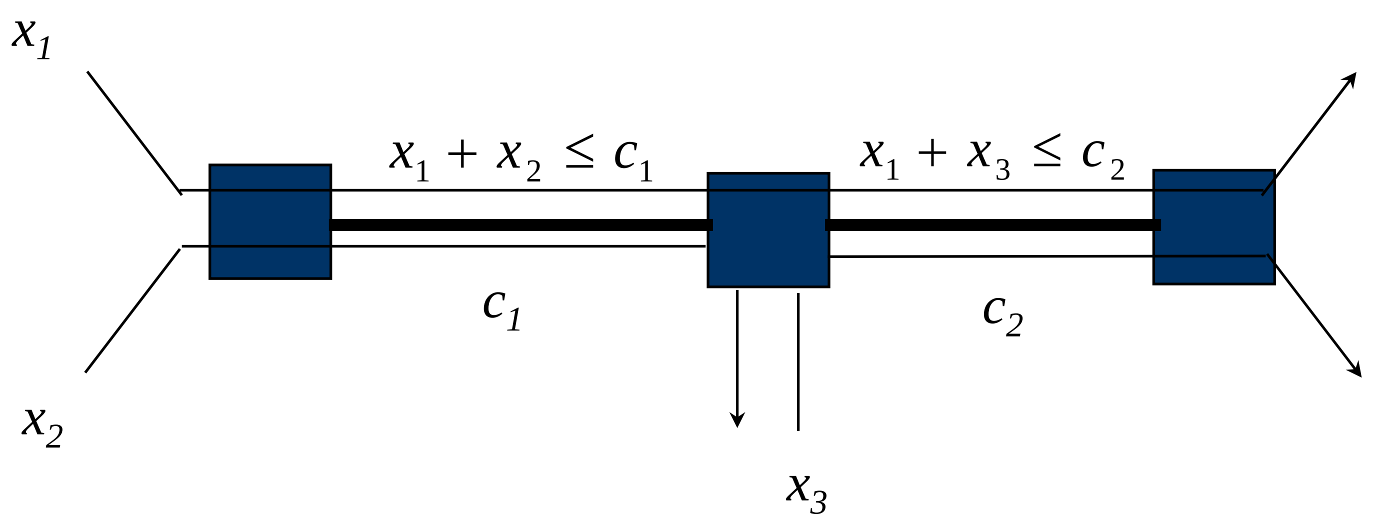
TCP Congestion Control

- Reverse engineering: B.NUM
- Forward engineering:
  - FAST
  - Heterogeneous protocols

Congestion control

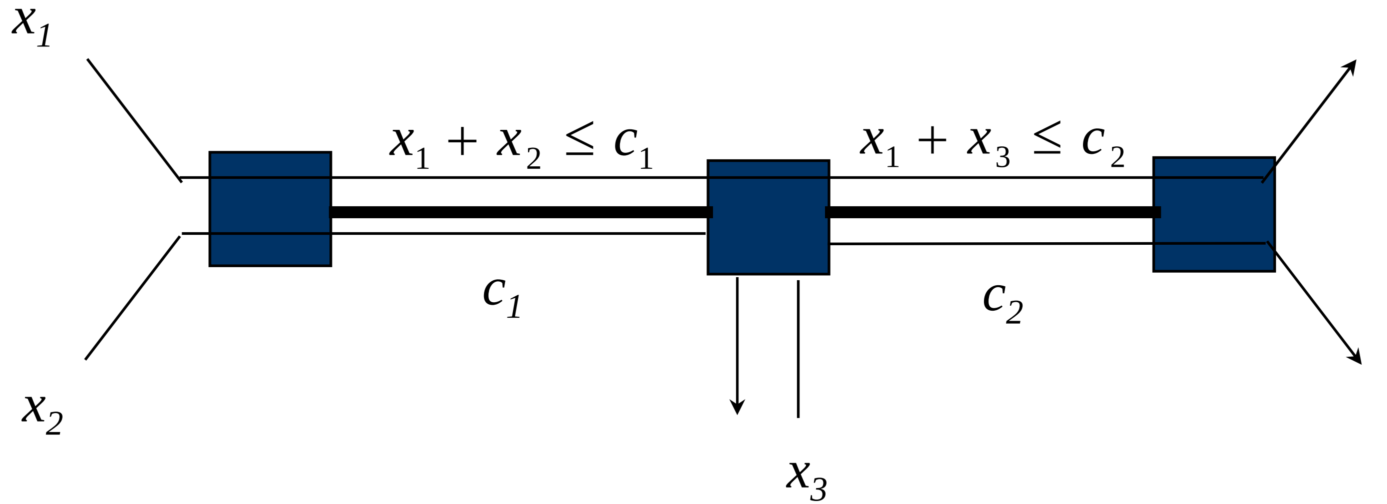
- Network: links l with capacity c_l
- Sources s: $L(s)$ - links used by source s
- TCP: dynamically adapts x_s to congestion to ensure

$$\sum_{i:l \in L(i)} x_i \le c_l \qquad \text{for all links } l$$

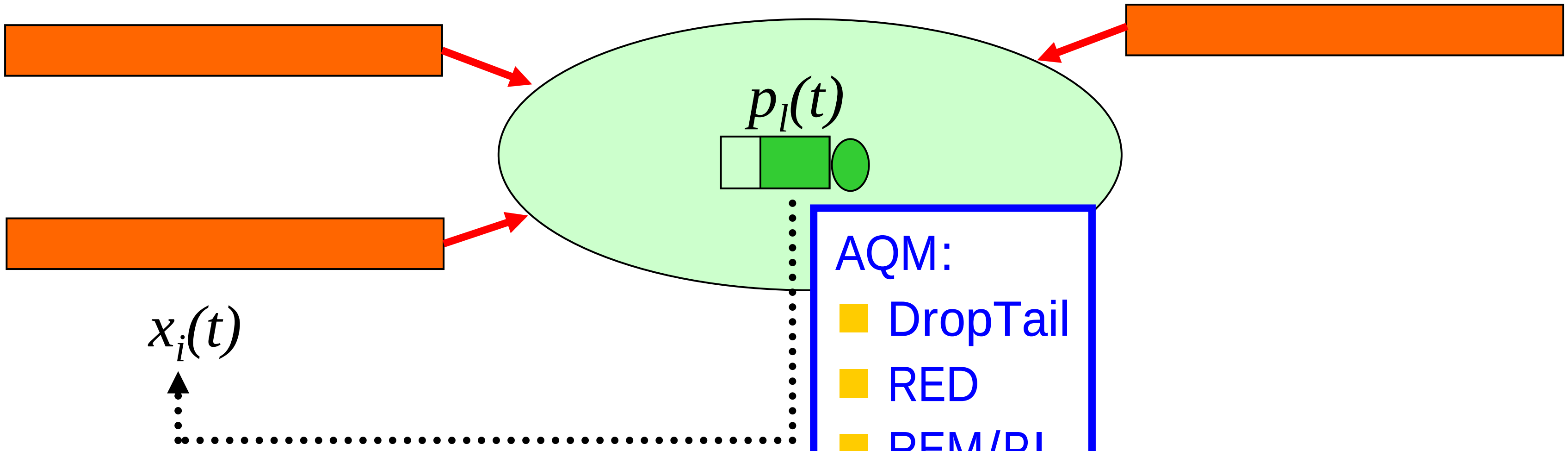


Congestion control

- Challenge: available info must be end-to-end
- Implicit congestion feedback
  - Loss probability: likelihood of a packet being delivered correctly
  - Round-trip time: time it takes for a packet to reach its destination and for its ack to return to the sender
- Explicit congestion feedback: marks, rates



TCP & AQM



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

TCP:
- Reno
- Vegas
- HSTCP
- FAST
- STCP

Reverse engineering

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

$$
\begin{aligned}
x(t+1) &= F\,(p(t),\ x(t)) \\
p(t+1) &= G\,(p(t),\ x(t))
\end{aligned}
$$

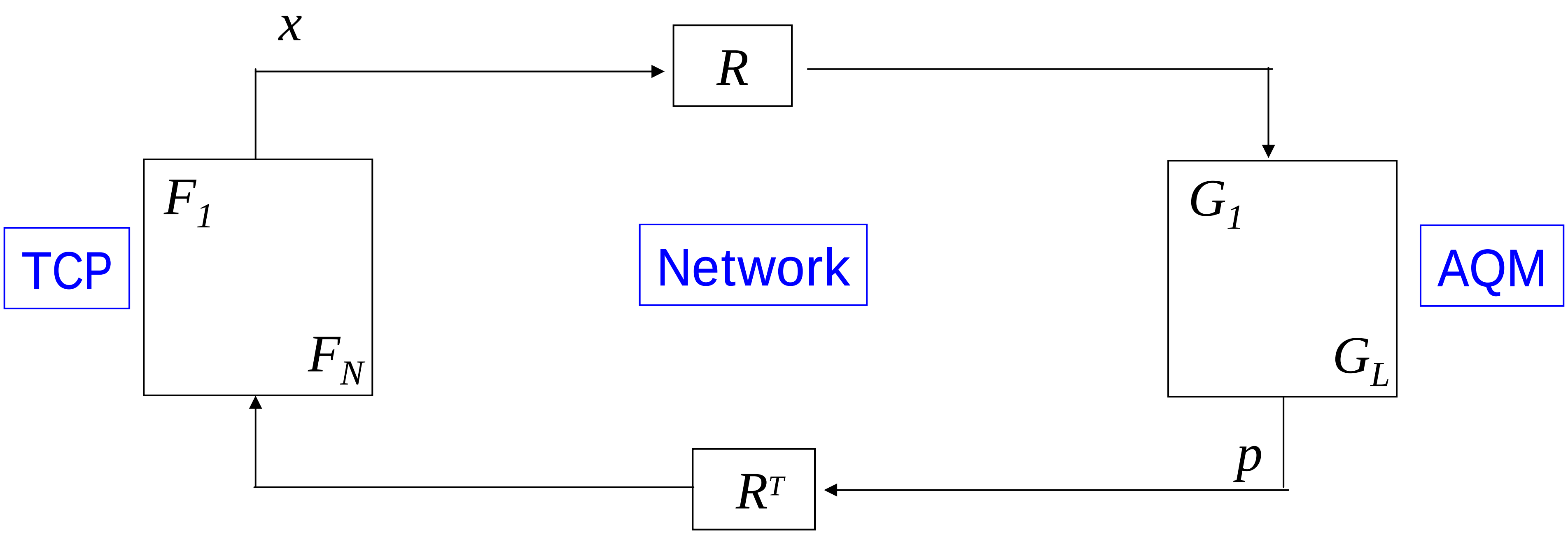
Equilibrium

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

Dynamics

- Local stability
- Global stability

Network model



$R_{li} = 1$ if source i uses link l ← IP routing

$$x(t+1) = F\,(R^T p(t),\ x(t))$$ ← Reno, Vegas

$$p(t+1) = G\,(p(t),\ Rx(t))$$ ← DT, RED, ...

Network model: example

Reno:
Jacobson
1989

```
for every RTT
{   W += 1   }
for every loss
{   W := W/2   }
```

(AI)

(MD)

AI

$$x_i(t+1) = \frac{1}{T_i^2} - \frac{x_i^2}{2} \sum_l R_{li} \, p_l(t)$$

MD

$$p_l(t+1) = G_l\left(\sum_i R_{li} \, x_i(t), \; p_l(t) \right)$$

TailDrop

Network model: example

FAST:

Jin, Wei, Low 2004

```
periodically
{
```

$$W := \frac{\text{baseRTT}}{\text{RTT}} W + \alpha$$

```
}
```

$$x_i(t+1) = x_i(t) + \frac{\gamma_i}{T_i}\left(\alpha_i - x_i(t)\sum_l R_{li}\, p_l(t)\right)$$

$$p_l(t+1) = p_l(t) + \frac{1}{c_l}\left(\sum_i R_{li}\, x_i(t) - c_l\right)$$

Duality model of TCP/AQM

- TCP/AQM
$$x^* = F(R^T p^*, x^*)$$
$$p^* = G(p^*, Rx^*)$$
- Equilibrium (x^*, p^*) primal-dual optimal:
$$\max_{x \geq 0} \sum U_i(x_i) \quad \text{subject to} \quad Rx \leq c$$
  - F determines utility function U
  - G guarantees complementary slackness
  - p^* are Lagrange multipliers

Kelly, Maloo, Tan 1998
Low, Lapsley 1999

Uniqueness of equilibrium
- x^* is unique when U is strictly concave
- p^* is unique when R has full row rank

Duality model of TCP/AQM

- TCP/AQM

$$x^* = F(R^T p^*, x^*)$$

$$p^* = G(p^*, Rx^*)$$

- Equilibrium (x^*, p^*) primal-dual optimal:

$$\max_{x \geq 0} \sum U_i(x_i) \quad \text{subject to} \quad Rx \leq c$$

  - F determines utility function U
  - G guarantees complementary slackness
  - p^* are Lagrange multipliers

Kelly, Maloo, Tan 1998
Low, Lapsley 1999

The underlying concave program also leads to simple dynamic behavior

Duality model of TCP/AQM

- Equilibrium *(x*,p*)* primal-dual optimal:

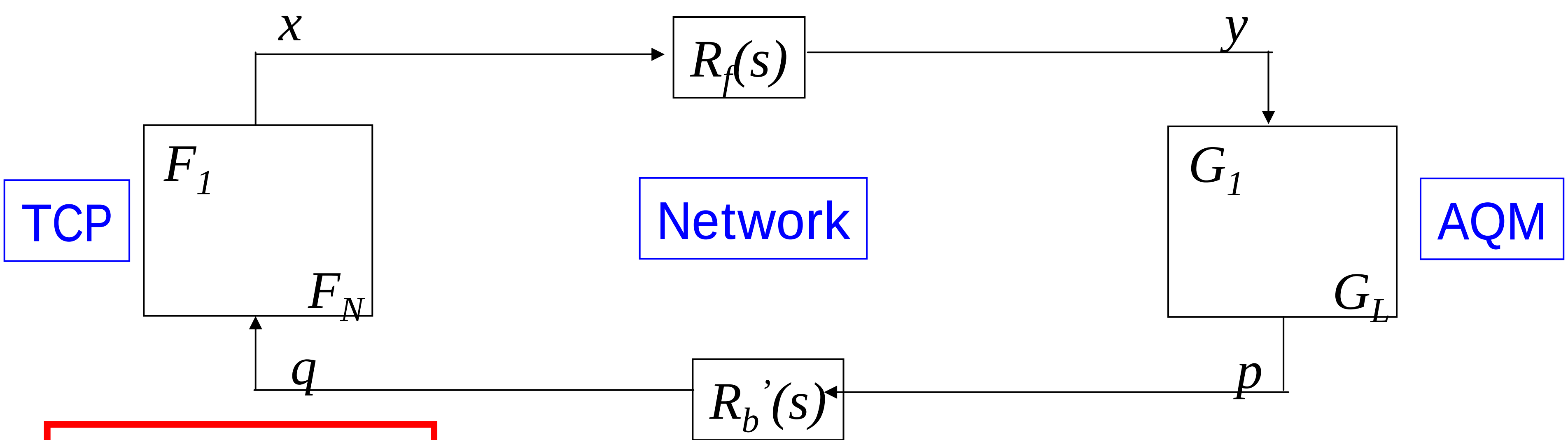
$$\max_{x \ge 0} \sum U_i(x_i) \qquad \text{subject to} \quad Rx \le c$$

Mo & Walrand 2000:

$$U_i(x_i) = \begin{cases} \log x_i & \text{if } \alpha = 1 \\ (1-\alpha)^{-1} x_i^{1-\alpha} & \text{if } \alpha \ne 1 \end{cases}$$

- $\alpha = 1$: Vegas, FAST, STCP
- $\alpha = 1.2$: HSTCP
- $\alpha = 2$: Reno
- $\alpha = \infty$: XCP (single link only)

Stability: Reno/RED



TCP:

- Small τ
- Small c
- Large N

RED:

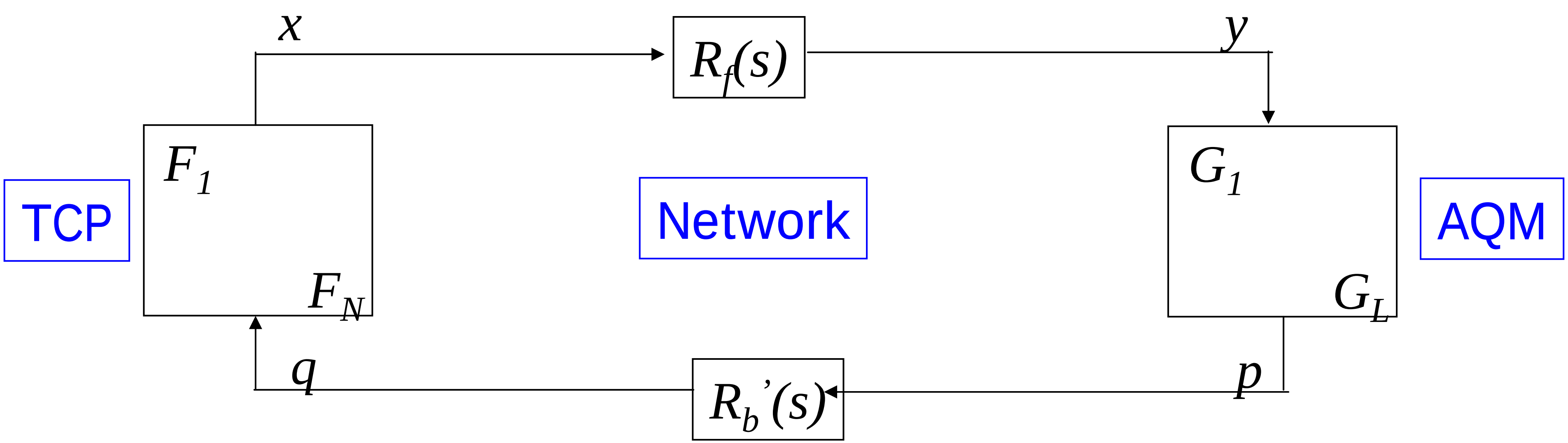
- Small ρ
- Large delay

Theorem (Low et al, Infocom'02)

Reno/RED is locally stable if

$$\frac{\rho}{2}\cdot\frac{c^3\tau^3}{N^3}(c\tau+N) \le \frac{\pi(1-\beta)^2}{\sqrt{4\beta^2+\pi^2(1-\pi)^2}}$$

Stability: scalable control

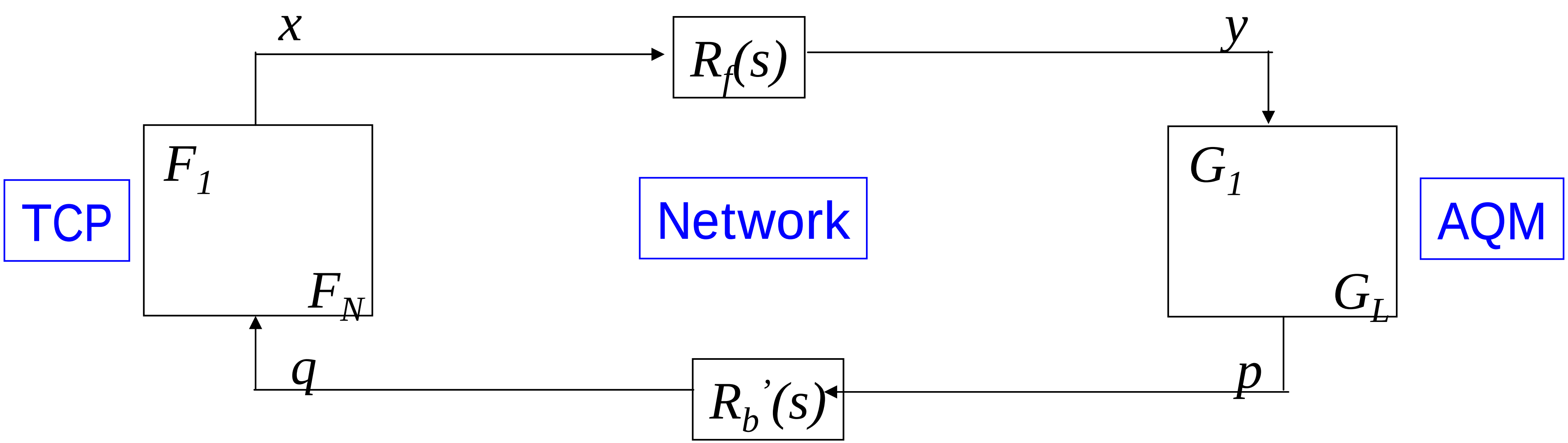


$$x_i(t) = \bar{x}_i \, e^{-\frac{\alpha_i}{\tau_i m_i} q_i(t)} \qquad \dot{p}_l(t) = \frac{1}{c_l}\left(y_l(t) - c_l\right)$$

<u>**Theorem**</u> (Paganini, Doyle, L, CDC'01)

Provided R is full rank, feedback loop is locally stable for arbitrary delay and capacity

Stability: FAST



$$\dot{w}_i = \kappa_i(t) \cdot \left(1 - \frac{q_i(t)}{u_i(t)}\right)$$

$$\boxed{\dot{p}_l(t) = \frac{1}{c_l}\left(y_l(t) - c_l\right)}$$

Application

- Stabilize TCP with current routers
- Queueing delay as congestion measure has right scaling
- Incremental deployment with ECN

Some implications

- Equilibrium
  - Always exists, unique if R is full rank
  - Bandwidth allocation independent of AQM or arrival
  - Can predict macroscopic behavior of large scale networks

- Counter-intuitive throughput behavior
  - Fair allocation is not always inefficient
  - Increasing link capacities do not always raise aggregate throughput

[Tang, Wang, Low, ToN 2006]

- FAST TCP
  - Design, analysis, experiments

[Jin, Wei, Low, ToN 2007]

Duality model

Historically

- Packet level implemented first
- Flow level understood as after-thought
- But flow level design determines
  - performance, fairness, stability

Now: can forward engineer

- Sophisticated theory on equilibrium & stability (optimization+control)
- Given (application) utility functions, can design provably scalable TCP algorithms

Packet level

- **Reno**
  AIMD(1, 0.5)

```
ACK:  W  ←  W + 1/W
Loss: W  ←  W – 0.5W
```

- **HSTCP**
  AIMD(a(w), b(w))

```
ACK:  W  ←  W + a(w)/W
Loss: W  ←  W – b(w)W
```

- **STCP**
  MIMD(a, b)

```
ACK:  W  ←  W + 0.01
Loss: W  ←  W – 0.125W
```

- **FAST**

$$\mathrm{RTT}: \mathrm{W} \leftarrow \mathrm{W} \cdot \frac{\mathrm{baseRTT}}{\mathrm{RTT}} + \alpha$$

Flow level: Reno, HSTCP, STCP, FAST

- **Common** flow level dynamics!

$$\dot{w}_i(t) \quad = \quad \kappa(t) \quad \cdot \quad \left(1 - \frac{p_i(t)}{U_i^{'}(t)}\right)$$

$$\boxed{\text{window adjustment}} \quad = \quad \boxed{\text{control gain}} \quad \boxed{\text{flow level goal}}$$

- **Different** gain κ and utility U_i
  - They determine equilibrium and stability
- **Different** congestion measure p_i
  - Loss probability (Reno, HSTCP, STCP)
  - Queueing delay (Vegas, FAST)

Flow level: Reno, HSTCP, STCP, FAST

- **Similar** flow level equilibrium

Reno $\quad x_i = \frac{1}{T_i} \cdot \frac{\alpha}{p_i^{0.5}}$ pkts/sec

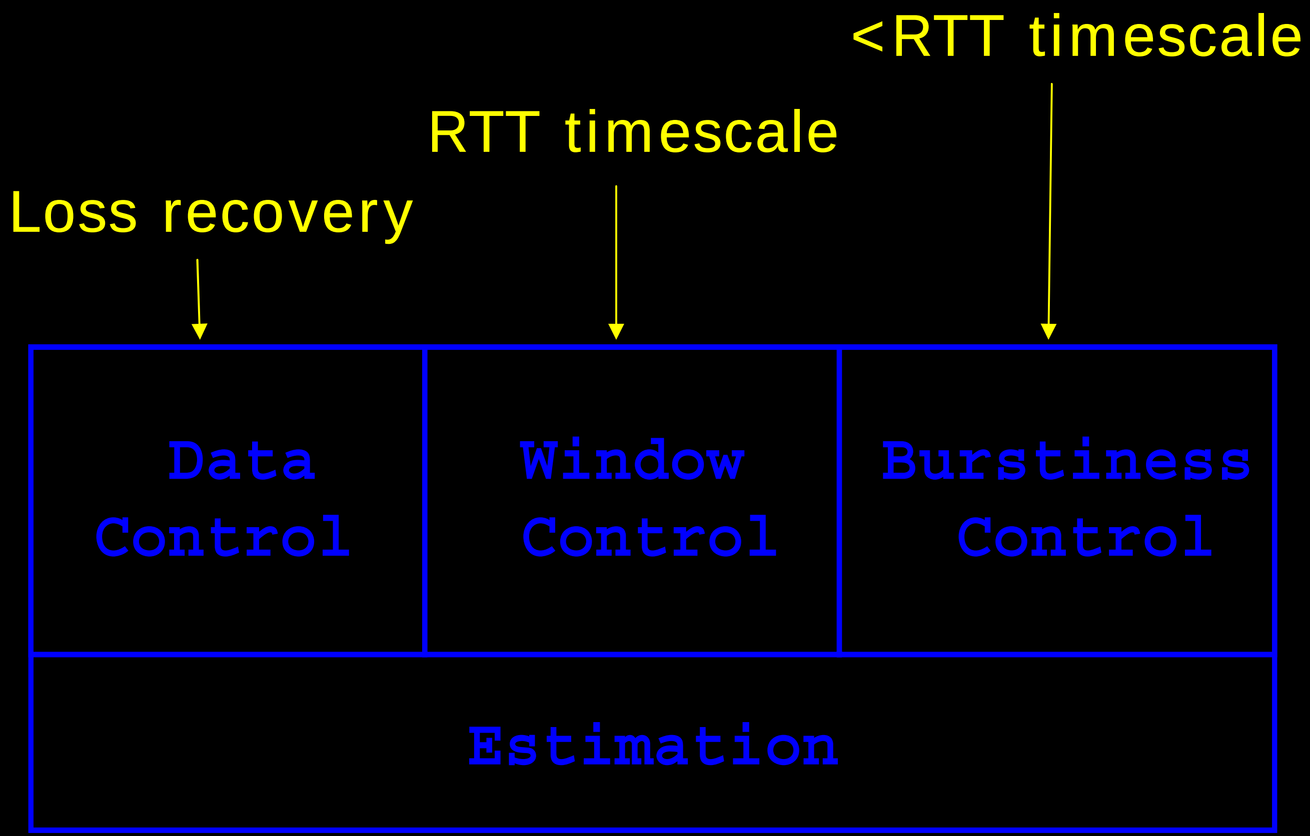
HSTCP $\quad x_i = \frac{1}{T_i} \cdot \frac{\alpha}{p_i^{0.84}}$

STCP $\quad x_i = \frac{1}{T_i} \cdot \frac{\alpha}{p_i}$

FAST $\quad x_i = \frac{\alpha}{p_i}$

α = 1.225 (Reno), 0.120 (HSTCP), 0.075 (STCP)

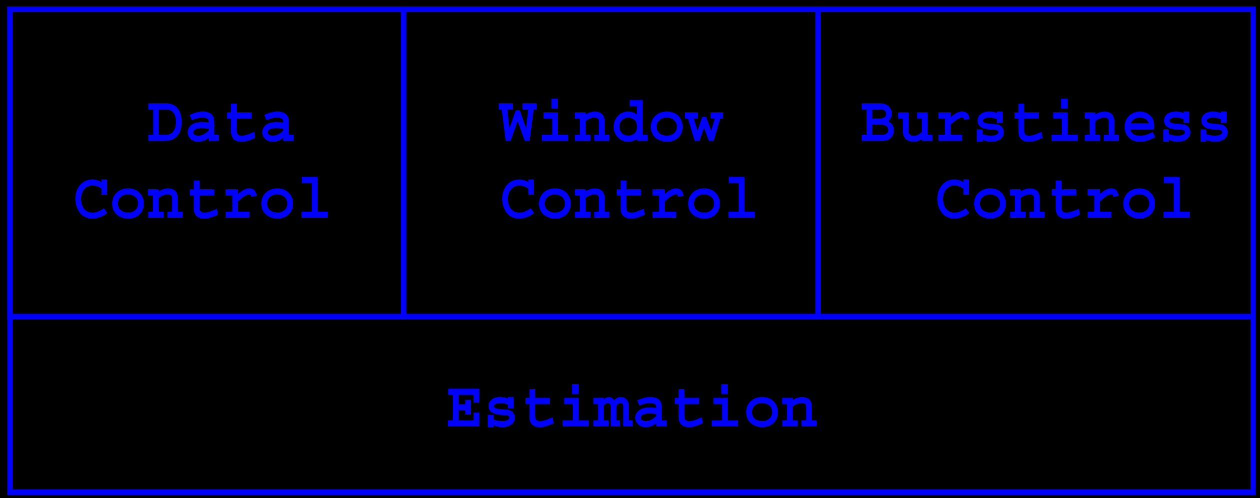
FAST Architecture



FAST Architecture

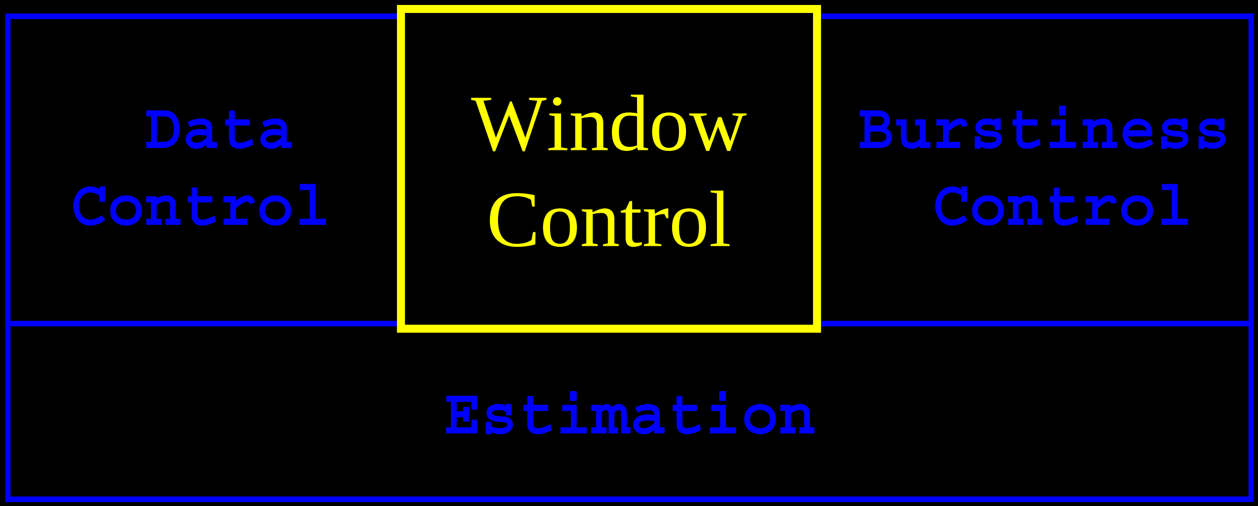
Each component

- designed independently
- upgraded asynchronously



FAST Architecture

Each component
- designed independently
- upgraded asynchronously



Window control algorithm

window update : $w_i(t+1) = w_i(t) + \gamma\left(\alpha_i - q(t)x_i(t)\right)$

self - clocking : $\sum \dfrac{w_i(t)}{d_i + q_i(t)} = c_l$

Window control algorithm

window update : $w_i(t+1) = w_i(t) + \gamma\left(\alpha_i - q(t)x_i(t)\right)$

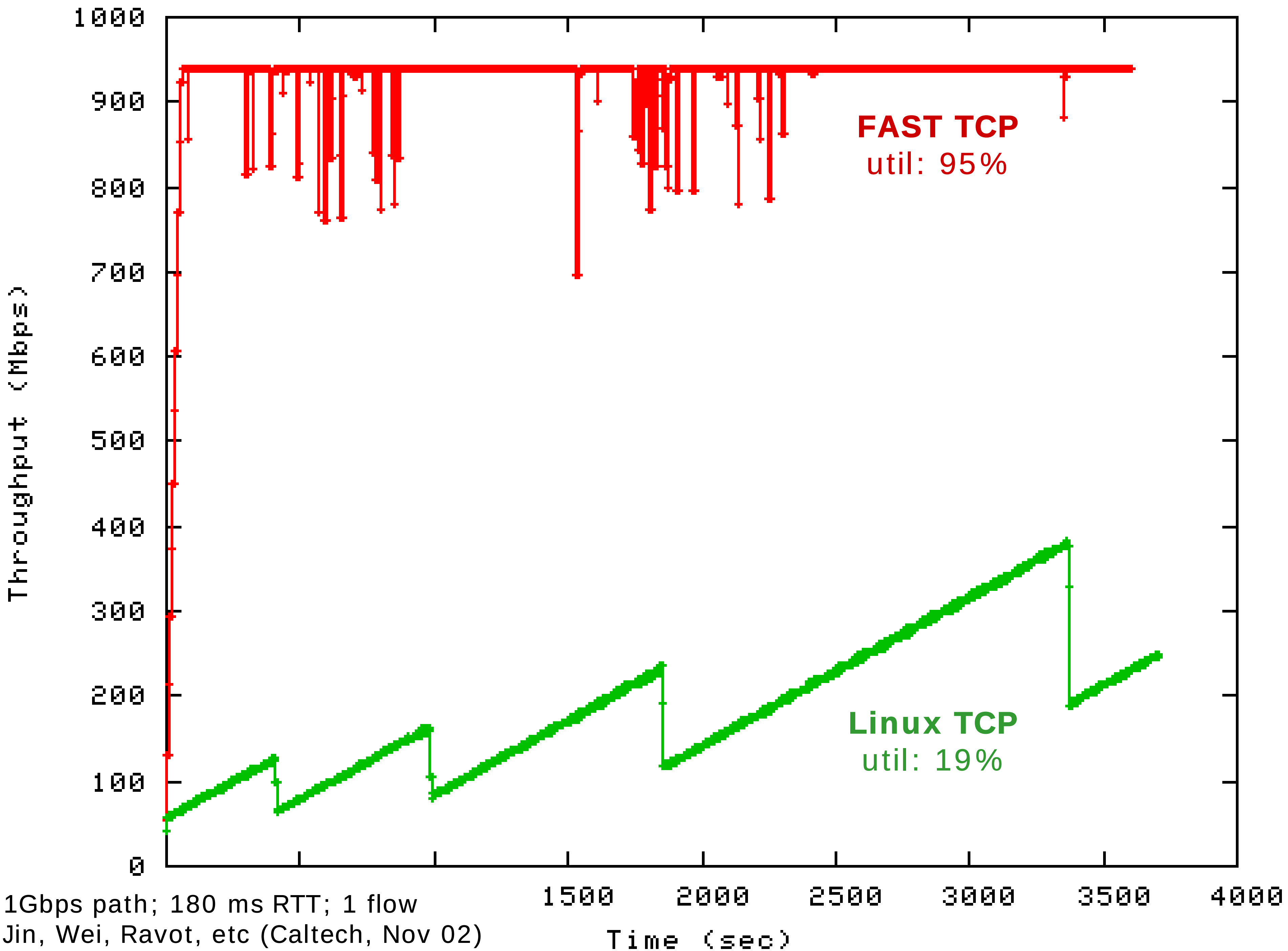
self - clocking : $\sum \frac{w_i(t)}{d_i + q_i(t)} = c_l$

Theorem (CDC04, Infocom05, IMA06)

- Utility function: α_i log x_i (proportional fairness)
- Locally stable in networks if feedback delays are homogeneous (e.g. zero)
- Global exponential convergence over a single link

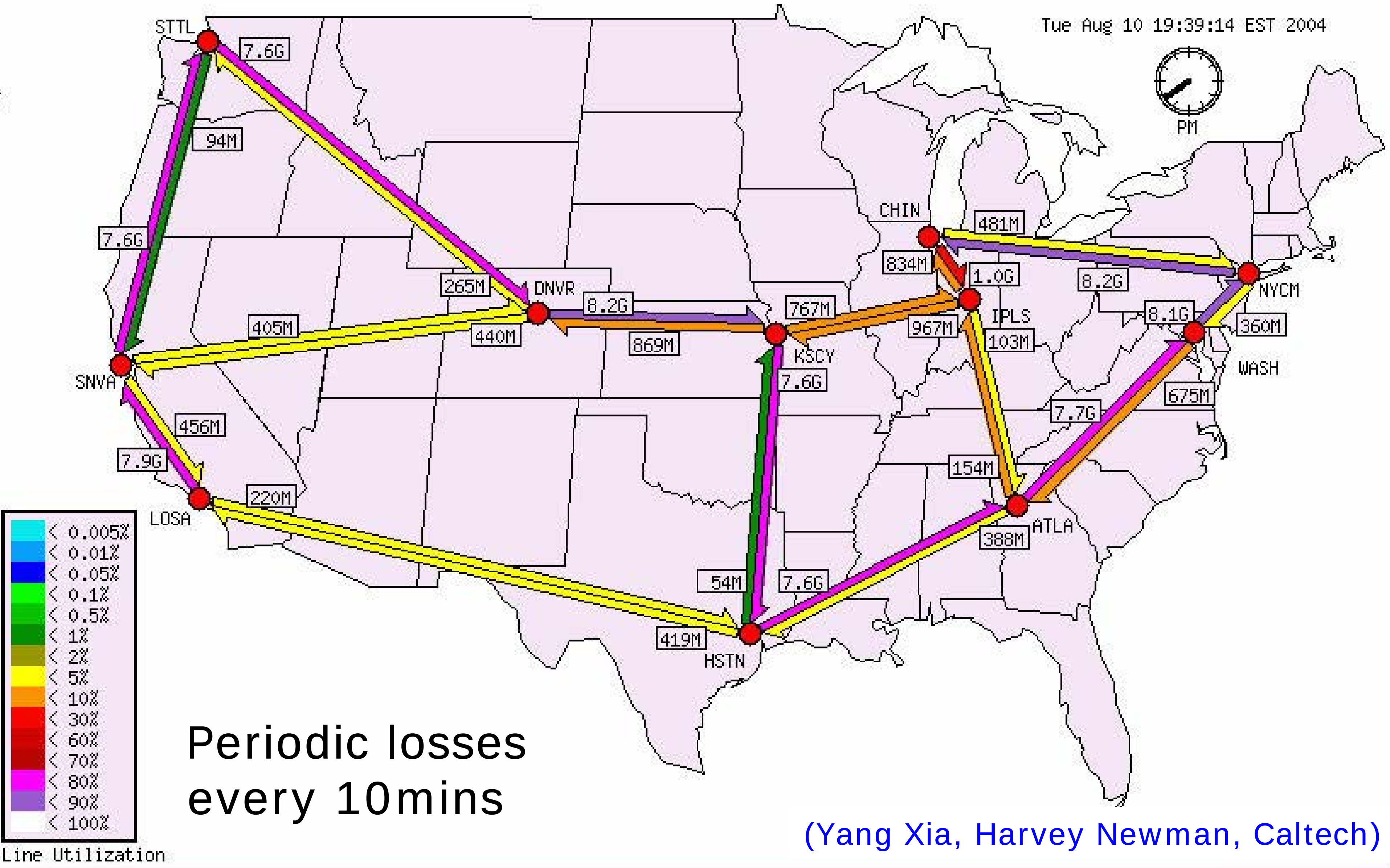
Network

(Sylvain Ravot, caltech/CERN)



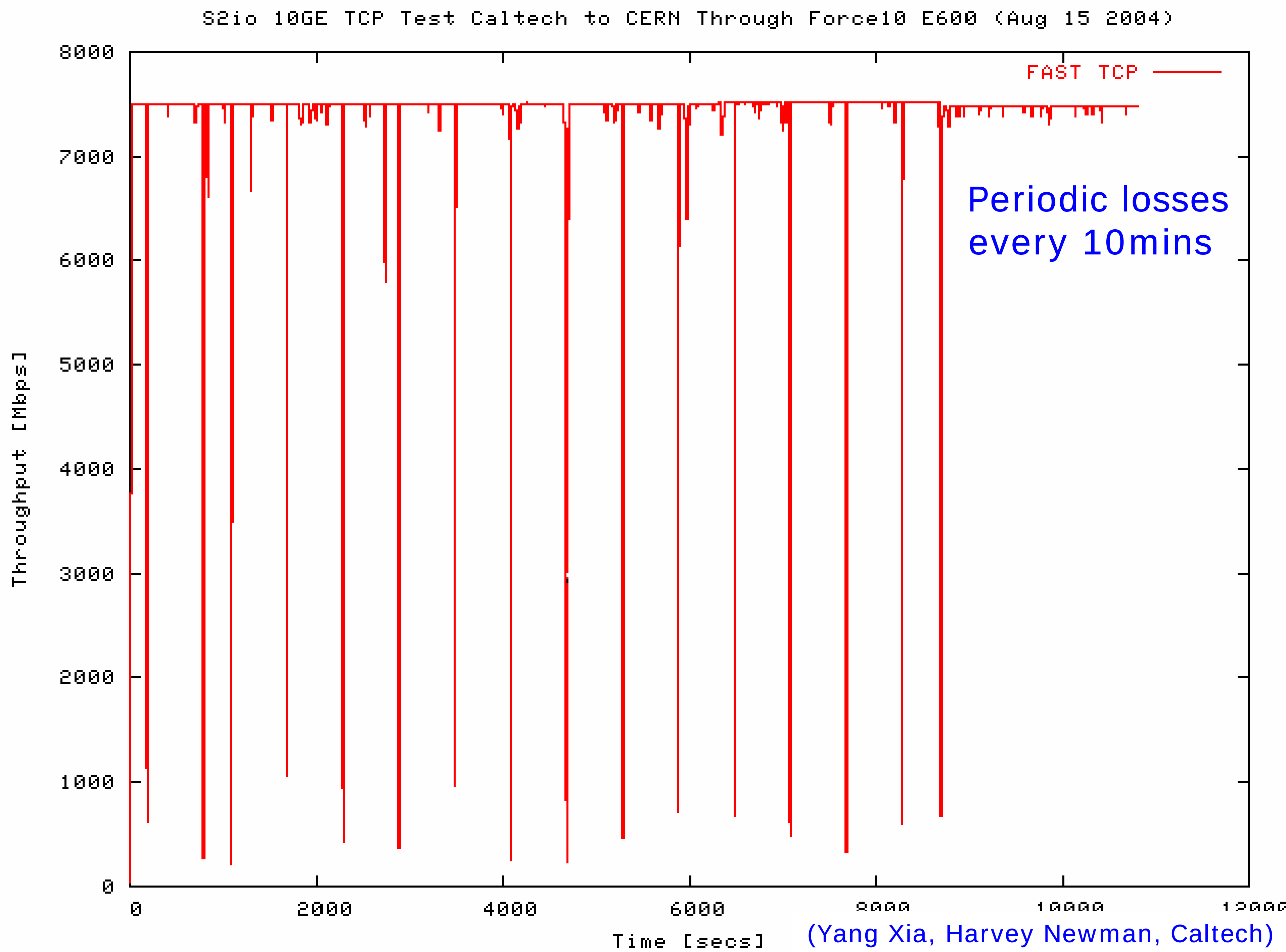
1Gbps path; 180 ms RTT; 1 flow
Jin, Wei, Ravot, etc (Caltech, Nov 02)

INDIANA UNIVERSITY ABILENE NOC WEATHERMAP



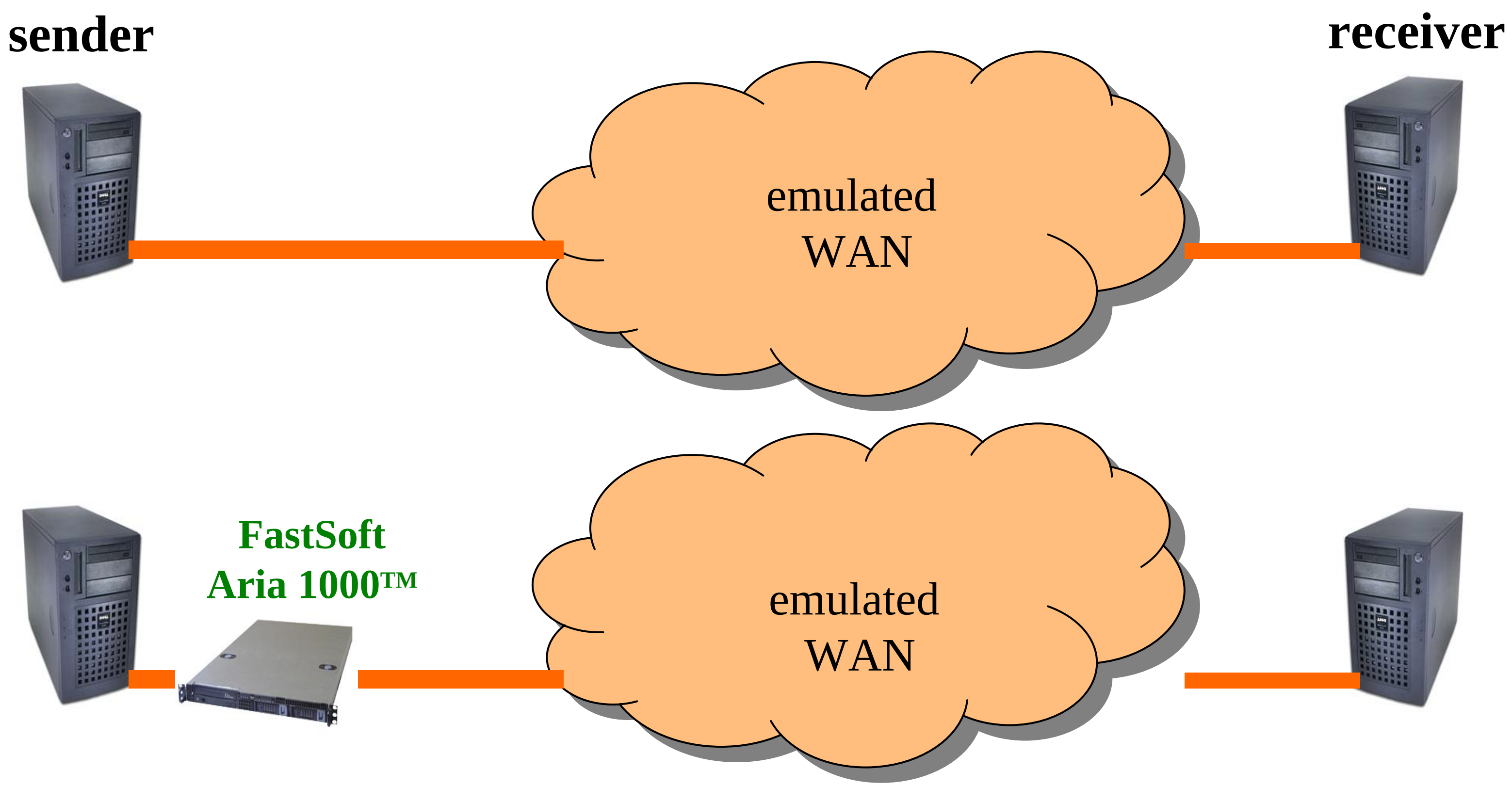
Periodic losses every 10mins

(Yang Xia, Harvey Newman, Caltech)



(Yang Xia, Harvey Newman, Caltech)

Benchmark testbed



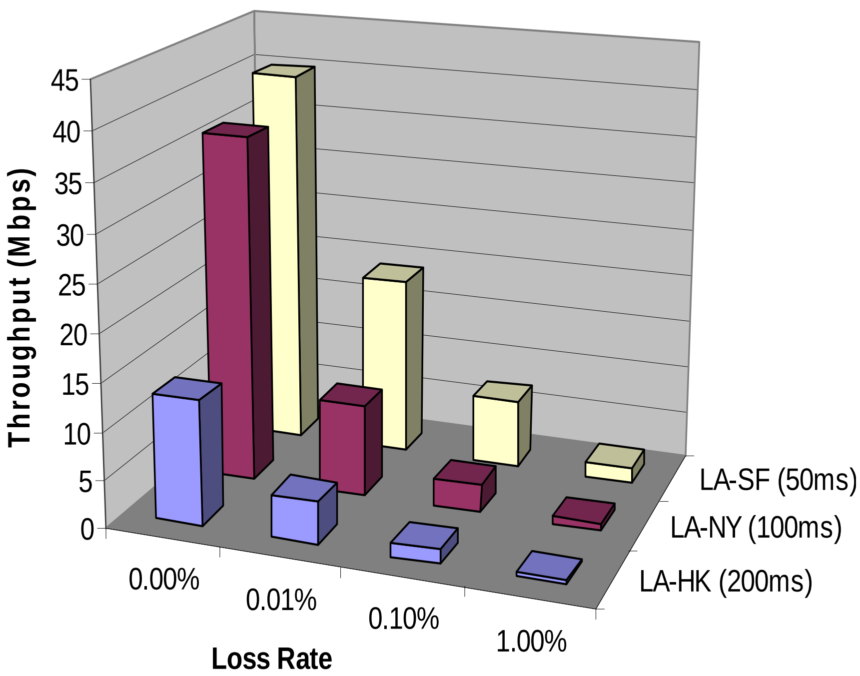
- Emulated WAN emulates different operating conditions
- Measure application throughput (iperf) with and without Aria 1000
- Throughput improvements are similar with Windows 2003 and Linux

Up to 22x improvement on T3

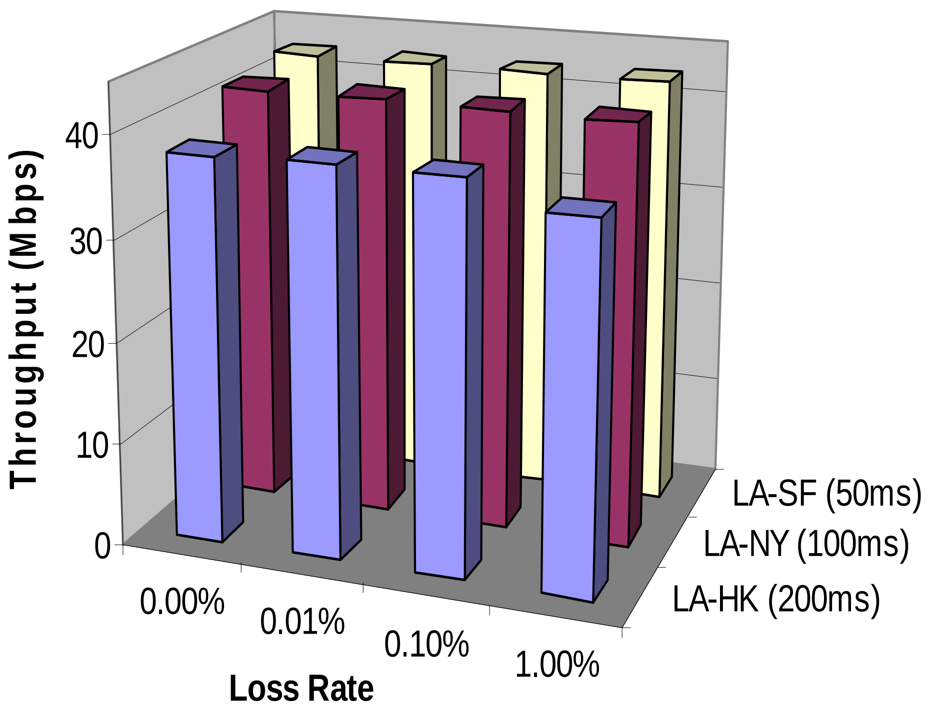
@ 0.1% loss rate

WAN speed: 45Mbps

Linux (tuned)



Linux (default)
+ Aria 1000

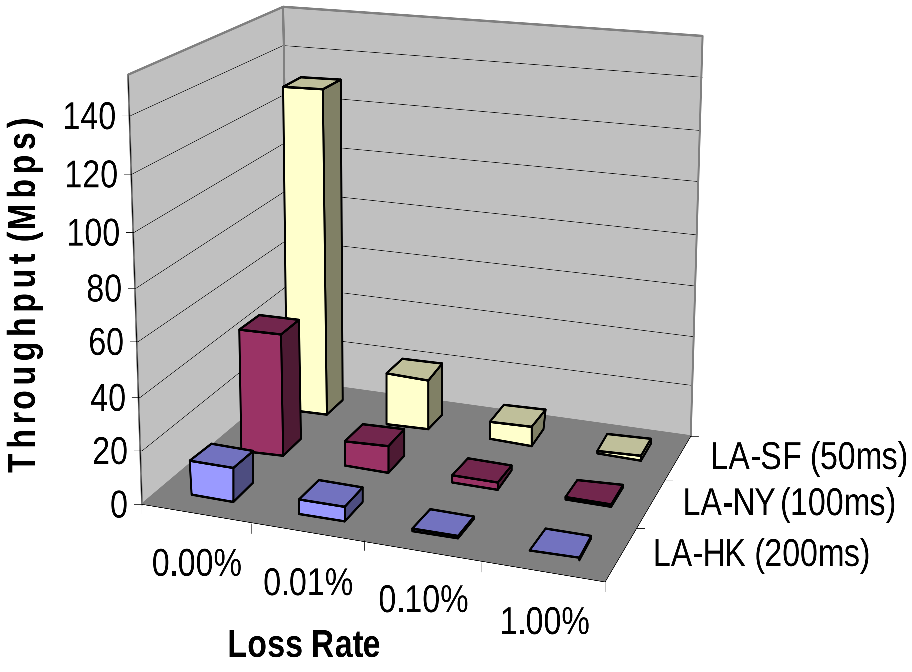


Up to 17x improvement on OC3

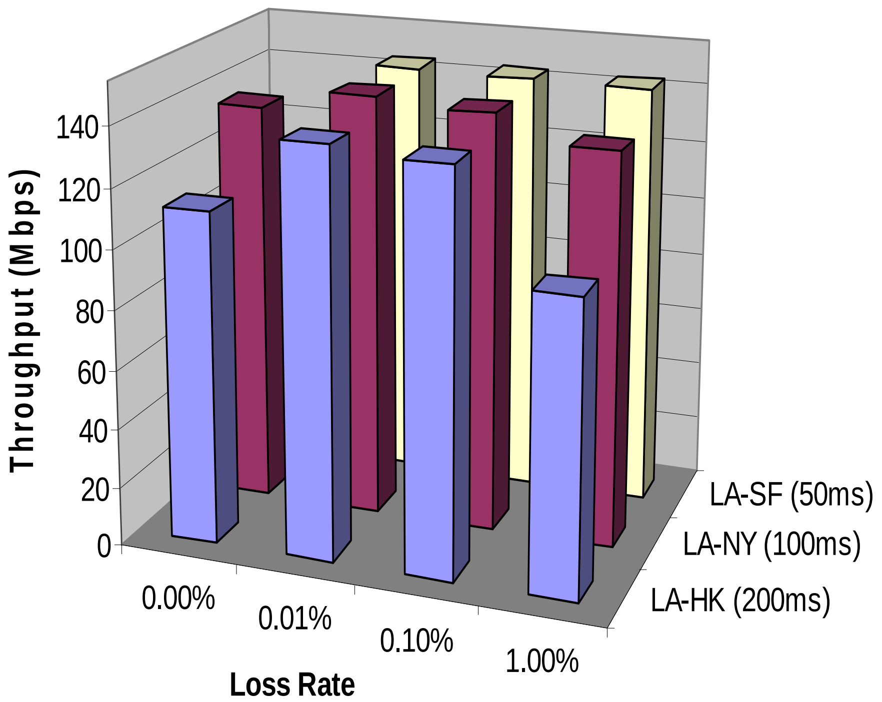
.01% loss rate

WAN speed: 155Mbps

Linux (tuned)



Linux (default)
+ Aria 1000



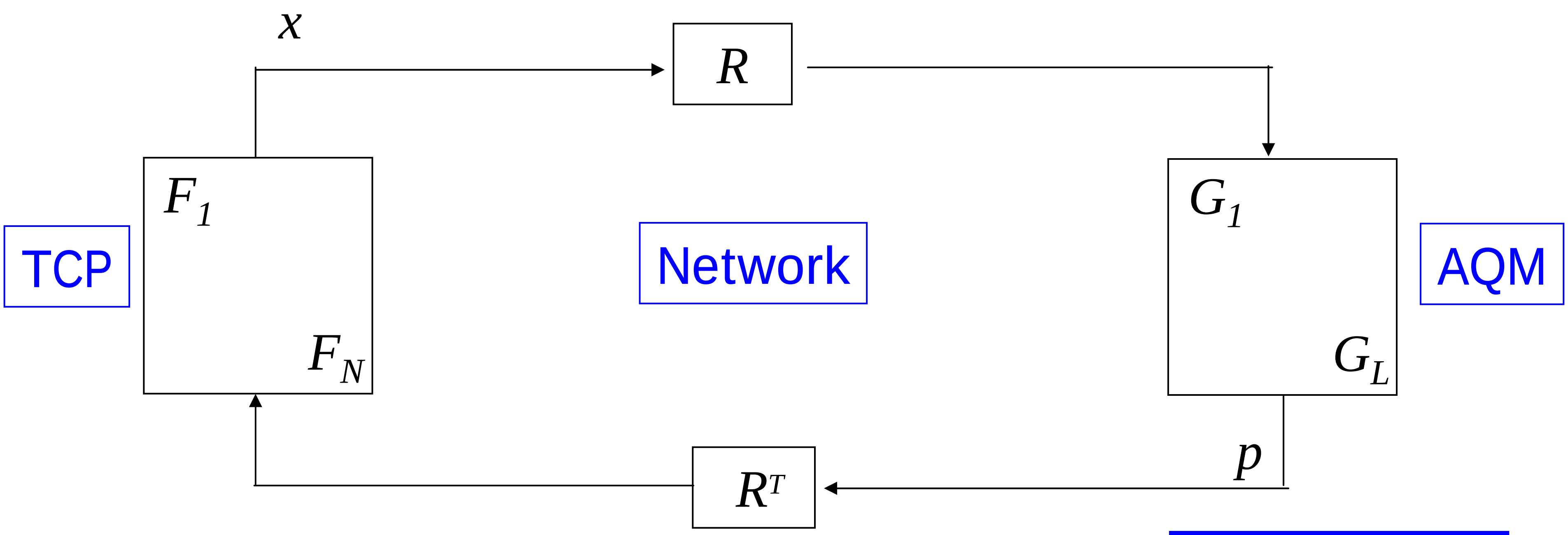
The world is heterogeneous…

- Linux 2.6.13 allows users to choose congestion control algorithms
- Many protocol proposals
  - Loss-based: Reno and a large number of variants
  - Delay-based: CARD (1989), DUAL (1992), Vegas (1995), FAST (2004), ...
  - ECN: RED (1993), REM (2001), PI (2002), AVQ (2003), ...
  - Explicit feedback: MaxNet (2002), XCP (2002), RCP (2005), ...

Some implications

	homogeneous	heterogeneous
equilibrium	unique	non-unique
bandwidth allocation on AQM	independent	dependent
bandwidth allocation on arrival	independent	dependent

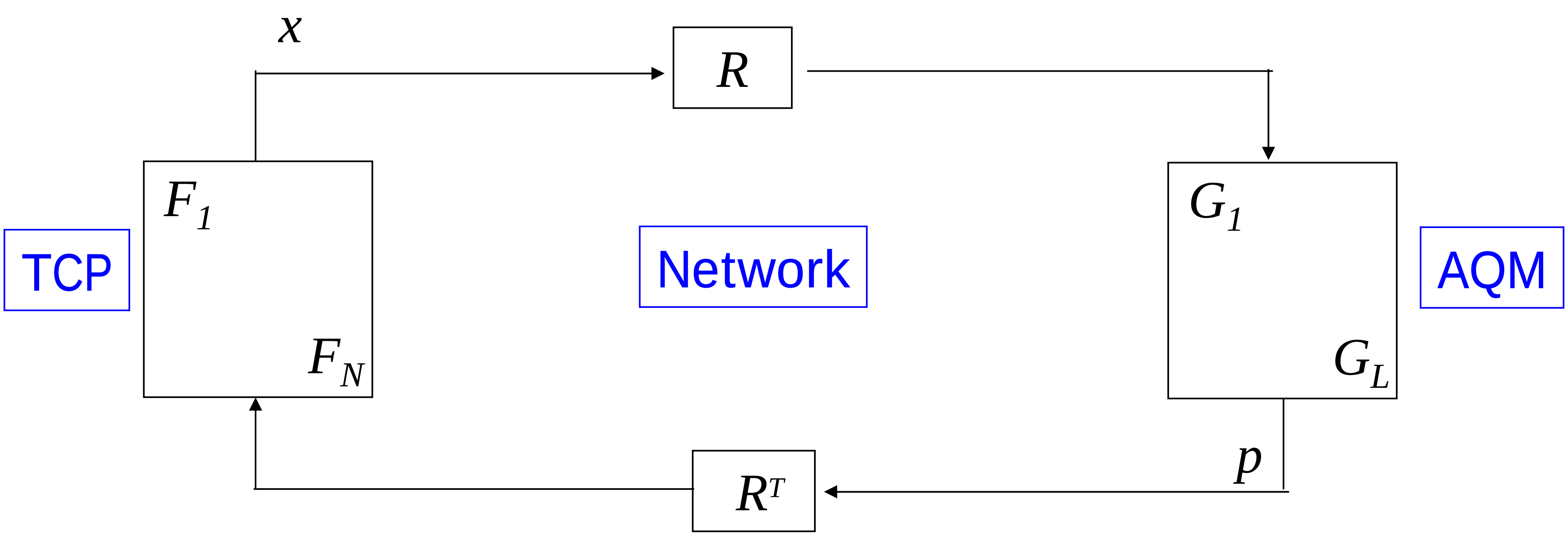
Homogeneous protocol



same price
for all sources

$$x_i(t+1) \;=\; F_i\left(\sum_l R_{li}\, p_l(t),\;\; x_i(t)\right)$$

Heterogeneous protocol



$$x_i(t+1) \;=\; F_i\left(\sum_l R_{li}\, p_l(t),\; x_i(t)\right)$$

$$x_i^j(t+1) \;=\; F_i^j\left(\sum_l R_{li}\, m_l^j\big(p_l(t)\big),\; x_i^j(t)\right)$$

heterogeneous prices for type *j* sources

Heterogeneous protocols

- Equilibrium: p that satisfies

$$x_i^j(p) = f_i^j\left(\sum_l R_{li} m_l^j(p_l)\right)$$

$$y_l(p) := \sum_{i,j} R_{li}^j x_i^j(p) \begin{cases} \le c_l \\ = c_l & \text{if } p_l > 0 \end{cases}$$

- Dynamic: dual algorithm

$$x_i^j(p(t)) = f_i^j\left(\sum_l R_{li} m_l^j(p_l(t))\right)$$

$$\dot{p}_l = \gamma_l\left(y_l(p(t)) - c_l\right)$$

Existence

Theorem

Equilibrium p exists, despite lack of underlying utility maximization

- Generally non-unique
  - There are networks with unique bottleneck set but infinitely many equilibria
  - There are networks with multiple bottleneck set each with a unique (but distinct) equilibrium

Regular networks

Definition

A *regular network* is a tuple (R, c, m, U) for which all equilibria p are locally unique, i.e.,

$$\det \mathbf{J}(p) := \det \frac{\partial y}{\partial p}(p) \neq 0$$

Theorem

- Almost all networks are regular
- A regular network has finitely many and odd number of equilibria (e.g. 1)

Proof: Sard's Theorem and Poincare-Hopf Index Theorem

Global uniqueness

$$\dot{m}_l^j \in [a_l, 2^{1/L} a_l] \quad \text{for any } a_l > 0$$

$$\dot{m}_l^j \in [a^j, 2^{1/L} a^j] \quad \text{for any } a^j > 0$$

Theorem

- If *Degree of heterogeneity* is small, then equilibrium is globally unique

Corollary

- If price mapping functions m_l^j are linear and link-independent, then equilibrium is globally unique

e.g. a network of RED routers almost always has globally unique equilibrium

Local stability: `uniqueness' → stability

$$\dot{m}_l^j \in [a_l, 2^{1/L} a_l] \ \ \text{for any } a_l > 0$$

$$\dot{m}_l^j \in [a^j, 2^{1/L} a^j] \ \ \text{for any } a^j > 0$$

Theorem

- If *Degree of heterogeneity* is small, then the unique equilibrium p is locally stable

Linearized dual algorithm: $\delta\dot{p} = \gamma\, \mathbf{J}(p^*)\ \delta p(t)$

Equilibrium p is *locally stable* if

$$\operatorname{Re} \lambda\big(\mathbf{J}(p)\big) < 0$$

Local stability: `converse'

Theorem

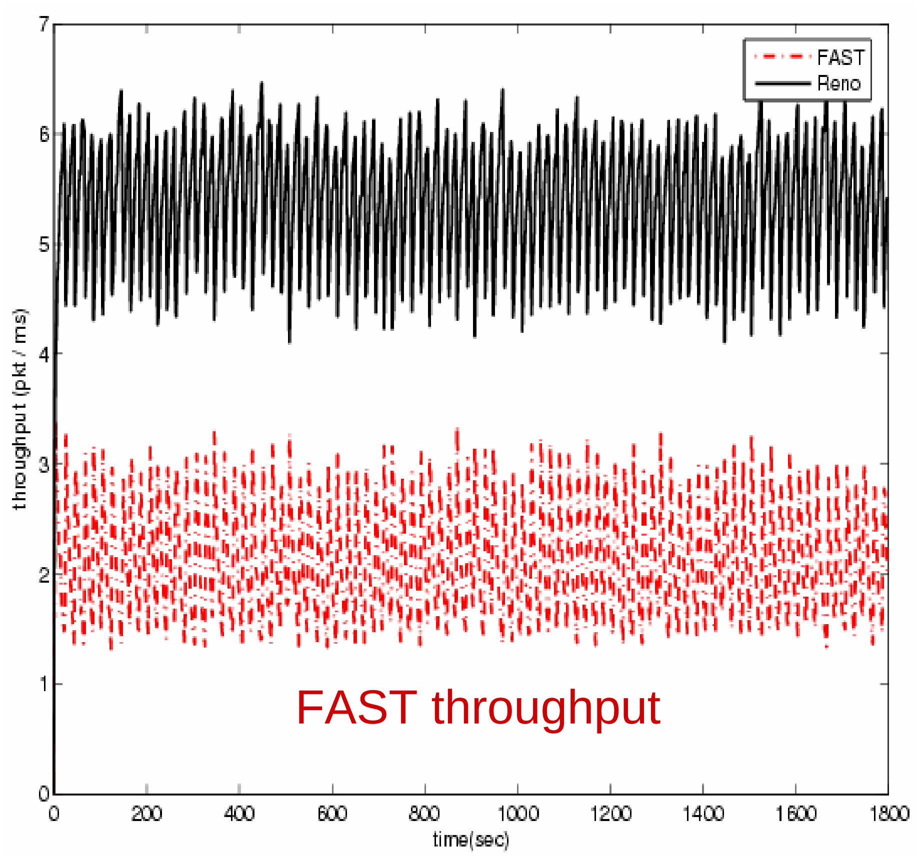
- If all equilibria p are locally stable, then it is globally unique

Proof idea:

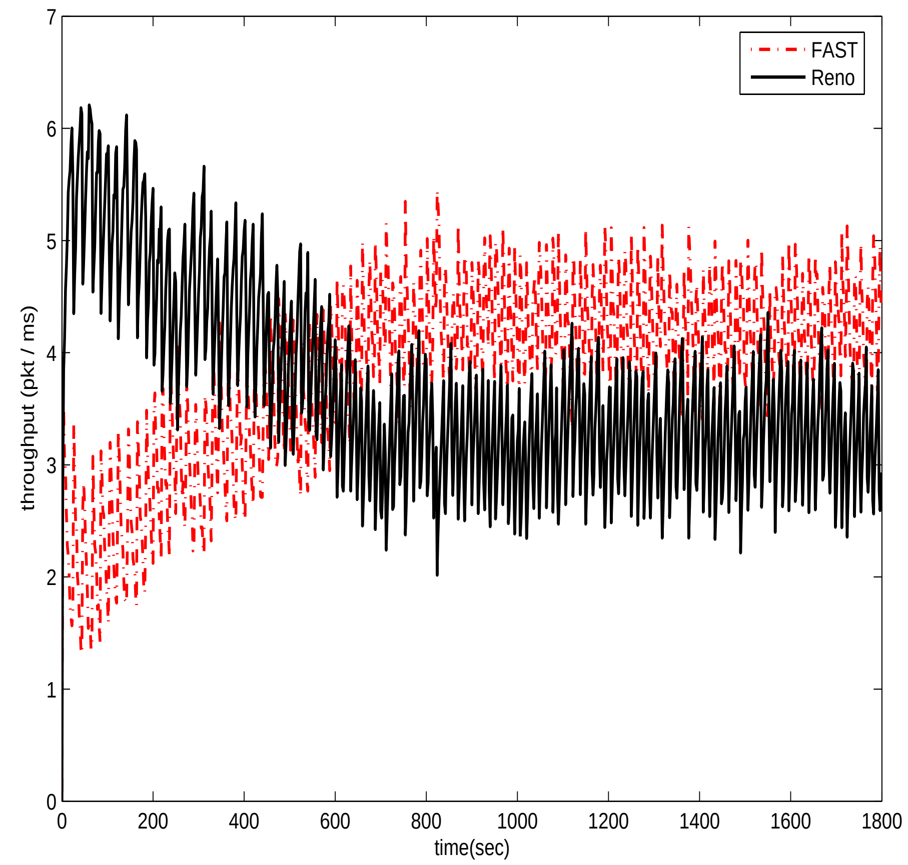
- For all equilibrium p: $I(p) = (-1)^L$
- Index theorem:

$$\sum_{\text{eq } p} I(p) = (-1)^L$$

Forward engineering: ns2 simulation



without slow timescale control



with slow timescale control

Part III

Medium Access Control: Reverse and Forward Engineering

Some References

- J. W. Lee, M. Chiang, and R. A. Calderbank, "Utility-optimal medium access control: reverse and forward engineering," *Proc. IEEE INFOCOM*, April 2006
- X. Wang and K. Kar, "Cross-layer rate control for end-to-end proportional fairness in wireless networks with ranom access", *IEEE Journal of Selected Areas in Communications*, vol. 24, no. 8, August 2006
- J. W. Lee, M. Chiang, and A. R. Calderbank, "Jointly optimal congestion and contention control", *IEEE Communication Letters*, vol. 10, no. 3, pp. 216-218, March 2006

Utility Approach

Two approaches of understanding and designing random access MAC:

- Queuing theoretic: stochastic stability
- Optimization theoretic: utility optimality

Eventually, we want both in a unifying framework

- A lot of results on the first
- This part of the tutorial is about the second

Reverse Engineering

What are heuristics-based protocol designs implicitly solving?

- Layer 4 TCP congestion control: Basic NUM
- Layer 3 IP inter-AS routing: Stable Paths Problem
- Layer 2 MAC backoff contention resolution: Non-cooperative Game

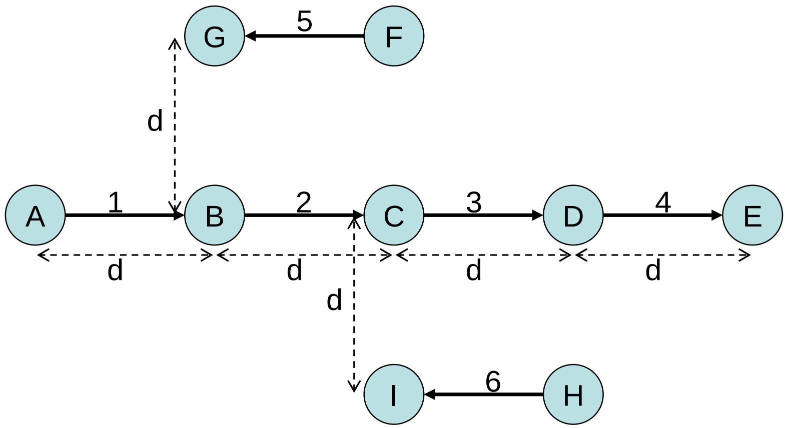
Focus of this part of the talk

Reverse Engineering

Different from imposing a game model:

- MacKenzie Wicker 2003
- Jin Kesidis 2004
- Marbach Yuen 2005
- Altman et. al. 2005

Network Topology



Directed graph $G(V, E)$

$L^I_{to}(l)$: set of links whose transmissions interfere with receiver of link l

$L^I_{from}(l)$: set of links whose transmissions get interferred by transmission on link l

Exponential Backoff MAC

MAC:

- Contention-free: centralized scheduling
- Contention-based: distributed random access (contention avoidance and resolution)

Contention resolution through exponential backoff

Binary Exponential Backoff (BEB) in IEEE 802.11 standard

Persistence probabilistic model:

- Each logical link l transmits with persistence probability p_l
- Successful transmission: $p_l = p_{l,max}$ (*i.e.*, $1/W_l^{min}$)
- Collided transmission: $p_l = \max\{\beta_l p_l, p_l^{min}\}, \;\; \beta_l \in (0, 1)$

It's A Game

TCP/AQM: social welfare (network utility) cooperative maximization

EB-MAC: non-cooperative game

- Coupled utility (due to collision)
- Inadequate feedback

Game: $[E, \prod_{l \in E} A_l, \{U_l\}_{l \in E}]$ with $A_l = \{p_l \mid p_l^{min} \leq p_l \leq p_l^{max}\}$

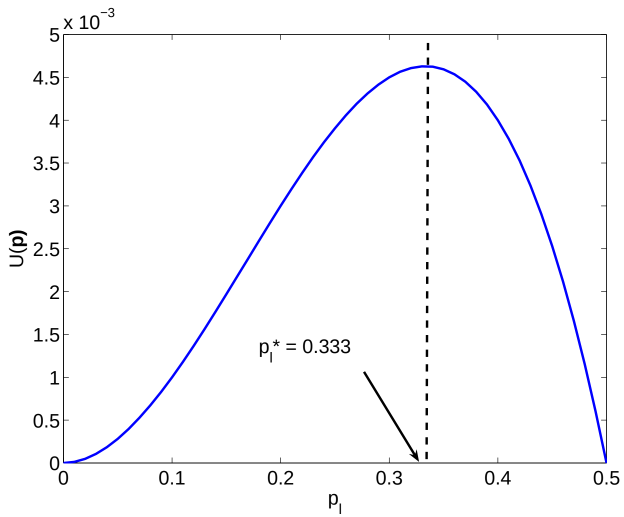
- $S(\mathbf{p}) = p_l \prod_{n \in L_{to}^I(l)} (1 - p_n)$: probability of transmission success
- $F(\mathbf{p}) = p_l (1 - \prod_{n \in L_{to}^I(l)} (1 - p_n))$: probability of collision

Reverse-Engineering Utility Function

Theorem: $U_l(\mathbf{p}) = R(p_l)S(\mathbf{p}) - C(p_l)F(\mathbf{p})$, with

$R(p_l) \stackrel{\text{def}}{=} p_l(\frac{1}{2}p_l^{max} - \frac{1}{3}p_l)$ (reward for transmission success)

$C(p_l) \stackrel{\text{def}}{=} \frac{1}{3}(1-\beta_l)p_l^3$ (cost for collision)



Example: Dependence of a utility function on its own persistence probability, for $\beta_l = 0.5$, $p_l^{max} = 0.5$, and $\prod_{n \in L_{to}^I(l)}(1-p_n) = 0.5$

Existence of NE

Theorem: There exists NE $\mathbf{p}^*$ characterized by:

$$p_l^* = \frac{p_l^{max} \prod_{n \in L_{to}^I(l)} (1 - p_n^*)}{1 - \beta_l (1 - \prod_{n \in L_{to}^I(l)} (1 - p_n^*))}$$

An example of the immediate corollaries that confirms intuition:

• Let $|L_{to}^I(l)| \to \infty$. If $p_l^* > 0$, then only a finite number of links among links in $L_{to}^I(l)$ have a positive persistence probability at NE

Reverse Engineering BEB Protocol

Is it a gradient-based maximization of $U_l(\mathbf{p})$ over p_l?

No, that requires explicit message passing among nodes

Theorem: EB maximizes U_l using stochastic subgradient ascent method (using only local information on success and collision):

$$p_l(t+1) = \max\{p_l^{min}, p_l(t) + v_l(t)\}$$

where

$$v_l(t) = p_l^{max}\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=0\}} + \beta_l p_l(t)\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=1\}} + p_l(t)\mathbf{1}_{\{T_l(t)=0\}} - p_l$$

and

$$\mathrm{E}\{v_l(t)|\mathbf{p}(t)\} = \frac{\partial U_l(\mathbf{p})}{\partial p_l}|_{\mathbf{p}=\mathbf{p}(t)}$$

Uniqueness and Stability of NE

Example: two links, $p_l^{max} = 1$, there are infinite number of NE

Question: under what conditions do we have uniqueness and stability (convergence of best response strategy) of NE?

Best response strategy:

$$p_l^*(t+1) = \underset{p_l^{min} \le p_l \le p_l^{max}}{\operatorname{argmax}} U_l(p_l, \mathbf{p}_{-l}^*(t))$$

Theorem: If $\mathbf{p}^*(0) = \mathbf{p}^{min}$,

$$\mathbf{p}^*(2t+1) \rightarrow \hat{\mathbf{p}} \text{ and } \mathbf{p}^*(2t) \rightarrow \tilde{\mathbf{p}} \text{ as } t \rightarrow \infty.$$

If $\hat{\mathbf{p}} = \tilde{\mathbf{p}}$, $\hat{\mathbf{p}}$ is a NE

Proof: S-modular game theory

Uniqueness and Stability of NE

Assume all links have same $p^{max} < 1$ and $p^{min} = 0$

Let $K = \max_l \{|L^I_{to}(l)|\}$

Uniqueness and stability of NE depend on

- K: amount of potential contention (given)
- β: speed of backoff (variable)
- p^{max}: minimum amount of backoff (variable)

Theorem: $\frac{p^{max} K}{4\beta(1-p^{max})} < 1$ implies uniqueness and global stability of NE

Proof: Contraction mapping verified through bounding infinity matrix norm of Jacobian

From Analysis To Design

Motivates forward-engineering:

- How to introduce limited, local message passing of pricing information to maximize social welfare through selfish utility maximization?
- Can choose utility functions, then design distributed algorithms.
- Prove convergence to global optimum?

Related work:

- Nandagopal et. al. 2000: Conflict graph
- Chen, Low, Doyle 2004: Deterministic approximation
- Kar, Sarkar, Tassiulas 2004: Proportional fair case

Problem Formulation

Nonconvex and coupled generalized NUM:

$$
\begin{array}{ll}
\text{maximize} & \sum_{l \in L} U_l(x_l) \\
\text{subject to} & x_l = c_l p_l \prod_{k \in N_{to}^I(l)} (1 - P^k), \ \ \forall l \\
& \sum_{l \in L_{out}(n)} p_l = P^n, \ \ \forall n \\
& x_l^{min} \leq x_l \leq x_l^{max}, \ \ \forall l \\
& 0 \leq P^n \leq 1, \ \ \forall n \\
& 0 \leq p_l \leq 1, \ \ \forall l \\
\text{variables} & \{x_l\}, \{p_l\}, \{P^n\}
\end{array}
$$

Algorithm Development

Three main steps:

- Reveal hidden decomposability: log change of variable

- Condition for convexity: application needs to be sufficiently elastic:

Utility function's curvature needs to be not just negative but bounded away from 0 by as much as $-\frac{dU_l^x(x_l)}{x_l dx_l}$

- Standard dual decomposition and distributed subgradient method

Utility-Optimal Random Access

Algorithm 1 (message passing from transmitters):

1: Each node n constructs its local interference graph to obtain sets $L_{out}(n)$, $L_{in}(n)$, $L^I_{from}(n)$, and $N^I_{to}(l)$, $\forall l \in L_{out}(n)$.

2: Each node n sets $t = 0$, $\lambda_l(1) = 1$, $\forall l \in L_{out}(n)$, $P^n(1) = \frac{|L_{out}(n)|}{|L_{out}(n)|+|L^I_{from}(n)|}$, and $p_l(1) = \frac{1}{|L_{out}(n)|+|L^I_{from}(n)|}$, $\forall l \in L_{out}(n)$.

3: Locally at each node n, repeat:

3.1: Set $t \leftarrow t + 1$.

3.2: Inform contention prices $\lambda_l(t)$ to nodes in $N^I_{to}(l)$, $\forall l \in L_{out}(n)$, and contention probability $P^n(t)$ to t_l, $\forall l \in L^I_{from}(n)$.

3.3: Set $k_n(t) = \sum_{l \in L_{out}(n)} \lambda_l(t) + \sum_{k \in L^I_{from}(n)} \lambda_k(t)$ and $\alpha(t) = \frac{1}{t}$.

3.4: Compute the following:

$$P^n(t+1) = \begin{cases} \frac{\sum_{l \in L_{out}(n)} \lambda_l(t)}{\sum_{l \in L_{out}(n)} \lambda_l(t) + \sum_{k \in L^I_{from}(n)} \lambda_k(t)}, & \text{if } k_n(t) \neq 0 \\ \frac{|L_{out}(n)|}{|L_{out}(n)| + |L^I_{from}(n)|}, & \text{if } k_n(t) = 0 \end{cases},$$

$$p_l(t+1) = \begin{cases} \frac{\lambda_l(t)}{\sum_{l \in L_{out}(n)} \lambda_l(t) + \sum_{k \in L^I_{from}(n)} \lambda_k(t)}, & \text{if } k_n(t) \neq 0 \\ \frac{1}{|L_{out}(n)| + |L^I_{from}(n)|}, & \text{if } k_n(t) = 0 \end{cases},$$

$$x'_l(t+1) = \underset{x'^{min}_l \leq x' \leq x'^{max}_l}{\operatorname{argmax}} \left\{ U'_l(x'_l) - \lambda_l(t) x'_l \right\},$$

and

$$\lambda_l(t+1) = \left[\lambda_l(t) - \alpha(t) \left(c'_l + \log p_l(t) + \sum_{k \in N^I_{to}(l)} \log\left(1 - P^k(t)\right) - x'_l(t) \right) \right].$$

3.5: Each node n decides if it will transmit data with a probability $P^n(t)$. If it decides to transmit data, it chooses to transmit on one of its outgoing links with probability $p_l(t)/P^n(t)$, $\forall l \in L_{out}(n)$.

Properties

- Theorem: convergence to global optimum
- Theory accurate predicts performance (unlike deterministic approx.)
- Efficiency-fairness flexible tradeoff

Extensions

- Variant: receiver based message passing (Algorithm 2)

From two-hop message passing to one-hop

- Quantification of message passing overhead
- Message passing reduction heuristics

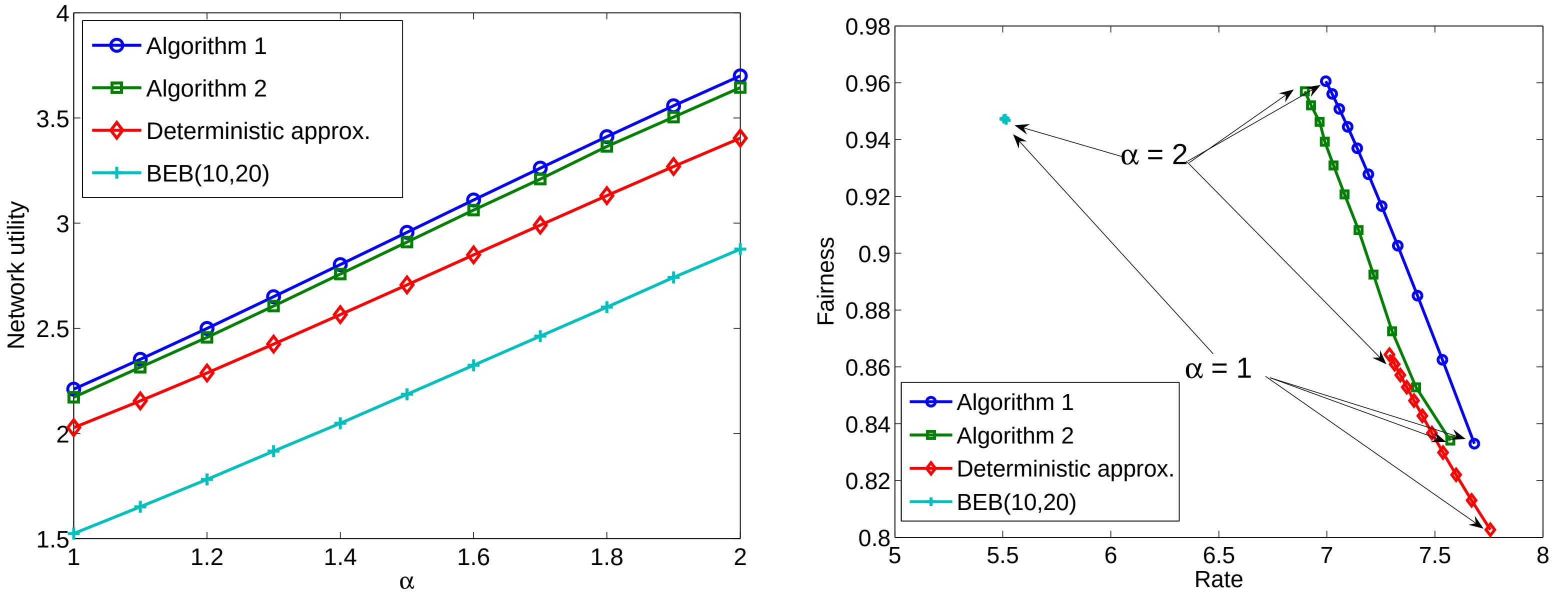
No need to pass P^n

Piggyback λ_l values to messages in Algorithm 2

Leverage broadcast property in wireless

- Robustness to time delays and outdated messages

Example



Algorithm 1
Algorithm 2
Deterministic approx.
BEB(10,20)
Network utility
α
4
3.5
3
2.5
2
1.5
1
1.2
1.4
1.6
1.8
2
Fairness
Rate
0.98
0.96
0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
5
5.5
6
6.5
7
7.5
8
α = 2
α = 1
Algorithm 1
Algorithm 2
Deterministic approx.
BEB(10,20)

Related Results

Reverse engineering:

- Convergence of stochastic subgradient

- Relationship between stochastic subgradient and best response dynamics

Forward engineering:

- Jointly optimal congestion and contention control

Part IV

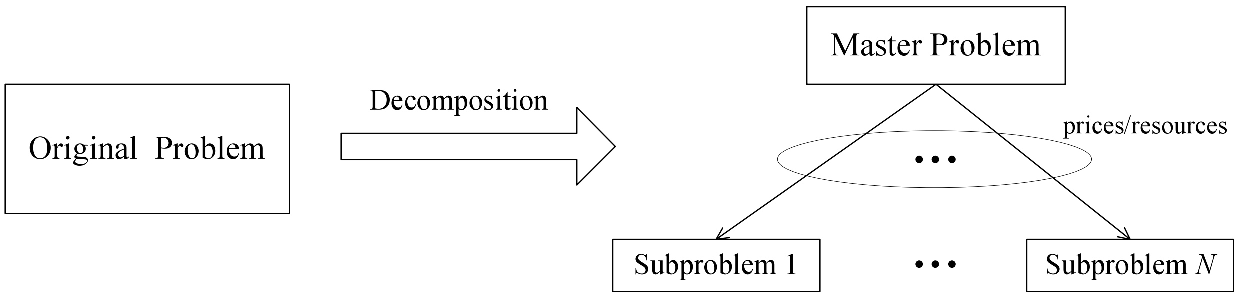
Decomposition Theory and Alternative Decompositions

Some References

- D. Palomar and M. Chiang, "A tutorial to decomposition methods for network utility maximization", *IEEE Journal of Selected Areas in Communications*, vol. 24, no. 8, August 2006

- D. Palomar and M. Chiang, "Alternative decompositions for distributed maximization of network utility: Framework and applications", *Proc. IEEE INFOCOM*, April 2006

Decomposition



Decompose a problem into subproblems coordinated by a master problem

Key idea in modularization and distributed control

Mathematical machineries available, but far from complete

Decomposition Theory

- Convexity: efficient solution to global optimality
- Decomposability: distributed algorithm

The two concepts are different

Related in the sense that convexity often leads to zero duality gap, thus allowing dual decomposition

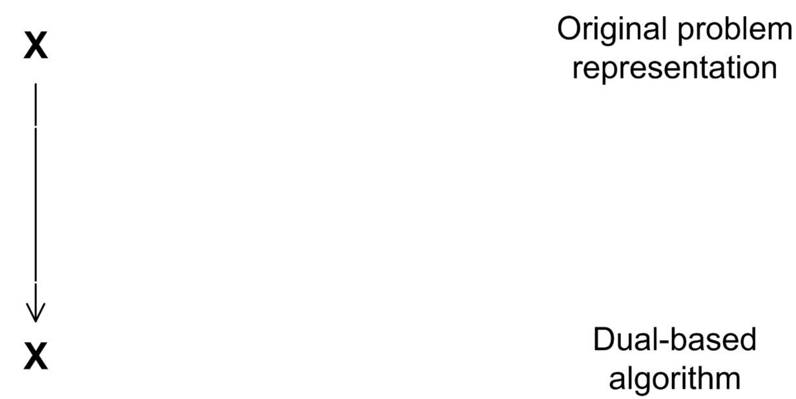
- No universally-agreed and concise definition of Decomposability
- Can be somewhat quantified by the amount of explicit and implicit message passing needed (and the growth order as the number of links, nodes, and processes increase)

Motivation

X

Original problem
representation

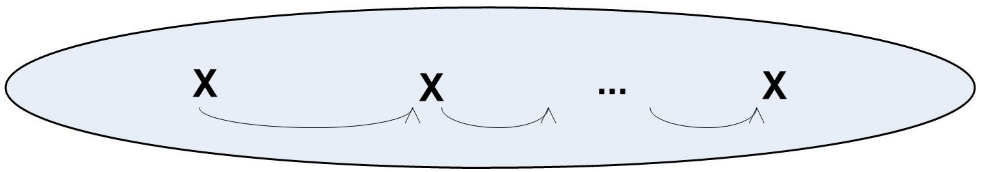
Motivation



Motivation

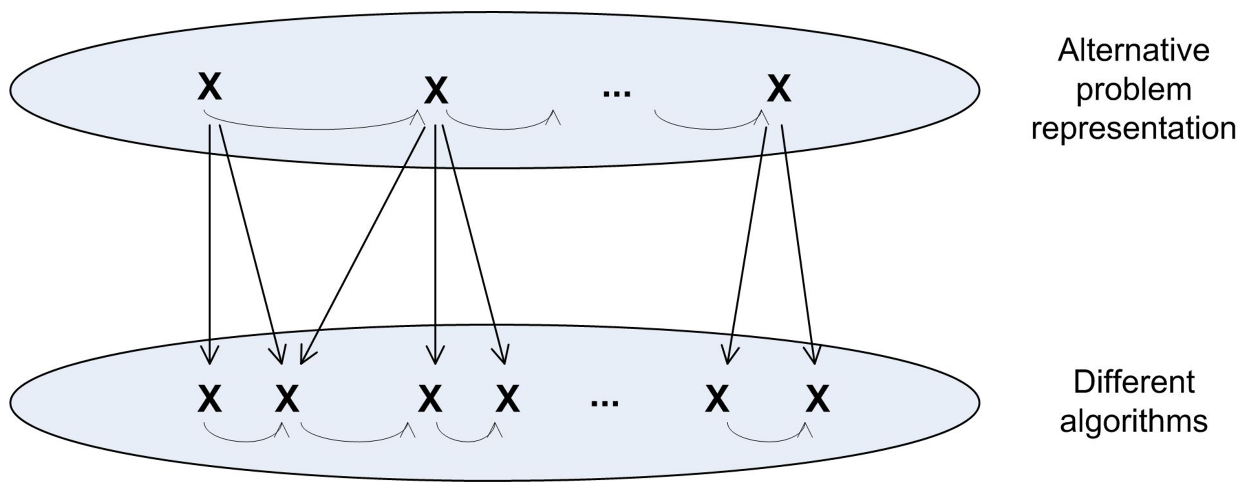
X	Original problem representation
↓	
X	Dual-based algorithm
↓	
□	Engineering implication

Motivation



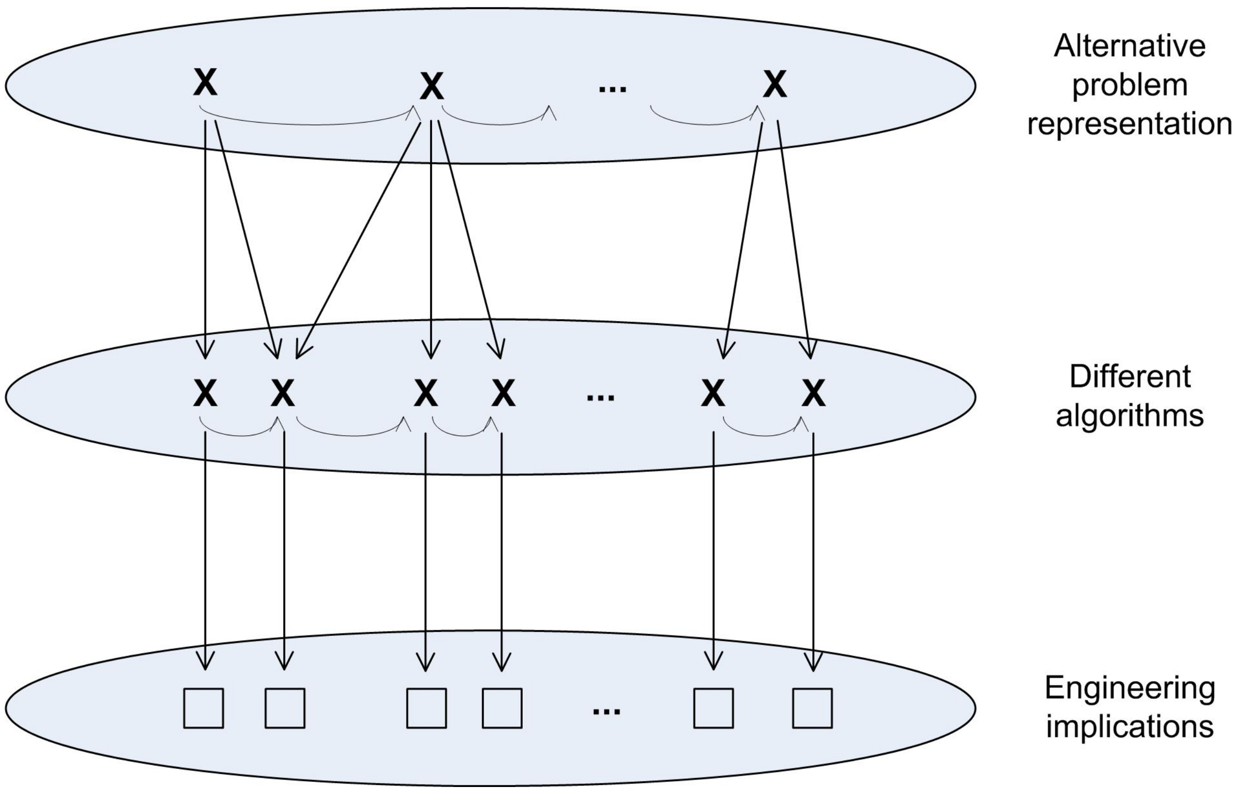
Alternative
problem
representation

Motivation



X
X
...
X
Alternative problem representation
X
X
X
X
...
X
X
Different algorithms

Motivation


X
X
...
X
Alternative problem representation
X
X
X
X
...
X
X
Different algorithms
...
Engineering implications

Building Blocks

- Primal/Dual decompositions
- Indirect decomposition
- Partial decomposition
- Multilevel/recursive decomposition

- Order of updates: sequential or parallel
- Timescale of updates: iterative or one-shot
- Timescale separation for multilevel decomposition

A variety of combinations from the above building blocks

Example: standard dual algorithm for BNUM is a direct, single-level, full, dual decomposition

Choices of Decomposition

But there can be many choices of alternative decompositions and alternative distributed algorithms for GNUM

Standard dual decomposition is not always the best

Even a different representation of GNUM can lead to a different set of choices of distributed algorithms

Three stages of development:

1. Layering can be understood as decomposition

2. Search through alternative decompositions

3. General methods to exhaust and compare the possibilities

Stages 1 and 2 are done, but not Stage 3

Comparing Decomposition

Metrics (sometimes competing):

- Amount and symmetry of message passing
- Amount and symmetry of local computation
- Speed of convergence (if iterative)
- Robustness (under signaling error, stochastic perturbance, failure of nodes)
- Ease and robustness of parameter tuning

Some are hard to be quantified/characterized or a full ordering relationship

Some tradeoffs require application-specific pick among Pareto-optimal choices of alternative decomposition

Decomposition Techniques: Dual Decomp.

- The dual of the following convex problem (with coupling constraint)

$$\begin{array}{lll} \underset{\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i(\mathbf{x}_i) & \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i & \forall i, \\ & \sum_i \mathbf{h}_i(\mathbf{x}_i) \leq \mathbf{c} & \end{array}$$

  is decomposed into subproblems:

$$\begin{array}{ll} \underset{\mathbf{x}_i}{\text{maximize}} & f_i(\mathbf{x}_i) - \boldsymbol{\lambda}^T \mathbf{h}_i(\mathbf{x}_i) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i. \end{array}$$

  and the master problem

$$\underset{\boldsymbol{\lambda} \geq \mathbf{0}}{\text{minimize}} \quad g(\boldsymbol{\lambda}) = \sum_i g_i(\boldsymbol{\lambda}) + \boldsymbol{\lambda}^T \mathbf{c}$$

  where $g_i(\boldsymbol{\lambda})$ is the optimal value of the ith subproblem.

Decomposition Techniques: Primal Decomp.

- The following convex problem (with coupling variable $\mathbf{y}$)

$$\begin{array}{ll} \underset{\mathbf{y},\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i(\mathbf{x}_i) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i \qquad \forall i \\ & \mathbf{A}_i \mathbf{x}_i \leq \mathbf{y} \\ & \mathbf{y} \in \mathcal{Y} \end{array}$$

is decomposed into the subproblems:

$$\begin{array}{ll} \underset{\mathbf{x}_i \in \mathcal{X}_i}{\text{maximize}} & f_i(\mathbf{x}_i) \\ \text{subject to} & \mathbf{A}_i \mathbf{x}_i \leq \mathbf{y} \end{array}$$

and the master problem

$$\underset{\mathbf{y} \in \mathcal{Y}}{\text{maximize}} \quad \sum_i f_i^\star(\mathbf{y})$$

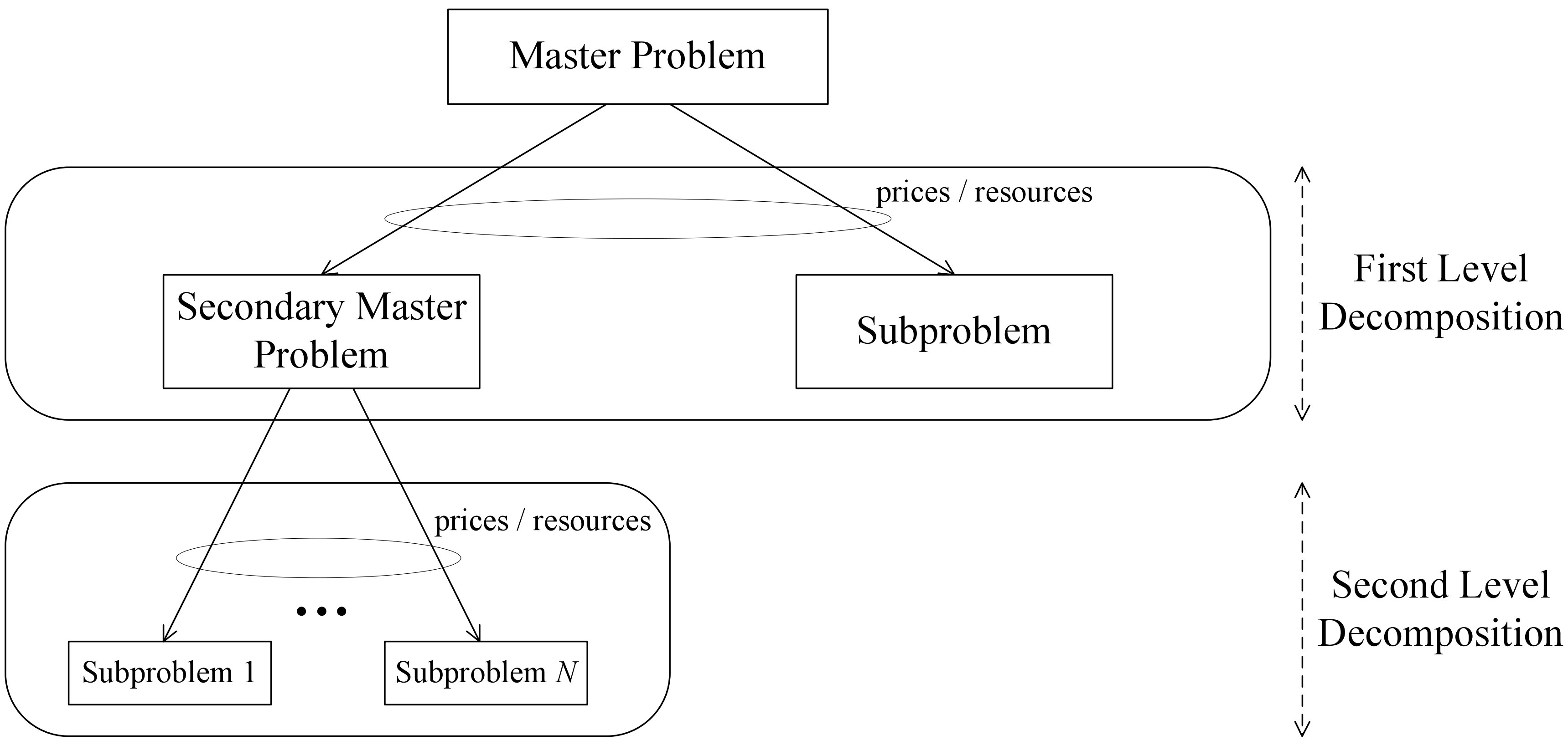
where $f_i^\star(\mathbf{y})$ is the optimal value of the ith subproblem.

Indirect Primal/Dual Decompositions

- Different problem structures are more suited for primal or dual decomposition.
- We can change the structure and use either a primal or dual decomposition for the same problem.
- Key ingredient: introduction of auxiliary variables.
- This will lead to different algorithms for same problem.

Multilevel Primal/Dual Decompositions

- Hierarchical and recursive application of primal/dual decompositions to obtain smaller and smaller subproblems:



Applic. 2: Cellular Downlink Power-Rate Control (I)

- Problem:

$$\begin{array}{ll} \underset{\{r_i, p_i\}}{\text{maximize}} & \sum_i U_i(r_i) \\ \text{subject to} & r_i \leq \log(g_i p_i) \qquad \forall i \\ & p_i \geq 0 \\ & \sum_i p_i \leq P_T. \end{array}$$

- Decompositions: i) primal, ii) partial dual, iii) full dual.

- Many variants of full dual decomposition: the master problem is

$$\underset{\boldsymbol{\lambda} \geq \mathbf{0}, \gamma \geq 0}{\text{minimize}} \quad g(\boldsymbol{\lambda}, \gamma)$$

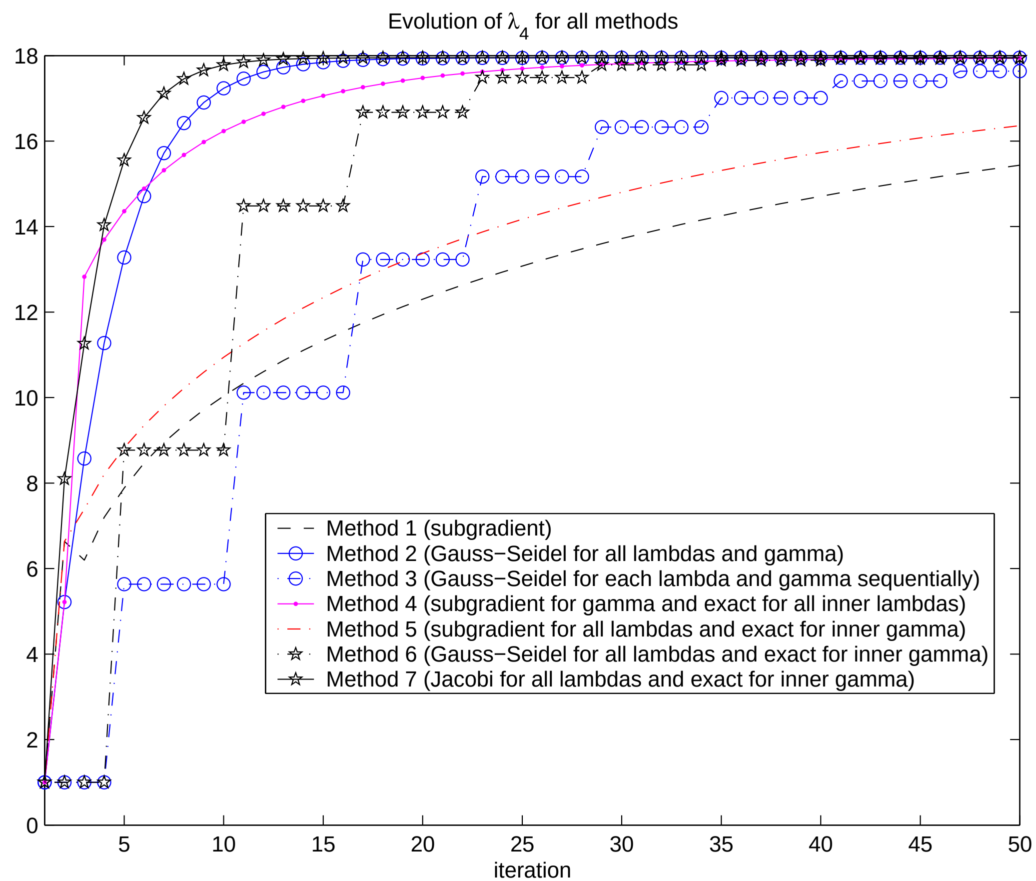
and can e solved as listed next.

Applic. 2: Cellular DL Power-Rate Control (II)

1. Direct subgradient update of $\gamma(t)$ and $\boldsymbol{\lambda}(t)$
2. Gauss-Seidel method for $g(\boldsymbol{\lambda}, \gamma)$: $\boldsymbol{\lambda} \to \gamma \to \boldsymbol{\lambda} \to \gamma \to \cdots$
3. Similar to 2), but optimizing $\lambda_1 \to \gamma \to \lambda_2 \to \gamma \to \cdots$
4. Additional primal decomp.: minimize $g(\gamma) = \inf_{\boldsymbol{\lambda} \geq \mathbf{0}} g(\boldsymbol{\lambda}, \gamma)$
5. Similar to 4), but changing the order of minimization
6. Similarly to 5), but with yet another level of decomposition: minimize $g(\boldsymbol{\lambda})$ sequentially (Gauss-Seidel fashion)
7. Similar to 5) and 6), but minimizing $g(\boldsymbol{\lambda})$ with in a Jacobi fashion

Applic. 2: Cellular DL Power-Rate Control (III)

- Downlink power/rate control problem with 6 nodes with utilities with utilities $U_i(r_i) = \beta_i \log r_i$. Evolution of λ_4 for all 7 methods:



Part V

Case 1: Joint Congestion Control and Coding

Some References

- J. W. Lee, M. Chiang, and A. R. Calderbank, "Price-based distributed algorithm for optimal rate-reliability tradeoff in network utility maximization", *IEEE Journal of Selected Areas in Communications*, vol. 24, no. 5, pp. 962-976, May 2006

- M. Chiang, "Balancing transport and physical layer in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE Journal of Selected Areas in Communications,* vol. 23, no. 1, pp. 104-116, January 2005

Signal Reliability

Application needs at sources : utility of signal reliability

Physical layer possibilities on links: adaptive coding/modulation

Intuition of the new opportunity:

- Link tradeoff: Fatter pipe, lower reliability
- Source tradeoff: Higher rate, lower quality

Signal quality and physical layer entirely missing from basic NUM

Problem Formulation

Assumptions: decode and reencode with small error probabilities

Reliability: R_s for source s

Code rate: $r_{l,s}$ on link l for source s

Error probability as a function of code rate: $E_l(r_{l,s})$

$$
\begin{array}{ll}
\text{maximize} & \sum_s U_s(x_s, R_s) \\
\text{subject to} & R_s = 1 - \sum_{l \in L(s)} E_l(r_{l,s}), \ \ \forall s \\
 & \sum_{s \in S(l)} \frac{x_s}{r_{l,s}} \leq C_l^{max}, \ \ \forall l \\
 & x_s^{min} \leq x_s \leq x_s^{max}, \ \ \forall s \\
 & 0 \leq r_{l,s} \leq 1, \ \ \forall l, s \\
\text{variables} & \mathbf{x}, \mathbf{R}, \mathbf{r}
\end{array}
$$

Overview

Difficulty: Neither convex nor separable problem

Goal: Derive globally optimal and distributed algorithm

- Develop such algorithms
- Extend pricing interpretation
- Sufficient conditions for convergence to global optimum
- Techniques to tackle nonconvexity and nonseparability issues

Integrated Policy

Each link maintains the same code rate for all sources traversing it

maximize	$\sum_s U_s(x_s, R_s)$
subject to	$R_s \leq 1 - \sum_{l \in L(s)} E_l(r_l), \ \ \forall s$
	$\sum_{s \in S(l)} \frac{x_s}{r_l} \leq C_l^{max}, \ \ \forall l$
variables	$\mathbf{x}, \mathbf{R}, \mathbf{r}$

Naturally decompose:

$$\sum_{s \in S(l)} x_s \leq C_l^{max} r_l, \ \forall l$$

Difficulty 1: Nonconvexity

Approximation of $E_l(r_l)$:

$$
\begin{aligned}
p_l &\leq \exp(-NE_0(r_l)) \\
E_0(r_l) &= \max_{0\leq\rho\leq 1} \max_{\mathbf{Q}} [E_o(\rho,\mathbf{Q}) - \rho r_l] \\
E_o(\rho,\mathbf{Q}) &= -\log \sum_{j=0}^{J-1} \left[\sum_{k=0}^{K-1} Q(k) P(j|k)^{1/(1+\rho)} \right]^{1+\rho}
\end{aligned}
$$

Lemma:

If absolute value of first derivatives of $E_0(r_l)$: bounded away from 0,

Absolute value of second derivative of $E_0(r_l)$: upper bounded,

Then for a large enough codeword block length N, $E_l(r_l)$ is a convex function

Standard Methodology

Next: Use standard Lagrangian relaxation and distributed subgradient algorithm to develop distributed algorithm

Distributed Algorithm 1

Source problem and reliability price update at source s:

- Source problem:

$$\begin{array}{ll} \text{maximize} & U_s(x_s, R_s) - \lambda^s(t)x_s - \mu_s(t)R_s \\ \text{subject to} & x_s^{min} \leq x_s \leq x_s^{max} \end{array}$$

where $\lambda^s(t) = \sum_{l \in L(s)} \lambda_l(t)$ is the end-to-end congestion price at iteration t

- Reliability price update:

$$\mu_s(t+1) = \left[\mu_s(t) - \alpha(t)\left(R^s(t) - R_s(t)\right)\right]^+$$

where $R^s(t) = 1 - \sum_{l \in L(s)} E_l(r_l(t))$ is the end-to-end reliability at iteration t

Link problem and congestion price update at link l:

- Link problem:

$$\begin{array}{ll} \text{maximize} & \lambda_l(t) r_l C_l^{max} - \mu^l(t) E_l(r_l) \\ \text{subject to} & 0 \leq r_l \leq 1 \end{array}$$

where $\mu^l(t) = \sum_{s \in S(l)} \mu_s(t)$ is the aggregate reliability price paid by sources using link l at iteration t

- Congestion price update:

$$\lambda_l(t+1) = \left[\lambda_l(t) - \alpha(t) \left(r_l(t) C_l^{max} - x^l(t) \right)\right]^+$$

where $x^l(t) = \sum_{s \in S(l)} x_s(t)$ is the aggregate information rate on link l at iteration t

Pricing Interpretation

- Source problem: maximize total net utility on

rate (with total congestion price) and

reliability (with signal quality price)

- Source algorithm:

local solution of source problem (2 variables)

updates signal quality price

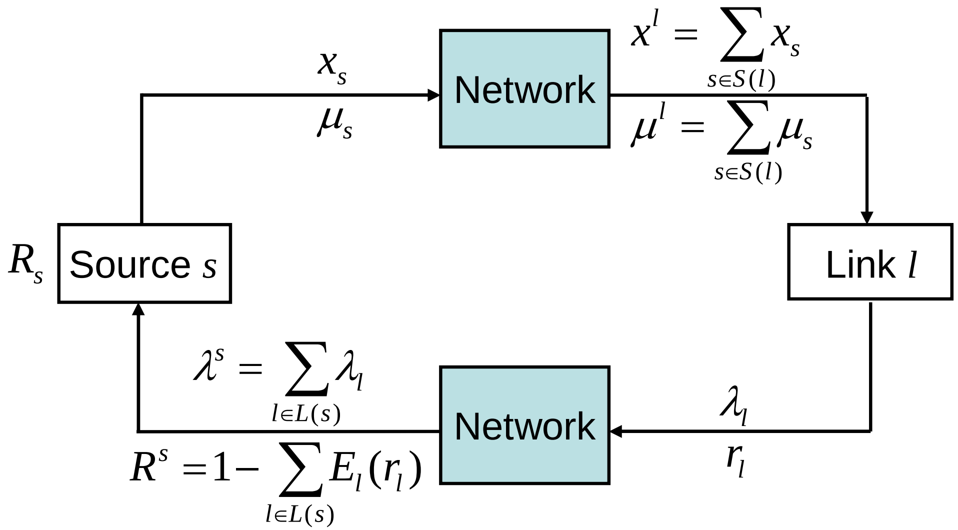
- Network problem: maximize net revenue:

receive revenue from rate

pay price for unreliability

- Link algorithm: update link congestion price

Distributed Algorithm 1



Theorem: Distributed Algorithm 1 converges to the globally optimal rate-reliability tradeoff for sufficiently strong codes

Differentiated Policy

Each link may give a different code rate for each of the sources traversing it

- Per-flow state needed
- Better rate-reliability tradeoff

$$
\begin{array}{ll}
\text{maximize} & \sum_s U_s(x_s, R_s) \\
\text{subject to} & R_s \leq 1 - \sum_{l \in L(s)} E_l(r_{l,s}), \ \ \forall s \\
& \sum_{s \in S(l)} \frac{x_s}{r_{l,s}} \leq C_l^{max}, \ \ \forall l \\
\text{variables} & \mathbf{x}, \mathbf{R}, \mathbf{r}
\end{array}
$$

Difficulty 2: Coupling

Step 1: Introduce auxiliary variables (a new scheduling layer):

$$
\begin{array}{ll}
\text{maximize} & \sum_s U_s(x_s, R_s) \\
\text{subject to} & R_s \le 1 - \sum_{l \in L(s)} E_l(r_{l,s}), \ \ \forall s \\
 & \frac{x_s}{r_{l,s}} \le c_{l,s}, \ \ \forall l, \ \ s \in S(l) \\
 & \sum_{s \in S(l)} c_{l,s} \le C_l^{max}, \ \ \forall l
\end{array}
$$

Step 2: Log change of variables (on $\mathbf{x}$):

$$
\begin{array}{ll}
\text{maximize} & \sum_s U'_s(x'_s, R_s) \\
\text{subject to} & R_s \le 1 - \sum_{l \in L(s)} E_l(r_{l,s}), \ \ \forall s \\
 & x'_s - \log r_{l,s} \le \log c_{l,s}, \ \ \forall l, \ \ s \in S(l) \\
 & \sum_{s \in S(l)} c_{l,s} \le C_l^{max}, \ \ \forall l
\end{array}
$$

Separable problem but $U'_s(x'_s, R_s)$ may not be concave

Difficulty 2: Coupling

Step 3: Concavity condition

$$
\begin{aligned}
g_s(x_s, R_s) &= \frac{\partial^2 U_s(x_s, R_s)}{\partial x_s^2} x_s + \frac{\partial U_s(x_s)}{\partial x_s}, \\
h_s(x_s, R_s) &= \Bigg(\left(\frac{\partial^2 U_s(x_s, R_s)}{\partial x_s \partial R_s} \right)^2 \\
&\quad - \frac{\partial^2 U_s(x_s, R_s)}{\partial x_s^2} \frac{\partial^2 U_s(x_s, R_s)}{\partial R_s^2} \Bigg) x_s \\
&\quad - \frac{\partial^2 U_s(x_s, R_s)}{\partial R_s^2} \frac{\partial U_s(x_s, R_s)}{\partial x_s},
\end{aligned}
$$

and

$$
q_s(x_s, R_s) = \frac{\partial^2 U_s(x_s, R_s)}{\partial R_s^2}.
$$

Lemma: If $g_s(x_s, R_s) < 0$, $h_s(x_s, R_s) < 0$, and $q_s(x_s, R_s) < 0$, then $U_s'(x_s', R_s)$ is a concave function of x_s' and R_s

Difficulty 2: Coupling

Special case 1: α-fair utilities

$$U_s(x_s, R_s) = \begin{cases} \log x_s R_s, & \text{if } \alpha = 1 \\ (1-\alpha)^{-1}(x_s R_s)^{1-\alpha}, & \text{otherwise} \end{cases}$$

If $\alpha \geq 1$, conditions for concavity is satisfied

Special case 2: if U_s is additive in x_s and R_s, its curvature needs to be not just negative but bounded away from 0 by as much as $-\frac{dU_s^x(x_s)}{x_s dx_s}$

The application traffic is sufficiently elastic

Distributed Algorithm 2

All descriptions same as in Algorithm 1 except one:

- Link problems:

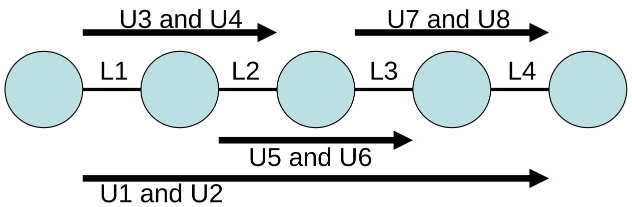
Bandwidth allocation problem

$$\begin{array}{ll} \text{maximize} & \sum_{s\in S(l)} \lambda_{l,s}(t)\log c_{l,s} \\ \text{subject to} & \sum_{s\in S(l)} c_{l,s} \leq C_l^{max} \end{array}$$

Code rate allocation problem for source s, $s \in S(l)$

$$\begin{array}{ll} \text{maximize} & \lambda_{l,s}(t)\log r_{l,s} - \mu_s(t)E_l(r_{l,s}) \\ \text{subject to} & 0 \leq r_{l,s} \leq 1 \end{array}$$

Numerical Example

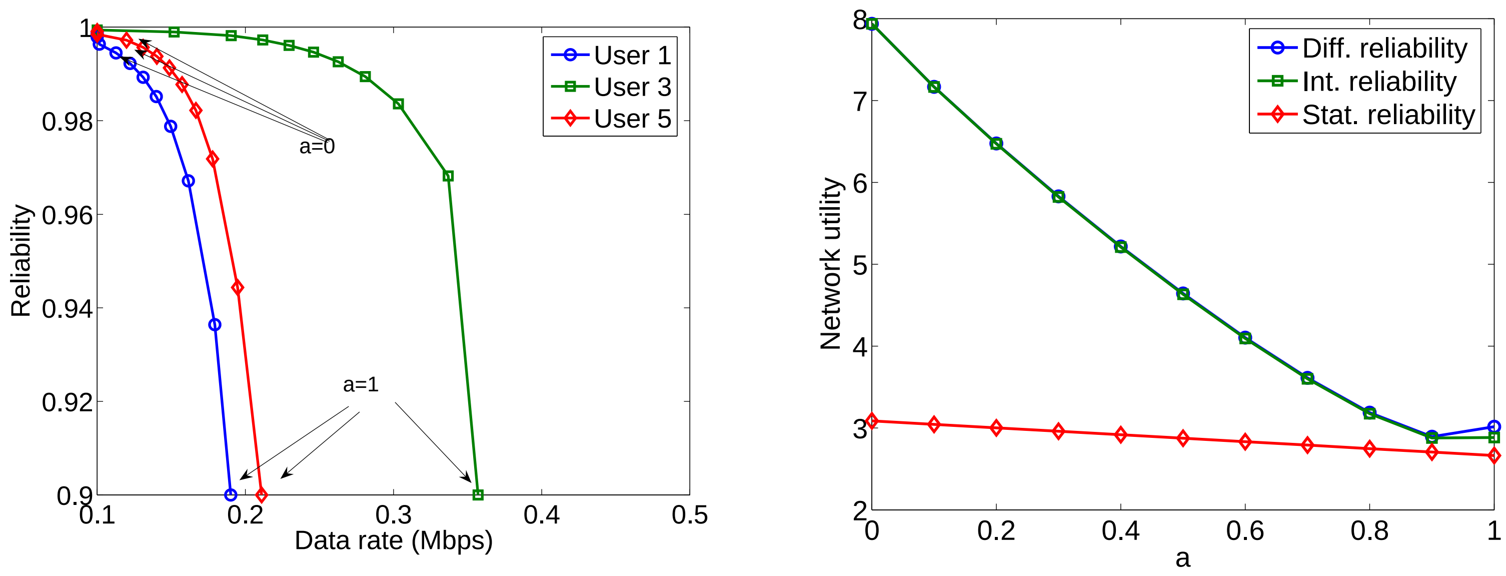


$$U_s(x_s, R_s) = a_s \frac{x_s^{1-\alpha} - x_s^{min(1-\alpha)}}{x_s^{max(1-\alpha)} - x_s^{min(1-\alpha)}} + (1 - a_s) \frac{R_s^{(1-\alpha)} - R_s^{min(1-\alpha)}}{R_s^{max(1-\alpha)} - R_s^{min(1-\alpha)}}$$

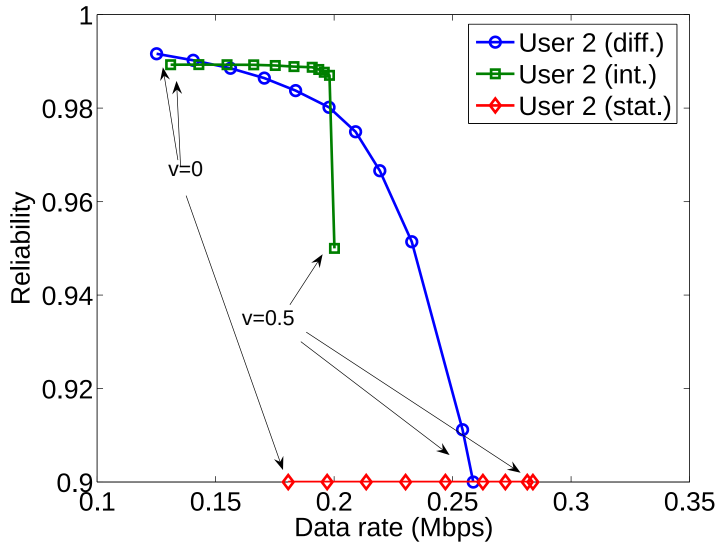
Example 1: $a_s = a$

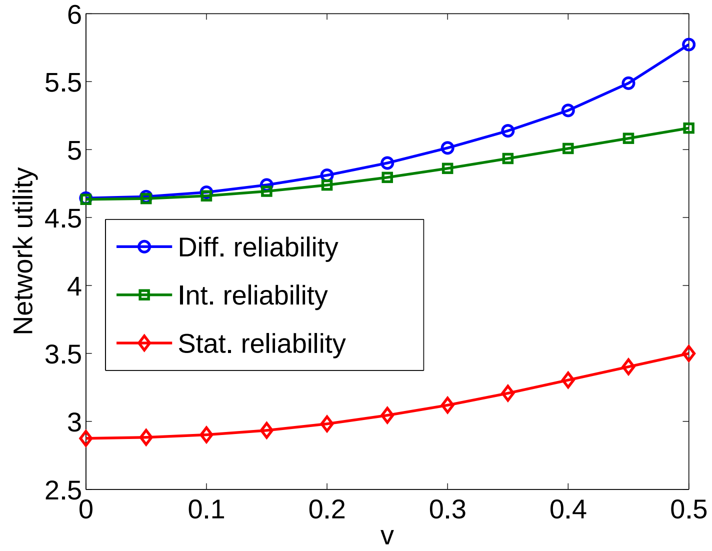
Example 2: $a_s = \begin{cases} 0.5 - v, & \text{if } s \text{ is an odd number} \\ 0.5 + v, & \text{if } s \text{ is an even number,} \end{cases}$

Numerical Example 1



Numerical Example 2





Extensions

Partial Dynamic Reliability Policy

- Only some links can adjust error correction capability
- DSL links in end-to-end path
- Substantial gain still observed.

Wireless MIMO rate reliability control

- Multiplexing gain vs. diversity gain
- Ignore multi-point joint coding
- Sufficiently high SNR (for convexity)
- Same mathematical structures as before

Key Messages and Methods

- Convexity of the generalized NUM is the key to devising a globally optimal solution.

Conditions on constraints and utility curvature for convexity to hold

- Decomposability of the generalized NUM is the key to devising a distributed solution.

Introducing new "layer" and log change of variable to reveal hidden decomposability structure

- User-generated pricing following end-to-end principle

Part VI

Case 2: Joint Congestion Control, Routing, and Scheduling

Some References

- L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Joint optimal congestion control, routing, and scheduling in wireless ad hoc networks," *Proc. IEEE INFOCOM*, April 2006

- A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," *Proc. IEEE INFOCOM*, March 2005

- X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," *Proc. IEEE INFOCOM,* March 2005

- M. Neely, E. Modiano, and C. Rohrs, "Dynamic power control and routing over time varying wireless networks", *IEEE Journal of Selected Areas in Communications*, vol. 23, no. 1, pp. 89-103, January 2005

- A. L. Stolyar, "Maximizing queueing network utility subject to statbility: greedy primal-dual algorithm", *Queueing Systems*, vol. 50, no. 4, pp. 401-457, 2005

Network Model

- Wireless network: a set of N nodes and a set of L logical links
- Each link $l \in L$ has fixed capacity c_l when active

- Primary interference: links that share common node cannot transmit simultaneously

Schedulability

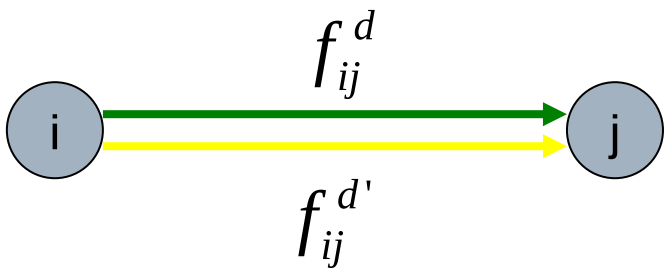
- **Independent set**: links that can transmit simultaneously
- An independent set e is represented by an L-dim rate vector r^e:

$$r_l^e = \begin{cases} c_l & \text{if} \quad l \in e \\ 0 & \text{otherwise} \end{cases}$$

- **Feasible rate region** is:

$$\boxed{\Pi = \left\{ r : r = \sum_e a_e r^e, a_e \geq 0, \sum_e a^e = 1 \right\}}$$

Schedulability constraint



$$f_{ij} := \sum_{d} f_{ij}^{d}$$

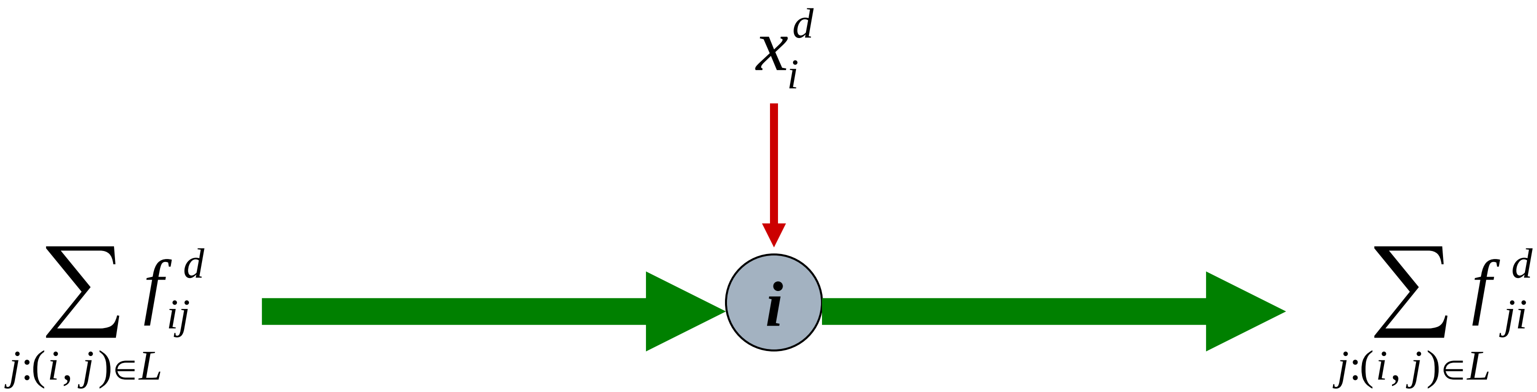
$$f := \left(f_{ij}, \forall \text{ links } (i, j)\right)$$

- f_{ij}^{d} : capacity of link (i, j) allocated to flow with destination d
- f_{ij} : capacity of link (i, j) allocated to all flows traversing link (i, j)
- Schedulability constraint:

$$f \in \Pi$$

Rate constraint

external flow
entering at node *i*



capacity allocated to
incoming transit flows

capacity allocated to
all outgoing flows

$$x_i^d + \sum_j f_{ji}^d \le \sum_j f_{ij}^d \qquad \text{for all } i \in N, d \in D$$

Problem formulation

- Generalized NUM:

$$\max_{x_{id},\, f_{ij}^{d}} \quad \sum_{i,d} U_{id}(x_i^d)$$

$$s.t. \quad x_i^d \le \sum_j f_{ij}^d - \sum_j f_{ji}^d, \quad i \ne d$$

$$f \in \Pi$$

Dual decomposition

- Dual decomposition:

$$D(p) = \max_{x_{id}, f_{ij}^d} \sum_{i,d} U_{id}(x_i^d) - \sum_{i,d} p_i^d \left(x_i^d - \sum_j f_{ij}^d + \sum_j f_{ji}^d\right)$$

$$s.t. \quad f \in \Pi$$

- Subgradient

$$g_i^d(p) = x_i^d(p) + \sum_j f_{ji}^d(p) - \sum_j f_{ij}^d(p)$$

- Subgradient algorithm to min *D(p)*

$$\boxed{p_i^d(t+1) = \left[p_i^d(t) + \gamma g_i^d(p(t))\right]^+}$$

Dual decomposition

- Dual decomposition:

$$D(p) = \max_{x_{id},\, f_{ij}^d} \sum_{i,d} U_{id}(x_i^d) - \sum_{i,d} p_i^d (x_i^d - \sum_j f_{ij}^d + \sum_j f_{ji}^d)$$

$$s.t. \quad f \in \Pi$$

- Dual problem has 2 subproblems:

$$D_1(p) = \max_{x_i^d} \sum_{i,d} U_{id}(x_i^d) - x_i^d p_i^d \longleftarrow \text{rate control}$$

$$D_2(p) = \max_{f_{ij}^d} \sum_{i,d} p_i^d (\sum_j f_{ij}^d - \sum_j f_{ji}^d) \longleftarrow \text{routing, scheduling}$$

$$\text{s.t.} \quad f \in \Pi$$

Cross-layer implementation

- 1^{st} subproblem: solved by rate control using local congestion price

$$D_1(p) = \sum_{i,d} \max_{x_i^d} \left(U_{id}(x_i^d) - x_i^d p_i^d \right)$$

Transport: rate control based on local price

$$x_i^d(t) = U_{id}^{'\,-1}(p_i^d(t))$$

Cross-layer implementation

- 2nd subproblem: equivalent form solved by routing and scheduling

scheduling

$$D_2(p) = \max_{f_{ij}} \sum_{i,j} f_{ij} \max_{d} \left(p_i^d - p_j^d \right)$$

routing

$$\text{s.t.} \quad f \in \Pi$$

- Nodes maintain a separate queue for each destination d

max diff. price: $w_{ij}(t) := \max_{d} \; p_i^d(t) - p_j^d(t)$

dest with max $w_{ij}(t)$: $d_{ij}(t) := \arg\max_{d} \; p_i^d(t) - p_j^d(t)$

Cross-layer implementation

- 2^{nd} subproblem: equivalent form solved by routing and scheduling

scheduling

$$D_2(p) = \max_{f_{ij}} \sum_{i,j} f_{ij} \max_{d} \left(p_i^d - p_j^d \right)$$

routing

$$\text{s.t.} \quad f \in \Pi$$

Network: routing based on differential price

- Output link (i, j) serves only queue $d_{ij}(t)$ with max differential price
- It transmits from queue $d_{ij}(t)$ at rate c_l if it is scheduled to send

Cross-layer implementation

- 2nd subproblem: equivalent form solved by routing and scheduling

scheduling

$$D_2(p) = \max_{f_{ij}} \sum_{i,j} f_{ij} \max_{d} \left(p_i^d - p_j^d\right)$$

routing

$$\text{s.t.} \quad f \in \Pi$$

Becomes stochastic with time-varying channels

Link: scheduling (centralized)

- Solve for maximum independent set $e(t)$:

$$e(t) := \arg\max_{e \in E} \sum_{(i,j) \in e} c_{ij} w_{ij}(t)$$

- All links l in $e(t)$ transmit at rates c_l

Summary of Cross-layer Algorithm

$$p_i^d(t+1) = \left[p_i^d(t) + \gamma g_i^d(p(t)) \right]^+$$

$$\begin{tikzcd}[column sep=huge]
\boxed{\text{TCP rate control}} \arrow[r, shift left, "x_i^d(t)"] & \boxed{\text{Queue at nodes}} \arrow[l, shift left, "p_i^d(t)"] \arrow[r, shift right, "p_i^d(t)"'] & \boxed{\text{Routing + Scheduling}} \arrow[l, shift right, "f_{ij}^d(t)"']
\end{tikzcd}$$

Stability

Theorem

The subgradient algorithm converges arbitrarily close to optimum

i.e., given any $\delta>0$, there exists a sufficiently small stepsize such that

primal objective function

$$\boxed{\begin{aligned} &\liminf_{t\to\infty} \; P(\overline{x}(t)) \geq P(x^*) - \delta \\ &\limsup_{t\to\infty} \; D(\overline{p}(t)) \leq D(p^*) + \delta \end{aligned}}$$

dual objective function

$\overline{x}(t),\ \overline{p}(t)$: running avg

x^*, p^*: optimum

Time-varying Channel

- Channel state $h(t)$ is an i.i.d. finite state process with distribution $q(h(t))$
- In channel state h
  - Link l's capacity is $c_l(h)$
  - Feasible rate region is $\Pi(h)$
- Extend the cross-layer algorithm with only a modification to scheduling

$$\max_{f_{ij}} \sum_{i,j} f_{ij} w_{ij}(t) \quad \text{s.t.} \quad f \in \Pi(h(t))$$

random

- Questions: stability? optimality?

Reference System

- Define mean feasible rate region

$$\overline{\Pi} = \left\{ \bar{r} : \bar{r} = \sum_h q(h) r(h),\ r(h) \in \Pi(h) \right\}$$

- Define reference system problem

$$\max_{x_i^d,\, f_{ij}^d} \quad \sum_{i,d} U_{id}(x_i^d)$$

$$s.t. \qquad x_i^d \le \sum_j f_{ij}^d - \sum_j f_{ji}^d,\ i \ne d$$

$$\boxed{f \in \overline{\Pi}}$$

Stability

- Congestion price is a positively recurrent Markov chain

$$p_i^d(t+1) = \left[p_i^d(t) + \gamma_t g_i^d(p(t))\right]^+$$

- Proof by stochastic Lyapunov analysis
  - Dual function of the reference problem: $\overline{D}(p)$
  - Optimal price: p^*
  - Lyapunov function: $V(p) = \| p - p^* \|_2^2$
  - Then:

$$E[V(p(t+1)) - V(p(t)) \mid p(t) = p] \leq -\gamma^2 G^2 \left(I_{p \in A^c} - I_{p \in A}\right)$$

where $A = \{p : \| p - p^* \|_2 \leq \delta\}$

$$\delta = \max_{\overline{D}(p) - \overline{D}(p^*) \leq \gamma G^2} \| p - p^* \|_2$$

Optimality

Theorem

The stochastic subgradient algorithm converges arbitrarily close to optimum

i.e., given any $\delta > 0$, there exists a sufficiently small stepsize such that

$$\boxed{\begin{array}{l} \overline{D}\big(E[p(\infty)]\big) \le \overline{D}(p^*) + \delta \\ \overline{P}\big(E[x(\infty)]\big) \ge \overline{P}(x^*) - \delta \end{array}}$$

$E\big[x(\infty)\big],\ E\big[p(\infty)\big]$: expected vaue in steady state

x^*, p^*: optimum of reference problem

Conclusion

- Joint rate control, routing, scheduling design for wireless networks
- Subgradient algorithm
  - Node prices adjusted according to excess demand
  - Traffic source controls its rate using marginal utility function based on local price
  - Only destination (queue) with maximum differential price is served over each link
  - Routing ‘absorbed’ into congestion control and scheduling
  - Combine backpressure and congestion pricing
- Extension to time-varying channel
- In general: dual solution to convex G.NUM remains stable and optimal (on average) under (Markov model) stochastically varying constraint set

Part VII

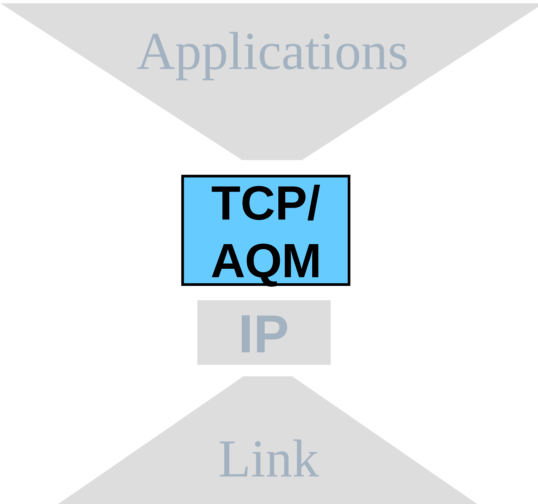
Case 3: TCP/IP Interactions

Some References

- J. Wang, L. Li, S. H. Low and J. C. Doyle, "Cross-layer Optimization in TCP/IP Networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 582-268, June 2005

- J. He, M. Chiang, and J. Rexford, "TCP/IP interaction based on congestion prices: Stability and optimality", *Proc. IEEE ICC*, June 2006

Protocol decomposition



TCP-AQM

$$\max_{x \geq 0} \quad \sum_i U_i(x_i)$$

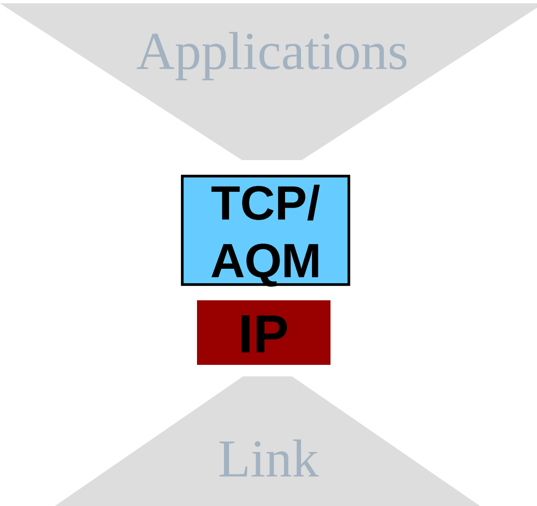
$$\text{subject to} \quad Rx \leq c$$

TCP-AQM:

- TCP algorithms maximize utility with different utility functions

Congestion prices coordinate across protocol layers

Protocol decomposition



$$\overset{\boxed{\text{IP}}}{\max_{R}} \quad \overset{\boxed{\text{TCP-AQM}}}{\max_{x \geq 0}} \quad \sum_i U_i(x_i)$$

$$\text{subject to} \quad Rx \leq c$$

TCP/IP:

- TCP algorithms maximize utility with different utility functions
- Shortest-path routing is optimal using congestion prices as link costs ......

Congestion prices coordinate across protocol layers

Two timescales

- Instant convergence of TCP/IP
- Link cost = $a\, p_l(t) + b\, d_l$

static

price

- Shortest path routing $R(t)$

TCP/AQM

IP

$a\ p(0)$ $\qquad$ $a\ p(1)$

$R(0)$ $\qquad$ $R(1)$ $\cdots$ $R(t),\ \ R(t+1), \cdots$

TCP-AQM/IP Model

TCP

$$x(t) = \arg\max_{x \geq 0} \sum_i U_i(x_i)$$

$$\text{subject to} \quad R(t)x \leq c$$

AQM

$$p(t) = \arg\min_{p \geq 0} \sum_i \left(\max_{x_i \geq 0} U_i(x_i) - x_i \sum_l R_{li}(t) p_l \right) + \sum_l c_l p_l$$

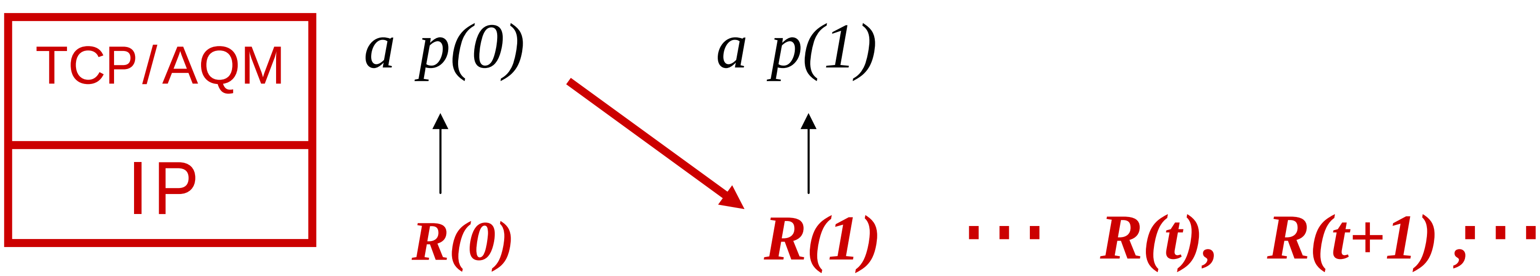
IP

Link cost

$$R_i(t+1) = \arg\min_{R_{li}} \sum_l R_{li}(a p_l(t) + b d_l)$$

Questions

- Does equilibrium routing R_a exist ?
- What is utility at R_a?
- Is R_a stable ?
- Can it be stabilized?



Delay-insensitive utility $U_i(x_i)$

Theorem

TCP/IP equilibrium solves the primal problem for general networks only if $b=0$

- i.e., if route based purely on congestion prices

$$\text{Primal:} \quad \max_{R} \max_{x \geq 0} \sum_i U_i(x_i) \quad \text{subject to } Rx \leq c$$

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

Delay-insensitive utility $U_i(x_i)$

Theorem

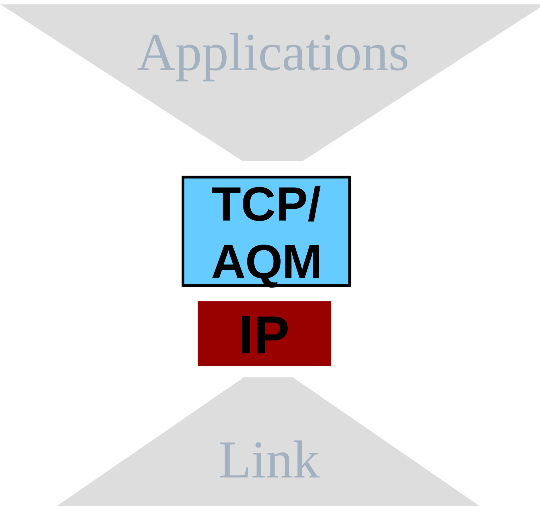
If $b=0$, R_a exists iff zero duality gap

- Shortest-path routing is optimal with congestion prices
- No penalty for not splitting

$$\text{Primal:}\quad \max_{R} \max_{x\geq 0} \sum_i U_i(x_i) \quad \text{subject to } Rx \leq c$$

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

Delay-insensitive utility $U_i(x_i)$



$$\overset{\boxed{\text{IP}}}{\max_{R}} \quad \overset{\boxed{\text{TCP-AQM}}}{\max_{x \geq 0}} \quad \sum_{i} U_i(x_i)$$

$$\text{subject to} \qquad Rx \leq c$$

TCP/IP: *b=0*

- Equilibrium of TCP/IP exists iff zero duality gap
- NP-hard, but subclass with zero duality gap is in P
- Equilibrium, if exists, can be unstable
- Can stabilize, but with reduced utility

Delay-sensitive utility $U_i(x_i, d_i)$

<u>Theorem</u>

If $a>0$, $b>0$, then TCP/IP equilibrium solves the primal problem for general networks if

$$U_i(x_i, d_i) = V_i(x_i) - \frac{b}{a} x_i d_i$$

Moreover no other "reasonable" class of utility functions work

$$\text{Primal: } \max_{R} \max_{x \geq 0} \sum_i U_i\left(x_i, \sum_l R_{li} \tau_l\right) \quad \text{subject to } Rx \leq c$$

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

Delay-sensitive utility $U_i(x_i, d_i)$

Theorem

Suppose $a>0$, $b>0$ and $U_i(x_i, d_i) = V_i(x_i) - ba^{-1}x_i d_i$

Then equilibrium routing exists iff zero duality gap

- Shortest-path routing is optimal with congestion prices
- No penalty for not splitting

$$\text{Primal:} \quad \max_{R} \max_{x \geq 0} \sum_i U_i\left(x_i, \sum_l R_{li}\tau_l\right) \quad \text{subject to} \quad Rx \leq c$$

$$\text{Dual:} \quad \min_{p \geq 0} \left(\sum_i \max_{x_i \geq 0}\left(U_i\left(x_i, \sum_l R_{li}\tau_l\right) - x_i \max_{R_i} \sum_l R_{li} p_l \right) + \sum_l p_l c_l \right)$$

Extensions

- Other timescale separations between TCP and IP dynamics (He, Chiang, Rexford 2006)
- Forward engineering:
  - DATE (Dynamic Adaptive Traffic Engineering)
- HTTP/TCP interactions (Chang Liu 2004)

Part VIII

Future Research Challenges and Summary

Future Research Issues

- Technical: Global stability under delay...
- Modeling: routing in ad hoc network, ARQ, MIMO...
- Time issues
- Why deterministic fluid model?

Shannon 1948: remove finite blocklength, Law of Large Numbers kicks in (later finite codewords come back...)

Kelly 1998: remove coupled queuing dynamics, optimization and decomposition view kicks in (later stochastics come back...)

- What if it's not convex optimization?

Rockafellar 1993: Convexity is the watershed between easy and hard (what if it's hard?)

- Is performance the only optimization objective?

Research Challenges

A sample of 30 bullets in three categories

Open Problems

- Stochastic stability for general filesize distribution, general utility functions and convex constraint set, without timescale separation?
- Performance (utility, delay...) under session, channel, and packet level stochastic?
- Impacts of stochastic feedback for multi-timescale decompositions?
- Validation of fluid model from packet level dynamics?
- Global convergence of successive convex approximations for signomial programming?
- Distributed Sum-of-Squares for nonconcave NUM
- Duality gap: estimation, bounding, and implications
- Tight bound on the rate of convergence of various distributed algorithms?
- Practical stepsize rules in asynchronous networks?
- Low spatial-temporal complexity scheduling algorithm?
- Global stability under feedback delay?

Open Issues

- Constraint set of G.NUM from information theory?
- How to systematically search alternative G.NUM representations and alternative decompositions?
- Adaptive slicing by primal decompositions?
- Modeling of routing (ad hoc network and BGP)?
- Dealing with utility as functions of delay and transient resource allocations for real-time flows?
- Degree of heterogeneity and price of heterogeneity?
- Topology level stochastic?
- New notions of fairness in S.NUM?
- Quantify suboptimality's impact on fairness?
- Characterize and bound instability?
- Hardware and application modeling?

New Mentalities

- Robustness-optimality tradeoff?
- Move away from optimality?

Suboptimal (with bounded loss of optimality) and simple algorithm for each module

Good architecture contains the "damage" to the overall system

- Stochastic network dynamics is good?

"Washes away" the corner cases?

- From focus on equilibrium to investigations of the transients (eg, how close to optimum within a given time, will resource allocation during transient drop below certain thresholds)?
- How to compare alternative architectures?
- Redesign architectures (especially the division between control protocols and network management systems) for optimizability?
- Quantify other Network X-ities?
- Managing complexity in networks through layering?

A Sample of 20 Methodologies

- Reverse engineering cooperative protocol as an optimization algorithm
- Lyapunov function construction to show stability
- Proving convergence of dual descent algorithm
- Proving stability by singular perturbation theory
- Proving stability by passivity argument
- Proving equilibrium properties through vector field representation
- Reverse engineering non-cooperative protocol as a game
- Verifying contraction mapping by bounding the Jacobian's norm
- Analyzing cross-layer interaction systematically through G.NUM
- Change of variable for decoupling, and computing minimum curvature needed

A Sample of 20 Methodologies

- Dual decomposition for jointly optimal cross layer design
- Computing conditions under which a general constraint set is convex
- Introducing an extra "layer" to decouple the problem
- End user generated pricing
- Different timescales of protocol stack interactions through different decomposition methods
- Maximum differential congestion pricing for node-based back-pressure scheduling
- Absorbing routing functionality into congestion control and scheduling
- Primal and dual decomposition for coupling constraints
- Consistency pricing for decoupling coupled objective
- Partial and hierarchical decompositions for architectural alternatives

10 Key Messages

- Existing protocols in layers 2,3,4 have been reverse engineered
- Reverse engineering leads to better design
- There is one unifying approach to cross-layer design
- Loose coupling through layering price
- Queue length often a right layering price, but not always
- Many alternatives in decompositions and layering architectures
- Convexity is key to proving global optimality
- Decomposability is key to designing distributed solution
- Still many open issues in modeling, stochastic dynamics, and nonconvex formulations
- Architecture, rather than optimality, is the key

Contacts

chiangm@princeton.edu
www.princeton.edu/∼chiangm
slow@caltech.edu
netlab.caltech.edu
calderbk@princeton.edu