15-744: Computer Networking

L-10 Wireless Broadcast

Taking Advantage of Broadcast

- Opportunistic forwarding
- Network coding
- Assigned reading
  - ExOR: Opportunistic Multi-Hop Routing for Wireless Networks
  - XORs In The Air: Practical Wireless Network Coding (sections 1-3)
- Optional reading:
  - Randomness in Opportunistic Wireless Routing

Outline

- Opportunistic forwarding (ExOR)
- Network coding (COPE)
- Combining the two (MORE)

Background

- Wireless communication is broadcast based
  - All nodes “within range” receive the packet and locally decide whether to accept it
  - Can listen in for free!
  - But is it really free?
- Properties of wireless channels are very dynamic unpredictable
  - Large scale path loss: distance, obstacles
  - Fast fading: mobility, multi-path
  - Fast fading is uncorrelated on different channels
Initial Approach: Traditional Routing

- Identify a route, forward over links
- Abstract radio to look like a wired link

Radios Aren’t Wires

- Every packet is broadcast
- Reception is probabilistic

Exploiting Probabilistic Broadcast

- Decide who forwards after reception
- Goal: only closest receiver should forward
- Challenge: agree efficiently and avoid duplicate transmissions

Why ExOR Might Increase Throughput

- Best traditional route over 50% hops: $3(1/0.5) = 6$ tx
- Throughput $\geq \frac{1}{6}$ transmissions
- ExOR exploits lucky long receptions: 4 transmissions
- Assumes probability falls off gradually with distance
Why ExOR Might Increase Throughput

- Traditional routing: \(\frac{1}{0.25} + 1 = 5\) tx
- ExOR: \(\frac{1}{1 - (1 - 0.25)^4} + 1 = 2.5\) transmissions
- Assumes independent losses

ExOR Batching

- Challenge: finding the closest node to have rx’d
- Send batches of packets for efficiency
- Node closest to the dst sends first
  - Other nodes listen, send remaining packets in turn
- Repeat schedule until dst has whole batch

Reliable Summaries

- tx: \{2, 4, 10 ... 97, 98\}
- batch map: \{1, 2, 6, ... 97, 98, 99\}
- Repeat summaries in every data packet
- Cumulative: what all previous nodes rx’d
- This is a gossip mechanism for summaries

Batch Map

- Each node has list of nodes, ranked based on how close they are to the destination
- Each node also listens in on all transmissions
- Batch map lists for each packet the node the node closest to the destination that has the packet
  - Based on overheard packets and the batch maps they included
  - Destination starts with its batch map
Priority Ordering

- Goal: nodes “closest” to the destination send first
- Sort by ETX metric to dst
  - Nodes periodically flood ETX “link state” measurements
  - Path ETX is weighted shortest path (Dijkstra’s algorithm)
- Source sorts, includes list in ExOR header

Using ExOR with TCP

- Batching requires more packets than typical TCP window

Discussion

- Exploits radio properties, instead of hiding them
  - Different from Internet: hides the physical layer
- Leverages properties of wireless channels
  - Fading on wireless channels is uncorrelated (if sufficiently separated)
- Scalability?
- Lots of parameters!
  - Waiting times, 10%, ETX, ..
- Overheads?

Outline

- Opportunistic forwarding (ExOR)
- Network coding (COPE)
- Combining the two (MORE)
Background

- Famous butterfly example:

- All links can send one message per unit of time
  - Coding increases overall throughput

Background

- Bob and Alice

Relay

Require 3 transmissions

Background

- Bob and Alice

Relay

Require 4 transmissions

Coding Gain

- Coding gain = 4/3
Throughput Improvement

- UDP throughput improvement ~ a factor 2 > 4/3 coding gain

Coding Gain: more examples

Without opportunistic listening, coding \([+MAC]\) gain=\(2N/(1+N)\) \(\rightarrow\) 2.
With opportunistic listening, coding gain + MAC gain \(\rightarrow\) \(\infty\)

COPE (Coding Opportunistically)

- Overhear neighbors’ transmissions
- Store these packets in a **Packet Pool** for a short time
- Report the packet pool info. to neighbors
- Determine what packets to code based on the info.
- Send encoded packets

Opportunistic Coding

<table>
<thead>
<tr>
<th>B’s queue</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>A</td>
</tr>
<tr>
<td>P2</td>
<td>C</td>
</tr>
<tr>
<td>P3</td>
<td>C</td>
</tr>
<tr>
<td>P4</td>
<td>D</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coding</th>
<th>Is it good?</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1+P2</td>
<td>Bad (only C can decode)</td>
</tr>
<tr>
<td>P1+P3</td>
<td>Better coding (Both A and C can decode)</td>
</tr>
<tr>
<td>P1+P3+P4</td>
<td>Best coding (A, C, D can decode)</td>
</tr>
</tbody>
</table>
Packet Coding Algorithm

- **When to send?**
  - Option 1: delay packets till enough packets to code with
  - Option 2: never delaying packets -- when there's a transmission opportunity, send packet right away

- **Which packets to use for XOR?**
  - Prefer XOR-ing packets of similar lengths
  - Never code together packets headed to the same next hop
  - Limit packet re-ordering
  - XOR-ing a packet as long as all its nexthops can decode it with a high enough probability

Packet Decoding

- **Where to decode?**
  - Decode at each intermediate hop

- **How to decode?**
  - Upon receiving a packet encoded with n native packets
    - find n-1 native packets from its queue
    - XOR these n-1 native packets with the received packet to extract the new packet

Prevent Packet Reordering

- Packet reordering due to async acks degrade TCP performance

- **Ordering agent**
  - Deliver in-sequence packets immediately
  - Order the packets until the gap in seq. no is filled or timer expires

Summary of Results

- Improve UDP throughput by a factor of 3-4

- Improve TCP by
  - wo/ hidden terminal: up to 38% improvement
  - w/ hidden terminal and high loss: little improvement

- Improvement is largest when uplink to downlink has similar traffic

- Interesting follow-on work using analog coding
Reasons for Lower Improvement in TCP

- COPE introduces packet re-ordering
- Router queue is small → smaller coding opportunity
  - TCP congestion window does not sufficiently open up due to wireless losses
- TCP doesn’t provide fair allocation across different flows

Discussion

- Wired vs. wireless coding
- Traffic patterns
- Scale

Outline

- Opportunistic forwarding (ExOR)
- Network coding (COPE)
- Combining the two (MORE)

Use Opportunistic Routing

Opportunistic routing promises large increase in throughput
But

- Overlap in received packets $\rightarrow$ Routers forward duplicates

ExOR

- State-of-the-art opp. routing, ExOR imposes a global scheduler:
  - Requires full coordination; every node must know who received what
  - Only one node transmits at a time, others listen

Global Scheduling

- Global coordination is too hard
  - One transmitter $\rightarrow$ You lost spatial reuse!

MORE (Sigcomm07)

- Opportunistic routing with no global scheduler and no coordination
  - Uses random network coding
  - Experiments show that randomness outperforms both current routing and ExOR
Go Random

Each router forwards random combinations of packets

Without coding → source retransmits all 4 packets

Randomness prevents duplicates

No scheduler; No coordination

Simple and exploits spatial reuse

Random Coding Benefits Multicast

Without coding → source retransmits all 4 packets

With random coding → 2 packets are sufficient

MORE

- Source sends packets in batches
- Forwarders keep all heard packets in a buffer
- Nodes transmit linear combinations of buffered packets
MORE

- Source sends packets in batches
- Forwarders keep all heard packets in a buffer
- Nodes transmit linear combinations of buffered packets

\[
\begin{align*}
A &= aP_1 + bP_2 + cP_3 = \alpha \beta \gamma \\
src &= P_1 \\
P_2 &= P_3 \\
\text{dst} &= 2 \cdot 4.1.3 + 1 \cdot 0.2.1 = 8.4.7
\end{align*}
\]

- Destination decodes once it receives enough combinations
  - Say batch is 3 packets
    \[
    \begin{align*}
    1: &
    \begin{align*}
    0 &= 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 1 \\
    5 &= 1 \cdot 0 + 4 \cdot 0 + 5 \cdot 1 \quad \Rightarrow \quad 5,4,5
    
    4 &= 1 \cdot 0 + 5 \cdot 0 + 5 \cdot 1 \quad \Rightarrow \quad 4,5,5
    \end{align*}
    \end{align*}
    \]
  - Destination acks batch, and source moves to next batch

Summary/Midterm

- Wireless broadcast enables new protocol designs
- Key challenge is coordination

- Midterm
  - Closed book, coverage includes today
  - Similar in style to sample
  - Will post HW solutions
  - Office hours: Monday 9-10
    - Will post Bin’s office hours

MORE

- Source sends packets in batches
- Forwarders keep all heard packets in a buffer
- Nodes transmit linear combinations of buffered packets

- Destination decodes once it receives enough combinations
  - Say batch is 3 packets
    \[
    \begin{align*}
    1: &
    \begin{align*}
    0 &= 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 1 \\
    5 &= 1 \cdot 0 + 4 \cdot 0 + 5 \cdot 1 \quad \Rightarrow \quad 5,4,5
    
    4 &= 1 \cdot 0 + 5 \cdot 0 + 5 \cdot 1 \quad \Rightarrow \quad 4,5,5
    \end{align*}
    \end{align*}
    \]
  - Destination acks batch, and source moves to next batch