Interference-Aware Wireless Network Optimization

Yin Zhang
University of Texas at Austin
yzhang@cs.utexas.edu

Joint work with Lili Qiu and Ratul Mahajan

IPAM Internet MRA Workshops IV
11/17/2008

Motivation

- Wireless networks are becoming ubiquitous
- Wireless networks are examples of highly dynamic graphs
  - What is a link?
  - Nearby nodes interfere with each other
  - Link characteristics (e.g. loss rate) are time varying, traffic dependent, environment dependent

Wireless vs. Wireline Management

- Managing wireless networks is hard

	Wireline	Wireless
Perform what-if analysis	✓	✗
Predict if given sending rates are achievable	✓	✗
Optimize sending rates for different objectives	✓	✗
Route optimization	✓	✗

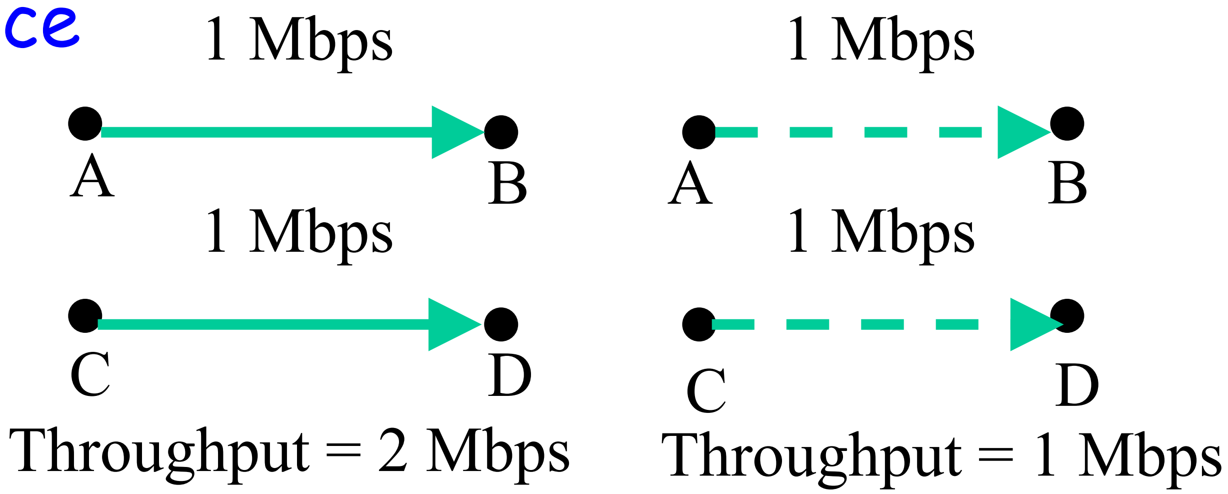
Mission Statement:

Make wireless management on par with wireline networks

Want *systematic*, *predictable* solution

Challenge: Wireless Network Management

- Interference is critical to wireless network performance



- The impact of interference was not well understood
- Make wireless network optimization/control hard
  - Often have to resort to indirect metrics to optimize

State of the Art

- Asymptotic performance bounds [GP00,LB+01,GT01,GV02]
  - Cannot model any specific networks
- Non-measurement based interference models
  - Inaccurate for real networks
    - Interference range is twice the communication range
  - Restricted scenarios
    - Single-hop networks (e.g., Bianchi's model)
    - Multihop networks with 2-flows (e.g., Garetto et al.)
- Direct interference measurement
  - Lack scalability and predictive power
- Measurement-based interference model [Reis06]
  - Promising to achieve both accuracy and scalability
  - Restricted scenario: only two saturated broadcast senders

Need a better measurement-based model!

Roadmap

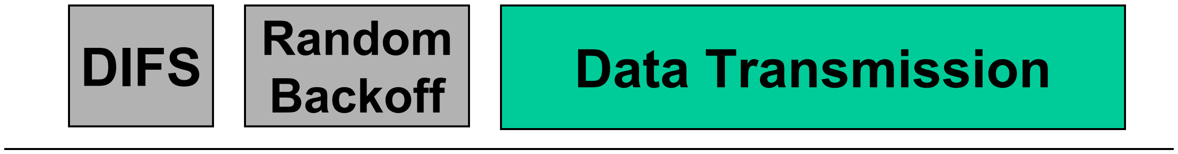
- Detailed model of 802.11 networks [mobicom07]
  - Focus: accurate what-if analysis
- Simple unicast model of 802.11 networks and its application to network optimization [sigcomm08]
  - Focus: simple model + predictable optimization

Introduction

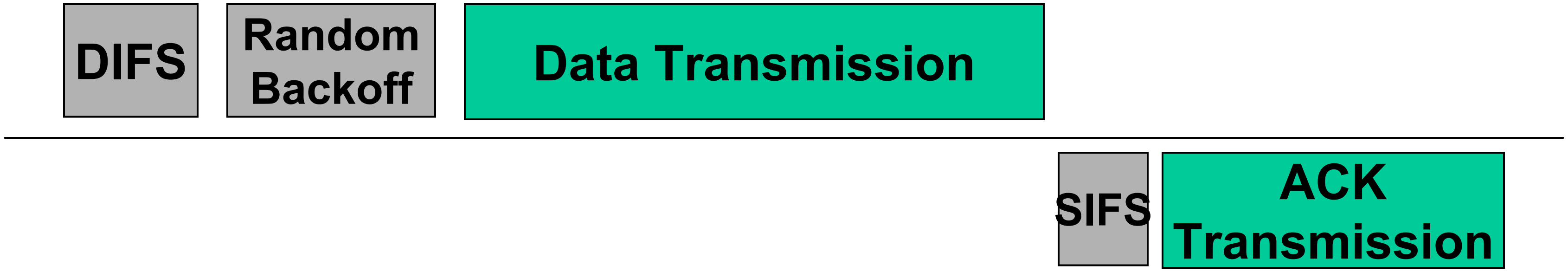
- A general interference model for IEEE 802.11 multihop wireless networks
  - Models non-binary interference among an arbitrary number of senders
  - Models both broadcast and unicast traffic
  - Models both saturated and unsaturated demands
- Easy to seed
  - Requires only O(N) broadcast measurements
- Highly accurate
  - Validated through experiments and simulations

Background on IEEE 802.11 DCF

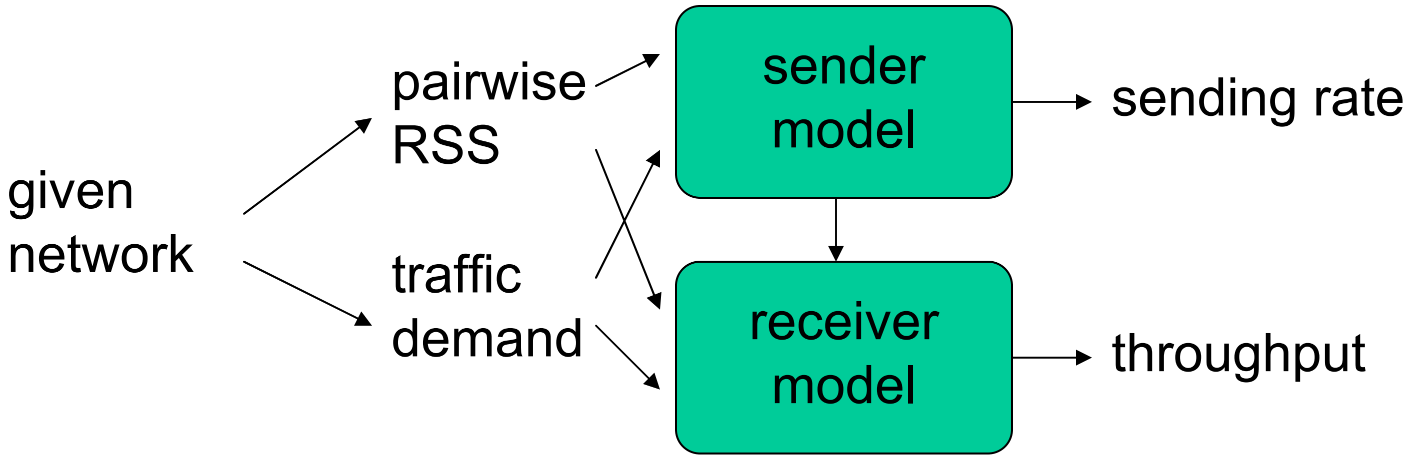
- Broadcast
  - If medium is idle for DIFS, transmit immediately
  - Otherwise, wait for DIFS and a random backoff between [0, CWmin]


DIFS
Random Backoff
Data Transmission

- Unicast
  - Use ACKs and retransmissions for reliability
  - Binary backoff
    - CW doubles after each failed transmission until CWmax
    - Restore CW to CWmin after a successful transmission or maxRetry count is reached


DIFS
Random Backoff
Data Transmission
SIFS
ACK Transmission

Model Overview

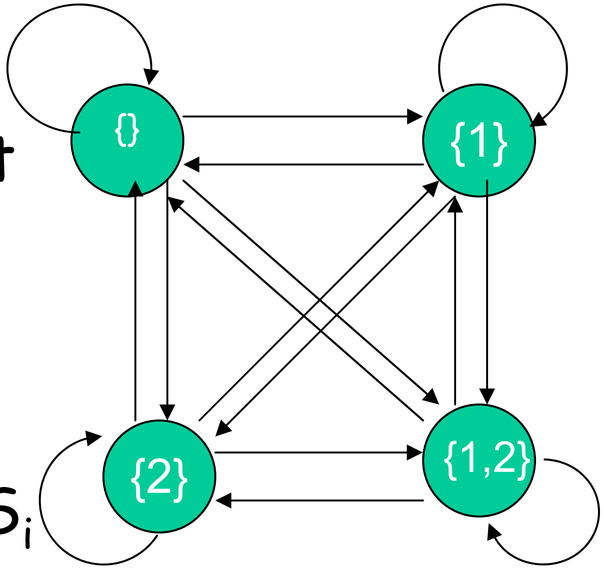


- How it works
  - Measure pairwise RSS via broadcast probes
    - One node broadcasts at a time, other nodes measure RSS
      ➔ only requiring O(n) probes
  - Use sender/receiver models to get sending rate/throughput
- Basic model: broadcast traffic
- Extension to unicast traffic
- Main application: What-if analysis

Assumptions

- A sender can transmit if
  - The total energy it received ≤ CCA threshold
    - CCA: clear-channel assessment
- A receiver correctly receives a transmission if its
  - RSS ≥ radio sensitivity
  - SINR ≥ SINR threshold
    - SINR: signal to interference-plus-noise ratio (= S/(I + N))
    - Can easily extend to BER-based model
- Assumes 1-hop traffic demands
  - Traffic is only sent over 1 hop and not routed further
  - Multi-hop demands need to be first mapped to 1-hop demands based on routing

Broadcast Sender: A Markov Model



- Challenge
  - Sender behavior depends on the set of nodes currently transmitting
- Solution
  - Associate a state i with every possible set of transmitting nodes S_i
  - Enhance scalability by pruning low-probability states and transitions (see paper)
- Algorithm
  1. Compute individual node's mode transitions in each state
  2. Compute state transition probabilities M(i,j)
  3. Compute stationary probabilities π_i by solving LP
  4. Compute throughput of node m: $t_m = \sum_{i:\, m \in Si} \pi_i$

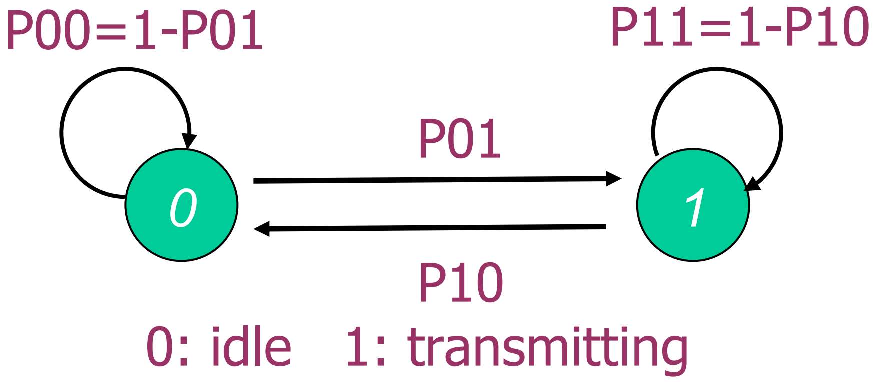
Individual Node Mode Transitions

$$P_{10}(n \mid S_i) = \frac{1}{\#\text{slots per tx}}$$

$$P_{01}(m \mid S_i)$$

$$= \Pr[\text{medium is clear } \& \text{ counter} = 0 \ \& \text{ m has data} \mid \text{S}_\text{i}]$$

$$= C(m|S_i) \times \frac{1}{CW/2 + DIFS} \times Q(m)$$



- $\Pr[\text{medium is clear} \mid \text{S}_\text{i}] = C(m \mid S_i) = \Pr[I_{m|S_i} \le \beta_m]$
  - $I_{m|Si}=W_m+B_m+\sum_{s\in Si\backslash\{m\}} R_{sm}$ (total interference at m caused by senders S_i)
  - Each term is modeled as a lognormal r.v. (validated experimentally)
  - Approximate the sum using a single moment-matching lognormal r.v.
- $\Pr[\text{counter} = 0 \mid \text{medium is clear}] = \dfrac{1}{CW/2 + DIFS}$
- $\text{Q(m)} = \Pr[\text{m has data} \mid \text{medium is clear } \& \text{ counter} = 0]$
  - Q(m) = 1 for saturated demands
  - Q(m) is estimated iteratively for unsaturated demands

State Transition Probabilities

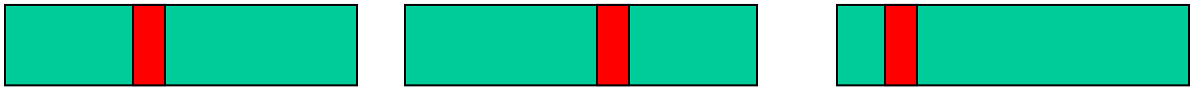
- Case 1: packet sizes are exponentially distributed
  - Different nodes' mode transitions are independent
  - M(i, j) = Π_n P{n's mode in state i → n's mode in state j}
- Case 2: packet sizes are similar
  - Mode transitions are dependent due to synchronization
    - Two nodes are in sync iff $C(m|\{n\}) < 0.1$ and $C(n|\{m\}) < 0.1$
    - A and B are in sync and have overlapping transmissions
      ➔ their transmissions start and end at the same time
  - Solution: fate sharing
    - m and n are in sync and both active in state i
      ➔ they have the same mode in next state
    - Probability for a group of k nodes in sync to all transition from 1 to 0 is $\frac{1}{\#\text{slots per tx}}$ instead of $\left(\frac{1}{\#\text{slots per tx}}\right)^k$

Broadcast Receiver

$$l^{rss}_{mn|Si} = \Pr[R_{mn} < \gamma_n]$$

$$l_{mn|S_i} = \Pr\{\frac{R_{mn}}{I_{mn|S_i}} < \delta_n\}$$

- Estimate slot-level loss rate
  - Loss due to low RSS
    - Based on pairwise RSS measurement
  - Loss due to low SINR
    - Our measurements show that RSS follows log-normal distribution
    - Approximate $\frac{R_{mn}}{I_{mn|S_i}}$ with a single moment-matching lognormal r.v.
- How to get packet-level loss rate?
  - Pkt loss rate ≠ slot-level loss rate under partial collisions (common in hidden terminal)



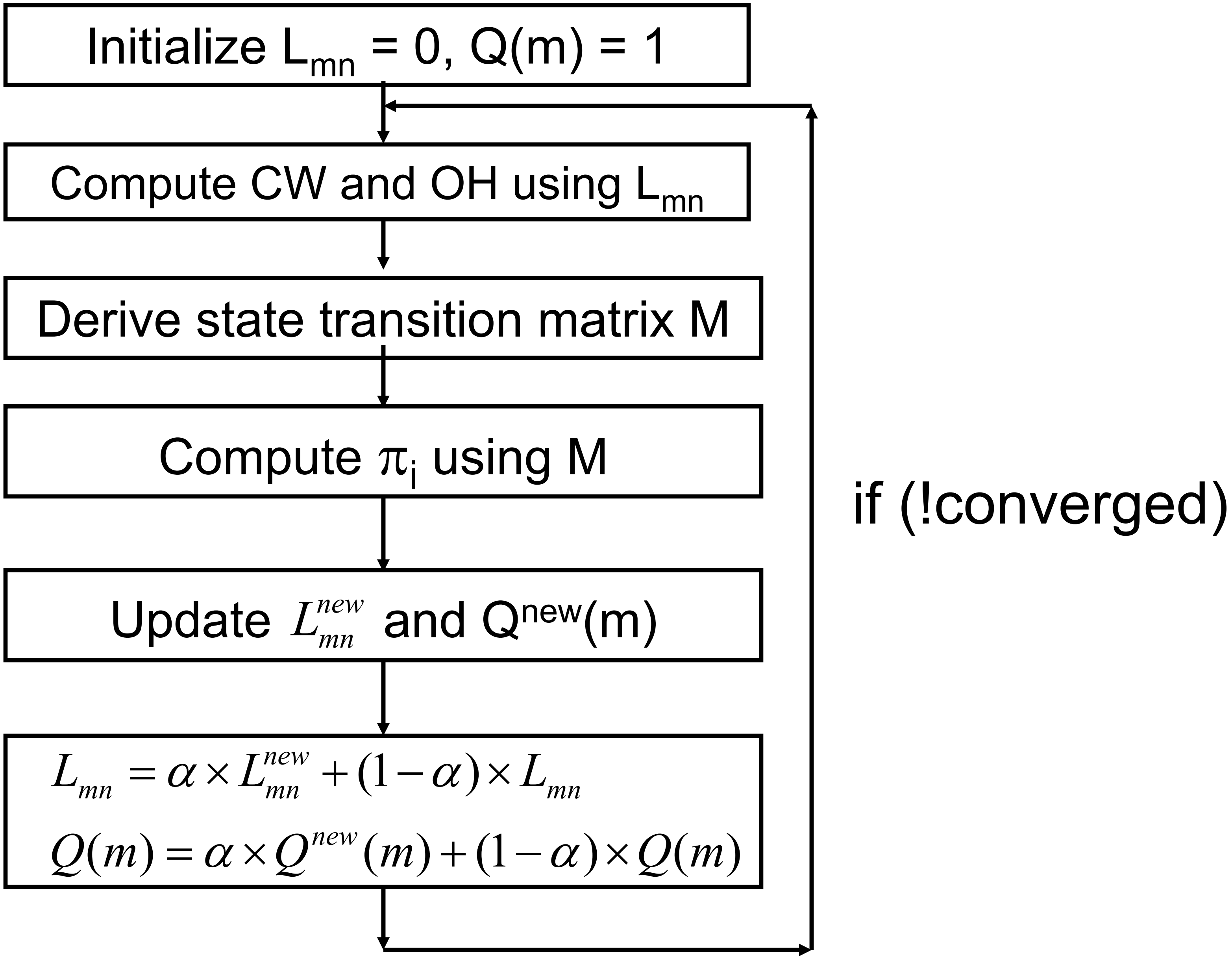
Slot loss=10%
Pkt loss = 100%

  - Differentiate losses due to synchronous collisions and asynchronous collisions
    - Synchronous collisions when a node is synchronized with at least one other node in the state
    - Otherwise, asynchronous collisions

Extensions to Unicast Demands

- Sender side extensions
  - Compute average CW based on binary backoff
  - Incorporate ACK/SIFS overhead (in addition to DIFS)
  - Derive Q(m) to ensure demands (w/ retx) not exceeded
- Receiver side extensions
  - Include low RSS induced losses for both data and ACK
  - Include low SINR induced loss due to collisions between data/ACK, ACK/data, ACK/ACK (in addition to data/data)
- Challenge
  - Inter-dependency between sending rate and loss rates
    - Sending rates depend on loss rates due to binary backoff
    - Loss rates depend on sending rate due to interference
- Solution: use an iterative framework

Unicast Model: Iterative Framework



Initialize $L_{mn} = 0$, $Q(m) = 1$

Compute CW and OH using L_{mn}

Derive state transition matrix M

Compute π_i using M

Update L_{mn}^{new} and $Q^{new}(m)$

$$L_{mn} = \alpha \times L_{mn}^{new} + (1-\alpha) \times L_{mn}$$

$$Q(m) = \alpha \times Q^{new}(m) + (1-\alpha) \times Q(m)$$

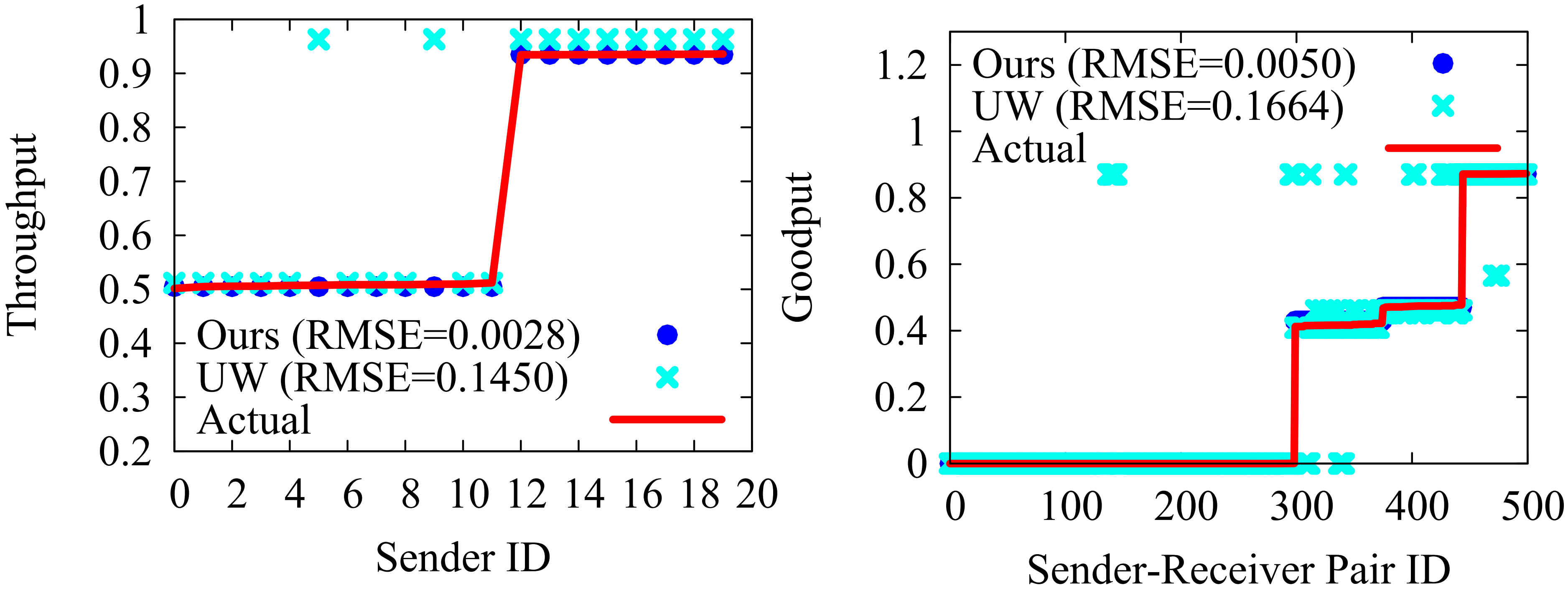
if (!converged)

Evaluation Methodology

- Qualnet simulation
  - Controlled environment and direct assessment of individual components in our model
  - Vary topologies, # senders, demand types, freq. band
- Testbed experiments
  - More realistic scenarios
    - RF fluctuation, measurement errors, and variation across hardware
  - UT traces
    - 22-node, 802.11 a/b/g NetGear WAG511, Madwifi, click
    - Vary # senders, demand types
  - UW traces (Reis et al.)
    - 14-node testbed inside an office building
    - 2 saturated broadcast sender traces

Simulation Evaluation: Saturated Broadcast

2 saturated broadcast



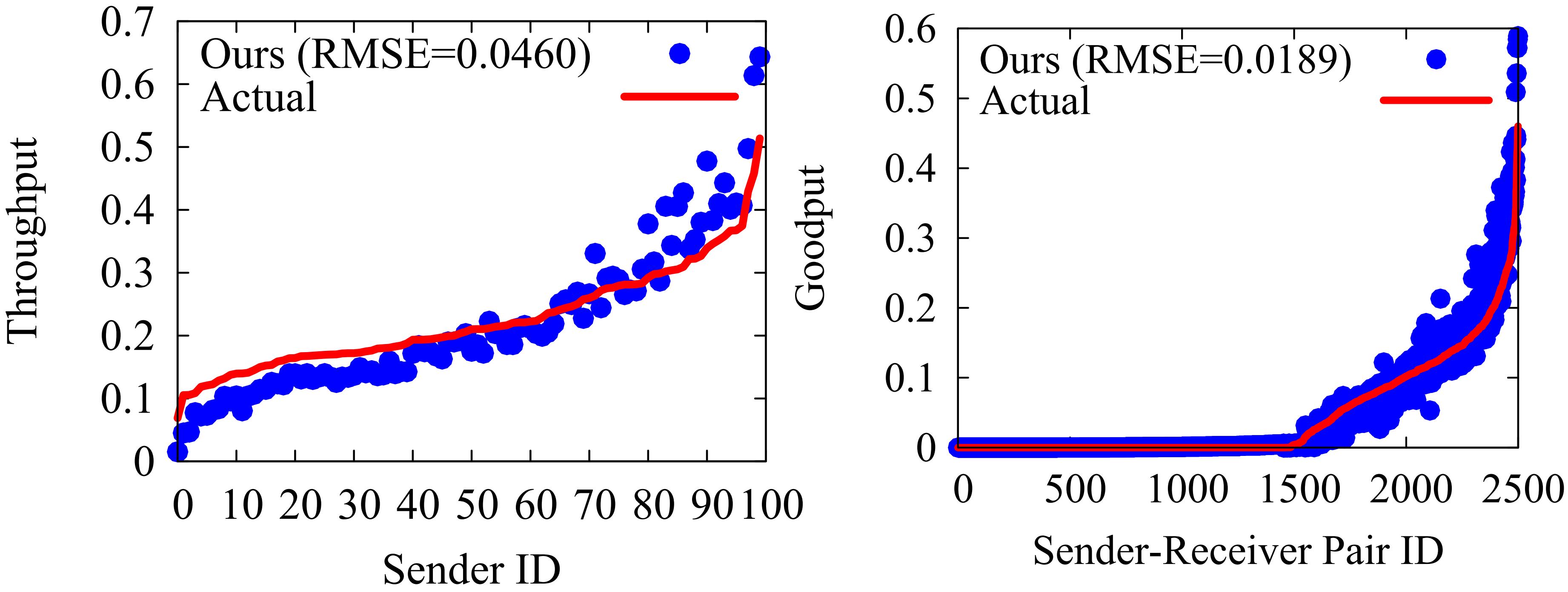
(a) sending rate

(b) throughput

More accurate than UW 2-node model

Simulation Evaluation: Saturated Broadcast

10 saturated broadcast



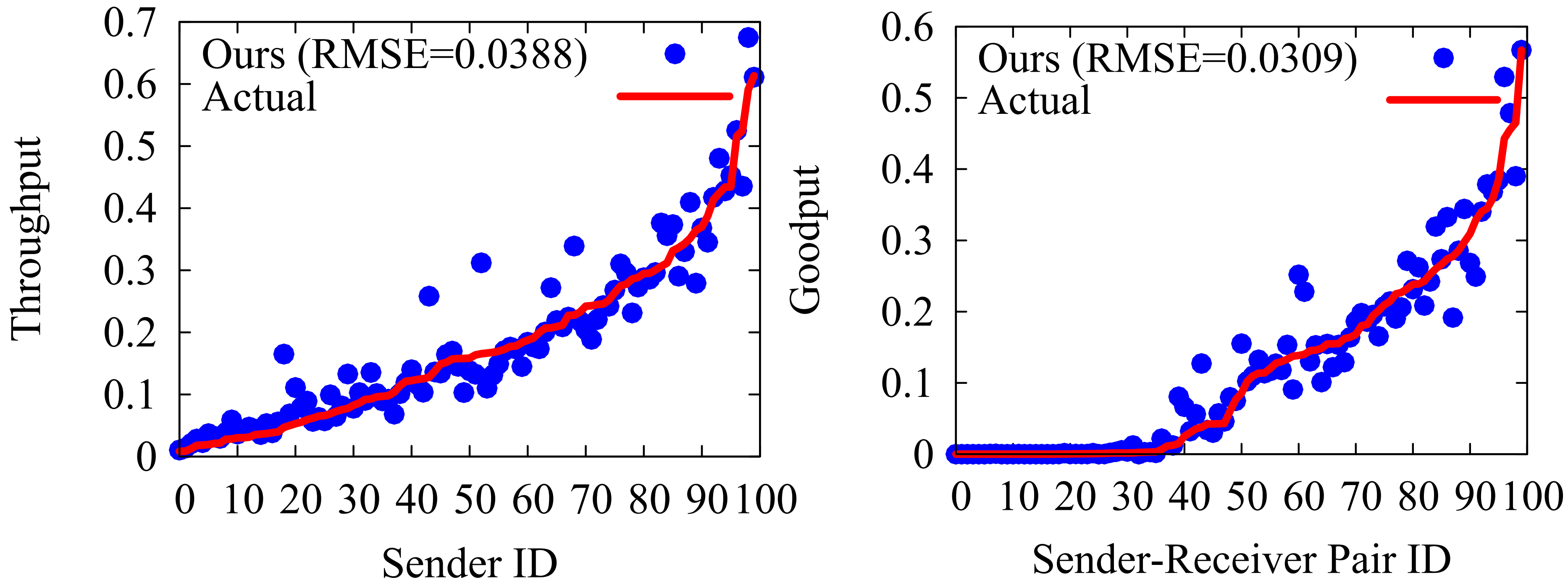
(a) sending rate

(b) throughput

Accurate for 10 saturated broadcast

Simulation Evaluation: Unsaturated Unicast

10 unsaturated unicast



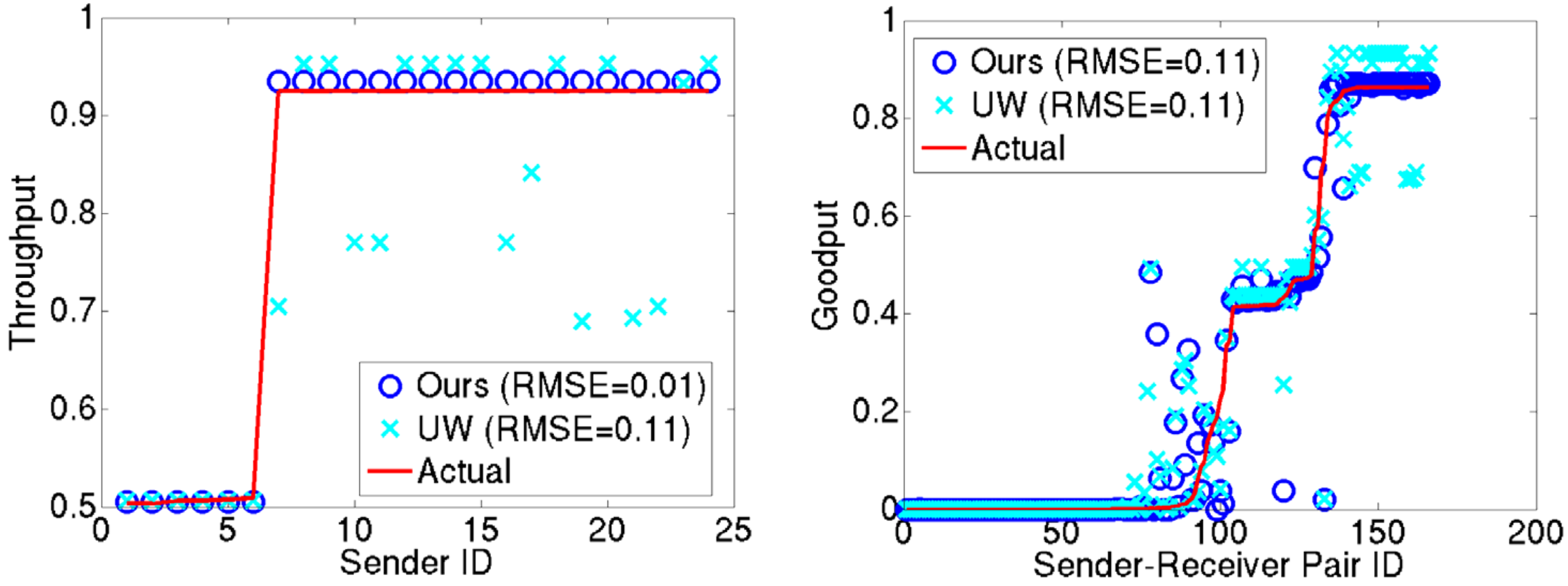
(a) sending rate

(b) throughput

Accurate for unsaturated unicast

Testbed Evaluation

UW traces: 2 saturated broadcast senders (30 mW)



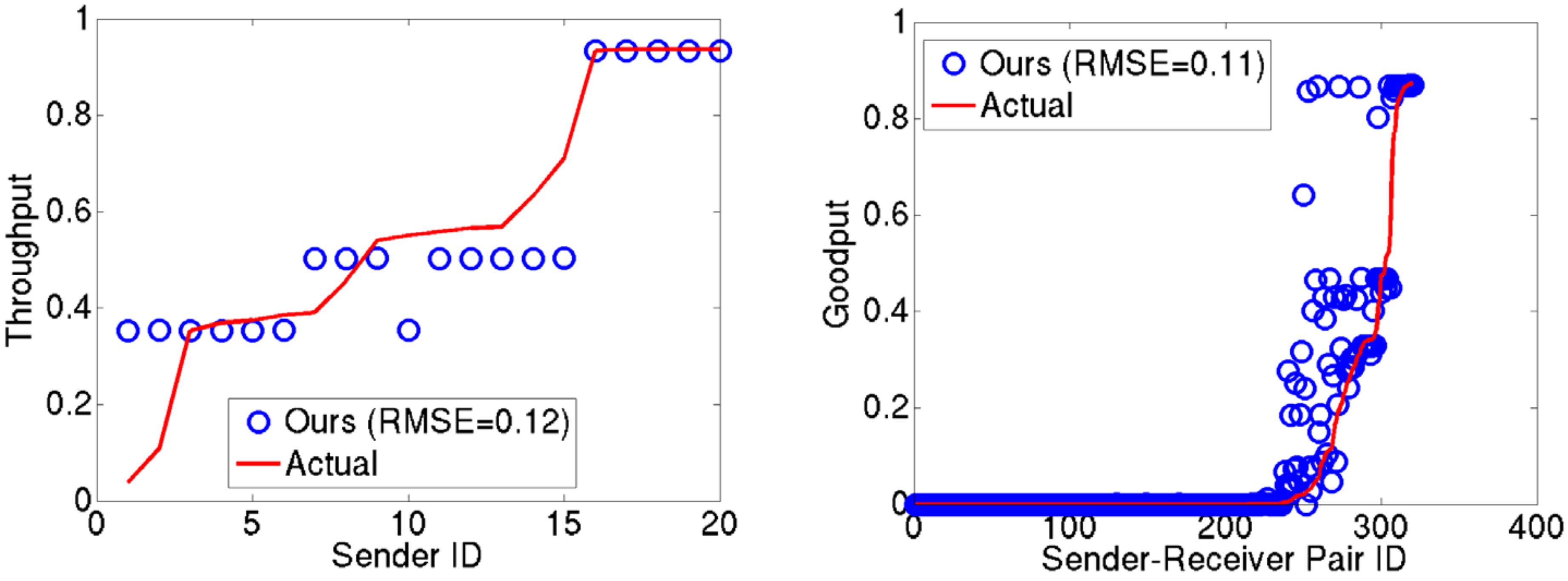
(a) sending rate

(b) throughput

More accurate than UW-model for 2-sender

Testbed Evaluation (Cont.)

UT traces: 5 saturated broadcast senders (30 mW)



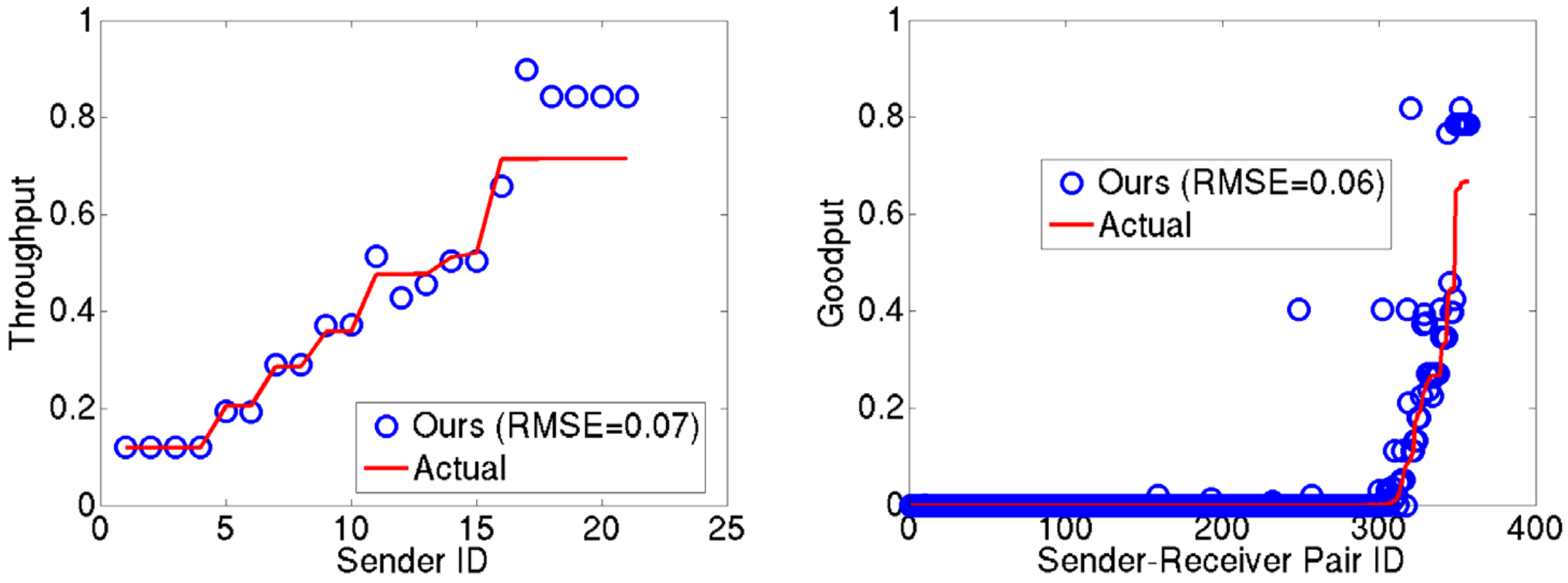
(a) sending rate

(b) throughput

Accurate for saturated broadcast

Testbed Evaluation (Cont.)

UT traces: 3 unsaturated broadcast senders (1mW)



(a) sending rate

(b) throughput

Accurate for unsaturated broadcast

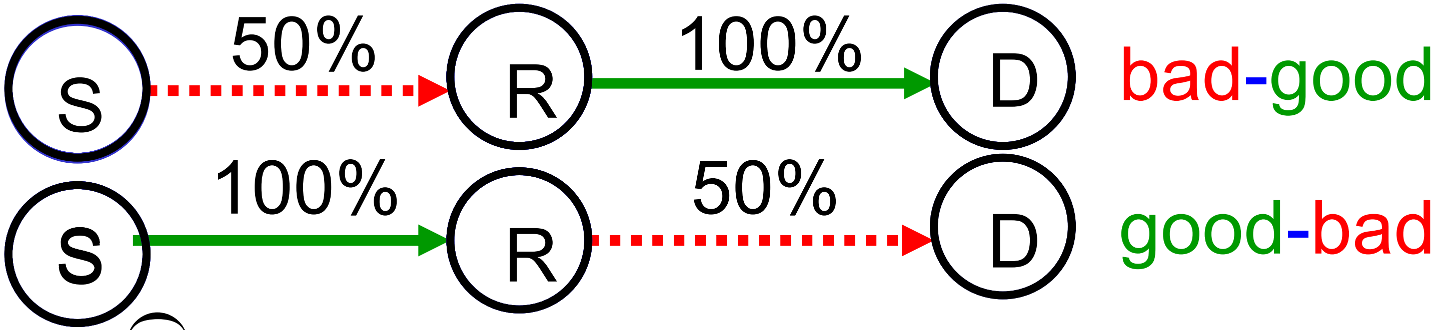
Summary

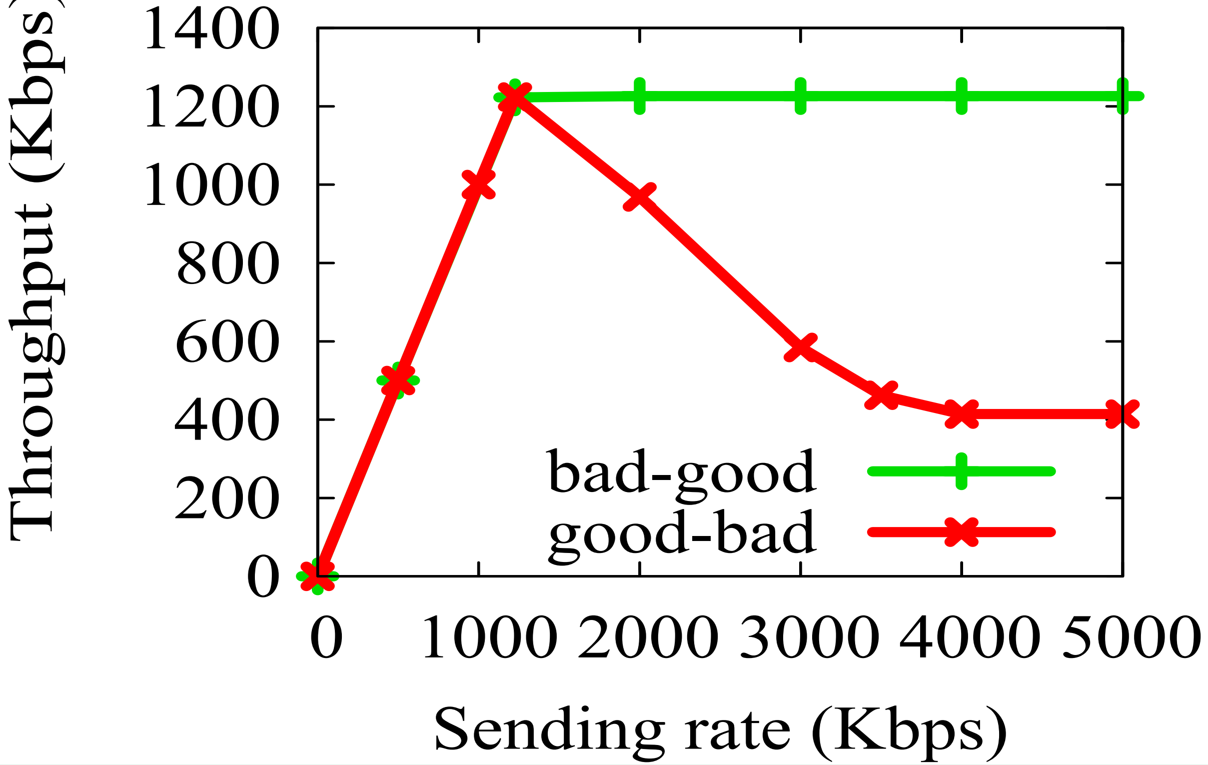
- A general interference model that handles
  - An arbitrary number of senders
  - Broadcast + unicast traffic
  - Saturated + unsaturated demands
- Validated by simulation and testbed evaluation
  - Achieve high accuracy under all types of traffic demands: broadcast/unicast + saturated/unsaturated
  - More accurate than state-of-art model for 2 saturated broadcast senders
  - Higher errors in testbed than in simulations due to
    - RF fluctuation
    - Errors in estimating actual RSS especially under high loss rates

Roadmap

- Detailed model of 802.11 networks [mobicom07]
  - Focus: accurate what-if analysis
- Simple unicast model of 802.11 networks and its application to network optimization [sigcomm08]
  - Focus: simple model + predictable optimization

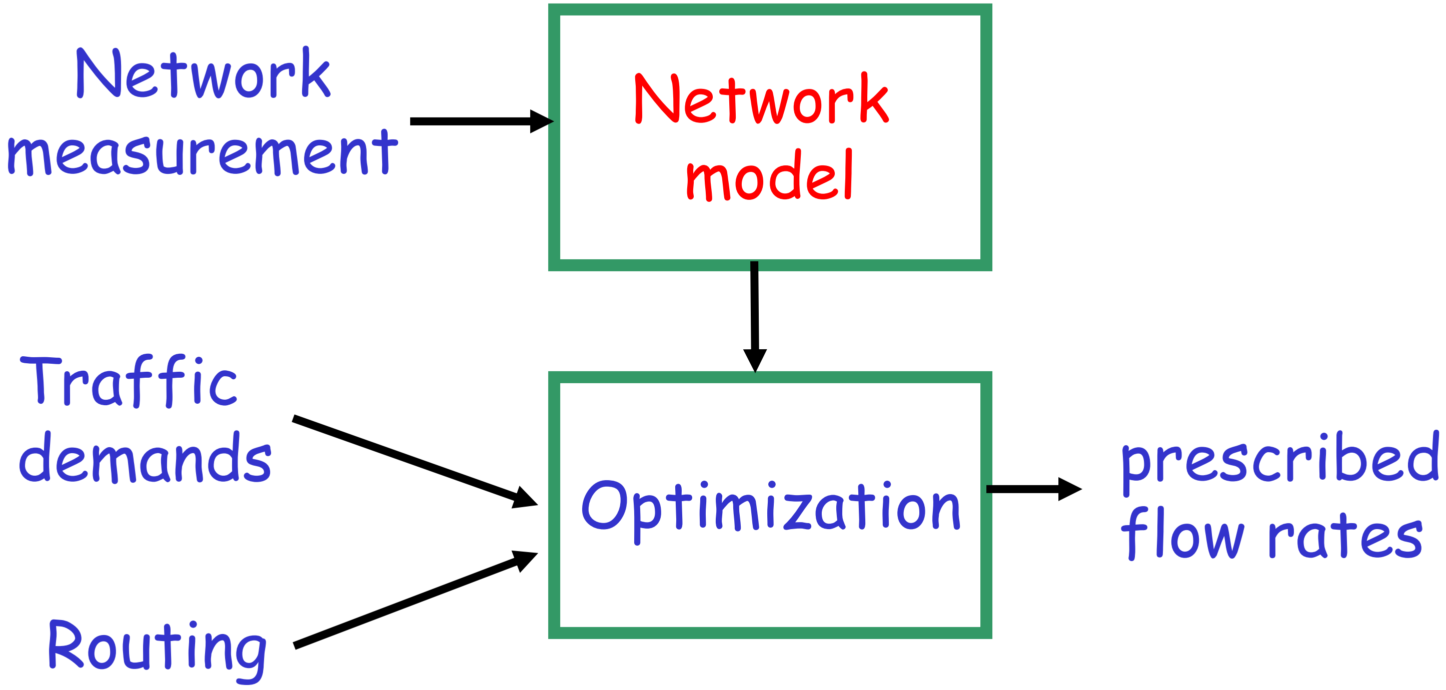
Unpredictability of wireless networks





Need predictable wireless performance optimization.

Model-driven optimization framework



Network measurement

Network model

Traffic demands

Optimization

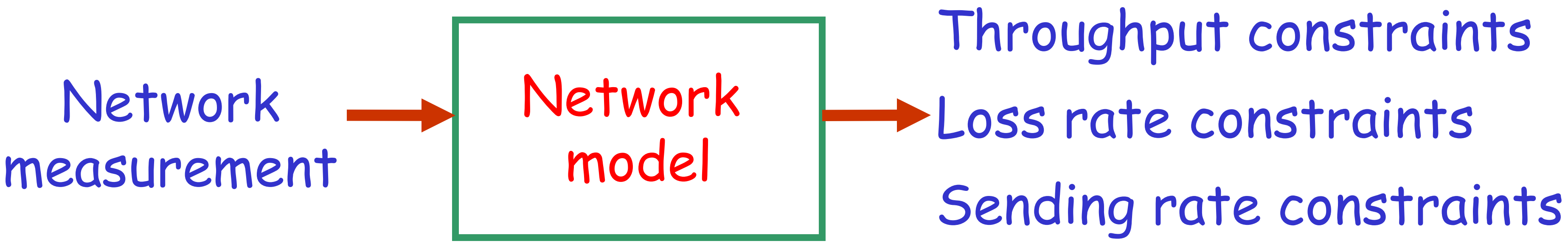
prescribed flow rates

Routing

Performance objectives:
- Maximize fairness, total throughput, …

Our network model

- Provide a compact characterization of feasible solution space to facilitate optimization
- Simple: O(N) constraints for N links
- Flexible and accurate
    - Handle asymmetric link loss rate
    - Handle asymmetric interference
    - Handle hidden terminals
    - Handle heterogeneous, multihop traffic demands



Network measurement

Network model

Throughput constraints

Loss rate constraints

Sending rate constraints

Throughput constraints

- Divide time into variable-length slot (VLS)
  - 3 types of slots:
    idle slot, transmission slot, deferral slot

Expected payload transmission time

Probability of starting tx in a slot

Success probability

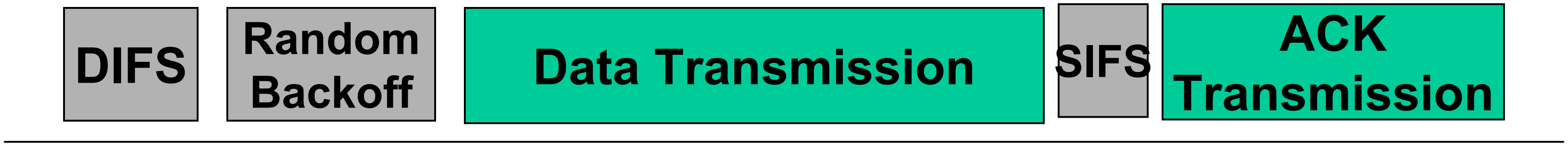
$$g_i = \frac{EP_i \times \tau_i \times (1 - p_i)}{\underbrace{(1 - \sum_j D_{ij}\tau_j) \times T_{slot} + \tau_i T_i + \sum_{j \neq i} D_{ij}\tau_j T_j}_{\text{Expected duration of a variable-length slot}}}$$

Loss rate constraints

- Inherent and collision loss are independent
- Inherent loss
  - Based on one-sender broadcast measurement
- Collision loss
  - Synchronous loss
    - Two senders can carrier sense each other
    - Occur when two transmissions start at the same time
  - Asynchronous loss
    - At least one sender cannot carrier sense the other
    - Occur when two transmissions overlap

Sending rate feasibility constraints

- 802.11 unicast



  - Random backoff interval uniformly chosen [0,CW]
  - CW doubles after a failed transmission until CW_{max}, and restores to CW_{min} after a successful transmission or when max retry count is reached
  - $CW(p_i)$: the expected contention window size under packet loss rate p_i [Bianchi00]
- Sending rate feasibility constraints

$$0 \leq \tau_i \leq \frac{1}{1 + CW(p_i)/2}$$

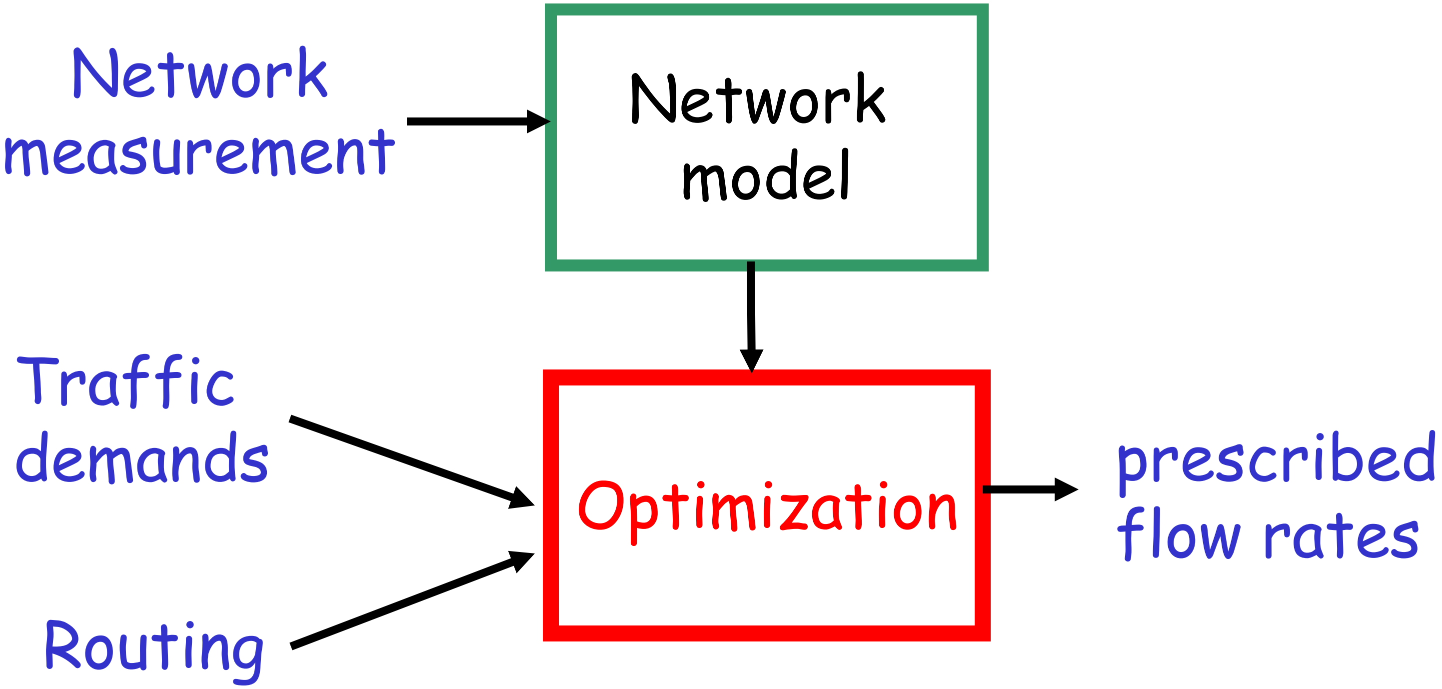
Extensions to the basic model

- RTS/CTS
  - Add RTS and CTS delay to VLS duration
  - Add RTS and CTS related loss to loss rate constraints
- Multihop traffic demands
  - Link load = routing matrix × e2e demand
  - Routing matrix gives the fraction of each e2e demand that traverses each link
- TCP traffic
  - Update the routing matrix:

$$R_{TCP} = R_{data} + \alpha \times R_{ack}$$

where α reflects the size & frequency of TCP ACKs

Model-driven optimization framework



Network measurement

Network model

Traffic demands

Routing

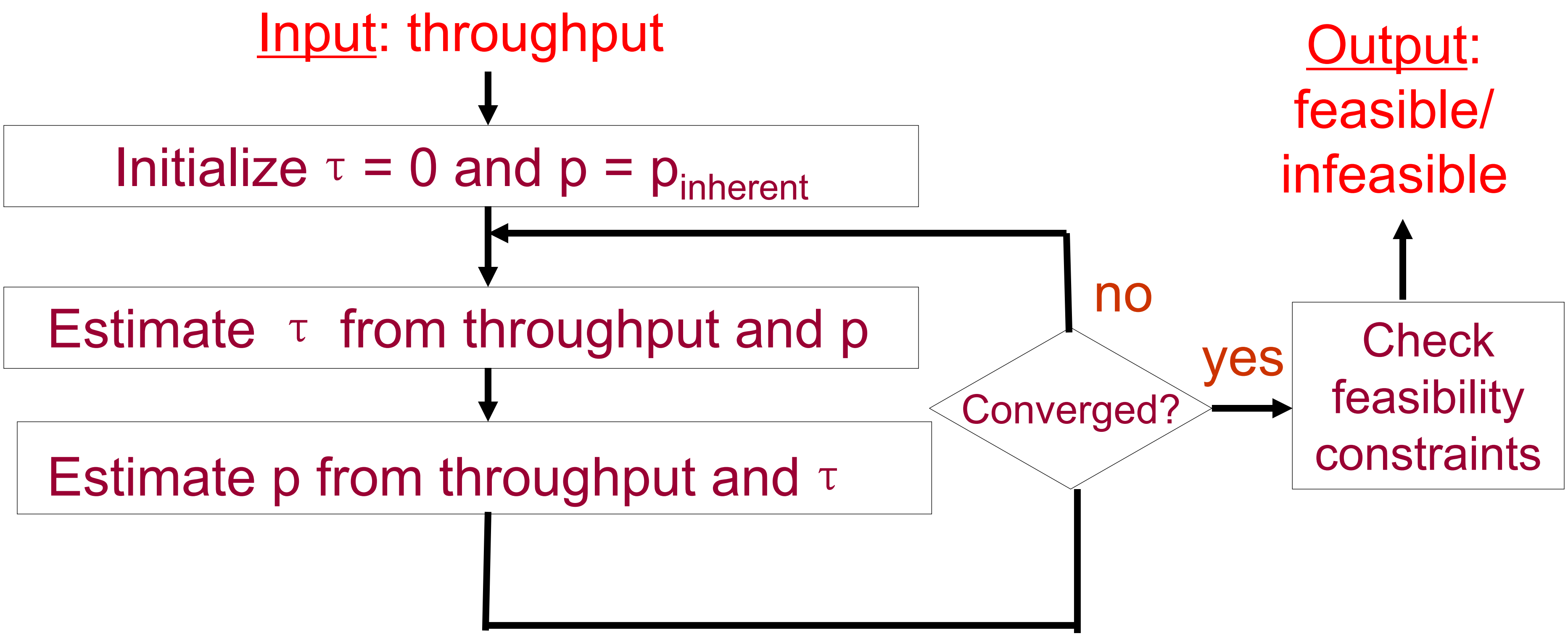
Optimization

prescribed flow rates

Performance objectives:
- Maximize fairness, total throughput, …

Flow throughput feasibility testing

- Test if given flow throughput are achievable
- Challenge: strong interdependency
- Our approach: iterative procedure



Input: throughput

Initialize $\tau = 0$ and $p = p_{inherent}$

Estimate τ from throughput and p

Estimate p from throughput and τ

Converged?

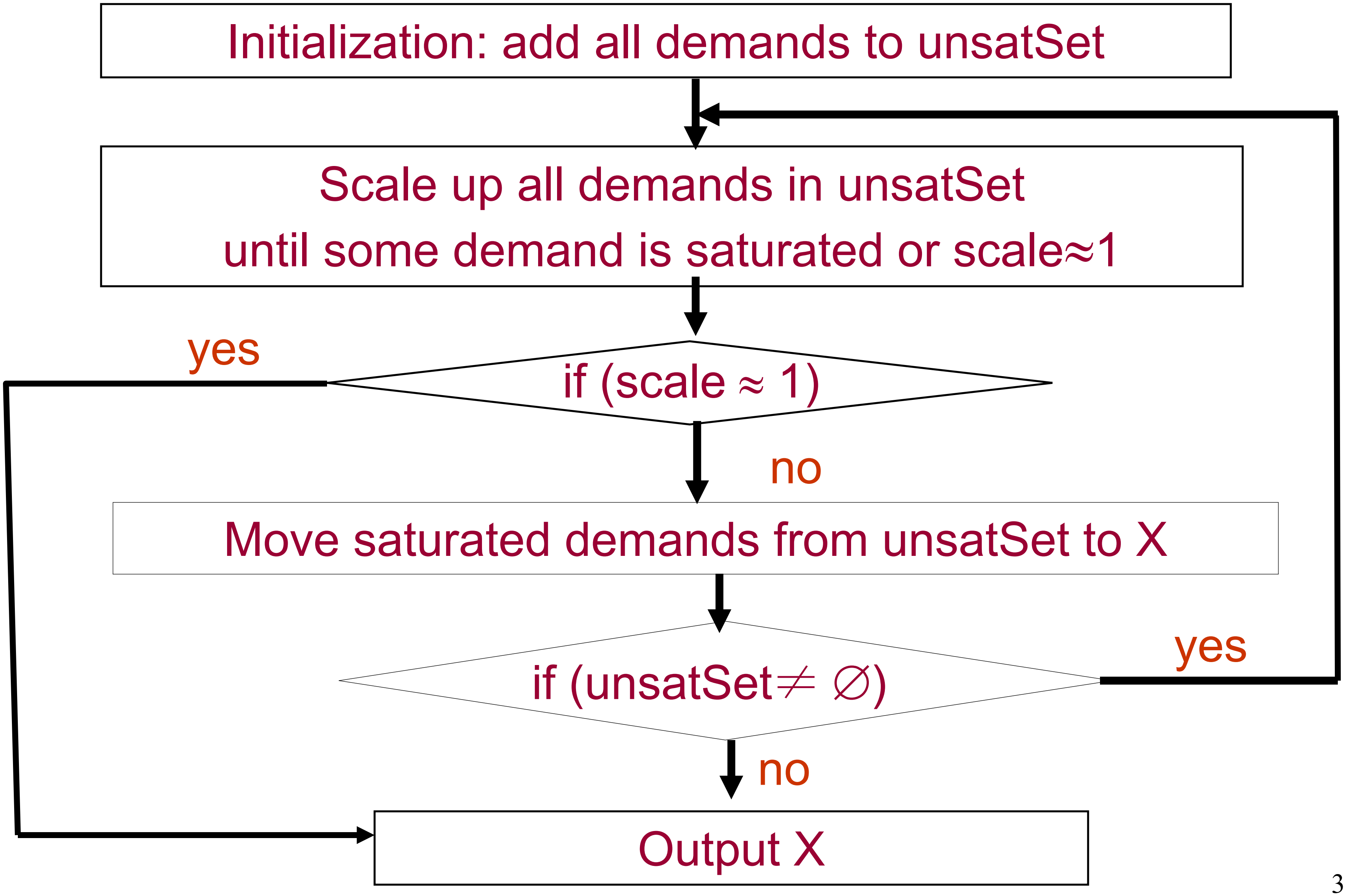
no

yes

Check feasibility constraints

Output: feasible/ infeasible

Fair rate allocation



Initialization: add all demands to unsatSet

Scale up all demands in unsatSet
until some demand is saturated or scale≈1

if (scale $\approx$ 1)

yes

no

Move saturated demands from unsatSet to X

if (unsatSet$\neq \emptyset$)

yes

no

Output X

Total throughput maximization

- Formulate a non-linear optimization problem (NLP)

$$\max \sum_d x_d$$

Maximize total throughput

$$s.t. \quad \sum_d R_{id} x_d \le \frac{EP_i \times \tau_i \times (1 - p_i)}{(1 - \sum_j \tau_j) T_{slot} + \sum_j D_{ij} \tau_j T_j}$$

Link load is bounded by throughput constraints

$$0 \le \tau_i \le \frac{1}{1 + CW(p_i)/2}$$

Sending rate is feasible

$$0 \le x_d \le x^*_d$$

E2e throughput is bounded by demand

- Solve NLP using iterative linear programming

Evaluation methodology

- Model validation
  - How to quantify over-prediction error?
    - Verify if prescribed rates are achievable
  - How to quantify under-prediction error?
    - Scale up all prescribed rates by a common factor

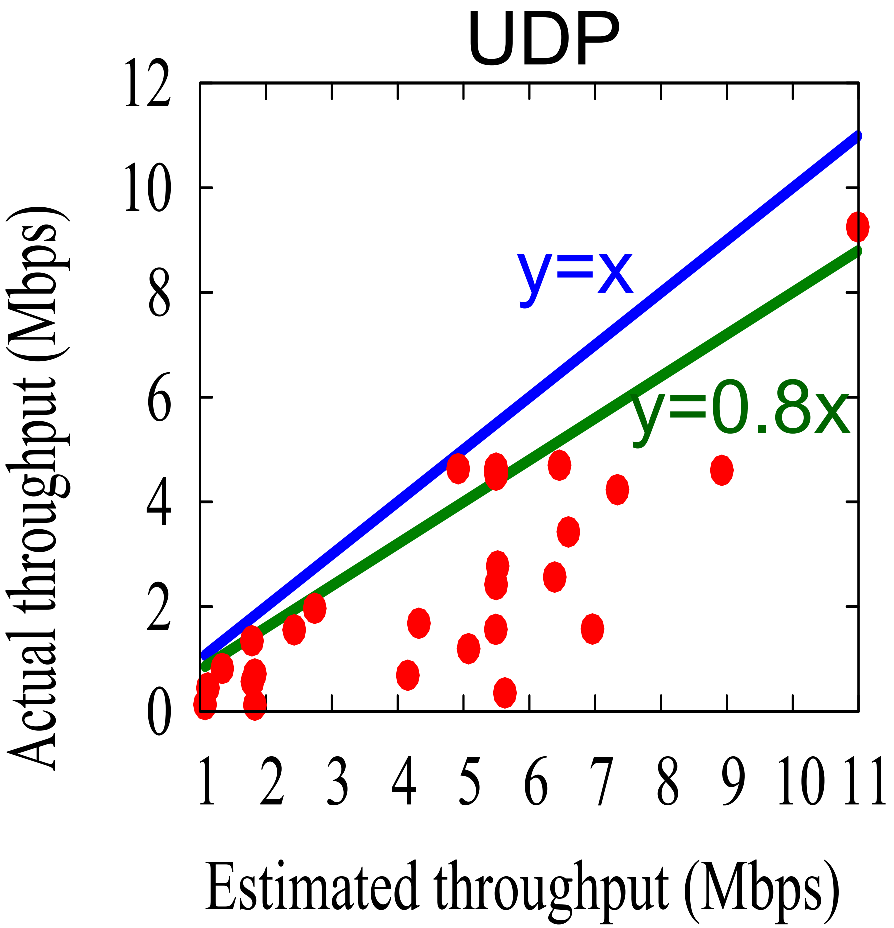
- Performance optimization
  - Fairness maximization: Jain's fairness index
    - $(\text{sum}_i\ x_i)^2 / (n * (\text{sum}_i\ x_i^2))$
  - Total throughput maximization

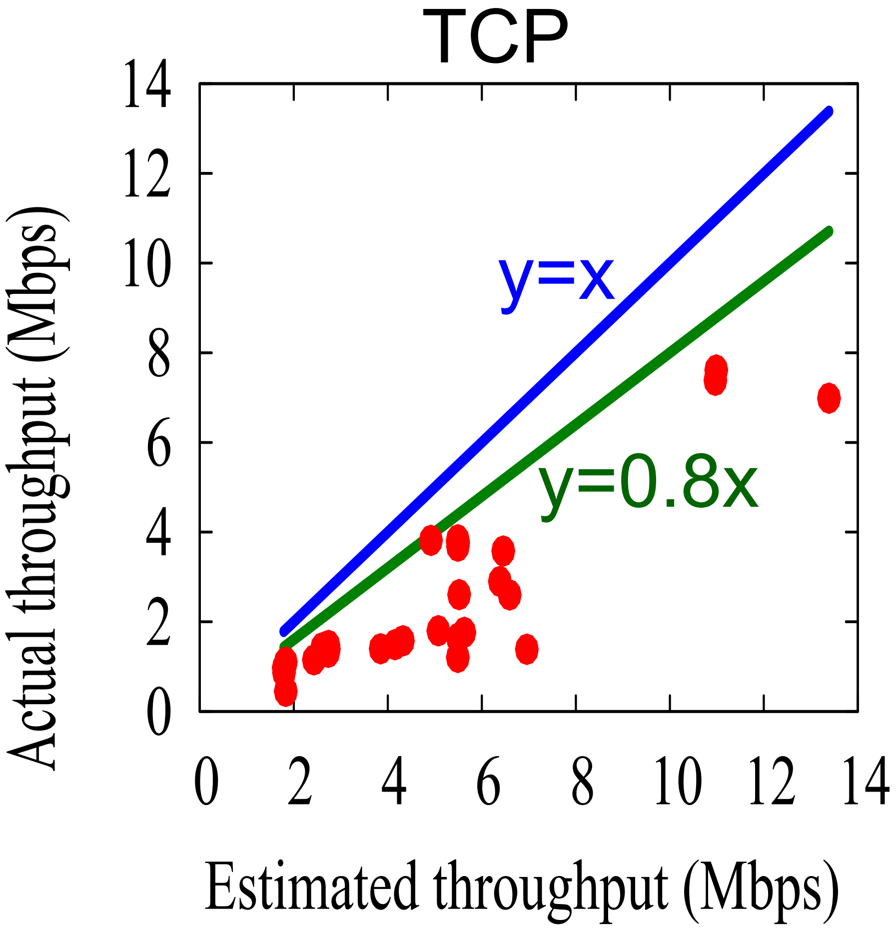
- This talk: testbed results only
  - 19 mesh nodes at UTCS building; up to 7 hops
  - Extensive simulation results are in the paper

Optimization schemes

- Our rate optimization
- No rate optimization (current practice)
- Conflict graph based optimization
  - Plug conflict graph model to our framework
  - Conflict graph assumes perfect scheduling [JPPQ03]
    - Represent each wireless link with a vertex
    - Draw an edge between the vertices if the corresponding links interfere
    - Derive clique constraints – all links in a clique in the CG cannot be active together

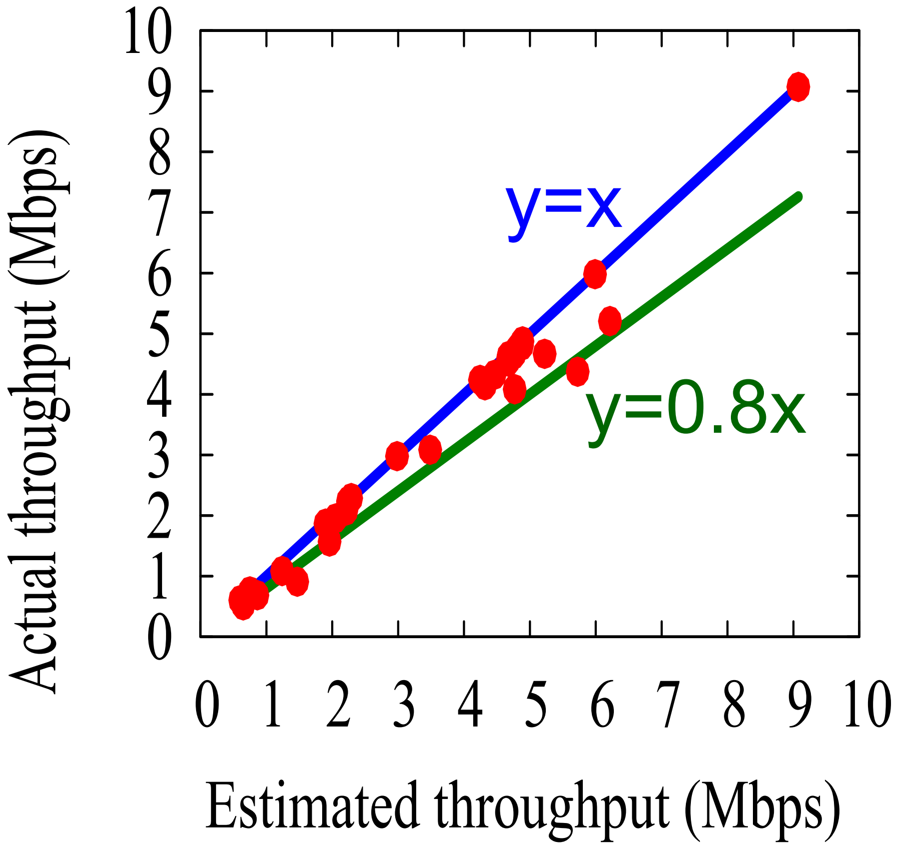
Baseline: conflict graph model

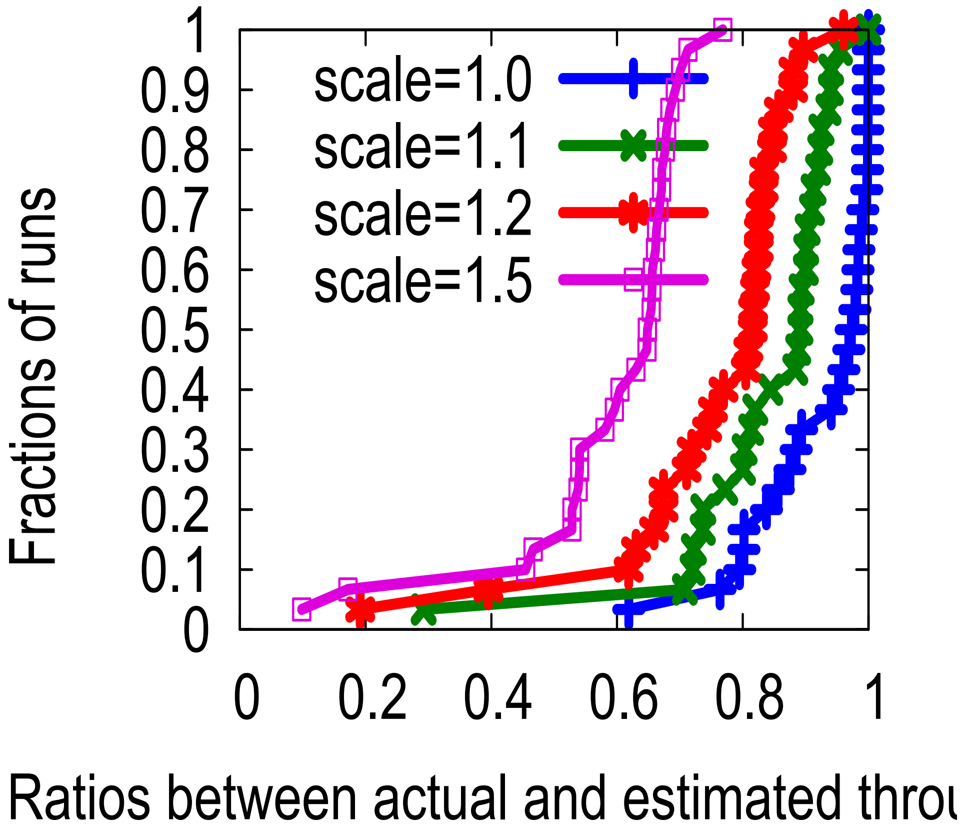




CG model significantly over-estimates sending rates.

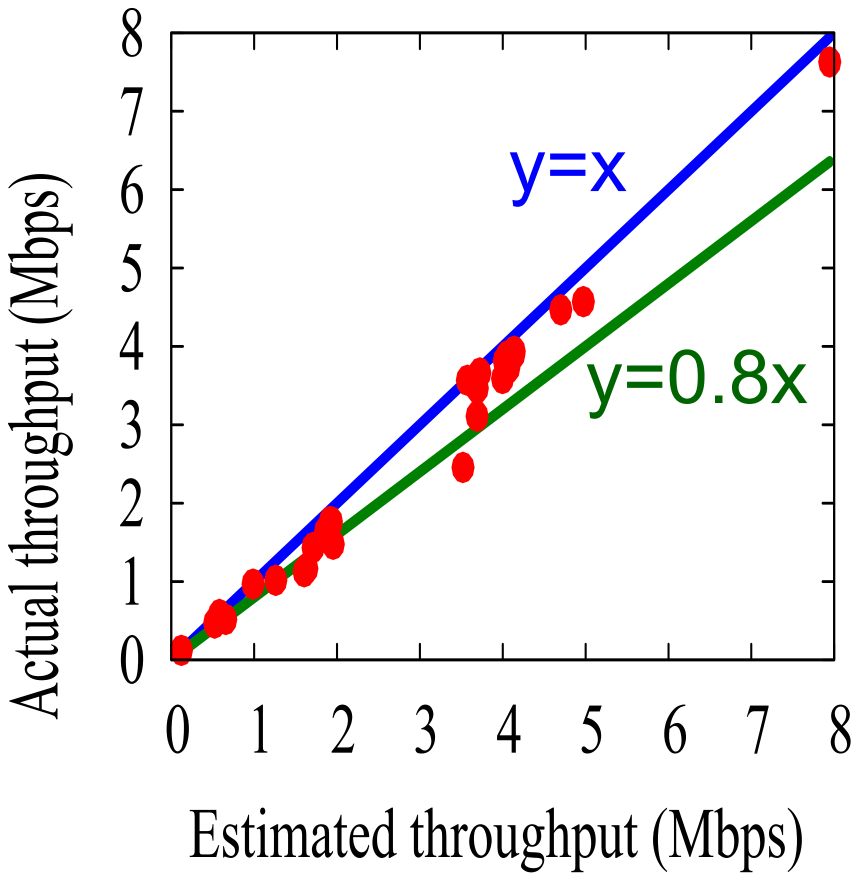
Model validation: UDP traffic

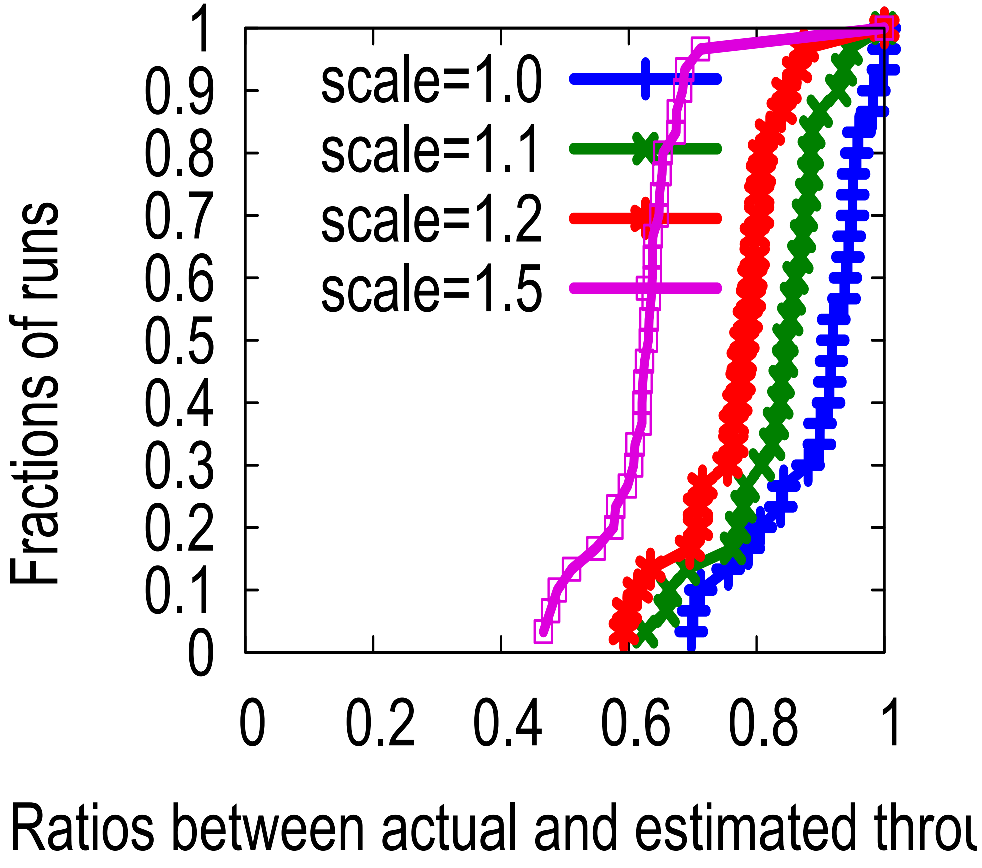




1) Most estimated rates are achievable within 20%.
2) Rates scaled up by just 10% become unachievable.

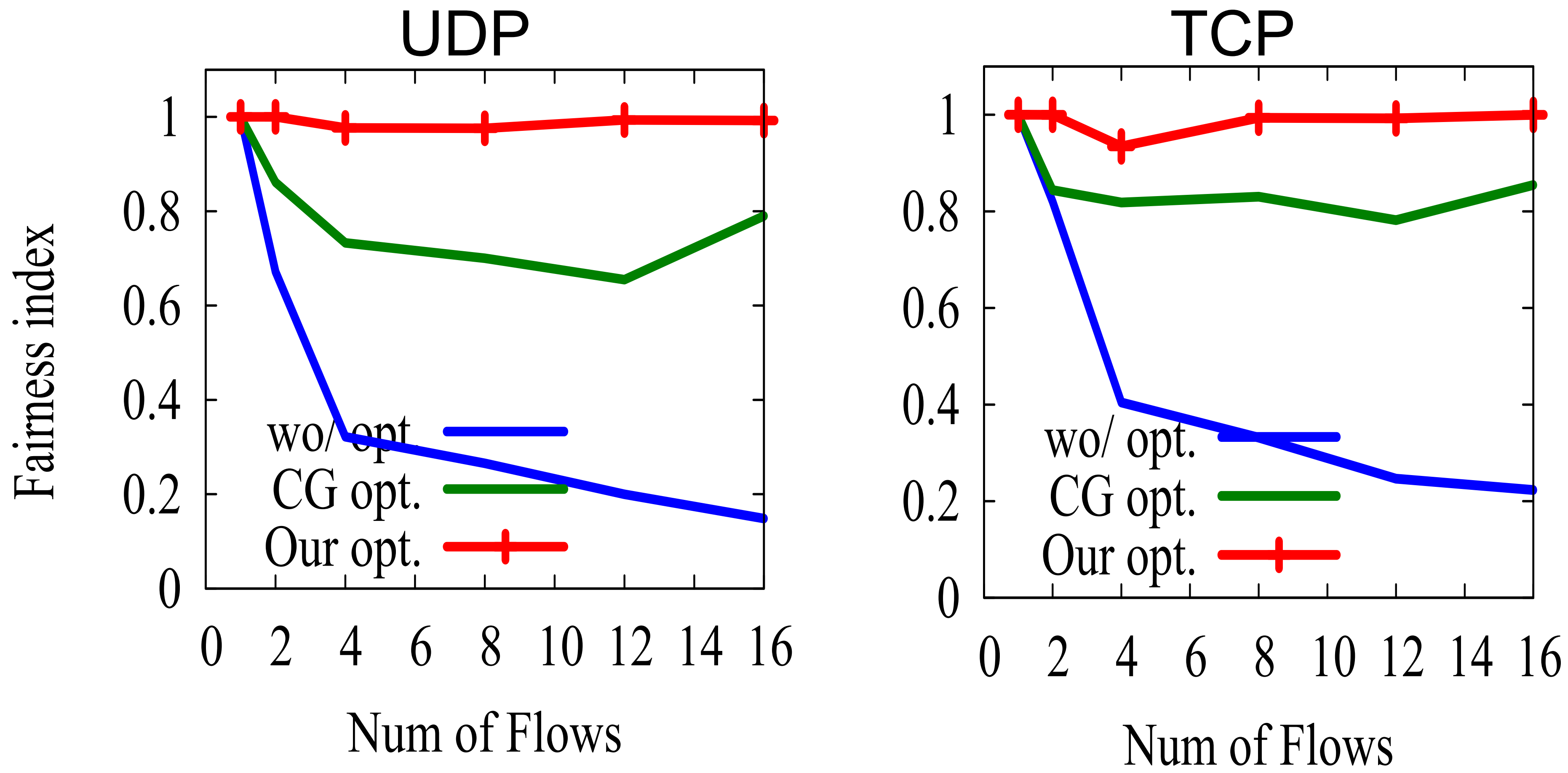
Model validation: TCP traffic





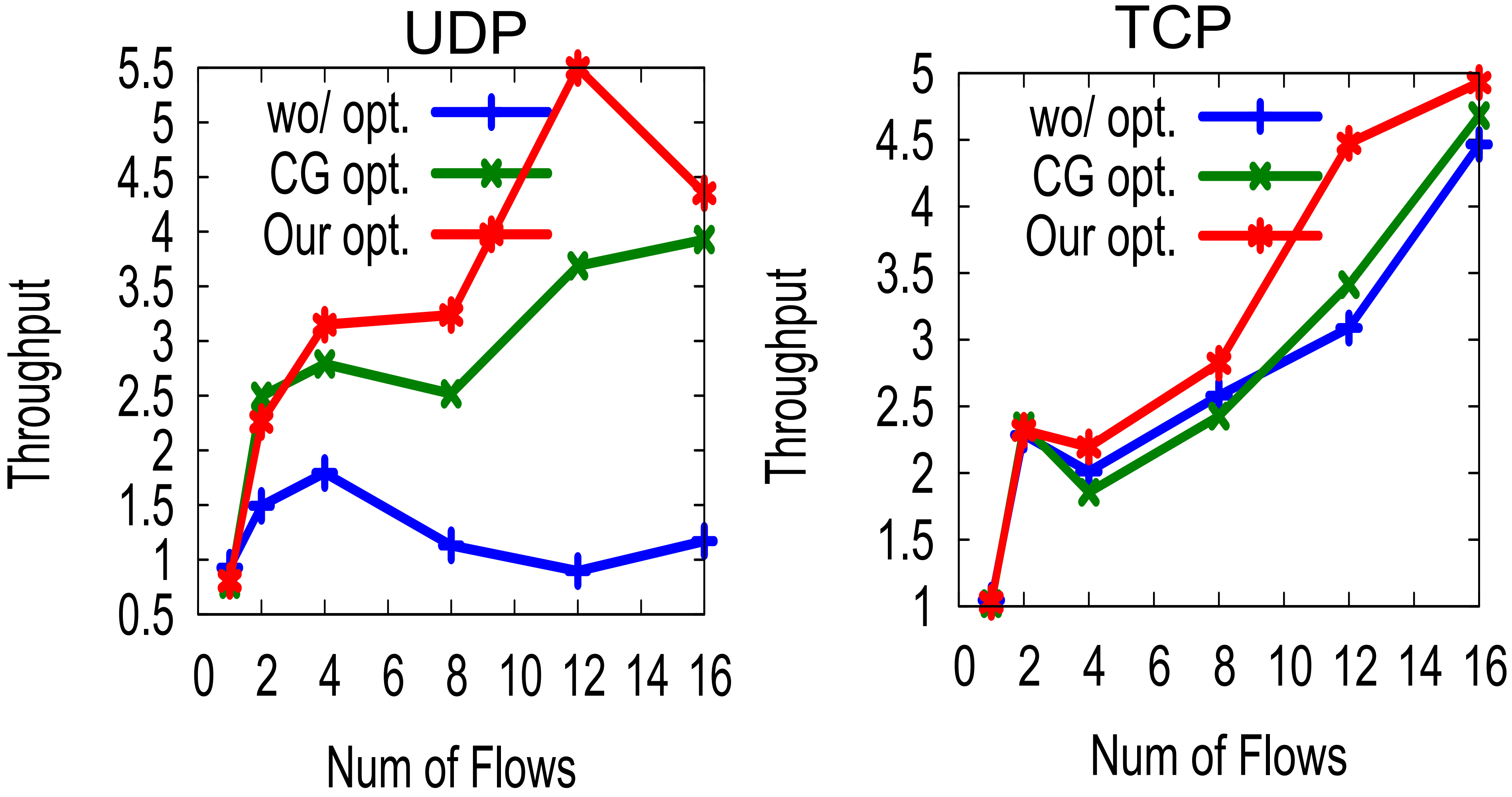
Our model is accurate for TCP traffic.

Maximizing fairness



Fairness index is close to 1 under our scheme, while it degrades quickly in other schemes.

Maximizing total throughput



Our scheme significantly increases total throughput.

Summary

- Predictable wireless performance optimization
  - A simple yet accurate wireless network model
  - Effective model-driven optimization algorithms

- Demonstrate their effectiveness through testbed experiments and simulation

Conclusion

- Interference-aware network management
  - Measuring wireless interference
  - Modeling wireless interference
  - Interference-aware network control to directly optimize e2e performance
    - Interference-aware rate limiting
- Relationship to MRA?
  - Q: How to live with dynamic graphs?
    - Need to consider it in specific applications
    - For wireless optimization, what really matters is a stable subgraph? How does this stable subgraph evolve?
  - Q: How can MRA help to improve scalability?
    - E.g., can we decompose large wireless networks into "soft" clusters and do intra- and inter-cluster optimization?

Thank you!

http://www.cs.utexas.edu/~yzhang/papers/