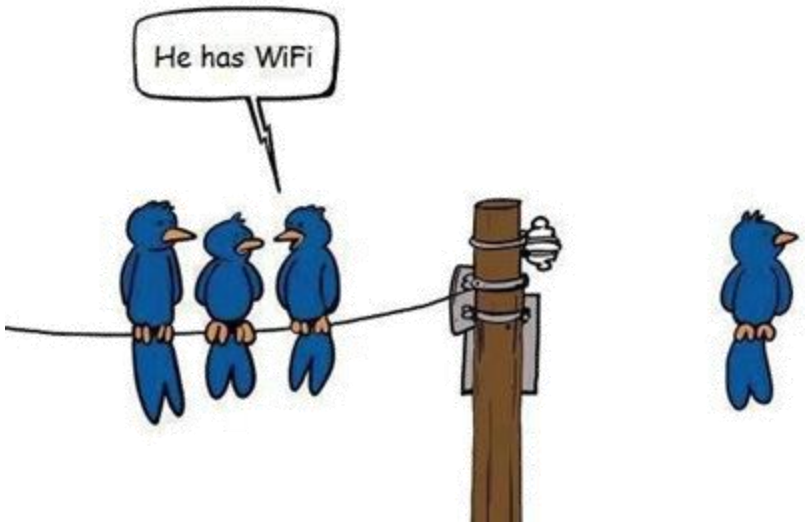
COM-405: Mobile Networks

Lecture 10.0: Multi-Hop Networks

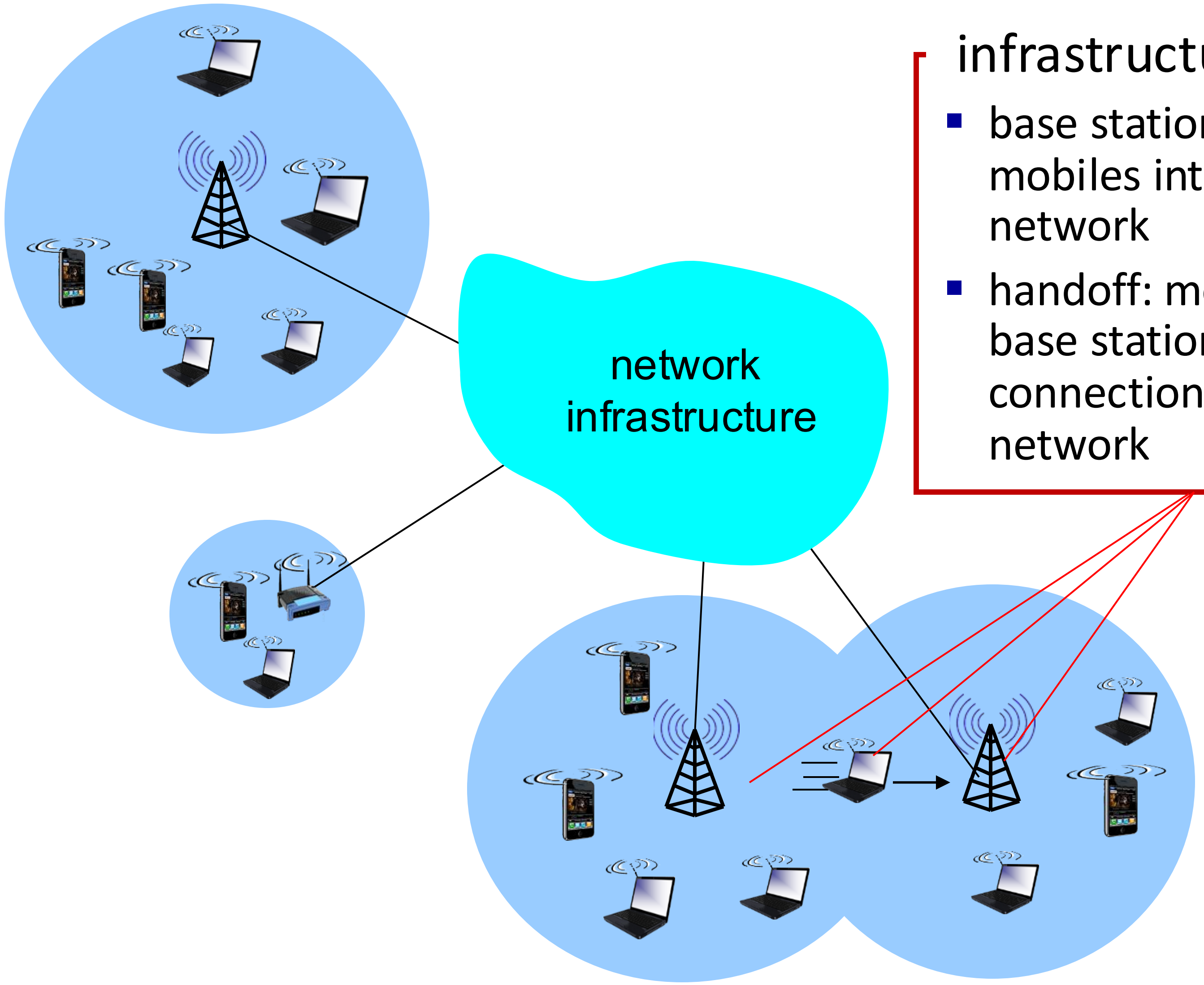
Haitham Hassanieh



EPFL

Laboratory of SEnsing
& Networking Systems

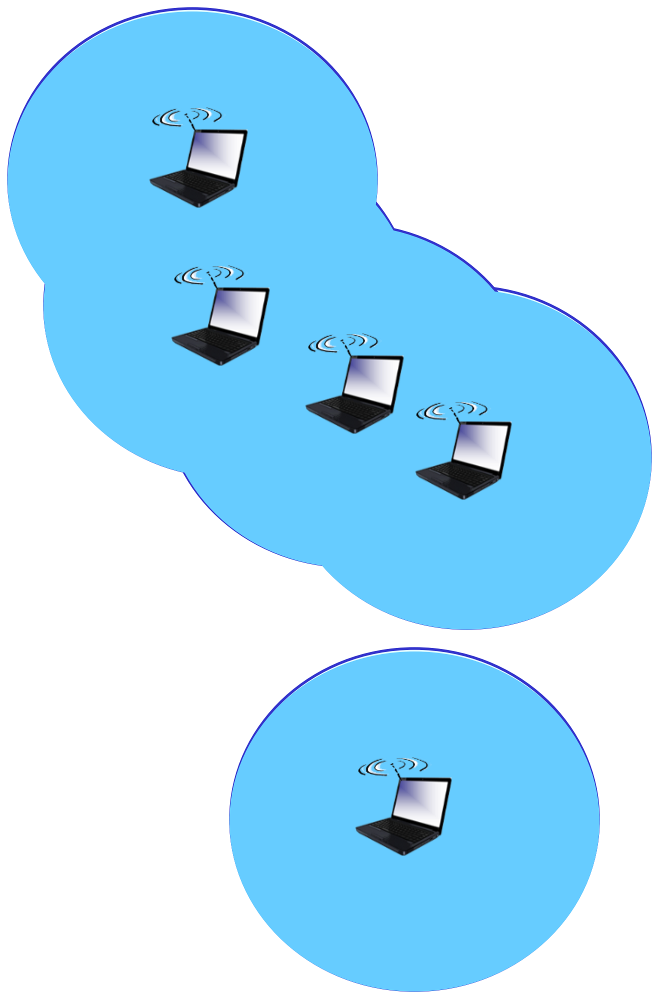
Elements of a wireless network



infrastructure mode

- base station connects mobiles into wired network
- handoff: mobile changes base station providing connection into wired network

Elements of a wireless network



ad hoc mode

- no base stations
- nodes can only transmit to other nodes within link coverage
- nodes organize themselves into a network: route among themselves

Wireless network taxonomy

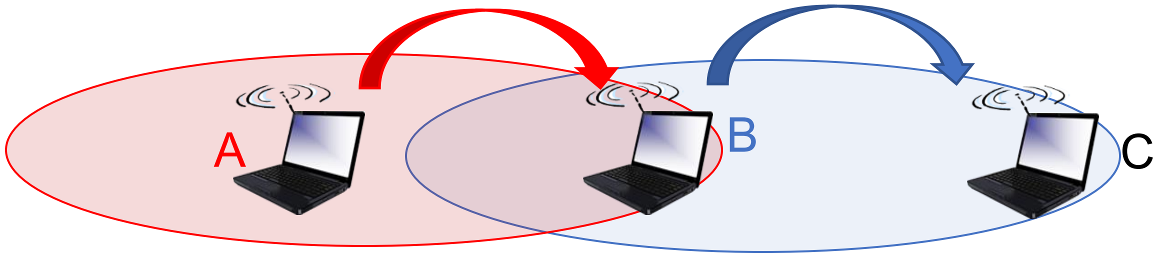
	single hop	multiple hops
infrastructure (e.g., APs)	host connects to base station (WiFi, WiMAX, cellular) which connects to larger Internet	host may have to relay through several wireless nodes to connect to larger Internet: *mesh net*
no infrastructure	no base station, no connection to larger Internet (Bluetooth, ad hoc nets)	no base station, no connection to larger Internet. May have to relay to reach other a given wireless node MANET, VANET

Wireless network characteristics

Problem: A wants to transmit a packet to C

Option 1: A increases its power such that its packet reaches C

Option 2: A sends that packet to B which intern send it to C

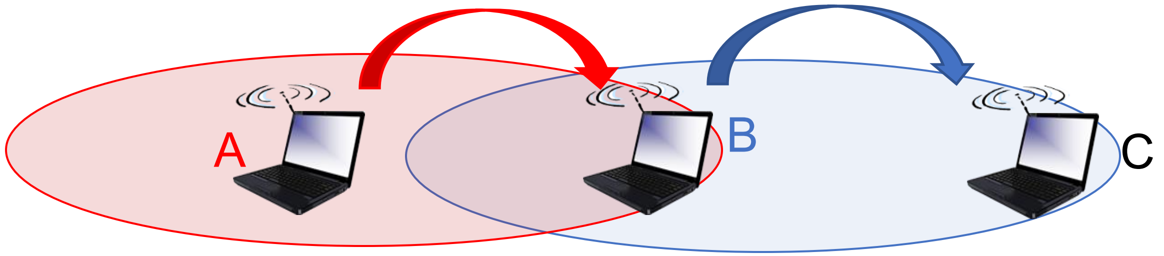


Wireless network characteristics

Problem: A wants to transmit a packet to C

To double transmission range, we need: 4x more overall power!

To transmit over two hops, we need: 2x more overall power!



Wireless network characteristics

Multi-hop wireless networks:

- Increase TX power: increase transmission range by N times, need $N^2 \times$ more power

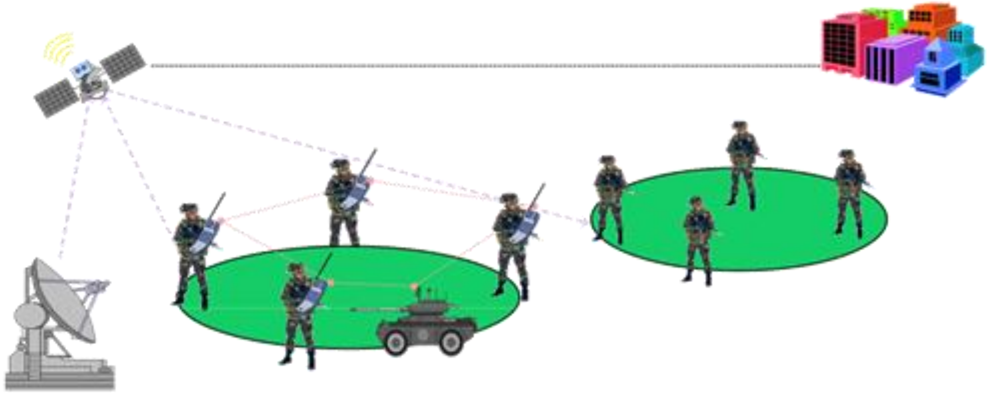
- Multi-hop links: increase transmission range by N times, need $N \times$ more power

Multi-hop wireless networks!

Multi-hop Wireless Networks

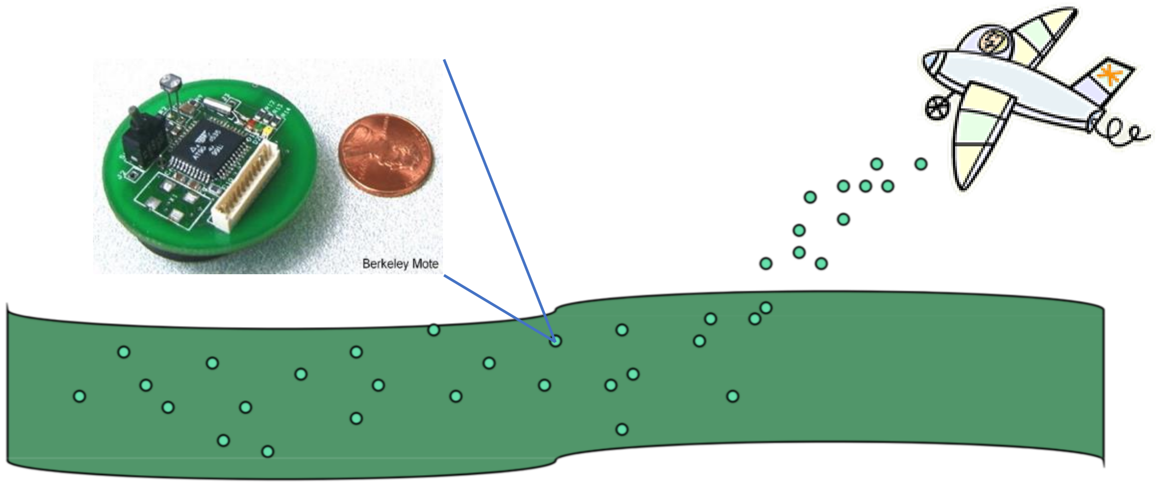
Wireless Ad Hoc Networks (WANETs)

A wireless Ad hoc network consists of a collection of mobile hosts dynamically forming a temporary network without the use of an existing network infrastructure.

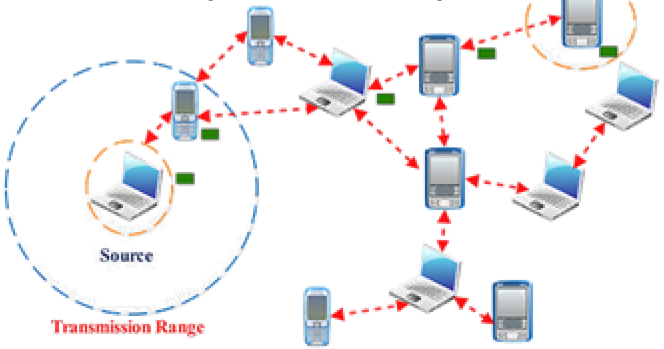


Wireless Sensor Networks (WSN)

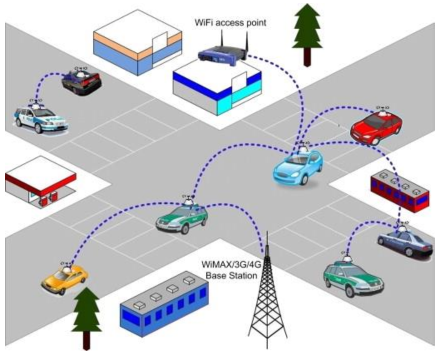
A sensor network consists of one or more base stations and a large number of inexpensive sensors.

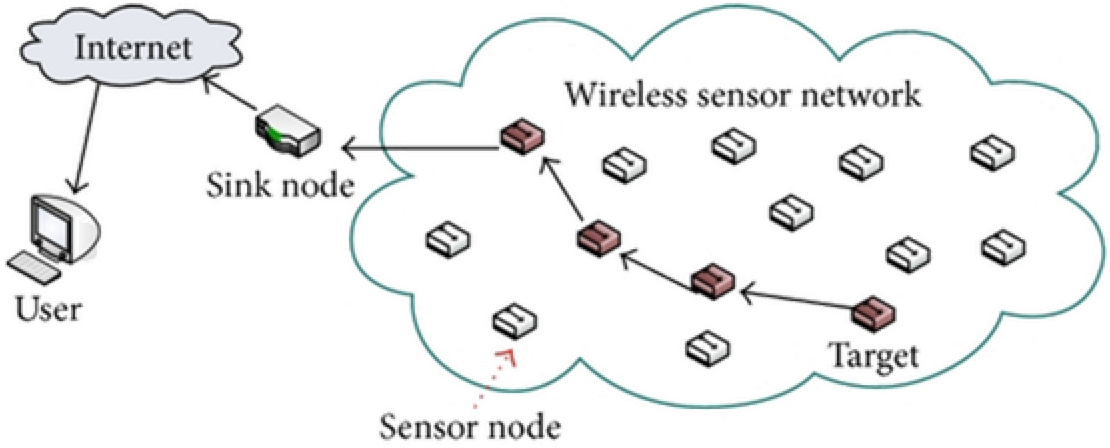


Mobile Ad Hoc Networks (MANETs)

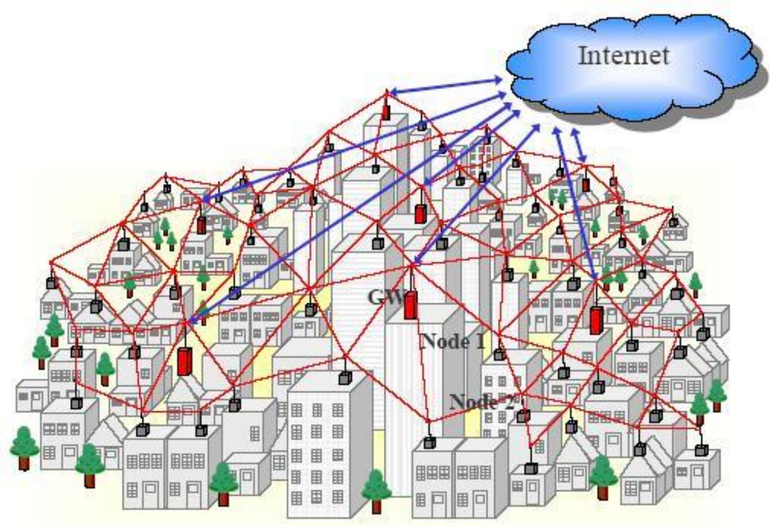


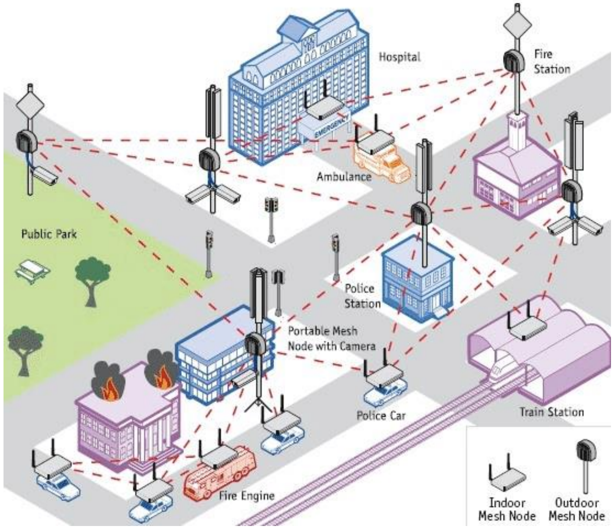
Vehicular Ad Hoc Networks (VANETs)





Wireless Mesh Networks

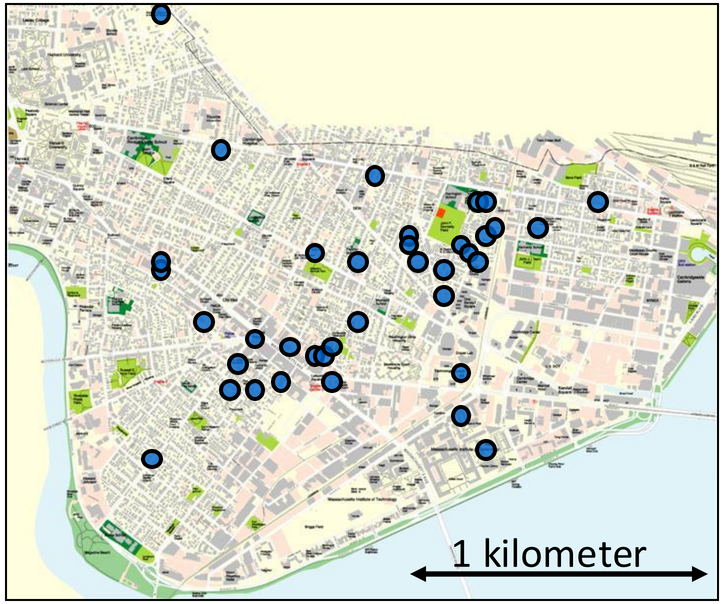




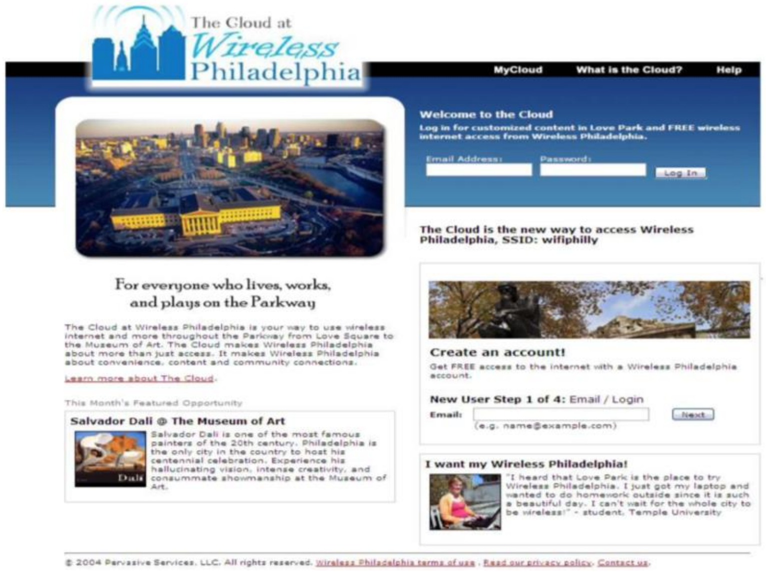
- Very low installation and maintenance cost
- Easy to provide coverage outdoors
- Ubiquitous access.
- Rapid deployment.

Wireless Mesh Networks

Roofnet

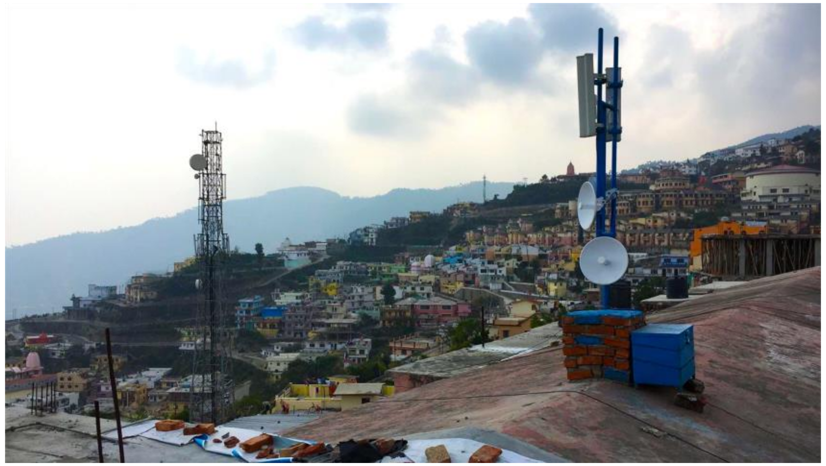


Wireless Philly



- Dense 802.11-based multi-hop network
- Goal is high-throughput in the presence of lossy links

Facebook WiFi Mesh Network



TARF Rice Mesh Network

Traditional Single Path Routing

Represent the wireless network as a graph

- Two nodes have an edge if they can communicate (i.e., are within radio range)
- Each edge is labeled with a weight (where a smaller weight indicates a preferred edge)

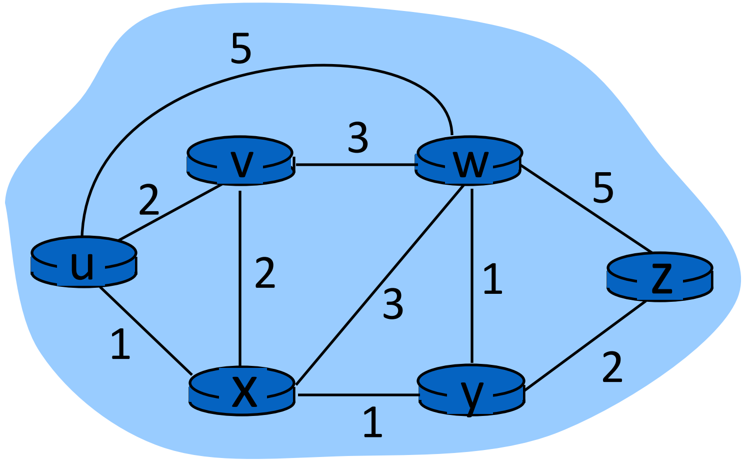
Run shortest path algorithm on the graph (e.g., Dijkstra, Bellman Ford)

- Produce the minimum weight path between every pair of nodes

How do you pick the edge weights?

- i.e., what metric should shortest path minimize?

Graph abstraction of the network



graph: G = (N,E)

N = set of routers = { u, v, w, x, y, z }

E = set of links ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

c(x,x') = cost of link (x,x') e.g., c(w,z) = 5

cost could always be 1, or inversely related to bandwidth, or related to congestion

cost of path $(x_1, x_2, x_3,\ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

key question: what is the least-cost path between u and z ?
routing algorithm: algorithm that finds that least cost path

Link-state Routing

Dijkstra's algorithm

- net topology, link costs known to all nodes
- computes least cost paths from one node ("source") to all other nodes
  - gives *forwarding table* for that node
- iterative: after k iterations, know least cost path to k dest.'s

1. Find w not in set S such that $D_x(w)$ is a minimum
2. Add w to S
3. Update $D_x(v)$ for all v adjacent to w and not in S

   $\boldsymbol{D_x(v)} = \min(\boldsymbol{D_x(v)}, \boldsymbol{D_x(w)} + \boldsymbol{c(w,v)})$
4. Loop to 1.

Distance Vector Routing

Bellman-Ford (Dynamic Programming)

let $D_x(y)$:= cost of least-cost path from x to y

then

$$D_x(y) = \min_v \{c(x,v) + D_v(y)\}$$

x maintains distance vector $\boldsymbol{D_x} = [Dx(y)\colon y \in N]$
x knows cost to each neighbor v: $c(x,v)$
x maintains its neighbors' distance vectors

1. From time-to-time, each node sends its own distance vector estimate to neighbors
2. When x receives new DV estimate from neighbor, it updates its own DV using B-F equation:

Cannot use classical wired routing in wireless!

Channel is Broadcast

- Interference: Cannot transmit on all links in parallel.
- Opportunity!

Unreliable

- Packet loss due to collision
- Packet loss due to bit errors

Overhead

- Nodes not as powerful (e.g. Sensor networks).
- Link bandwidth is typically lower than wired.

Routing in Wireless Ad-Hoc Networks

Dynamic Source Routing (DSR)

- On demand routing
  - → No periodic updates
- When a source wants to route packets, routes generated
- Consists of two parts
  - Route discovery
  - Route maintenance

Ad hoc On-demand distance vector (AODV)

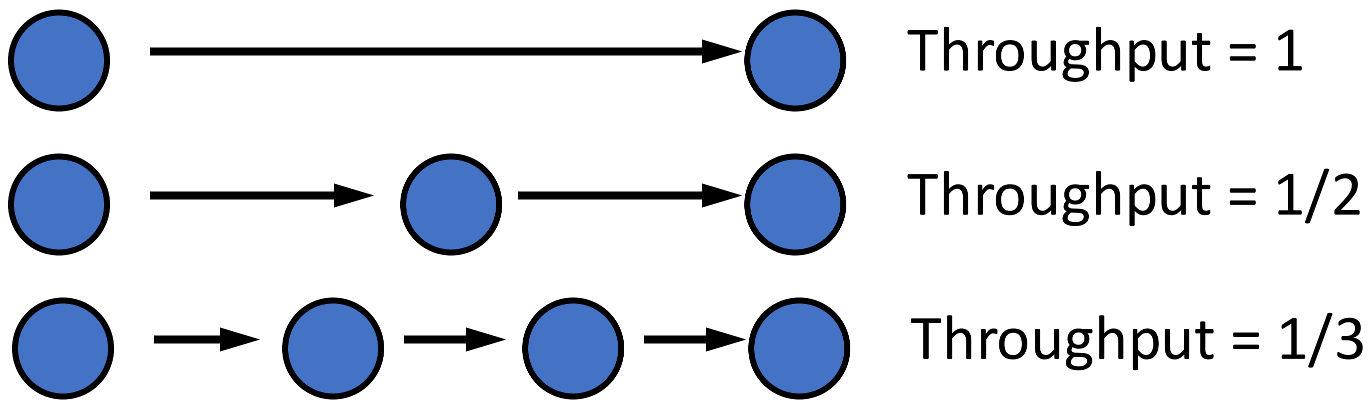
- Best of both worlds of DSR and DV
- Like DV it is next hop based routing, full route path not included in packet
- Like DSR, does not need periodic route maintenance messages

Challenge: How to select the link cost?

A straw-man route metric (1):
Assign all edges the same weight → Minimize number of hops

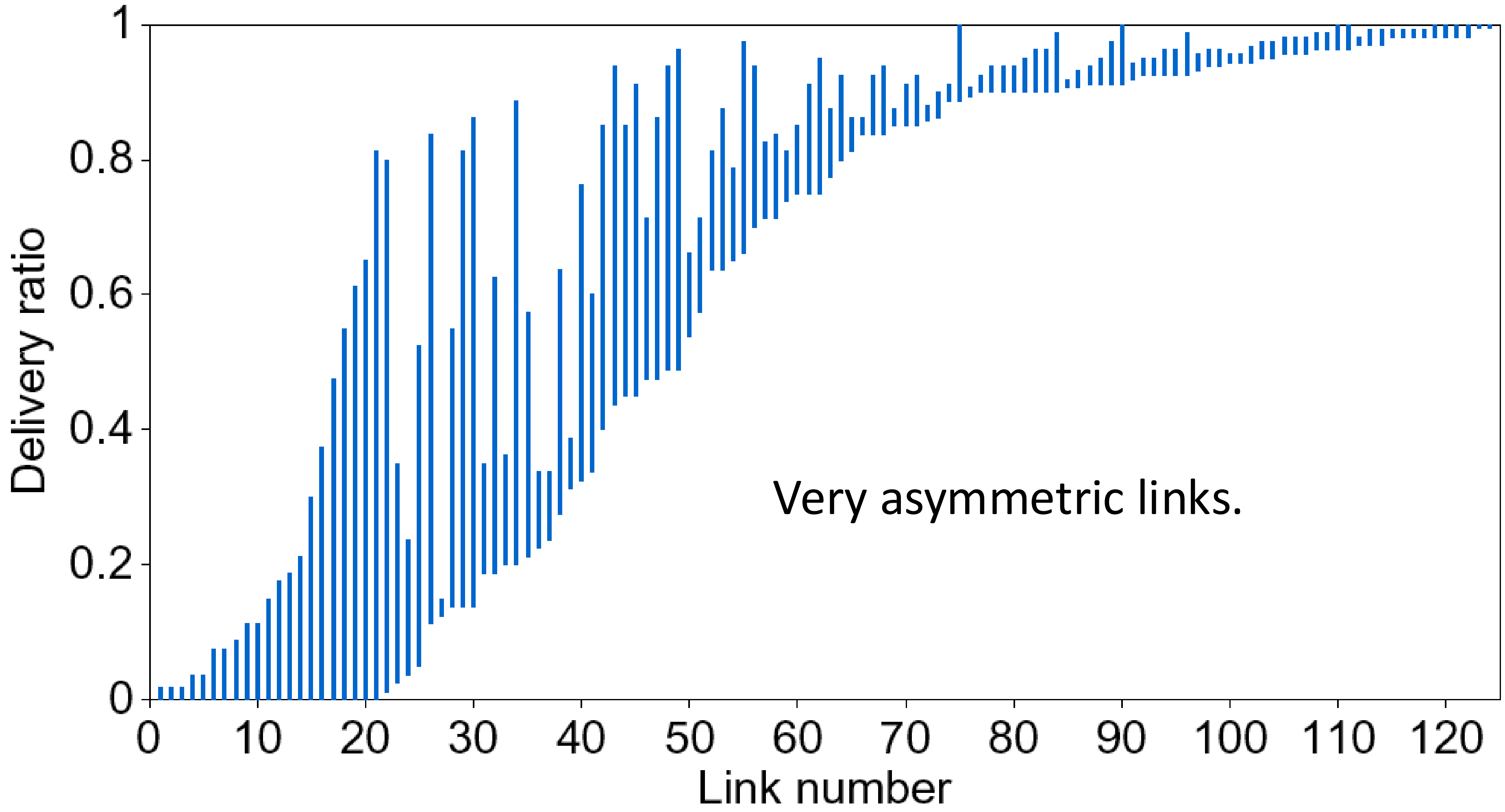
Reasoning:

- Links in route share radio spectrum
- Extra hops reduce throughput



But is not good enough because different edges many have very different packet loss rates

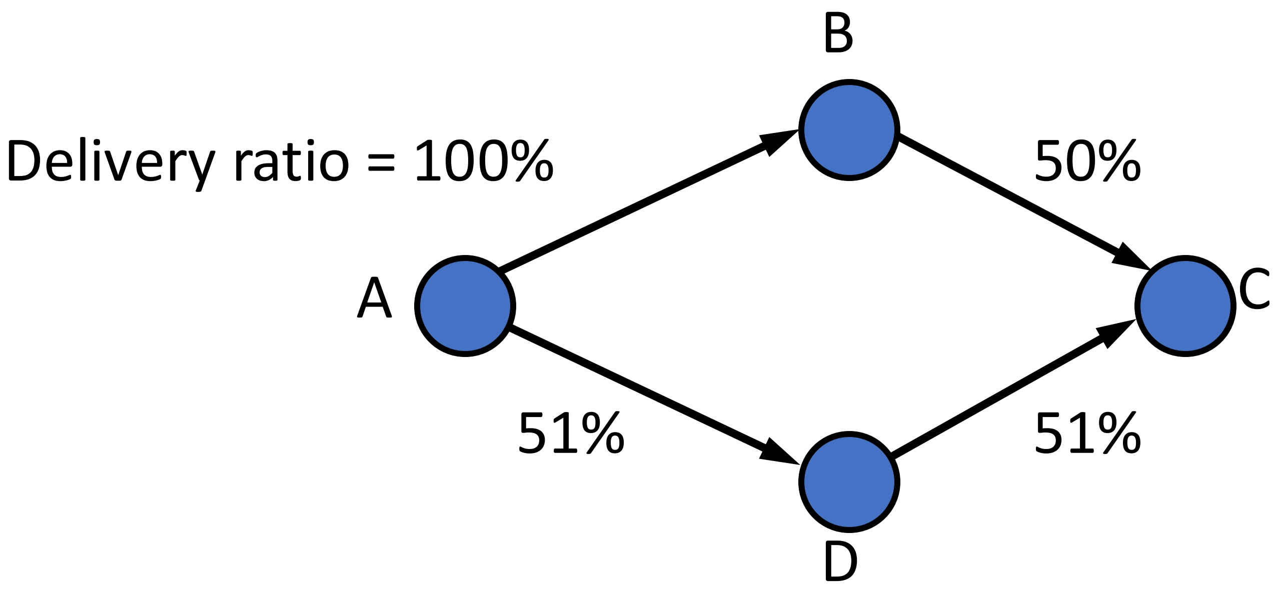
Challenge: links are lossy and asymmetric



Different links have different loss rates
Further, the loss rate may be different in each direction

A straw-man route metric (2):

Maximize bottleneck throughput

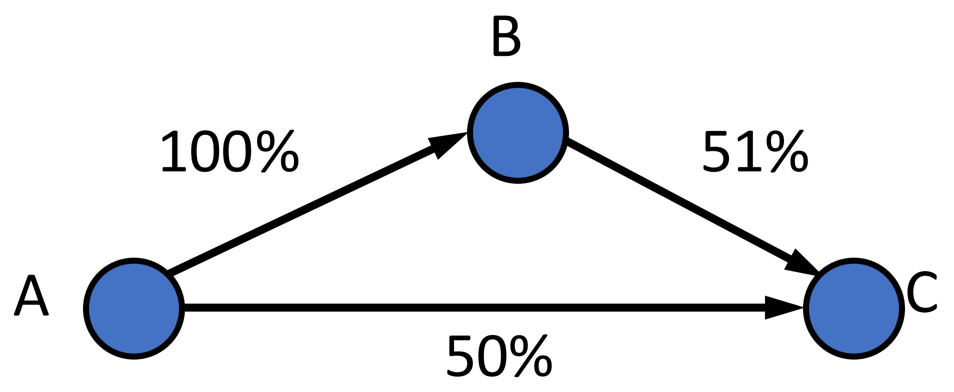


Bottleneck throughput:
- A-B-C = 50%
- A-D-C = 51%

Actual throughput:
- A-B-C : ABBABBABB = 33%
- A-D-C : AADDAADD = 25%

Key Idea: In a shared medium links are not independent

A straw-man metric (3):
Maximize end-to-end delivery ratio



End-to-end delivery ratio:
- A-B-C = <u>51%</u>
- A-C = 50%

Actual throughput:
- A-B-C : ABBABBABB = 33%
- A-C : AAAAAAAA = <u>50%</u>

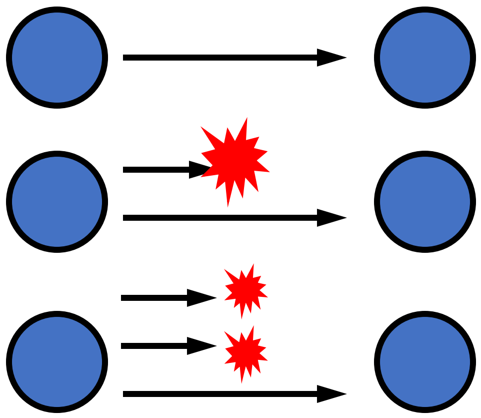
Key Idea: Again, links are not independent

Wireless routing metric: ETX

Minimize total transmissions per packet
(ETX, ‘Expected Transmission Count’)

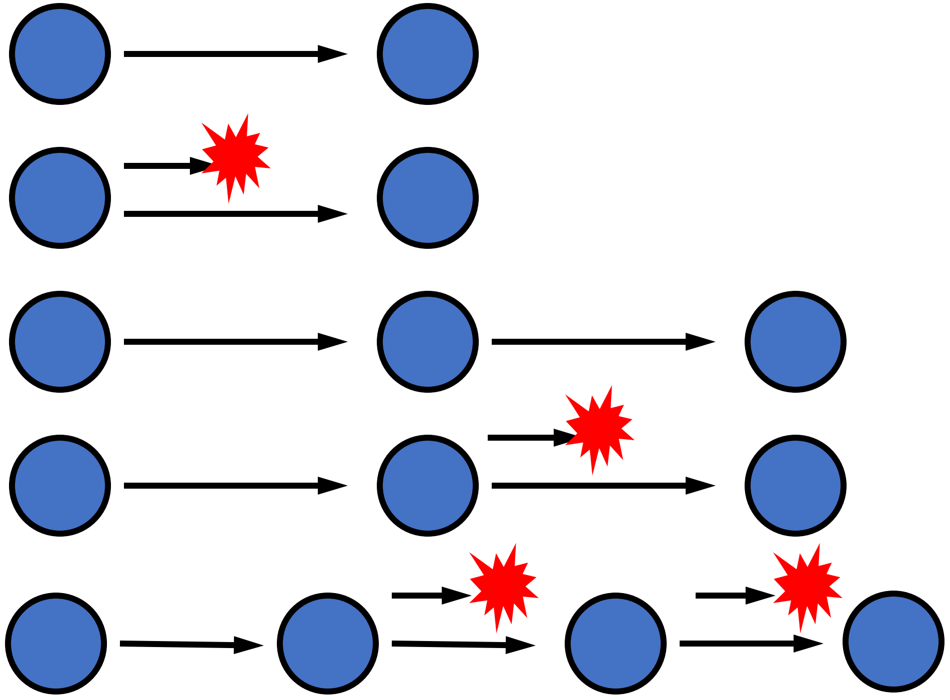
Link throughput ≈ 1/ Link ETX

Delivery Ratio	Link ETX	Throughput
100%	1	100%
50%	2	50%
33%	3	33%



Route ETX

Route ETX = Sum of link ETXs



Route ETX	Throughput
1	100%
2	50%
2	50%
3	33%
5	20%

Calculating Link ETX

Assuming 802.11 link-layer acknowledgments (ACKs) and retransmissions:

$$P(TX\ success) = P(Data\ success) \times P(ACK\ success)$$

$$Link\ ETX = \frac{1}{P(TX\ success)} = \frac{1}{P(Data\ success) \times P(ACK\ success)}$$

- Estimating link ETX

  P(Data success) ≈ measured fwd delivery ratio r_{fwd}

  P(ACK success) ≈ measured rev delivery ratio r_{rev}

  Link ETX $\approx \frac{1}{r_{fwd} \times r_{rev}}$

Measuring delivery ratios

- Each node broadcasts small link probes once per second
- Nodes remember probes received over past 10 seconds
- Reverse delivery ratios estimated as

$$r_{\text{rev}} \approx \text{pkts received / pkts sent}$$

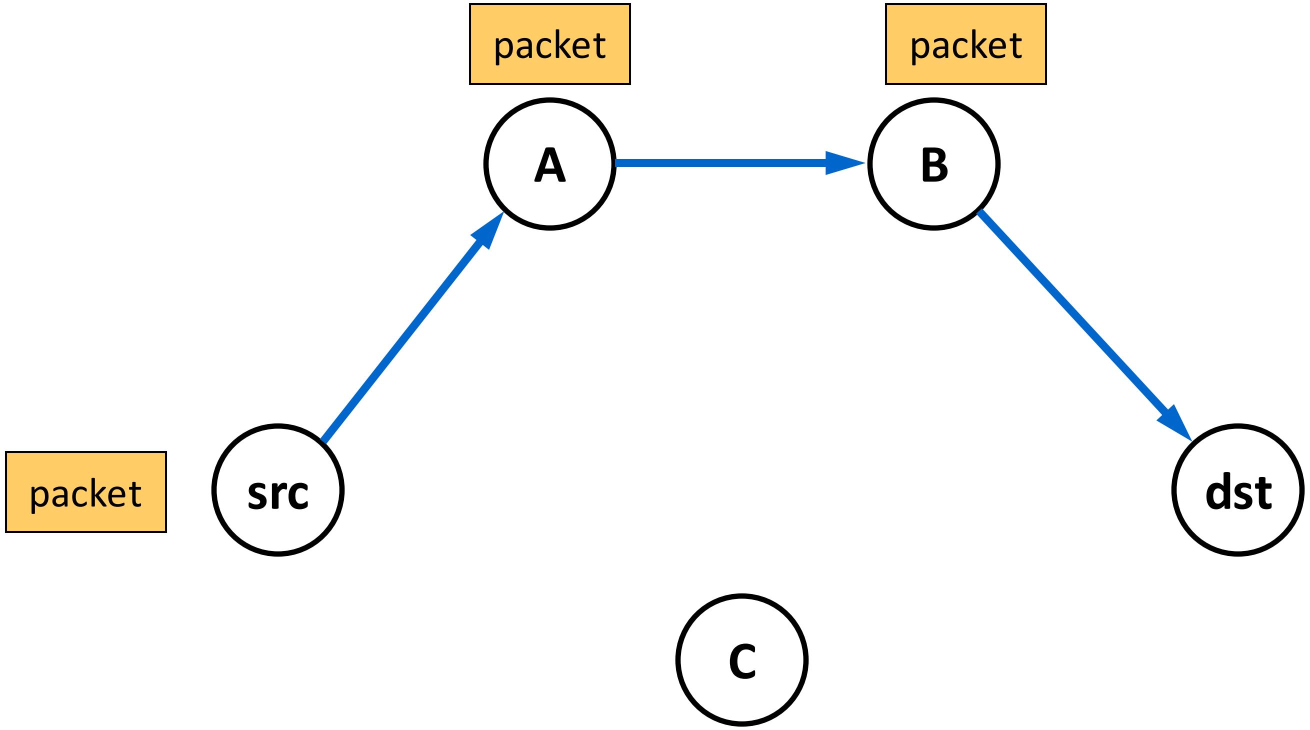
- Forward delivery ratios obtained from neighbors (piggybacked on probes)

ETX Caveats

- It is really hard to measure link quality/loss
  - ➢ Changes as a function of load
  - ➢ Changes with time
- ETX ignores differences in bit-rate and packet size

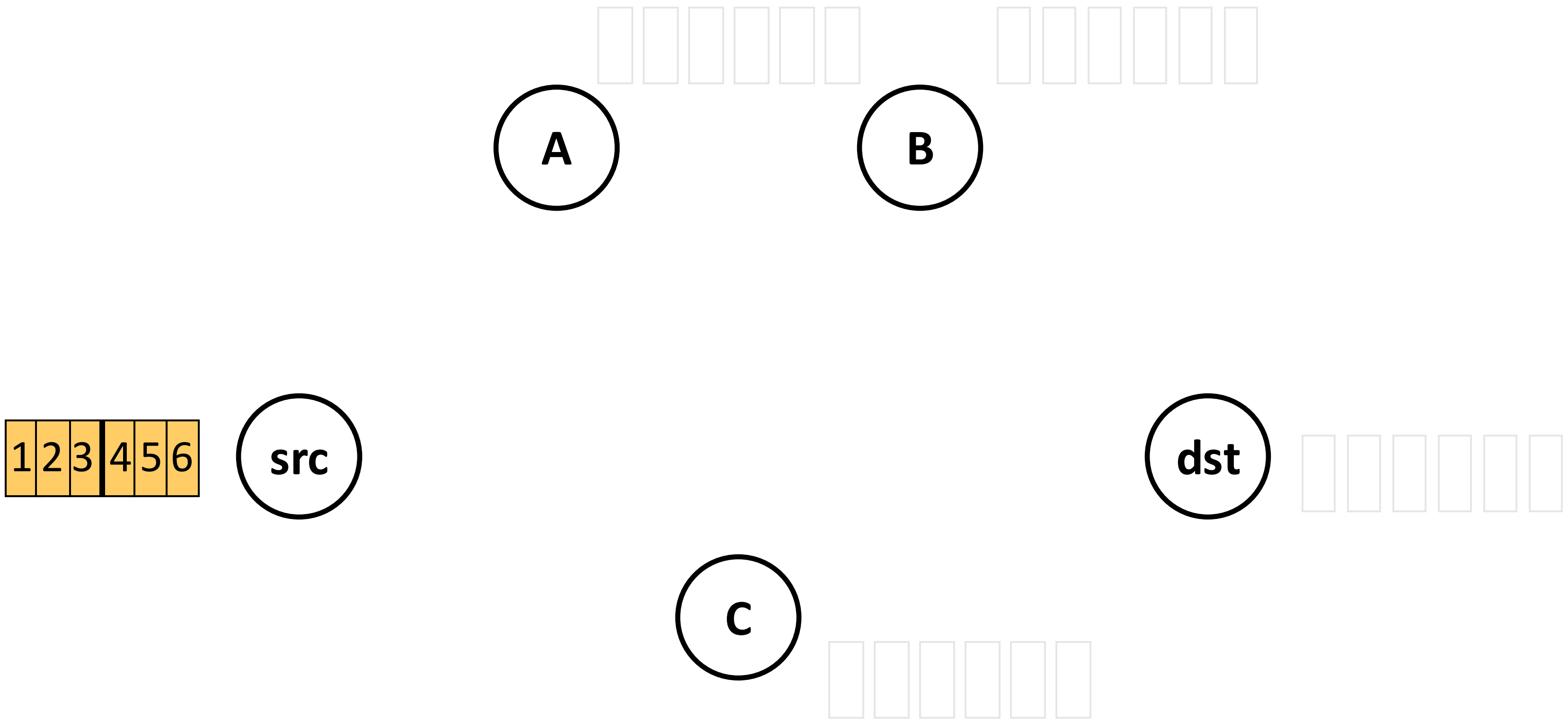
  ETT = ETX *(pkt_size/link-bit-rate)
- ETX ignores spatial re-use (i.e., assumes all links interfere)

Traditional routing



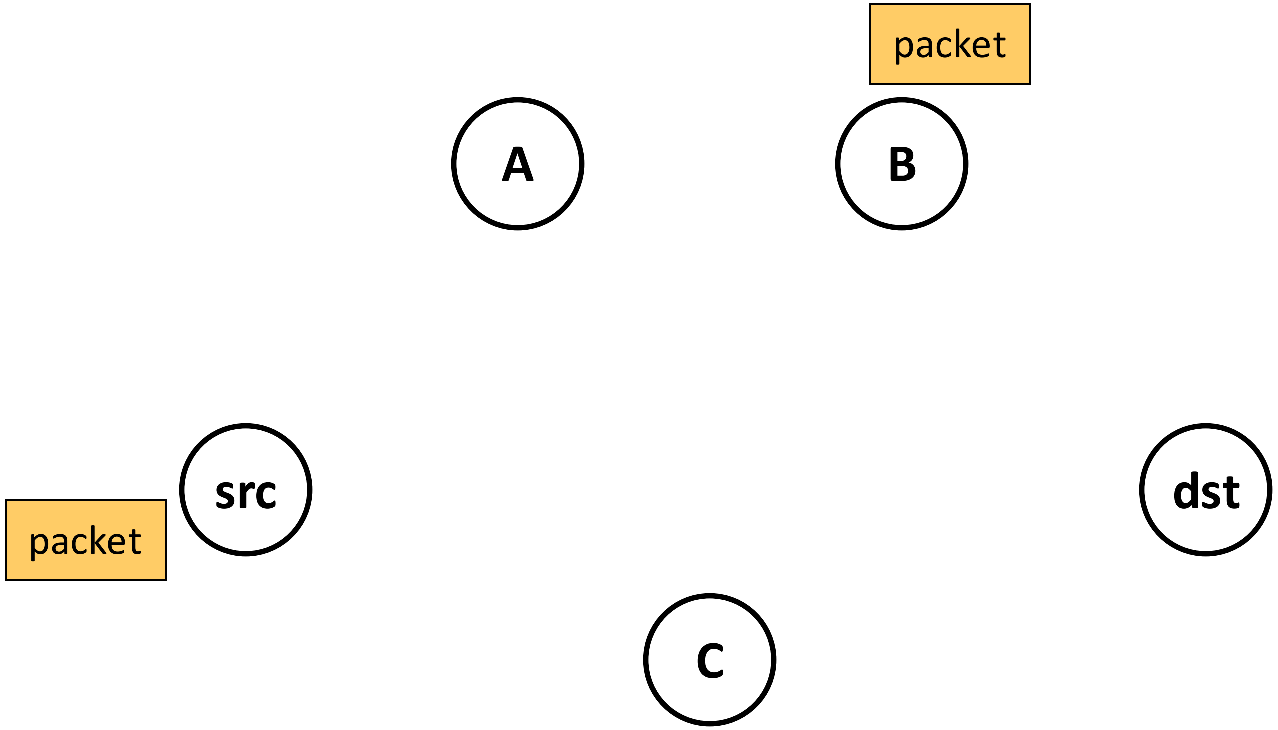
- Identify a route, forward over links
- Abstract radio to look like a wired link

Radios aren't wires



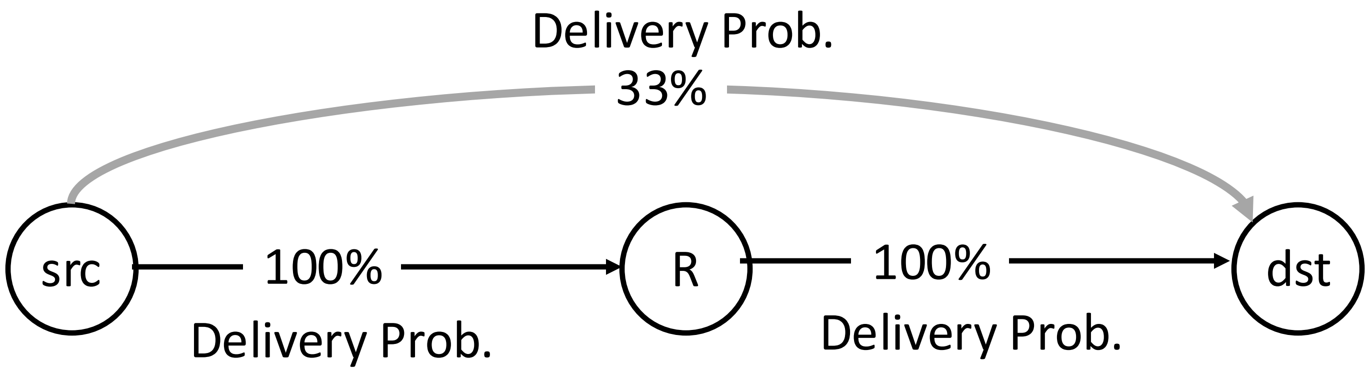
- Every packet is broadcast
- Receptions are probabilistic and independent (Spatial diversity)

ExOR Idea: exploit probabilistic broadcast



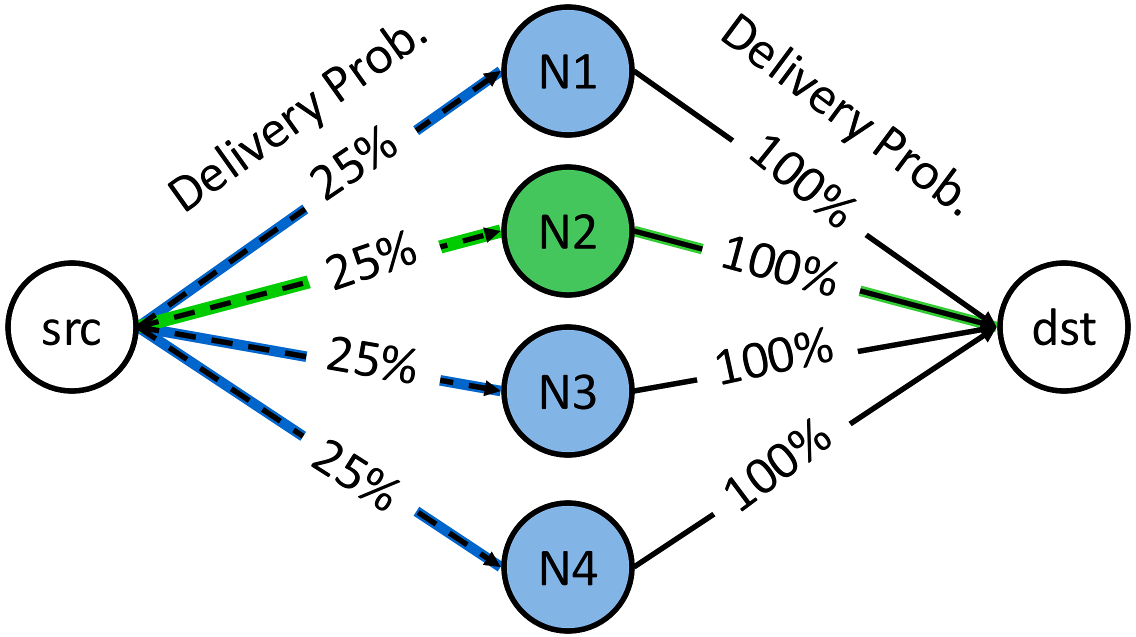
- Decide who forwards after reception
- Goal: for each packet, receiver closest to the destination should forward
- Challenge: agree efficiently on which node should forward, and avoid duplicate transmissions

Why ExOR might increase throughput (1)



- Traditional routing picks the path via R → on average 2 tx per packet
- Throughput $\cong {}^{1}/_{\#\ \text{transmissions}}$
- Traditional routing ignores that 33% of the packets make it to the destination in one transmission
- ExOR exploits these opportunistic receptions → 1.67 tx per packet

Why ExOR might increase throughput (2)



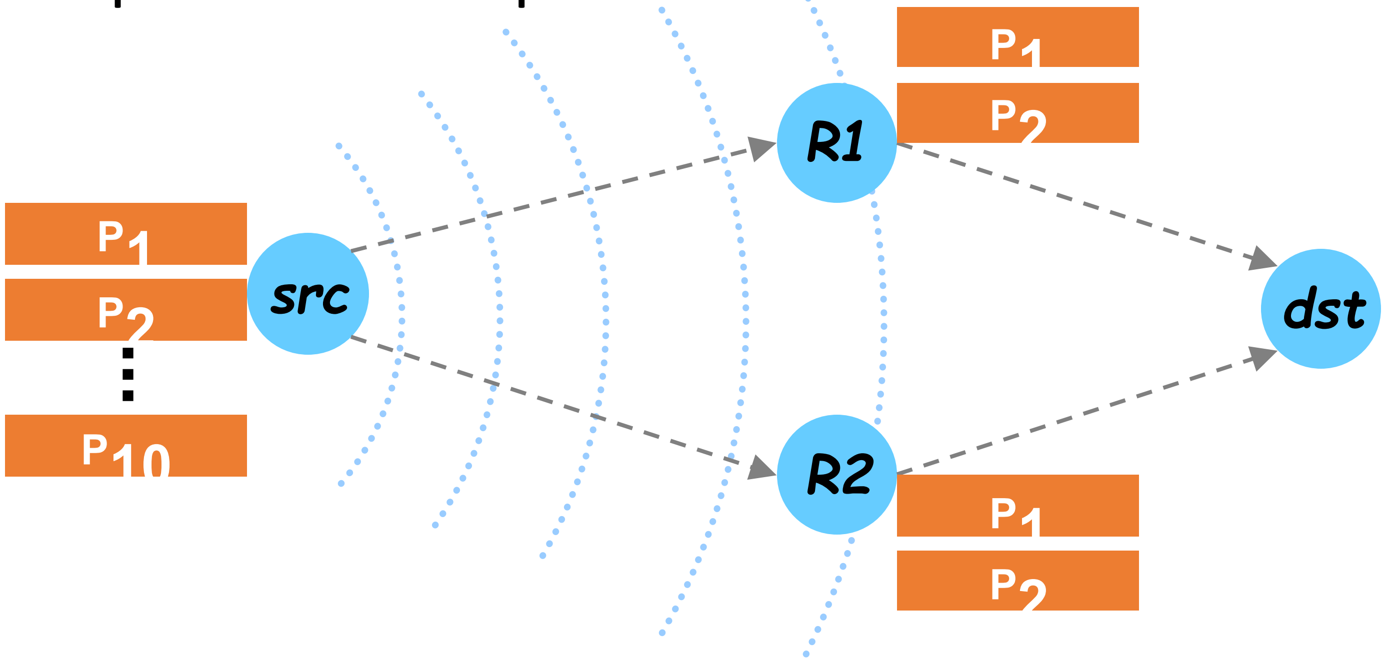
- Traditional routing: $^1/_{0.25} + 1 = 5$ tx
- ExOR: $^1/_{(1 - (1 - 0.25)^4)} + 1 = 2.5$ transmissions

Challenge

Overlap in received packets → Routers forward duplicates

Challenge

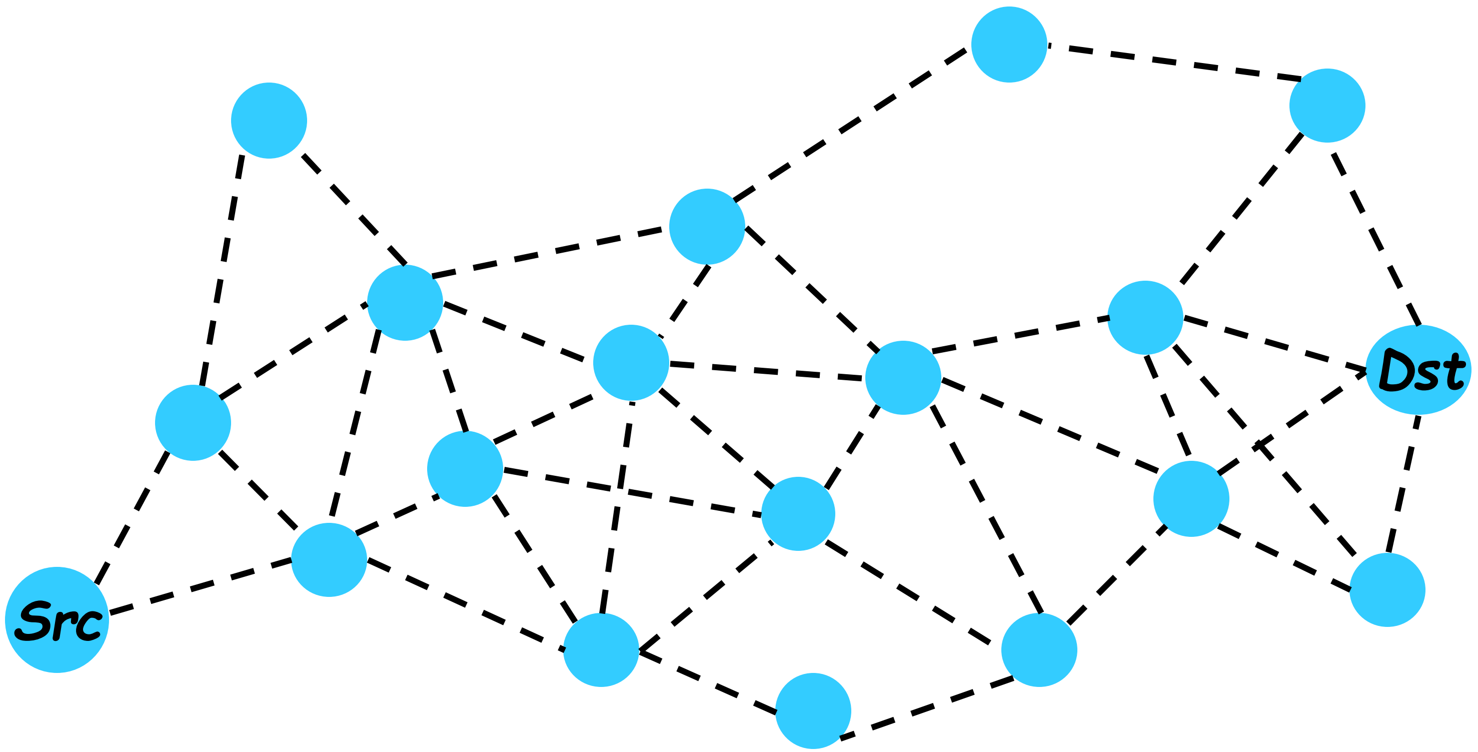
Overlap in received packets → Routers forward duplicates



ExOR imposes a global scheduler:

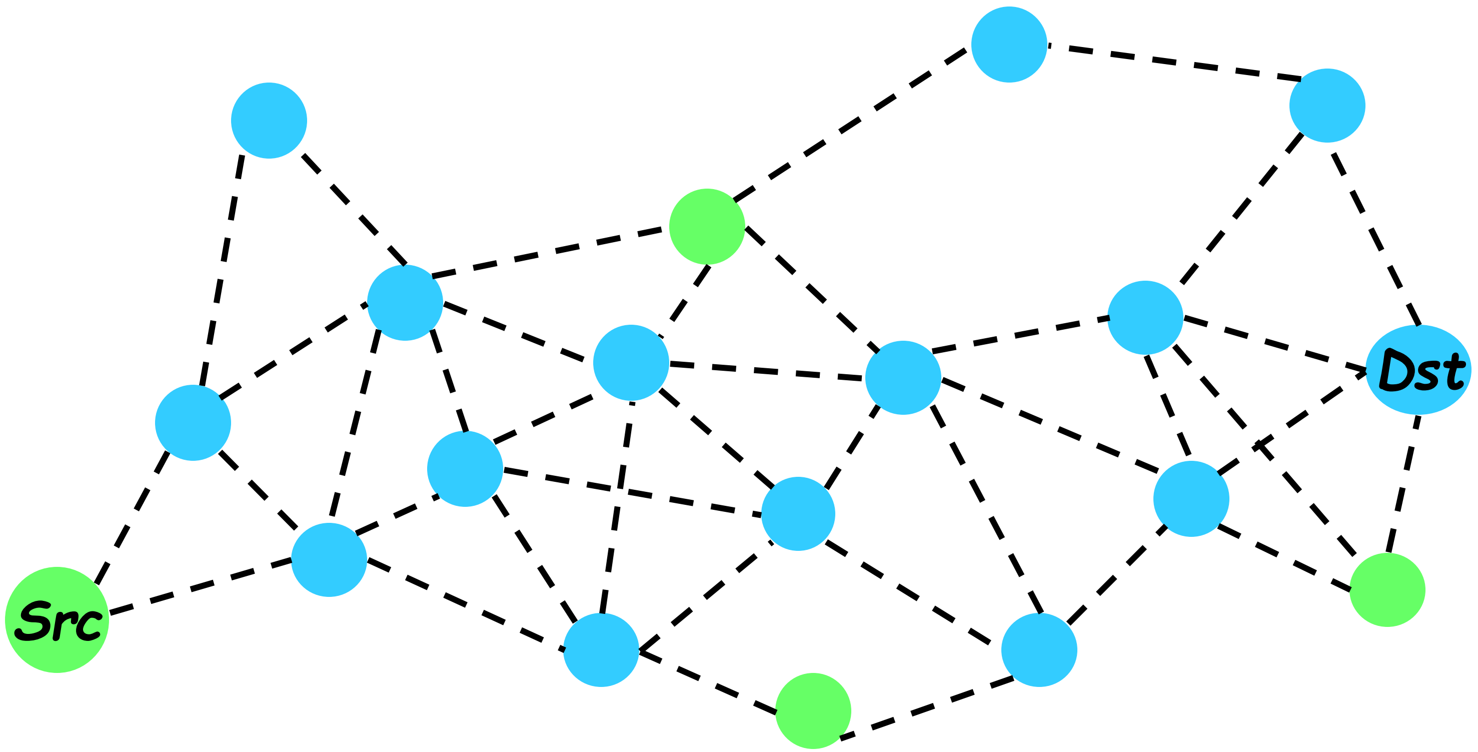
- Nodes have to agree on who transmits what
- Only one node transmits at a time, others listen

Limitations of ExOR



- Learning who received what → too much overhead
- Forcing only one transmitter at a time → prevents spatial reuse of the medium

Limitations of ExOR



- Learning who received what → too much overhead
- Forcing only one transmitter at a time → prevents spatial reuse of the medium

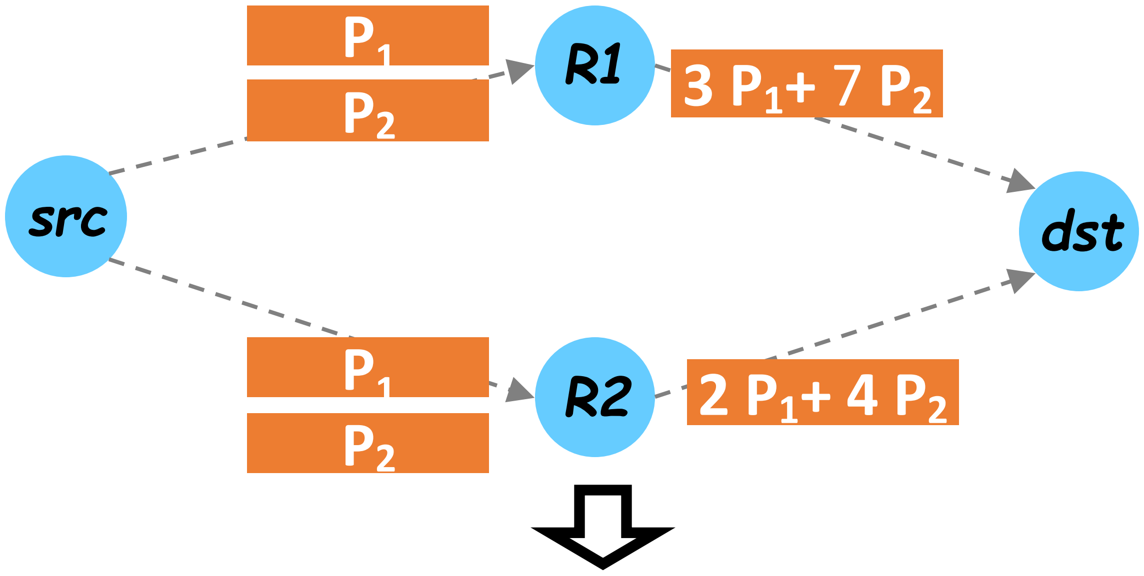
Can we eliminate these problems?

Solution: Random Network Coding

Each router forwards random combinations of packets

Solution: Random Network Coding

Each router forwards random combinations of packets

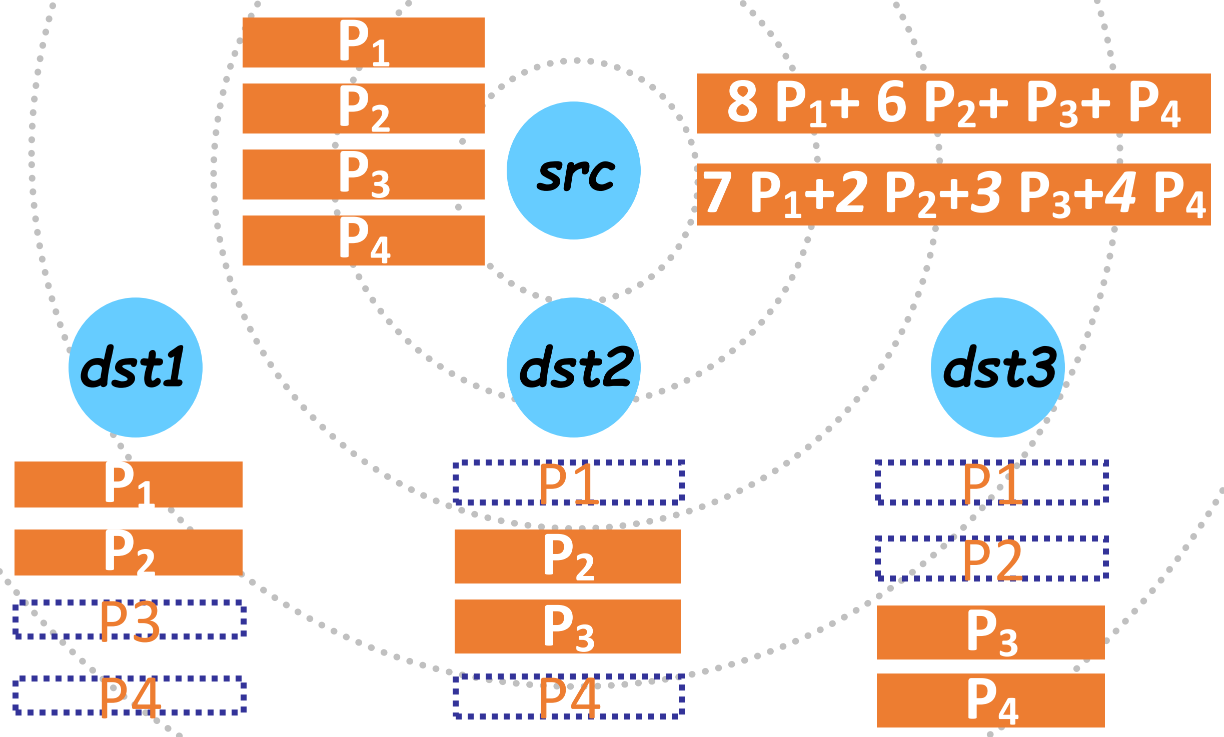


Randomness prevents duplicates

No need to know who received what

Can exploit spatial reuse

Network Coding Also Benefits Multicast



Without coding → source has to retransmit all 4 packets

With network coding → 2 packets are sufficient

MORE

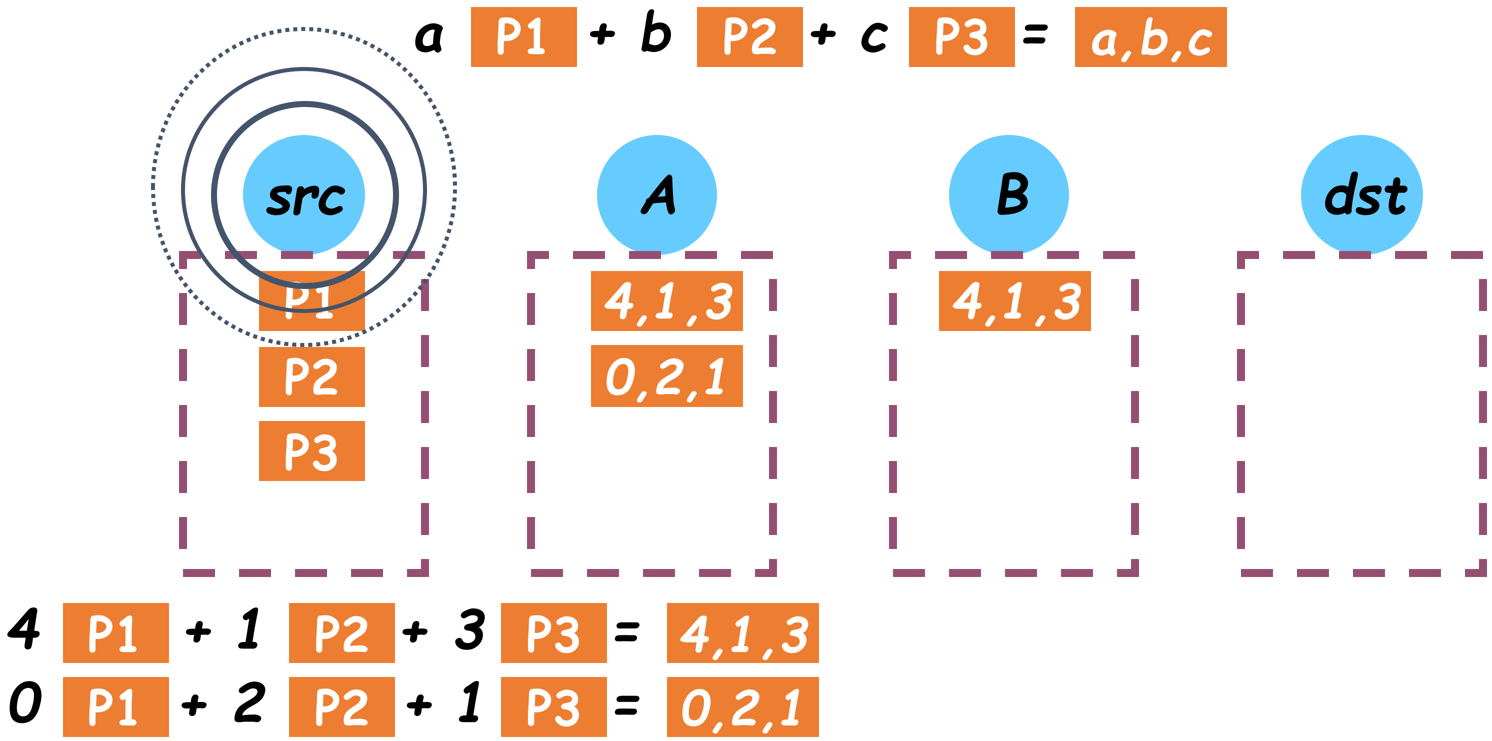
- MAC-independent Opportunistic Routing & Encoding.

- An opportunistic routing protocol that reduces overhead and enables spatial reuse

- It is based on network coding, where routers code packets together before forwarding them

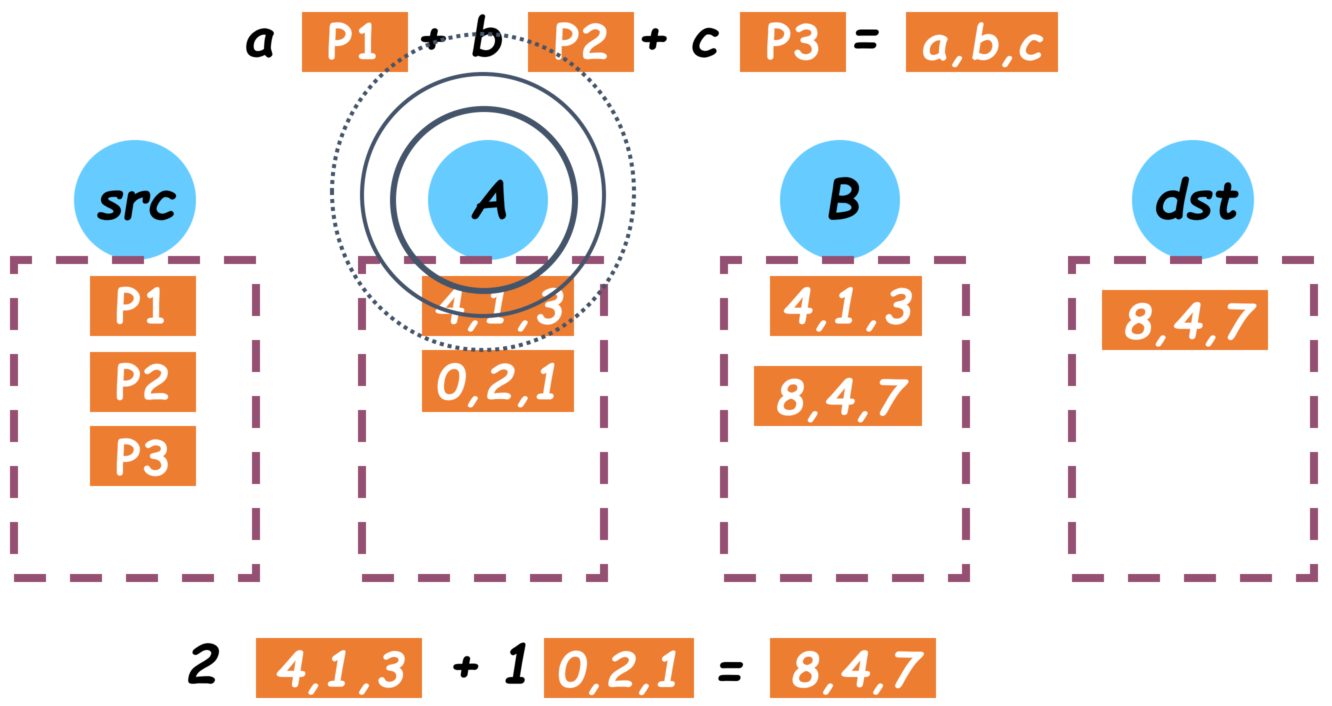
How Does MORE Work?

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



How Does MORE Work?

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



How Does MORE Work?

- Source sends packets in batches
- Forwarders keep all heard (innovative) 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

- Decoding is solving linear equations
- Once it decoded a batch, the destination acks the batch and the source moves to next batch

Network Coding

- Requires less coordination
  - No scheduler
- More flexibility
  - One framework for unicast and multicast
- More throughput
  - 22% more than ExOR and 95% more than current shortest path routing

Two Types of Network Coding

Intra-flow

- Codes packets within a connection

- Robustness to packet loss

- Unicast and Multicast

- E.g., MORE

Two Types of Network Coding

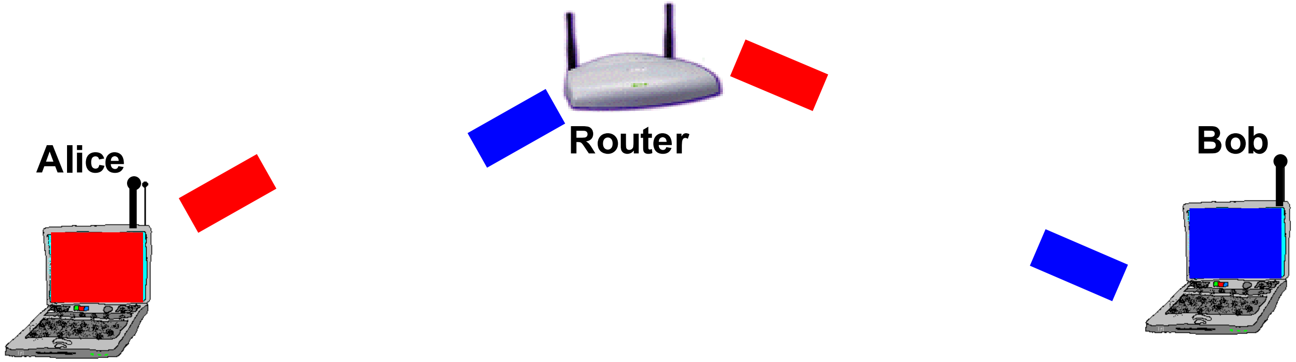
Intra-flow

- Codes packets within a connection
- Robustness to packet loss
- Unicast and Multicast
- E.g., MORE

Inter-flow

- Codes packets across connections
- Higher throughput
- Mainly Unicast
- E.g., COPE

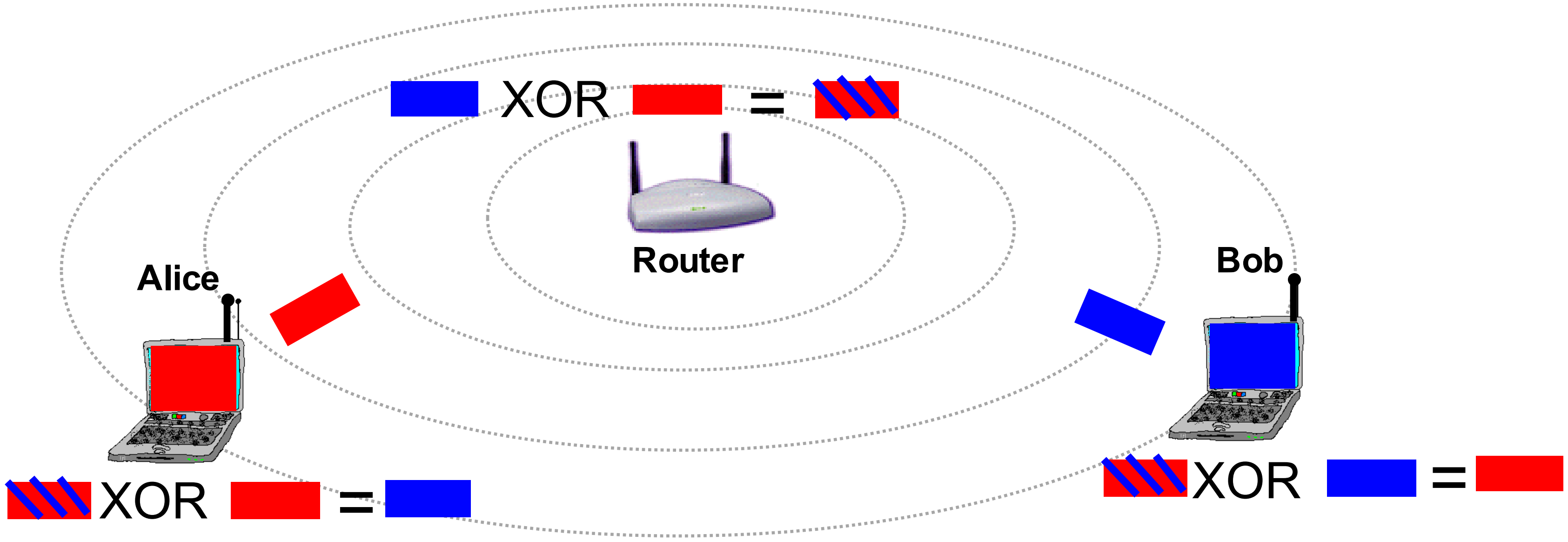
Traditional Approach



Requires 4 transmissions

- Alice to router; Router to Bob; Bob to router; Router to Alice
- Can we exploit broadcast to do better?

COPE

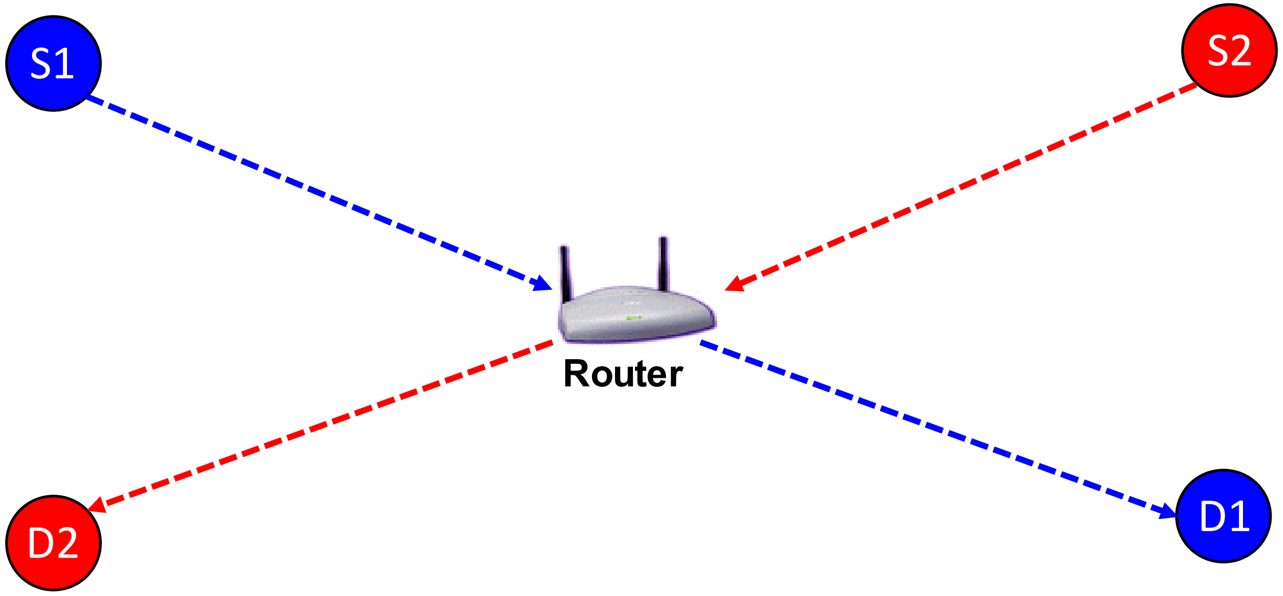


Requires 3 transmissions instead of 4

- Alice to router; Bob to router; and router to both Alice and Bob

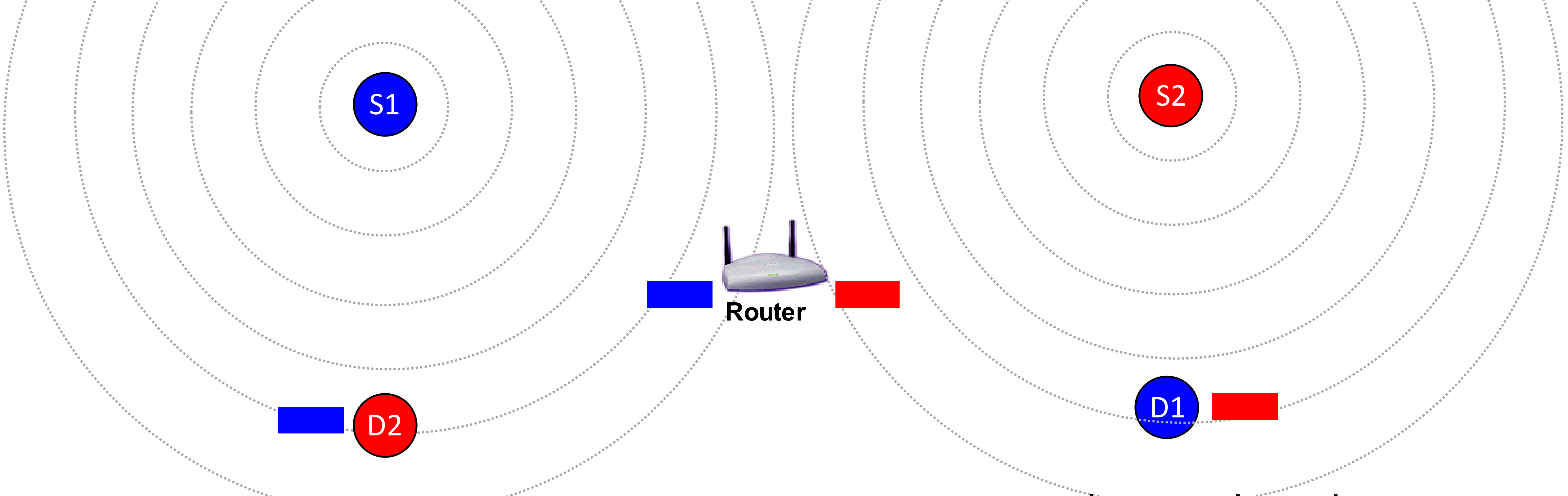
Network Coding → 3 Transmissions instead of 4 → Increases Throughput

Beyond Duplex Communications



Two communication flows that intersect at a router

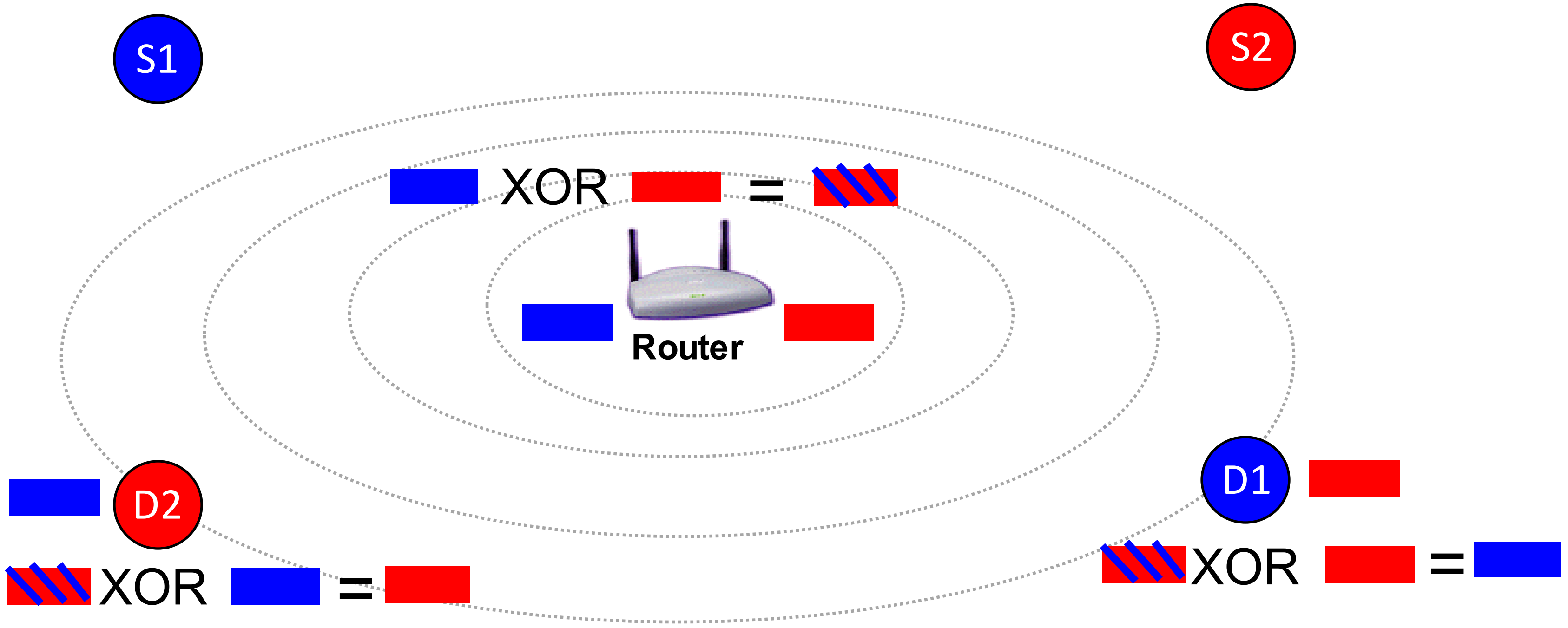
Beyond Duplex Communications



D2 overhears S1's packet

D1 overhears S2's packet

Beyond Duplex Communications



3 transmissions instead of 4 → Higher Throughput

COPE

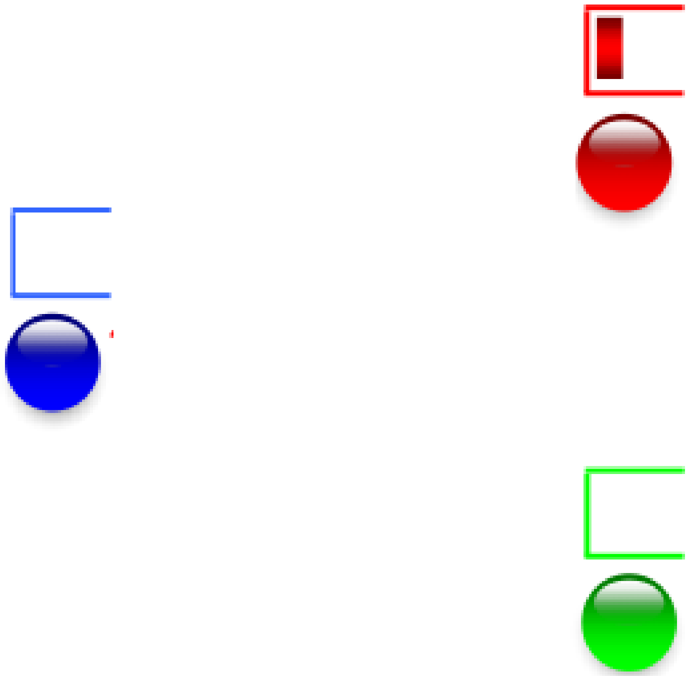
- Opportunistic Listening
- Opportunistic Coding

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

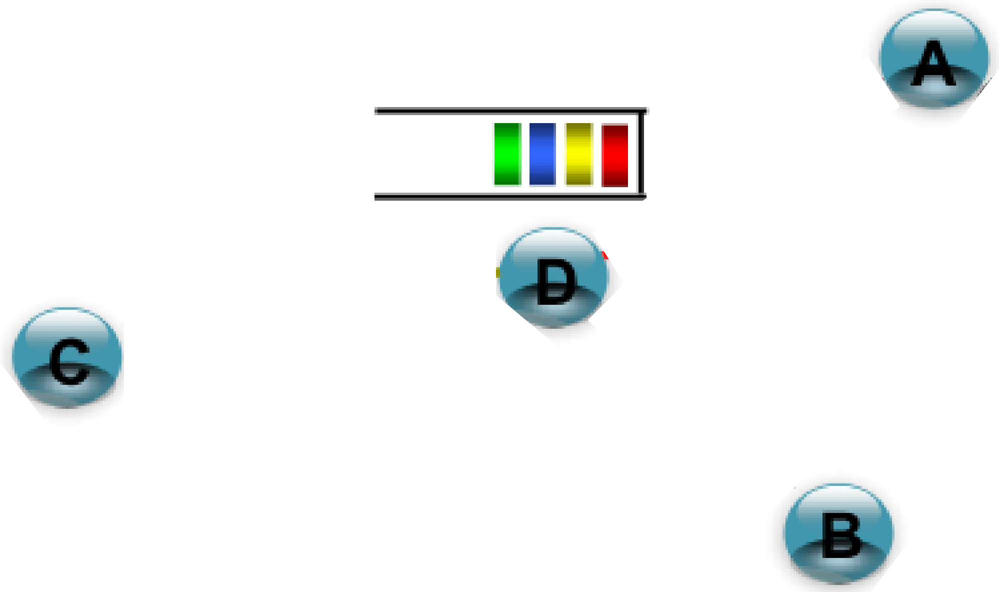
- Exploit wireless broadcast
- Every node snoops on all packets
- A node stores all heard packets for a limited time

- 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

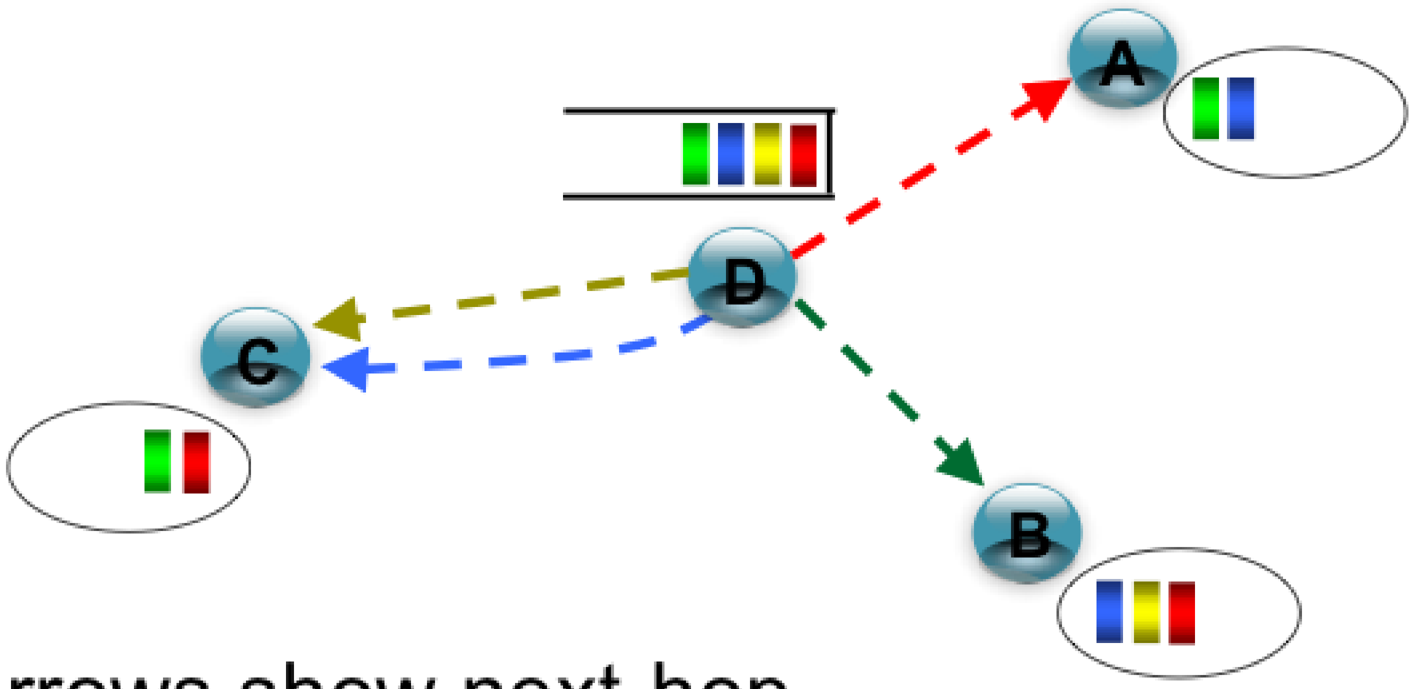
Opportunistic Coding

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

Which Packets to Code Together?

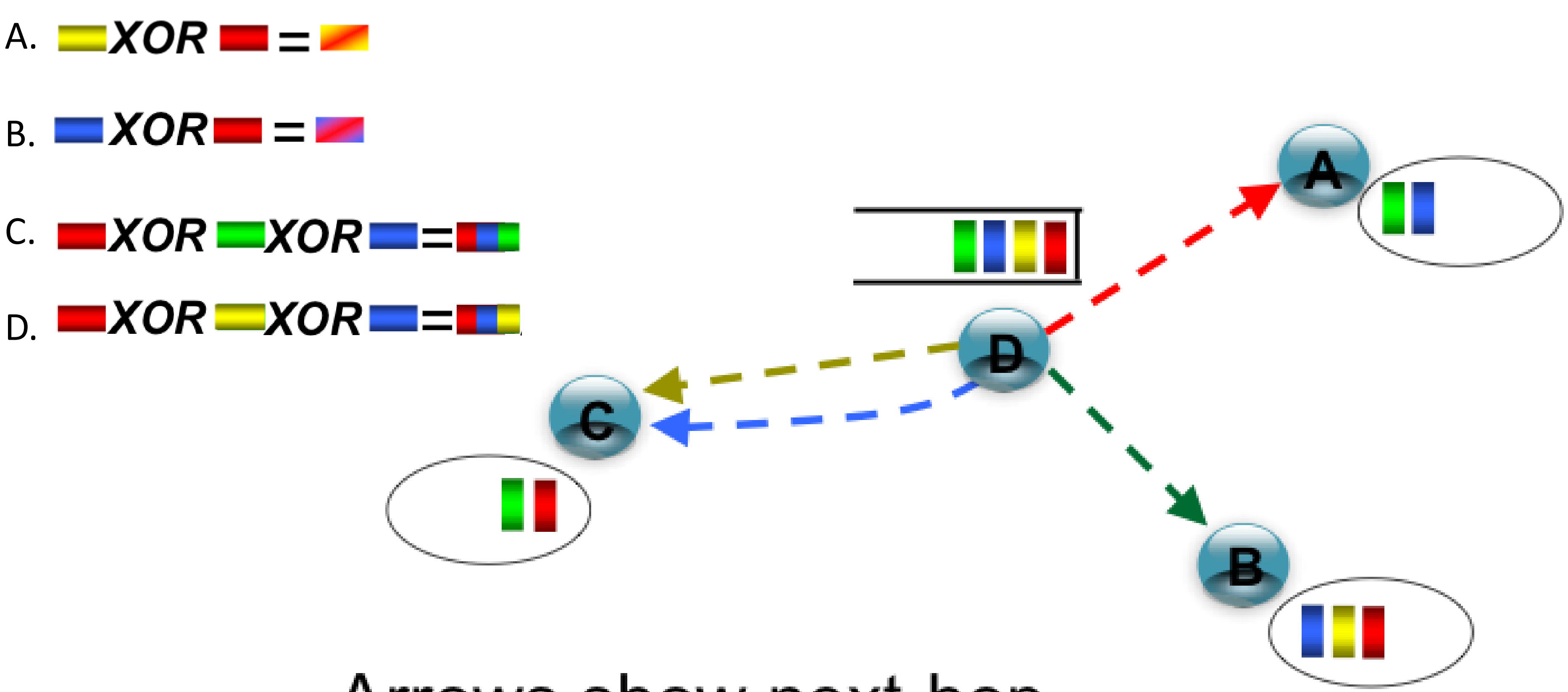


Which Packets to Code Together?



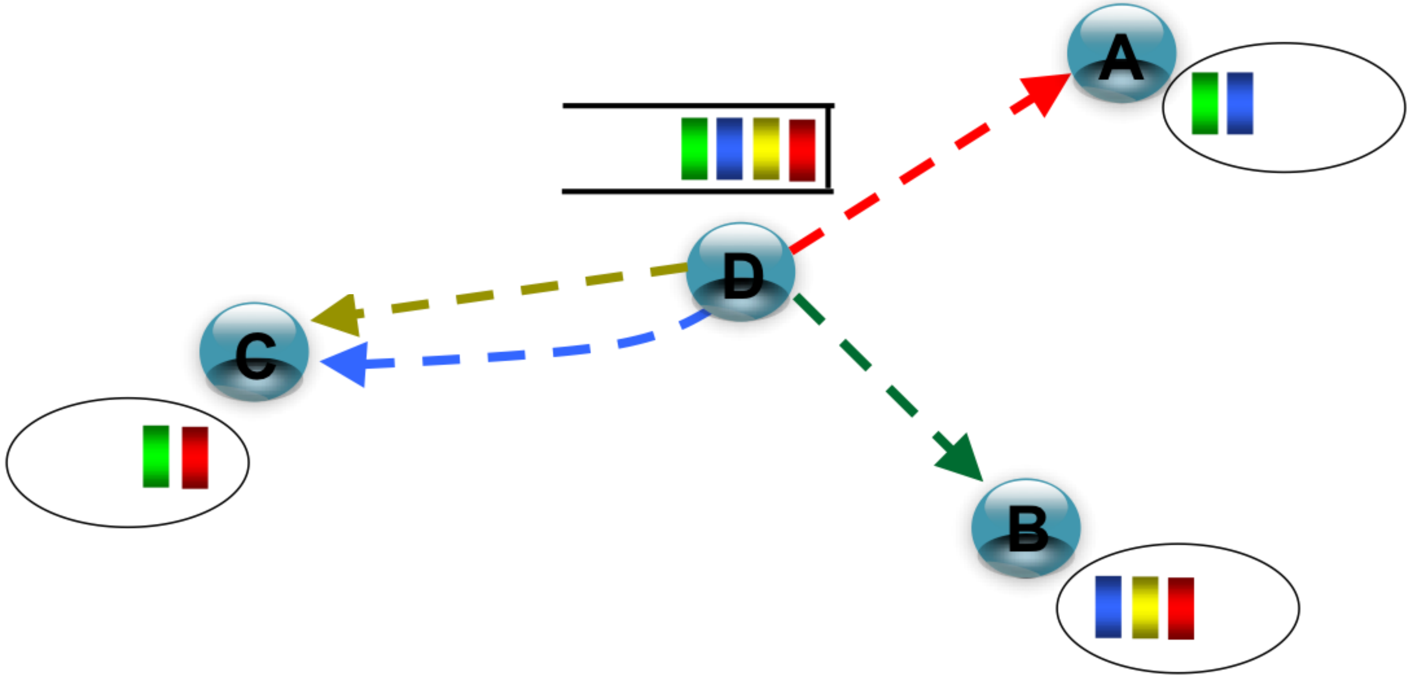
Arrows show next-hop

Which Packets to Code Together?

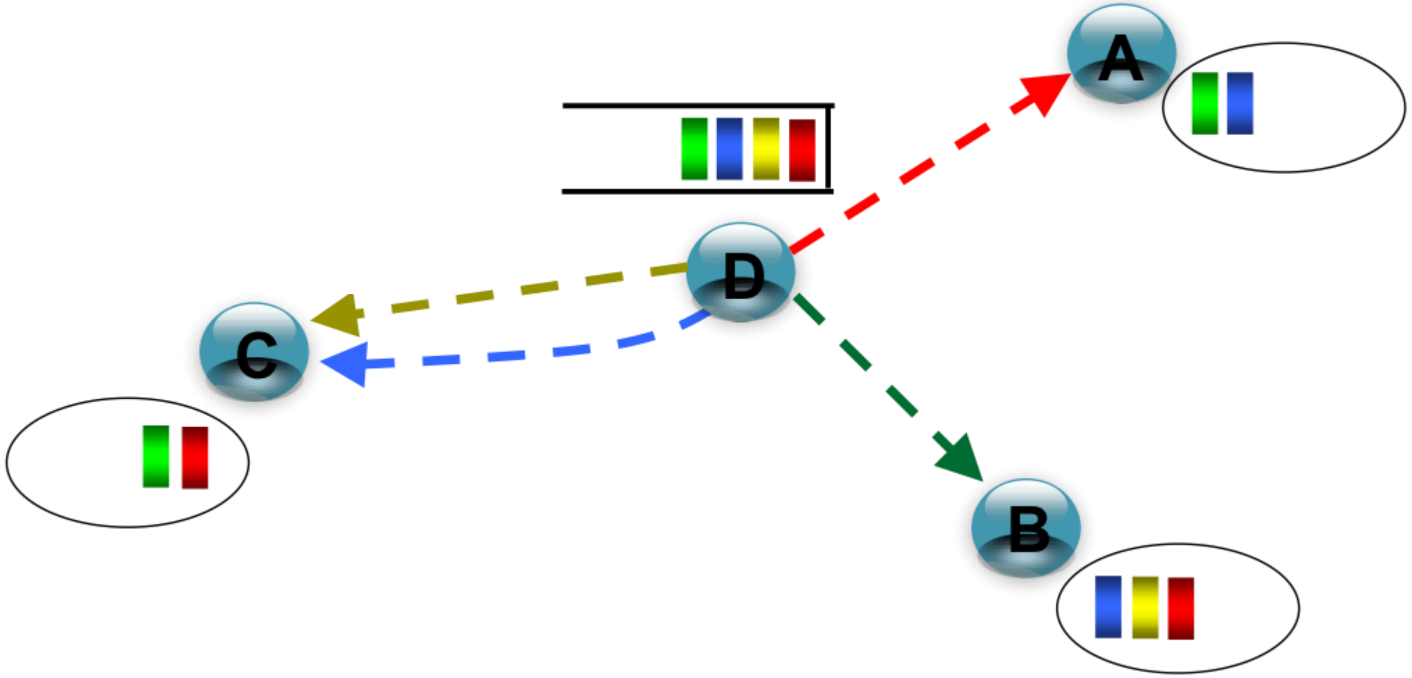


Arrows show next-hop

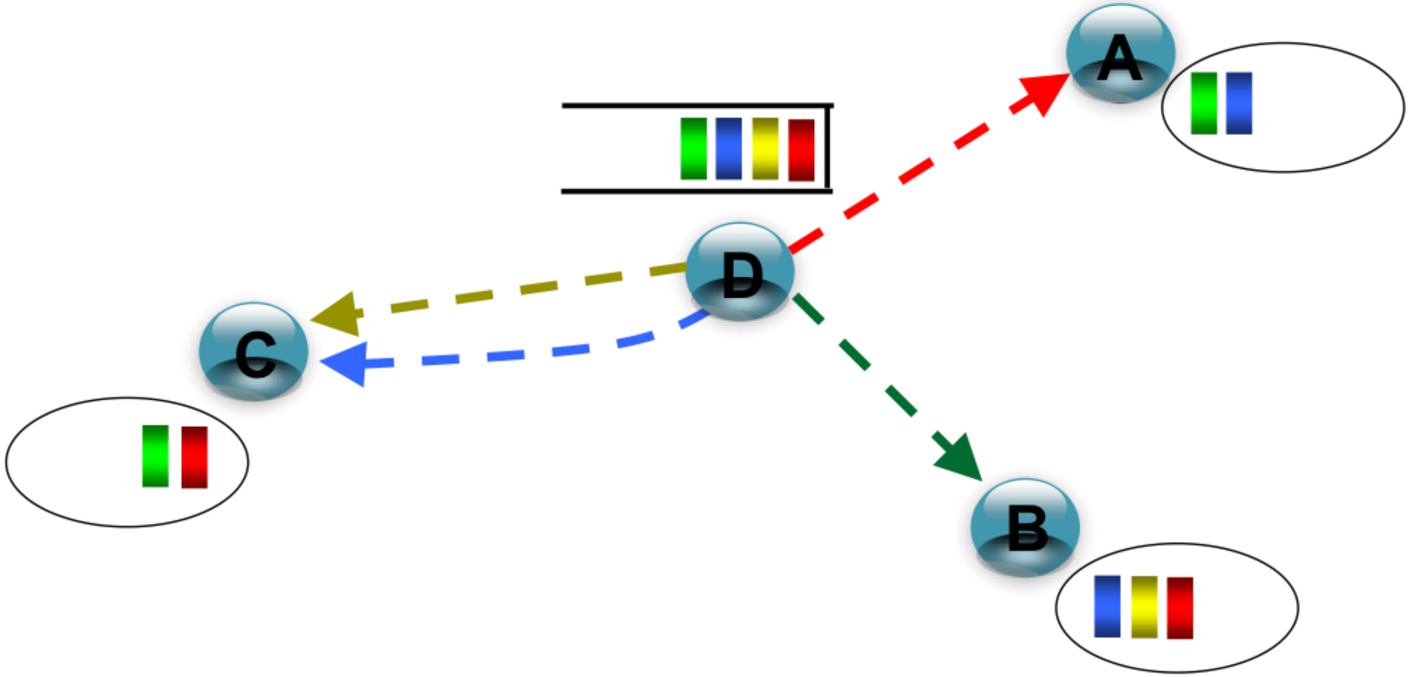
Which Packets to Code Together?



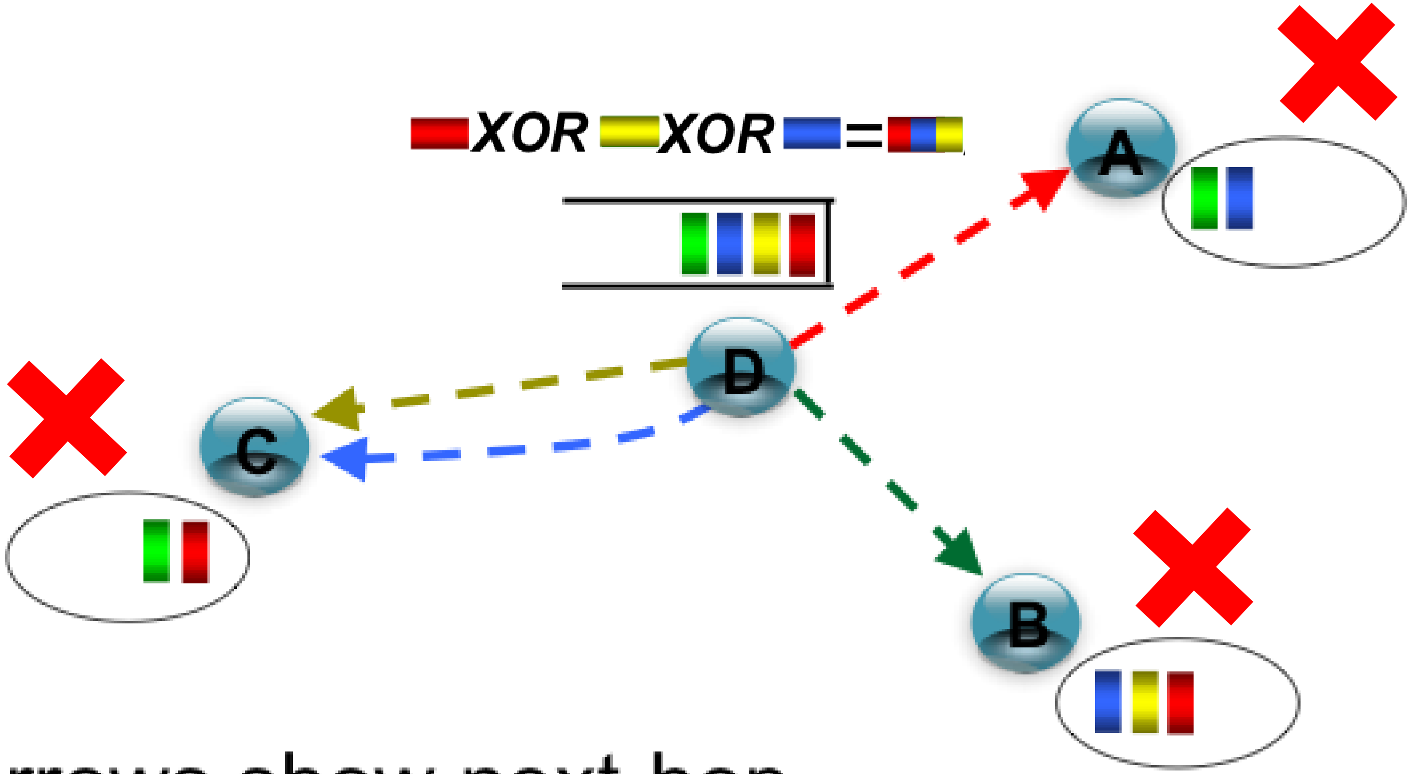
Which Packets to Code Together?



Which Packets to Code Together?

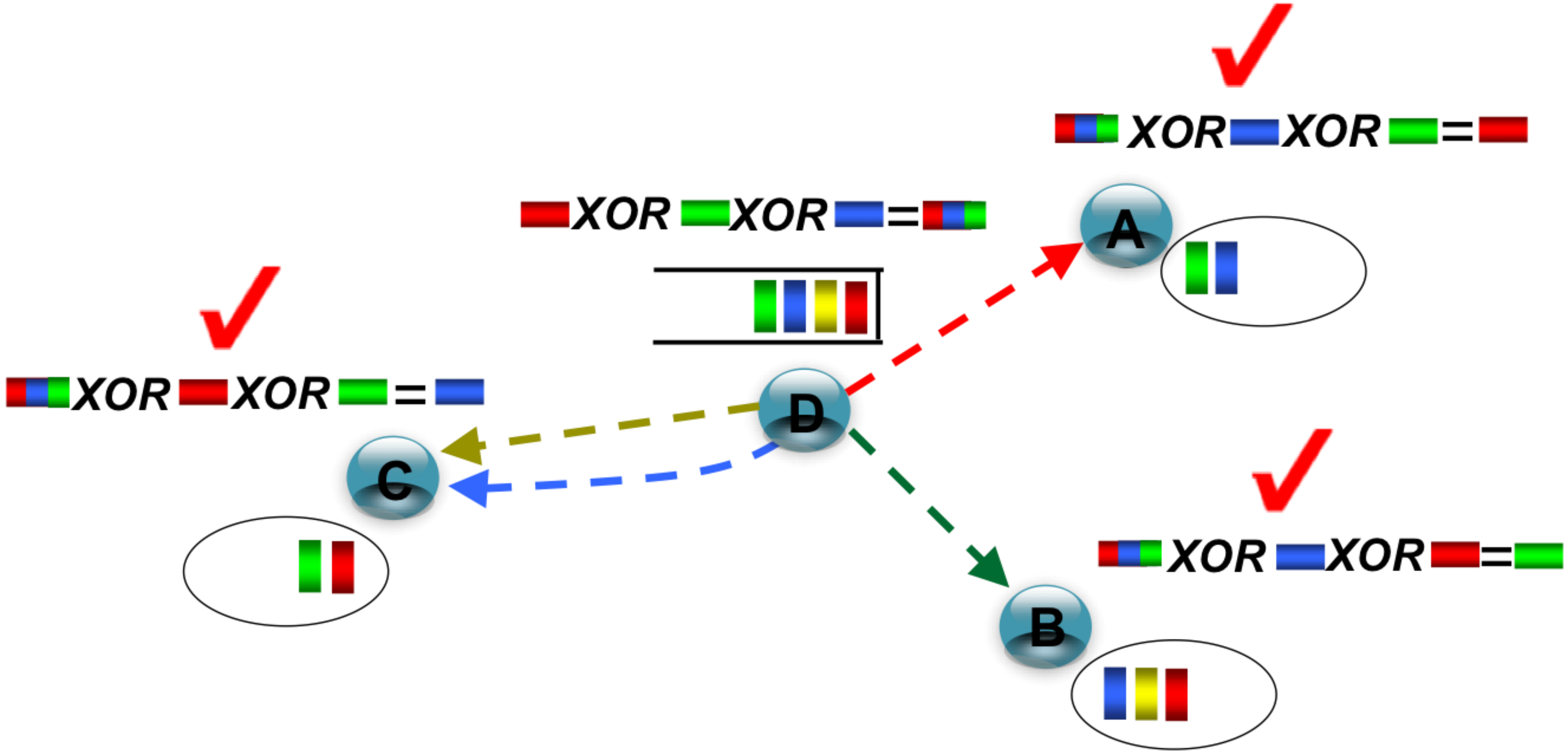


Which Packets to Code Together?



Arrows show next-hop

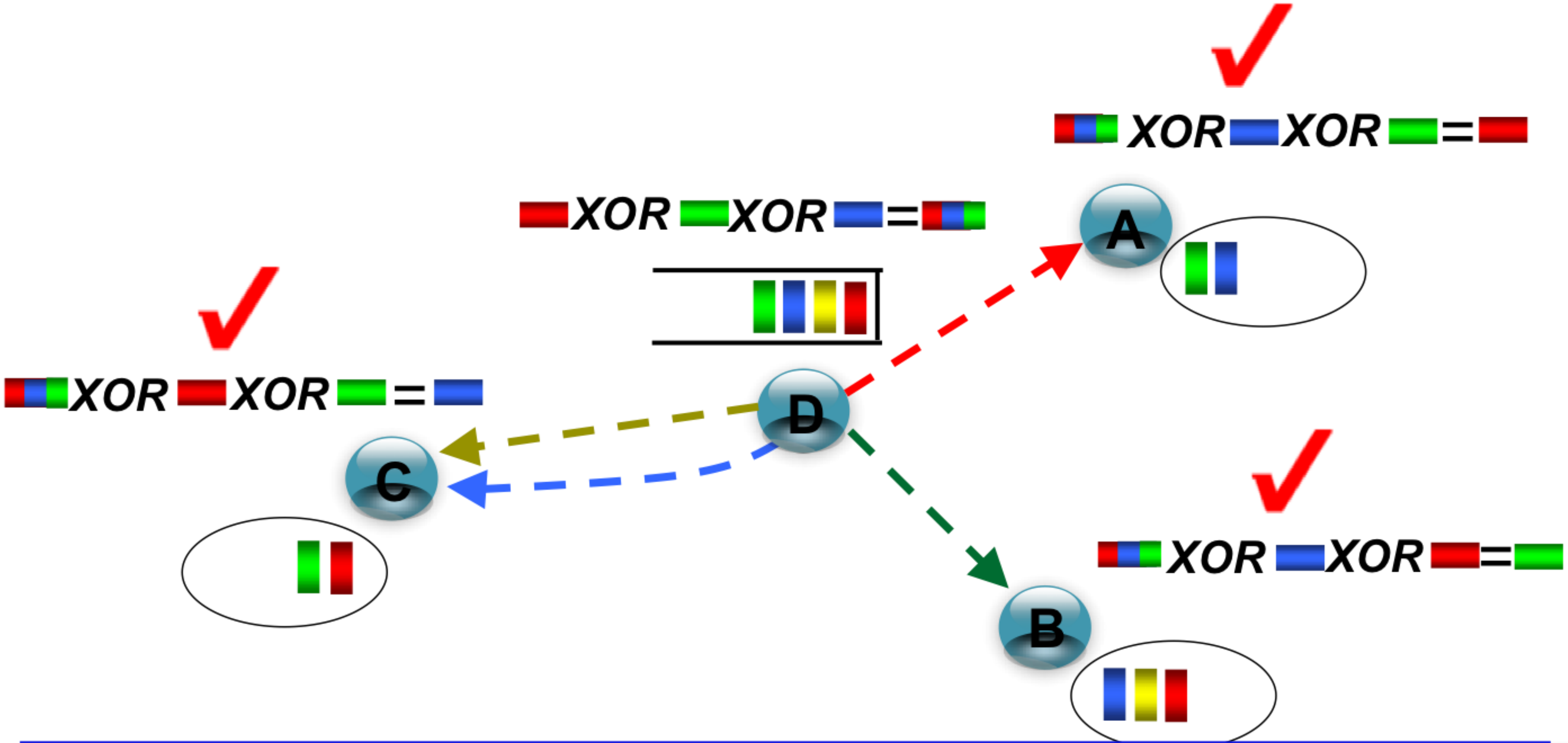
Which Packets to Code Together?



Best Coding

Three neighbors benefit from one transmission!

Which Packets to Code Together?



XOR n packets together iff the next hop of each packet already has the other ***n-1*** packets apart from the one it wants

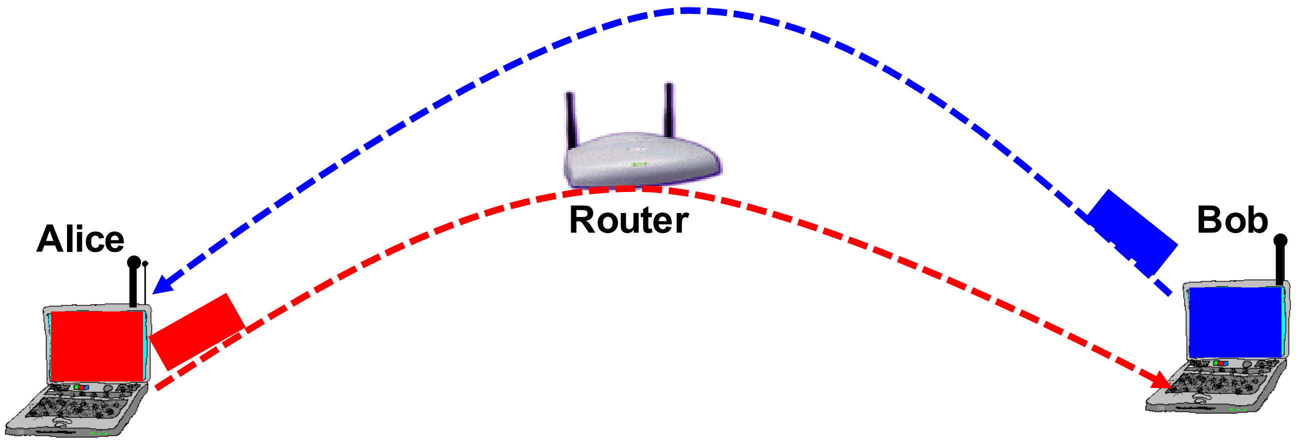
But, to decode a node needs to know which packets are XOR-ed

COPE’ s Characteristics

- COPE is a forwarding mechanism
  - It sits transparently between IP and MAC
  - Routing is unmodified (i.e., shortest path)

Performance

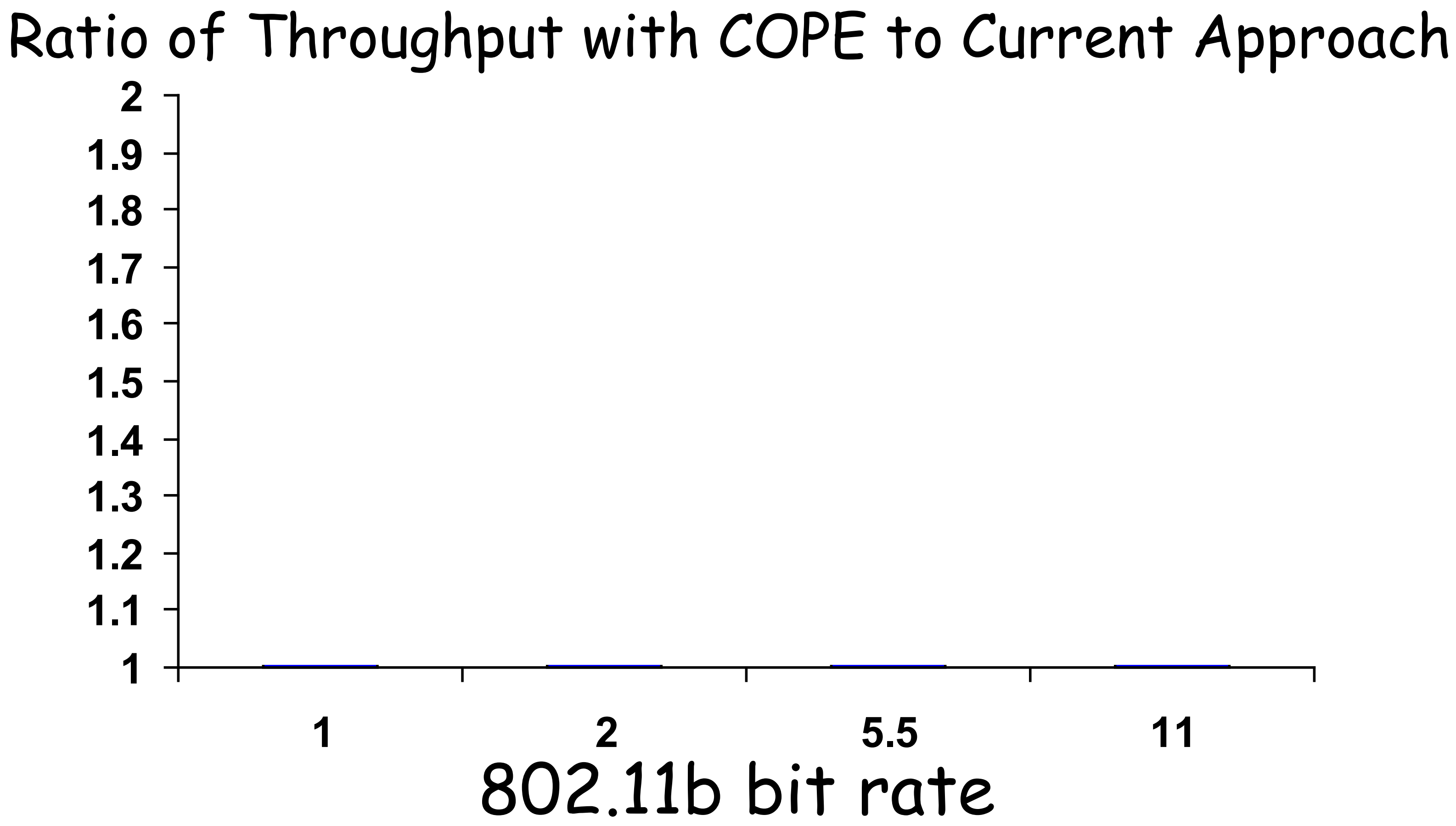
Alice and Bob Experiment



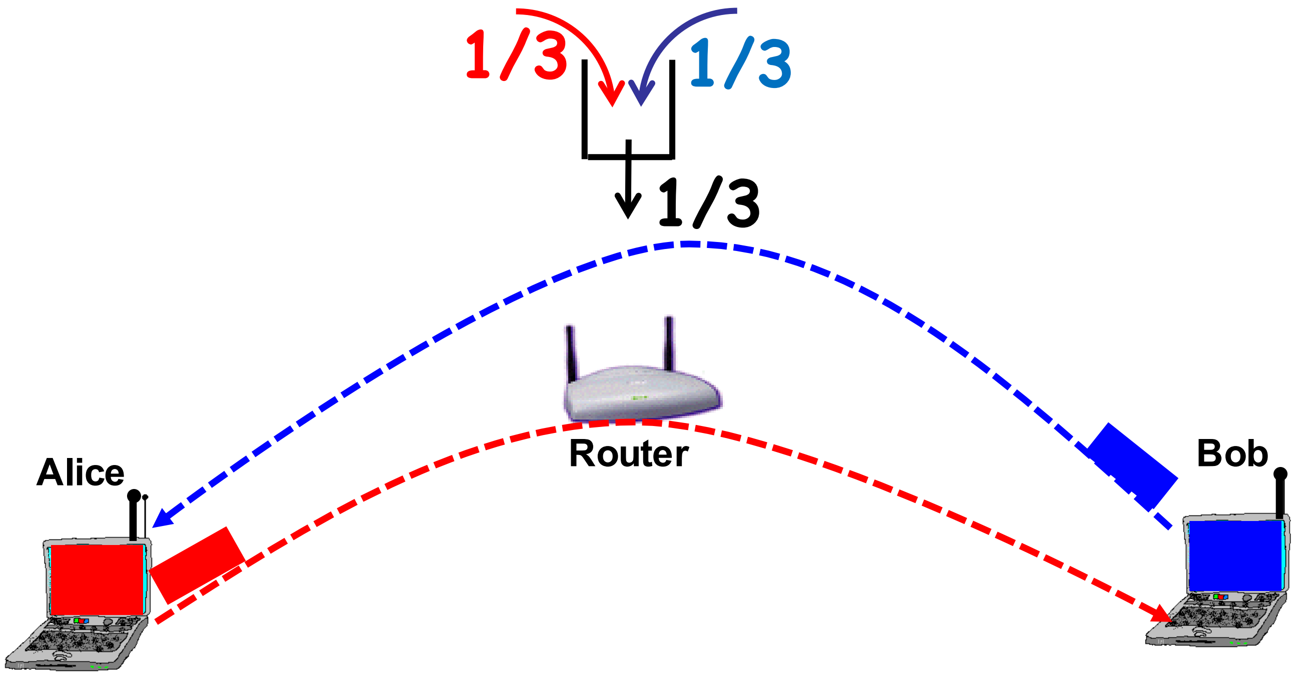
3 transmissions instead of 4

→ Throughput Gain = 4/3 = 1.3333

Results of the Alice-and-Bob



Why the Gain is more than 1.33 ?



802.11 is fair → Each node transmits 1/3

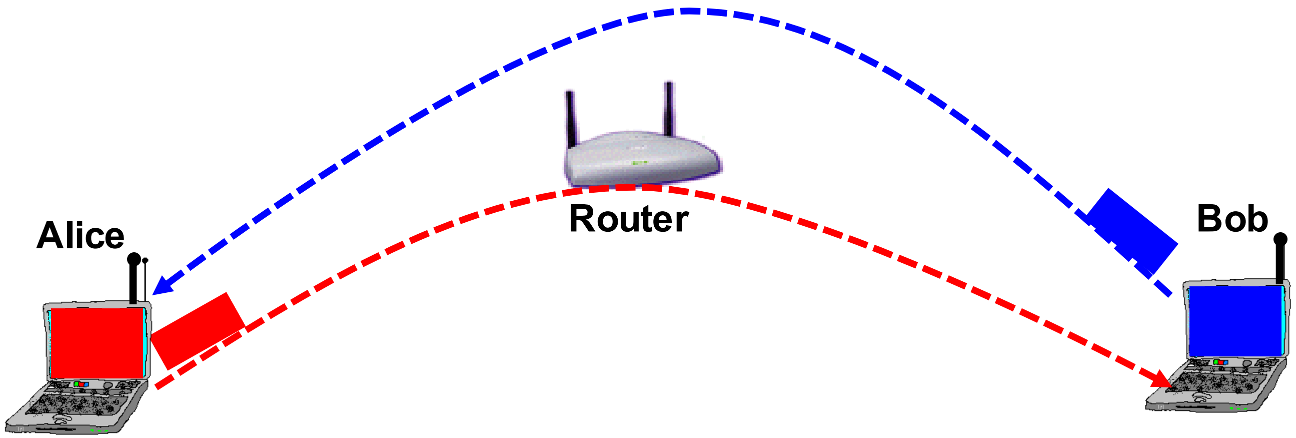
- Without COPE: Router needs to transmit twice as much
- With COPE: All nodes need equal rate.

COPE alleviates the mismatch between MAC's allocation and the congestion at a node

Coding Gain

Coding+MAC Gain

Limitations of COPE



Traditional Approach: requires 4 transmissions

COPE: requires 3 transmissions

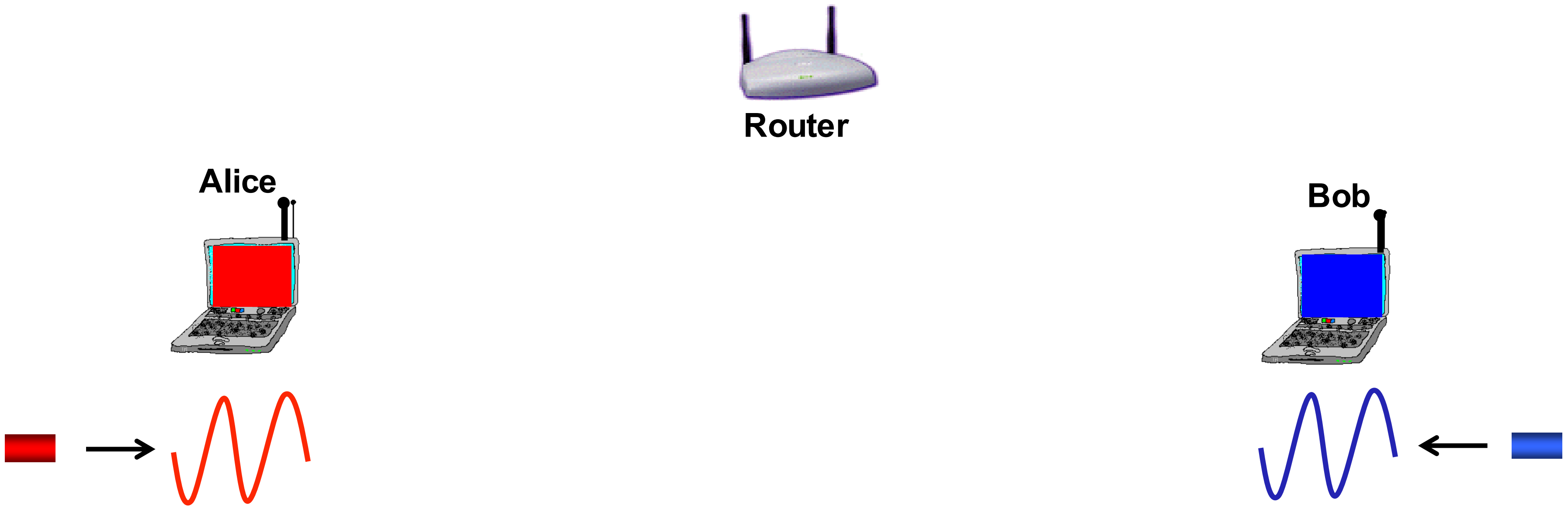
Can we do it in 2 transmissions?

Instead of router mixing packets…

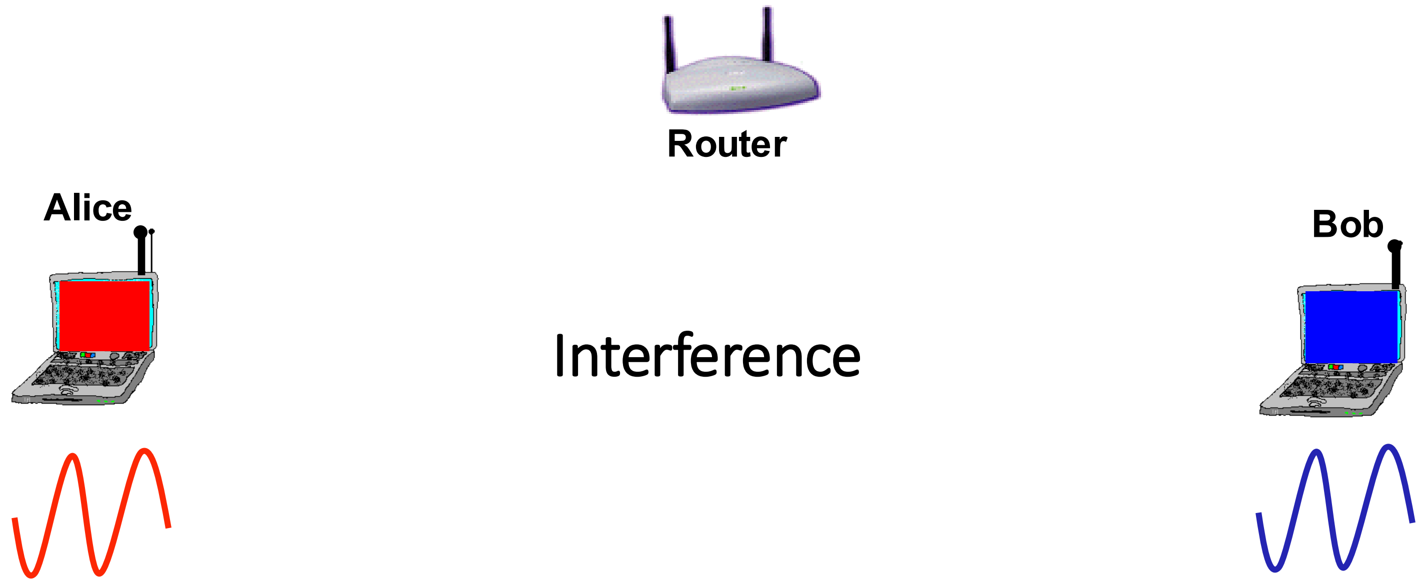
Exploit that the wireless *channel naturally mixes signals*

Analog Network Coding (ANC)

Analog Network Coding

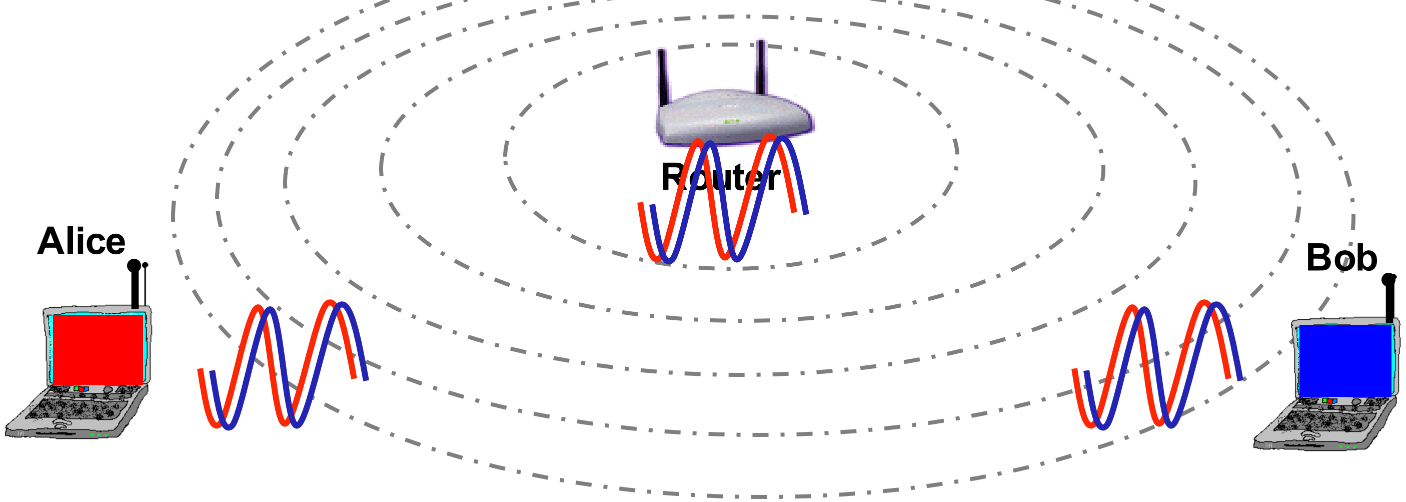


Analog Network Coding



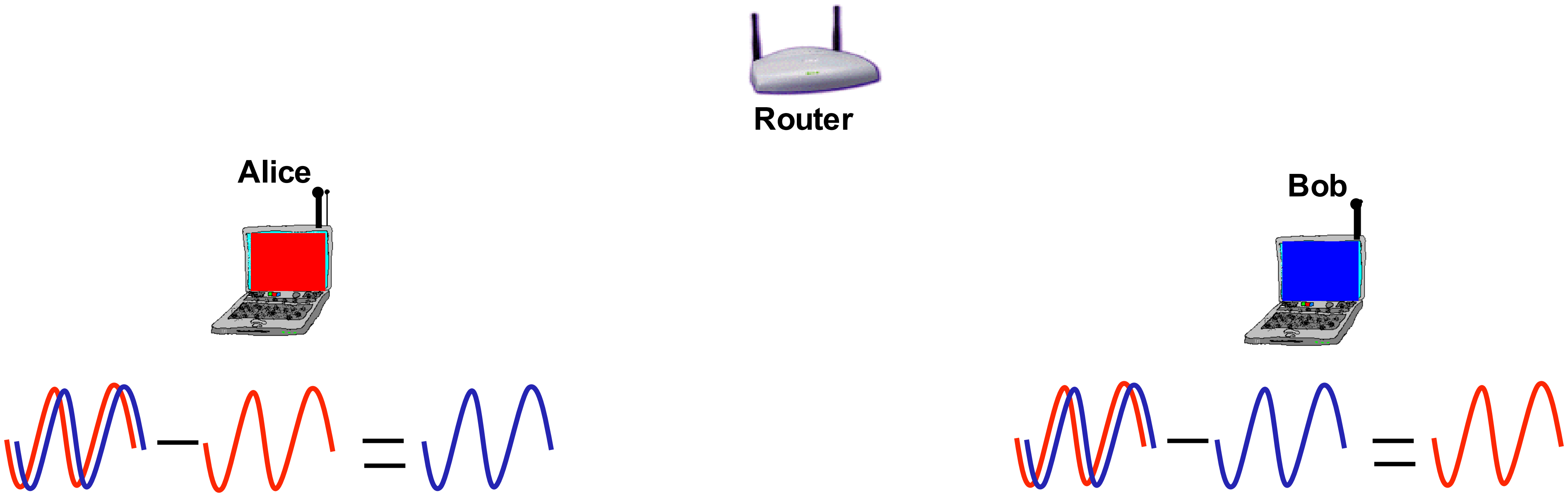
1) Alice and Bob transmit simultaneously

Analog Network Coding



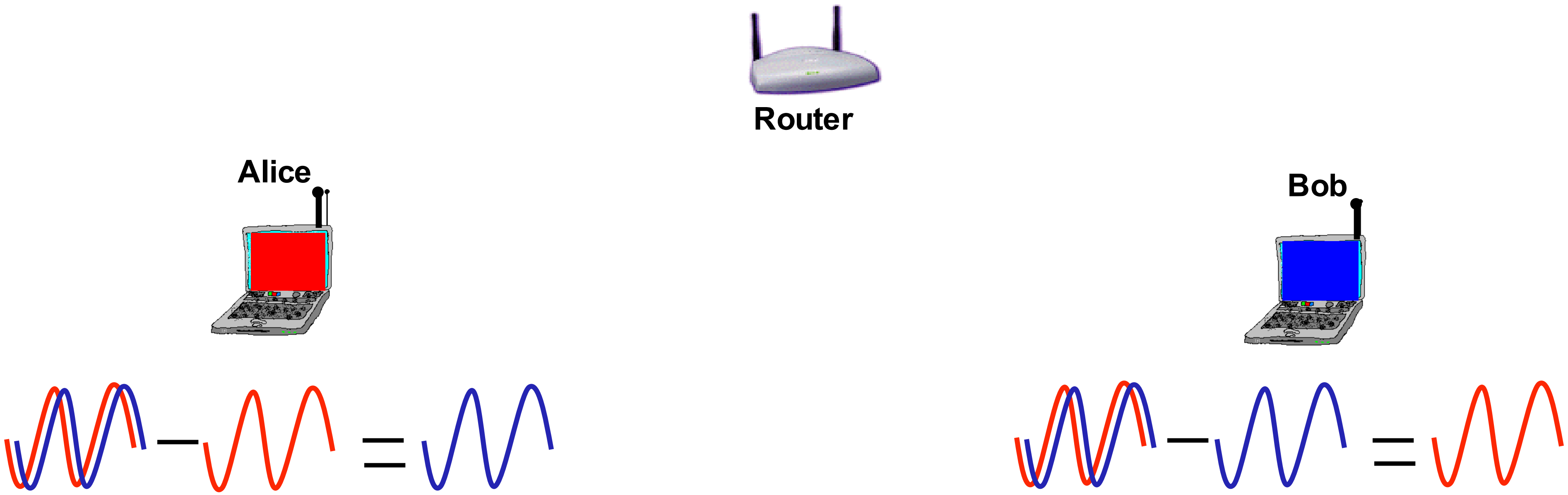
1) Alice and Bob transmit simultaneously
2) Router amplifies and broadcasts interfered signal

Analog Network Coding



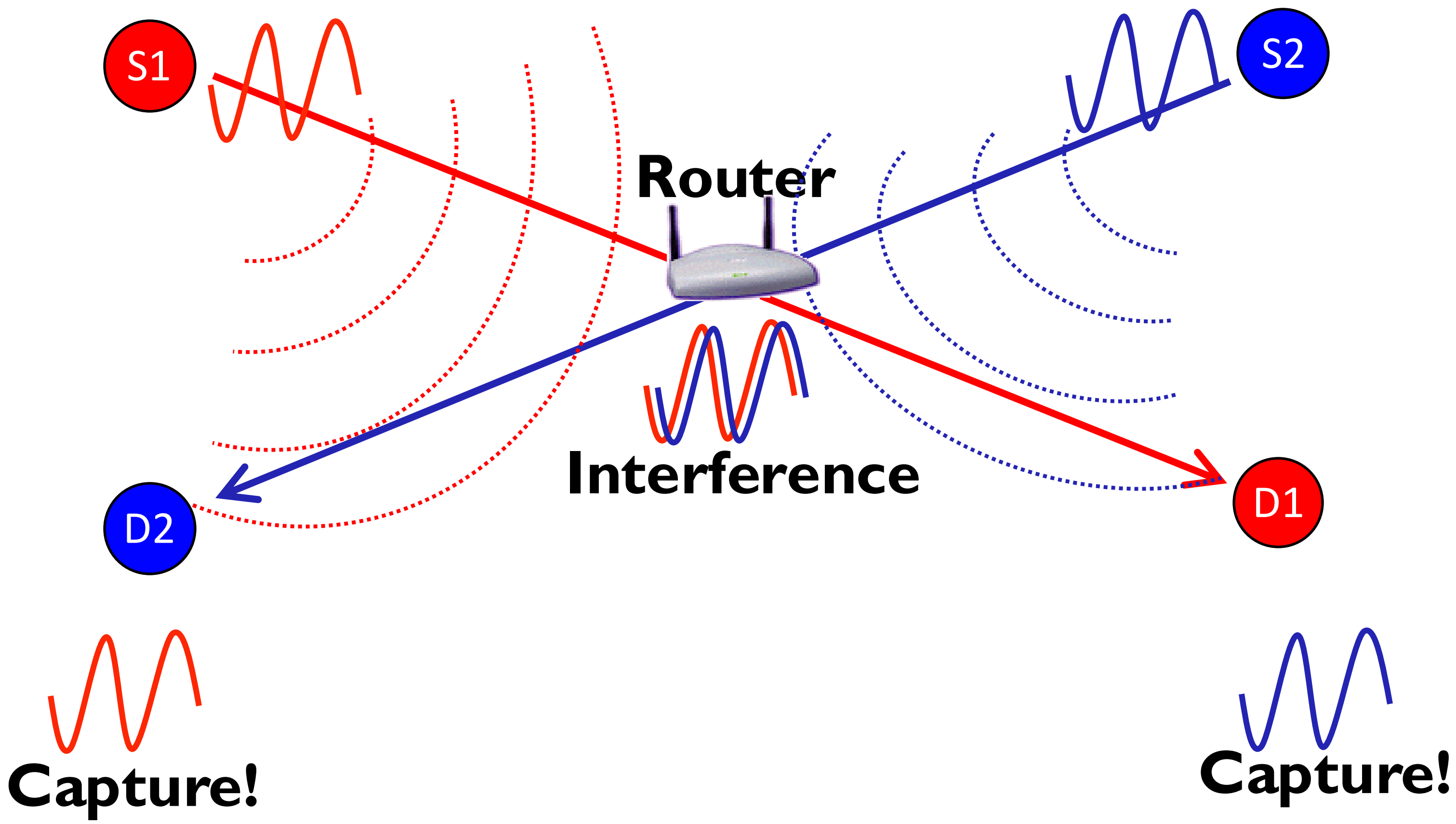
1) Alice and Bob transmit simultaneously
2) Router amplifies and broadcasts interfered signal
3) Alice subtracts known signal from interfered signal

Analog Network Coding

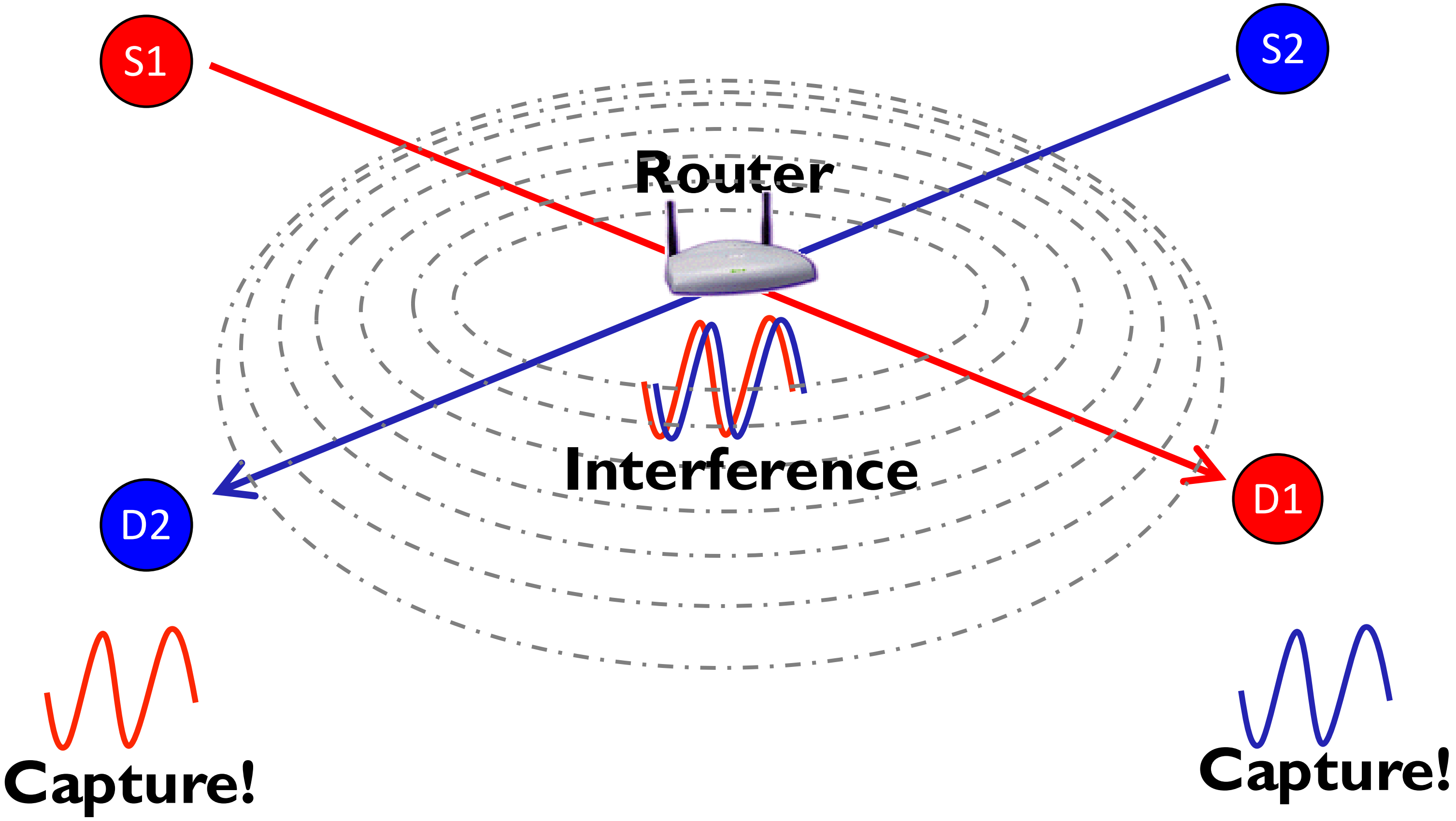


Analog Network Coding requires 2 time slots
→Higher throughput

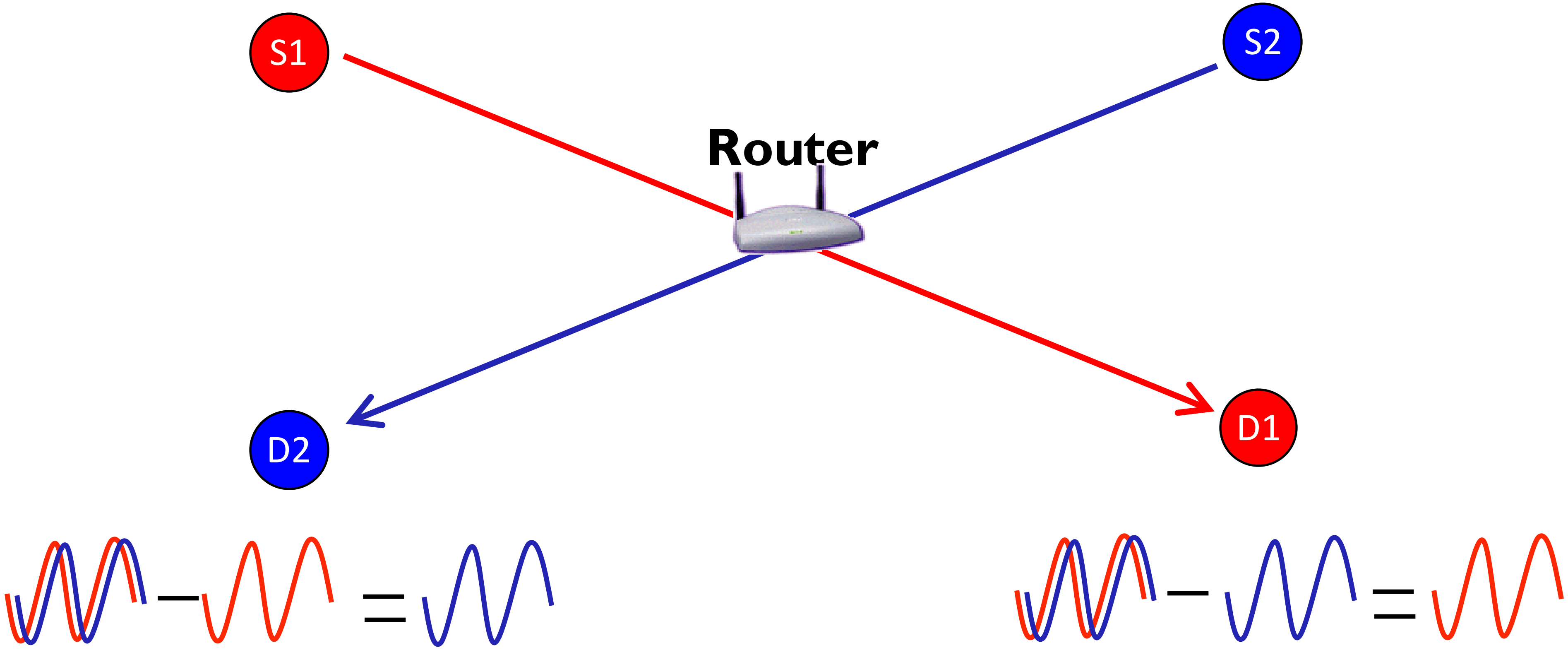
X topology



X topology



X topology



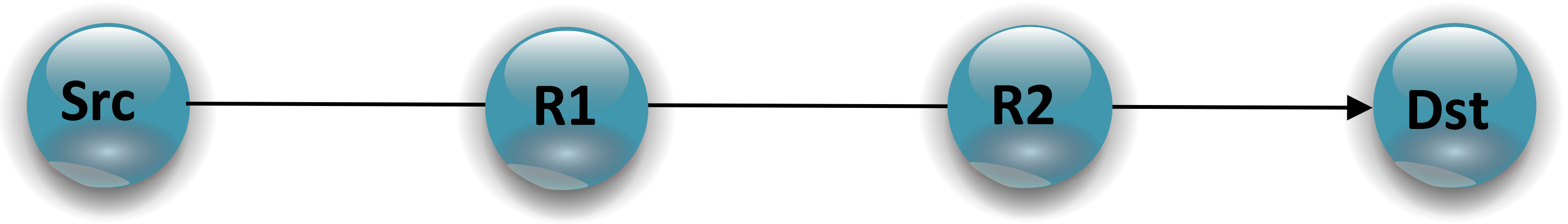
ANC decodes interference using overheard signals

It Is More Than Going From 3 To 2!

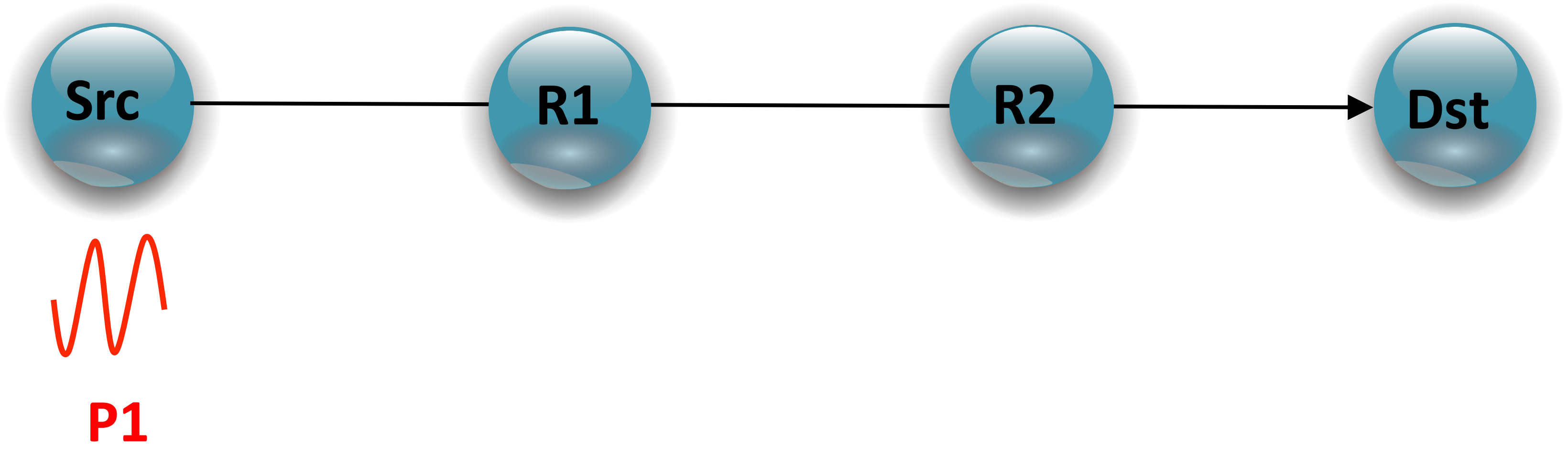
- Philosophical shift in dealing with interference
  - Strategically exploit interference instead of avoiding it

- Promises new ways of dealing with hidden terminals

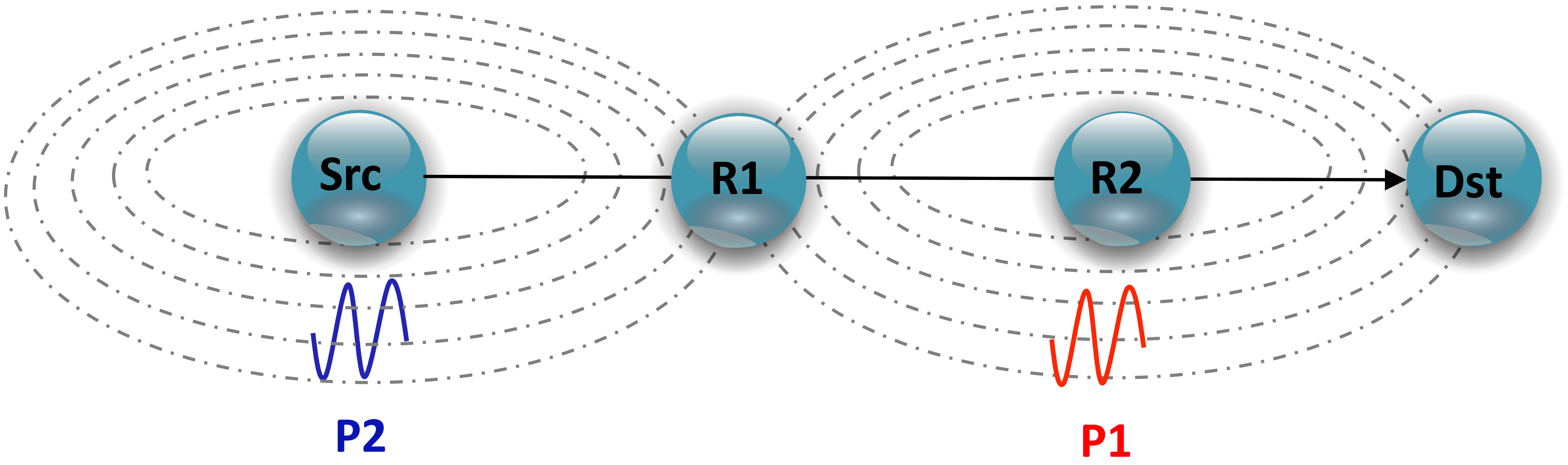
Hidden Terminal Scenario



Hidden Terminal Scenario

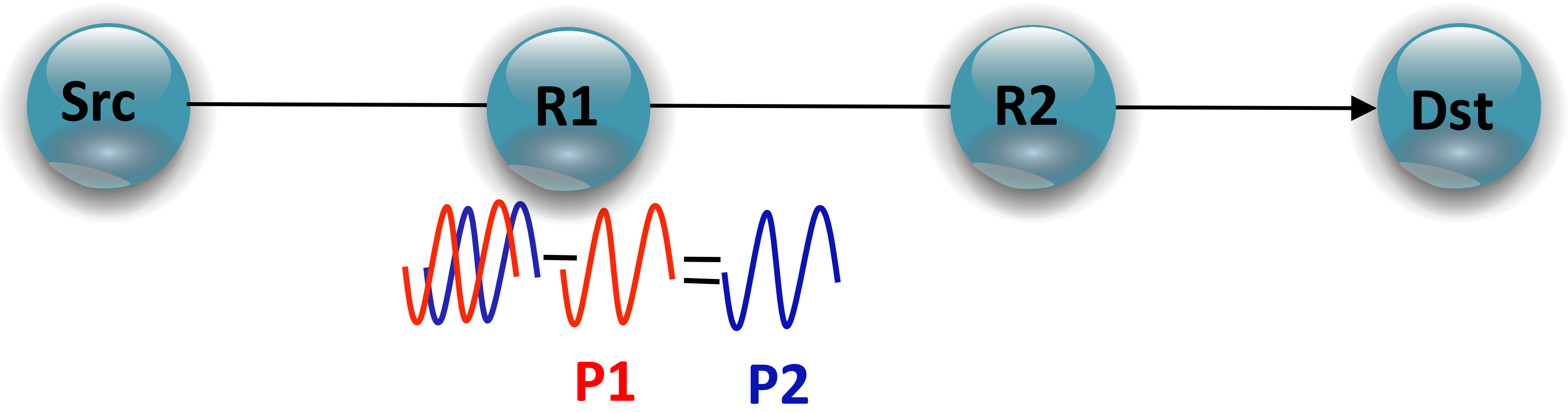


Hidden Terminal Scenario



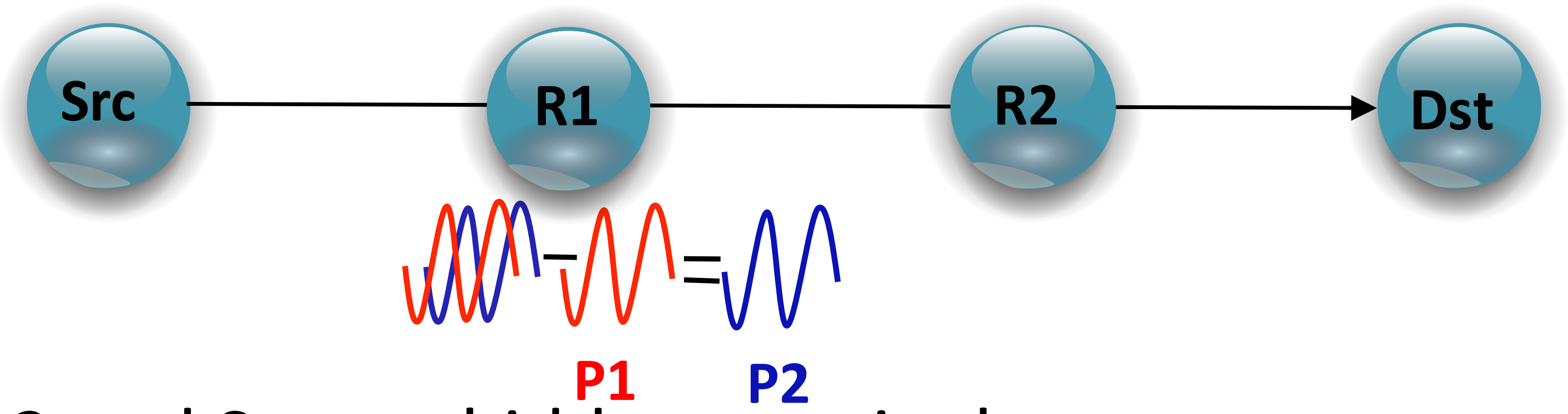
1) Src and R2 transmit simultaneously

Hidden Terminal Scenario



1) Src and R2 transmit simultaneously
2) R1 subtracts P1, which he relayed earlier to recover P2 that he wants

Hidden Terminal Scenario



R2 and Src are hidden terminals

- Today : Simultaneous transmission → Collision
- ANC : Simultaneous transmission → Success!