XORs in the Air: Practical Wireless Network Coding

Sachin Katti, Hariharan Rahul, Wenjun Wu, Dina Katabi, Muriel Médard, Jon Crowcroft

ACM SIGCOMM 2006
The Problem

Increasing the throughput of dense wireless mesh networks
Current Approach

- Requires 4 transmissions
- Can it be done in fewer transmissions?
A Network Coding Approach

- Requires 3 transmissions instead of 4
- Increased throughput
Beyond Duplex Flows

- Two flows that intersect at a router
Beyond Duplex Flows
Beyond Duplex Flows
Beyond Duplex Flows
Beyond Duplex Flows
Beyond Duplex Flows

- Again 3 transmissions instead of 4
Two Departures

- Accept wireless as a broadcast medium
  - Dispose of the point to point abstraction
- Routers mix bits in packets, then forward them: Network Coding!
COPE (Coding Opportunistically)

- Large throughput increase
- First integration of network coding into the network stack
- New network coding algorithm that deals with general unicast flows
COPE - Snooping

- Exploit wireless broadcast
- Every node snoops on all packets
- A node stores all heard packets for a limited time
COPE - Snooping

- Exploit wireless broadcast
- Every node snoops on all packets
- A node stores all heard packets for a limited time
COPE - Snooping

- Exploit wireless broadcast
- Every node snoops on all packets
- A node stores all heard packets for a limited time

Node sends Reception Reports to tell its neighbors what packets it heard
- Reports are piggybacked on packets
- If no packets to send, periodically send reports
COPE - Coding

- To send packet $p$ to neighbor A, XOR $p$ with packets already known to A
  - Thus, A can decode

- But how can multiple neighbors benefit from a single transmission?
Efficient Coding

Arrows show next-hop
Efficient Coding

- **Bad Coding**: Only one neighbor benefits from one transmission
Efficient Coding

- **Good Coding**: Two neighbors benefit from one transmission!
Efficient Coding

- **Best Coding**: Three neighbors benefit from one transmission!
Efficient Coding

- XOR \( n \) packets together iff the next hop of each packet already has the other \( n-1 \) packets apart from the one he wants
But how does a node know what packets a neighbor has?

- Reception reports
- But reception reports may be late or get lost
- Make informed guesses based on delivery rate between the two nodes
- If error occurs, recover by retransmission
Design Choices

- Sit transparently between IP and MAC
- **Opportunistic**: Code packets if possible, if not forward without coding
- Do not delay packets
Performance
COPE Implementation

- Linux
- Click + Roofnet
- Userspace module
Alice-and-Bob

- Requires 3 transmissions instead of 4
  - Expected throughput gain of $\frac{4}{3} = 1.33$
Alice-and-Bob (TCP)

Throughput increase in line with analysis
Alice-and-Bob (UDP)

- COPE almost doubles the throughput
Alice-and-Bob (UDP)

COPE almost doubles the throughput
Why More Than 1.33?

COPE alleviates the mismatch between MAC's capacity allocation and the congestion at a node.
Coding Gain

- Reduction in #Transmissions
- For Alice-and-Bob scenario, Coding Gain is 4/3
-反射 gains when nodes are not backlogged

Coding + MAC Gain

- Improvement of draining rate at bottlenecks
- For Alice-and-Bob scenario, Coding + MAC Gain is 2
- Reflects gains when nodes are backlogged

Theoretically,

- Coding gain is bounded by 2
- Coding + MAC gain can be infinite
Large-Scale Experiments

- Wireless testbed
  - 20 nodes
  - 2 floors

- Experiments
  - Pick sender and receiver randomly
  - Transfer size based on actual measurements
  - Flow arrivals are Poisson
TCP in large network

<table>
<thead>
<tr>
<th>With Hidden Terminals</th>
<th>No Hidden Terminals</th>
</tr>
</thead>
<tbody>
<tr>
<td>With or without coding</td>
<td>With or without coding</td>
</tr>
<tr>
<td>■ High loss rates (14-40%) due to collisions</td>
<td>■ Low loss rates (1-2%)</td>
</tr>
<tr>
<td>■ TCP doesn’t send much</td>
<td>■ TCP sends</td>
</tr>
<tr>
<td>■ Medium under-utilized</td>
<td>■ Coding opportunities</td>
</tr>
<tr>
<td>■ No coding opportunities</td>
<td></td>
</tr>
</tbody>
</table>
TCP Without Hidden Terminals

With no hidden terminals, COPE substantially increases TCP throughput.
UDP is the same with or without hidden terminals.
UDP in large network

About 4-fold throughput increase in congested network
Conclusion

- COPE: a new approach to wireless
- Large throughput increase
- First integration of network coding into the network stack
- New network coding algorithm that deals with general unicast flows
Pros and Cons

- Cross the chasm by putting network coding into practice
- Research Exemplar
  - A radical new idea
  - Careful engineering
  - Extensive testing
- Applicable to sensor network with infrastructure
References

- Talk Slides for Allerton 2005 and ACM SIGCOMM 2006