

15-744: Computer Networking

L-12 Wireless Broadcast



Taking Advantage of Broadcast



- Opportunistic forwarding
- Network coding
- Assigned reading
 - XORs In The Air: Practical Wireless Network Coding
 - ExOR: Opportunistic Multi-Hop Routing for Wireless Networks

2

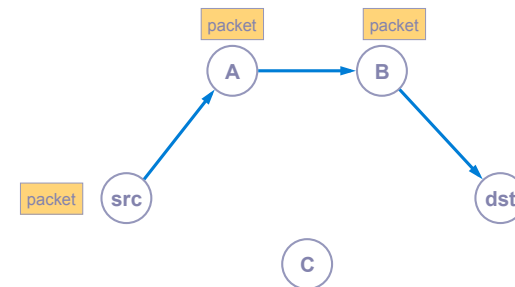
Outline



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

3

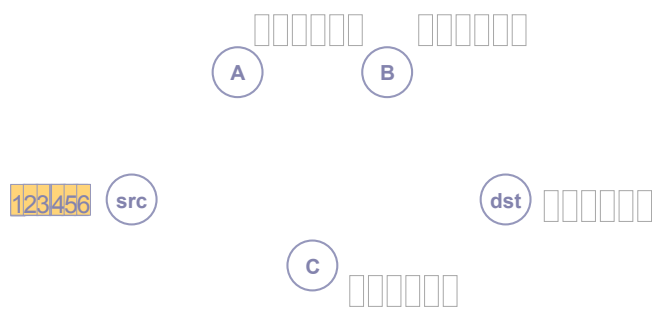
Initial Approach: Traditional Routing



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

4

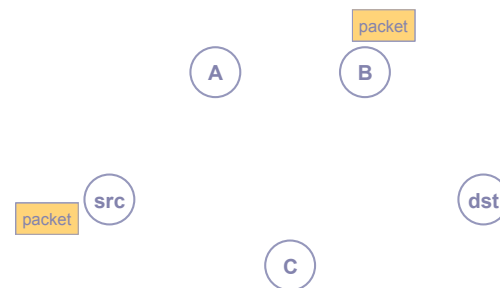
Radios Aren't Wires



- Every packet is broadcast
- Reception is probabilistic

5

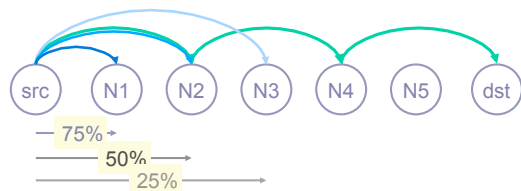
Exploiting Probabilistic Broadcast



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

6

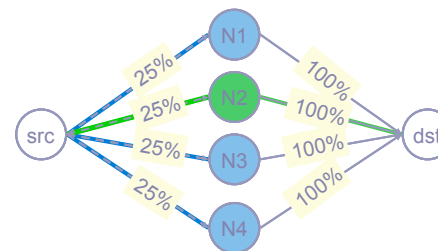
Why ExOR Might Increase Throughput



- Best traditional route over 50% hops: $3(1/0.5) = 6$ tx
- Throughput $\cong 1/\#$ transmissions
- ExOR exploits lucky long receptions: 4 transmissions
- Assumes probability falls off gradually with distance

7

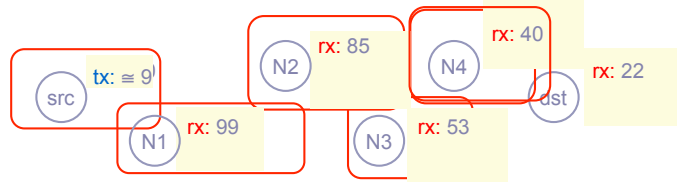
Why ExOR Might Increase Throughput



- Traditional routing: $1/0.25 + 1 = 5$ tx
- ExOR: $1/(1 - (1 - 0.25)^4) + 1 = 2.5$ transmissions
- Assumes independent losses

8

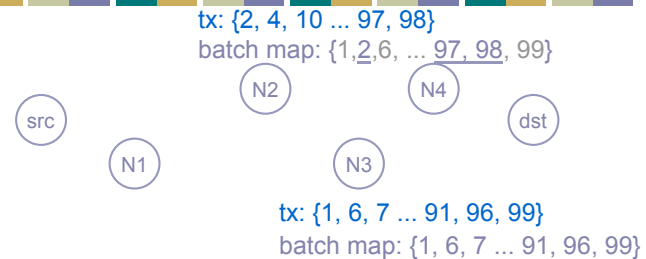
ExOR Batching



- 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

9

Reliable Summaries



- Repeat summaries in every data packet
- Cumulative: what all previous nodes rx'd
- This is a gossip mechanism for summaries

10

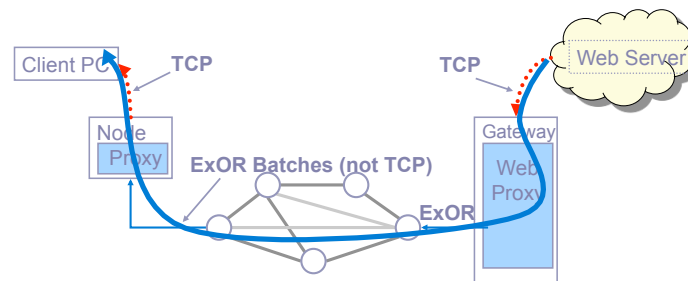
Priority Ordering



- 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

11

Using ExOR with TCP



- Batching requires more packets than typical TCP window

12

Discussion

- Exploits radio properties, instead of hiding them
- Scalability?
- Parameters – 10%?
- Overheads?

13

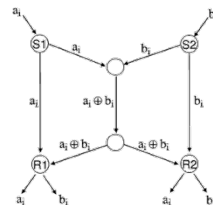
Outline

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

14

Background

- Famous butterfly example:

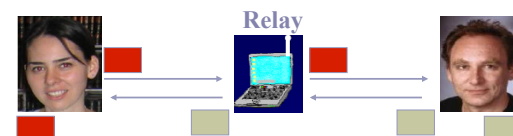


- All links can send one message per unit of time
 - Coding increases overall throughput

15

Background

- Bob and Alice

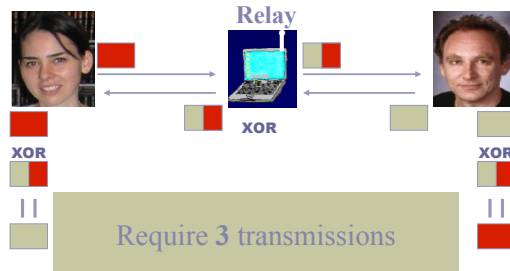


Require 4 transmissions

16

Background

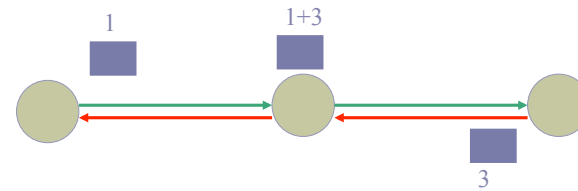
- Bob and Alice



17

Coding Gain

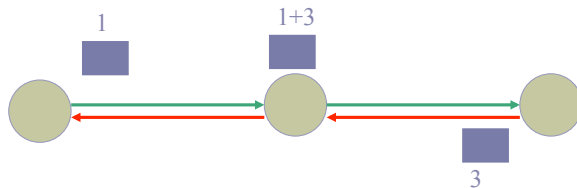
- Coding gain = $4/3$



18

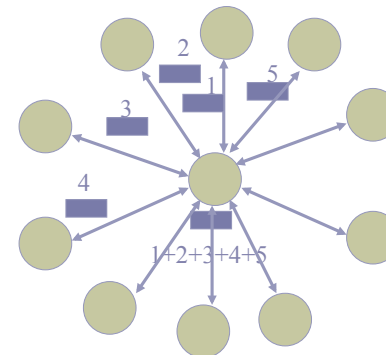
Throughput Improvement

- UDP throughput improvement $\sim$ a factor 2 > $4/3$ coding gain



19

Coding Gain: more examples



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

20

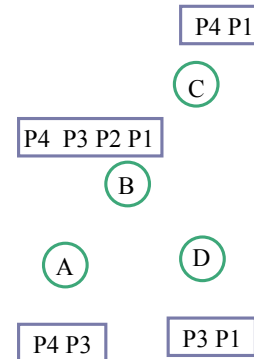
COPE (Coding Opportunistically)



- 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

21

Opportunistic Coding



B's queue	Next hop
P1	A
P2	C
P3	C
P4	D

Coding	Is it good?
P1+P2	Bad (only C can decode)
P1+P3	Better coding (Both A and C can decode)
P1+P3+P4	Best coding (A, C, D can decode)

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
 - XORing a packet as long as all its nexthops can decode it with a high enough probability

23

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

24

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

25

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

26

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

27

Discussion



- Wired vs. wireless coding
- Traffic patterns
- Scale

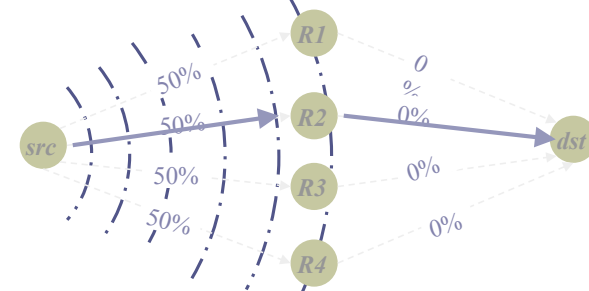
28

Outline

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

29

Use Opportunistic Routing

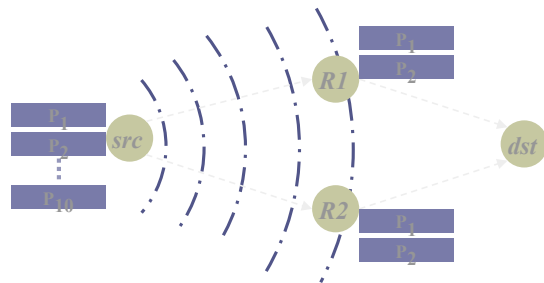


Opportunistic routing promises large increase in throughput

30

But

- Overlap in received packets → Routers forward duplicates



31

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

32

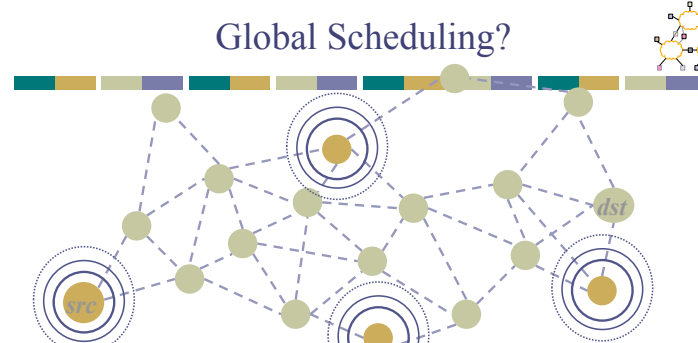
Global Scheduling?



- Global coordination is too hard
- One transmitter

33

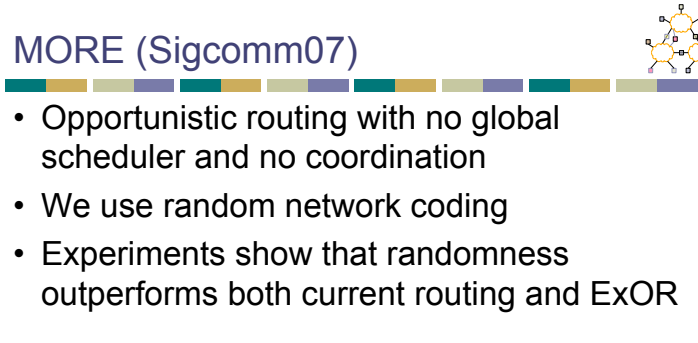
Global Scheduling?



Does opportunistic routing have to be so complicated?

34

MORE (Sigcomm07)

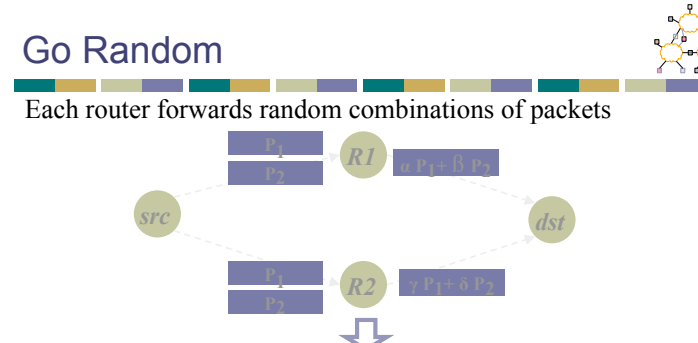


- Opportunistic routing with no global scheduler and no coordination
- We use random network coding
- Experiments show that randomness outperforms both current routing and ExOR

35

Go Random

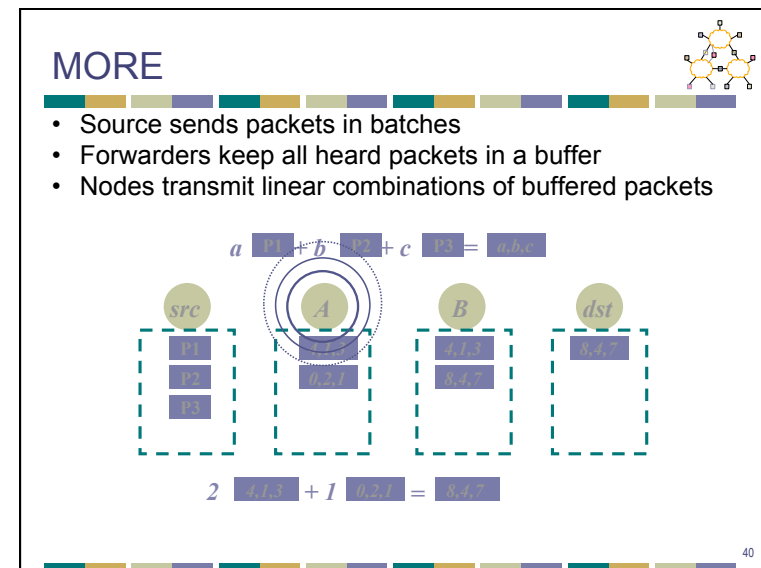
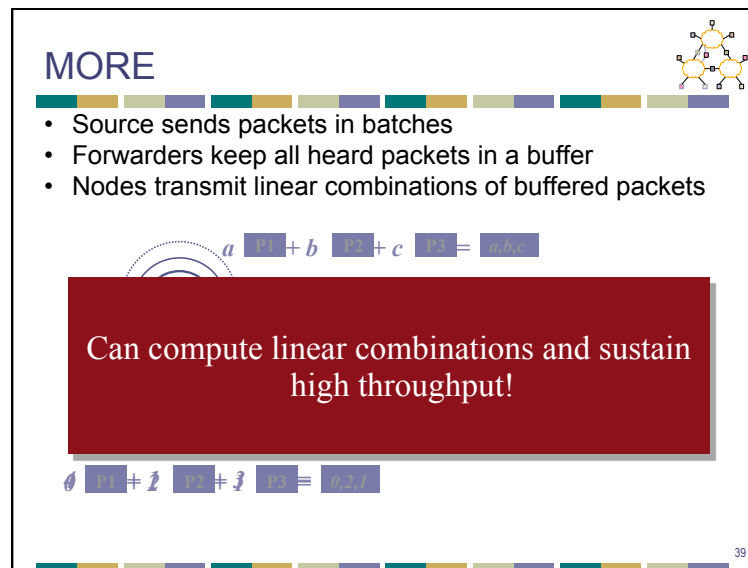
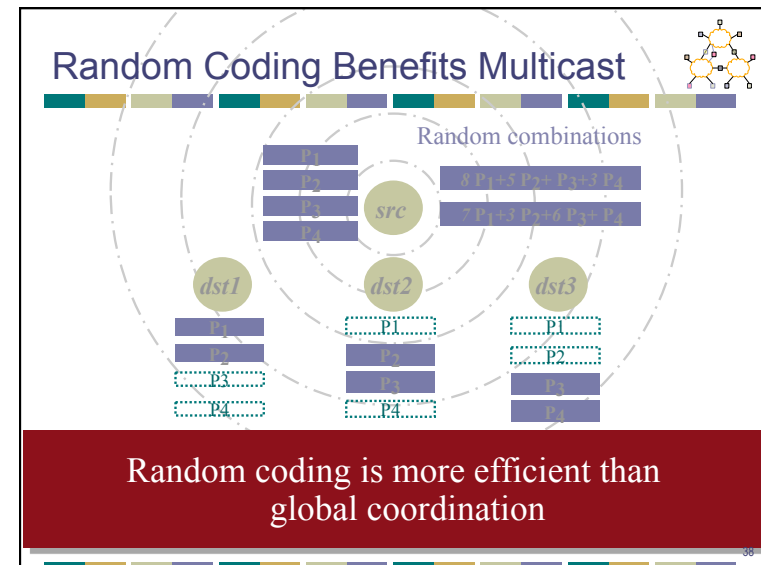
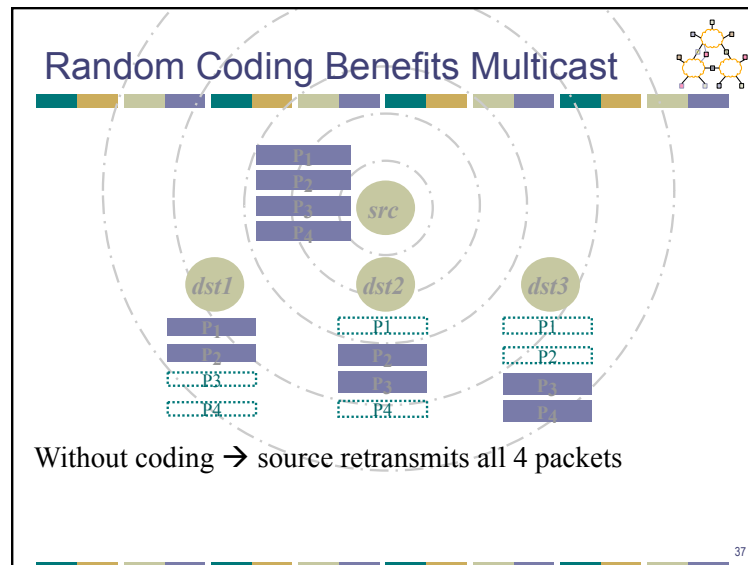
Each router forwards random combinations of packets



Randomness prevents duplicates

No scheduler; No coordination
Simple and exploits spatial reuse

36



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

$$1 \begin{bmatrix} p_1 + 3 & p_2 + 2 & p_3 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 2 \end{bmatrix}$$

$$5 \begin{bmatrix} p_1 + 4 & p_2 + 5 & p_3 \end{bmatrix} = \begin{bmatrix} 5 & 4 & 5 \end{bmatrix}$$

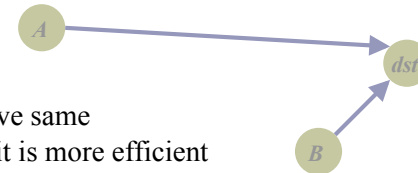
$$4 \begin{bmatrix} p_1 + 5 & p_2 + 5 & p_3 \end{bmatrix} = \begin{bmatrix} 4 & 5 & 5 \end{bmatrix}$$

- Destination acks batch, and source moves to next batch

41

But How Do We Get the Most Throughput?

- Naïve approach transmits whenever 802.11 allows

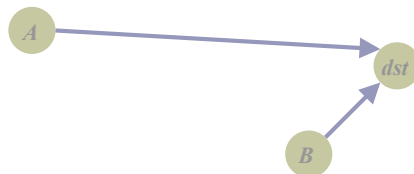


If A and B have same information, it is more efficient for B to send it

Need a Method to Our Madness

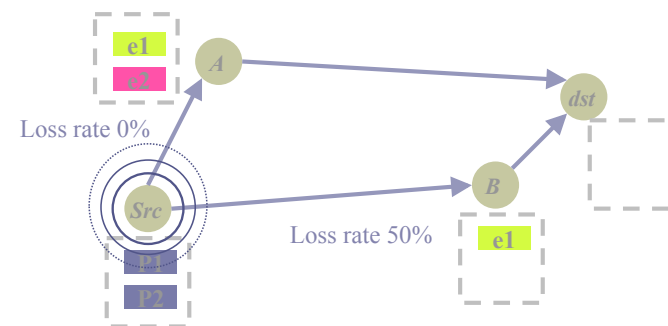
42

Probabilistic Forwarding



43

Probabilistic Forwarding

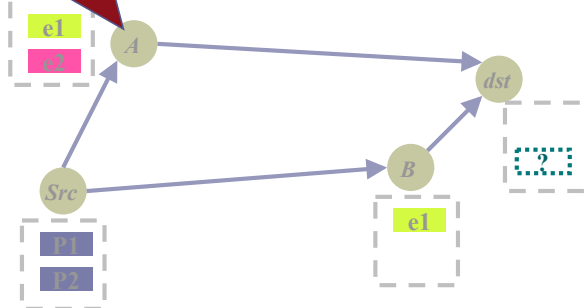


44

Probabilistic Forwarding

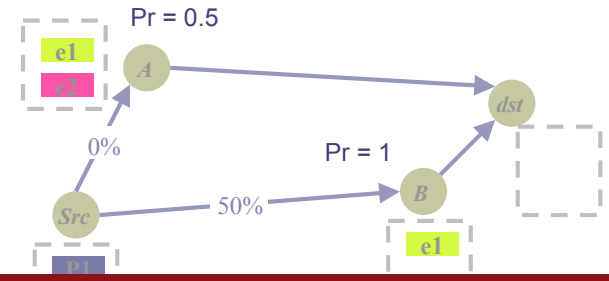
How many packets should I forward?

50% of buffer



45

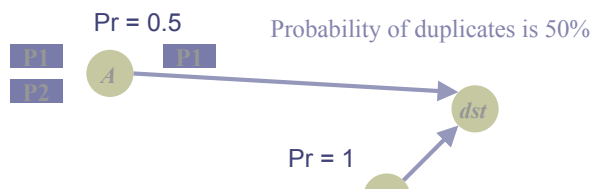
Probabilistic Forwarding



Compute forwarding probabilities without coordination using loss rates

46

Can ExOR Use Probabilistic Forwarding To Remove Coordination?



- Without random coding $\rightarrow$ need to know the exact packets to forward every time
- With random coding $\rightarrow$ need to know only the average amount of overlap

47

Adapting to Short-term Dynamics

- Need to balance sent information with received information
- MORE triggers transmission by receptions
- A node has a credit counter
 - Upon reception, increment the counter using forwarding probabilities
 - Upon transmission, decrement the counter
- Source stops $\rightarrow$ No triggers $\rightarrow$ Flow is done

48

Opportunistic Coding



- Three ways to get neighbor state
 - Reception report
 - Guess
 - Based on ETX metric (delivery probability)
 - Estimate the probability that packets are overheard
 - The neighbor is the previous hop of the packet

49

COPE Design



- Pseudo Broadcast
 - Cons of broadcast
 - Unreliable due to no ACK
 - Lack of backoff
 - Piggy back on unicast
 - Set one of intended node as Mac address
 - List all others in COPE header (between MAC and IP header)
 - Receiver: if it is on the list, decode the packet, else store the packet in its pool

50