Network Coding

Lecture 1

Dina Katabi
What is Network Coding?

- Generalization of store and forward networks
- Routers mix/code packets’ content before forwarding
The Butterfly Example

Source wants to multicasts $a$ and $b$ to both destinations

Link capacity is 1 message/second
The Butterfly Example

Source wants to multicasts $a$ and $b$ to both destinations

Link capacity is 1 message/second

\[ a \oplus (a \oplus b) = b \]

This multicast connection is impossible without network coding
What is Network Coding Good for?

- Throughput
Improves Multicast Throughput

Without network coding, multicast throughput is 1.5
With network coding, multicast throughput is 2
Improves Unicast Throughput

s1 wants to send a to d1, and s2 wants to send b to d2
Improves Unicast Throughput

s1 wants to send a to d1, and s2 wants to send b to d2

Without net-coding, throughput is 1
With net-coding, throughput is 2
What is Network Coding Good for?

- Throughput
- Robustness
Robustness to Packet Loss

Each link has a loss rate of $\varepsilon$

With source coding, the maximum rate is $(1 - \varepsilon)(1 - \varepsilon)$

E.g., $\varepsilon = 0.2 \rightarrow$ max-rate is 0.64

With network coding, the maximum rate is $(1 - \varepsilon)$
Robustness to Packet Loss

Each link has a loss rate of \( \varepsilon \)

\[
\text{Delivery} = (1 - \varepsilon)
\]

With source coding, the maximum rate is \((1 - \varepsilon)(1 - \varepsilon)\)

- E.g., \(\varepsilon = 0.2 \rightarrow \text{max-rate is 0.64}\)

With network coding, the maximum rate is \((1 - \varepsilon)\)

- E.g., \(\varepsilon = 0.2 \rightarrow \text{max-rate is 0.8}\)
Robustness to Randomness

Coupon Collector Problem

Problem: \( n \) unique coupons; a collector samples randomly

Without coding
- Need a sample size of about \( n \log(n) \) to collect all unique coupons

With random coding
- Need \( n \) samples to collect all unique coupons
  - e.g., 3 coupons \( c_1, c_2 \) and \( c_3 \)

\[
Y_1 = c_1 + 3c_2 + 4c_3 \\
Y_2 = 5c_1 + c_2 + 7c_3 \\
Y_3 = c_1 + c_2
\]
Two Types of Network Coding

Inter-session

- Codes packets across connections
- Increases Throughput
- Mainly Unicast

Intra-session

- Codes packets within a connection
- Robustness to packet loss
- Mainly multicast
COPE
An Example of Inter-Session Network Coding
Increased Demands for Wireless Networks

City-wide Network

Digital Home

Emergency Networks
But, wireless networks struggle with low throughput, particularly in dense deployments

Need a solution!
Current Approach

Current Approach \( \rightarrow \) Requires 4 transmissions
But can we do better?
Network Coding $\rightarrow$ 3 transmissions instead of 4

Network Coding Increases Throughput
Beyond Duplex Communications

Two communications that intersect at a router
Beyond Duplex Communications

S1

S2

D1 overhears packet

D2 overhears packet

Router
Beyond Duplex Communications

Router

\[ \text{XOR} \]

S1

S2

D2

D1

3 transmissions instead of 4 \(\rightarrow\) Higher Throughput
Differences from Traditional Wireless Networks

- Embrace the *broadcast* nature of wireless
- Dispose of the point-to-point abstraction

- Routers mix bytes across packets, then forward them → Network Coding
COPE

- Opportunistic Listening
- Opportunistic Coding
Opportunistic Listening

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

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

- 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
Opportunistic 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?
Which Packets to Code Together?

Arrows show next-hop
Which Packets to Code Together?

Bad Coding
Only one neighbor benefits from one transmission
Which Packets to Code Together?

Good Coding
Two neighbors benefit from one transmission!
Which Packets to Code Together?

Best Coding
Three neighbors benefit from one transmission!
Which Packets to Code Together?

\[ \text{xor} \ n \text{ packets together iff the next hop of each packet already has the other } n-1 \text{ packets apart from the one it wants} \]
But, to decode a node needs to know which packets are XOR-ed

- Put that information in the header of the coded packet
- E.g., $P = P_1 + P_2 \rightarrow P$’s header will say $P_1, P_2$
But how does a node know what packets a neighbor has?

- Reception Reports
- But reception reports may get lost or arrive too late
- Make informed guesses based on the delivery rate between the two nodes
- Yes, error might occur and we recover by encoding and retransmitting
COPE’s Characteristics

- COPE is a forwarding mechanism
  - It sits transparently between IP and MAC
  - Routing is unmodified (i.e., shortest path)
- Opportunistic → Code packets if possible, if not forward without coding
- Does not delay packets
Performance
COPE is implemented in Linux
Alice-and-Bob Experiment

3 transmissions instead of 4
⇒ Throughput gain is $4/3 = 1.33$
Results of the Alice-and-Bob

Ratio of Throughput with COPE to Current Approach

COPE almost doubles the throughput
Why More Gain than 1.33?

COPE alleviates the mismatch between MAC's capacity allocation and the congestion at a node.
<table>
<thead>
<tr>
<th>Coding Gain</th>
<th>Coding+MAC Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reduction in #Transmissions</strong></td>
<td><strong>Improvement of Draining Rate at Bottlenecks</strong></td>
</tr>
<tr>
<td>In George-and-Willy scenario, Coding Gain is $4/3 = 1.33$</td>
<td>In George-and-Willy scenario, Coding+MAC Gain is 2</td>
</tr>
<tr>
<td>Nodes not backlogged</td>
<td>Nodes backlogged</td>
</tr>
</tbody>
</table>

*Can show that*

- 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 from Internet measurements
- Flows arrive according to Poisson and send as fast as the MAC permits
- UDP traffic
Experiments in a 20-Node Network
Experiments in a 20-Node Network

Network Throughput (Mb/s)

Traffic Demands (Mb/s)
Experiments in a 20-Node Network

Network Throughput (Mb/s)

Traffic Demands (Mb/s)

Current Approach
Experiments in a 20-Node Network

Network Throughput (Mb/s)

COPE

Current Approach

Several fold throughput increase!
TCP in large network

With Hidden Terminals
- With or without coding
  - High loss rates (14-40%) due to collisions
  - TCP doesn’t send much
  - Medium under-utilized
  - No coding opportunities

No Hidden Terminals
- With or without coding
  - Low loss rates (1-2%)
  - TCP sends
  - Coding opportunities
With no hidden terminals, COPE substantially increases TCP throughput.
Video with/out Network Coding
Extension: Coding-Aware Routing [SRB07]

Red Flow:  A ➔ D
Green Flow: D ➔ F

6 transmissions to send two packets

No coding opportunities since flows take different routes
Extension: Coding-Aware Routing [SRB07]

Pick routes that enable coding

5 transmissions instead of 6

Modified routing that maximizes throughput given coding

C & E can now code packets

Red Flow: A → D
Green Flow: D → F
Summary

- COPE harnesses broadcast to achieve in-network compression
- Inter-session network coding for unicast
- Significant throughput gains depending on topology and traffic