Improving Wireless Mesh Network Throughput with Superposition Coding and Packet Mixing

Richard Alimi
3rd Year Ph.D. Student

October 23, 2007

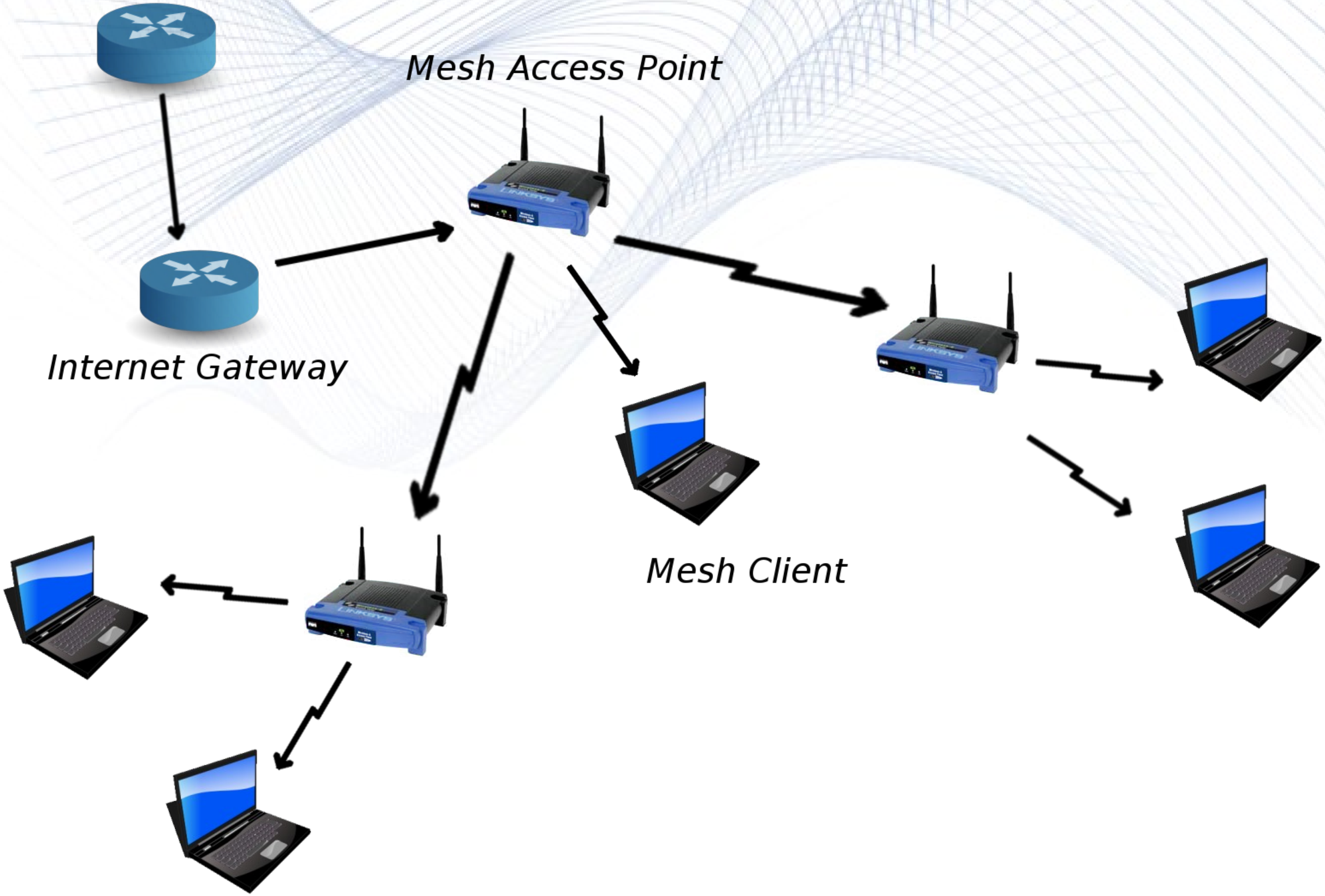
OGST
Yale University

Joint work with: Li (Erran) Li, Ramachandran Ramjee, Jingpu Shi, Yanjun Sun, Harish Viswanathan, Richard Yang

Wireless Mesh Networks

- City- and community-wide mesh networks becoming popular
  - Seen as a new approach to the “last mile” of Internet service
  - 186 deployed and 164 in the planning stages in the United States alone as of March 2006 [muniwireless.com]
- Mesh network structure
  - Access Points (APs) deployed, some connect directly to Internet
    - Street lamps
    - Traffic lights
    - Public buildings
  - Clients associate with nearest AP
  - Traffic routed to and from Internet via APs
    - Possibly along multi-hop path

Mesh Network Structure



Limited Capacity of Mesh Networks

- Current mesh networks have limited capacity [Li *et al.* 2001, dailywireless.org 2004]
- Increased popularity will only worsen congestion
  - Larger downloads
  - P2P applications
  - Video streaming
- Network-wide transport capacity does not scale [Gupta and Kumar 2001]
  - $O(\sqrt{n})$ where n is the number of users
- Must bypass traditional constraints
  - Use benefits of the wireless medium

The Wireless *Broadcast* Channel

- Transmissions heard by multiple receivers
- Fundamental observation to increase capacity
  - Bypass single-receiver constraint in Gupta and Kumar's upper bound on capacity
  - Mix multiple packets together in the same transmission such that each receiver can recover its own packet

Use messages not addressed to you to decode

messages that are *addressed to you*

Packet Mixing Techniques

- Transmitter-side mixing
  - Downlink superposition coding
  - XOR-style network coding
- Receiver-side mixing
  - Uplink superposition coding
  - Analog and physical-layer network coding
  - Relay channel
- *Ultimate question: can we develop a framework for constructing mixed packets?*

Talk Outline

- ***Related work***
- Introduction to basic concepts
  - Downlink superposition coding
  - XOR-style network coding
- Construction of mixed packets
  - Analysis and motivation
  - Algorithms
  - Evaluations
- GNU Radio testbed
- Future work

Related Work: Information Theory

- Superposition coding introduced [Cover 1972, Bergmans and Cover 1974]
- Superposition coding achieves optimal capacity for AWGN channel [Bergmans 1974]
- Superposition coding had not yet been investigated for mesh networks
- Challenges
  - Must maintain good channel feedback
  - Need appropriate scheduling algorithms

Related Work: Coding in Wireless

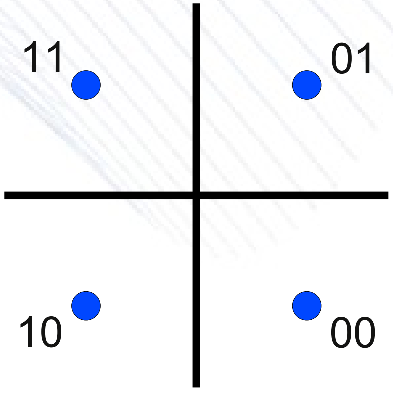
- Generalized network coding proposed by to achieve multicast capacity of general networks [Ahlswede *et al.* 2000]
  - Computationally expensive (requires large matrix multiplications)
- COPE: XOR-style network coding [Katti *et al.* 2006]
  - Less suitable for traffic patterns common to mesh networks
- Analog network coding [Katti *et al.* 2007] and Physical-layer network coding [Zhang *et al.* 2006]
  - Increased receiver complexity
  - Less suitable for traffic patterns common to mesh networks

Talk Outline

- Related work
- ***Introduction to basic concepts***
  - ***Downlink superposition coding***
  - ***XOR-style network coding***
- Construction of mixed packets
  - Analysis and motivation
  - Algorithms
  - Evaluations
- GNU Radio testbed
- Future work

Signal Modulation

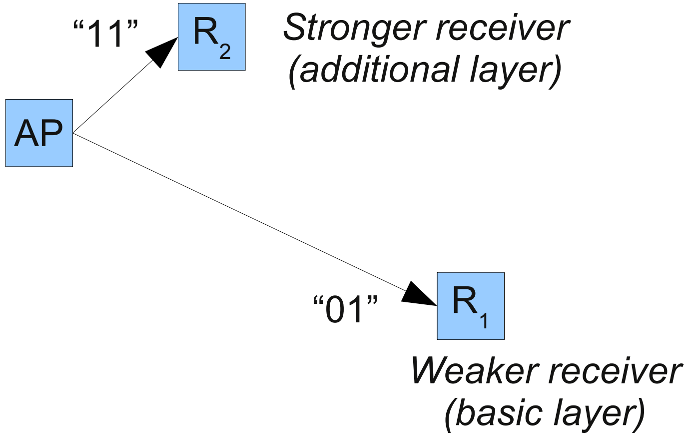
- Signal has two components: I and Q
- Typically represented on complex plane
- Sender
  - Map bits to symbol (*i.e.*, a complex number)
  - Transmit signal corresponding to that symbol
- Receiver
  - Observe noisy version of the signal
  - Determine most likely symbol and emit corresponding bits
- # of constellation points partially determines bitrate



QPSK

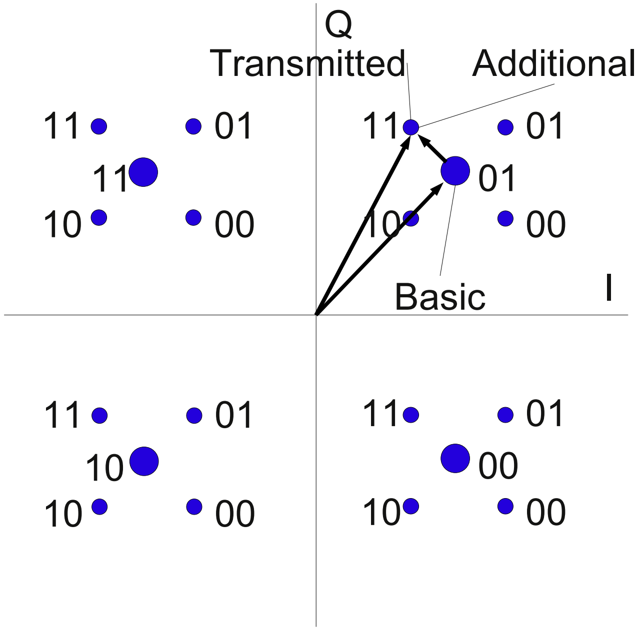
Downlink Superposition Coding (SC)

- Basic idea
  - Multiple receivers, message queued for each
  - Transmit independent message to each simultaneously
  - Each message transmitted in a different *layer*
- Example



Stronger receiver (additional layer)

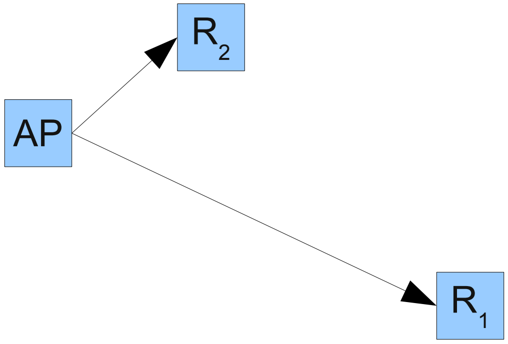
Weaker receiver (basic layer)



2-layer QPSK

Decoding Technique: SIC

- Receivers use *successive interference cancellation*
- Weaker receiver (R_1)
  - Decodes normally (with slightly-higher interference)
- Stronger receiver (R_2)
  - First decodes signal to weaker receiver
  - Reconstructs signal and subtracts from original
  - Only stronger receiver's signal remains
- Generalizes to >2 receivers

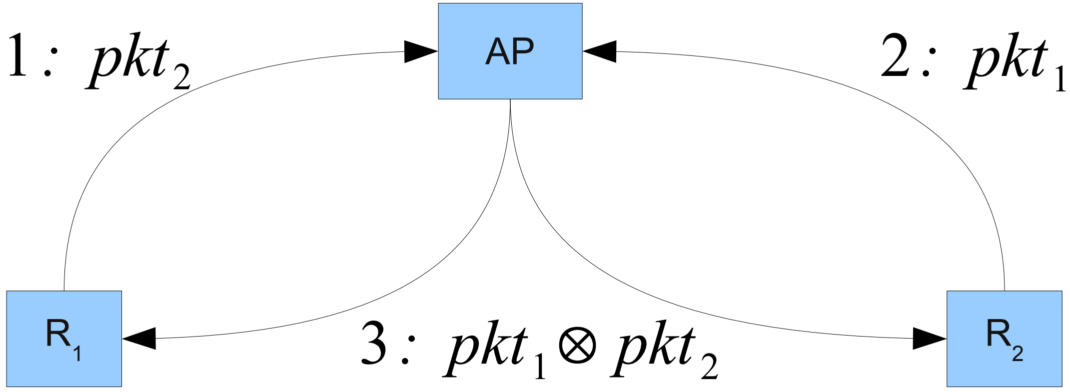


Superposition Coding in Mesh Networks

- Favorable traffic patterns
  - Flows from Internet to clients (*e.g.*, HTTP/FTP downloads)
  - Flows between clients of same AP
- Exploits client diversity
  - Varying distances to receivers
  - Building materials of surroundings

XOR-style Network Coding

- Basic idea
  - Nodes remember overheard and sent messages
  - Transmit bitwise XOR of packets: $pkt_1 \otimes pkt_2 \otimes \cdots \otimes pkt_n$
  - Receivers decode if they already know *n-1* packets
- Example



$1: \; pkt_2$

AP

$2: \; pkt_1$

R_1

$3: \; pkt_1 \otimes pkt_2$

R_2

Network Coding in Mesh Networks

- Favorable traffic patterns
  - Flows between clients of same AP
- Exploits broadcast channel and traffic patterns
  - Clients within overhearing distance; *or*
  - Prior transmissions from clients

Talk Outline

- Related work
- Introduction to basic concepts
  - Downlink superposition coding
  - XOR-style network coding
- ***Construction of mixed packets***
  - ***Analysis and motivation***
  - ***Algorithms***
  - ***Evaluations***
- GNU Radio testbed
- Future work

Packet Mixing in Mesh Networks

- Objective
  - Develop algorithms to take advantage of mixed packets
  - Given current state, construct a mixed packet with:
    - Maximum effective throughput
    - Sufficiently-high probability that receivers decode their messages
    - Mixture of coding techniques
- Currently consider two techniques
  - Downlink superposition coding (will be denoted by *SC*)
  - XOR-style network coding (will be denoted by *NC*)

Analysis and Motivation

- Does 802.11a/b/g offer any benefits to SC?
  - We derive formula for rate quantization gains
- SC theoretical scalability analysis
  - We show network-wide transport capacity is $O(n)$ instead of $O(\sqrt{n})$
- We prove optimal scheduling with packet mixing is NP Hard
  - Thus, we extend existing greedy algorithm for constructing NC packets

SC Quantization Gains

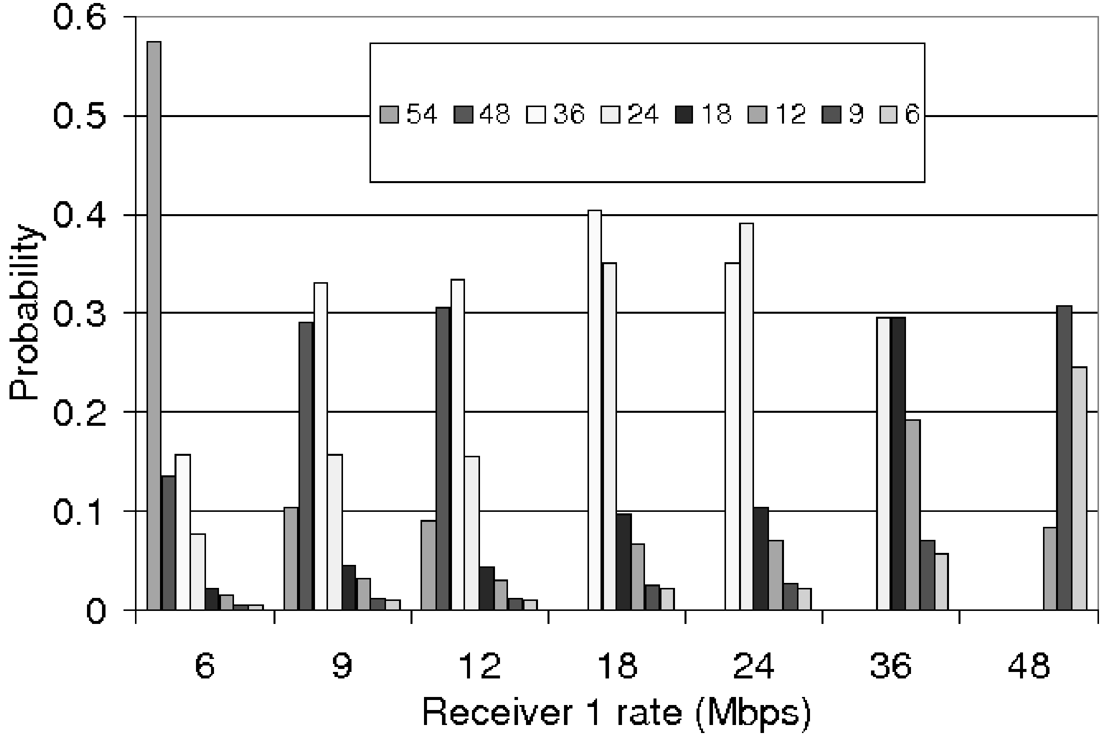
- Discrete rates are common in standards (including 802.11)
- In other words...
  - Channel qualities are quantized
- Idea
  - Steal extra power from one receiver without affecting data rate
  - Allocate extra power to second layer destined for a stronger receiver
    - Use largest rate achievable with this extra power
- We have derived formula for computing these gains:

$$\Pr\{R_2 \geq r_2 | R_1 = r_1^*\} = \frac{\left(\frac{Z(r_1^{*+})}{Z(r_1^*)}\right)^{\frac{2}{k}} \left(1 - \frac{N_0 Z(r_2)\left(1+Z(r_1^*)\right)}{P h_2}\right)^{\frac{2}{k}} - 1}{\left(\frac{Z(r_1^{*+})}{Z(r_1^*)}\right)^{\frac{2}{k}} - 1}$$

where:

$$Z(r) = \frac{1}{I_m}\left(2^{\frac{r}{W}} - 1\right)$$

SC Quantization Gains in 802.11a/g



- Distances
  - R_1: varies
  - R_2: 40 m
- Path-loss Exponent: 4
- Noise: -90 dBm
- Remaining parameters consistent with Cisco Aironet 802.11g card

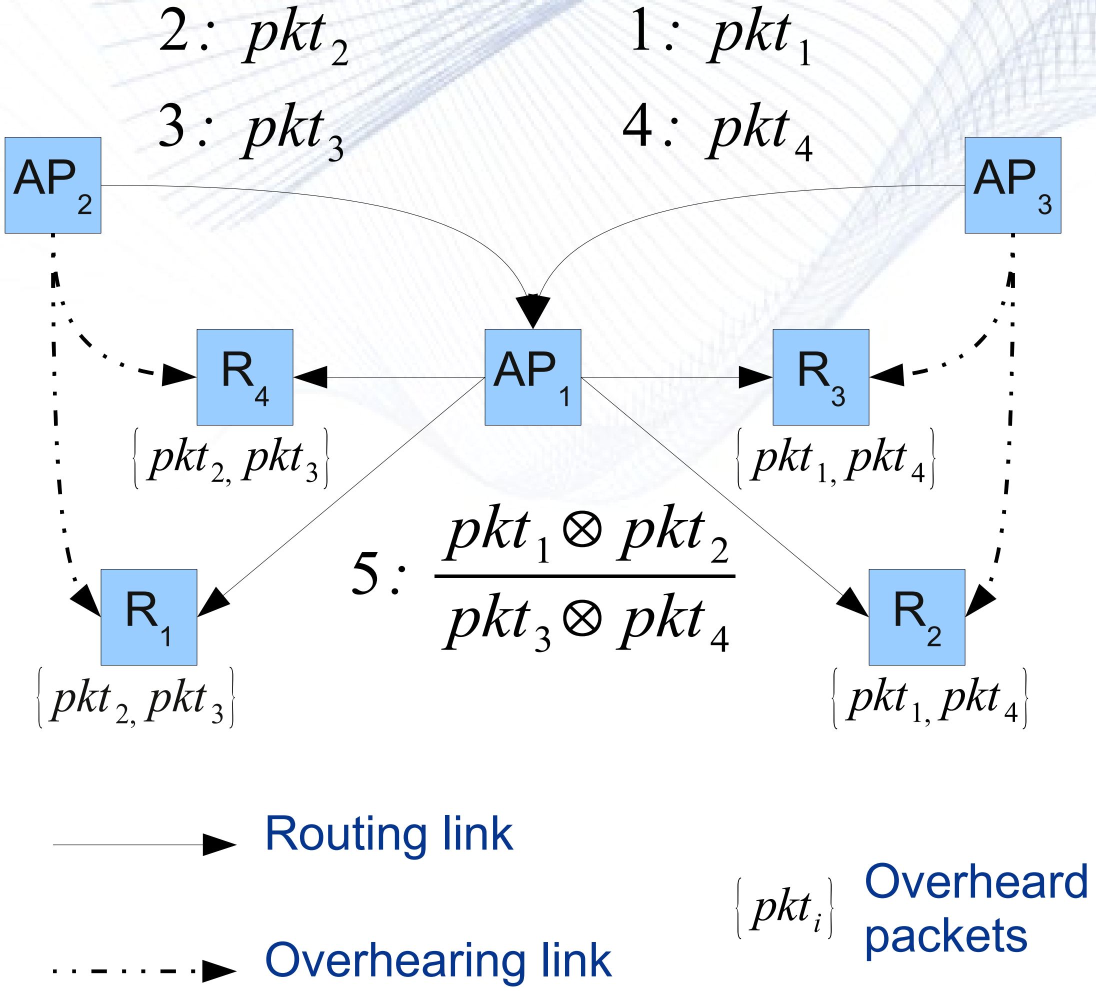
SC Scalability Analysis

- Gupta and Kumar result assumes two constraints
  - At most *n / 2* concurrent transmissions in the network
  - Interference constraint
- Story changes with superposition coding
  - At most *n-1* concurrent transmissions (superposition coding with *n-1* layers)
  - # of interfering transmissions reduced by a factor of *n*
- Modified constraints produce *O(n)* upper bound
- Achievable when all nodes within transmission distance of each other

Hardness of Optimal Packet Mixing

- Problem formulation
  - Nodes maintain cache of transmitted and overheard packets
  - Assume perfect knowledge of neighbor's packet caches
  - Determine maximum number of packets that can be mixed in a single transmission
- Proof:
  - Reduction from maximum clique problem

Motivating Example: Why Packet Mixing?



- Flows
  - $AP_2 \rightarrow R_2$, $AP_2 \rightarrow R_3$
  - $AP_3 \rightarrow R_1$, $AP_3 \rightarrow R_4$
  - pkt_d has destination R_d
- Without packet mixing
  - 8 transmissions required
- With packet mixing
  - 5 transmissions required

Applying Packet Mixing in the Mesh

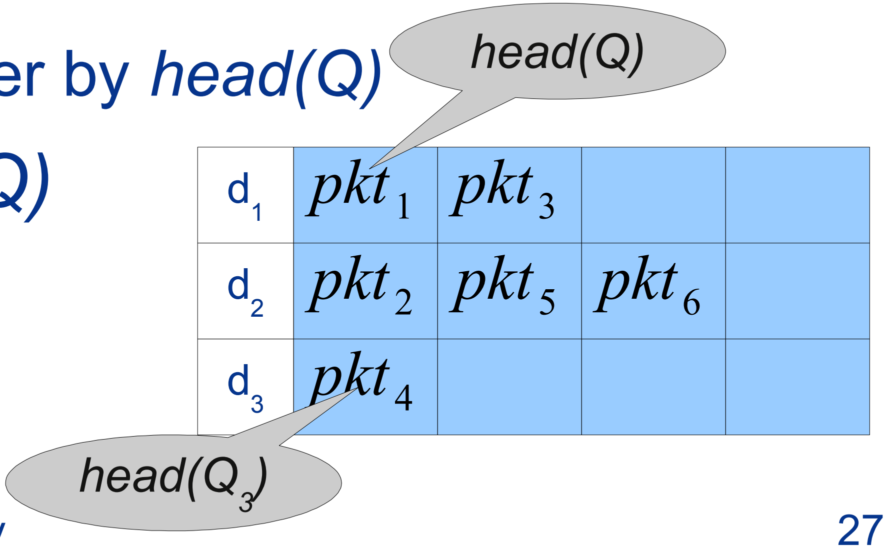
- Receivers must have high probability of decoding packets
- Metric: effective throughput per transmission
- Use best packet mixing strategy given
  - Current link qualities (for SC and NC)
    - Requires channel estimation
  - Current traffic flows (for NC)
    - For determining overheard packets

Channel Estimation in 802.11

- Via RTS/CTS
  - Receiver stamps received power in CTS packet
- Via DATA/ACK
  - Receiver stamps received power in ACK packet
- Estimations not available for initial packets
- Long-lived flows maintain accurate channel estimations

Data Structures for Scheduling

- Maintain per-neighbor FIFO packet queues
  - Q_d is queue for neighbor *d*
  - Denote head-of-queue packet by $head(Q_d)$
- Maintain total order on packets in all queues
  - Ordered by arrival time
  - Denote first packet in total order by *head(Q)*
- Rule: always transmit *head(Q)*
  - Prevents starvation



d_1	pkt_1	pkt_3		
d_2	pkt_2	pkt_5	pkt_6	
d_3	pkt_4			

SC Scheduler: G_{opp}

- Algorithm sketch
  - Select packet *head(Q)* for first layer, denote destination as d_1
  - Iterate over each discrete rate r_1 and destination $d \neq d_1$
    - Constraint: only consider rates supported by destination d_1
    - Compute maximum rate r_2 supported by d in second layer
    - Skip if scheduling transmissions serially has better throughput
    - $N_{d,r1,r2}$ packets from Q_d can be put in second layer; $N_{d,r1,r2} = floor(r_2 / r_1)$
  - Select combination with best throughput: $\max_{(d,\, r1)} \{ r_1 \bullet (1 + N_{d,r1,r2}) \}$

Example mixed packet:

Layer 1	*head(Q) = head(Q_2)*	Destination 2, rate 12 Mbps
Layer 2	*3 packets from Q_4*	Destination 4, rate 36 Mbps

Effective transmission rate is 12 Mbps with 4 packets transmitted, thus effective throughput is 48 Mbps.

Multi-rate NC: *mnetcode*

- SC requires multirate for better gains
- Must extend NC to multirate
- Algorithm sketch
  - Run single-rate COPE algorithm *snetcode(r)* for each rate *r*
    - Constraint: only consider neighbors that support rate *r*
    - Constraint: only consider $head(Q_i)$ for each neighbor *i*
    - Denote number of packets XOR'd at rate *r* as N_r
  - Select combination with best throughput: $\max_r \{ N_r \cdot r \}$

Adding a Second Layer: *SC1*

- Algorithm sketch
  - Select NC packet via *mnetcode*
    - Constraint: must include *head(Q)*
  - Add second layer via G_{opp}

Assume $head(Q) = head(Q_1)$

Layer 1	$head(Q_1) \otimes head(Q_2)$
Layer 2	*3 packets from* Q_4

- Problems
  - No NC used in second-layer packets
    - There is often >1 second-layer packets
  - Limited rate combinations

Joint Algorithm: *SCJ*

- Algorithm sketch
  - Search possible discrete rates for each layer: denote as r_1, r_2
  - Use *snetcode*(r_1) to select packet for first layer
    - Denote number of packets by N_{r1}
  - Use *snetcode*(r_2) to select NC packet(s) for second layer
    - Constraint: only consider neighbors that support r_2 in second layer
    - Repeat $N_{r1,r2}$ times; $N_{r1,r2} = floor(r_2 / r_1)$
  - Select combination with best throughput: $\max_{(r1,\, r2)} \{ r_1 \cdot (N_{r1} + N_{r1,r2}) \}$

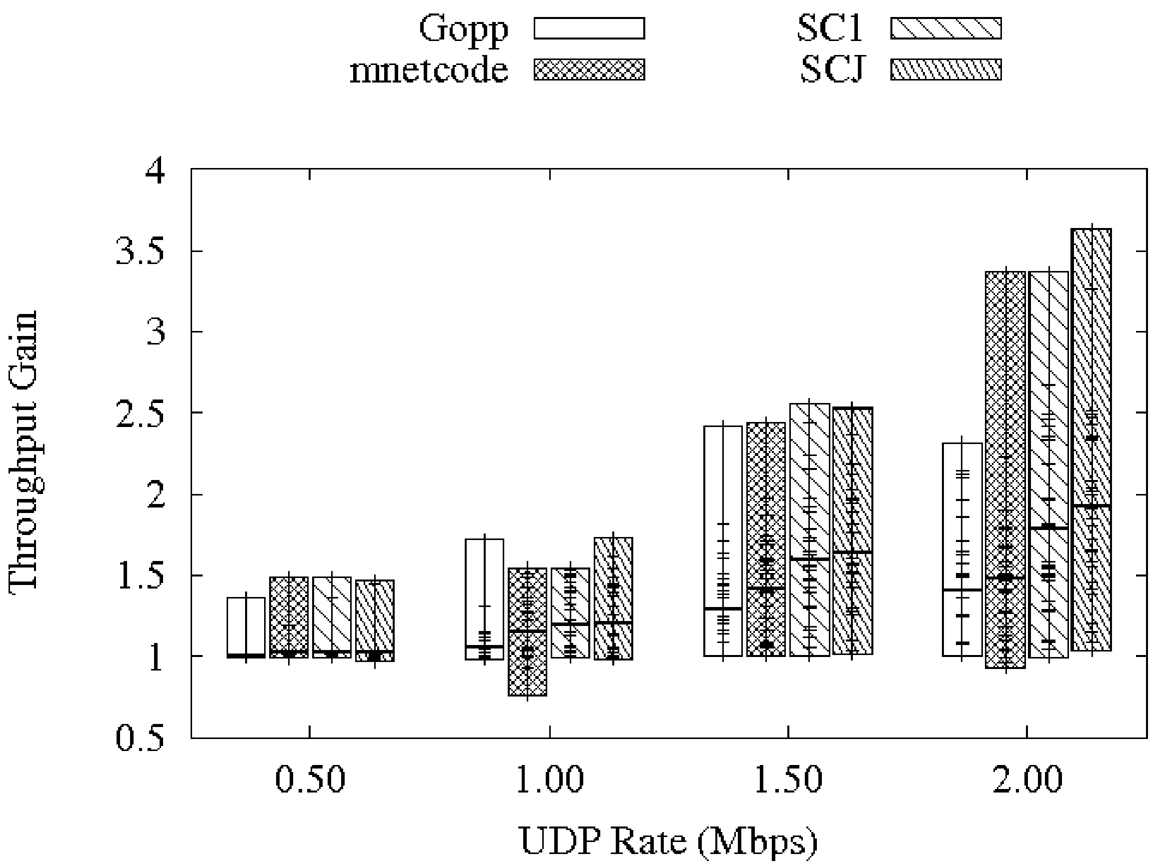
Assume $head(Q) = head(Q_1)$

Layer 1	$head(Q_1) \otimes head(Q_2)$
Layer 2	$head(Q_3) \otimes head(Q_4)$, $head(Q_5) \otimes head(Q_6)$, $head(Q_7) \otimes head(Q_8)$

Evaluations: Setup

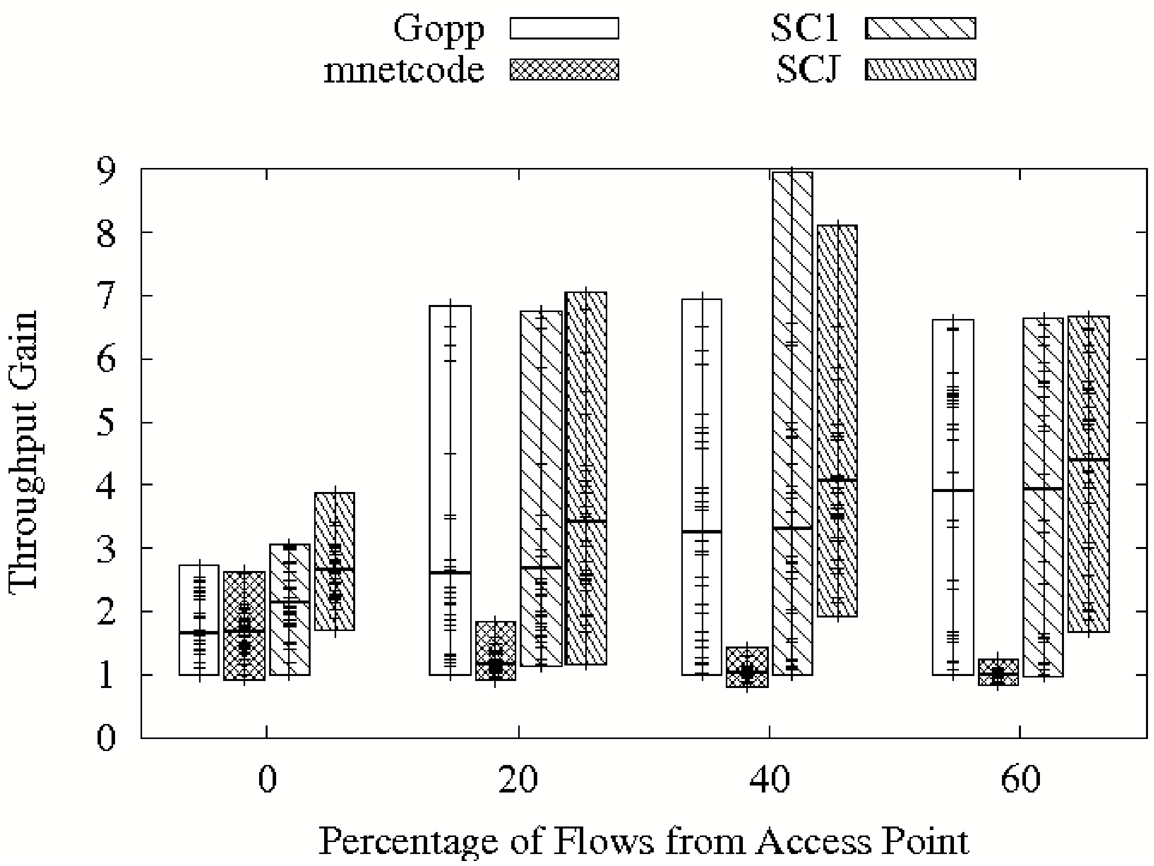
- Algorithms implemented in *ns*-2 version 2.31
- Careful attention to physical layer model
  - Standard *ns*-2 physical layer model does not suffice
  - Use packet error rate curves from actual 802.11a measurements [Doo *et al.* 2004]
  - Packet error rates used for physical layer decoding and rate calculations
- Realistic simulation parameters
  - Parameters produce similar transmission ranges as Cisco Aironet 802.11g card in outdoor environment

Evaluations: Network Demand



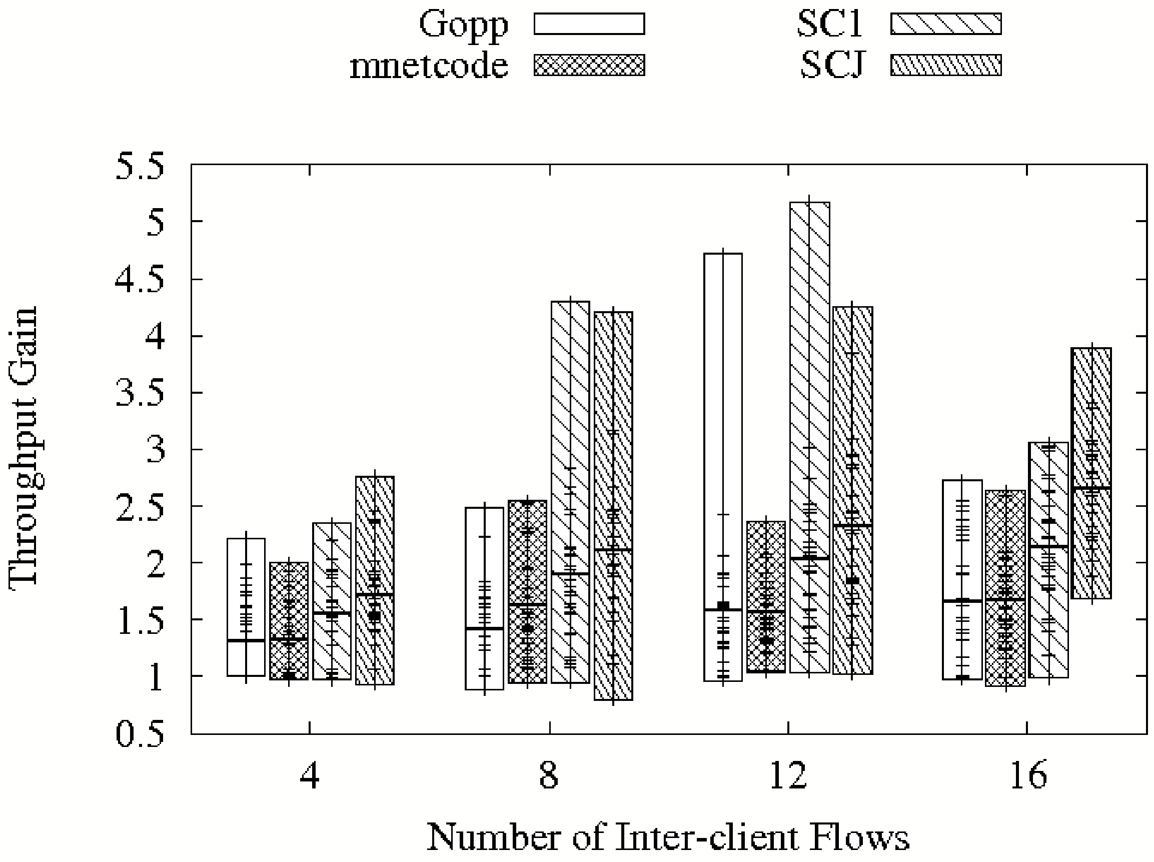
- Setup
  - 1 AP
  - 10 clients
  - 8 flows
  - Vary client sending rate
- Packet mixing gains are sensitive to network demand
- Queues are usually empty with low demand
  - Few mixing opportunites
- NC shows ~3% gain with TCP [Katti *et al.* 2006]

Evaluations: Internet → Client Flows



- Setup
  - 1 AP
  - 20 clients
  - 16 flows
  - Backlogged flows
  - Vary % of flows originating at AP
- SC mixing superior when Internet → client flows are common
- Throughput gains as high as 4.24

Evaluations: Client → Client Flows



- Setup
  - 1 AP
  - 20 clients
  - Backlogged flows
  - Vary # of flows
- Both SC and NC mixing alone improve with # of flows
  - More opportunities
- Gains each SC and NC exploited successfully by *SC1* and *SCJ* schedulers

Talk Outline

- Related work
- Introduction to basic concepts
  - Downlink superposition coding
  - XOR-style network coding
- Construction of mixed packets
  - Analysis and motivation
  - Algorithms
  - Evaluations
- ***GNU Radio testbed***
- Future work

GNU Radio

- Open Source software radio environment
- Architecture
  - RF hardware (USRP) receives and sends raw signals
    - Signals transferred to and from computer via USB
    - Daughtercards available for various frequencies
  - Software environment
    - Signal processing blocks written in C++
    - Python code “glues” the blocks together
- Issues
  - Signal processing is far slower in software
  - Latency between hardware and userspace

Implementation in GNU Radio

- Implementation of proposed schemes is important
  - Handle issues that would block an actual deployment
  - Find other research problems
  - Credibility – most people don't trust wireless simulations
- Contribution thus far
  - First (known) implementation of SC in GNU Radio environment
  - 802.11 MAC implemented (DCF mode only)
  - ~4000 lines C++ code, ~3000 lines Python code

Enabling Further Research

- Many schemes from Information Theory require modifications at physical layer
- Modifying physical layer of normal wireless cards is difficult and time-consuming
    - Impossible for some things
- Benefits
    - Becoming familiar with physical layer enables better designs at the upper layers
    - Realistic evaluation of cross-layer schemes that include the physical layer modifications
    - It's fun

Talk Outline

- Related work
- Introduction to basic concepts
  - Downlink superposition coding
  - XOR-style network coding
- Construction of mixed packets
  - Analysis and motivation
  - Algorithms
  - Evaluations
- GNU Radio testbed
- ***Future work***

Future Work

- Finish setup of GNU Radio testbed
  - Improve performance with higher data rates
- Improvements to TCP are minimal
  - Adjust TCP to keep more packets in network queues
- Handle simultaneous ACKs
  - Each receiver of packet sends ACK to single sender
- Generalize framework for mixing packets
  - Account for other transmission schemes (*e.g.*, relays)
- Design protocols enabling schemes from Info. Theory
- Enable deployment of new schemes in existing networks

Thanks!