iPack: in-Network Packet Mixing for High Throughput Wireless Mesh Networks

Richard Alimi, L. Erran Li, Ramachandran Ramjee, Harish Viswanathan, Y. Richard Yang

Yale University
Bell Labs, Lucent Technologies
Microsoft Research India

IEEE INFOCOM 2008
Wireless Mesh Networks

- City- and community-wide mesh networks widely used
  - New approach to the “last mile” of Internet service
  - In United States alone [muniwireless.com, Jan 16, 2007]:
    - 188 deployed
    - 148 in-progress or planned
Mesh Network Structure

- APs deployed, some connect directly to Internet
  - Street lamps, traffic lights, public buildings
- Clients associate with nearest AP
- Traffic routed to/from Internet via APs (possibly multi-hop)
Limited Capacity of Mesh Networks

• Current mesh networks have limited capacity
  [Li et al. 2001, dailywireless.org 2004]

• Increased usage will only worsen congestion
  – More devices
  – Larger downloads, P2P, video streaming
  – Limited spectrum

• Network-wide transport capacity does not scale
  [Gupta and Kumar 2001]

• Must bypass traditional constraints
Current Coding Techniques

- **Transmitter-side**
  - Downlink superposition coding
    [Cover 1972, Bergmans and Cover 1974]
  - XOR-style network coding
    [Katti et al. 2006]

- **Receiver-side**
  - Uplink superposition coding
  - Analog [Katti et al. 2007] and physical-layer [Zhang et al. 2006] network coding
Packet Mixing for Increased Capacity

- Multiple packets transmitted simultaneously
  - Same timeslot
  - Cross-layer coding techniques
  - No spreading (unlike CDMA)

- Receiver(s) decode own packets
  - Possibly use side-information
Packet Mixing in Mesh Networks

- **Objective**
  - Scheduling algorithms to take advantage of mixing
  - Construct a mixed packet with
    - Maximum effective throughput
    - Sufficiently-high decoding probability at receivers
    - Mixture of coding techniques

- **Currently consider two techniques**
  - Downlink superposition coding (*denoted by SC*)
  - XOR-style network coding (*denoted by NC*)
Talk Outline

- **Basic concepts**
  - *Downlink superposition coding*
  - *XOR-style network coding*
- Mixed packet construction
  - Algorithms
  - Evaluations
- GNU Radio implementation
- Conclusions and future work
Physical Layer Signal Modulation

- Signal has two components: I and Q
- Represented on complex plane

- **Sender**
  - Map bits to symbol (*constellation point*)
- **Receiver**
  - Determine closest symbol and emit bits
Downlink Superposition Coding (SC)

- Basic idea
  - Different message queued for each receiver
  - Transmit messages simultaneously
    - Exploit client channel diversity

- Example
Downlink Superposition Coding (SC)

- **Basic idea**
  - Different message queued for each receiver
  - Transmit messages simultaneously
  - Exploit client channel diversity
- **Example**
  
  ![Diagram showing two laptops receiving different messages]

  - **Weaker receiver**
    - Layer 1
    - “Low resolution”
  - **Stronger receiver**
    - Layer 2
    - “High resolution”
Downlink Superposition Coding (SC)

- **Basic idea**
  - Different message queued for each receiver
  - Transmit messages simultaneously
  - Exploit client channel diversity

- **Example**
  - **Stronger receiver**
    - **Layer 2**
    - "High resolution"
  - **Weaker receiver**
    - **Layer 1**
    - "Low resolution"
SC Decoding: Successive Interference Cancellation

Weaker Receiver

Decode Layer 1
SC Decoding: Successive Interference Cancellation

(1) Decode Layer 1

Stronger Receiver
SC Decoding: Successive Interference Cancellation

(1) Decode Layer 1

(2) Subtract and decode Layer 2

Stronger Receiver
XOR-style Network Coding

- **Basic idea**
  - Nodes remember overheard and sent messages
  - Transmit bitwise XOR of packets: $1 \oplus 2 \oplus \ldots \oplus n$
  - Receivers decode if they already know $n-1$ packets

- **Example**
Talk Outline

• Basic concepts
  – Downlink superposition coding
  – XOR-style network coding

• Mixed packet construction
  – Algorithms
  – Evaluations

• GNU Radio implementation

• Conclusions and future work
Packet Mixing Example

- 4 Flows
  - Packet $d$ has dest $R_d$
- Without packet mixing
  - 8 transmissions required
- With packet mixing
  - 5 transmissions required

Routing link          Overhearing link
Packet Mixing Example

(1) $R_2$ sends Pkt 1 to AP
Packet Mixing Example

(1) $R_2$ sends Pkt 1 to AP

(2) $R_4$ sends Pkt 2 to AP
Packet Mixing Example

(1) $R_2$ sends Pkt 1 to AP
(2) $R_4$ sends Pkt 2 to AP
(3) $R_1$ sends Pkt 3 to AP
Packet Mixing Example

(1) $R_2$ sends Pkt 1 to AP
(2) $R_4$ sends Pkt 2 to AP
(3) $R_1$ sends Pkt 3 to AP
(4) $R_3$ sends Pkt 4 to AP
Packet Mixing Example

(1) $R_2$ sends Pkt 1 to AP

(2) $R_4$ sends Pkt 2 to AP

(3) $R_1$ sends Pkt 3 to AP

(4) $R_3$ sends Pkt 4 to AP

(5) AP sends mixed packet using SC:

(a) Layer 1: \[ \begin{array}{cc} 1 & \otimes & 2 \\ \end{array} \]

(b) Layer 2: \[ \begin{array}{cc} 3 & \otimes & 4 \\ \end{array} \]
Scheduling under Packet Mixing

- Per-neighbor FIFO packet queues
  - $Q_d$ is queue for neighbor $d$ – first denoted by $\text{head}(Q_d)$

- Total order on packets in all queues
  - Ordered by arrival time – first denoted by $\text{head}(Q)$

- Rule: always transmit $\text{head}(Q)$
  - Prevents starvation
SC Scheduler: $G_{opp}$

- Prior work: mesh network scheduling with SC [Li et al. 2007]

- Overview
  - SC Layer 1: Select $\text{head}(Q)$ with dest $d_1$ at rate $r_1$
  - SC Layer 2: Select $\text{floor}(r_2 / r_1)$ packets for dest $d_2 \neq d_1$ at rate $r_2$
  - Allows different rates for each layer
    - Selects best rates given current channels
    - Ensures sufficient decoding probabilities

<table>
<thead>
<tr>
<th>Layer 1</th>
<th>$\text{head}(Q_2)$</th>
<th>Destination 2, rate 12 Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 2</td>
<td>3 packets from $Q_4$</td>
<td>Destination 4, rate 36 Mbps</td>
</tr>
</tbody>
</table>

- Rate: 12 Mbps
- Packets: 4
- Throughput: 48 Mbps
Multirate NC: *mnetcode*

- SC requires multirate for better gains, so extend NC as well

- **Algorithm**
  - Run single-rate COPE algorithm $snetcode(r)$ for each rate $r$ and select best
    - Skip rate if not supported by neighbor
    - Only consider $head(Q_d)$ for each neighbor $d$
    - $N$ packets XOR’d at rate $r$
    - Effective throughput is $N \cdot r$
Simple Cross-layer Mixing: SC1

- Utilize physical- and network-layer coding

- Algorithm
  - SC Layer 1: Select NC packet with \textit{mnetcode}
    - Must include \textit{head}(Q)
  - SC Layer 2: Select packet with \textit{G}_{opp}

- Problems
  - No NC used in Layer 2 packets
  - Limited rate combinations
Joint Algorithm: SCJ

- Improved utilization of physical- and network-layer coding

- Algorithm
  - Iterate over discrete rates for each layer, $r_1$ and $r_2$
    - SC Layer 1: $snetcode(r_1)$ selects packet, $N_1$ packets encoded
    - SC Layer 2: $snetcode(r_2)$ selects $\text{floor}(r_2 / r_1)$, $N_2$ packets encoded
      - Only consider neighbors that support $r_2$ in second layer
    - Effective throughput is $r_1 \cdot (N_1 + N_2)$

<table>
<thead>
<tr>
<th>Layer 1</th>
<th>(\text{head}(Q_1) \otimes \text{head}(Q_2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 2</td>
<td>(\text{head}(Q_3) \otimes \text{head}(Q_4), \text{head}(Q_5) \otimes \text{head}(Q_6), \text{head}(Q_7) \otimes \text{head}(Q_8))</td>
</tr>
</tbody>
</table>
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 opportunities

- 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

- Basic concepts
  - Downlink superposition coding
  - XOR-style network coding
- Mixed packet construction
  - Algorithms
  - Evaluations
- GNU Radio implementation
- Conclusions and future work
GNU Radio Implementation

- Open Source software radio
  - RF frontend hardware (USRP)
  - Signal processing in software

- Components
  - Implementation of SC in GNU Radio environment
  - 802.11 MAC implemented with NC support

- Measurement Results

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Norm. exp. trans. time</th>
<th>Gain ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Coding</td>
<td>3.92</td>
<td>1</td>
</tr>
<tr>
<td>Superposition</td>
<td>2.88</td>
<td>1.4</td>
</tr>
<tr>
<td>Network coding</td>
<td>2.30</td>
<td>1.7</td>
</tr>
<tr>
<td>iPack</td>
<td>2.07</td>
<td>2.0</td>
</tr>
</tbody>
</table>
Talk Outline

• Basic concepts
  – Downlink superposition coding
  – XOR-style network coding

• Construction of mixed packets
  – Algorithms
  – Evaluations

• GNU Radio implementation

• Conclusions and future work
Conclusions and Future Work

- Packet mixing increases throughput
  - Exploit packet mixing at network and physical layers
  - Cross-layer coding techniques can significantly improve throughput

- Ongoing and Future Work
  - Expand and improve implementation testbed
  - Improve TCP gains
  - Simultaneous ACKs
  - Generalize packet-mixing framework
Thanks!