Network Coding With Wireless Applications

William Wu[1]

[1]Department of Electrical Engineering
Information Systems Laboratory
Stanford University

Motivation

- Network coding is a theory for communicating information across networks more efficiently.

Motivation

- Network coding is a theory for communicating information across networks more efficiently.
- Although it has been around since the year 2000, there still isn't a single deployed product that uses it.

Motivation

- Network coding is a theory for communicating information across networks more efficiently.
- Although it has been around since the year 2000, there still isn't a single deployed product that uses it.
- **Question:**

Motivation

- Network coding is a theory for communicating information across networks more efficiently.
- Although it has been around since the year 2000, there still isn't a single deployed product that uses it.
- **Question:**
  - Is network coding only an impractical theory?

Motivation

- Network coding is a theory for communicating information across networks more efficiently.
- Although it has been around since the year 2000, there still isn't a single deployed product that uses it.
- **Question:**
  - Is network coding only an impractical theory?
  - Or does industry just need to learn about it?

Motivation

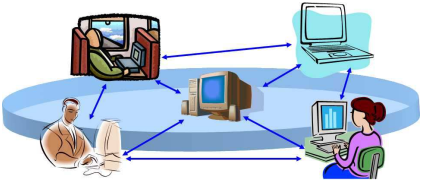
- Network coding is a theory for communicating information across networks more efficiently.
- Although it has been around since the year 2000, there still isn't a single deployed product that uses it.
- **Question:**
  - Is network coding only an impractical theory?
  - Or does industry just need to learn about it?
- **Goal of Talk:**

  Understand the main results in network coding to date.

Outline of Talk

- What is Network Coding?
- Theory of Network Coding
- Wireless Applications
- Conclusion

What is Network Coding?



- **Network coding is a strategy for sending data across a communication network.**
- Instead of forwarding the data, we **transform** it along the way.
- This allows us to **communicate more efficiently**!

Preliminaries: The XOR Operator

Preliminaries: The XOR Operator

- In network coding, we often like to transform data by using the **"XOR"** operator, denoted by $\oplus$.
- XOR is a binary operator that takes two bits as input, and returns one bit as output, as defined by this truth table:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Preliminaries: The XOR Operator

- In network coding, we often like to transform data by using the **"XOR"** operator, denoted by $\oplus$.
- XOR is a binary operator that takes two bits as input, and returns one bit as output, as defined by this truth table:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

- Summary: (Different inputs) $\Rightarrow$ 1. (Same inputs) $\Rightarrow$ 0.

- XOR naturally extends in a pairwise fashion to vectors of bits:

$$\begin{aligned} \mathbf{a} &= 010110 \\ \mathbf{b} &= 111011 \\ \mathbf{a} \oplus \mathbf{b} &= 101101 \end{aligned}$$

- XOR naturally extends in a pairwise fashion to vectors of bits:

$$\begin{aligned} \mathbf{a} &= 010110 \\ \mathbf{b} &= 111011 \\ \mathbf{a} \oplus \mathbf{b} &= 101101 \end{aligned}$$

- Fact: $\mathbf{a} \oplus (\mathbf{a} \oplus \mathbf{b}) = (\mathbf{a} \oplus \mathbf{a}) \oplus \mathbf{b} = 0 \oplus \mathbf{b} = \mathbf{b}$.

- XOR naturally extends in a pairwise fashion to vectors of bits:

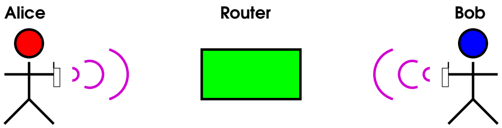
$$\begin{aligned} \mathbf{a} &= 010110 \\ \mathbf{b} &= 111011 \\ \mathbf{a} \oplus \mathbf{b} &= 101101 \end{aligned}$$

- Fact: $\mathbf{a} \oplus (\mathbf{a} \oplus \mathbf{b}) = (\mathbf{a} \oplus \mathbf{a}) \oplus \mathbf{b} = 0 \oplus \mathbf{b} = \mathbf{b}$.

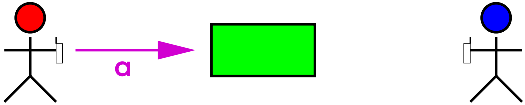
- **Main Point:**

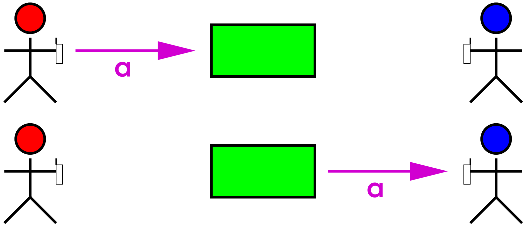
If I know **a**, and someone gives me $\mathbf{a} \oplus \mathbf{b}$, *I can decode* **b**.

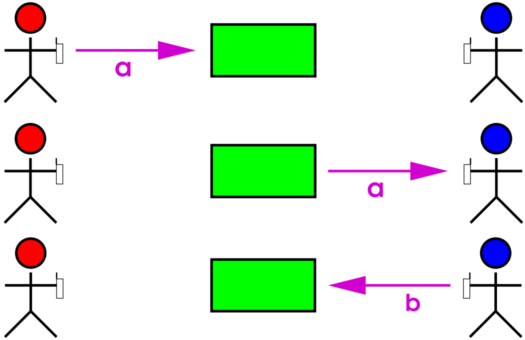
Wireless Exchange

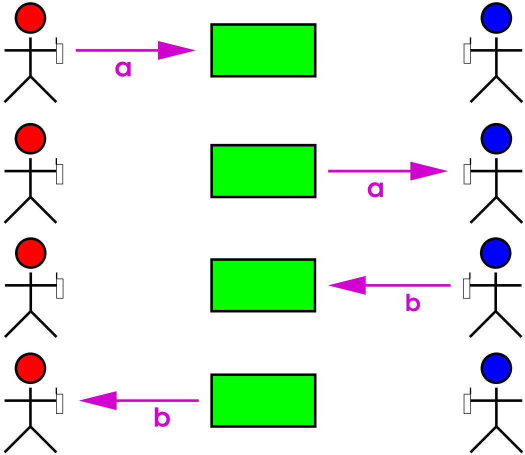


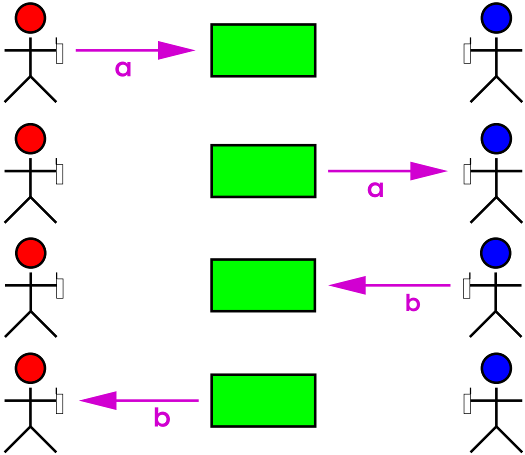
- Alice and Bob are wireless users.
- Alice wants to send message **a** to Bob.
- Bob wants to send message **b** to Alice.
- Because Alice and Bob are too far away from each other, they must send their messages to a router.


a

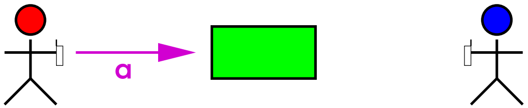

a
a

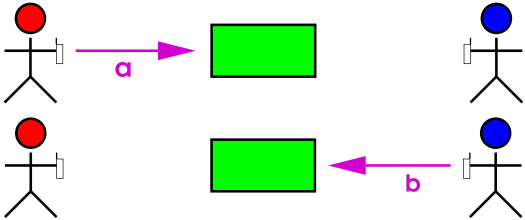

a
a
b

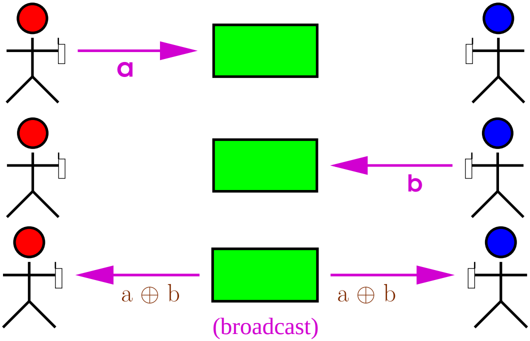

a
a
b
b



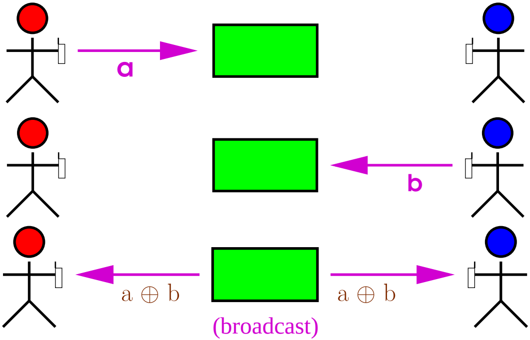
Number of Transmissions = 4. Can we do better?


a

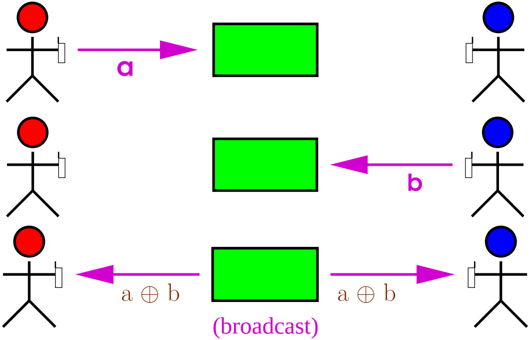

a
b



- Network coding at the router: broadcast **a** $\oplus$ **b**.

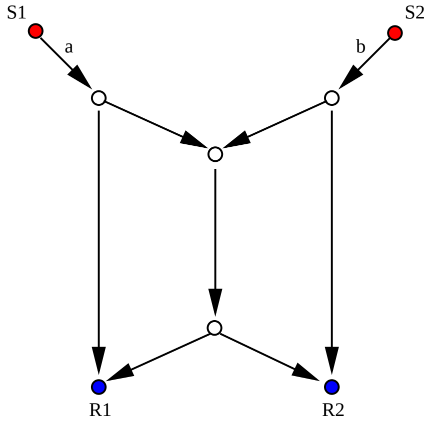


- Network coding at the router: broadcast $\mathbf{a} \oplus \mathbf{b}$.
- Alice knows **a** already (she sent it!).
- So Alice can decode $\mathbf{b} = \mathbf{a} \oplus (\mathbf{a} \oplus \mathbf{b})$. Similarly for Bob.



- Network coding at the router: broadcast $\mathbf{a} \oplus \mathbf{b}$.
- Alice knows **a** already (she sent it!).
- So Alice can decode $\mathbf{b} = \mathbf{a} \oplus (\mathbf{a} \oplus \mathbf{b})$. Similarly for Bob.
- Number of Transmissions = 3.

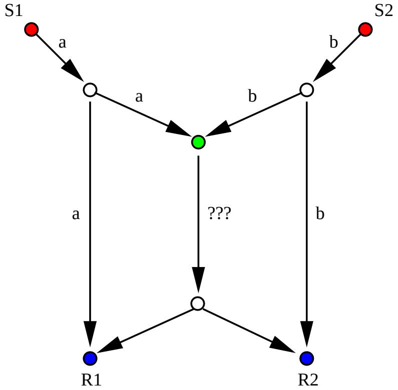
Multicast: Butterfly



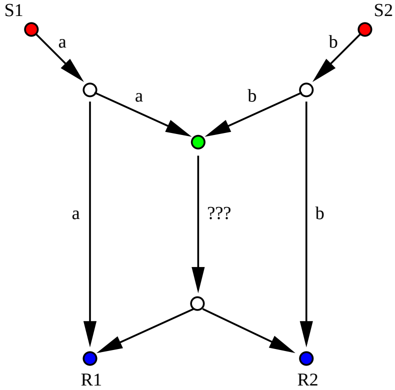
- Senders: $S1, S2$
- Receivers: $R1, R2$
- Multicasting:
  - $S1$ wants to send **a** to *both* receivers.
  - $S2$ wants to send **b** to *both* receivers.
- Every edge in the communication network has the same capacity.

The immediate recipients can do nothing but forward the data.

The immediate recipients can do nothing but forward the data.

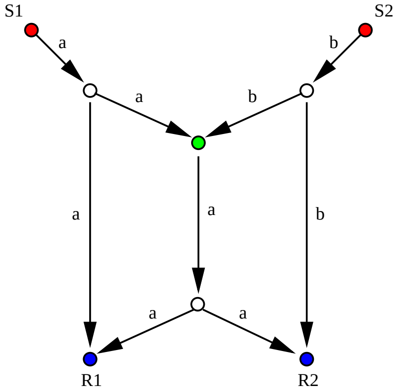


The immediate recipients can do nothing but forward the data.

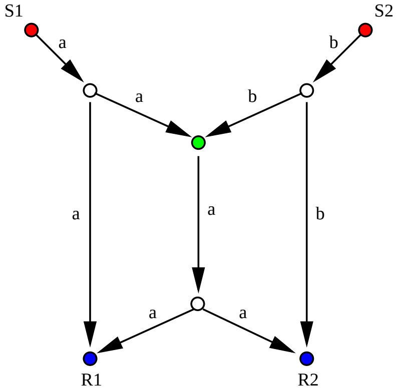


But what should the green node do? (ask audience)

Suppose he forwards one of the two messages; let's say **a** . . .

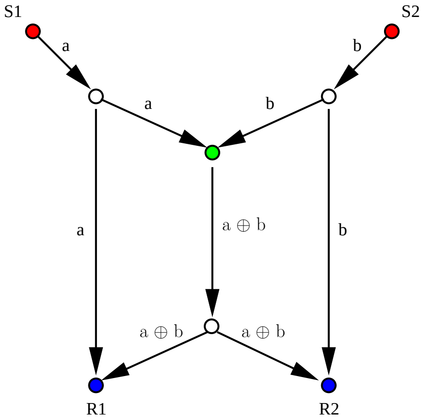


Suppose he forwards one of the two messages; let's say **a** . . .

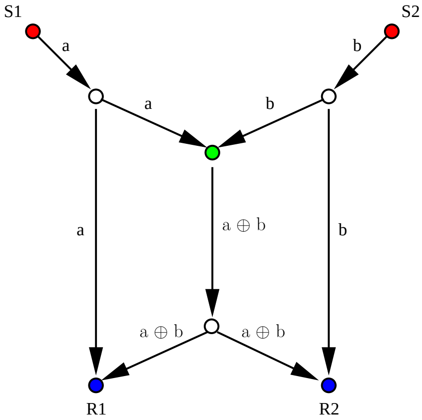


Then $R2$ receives both **a** and **b**, but $R1$ only receives **a**.

- Better idea: Use XOR to *mix* the information.



- Better idea: Use XOR to *mix* the information.
- $R1$ receives **a** and $\mathbf{a} \oplus \mathbf{b}$. Decode $\mathbf{b} = \mathbf{a} \oplus (a \oplus b)$.
- $R2$ receives **b** and $\mathbf{a} \oplus \mathbf{b}$. Decode $\mathbf{a} = (\mathbf{a} \oplus \mathbf{b}) \oplus \mathbf{b}$.



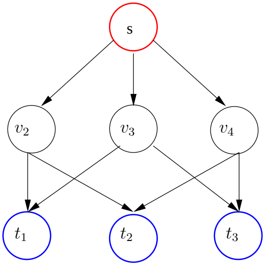
- Better idea: Use XOR to *mix* the information.
- $R1$ receives **a** and $\mathbf{a} \oplus \mathbf{b}$. Decode $\mathbf{b} = \mathbf{a} \oplus (a \oplus b)$.
- $R2$ receives **b** and $\mathbf{a} \oplus \mathbf{b}$. Decode $\mathbf{a} = (\mathbf{a} \oplus \mathbf{b}) \oplus \mathbf{b}$.
- Both get two messages! Network coding increases **capacity**.

The Key Idea

Key Idea of Network Coding

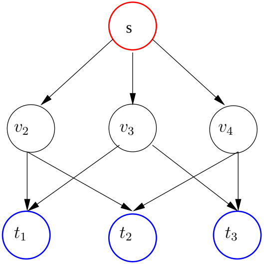
- Information is not a physical commodity!
- We don't have to keep it in its original packaging. (*routing*)
- Sometimes we should open the package and change it! (*network coding*)

Multicast: 3-ary Graph



- Node s wants to send the *same set of messages* to three different receivers t_1, t_2, and t_3. (This is called "multicast".) Every edge has the same capacity.

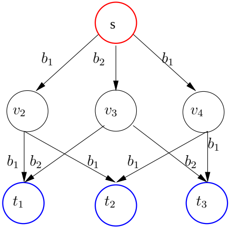
Multicast: 3-ary Graph



- Node s wants to send the *same set of messages* to three different receivers t_1, t_2, and t_3. (This is called "multicast".) Every edge has the same capacity.
- How many different messages can s send simultaneously? (ask audience)

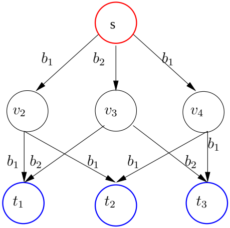
- Routing cannot even multicast two messages. (Why?)

- Routing cannot even multicast two messages. (Why?)



routing

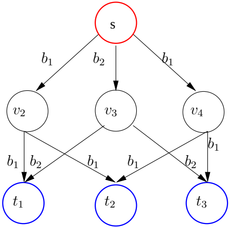
- Routing cannot even multicast two messages. (Why?)



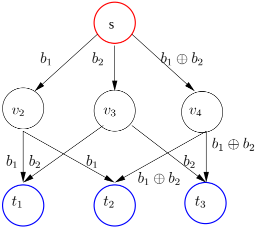
routing

- Solution: Use coding before and after a relay.

- Routing cannot even multicast two messages. (Why?)
- Solution: Use coding before and after a relay.

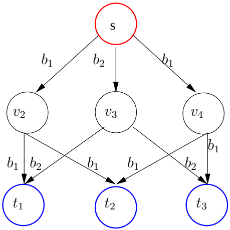


routing

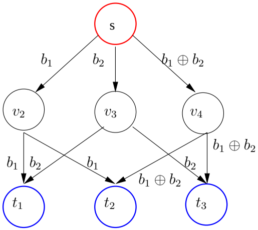


network coding

- Routing cannot even multicast two messages. (Why?)
- Solution: Use coding before and after a relay.



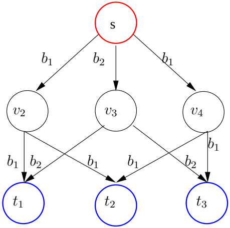
routing



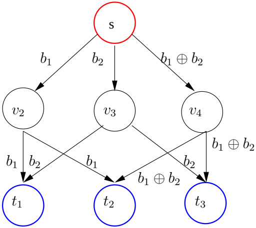
network coding

- This was harder than the previous problems.

- Routing cannot even multicast two messages. (Why?)
- Solution: Use coding before and after a relay.



routing



network coding

- This was harder than the previous problems.
- How do we know that we cannot send three messages?

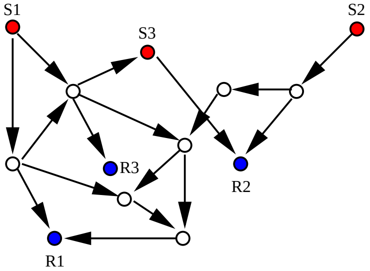
Natural Questions

Natural Questions

- Given a network, what is the most information we can send?

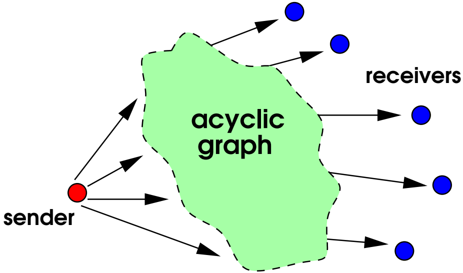
Natural Questions

- Given a network, what is the most information we can send?
- How can we do network coding on a complex network?

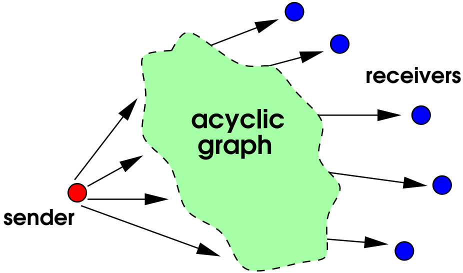


(XORs aren't good enough ...)

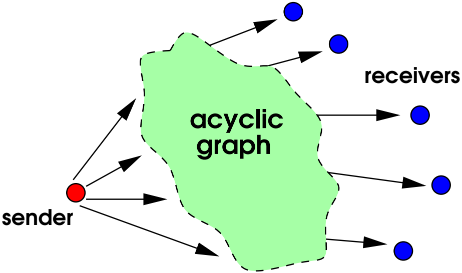
- Satisfying answers to these questions are available for **one sender multicasting on an acyclic graph.**



- Satisfying answers to these questions are available for **one sender multicasting on an acyclic graph.**
- **Extension:** *Many senders multicasting to the same receivers* is just like having only one sender. (Why?)

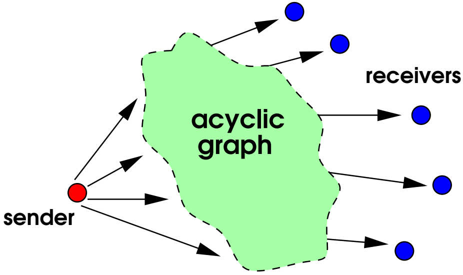


- Satisfying answers to these questions are available for **one sender multicasting on an acyclic graph.**
- **Extension:** *Many senders multicasting to the same receivers* is just like having only one sender. (Why?)
- Other scenarios are open problems.



- Satisfying answers to these questions are available for **one sender multicasting on an acyclic graph.**

- **Extension:** *Many senders multicasting to the same receivers* is just like having only one sender. (Why?)

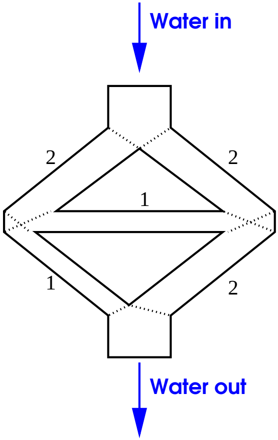


- Other scenarios are open problems.

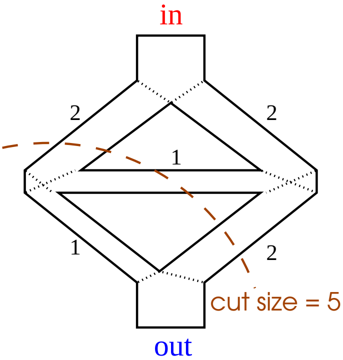
- To understand the existing answers, consider flowing water . . .

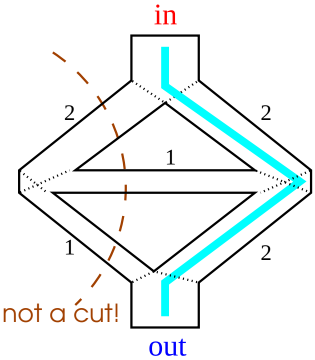
Preliminaries: Max-Flow Min-Cut Theorem

- Consider a network of water pipes. There is a single input pipe, and a single output pipe.
- Every pipe has a certain flow capacity that it can support (e.g., 2 gal/sec).
- **Question:** *What is the maximum water flow between input and output?*



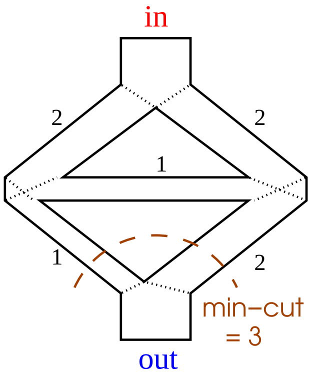
- **Definition:** A **cut** is a set of pipes that together completely separate the input and output.
- **Definition:** The **size** of a cut is the sum of the capacities of all the pipes in the cut.





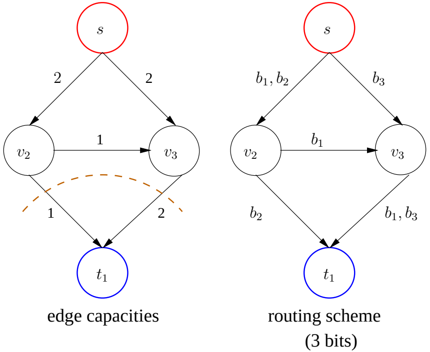
- **Ford-Fulkerson Max-Flow Min-Cut Theorem:**

The maximum flow is equal to the size of the smallest cut.

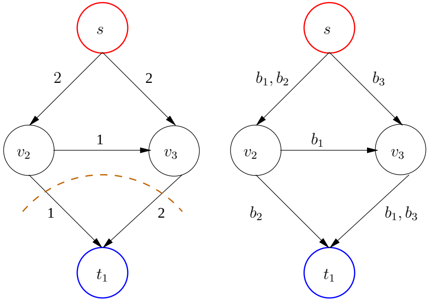


- The smallest cut is called the "min-cut".

- This result extends to information transfer!



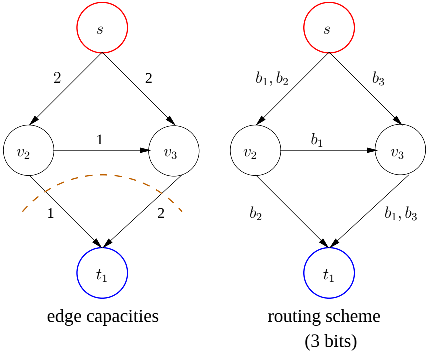
- This result extends to information transfer!
- **New Question:** Given a graph, what is the maximum number of bits we can route from s to t?



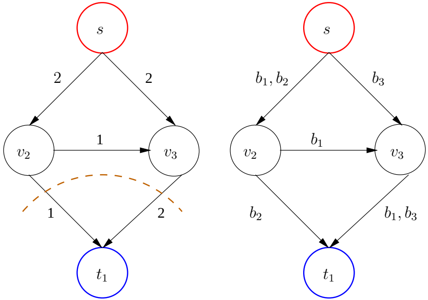
edge capacities

routing scheme (3 bits)

- This result extends to information transfer!
- **New Question:** Given a graph, what is the maximum number of bits we can route from s to t?
- Answer: The size of the min-cut. (Why?)



- This result extends to information transfer!
- **New Question:** Given a graph, what is the maximum number of bits we can route from s to t?
- Answer: The size of the min-cut. (Why?)



edge capacities
routing scheme (3 bits)

Full solution to one-sender one-receiver problem.
Ford-Fulkerson routing achieves optimal throughput.

Multicasting Problem Statement

Now let's look at one sender and multiple receivers.

Multicasting Problem Statement

Now let's look at one sender and multiple receivers.

- **Given:** A graph $G = (V, E, w)$, where
  - V is the set of nodes,
  - E is the set of edges, and
  - w is a mapping s.t. for $e \in E$, $w(e)$ is the bitrate capacity of e.

Multicasting Problem Statement

Now let's look at one sender and multiple receivers.

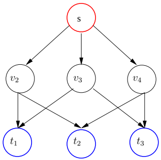
- **Given:** A graph $G = (V, E, w)$, where
  - V is the set of nodes,
  - E is the set of edges, and
  - w is a mapping s.t. for $e \in E$, $w(e)$ is the bitrate capacity of e.
- **Problem 1 (Multicast Rate):** Find maximum number of "symbols" h that node $s \in V$ can simultaneously send to a set of receivers $T \triangleq \{t_1, t_2, \ldots, t_n\} \subset V$, such that every receiver can decode the same h symbols.

Multicasting Problem Statement

Now let's look at one sender and multiple receivers.

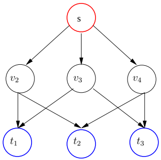
- **Given:** A graph $G = (V, E, w)$, where
  - V is the set of nodes,
  - E is the set of edges, and
  - w is a mapping s.t. for $e \in E$, $w(e)$ is the bitrate capacity of e.
- **Problem 1 (Multicast Rate):** Find maximum number of "symbols" h that node $s \in V$ can simultaneously send to a set of receivers $T \triangleq \{t_1, t_2, \ldots, t_n\} \subset V$, such that every receiver can decode the same h symbols.
- **Problem 2 (Code):** Find the routing/coding scheme which achieves the maximum rate.

Example: 3-ary Multicast, Again



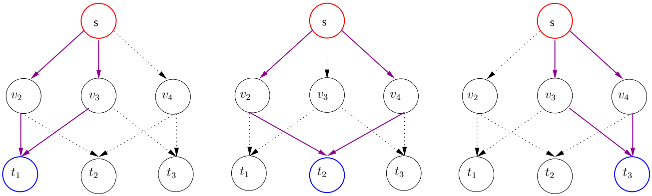
- One sender s, and three receivers $T \triangleq \{t_1, t_2, t_3\}$.

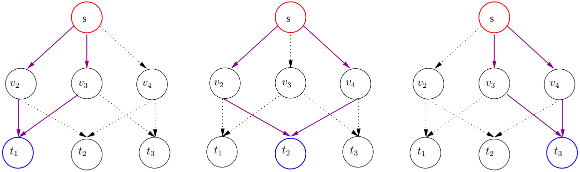
Example: 3-ary Multicast, Again



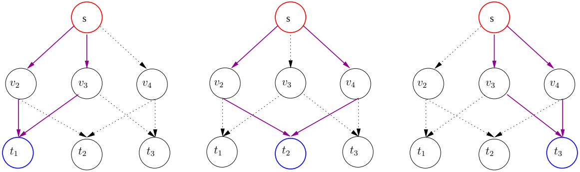
- One sender s, and three receivers $T \triangleq \{t_1, t_2, t_3\}$.

- For each $t \in T$, define the "subgraph" G_t to be the graph consisting of all paths which run from s to t.

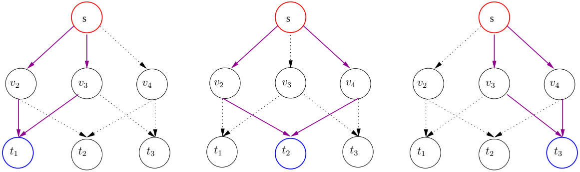




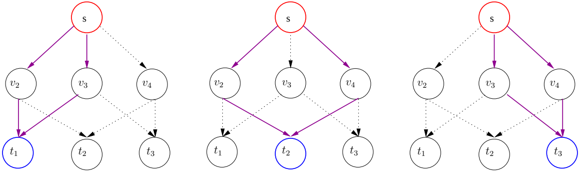
- **Receiver's Perspective:** "If I were the only receiver, then s ought to send me data at rate $MINCUT(G_t)$."



- **Receiver's Perspective:** "If I were the only receiver, then s ought to send me data at rate $MINCUT(G_t)$."
- **Sender's Perspective:** "I cannot multicast at a rate higher than $\min_t MINCUT(G_t)$." (Why?)



- **Receiver's Perspective:** "If I were the only receiver, then s ought to send me data at rate $MINCUT(G_t)$."
- **Sender's Perspective:** "I cannot multicast at a rate higher than $\min_t MINCUT(G_t)$." (Why?)
- **Graph's Perspective:** "Subgraphs overlap, so if you hope to multicast at rate $\min_t MINCUT(G_t)$, you need coding!"



- **Receiver's Perspective:** "If I were the only receiver, then s ought to send me data at rate $MINCUT(G_t)$."
- **Sender's Perspective:** "I cannot multicast at a rate higher than $\min_t MINCUT(G_t)$." (Why?)
- **Graph's Perspective:** "Subgraphs overlap, so if you hope to multicast at rate $\min_t MINCUT(G_t)$, you need coding!"
- **The Theorem**: $MAXRATE = \min_t MINCUT(G_t)$.

Main Theorem of Network Coding

Main Theorem of (Multicast) Network Coding

Let G_t be the subgraph between s and $t \in T$. Then $MINCUT(s \to t)$ is the min-cut between s and t in G_t. Then, the *maximum reliable multicast rate* is:

$$MAXRATE = \min_{t \in T} MINCUT(s \to t)$$

This rate can be achieved with *linear codes* which can be found in polynomial time $O(|E| \cdot |T| \cdot (h^2 + |T|^2))$.

How To Find The Code?

1. **Key Idea:** With every edge $e_{ij} \in E$, we will associate a vector $\mathbf{b}(e_{ij})$ representing the information on that edge.

How To Find The Code?

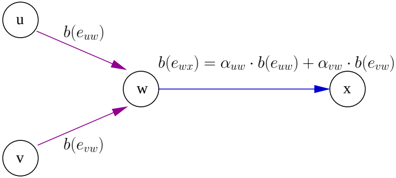
1. **Key Idea:** With every edge $e_{ij} \in E$, we will associate a vector $\mathbf{b}(e_{ij})$ representing the information on that edge.
2. Find the maximum symbol rate $h \triangleq \min_{t \in T} MINCUT(s \rightarrow t)$.

How To Find The Code?

1. **Key Idea:** With every edge $e_{ij} \in E$, we will associate a vector $\mathbf{b}(e_{ij})$ representing the information on that edge.
2. Find the maximum symbol rate $h \triangleq \min_{t \in T} MINCUT(s \rightarrow t)$.
3. Represent each of the h symbols generated at s by unit vectors:

$$\mathbf{b}(e_i) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbf{b}(e_2) = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \mathbf{b}(e_h) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

4. Linear Coding

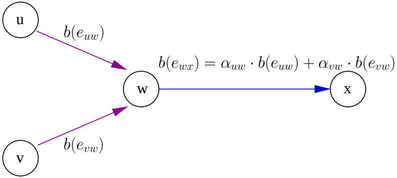


- $\mathbf{b}(e_{ij})$ is a *random linear combination* of information received from incoming edges $\mathbf{b}(e_{ki})$:

$$\mathbf{b}(e_{ij}) = \sum_{e_{ki} \in E} \alpha_e(e_{ki})\mathbf{b}(e_{ki})$$

where $\alpha_e(e_{ki})$ are drawn randomly from a field (set) $\mathcal{F}$.

4. **Linear Coding**



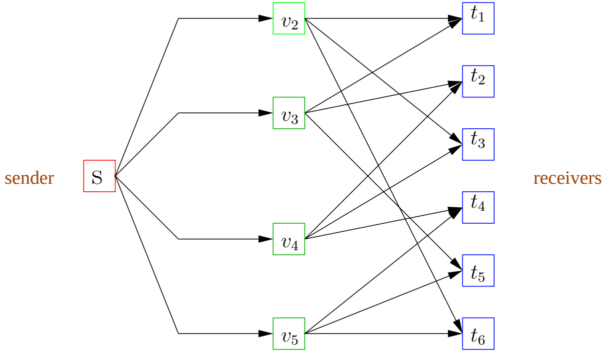
- $\mathbf{b}(e_{ij})$ is a *random linear combination* of information received from incoming edges $\mathbf{b}(e_{ki})$:

$$\mathbf{b}(e_{ij}) = \sum_{e_{ki} \in E} \alpha_e(e_{ki})\mathbf{b}(e_{ki})$$

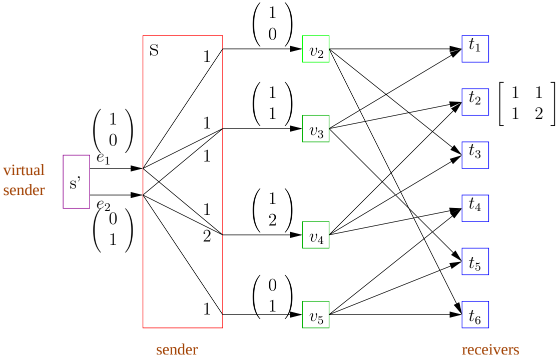
  where $\alpha_e(e_{ki})$ are drawn randomly from a field (set) $\mathcal{F}$.

5. If $|\mathcal{F}| >> |T|$, we will successfully multicast at rate h with high probability.

Example



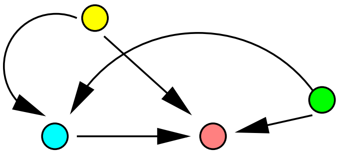
- The min-cut of each sender-to-receiver subgraph is 2.
- So $h = 2$.



- Introduce a virtual sender s' which supplies the symbols.
- Our code can multicast if and only if for every receiver t, the determinant of the matrix of vectors entering t is nonzero.

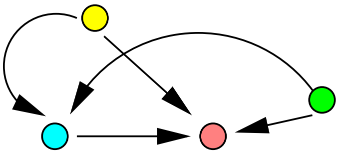
Toward Reality

- We have been looking at networks which are
  - noiseless
  - have clearly defined communication links.

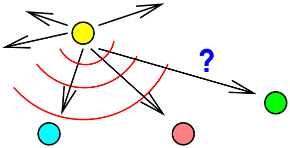


Toward Reality

- We have been looking at networks which are
  - noiseless
  - have clearly defined communication links.



- Yet, real wireless networks
  - have noisy links
  - are broadcast in nature (unintended listeners).

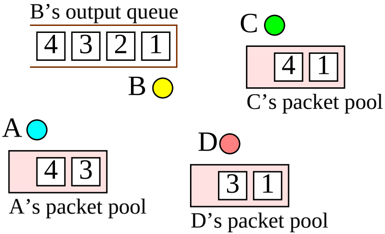


The Key Idea

Key Idea of Wireless Network Coding

- In wireless networks,
  - information is always broadcast to many users, and
  - information can be lost.
- Therefore,
  - Sometimes Alice will hear something that Bob didn't.
  - Sometimes Bob will hear something that Alice didn't.
- Network coding can exploit this *diversity*!
- The wireless channel is naturally suited for network coding.

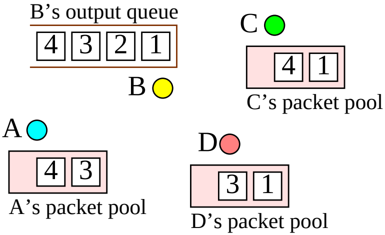
"Coding Opportunistically" (COPE)



packet in B's queue		next hop
1	$\longrightarrow$	A
2	$\longrightarrow$	C
3	$\longrightarrow$	C
4	$\longrightarrow$	D

coding option	who can decode

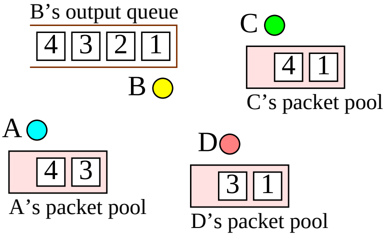
"Coding Opportunistically" (COPE)



packet in B's queue		next hop
1	→	A
2	→	C
3	→	C
4	→	D

coding option	who can decode
1 $\oplus$ 2	C

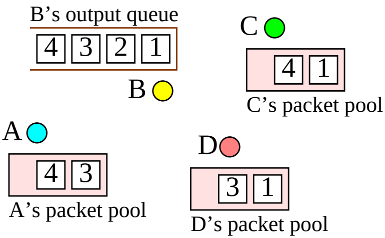
"Coding Opportunistically" (COPE)



packet in B's queue		next hop
1	→	A
2	→	C
3	→	C
4	→	D

coding option	who can decode
1 ⊕ 2	C
1 ⊕ 3	A,C

"Coding Opportunistically" (COPE)



packet in B's queue		next hop
1	→	A
2	→	C
3	→	C
4	→	D

coding option	who can decode
1 ⊕ 2	C
1 ⊕ 3	A,C
1 ⊕ 3 ⊕ 4	A,C,D

Framework of COPE

- Opportunistic Listening
  - Set all nodes to *promiscuous* mode.
  - Everyone *records* what they have heard for a while.
  - Send *reception reports* stating what you have heard.

Framework of COPE

- Opportunistic Listening
  - Set all nodes to *promiscuous* mode.
  - Everyone *records* what they have heard for a while.
  - Send *reception reports* stating what you have heard.
- Learning Neighbor State
  - From reception reports and probability modeling, make assumptions about what your neighbors know.

Framework of COPE

- Opportunistic Listening
  - Set all nodes to *promiscuous* mode.
  - Everyone *records* what they have heard for a while.
  - Send *reception reports* stating what you have heard.
- Learning Neighbor State
  - From reception reports and probability modeling, make assumptions about what your neighbors know.
- Opportunistic Coding
  - When sending, XOR together as many packets we can in order to maximize the number of intended receivers who can decode.
  - Never delay packets.

Experimental Results

Fully-implemented 20-node wireless testbed at MIT

- <u>Wireless Ad-Hoc Network</u>

Protocol	Throughput Gain
TCP	2-3% 20-30% when nodes are closely packed
UDP	300-400%

(TCP backs off excessively due to collision-based losses.)

- <u>Wireless Mesh Access</u>

Protocol	Throughput Gain
UDP	5-15% when UL/DL ratio $< 1/2$
UDP	70% when UL/DL ratio $> 1/2$

(higher uplink traffic = more diversity at output queues)

Reliable Broadcast

Sender s broadcasts to receivers $R1$ and $R2$. Packets are lost.

Received by R1	x	2	3	x	5	6	x	8	x
Received by R2	1	2	x	4	x	6	x	8	9

Reliable Broadcast

Sender s broadcasts to receivers $R1$ and $R2$. Packets are lost.

Received by R1	x	2	3	x	5	6	x	8	x
Received by R2	1	2	x	4	x	6	x	8	9

From negative acknowledgements (opposite of ACK), s knows who did not receive what. Use XOR to retransmit efficiently.

Received by R1	x	2	3	x	5	6	x	8	x
Received by R2	1	2	x	4	x	6	x	8	9

In practice, use a combination of FEC and network coding.

Analog Network Coding (ANC)

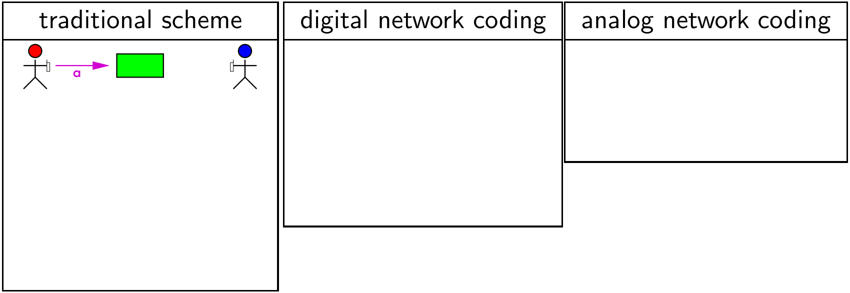
Idea: Increase throughput by letting analog signals collide.

traditional scheme	digital network coding	analog network coding

How can we get away with this?

Analog Network Coding (ANC)

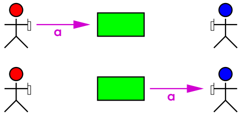
Idea: Increase throughput by letting analog signals collide.



How can we get away with this?

Analog Network Coding (ANC)

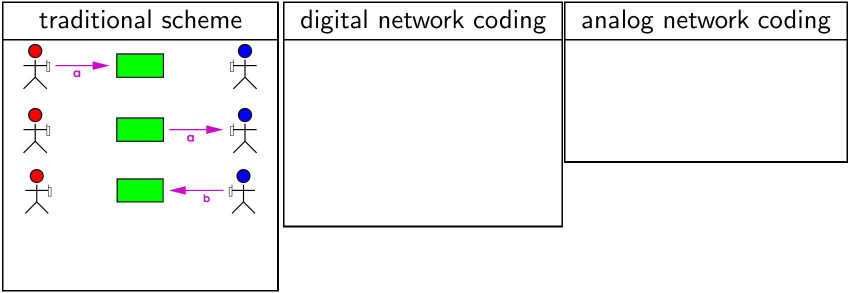
Idea: Increase throughput by letting analog signals collide.

traditional scheme	digital network coding	analog network coding
		

How can we get away with this?

Analog Network Coding (ANC)

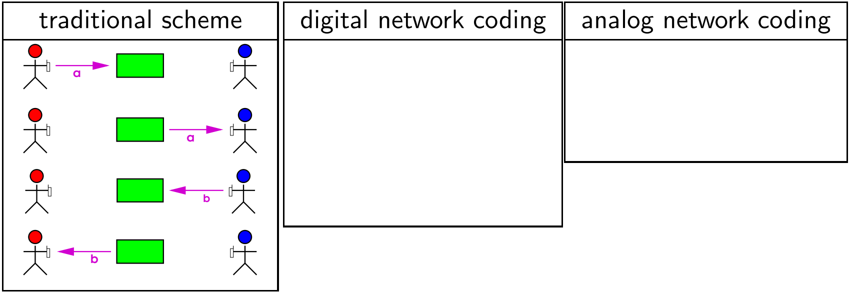
Idea: Increase throughput by letting analog signals collide.



How can we get away with this?

Analog Network Coding (ANC)

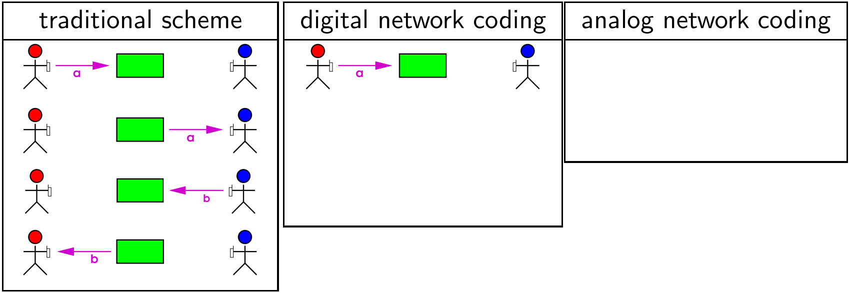
Idea: Increase throughput by letting analog signals collide.



How can we get away with this?

Analog Network Coding (ANC)

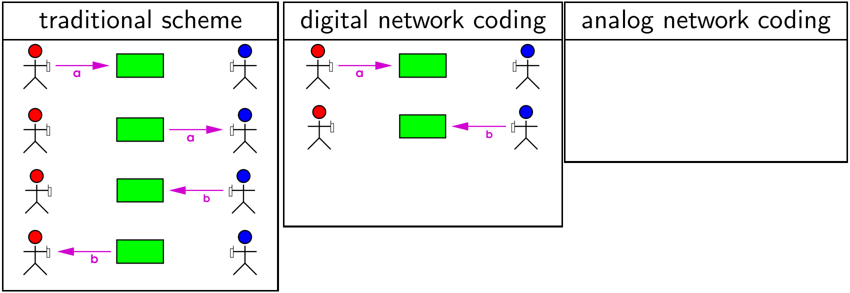
Idea: Increase throughput by letting analog signals collide.



How can we get away with this?

Analog Network Coding (ANC)

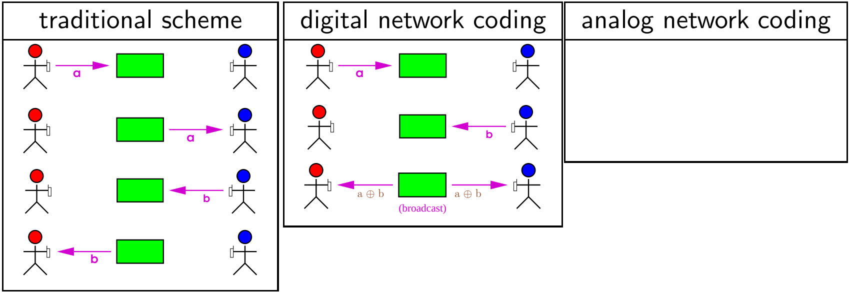
Idea: Increase throughput by letting analog signals collide.



How can we get away with this?

Analog Network Coding (ANC)

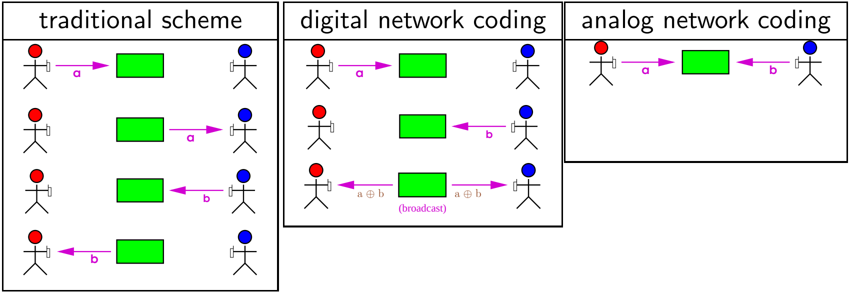
Idea: Increase throughput by letting analog signals collide.



How can we get away with this?

Analog Network Coding (ANC)

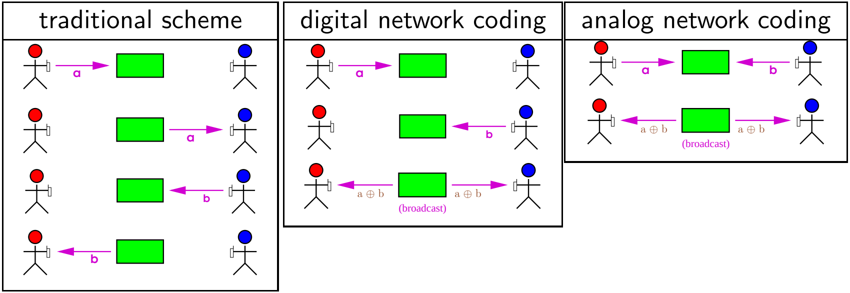
Idea: Increase throughput by letting analog signals collide.



How can we get away with this?

Analog Network Coding (ANC)

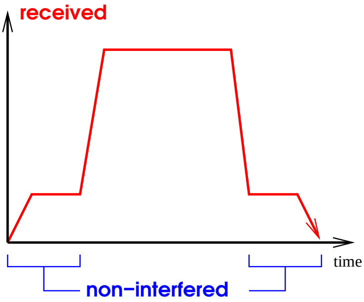
Idea: Increase throughput by letting analog signals collide.



How can we get away with this?

Key Trick:

- The two simultaneously sent signals will not be exactly synchronized.
- By using *MSK modulation*, we can deduce the original signals by analyzing the non-interfered parts of the combined signal.



Result (software radios): Two senders $\Longrightarrow \sim 70\%$ gain

Summary

Summary of Key Ideas

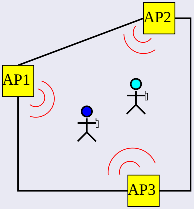
- Information is not a physical commodity. We can transform it at intermediate nodes.
- For multicasting between s and a set of receivers T,

$$MAXRATE = \min_{t \in T} MINCUT(s \rightarrow t).$$

  Achievable with linear codes found in polynomial time.
- The wireless channel is naturally suited for network coding, since there is diversity in the received information.

Further Investigation

- How to use network coding ideas effectively in an indoor Wireless LAN?
  - Wired APs in building
  - Wireless users



- How can we improve on COPE (Coding Opportunistically)?
- New ideas in applying network coding to ad-hoc networks?
- How to best use network coding ideas in unicast scenarios?
- Thanks for listening!
- Comments and collaboration: **willywutang@gmail.com**

References

- R. Ahlswede, N. Cai, S. Li, and R. Yeung, *Network Information Flow*, IEEE Trans. Inform. Theory, IT-46: 1204-1216, 2000.
- P. Sanders, S. Egner, and L.M.G.M. Tolhuizen, *Polynomial time algorithms for network information flow*, Proc. 15-th ACM Symposium on Parallelism in Algorithms and Architectures, pp. 286–294, June 2003.
- S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, *XORs in the air: Practical wireless network coding*, Sigcomm 2006.
- S. Katti, S. Gollakota, D. Katabi, *Embracing Wireless Interference: Analog Network Coding*, Sigcomm 2007.
- D. Nguyen, T. Nguyen, B. Bose, *Wireless Broadcasting Using Network Coding*, submitted to IEEE Transactions on Vehicular Technology, 2007.
- B. Nazer and M. Gastpar, *Computing over Multiple-Access Channels with Connections to Wireless Network Coding*, 2006.