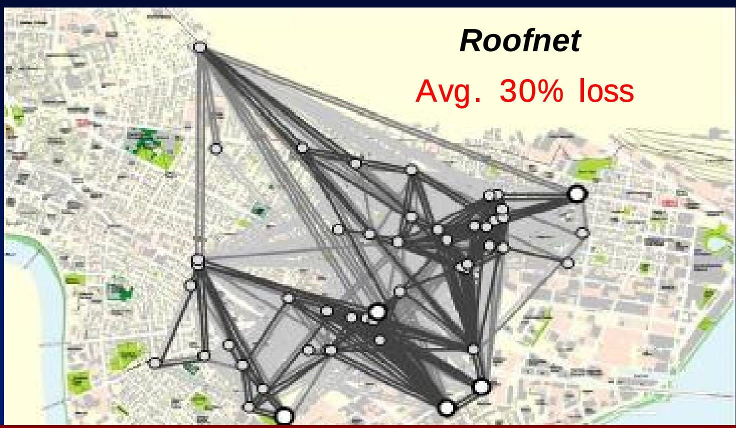
Trading Coordination For Randomness

Szymon Chachulski

Mike Jennings, Sachin Katti, and Dina Katabi

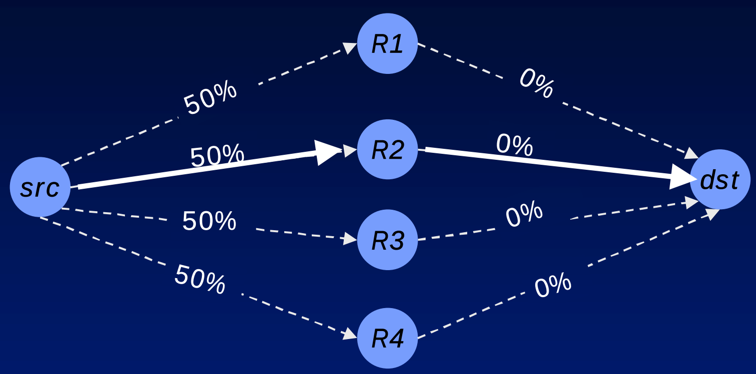
Wireless mesh networks have high loss rates



Objective:
High throughput despite lossy links

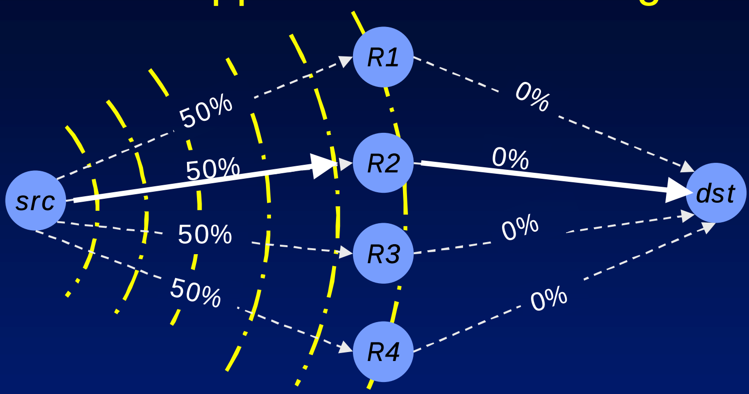
Use Opportunistic Routing

Use Opportunistic Routing



- Best single path à loss prob. 50%

Use Opportunistic Routing



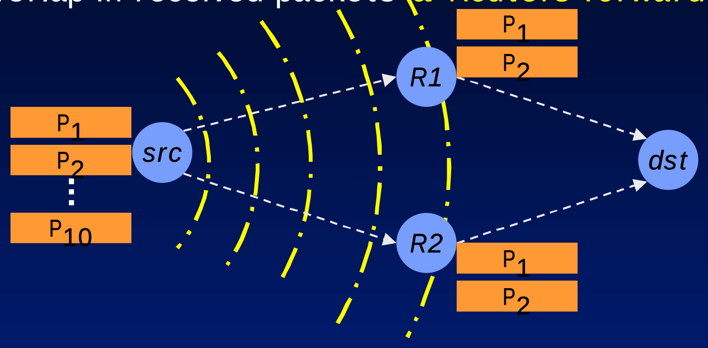
Opportunistic routing promises large increase in throughput

But

Overlap in received packets **à** Routers forward duplicates

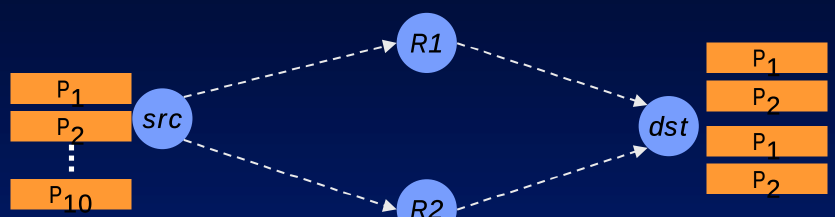
But

Overlap in received packets à Routers forward duplicates



But

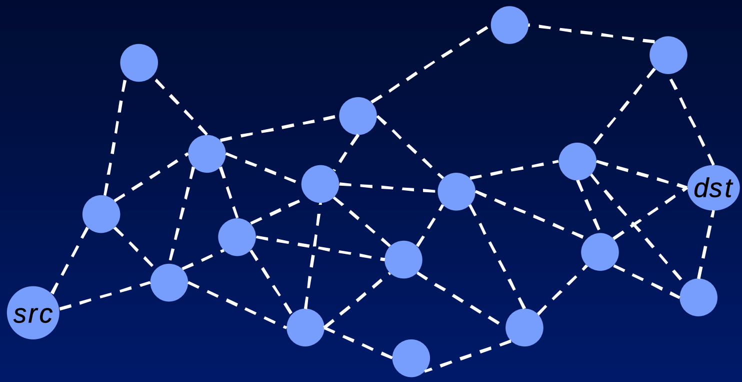
Overlap in received packets à Routers forward duplicates



State-of-the-art opp. routing, ExOR imposes a global scheduler:

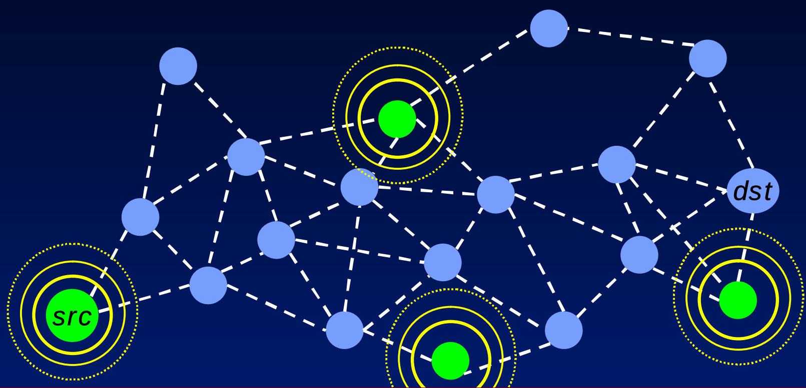
- Requires full coordination; every node must know who received what
- Only one node transmits at a time, others listen

Global Scheduling?



- Global coordination is too hard
- One transmitter

Global Scheduling?



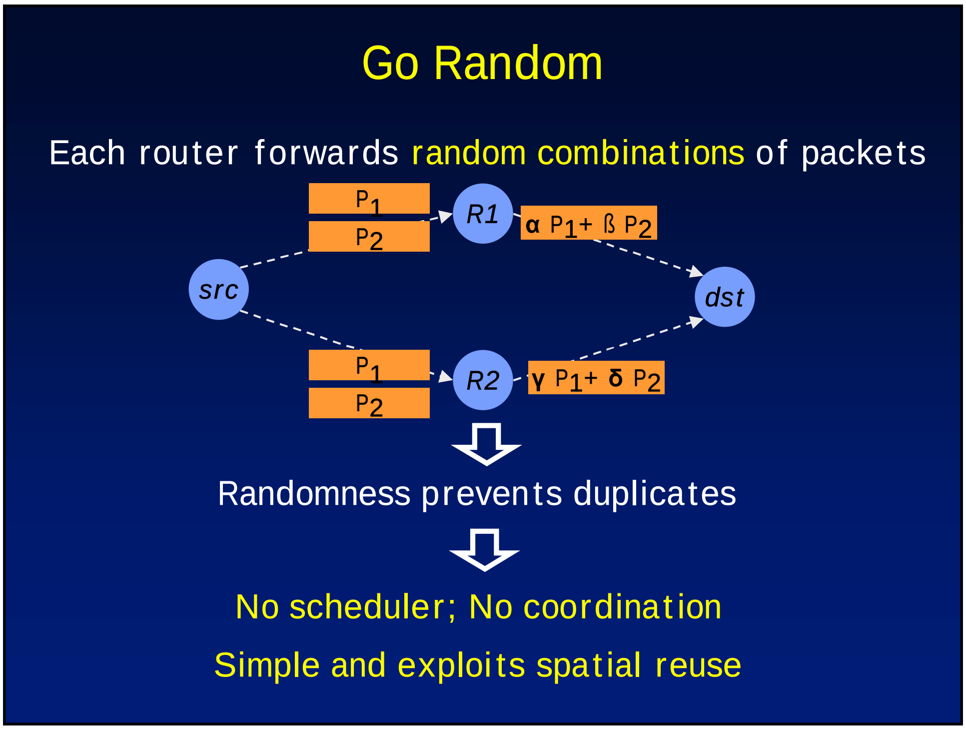
Does opportunistic routing have to be so complicated?

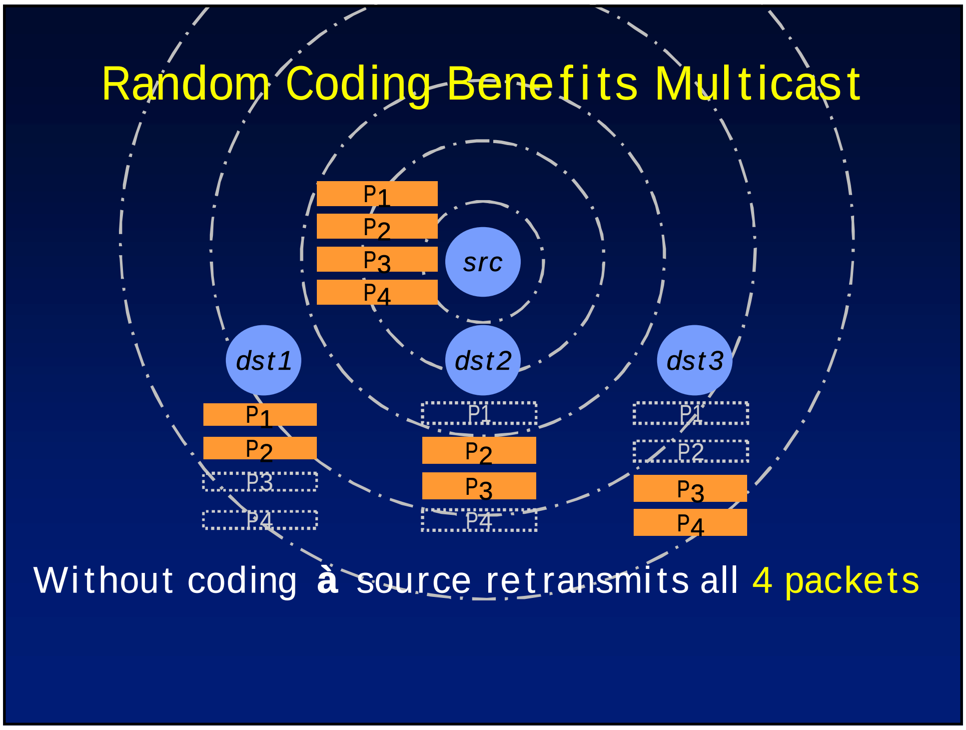
Our Contributions

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

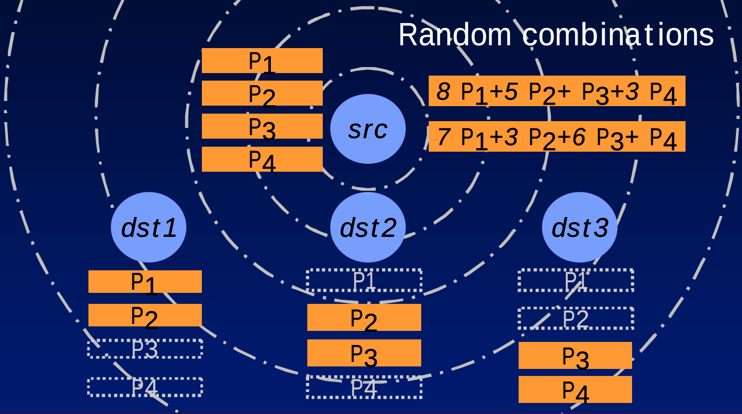
Go Random

Each router forwards random combinations of packets


Go Random
Each router forwards random combinations of packets
P_1
P_2
R1
$\alpha P_1 + ß P_2$
src
dst
P_1
P_2
R2
$\gamma P_1 + \delta P_2$
Randomness prevents duplicates
No scheduler; No coordination
Simple and exploits spatial reuse

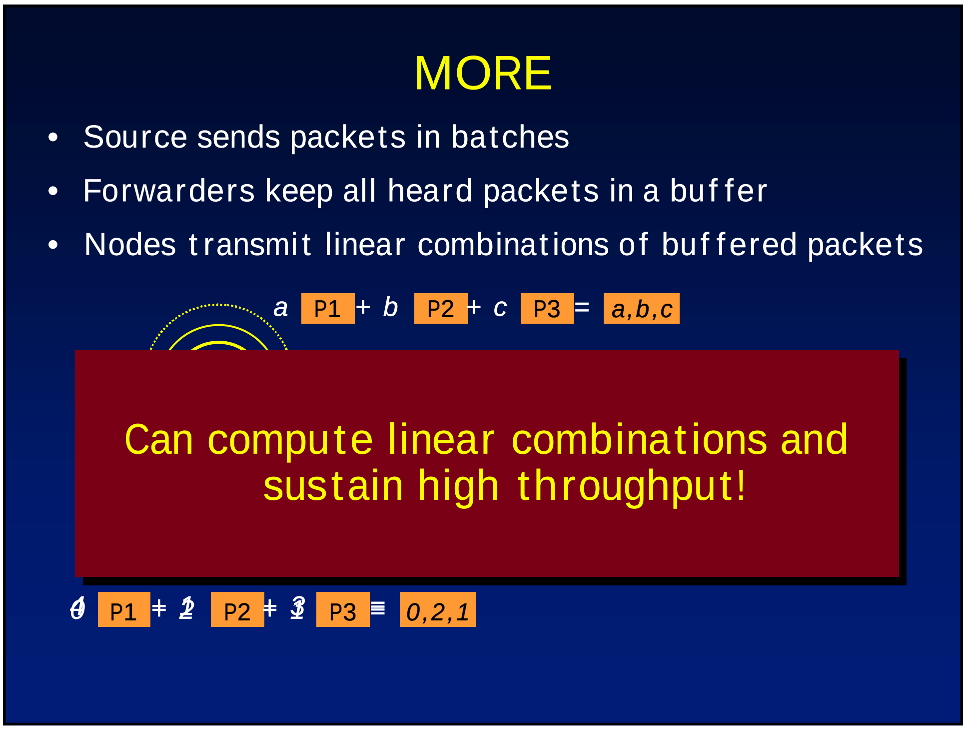

Random Coding Benefits Multicast
P_1
P_2
P_3
P_4
src
dst1
dst2
dst3
P_1
P_2
P3
P4
P1
P_2
P_3
P4
P1
P2
P_3
P_4
Without coding à source retransmits all 4 packets

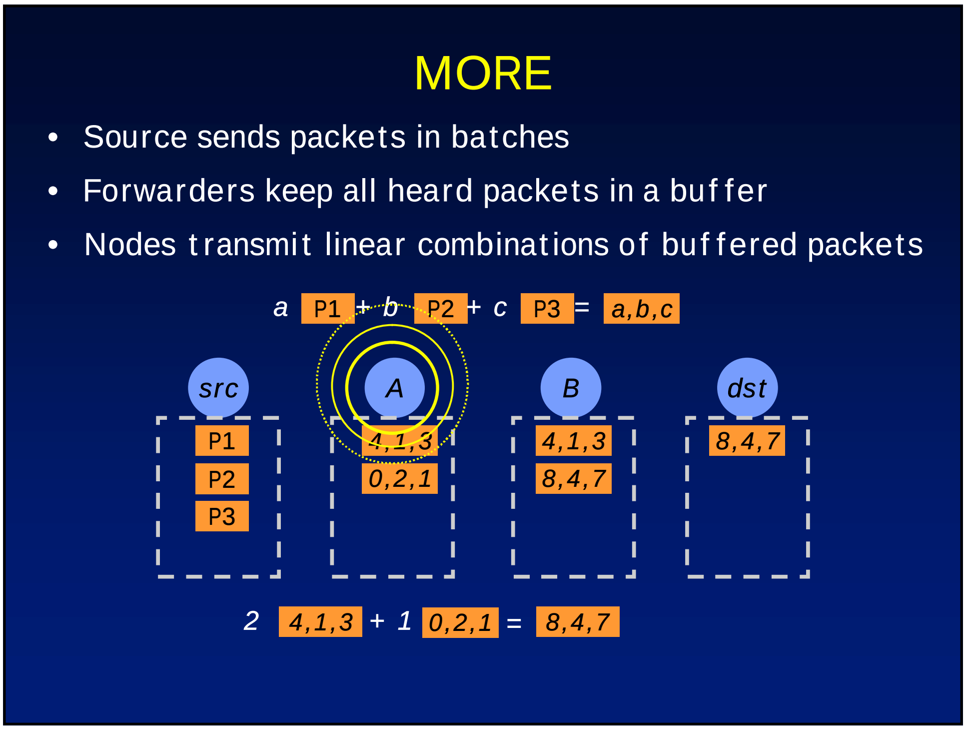
Random Coding Benefits Multicast



Random coding is more efficient than global coordination

MORE


MORE
• Source sends packets in batches
• Forwarders keep all heard packets in a buffer
• Nodes transmit linear combinations of buffered packets
a P1 + b P2 + c P3 = a,b,c
Can compute linear combinations and sustain high throughput!
P1 + P2 + P3 = 0,2,1


MORE
• Source sends packets in batches
• Forwarders keep all heard packets in a buffer
• Nodes transmit linear combinations of buffered packets
a P1 + b P2 + c P3 = a,b,c
src
A
B
dst
P1
P2
P3
4,1,3
0,2,1
4,1,3
8,4,7
8,4,7
2 4,1,3 + 1 0,2,1 = 8,4,7

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 P1 + *3* P2 + *2* P3 = *1,3,2*

5 P1 + *4* P2 + *5* P3 = *5,4,5*

4 P1 + *5* P2 + *5* P3 = *4,5,5*

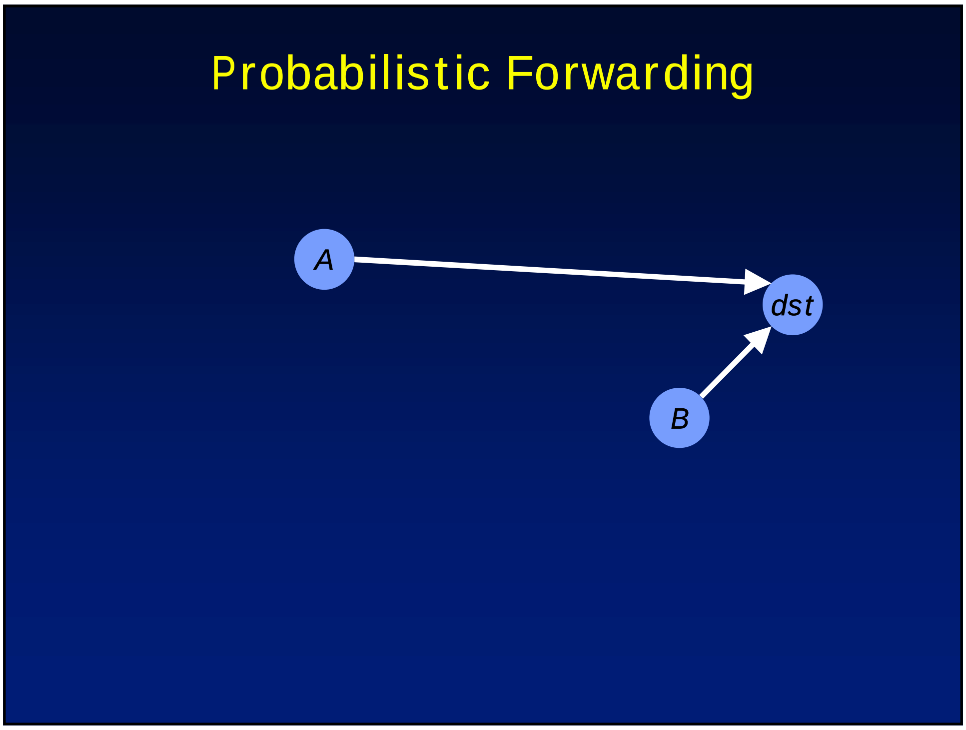
- Destination acks batch, and source moves to next batch

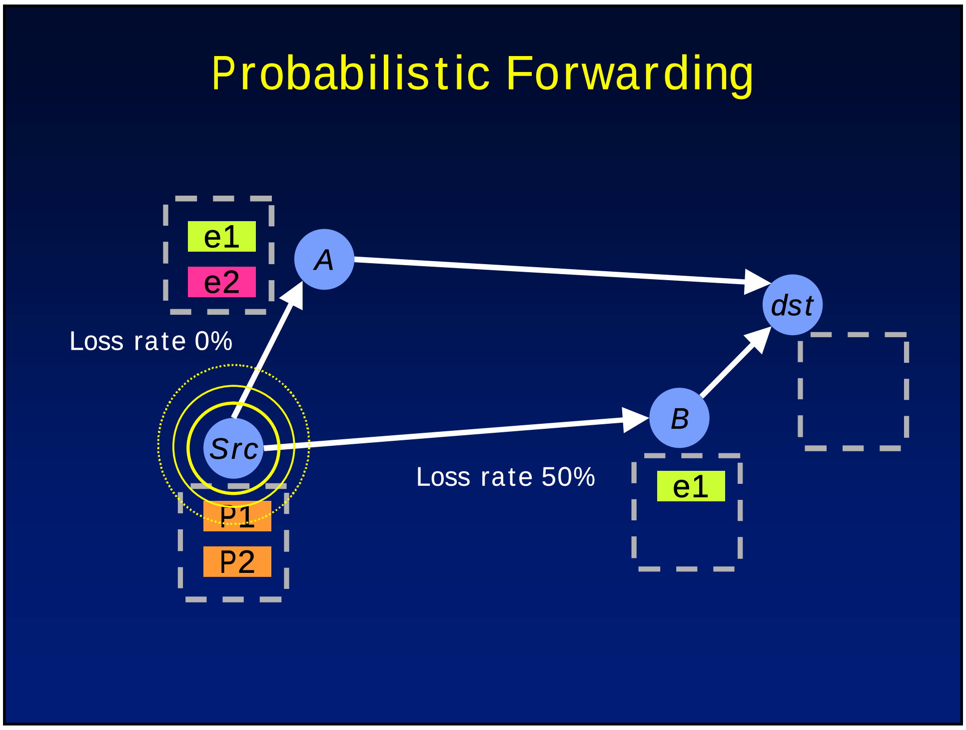
But How Do We Get the Most Throughput?

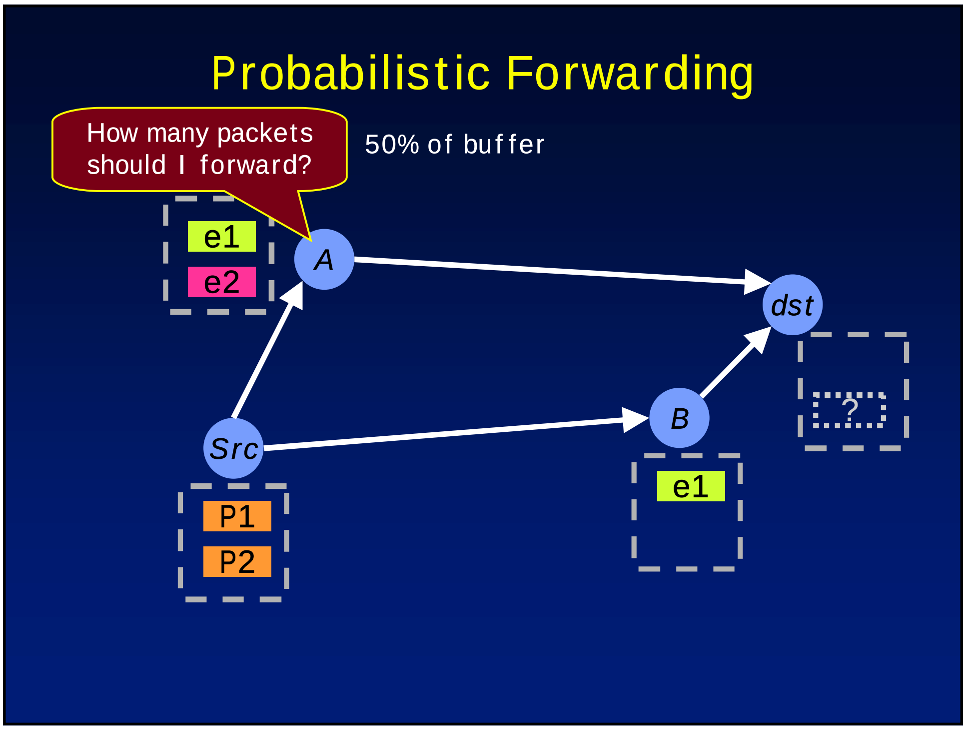
- Naïve approach transmits whenever 802.11 allows

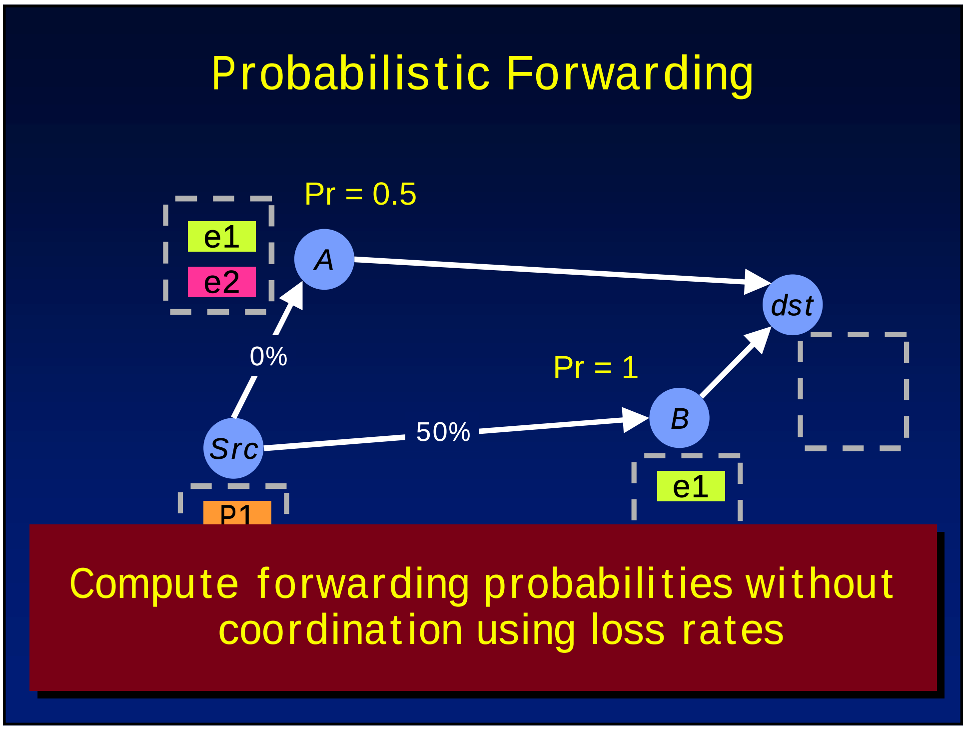
If *A* and *B* have same information, it is more efficient for B to send it

Need a Method to Our Madness

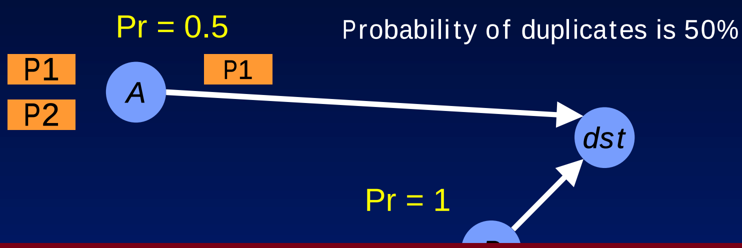

Probabilistic Forwarding
A
dst
B


Probabilistic Forwarding
e1
e2
A
dst
Loss rate 0%
Src
B
Loss rate 50%
e1
P1
P2


Probabilistic Forwarding
How many packets should I forward?
50% of buffer
e1
e2
A
dst
?
B
Src
e1
P1
P2


Probabilistic Forwarding
Pr = 0.5
e1
e2
A
dst
0%
Pr = 1
B
Src
50%
e1
P1
Compute forwarding probabilities without coordination using loss rates

Can ExOR Use Probabilistic Forwarding To Remove Coordination?



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

Long-term averages are great, but…
Wireless is unpredictable over short time-scales

> There are known knowns. These are things we know that we know. There are known unknowns. That is to say, there are things that we know we don't know. But there are also unknown unknowns.

MORE needs to adapt to short-term dynamics

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 à No triggers à Flow is done

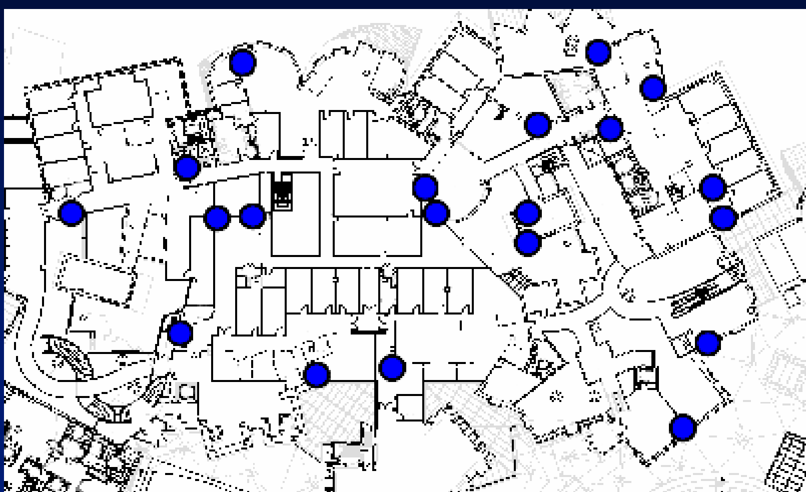
Performance

Experimental Setup

- We implemented MORE in Linux
- 20-node testbed
- Compare MORE with:
  - Current Routing (Single Best Path)
  - ExOR (State-of-the-art Opportunitic Routing)
- Experiment
  - Random source-destination pairs
  - Transmit 5 MB file

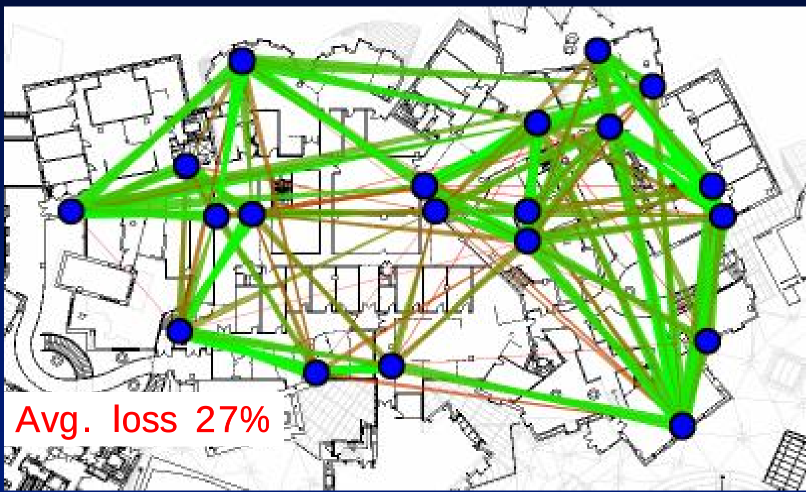
Testbed

- 20-node testbed over three floors

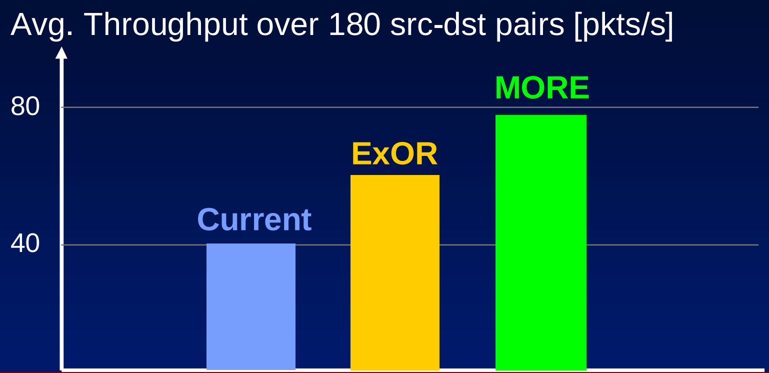


Testbed

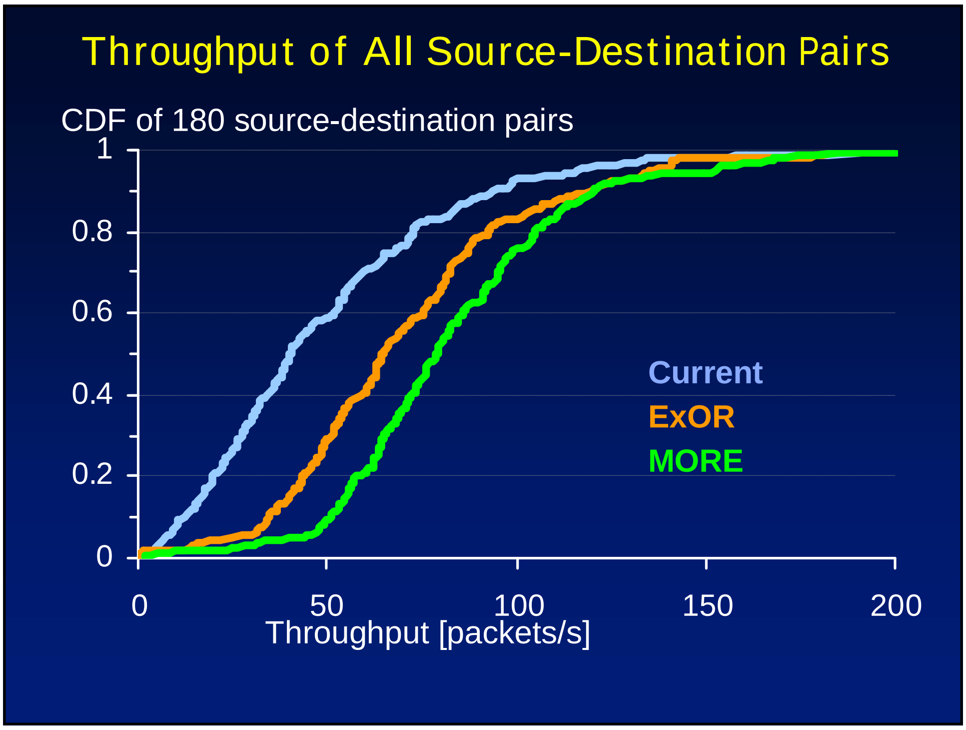
- 20-node testbed over three floors

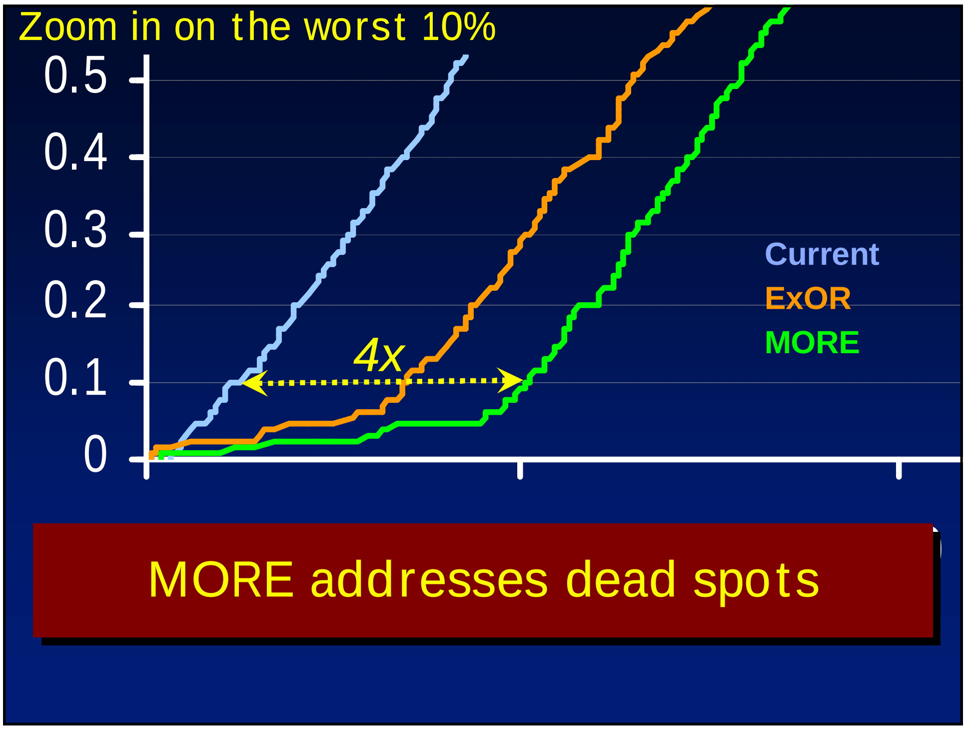


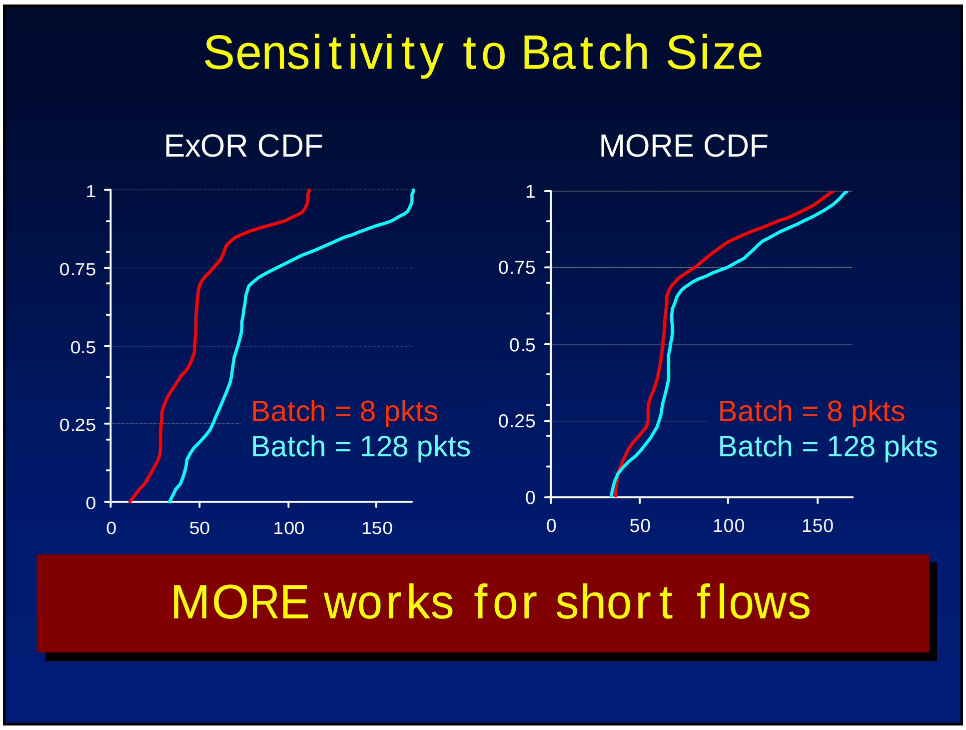
Does MORE Improve Wireless Throughput?

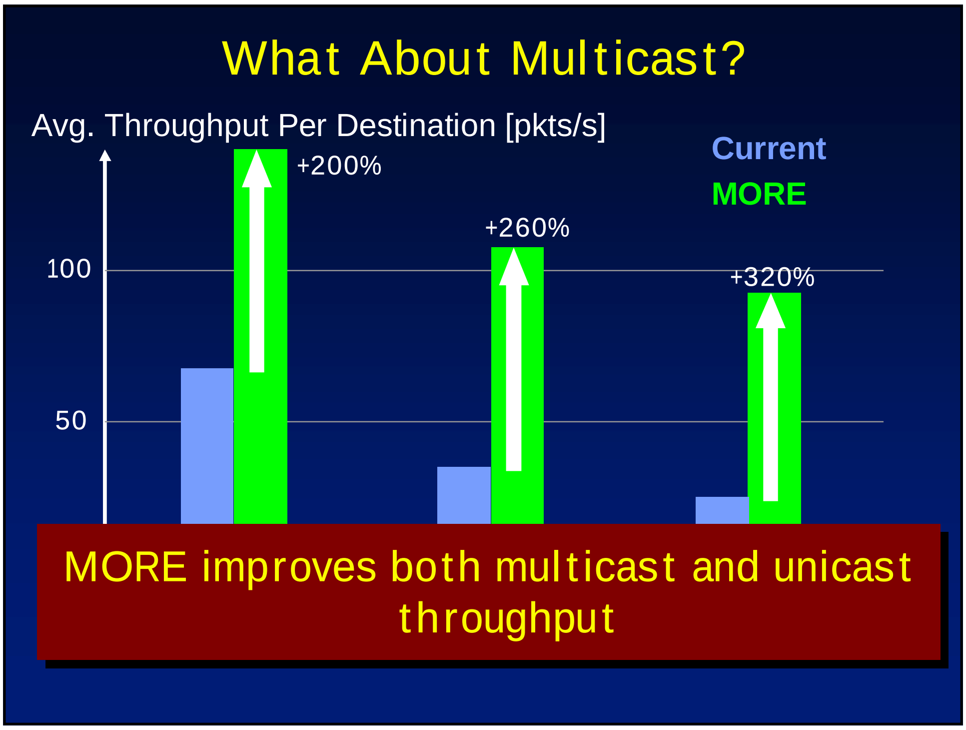


MORE's throughput is
- 2x better than current routing
- 22% better than ExOR


Throughput of All Source-Destination Pairs
CDF of 180 source-destination pairs
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
Throughput [packets/s]
Current
ExOR
MORE


Zoom in on the worst 10%
0.5
0.4
0.3
0.2
0.1
0
4x
Current
ExOR
MORE
MORE addresses dead spots


Sensitivity to Batch Size
ExOR CDF
MORE CDF
1
0.75
0.5
0.25
0
0
50
100
150
Batch = 8 pkts
Batch = 128 pkts
MORE works for short flows


What About Multicast?
Avg. Throughput Per Destination [pkts/s]
Current
MORE
+200%
+260%
+320%
100
50
MORE improves both multicast and unicast throughput

MORE for Less!

- Lesser coordination and lesser rigidity
  - No scheduler
- More flexibility
  - Works on top of 802.11 à enjoy spatial reuse
  - One framework for unicast and multicast
- More throughput
  - 22% better than ExOR
  - 2x better than current routing