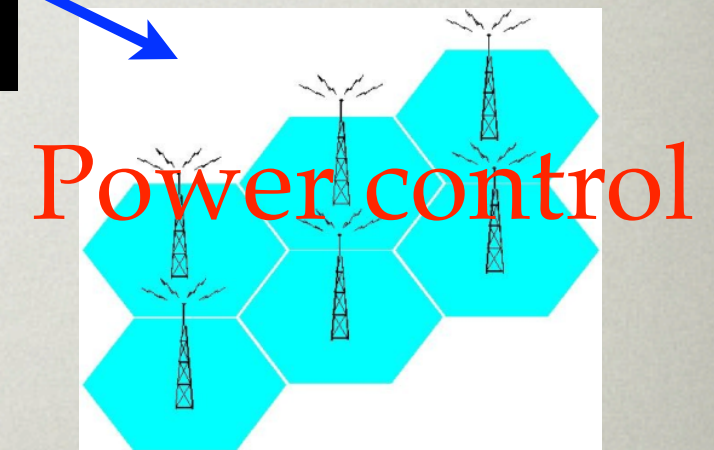
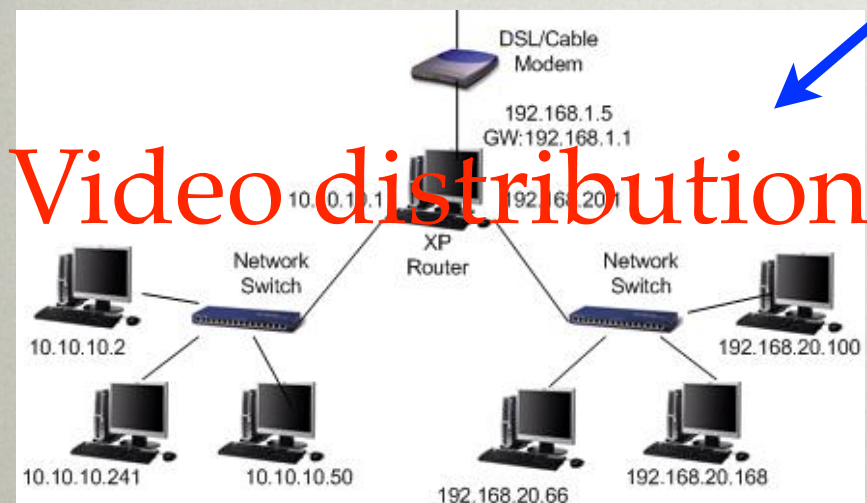
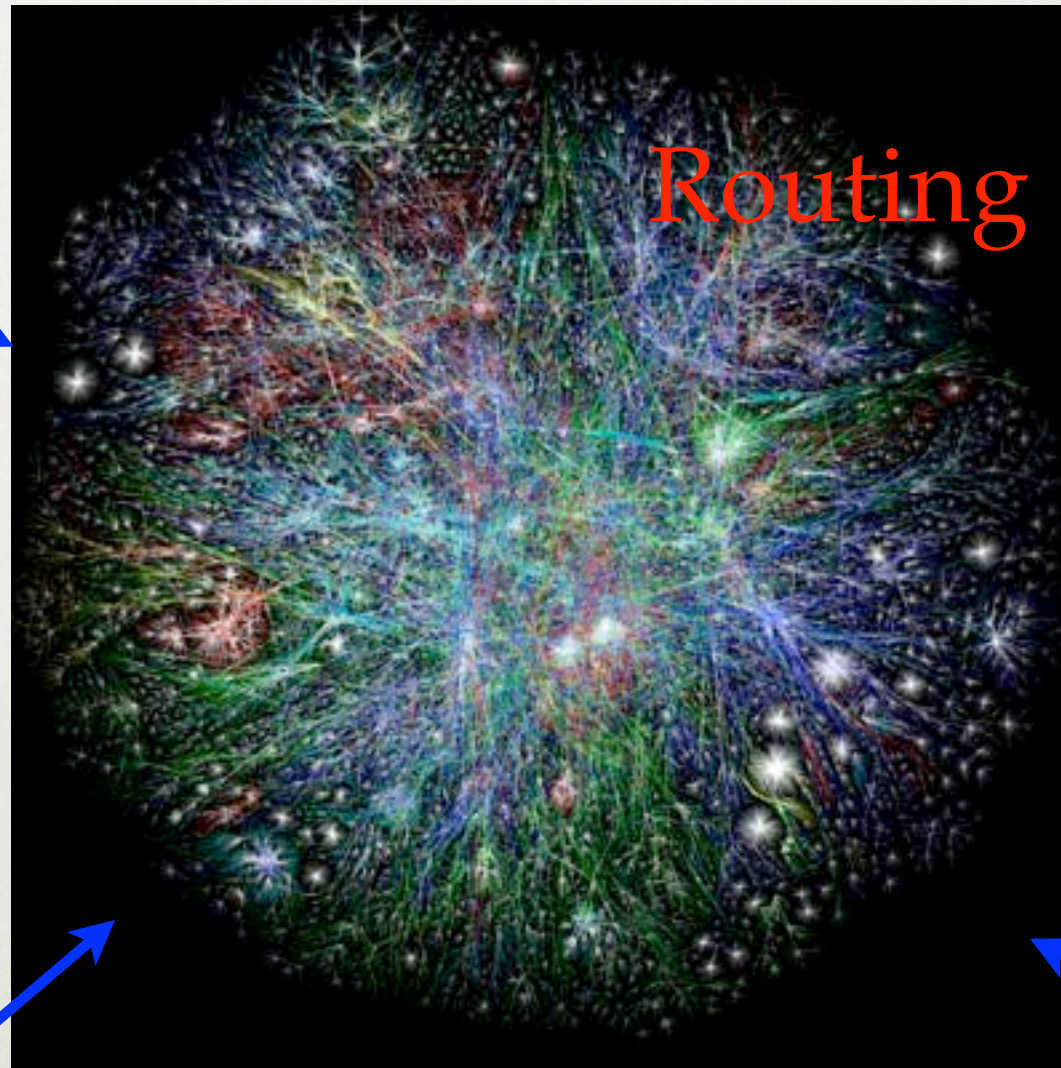
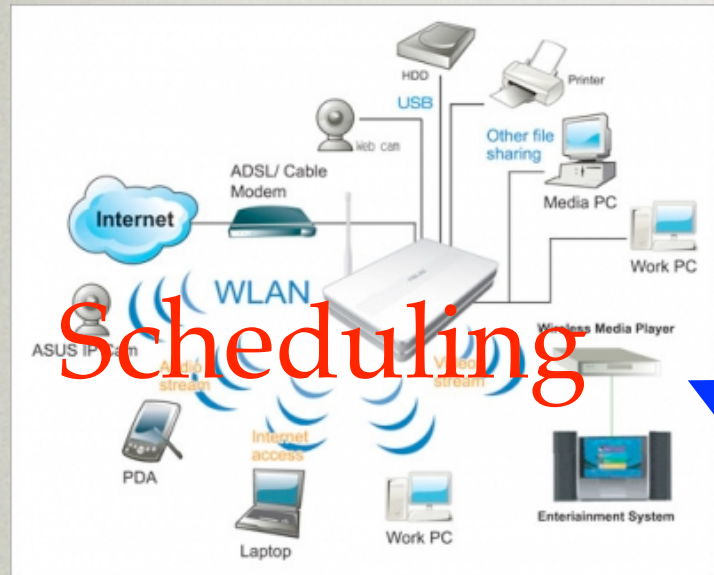


OPTIMIZATION IN NETWORKING

MUNG CHIANG
PRINCETON UNIVERSITY

10TH MOPTA
AUGUST 18, 2010



OPTIMIZATION IN NETWORKING

- Distributed optimization
 - Wireless power control
- Combinatorial optimization
 - P2P streaming capacity
- Nonconvex optimization
 - Internet IP routing
- Stochastic optimization
 - Wireless scheduling
- Optimization as a language for networking

All are recent updates on long-time questions, with interesting math and visible impact

“Distributed” is a keyword

DUAL DECOMPOSITION: THE SIMPLEST CASE

Primal

$$\begin{array}{ll} \text{maximize} & f(x) + g(y) \\ \text{subject to} & x + y \leq 1 \end{array}$$



Lagrangian

$$L(x, y, \lambda) = (f(x) - \lambda x) + (g(y) - \lambda y) + \lambda$$



Dual

$$\begin{array}{ll} \text{minimize} & \max_{x,y} L(x, y, \lambda) \\ \text{subject to} & \lambda \geq 0 \end{array}$$

$$\begin{array}{ll}
\text{maximize} & \sum_i U_i(\gamma_i) \\
\text{subject to} & \frac{G_{ii}p_i}{\sum_{j \neq i} G_{ij}p_j + n_i} \geq \gamma_i \\
\text{variables} & \{p_i, \gamma_i\}
\end{array}$$

How to solve it in a **distributed** way?

Turns out to be a **power control** problem in wireless

$$\begin{array}{ll}
\text{maximize} & \sum_{t \in T} y_t \\
\text{subject to} & \sum_{t \in T} m_{v,t} y_t \leq C_v, \quad \forall v \\
& y_t \geq 0, \quad \forall t \in T \\
\text{variables} & \{y_t\}
\end{array}$$

How to solve this combinatorial tree-embedding problem in **polynomial-time**?

Turns out to be video **streaming capacity** in P2P

$$\begin{array}{ll}
\text{minimize} & \Phi(\{f_{u,v}, c_{u,v}\}) \\
\text{subject to} & \sum_v f_{s,v}^t - \sum_u f_{u,s}^t = D(s, t), \quad \forall s \neq t \\
& f_{u,v} \leq c_{u,v} \quad \forall (u, v) \\
& f_{u,v} = \sum_t f_{u,v}^t, \quad \forall (u, v) \\
\text{variables} & \{f_{u,v}^t, f_{u,v}\}
\end{array}$$

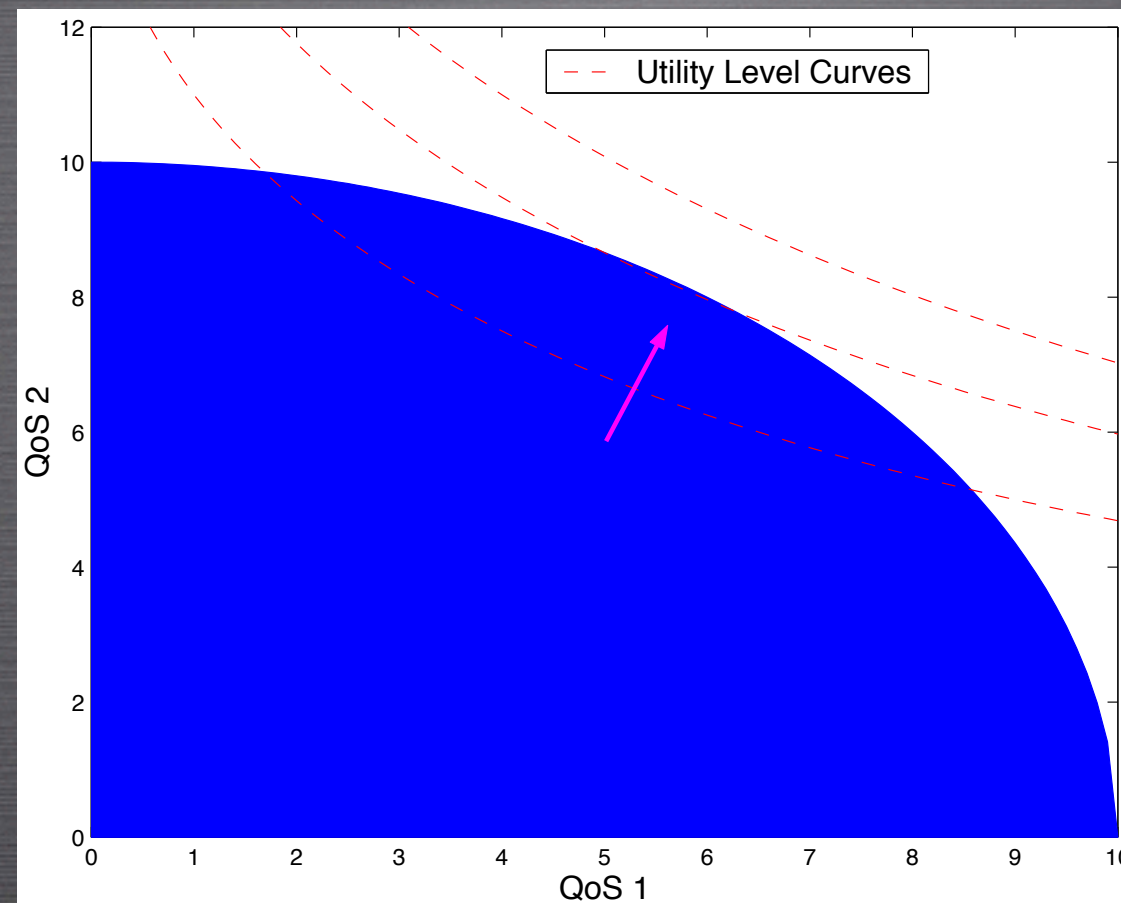
Multi-commodity flow with a twist: can only solve via
update of weights used at each node

Turns out to be **IP routing** in the Internet

$$\begin{array}{ll}
\text{maximize} & \sum_l U_l(x_l) \\
\text{subject to} & x_l \leq \sum_{\mathbf{s} \in \mathcal{S}: s_l=1} \pi_{\mathbf{s}}, \quad \forall l \\
& \pi_{\mathbf{s}} \geq 0, \quad \forall \mathbf{s} \\
& \sum_{\mathbf{s} \in \mathcal{S}} \pi_{\mathbf{s}} = 1 \\
\text{variables} & \{x_l, \pi_{\mathbf{s}}\}
\end{array}$$

How to approach optimality based on **local observations** of stochastic network state?

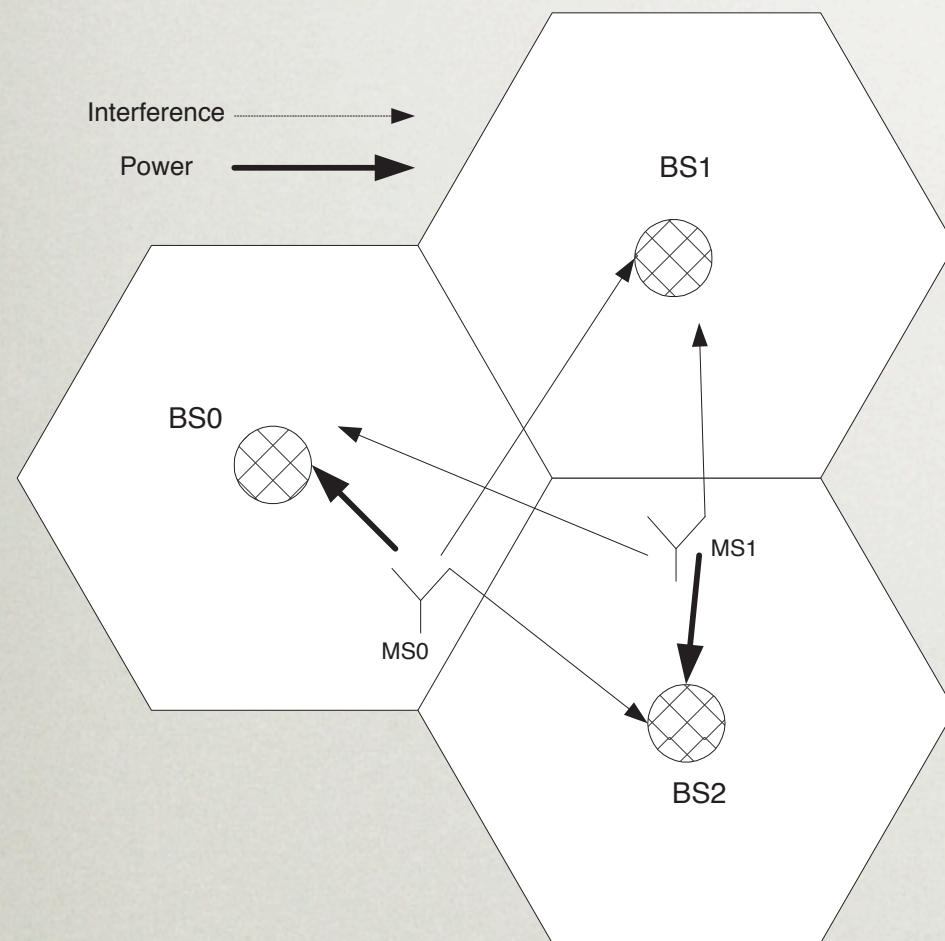
Turns out to be random access **scheduling** in wireless



POWER CONTROL

P. HANDE, S. RANGAN, M. CHIANG, X. WU, "DISTRIBUTED UPLINK POWER CONTROL FOR OPTIMAL SIR ASSIGNMENT IN CELLULAR DATA NETWORKS", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 16, NO. 6, PP. 1430-1443, NOVEMBER 2008

POWER CONTROL: SYSTEM MODEL



Mobile Stations (MS)

Base Stations (BS)

Each MS served by a BS

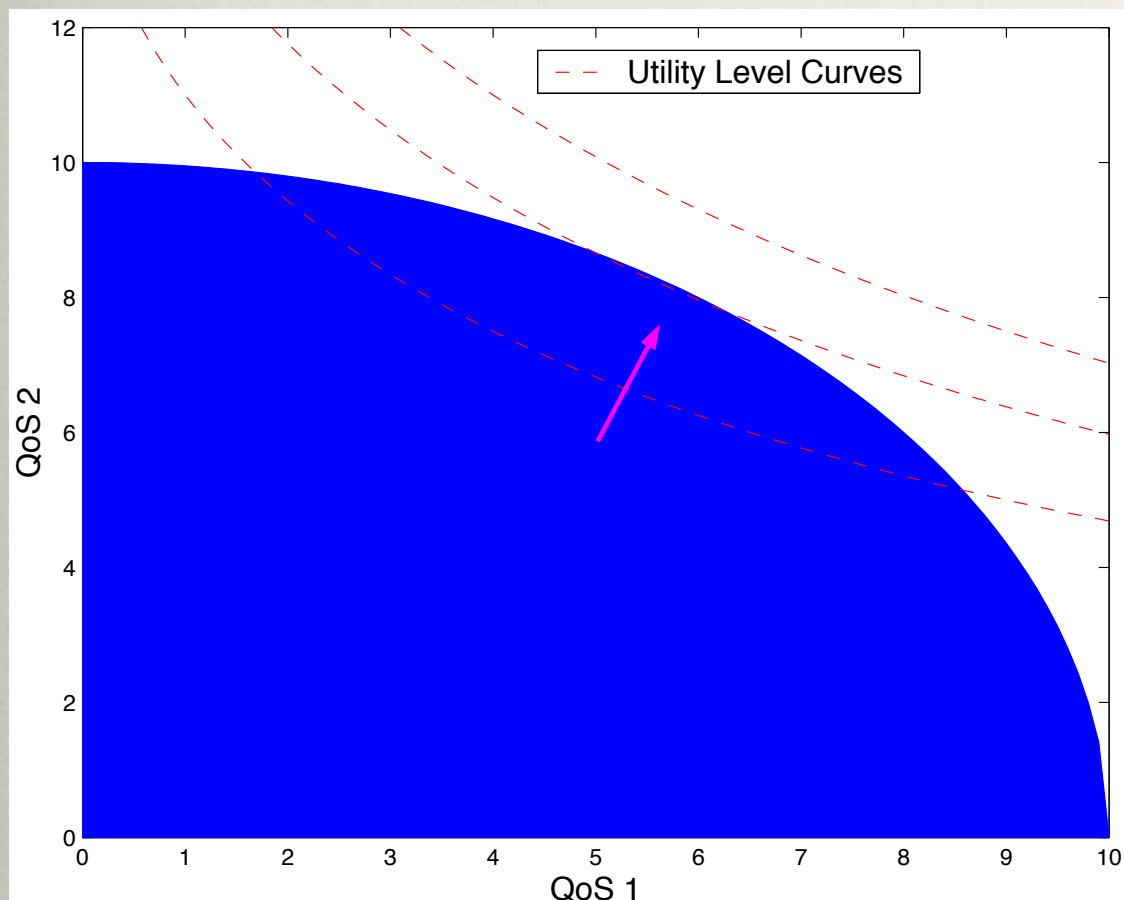
Each BS serving a set of MS

Interference-limited wireless
data networks

Transmit power control in 2G -> 3G -> 4G networks

POWER CONTROL: OPTIMIZATION FORMULATION

$$\text{SIR : } \gamma_i = \frac{p_i G_{ii}}{\sum_{j \neq i} p_j G_{ij} + n_i}$$



Maximize: utility function of SIR
Subject to: SIR feasibility
Variables: transmit power **and** SIR assignments

POWER CONTROL: PARAMETERIZATION

γ is feasible iff there exists an $\mathbf{s} \succ 0$ such that

$$\mathbf{s}^T \mathbf{GD}(\gamma) = \mathbf{s}^T$$

$\mathbf{s}$ load vector $\mathbf{r}$ spillage vector $\mathbf{r} = \mathbf{G}^T \mathbf{s}$

New (**left-eigenvector**) parametrization of SIR feasibility
boundary: $\gamma = \mathbf{s}/\mathbf{r}$

Intuition: assign high SIR to MS with

- good direct channel
- weak interfering channel

POWER CONTROL: LOAD SPILLAGE ALGORITHM

Initialize: Arbitrary $s(0) \succ 0$

BS broadcasts load factor sum $l_k(t) = \sum_{i \in k} s_i(t)$

MS

• computes spillage factor $r_i(t) = \sum_{k \neq i} h_{ki} l_k$

• assign target SIR value $\gamma_i(t) = s_i(t) / r_i(t)$

• update power to attain target

• measure interference $q_i(t)$

• update load factor $s_i(t+1) = s_i(t) + b(t) \left(\frac{U'_i(\gamma_i) \gamma_i}{q_i} - s_i(t) \right)$

Continue: $t := t+1$

POWER CONTROL: CONVERGENCE AND OPTIMALITY

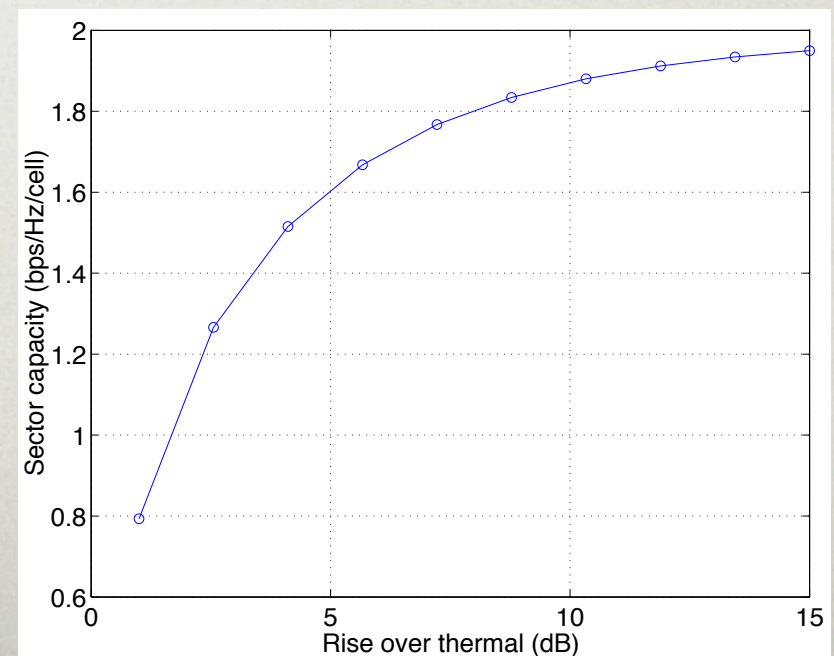
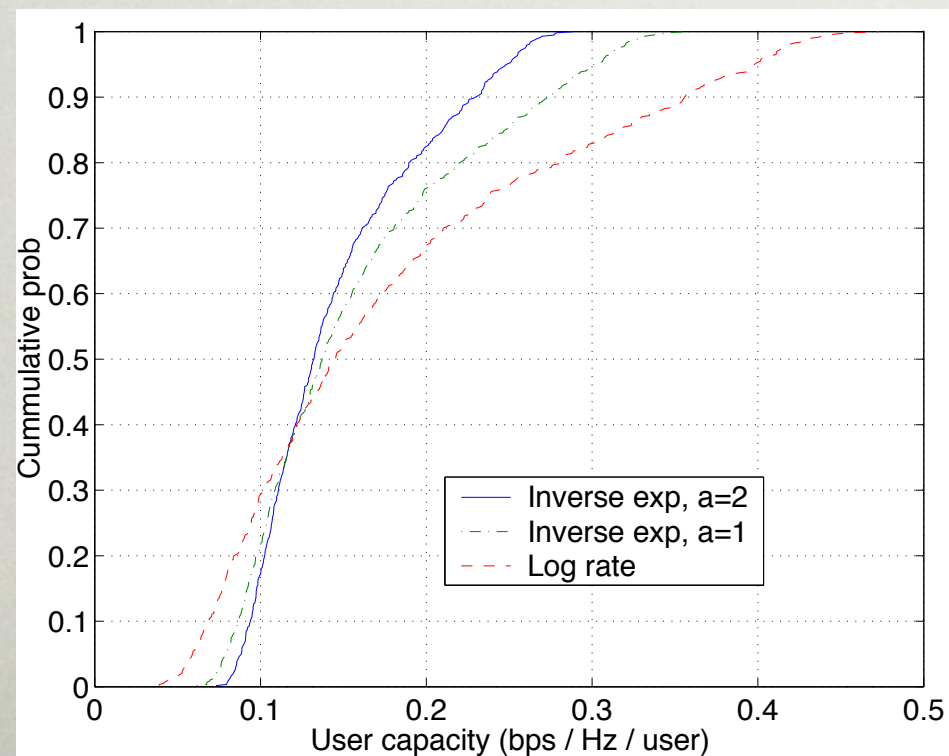
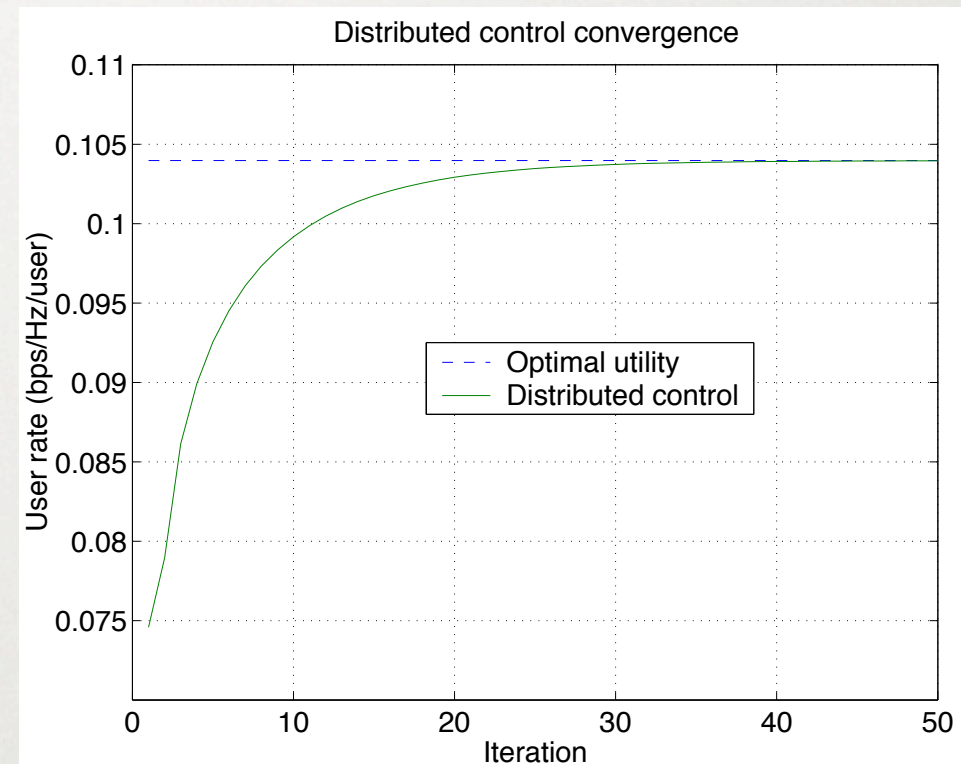
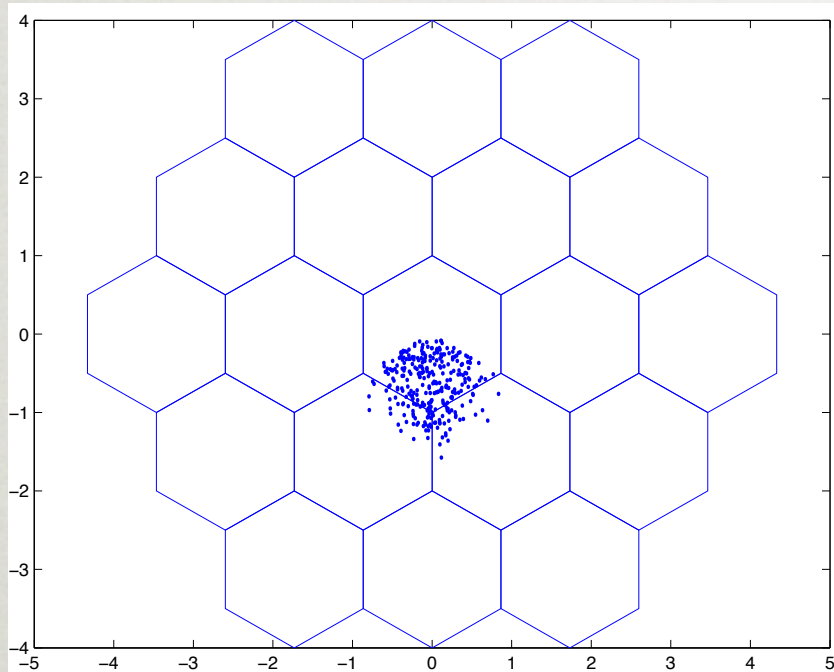
For sufficiently concave and starvation-free utility function,
algorithm converges to global optimizer of

$$\begin{array}{ll} \text{maximize} & \sum_i U_i(\gamma_i) \\ \text{subject to} & \frac{G_{ii}p_i}{\sum_{j \neq i} G_{ij}p_j + n_i} \geq \gamma_i \\ \text{variables} & \{p_i, \gamma_i\} \end{array}$$

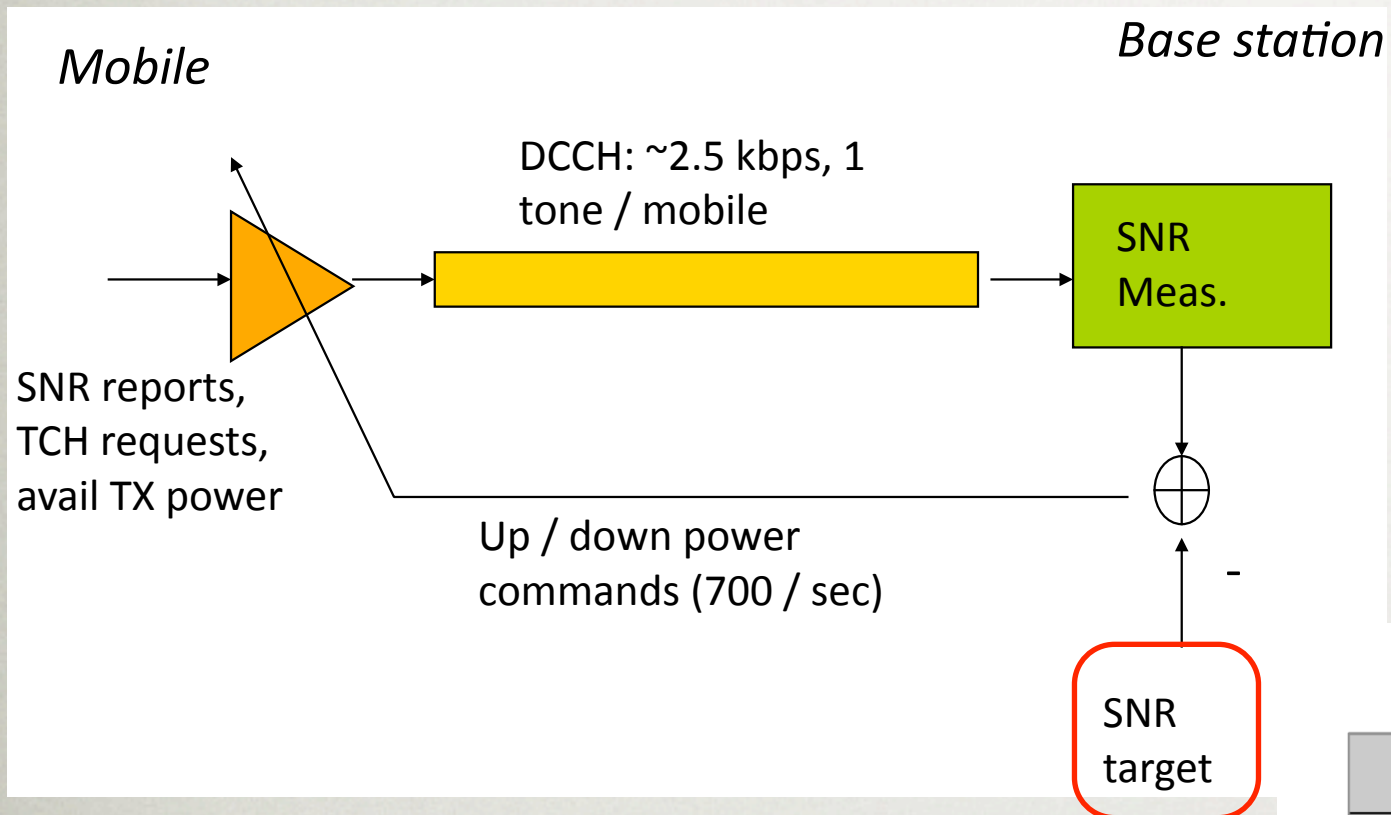
Proof key steps:

- 🔧 Develop a **locally computable** ascent direction
- 🔧 Evaluate KKT conditions
- 🔧 Ensure Lipschitz condition

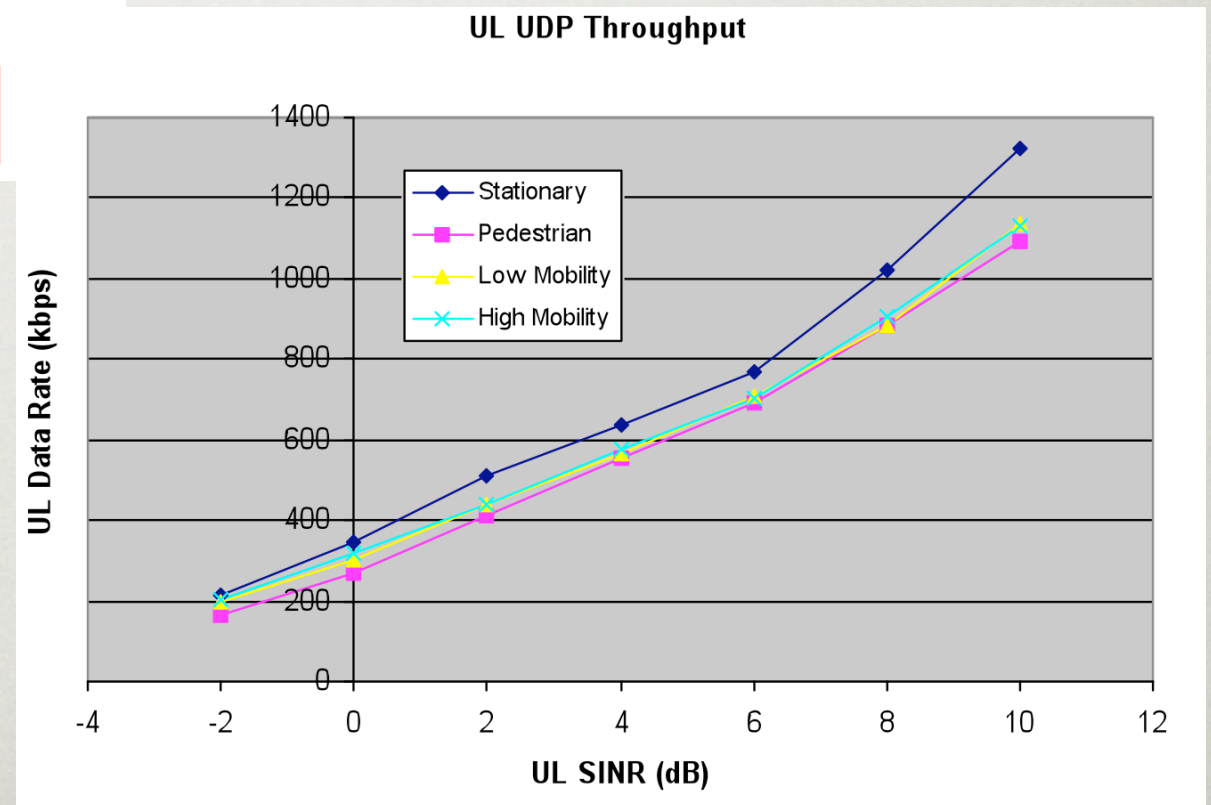
POWER CONTROL: 3GPP SIMULATION

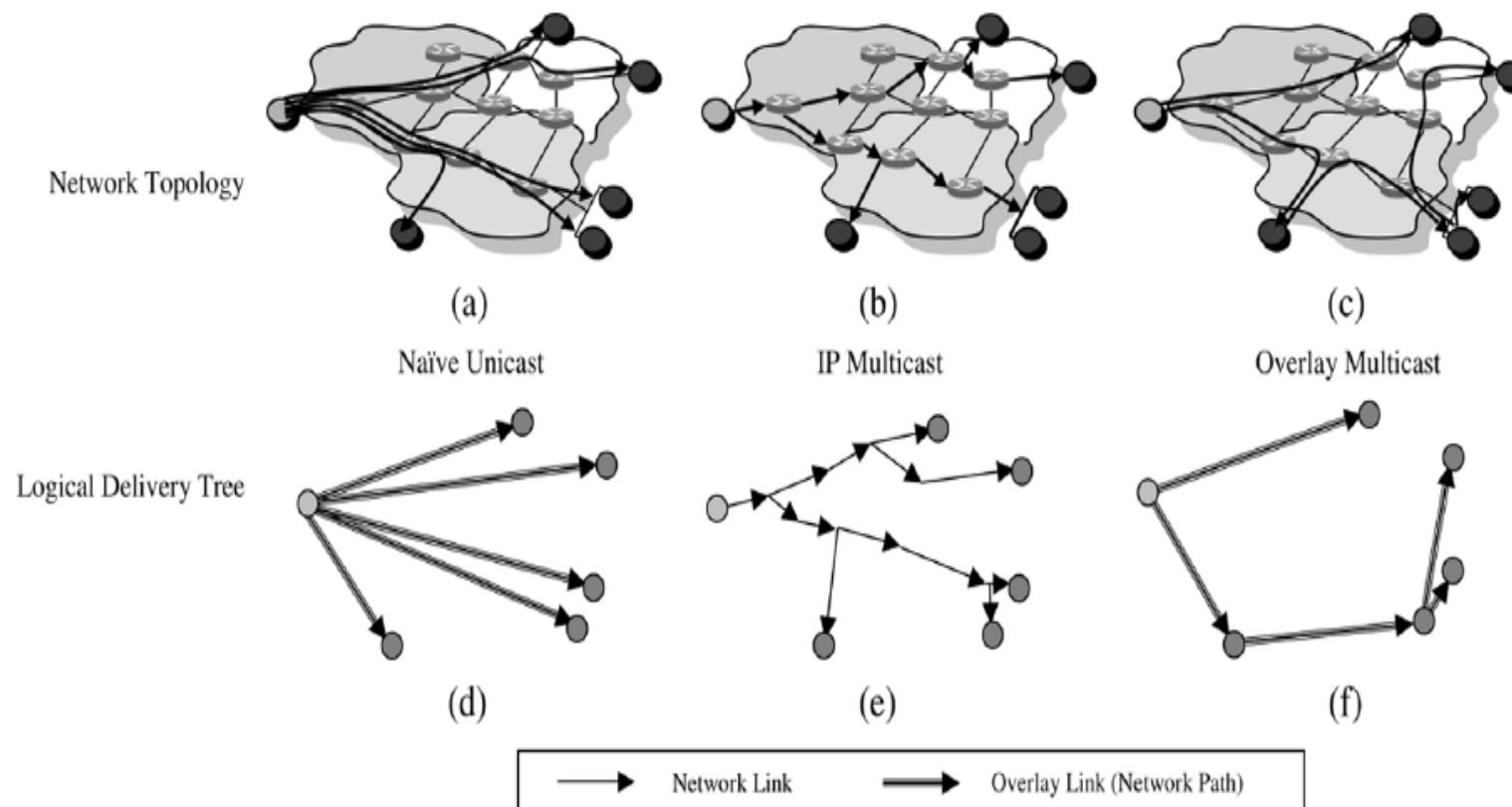


POWER CONTROL: QUALCOMM IMPLEMENTATION



Factor of 4 improvement
in spectral efficiency





P2P STREAMING

M. CHEN, S. LIU, S. SENGPOTA, M. CHIANG, J. LI, AND P. A. CHOU, "P2P STREAMING CAPACITY UNDER NODE DEGREE BOUNDS", PROCEEDINGS OF IEEE ICDCS, JUNE 2010

P2P STREAMING: SCALABLE, HOW FAST?

Rethinks who sends to whom?

- Client-server: not scalable
- Peer to peer: **scalable** for massive amount of sharing

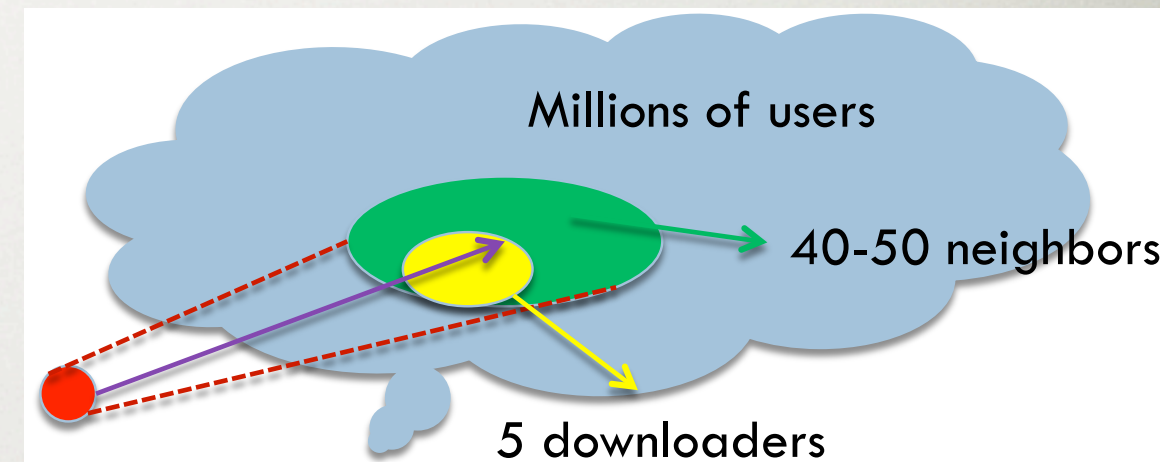
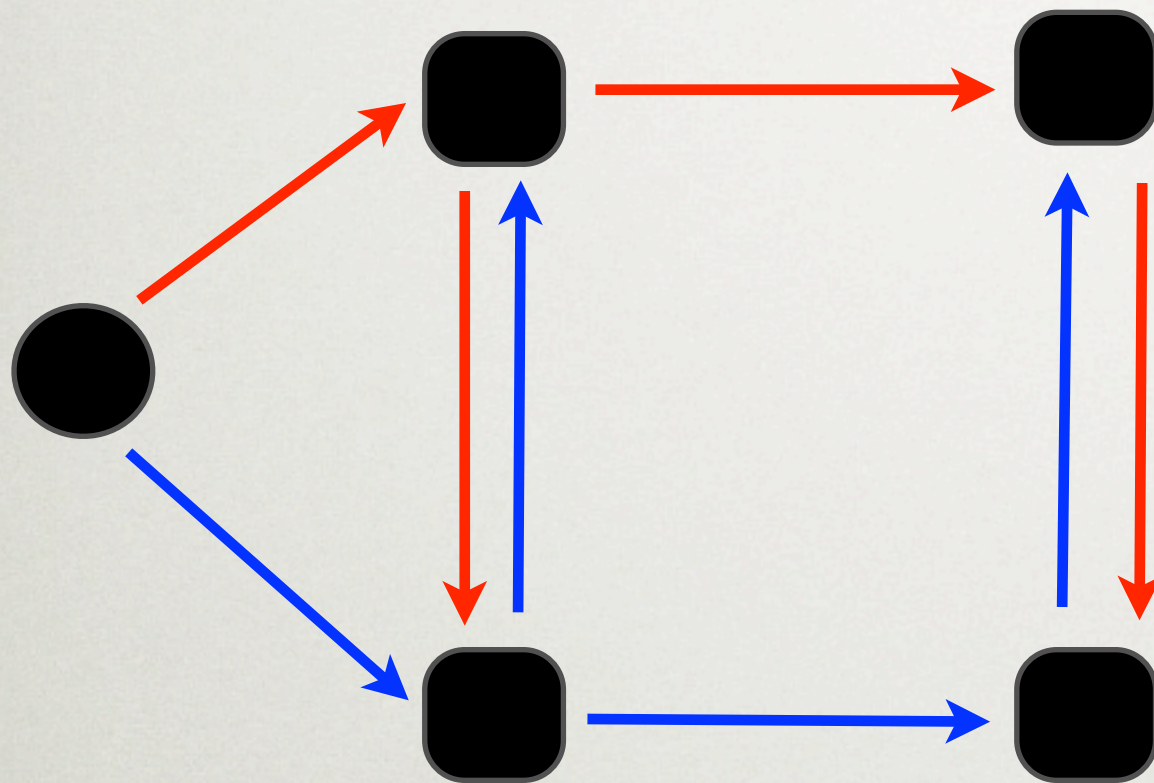
Extremely popular, once 70% of Internet traffic

- File sharing: BitTorrent...
- Video streaming: PPLive...
- Video on demand...

What is the limit of P2P streaming rate?

How to achieve it?

P2P STREAMING: EMBEDDING MULTI-TREES



What is the **highest possible rate** to all the receivers by
optimizing over the overlay topology?

P2P STREAMING: TAXONOMY

8 variations of the problem:

- Given graph full mesh or not?
- Node degree constrained or not?
- Helper nodes exist or not?

• Some are solved exactly

• Some are solved arbitrarily closely

full mesh, degree bound, helper

non full mesh, no degree bound, no helper

• Some are approximated

• One is open

P2P STREAMING: INTUITION

- Constrained multi-tree embedding is too hard
- Turn combinatorial problem into continuous optimization
- Too many trees to search through
- Primal-dual iterative outer loop to guide tree search by price
- Outer loop: update price
- Inner loop: easier combinatorial tree construction

P2P STREAMING: NOTATION

Source: s

Set of receivers: R

Uplink rate: U_v

Uplink capacity: C_v

Tree: t

Set of allowed trees: T

Outgoing degree: $m_{v,t}$

Price: p_v

Total price: $Q(t, \mathbf{p}) = \sum m_{v,t} p_v$

Min price: $\alpha(\mathbf{p}) = \min_t Q(t, \mathbf{p})$

$$\text{rate } r = \sum_t y_t$$

$$U_v = \sum_t m_{v,t} y_t$$

P2P STREAMING: PRIMAL AND DUAL

$$\begin{array}{ll}\text{maximize} & \sum_{t \in T} y_t \\ \text{subject to} & \sum_{t \in T} m_{v,t} y_t \leq C_v, \quad \forall v \\ & y_t \geq 0, \quad \forall t \in T \\ \text{variables} & \{y_t\}\end{array}$$

$$\begin{array}{ll}\text{minimize} & \sum_{v \in V} C_v p_v \\ \text{subject to} & \sum_{v \in V} m_{v,t} p_v \geq 1, \quad \forall t \in T, \\ & p_v \geq 0 \quad \forall v \in V\end{array}$$

P2P STREAMING: ALGORITHM

initialize

while (tree-price small enough)

pick allowed tree with smallest price

$$y = \min_{v \in I_t} C_v / m_{v,t}$$
$$p_v = p_v \left(1 + \epsilon \frac{y}{C_v / m_{v,t}} \right)$$

update counters

end while

normalize and output capacity

P2P STREAMING: EFFICIENCY

📌 Approximation's accuracy: $\epsilon_{\text{tree}} - \epsilon$
for appropriately chosen δ

📌 Time complexity: $\mathcal{O}\left(\frac{N \log N}{\epsilon^2} T_{\text{tree}}\right)$
Use Garg and Konemann 1998

Find smallest-price-tree with small T_{tree} and big ϵ_{tree}

📌 Direction construction

📌 New combinatorial algorithm

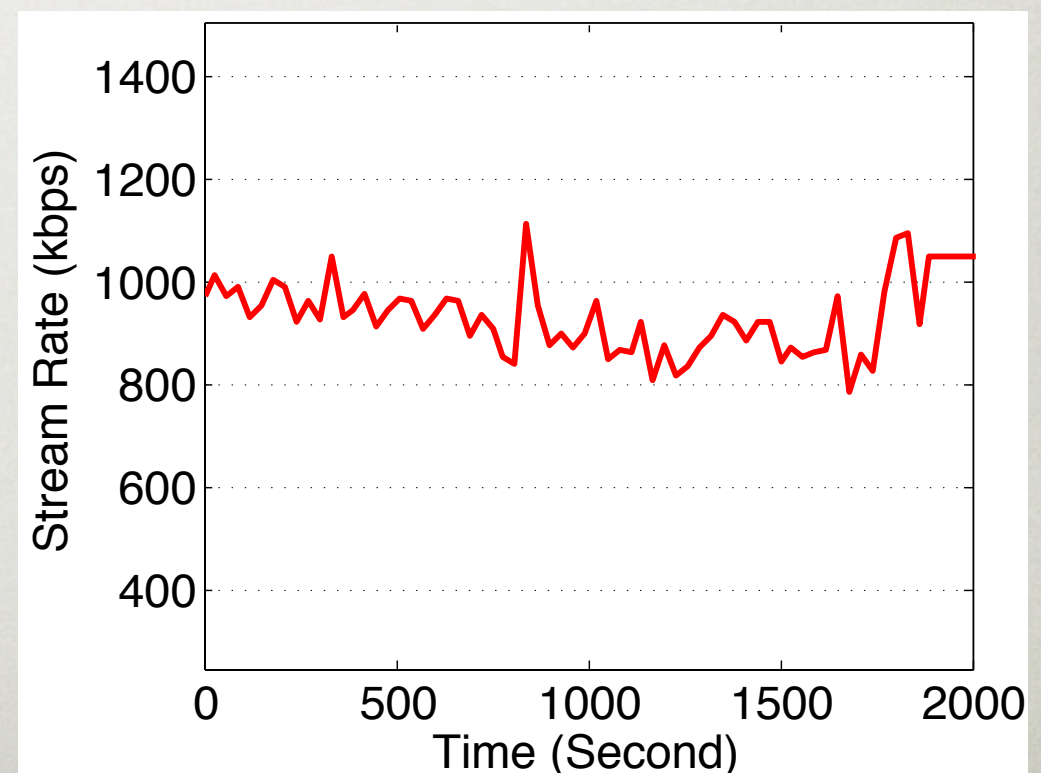
📌 Translation to: shortest arborescence, min cost group Steiner tree, degree constrained survivable network...

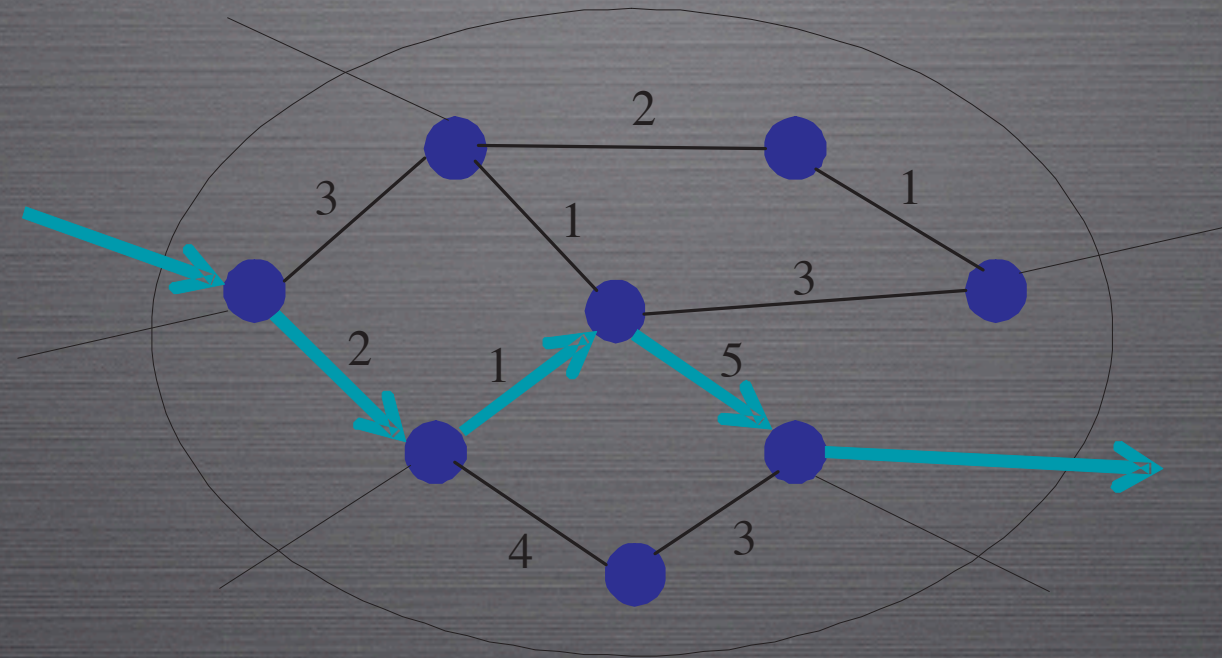
P2P STREAMING: GLOBAL TESTBED



Achieve over 1Mbps high
quality video, about 80% of
streaming capacity

Hundreds of peers around
the world. Joint with G.
Chan, HKUST

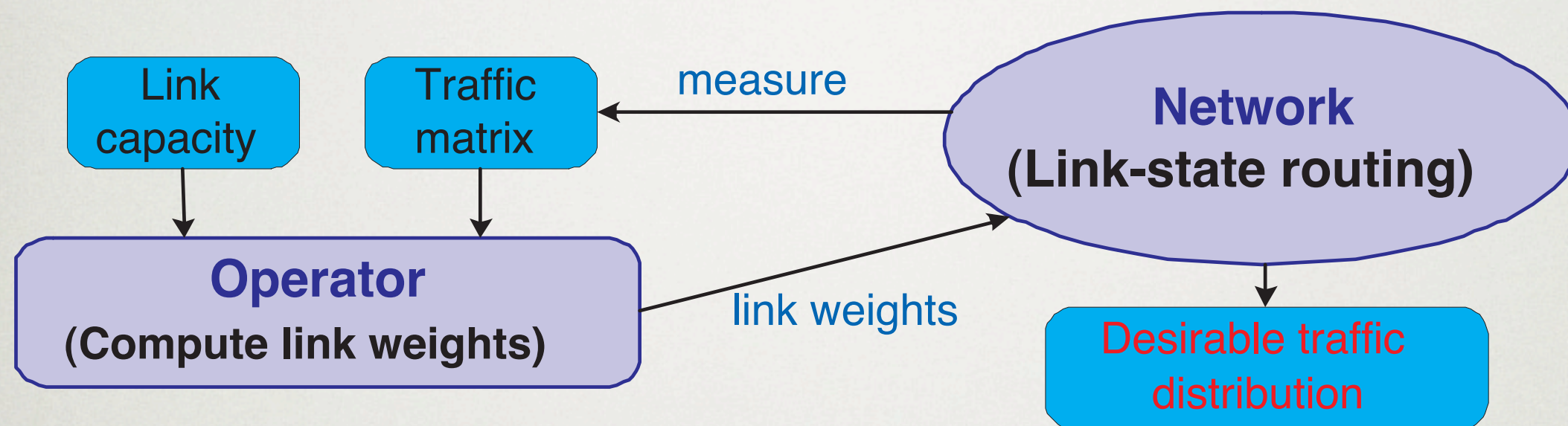




IP ROUTING

D. XU, M. CHIANG, J. REXFORD, "LINK STATE ROUTING ACHIEVES OPTIMAL TRAFFIC ENGINEERING", PROCEEDINGS OF IEEE INFOCOM, MAY 2008

IP ROUTING: PRACTICE TODAY



Internet Routing: a **reverse** shortest path method

- Take in traffic matrix (constraint)
- Vary link weights (variables)
- Hope to minimize sum of link cost function (objective)

In **OSPF**, router **evenly split** traffic along shortest paths
Computing optimal link weights is **NP-hard**

IP ROUTING:

LINK STATE ROUTING

OSPF is just one member of a family called
link state routing with hop by hop forwarding

Involves 3 steps:

- 📌 Centralized **computation for setting link weights**
- 📌 Distributed way of **using these link weights** to split traffic
- 📌 Hop by hop, destination based packet forwarding

A new way to use link weights:

Split traffic on all paths but
exponentially penalize longer ones

A new way to compute them

IP ROUTING: NOTATION

weight for link (u,v): $w_{u,v}$

shortest distance from u to t: d_u^t

distance from u to t if through v: $d_v^t + w_{u,v}$

gap: $h_{u,v}^t = d_v^t + w_{u,v} - d_u^t$

incoming flow at u for destination t: f_u^t

flow on link (u,v) for destination t: $f_{u,v}^t$

$$f_{u,v}^t = f_u^t \frac{\Gamma(h_{u,v}^t)}{\sum_{u,j} \Gamma(h_{u,j}^t)}$$

IP ROUTING: PEFT/DEFT

OSPF:

$$\Gamma_O(h_{u,v}^t) = \begin{cases} 1, & \text{if } h_{u,v}^t = 0 \\ 0, & \text{if } h_{u,v}^t > 0. \end{cases}$$

PEFT:

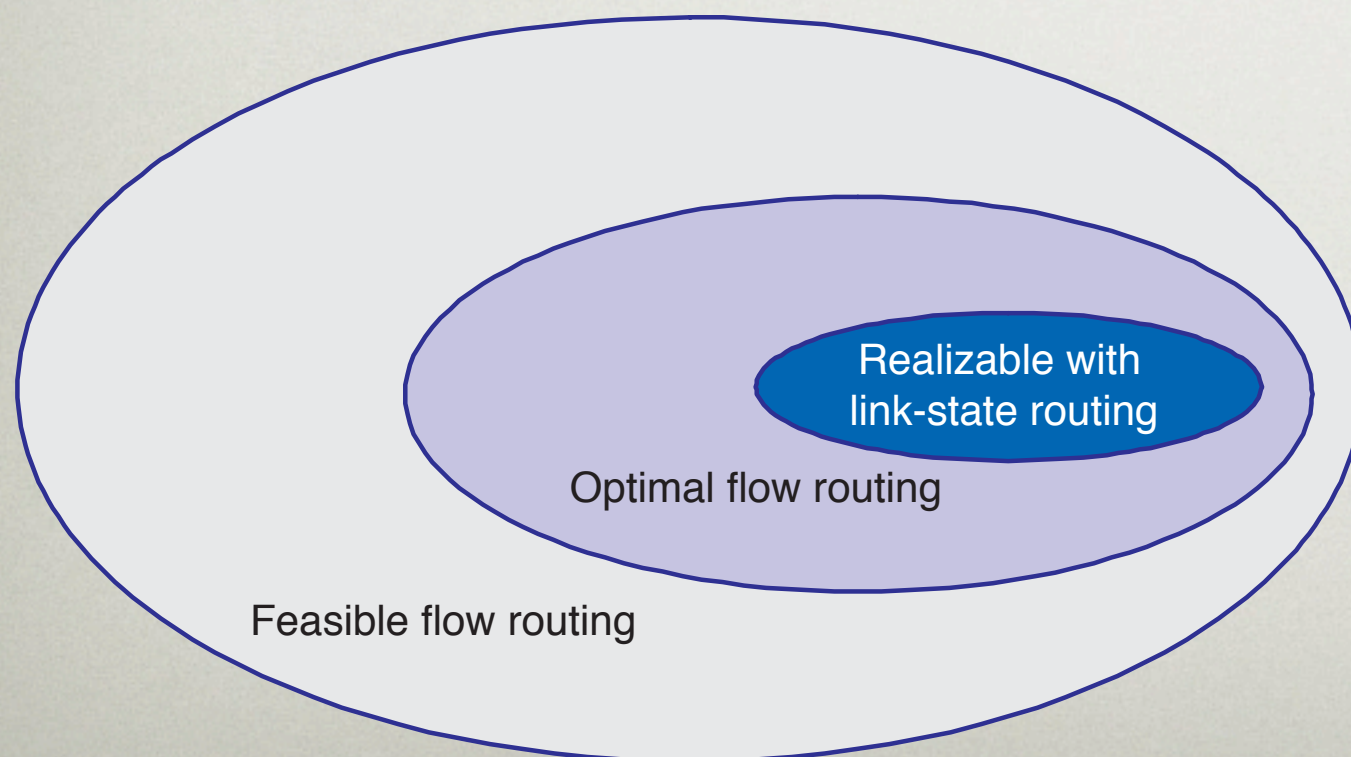
$$\Gamma_P(h_{u,v}^t) = \Upsilon_v^t e^{-h_{u,v}^t}$$

$$\Upsilon_u^t = \sum_{(u,v) \in \mathbb{E}} \left(e^{-h_{u,v}^t} \Upsilon_v^t \right)$$

IP ROUTING: EFFICIENCY

📌 PEFT achieves optimal traffic engineering

📌 Optimal link weights can be computed by a convex optimization (2000 times faster than local search algorithms for OSPF link weight computation)



Find an objective function that picks out only link state realizable traffic distribution

Entropy is the (only) right choice

IP ROUTING: NETWORK ENTROPY MAX

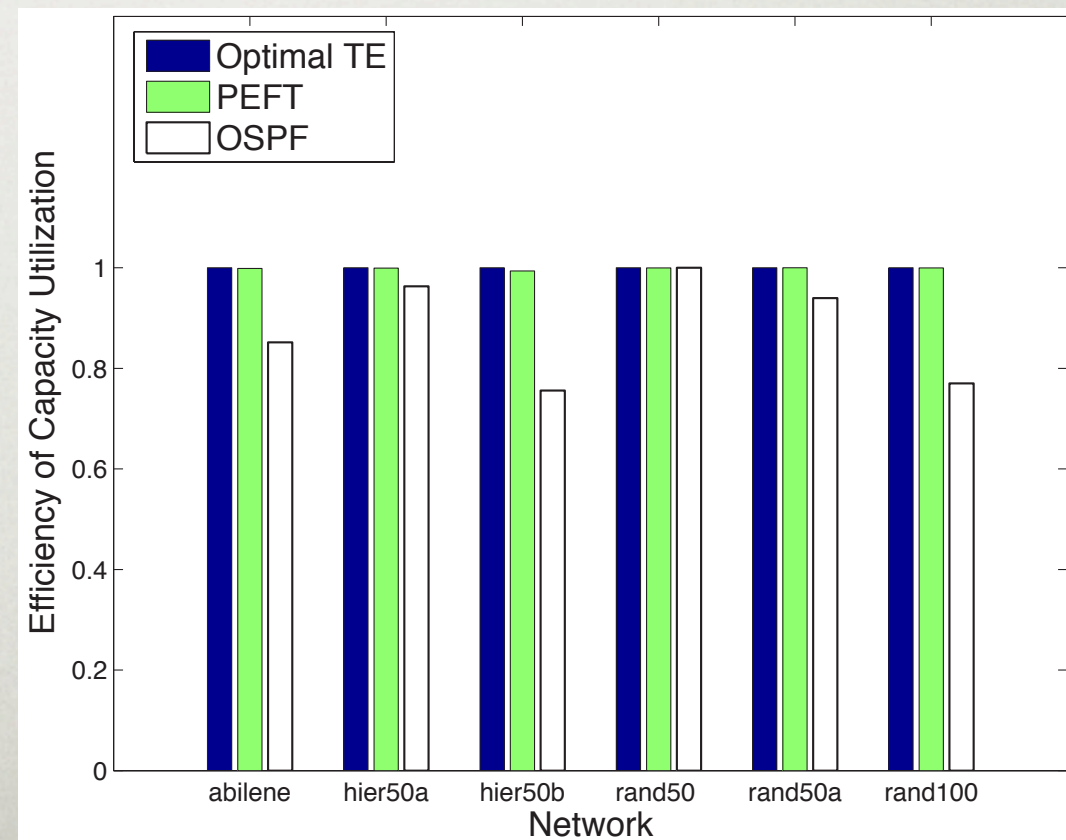
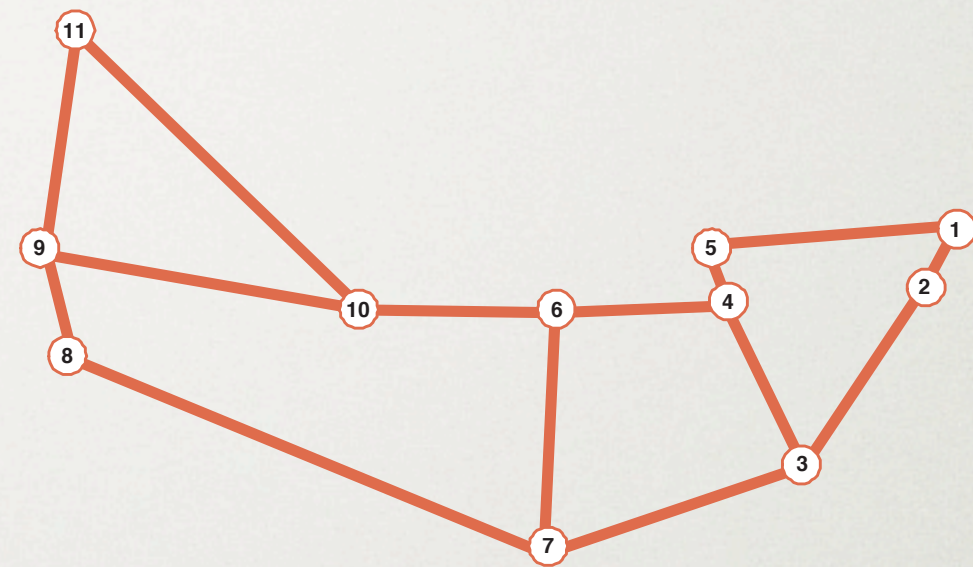
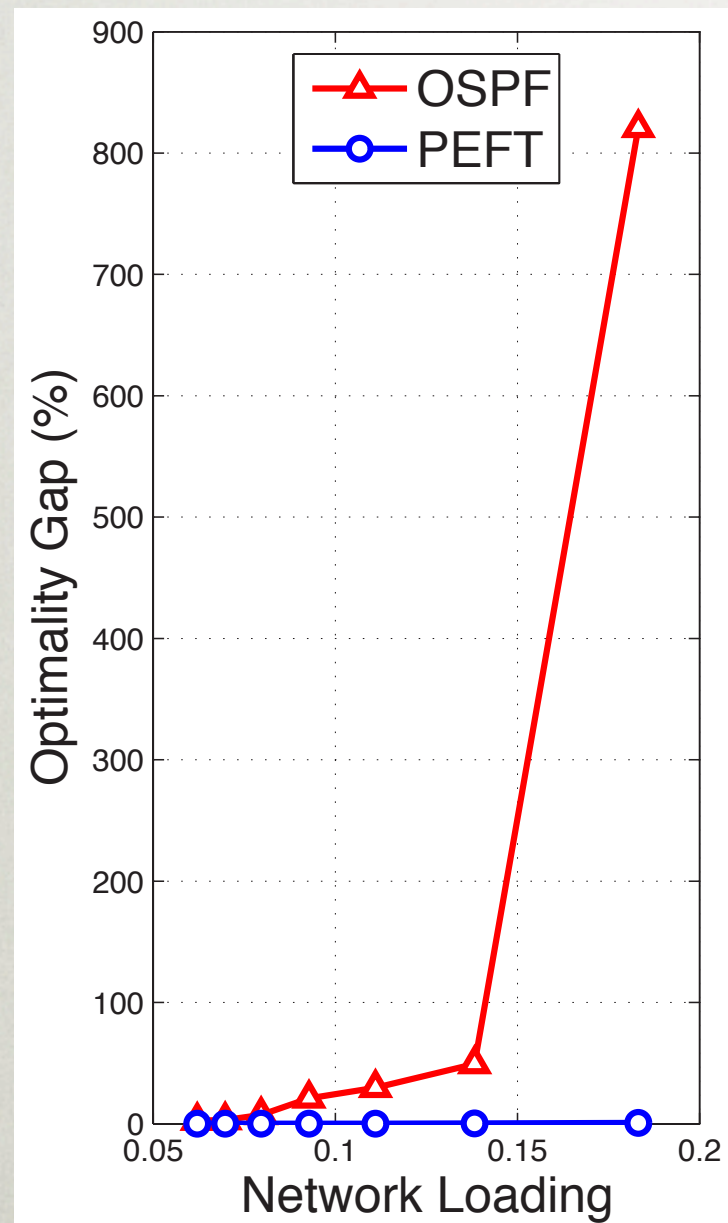
Entropy $z(x_{s,t}^i) = -x_{s,t}^i \log x_{s,t}^i$ for source-destination pair (s, t)

$$\begin{aligned} &\text{maximize} && \sum_{s,t} \left(D(s,t) \sum_{P_{s,t}^i} z(x_{s,t}^i) \right) \\ &\text{such that} && \sum_{s,t,i:(u,v) \in P_{s,t}^i} D(s,t) x_{s,t}^i \leq \tilde{c}_{u,v}, \forall (u,v) \\ &&& \sum_i x_{s,t}^i = 1, \forall (s,t) \\ &\text{variables} && x_{s,t}^i \geq 0. \end{aligned}$$

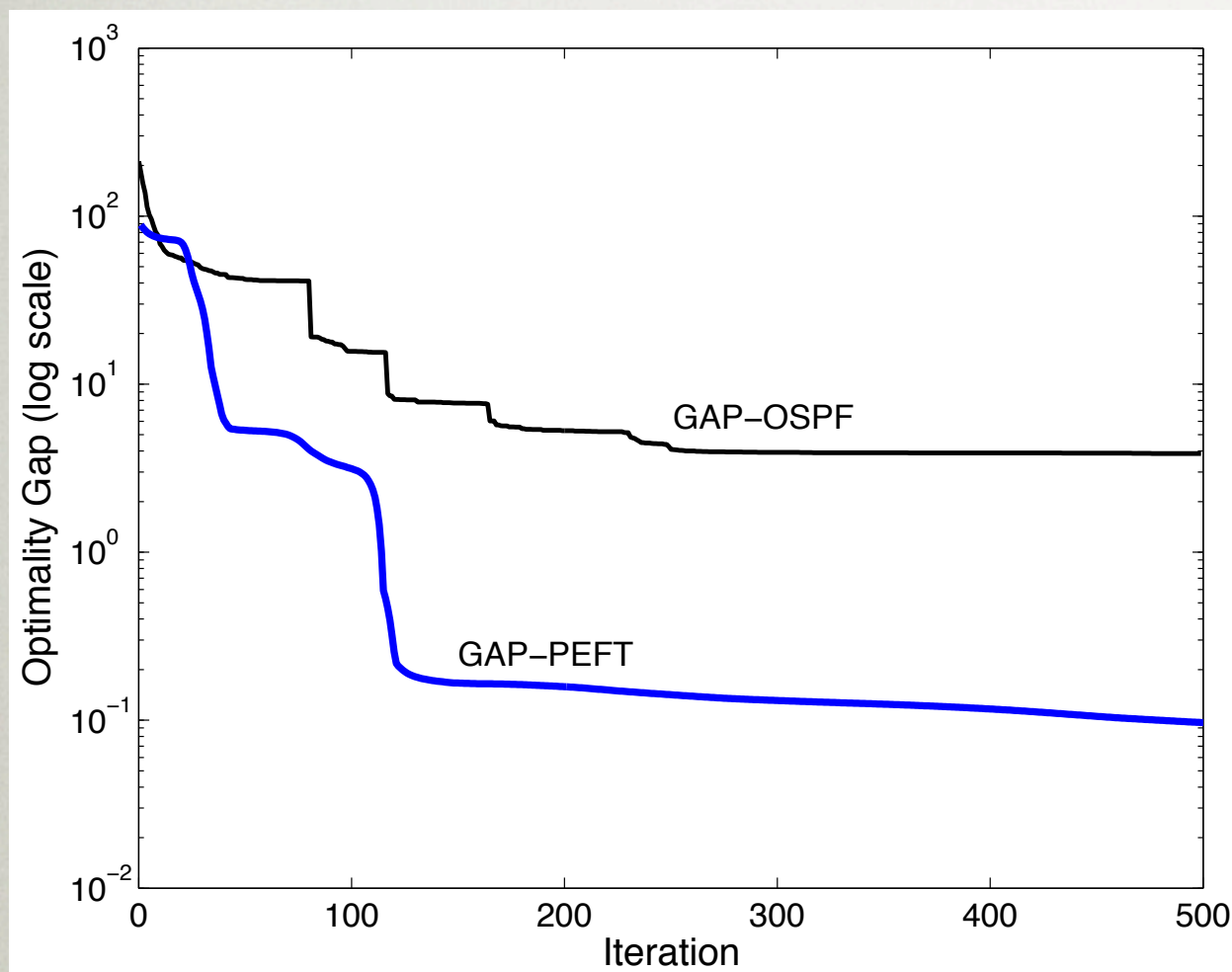
Characterization of optimality:

$$\frac{x_{s,t}^{i*}}{x_{s,t}^{j*}} = \frac{e^{-(\sum_{(u,v) \in P_{s,t}^i} w_{u,v})}}{e^{-(\sum_{(u,v) \in P_{s,t}^j} w_{u,v})}}$$

IP ROUTING: PERFORMANCE



IP ROUTING: EFFICIENCY



Abilene:

0.002s vs. 6s

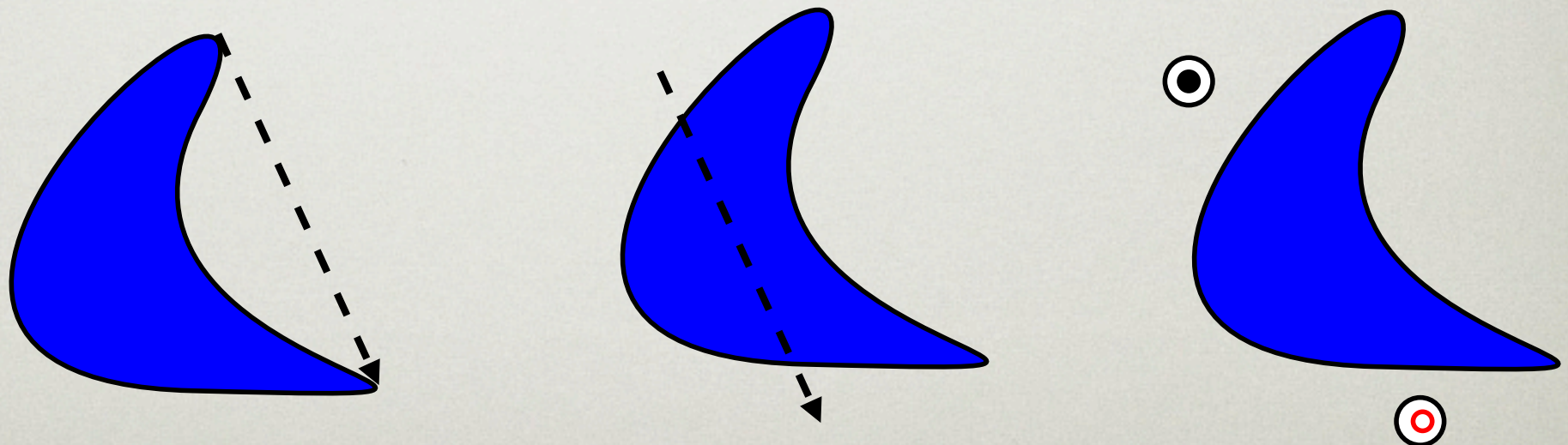
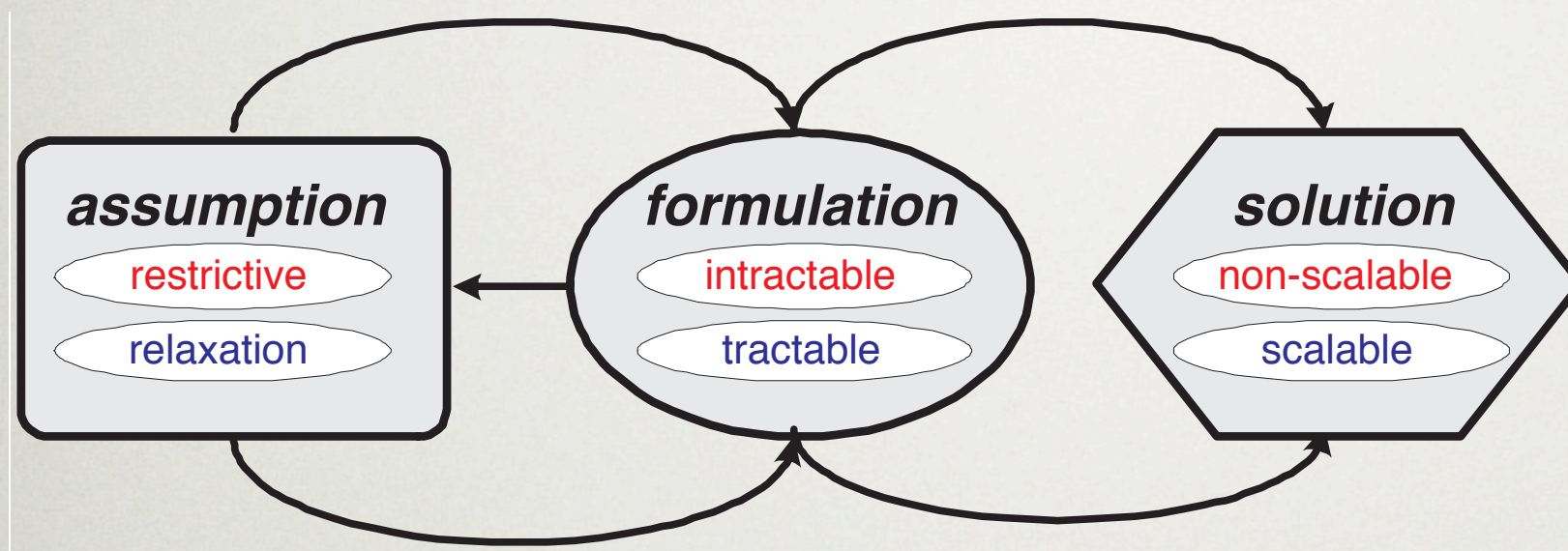
100 node 403 link: 0.042s vs. 39.5s

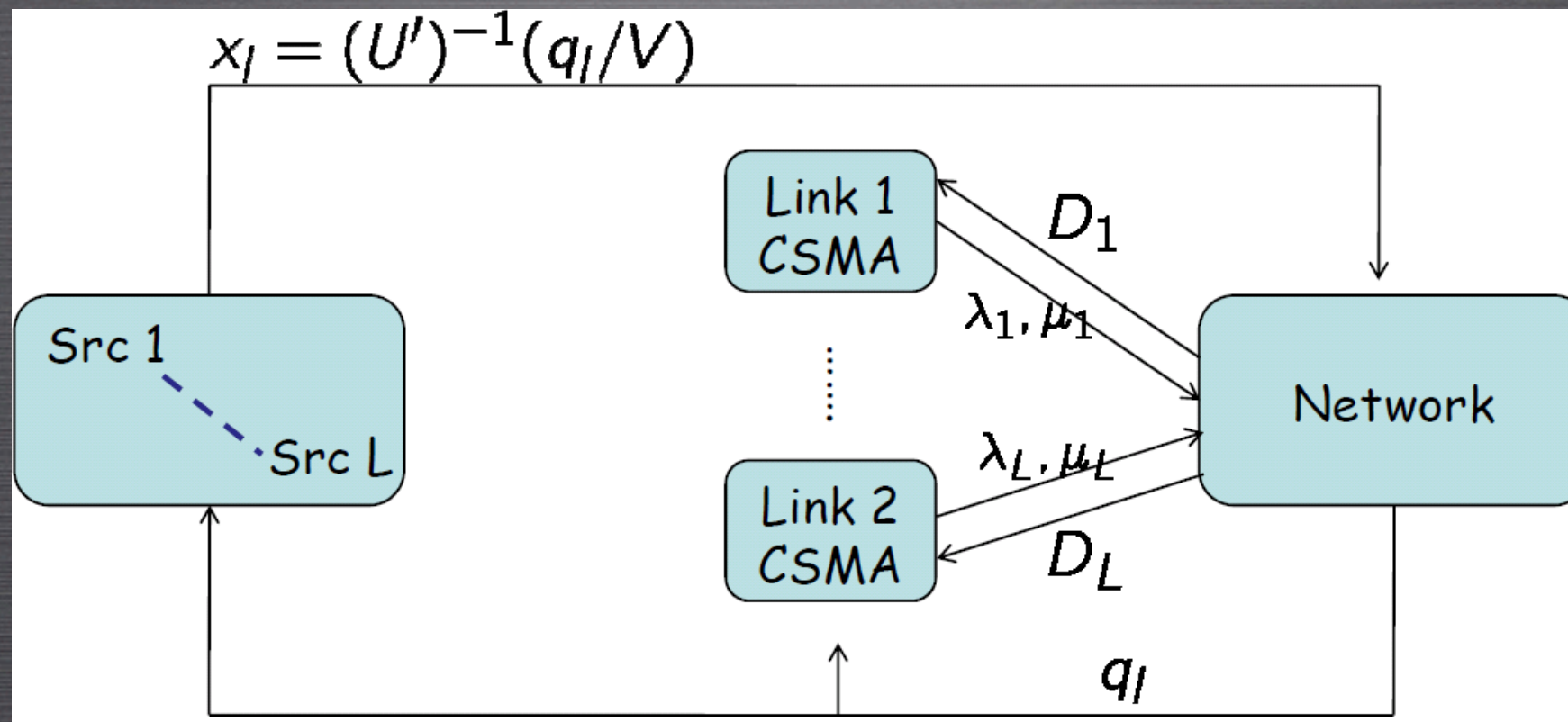
IP ROUTING: OPTIMAL AND SIMPLE

	Commodity Routing	Link-State Routing	
		OSPF	PEFT
Traffic Splitting	Arbitrary	Even	Exponential
Scalability	Low	High	High
Optimal TE	Yes	No	Yes
Complexity Class	Convex Optimization	NP Hard	Convex Optimization



IP ROUTING: DESIGN FOR OPTIMIZABILITY





WIRELESS SCHEDULING

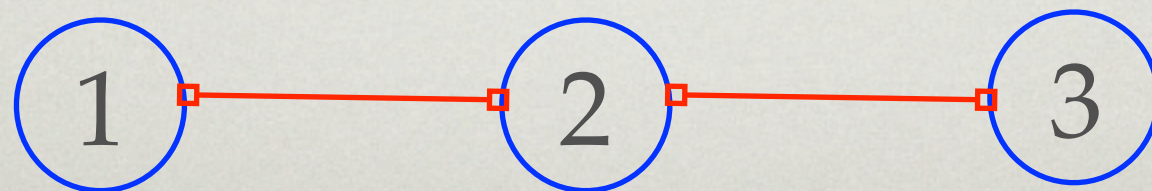
J. LIU, Y. YI, A. PROUTIERE, M. CHIANG, AND H. V. POOR, "TOWARDS UTILITY-OPTIMAL RANDOM ACCESS WITHOUT MESSAGE PASSING", SPECIAL ISSUE OF WILEY JOURNAL OF WIRELESS COMMUNICATION AND MOBILE COMPUTING, VOL. 10, NO. 1, PP. 115-128, JANUARY 2010

WIRELESS SCHEDULING: PROBLEM

Revisit interference in wireless networks

The other degree of freedom is “time”: **who talks when**

Interference (0-1 matrix):	A	maximize	$\sum_l U_l(x_l)$
Schedule (0-1 vector):	$\mathbf{s}$	subject to	$x_l \leq \sum_{\mathbf{s} \in \mathcal{S}: s_l=1} \pi_{\mathbf{s}}, \quad \forall l$
Set of feasible schedules:	$\mathcal{S}(A)$		$\pi_{\mathbf{s}} \geq 0, \quad \forall \mathbf{s}$
Time fraction of activation:	$\pi_{\mathbf{s}}$		$\sum_{\mathbf{s} \in \mathcal{S}} \pi_{\mathbf{s}} = 1$
Throughput:	x_l	variables	$\{x_l, \pi_{\mathbf{s}}\}$



WIRELESS SCHEDULING: HOW GOOD CAN CSMA BE?

CSMA: Carrier Sense Multiple Access:
When to contend, and How long to hold the channel



Adaptive CSMA without message passing:
Adjust contention and holding time (λ, μ)

Timescale separation assumption:
Network state converges to stationary distribution before
parameter update

Real system does **not** obey this assumption

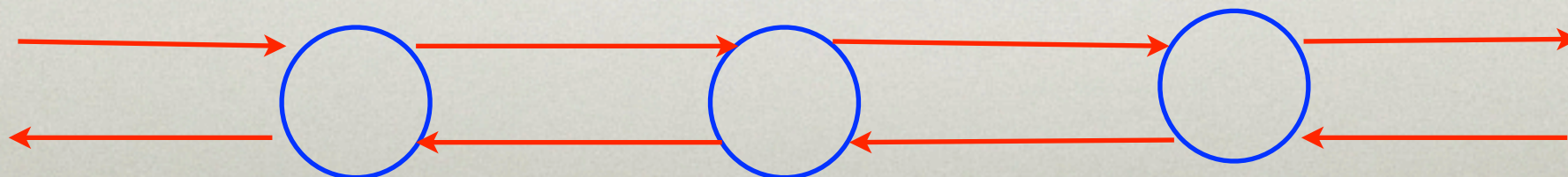
WIRELESS SCHEDULING: ALGORITHM

Update “virtual queue length” based on service rate
No message passing needed:

$$q_l[t + 1] = \left[q_l[t] + \frac{b[t]}{q_l[t]} \left(U_l'^{-1} \left(\frac{q_l[t]}{V} \right) - D_l[t] \right) \right]_{q_{min}}^{q_{max}}$$

Adjust Poisson contention rate or exponential holding time

$$\frac{\lambda_l[t + 1]}{\mu_l[t + 1]} = \exp(q_l[t + 1])$$



WIRELESS SCHEDULING: PERFORMANCE

Algorithm converges to $\lim_{t \rightarrow \infty} \mathbf{q}[t] = \mathbf{q}^*$ such that $\mathbf{x}(\mathbf{q}^*)$ solves

$$\begin{aligned} &\text{maximize} && V \sum_l U_l(x_l) - \sum_{\mathbf{s}} \pi_{\mathbf{s}} \log \pi_{\mathbf{s}} \\ &\text{subject to} && x_l \leq \sum_{\mathbf{s}: s_l=1} \pi_{\mathbf{s}}, \quad \forall l \\ &&& \pi_{\mathbf{s}} \geq 0, \quad \forall \mathbf{s} \\ &&& \sum_{\mathbf{s}} \pi_{\mathbf{s}} = 1 \end{aligned}$$

Approximation error **bounded** by $\log |\mathcal{S}|/V$

Pick V large enough and grows $\mathcal{O}(L)$

WIRELESS SCHEDULING: PROOF

A stochastic subgradient algo. modulated by a Markov chain

 Step 1: show averaging over fast timescale is valid

Interpolation of discrete q converges a.s. to a continuous q solving a system of ODE

 Step 2: show the resulting averaged process converges

The system of ODE describes the trajectory of subgradient solving the dual of the approximation problem

 Step 3: standard results in convex optimization and duality to show convergence and optimality

WIRELESS SCHEDULING: DISCRETE TIMESLOTS

- More realistic than Poisson clock model
- **Collision** (in addition to algorithmic inefficiency)
- Form a sequence of systems converging to Poisson model
- Scale both contention probability and channel holding time

Efficiency-Fairness Tradeoff:

Utility gap: δ
bound on suboptimality

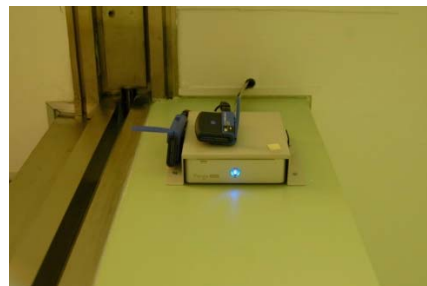
Short-term fairness: β
1 / ave. number of periods of
no transmission

$$\beta \leq \frac{\delta}{C_1 \exp(C_2/\delta)}$$

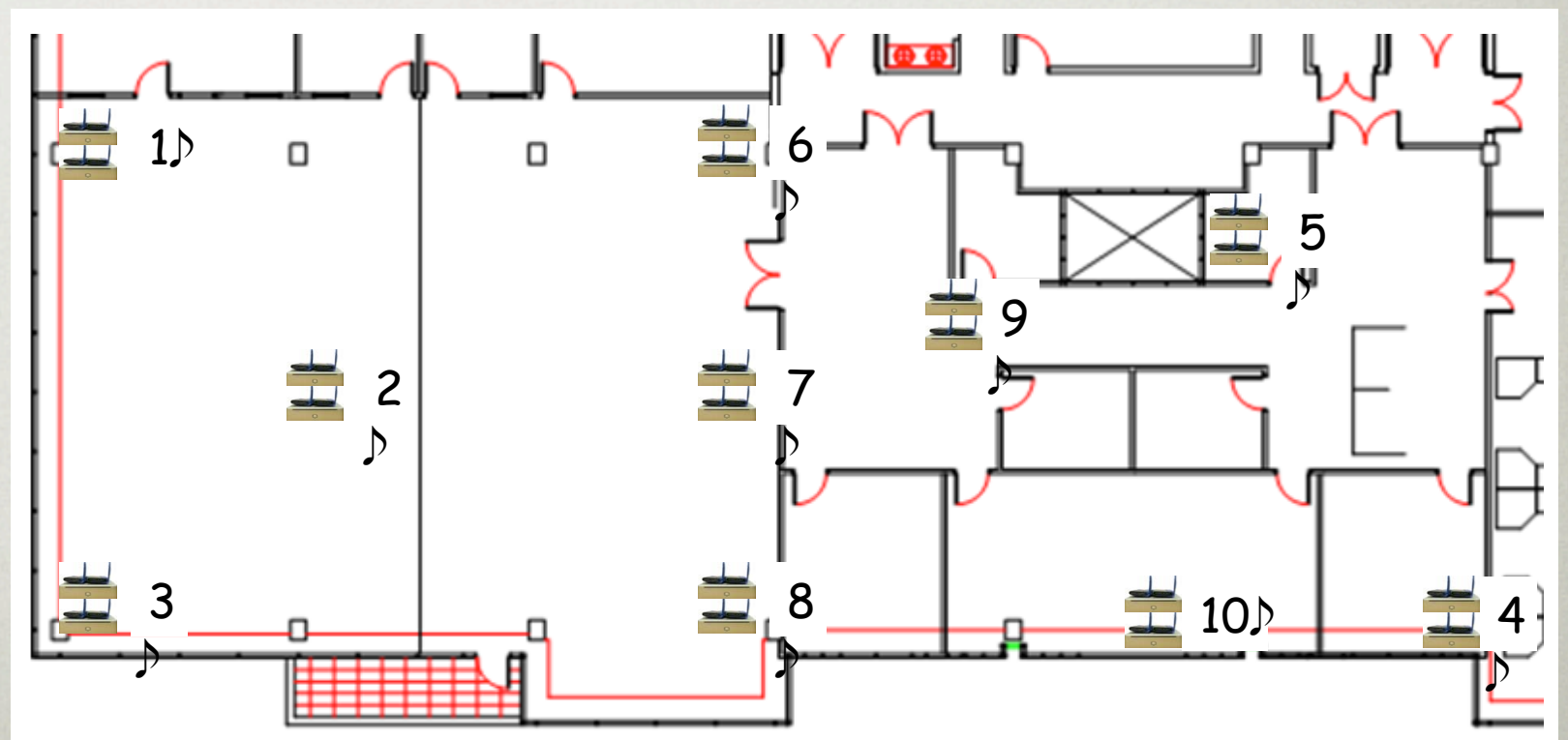
WIRELESS SCHEDULING: IMPLEMENTATION OVER WIFI



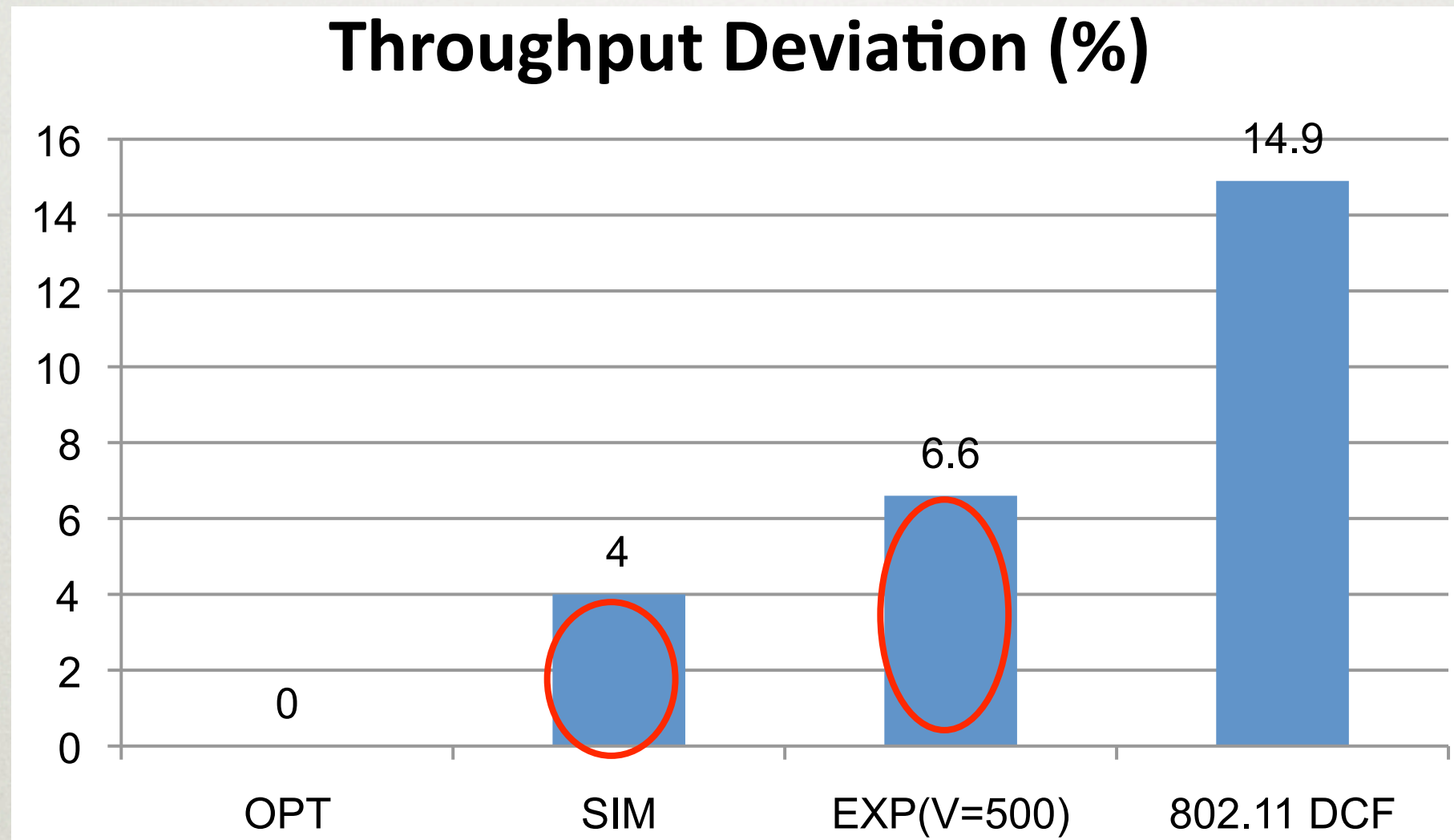
[Mesh Router]



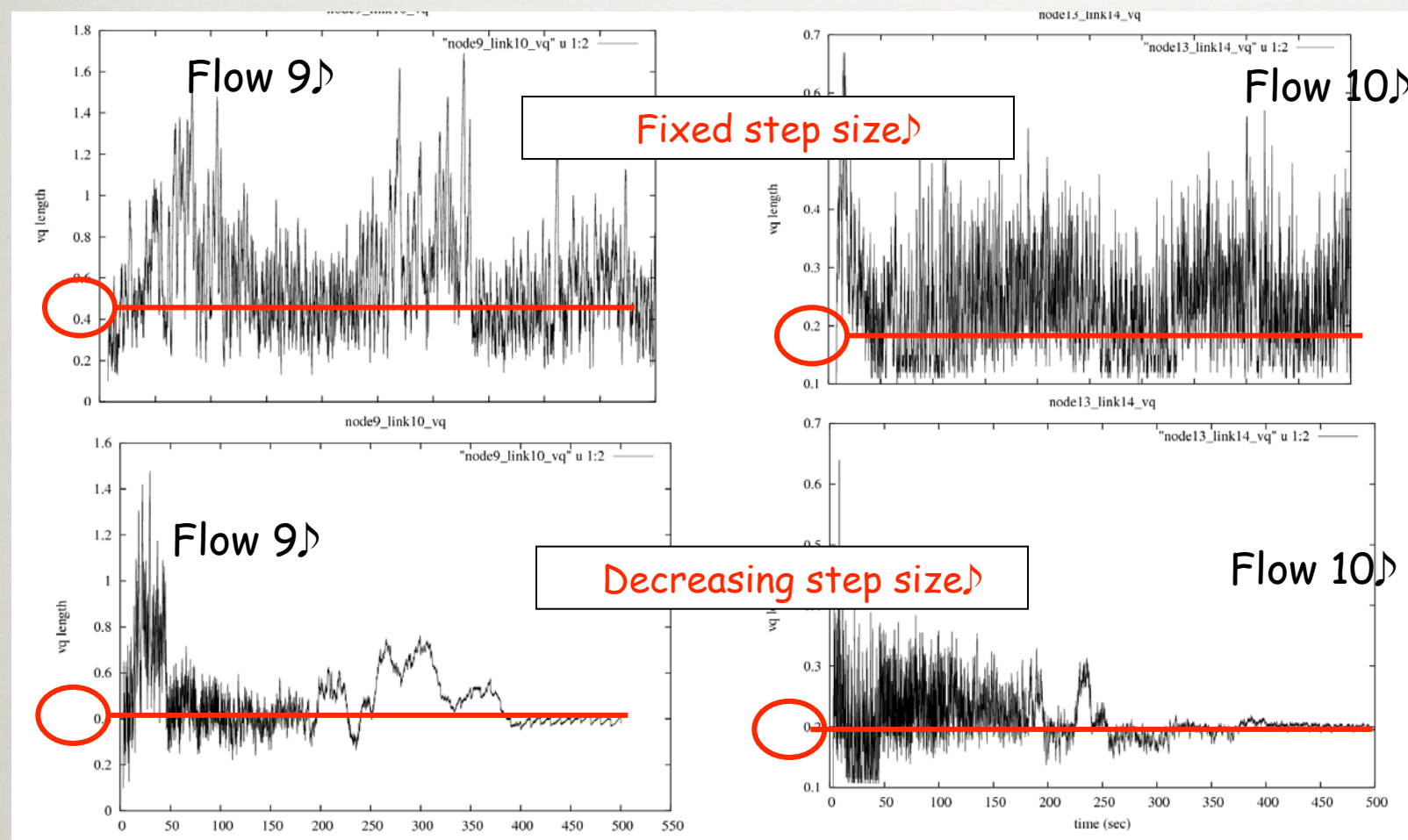
With Y. Yi and S.
Chong at KAIST



WIRELESS SCHEDULING: PERFORMANCE

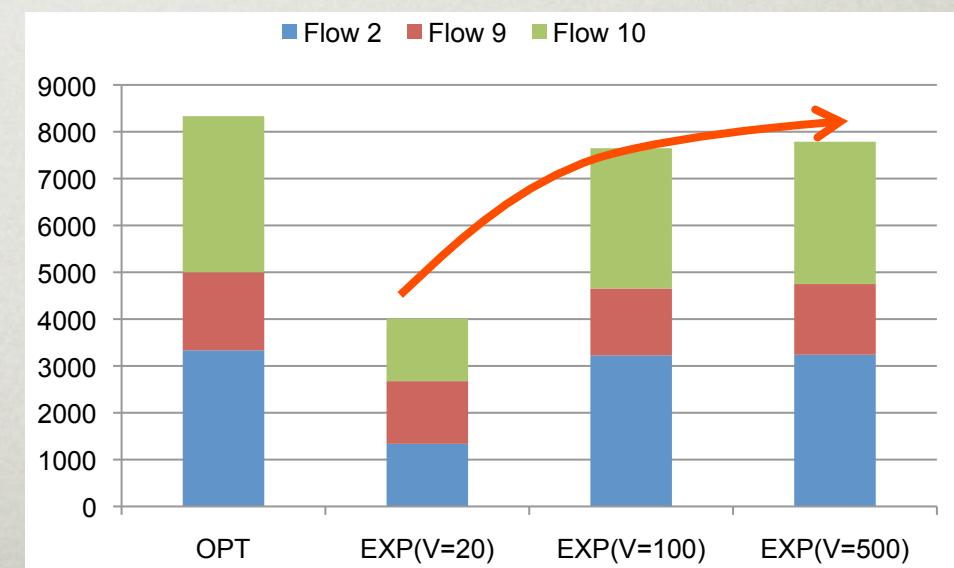


WIRELESS SCHEDULING: THEORY PREDICTIONS



Impact of V choice

Impact of stepsize choice



WIRELESS SCHEDULING: THEORY-PRACTICE GAPS

Theory ↔ Simulation ↔ Experiment ↔ Legacy

- 📌 **Interference:** asymmetric
- 📌 **Sensing:** imperfect
- 📌 **Receiving:** SIR based
- 📌 **Holding:** imperfect

📌 **Assumed away:**

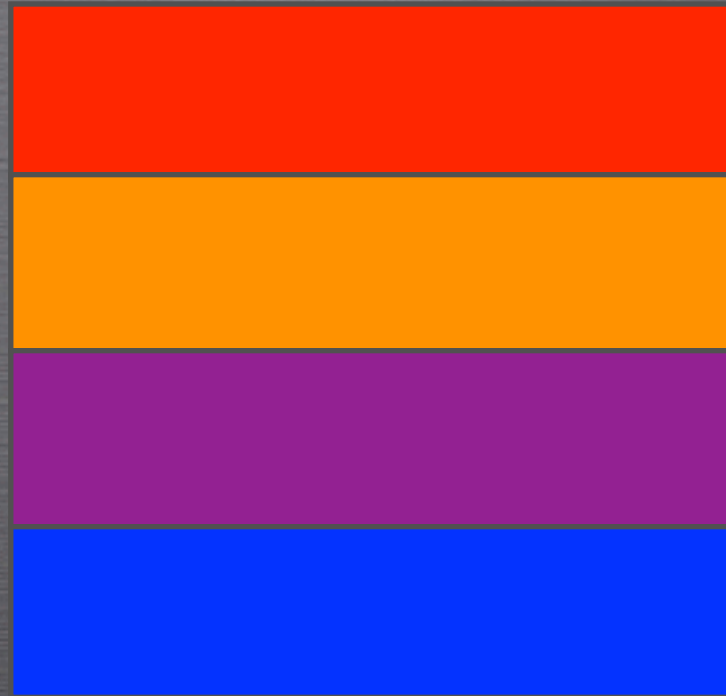
overhead, asymmetry, control granularity

📌 **Modeled simplistically:**

imperfect holding and sensing
SIR collision model with capture

📌 **Analyzed loosely:**

convergence speed
transient behavior
parameter choice

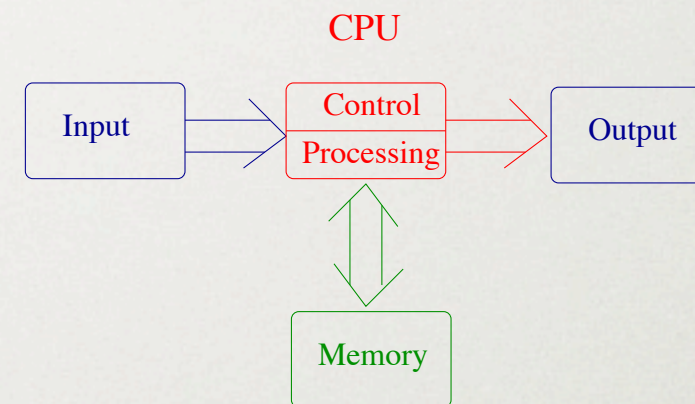
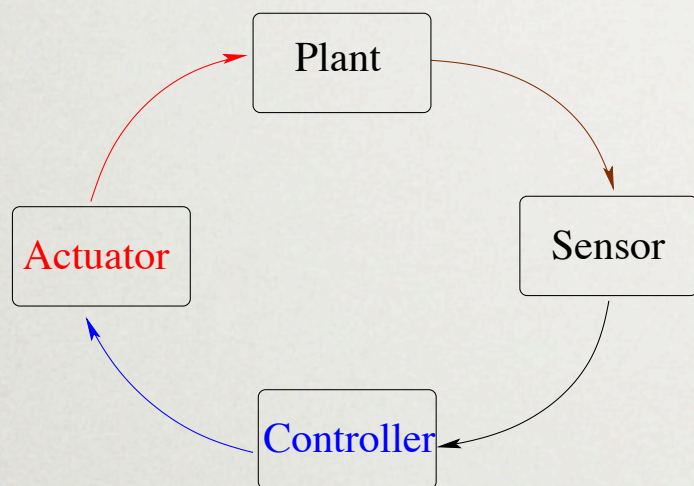


NETWORK ARCHITECTURE

M. CHIANG, S. H. LOW, A. R. CALDERBANK, J. C. DOYLE, "LAYERING AS
OPTIMIZATION DECOMPOSITION: A MATHEMATICAL THEORY OF NETWORK
ARCHITECTURE", PROCEEDINGS OF THE IEEE, VOL. 57, NO. 1, PP. 255-312,
JANUARY 2007

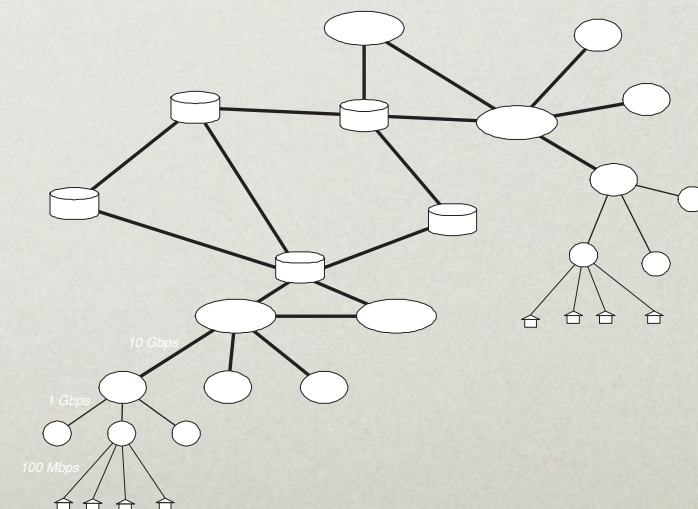
NETWORK ARCHITECTURE: ANALYTIC FOUNDATIONS

Architectures well-understood in control and computation



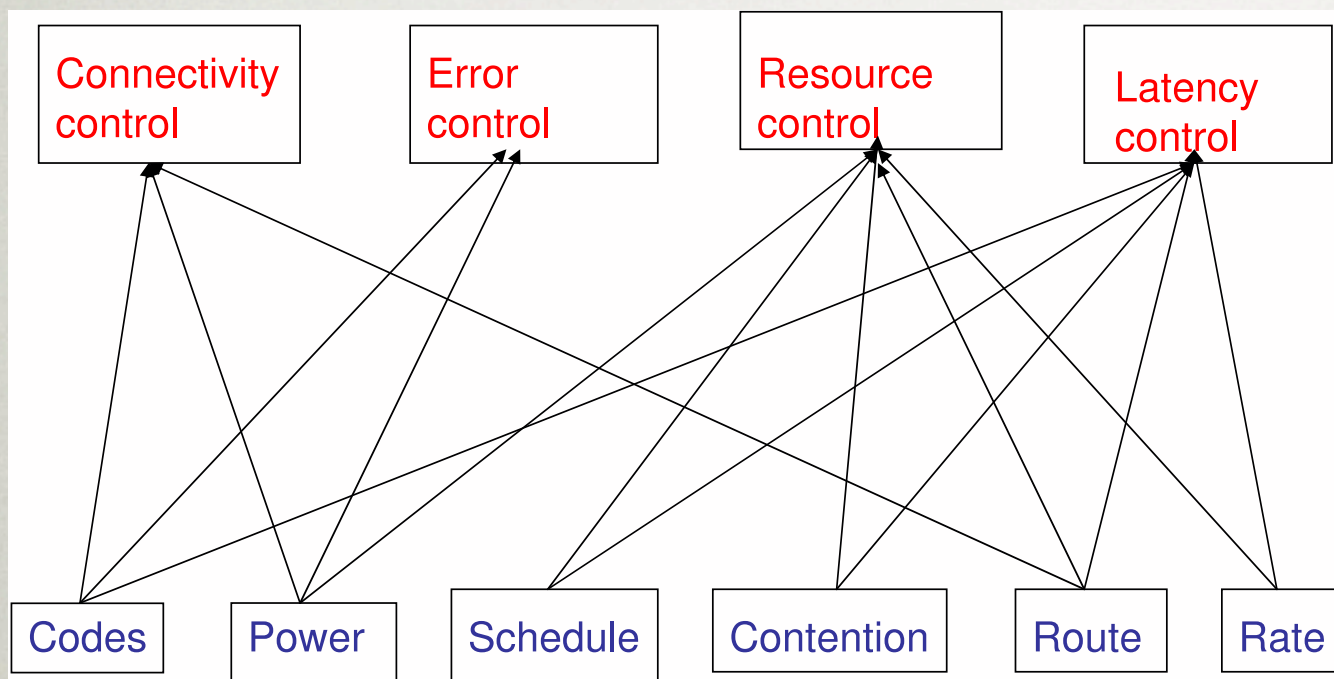
What about network architecture?

Application
Presentation
Session
Transport
Network
Link
Physical



NETWORK ARCHITECTURE: FUNCTIONALITY ALLOCATION

Who should do what and how to connect them



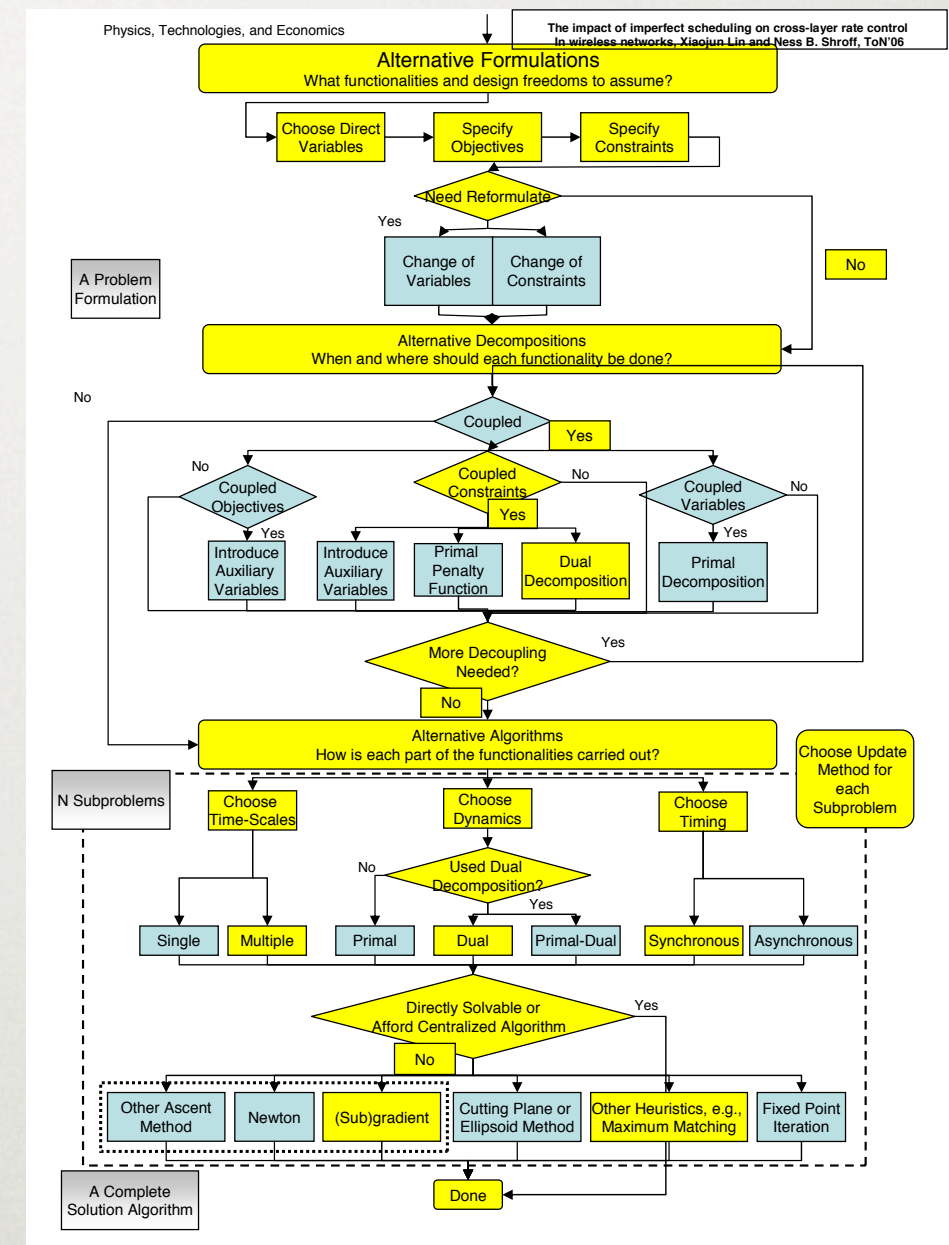
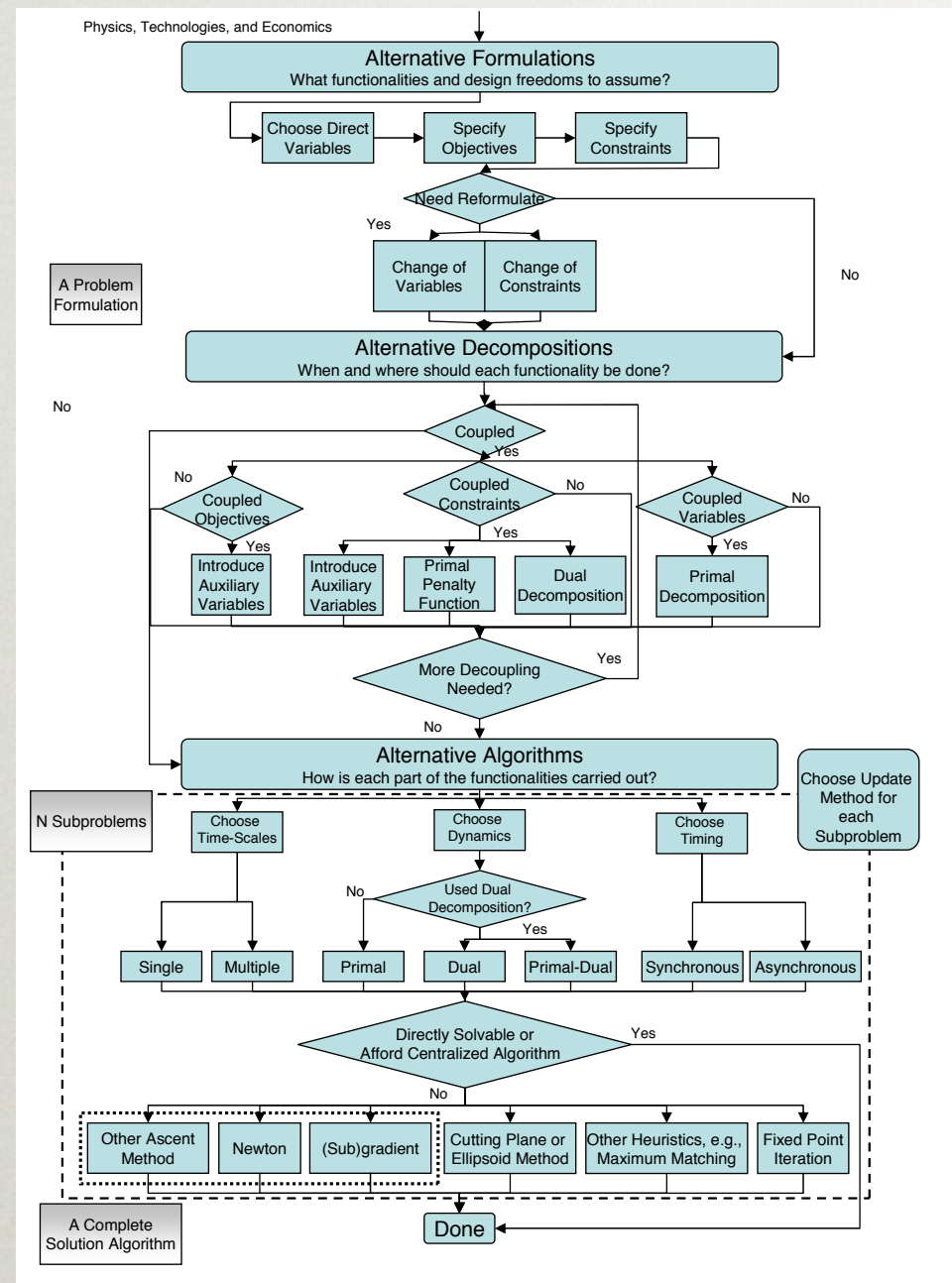
Network: Generalized NUM

Layering: Decomposition

Layers: Decomposed subproblems

Interfaces: Functions of primal/dual var.

NETWORK ARCHITECTURE: LAYERING AS DECOMPOSITION



<http://num.ie.cuhk.edu.hk>

NOT JUST A HAMMER

DESCRIPTIVE -> EXPLANATORY MODEL

REVERSE ENGINEERING NETWORK AS OPTIMIZER

GET TO THE ROOT OF KNOWLEDGE TREE

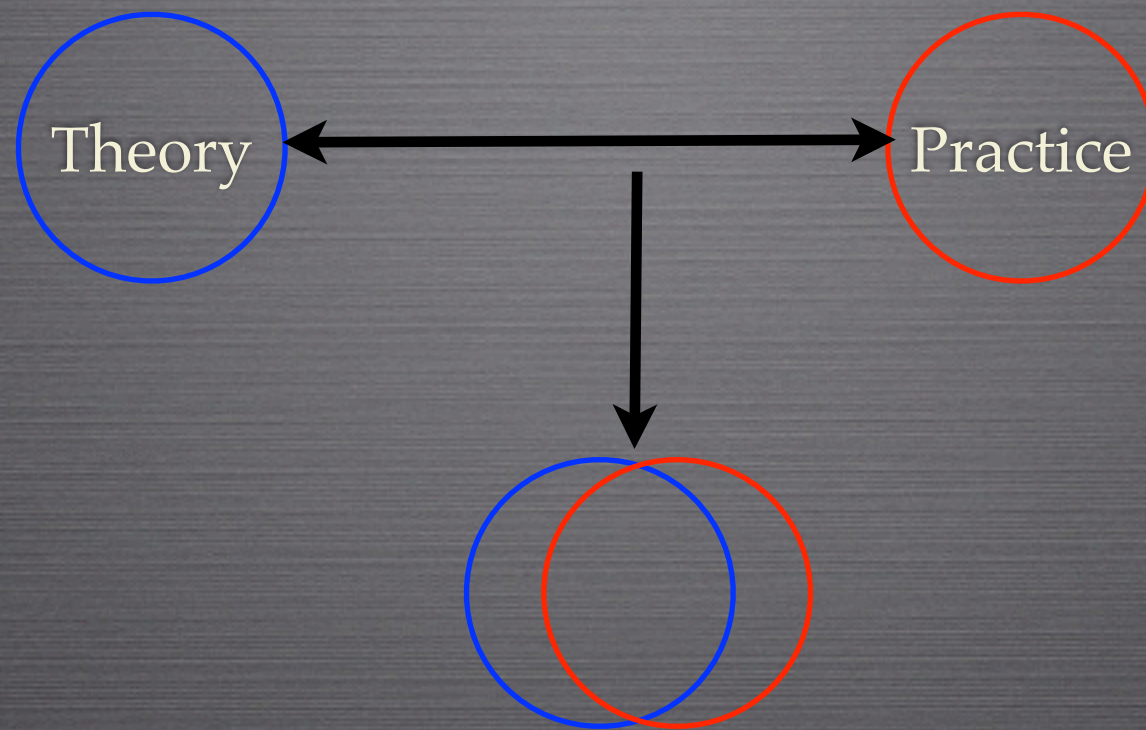


TOP-DOWN FIRST-PRINCIPLED DESIGN

NEW ANGLES ON NETWORKING RESEARCH

OPTIMIZATION BEYOND OPTIMALITY

MODELING
ARCHITECTURE
ROBUSTNESS
DESIGN FOR OPTIMIZABILITY



[HTTP://SCENIC.PRINCETON.EDU](http://scenic.princeton.edu)



ACKNOWLEDGEMENTS

COAUTHORS

AFOSR, DARPA, NSF, ONR

AT&T, CISCO, , MICROSOFT, QUALCOMM

**ALL PAPERS DOWNLOADABLE:
WWW.PRINCETON.EDU/~CHIANGM**