

Deutsche Telekom Chair of Communication Networks  
Technische Universität Dresden

# Advanced Topics in Signal Processing

Frank Fitzek // Summer Semester 2018

# Lecture / Exercise Dates - [goo.gl/BEUp7C](http://goo.gl/BEUp7C)

- Here you find all the information for the lecture and the exercise can be found. Please check every week in case of changes (might happen)
- Slides
- Links



Main lecturer:  
Professor Frank Fitzek



Co-lecturer:  
M.Sc. Juan Cabrera



Co-lecturer:  
Dipl.-Ing. Zuo Xiang

DATE	TYPE	ROOM	TOPIC
13.Apr.2018	Module 1	TBD	Organisation of the course; Motivation for NC and NC use cases (5G, IoT, Wireless Mesh); Butterfly example; min cut max flow. Inter Flow NC. Index Coding. Zick Zack Coding. CATWOMAN
20.Apr.2018	Module 2	TBD	Alice Relay and Bob scenario under IEEE802.11 networks and asymmetric traffic. Analog Inter Flow NC. Random Linear Network Coding (Basics)
04.May.2018	Exercise 1	TBD	Basic UDP transmissions over WIFI with Python. Example of Inter-Session Network coding: XORing packets. Introduction to KODO. Lossless Encoding and Decoding
27.Apr.2018	Module 3	TBD	RLNC advanced 1 (systematic, sparse, and sliding window RLNC)
11.May.2018	Module 4	TBD	KODO overhead and energy consumption. Heterogeneous packet lengths Network coding for transport
18.May.2018	Module 5	TBD	RLNC extensions: Telescopic codes, FULCRUM codes, and the CORE protocol. Network coding for storage
08.Jun.2018	Exercise 2	TBD	Code efficiency. Number of linear dependencies. Where does the linear dependencies occur? Multi-path and Multi-hop with NC
01.Jun.2018	Module 6	TBD	Combination of Network Coding with Source/Channel Coding. Combination of Network Coding and Compressed Sensing

# Aim of this lecture module

Explain network coding in theory and practice

Explain the uniqueness of network coding

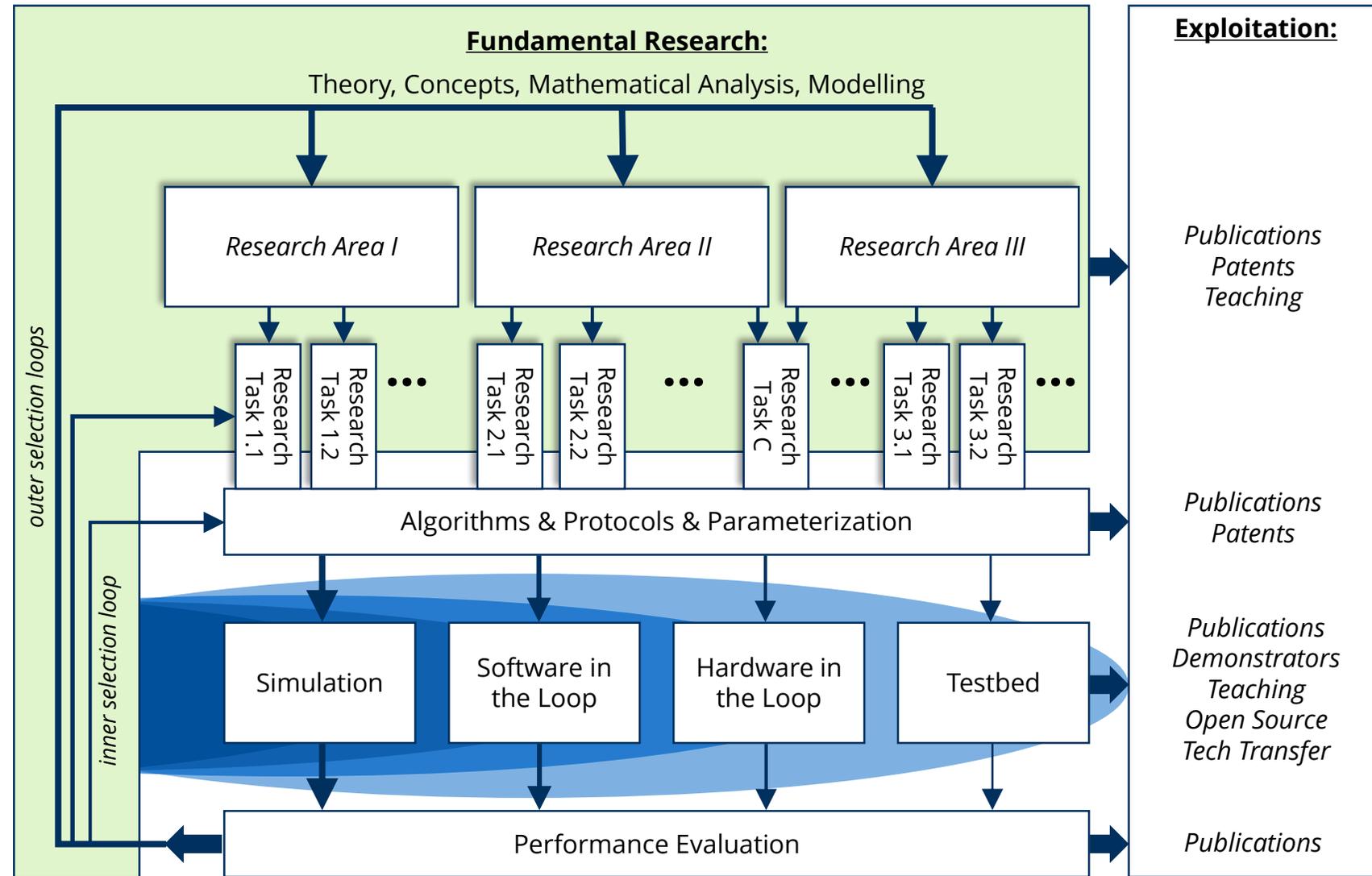
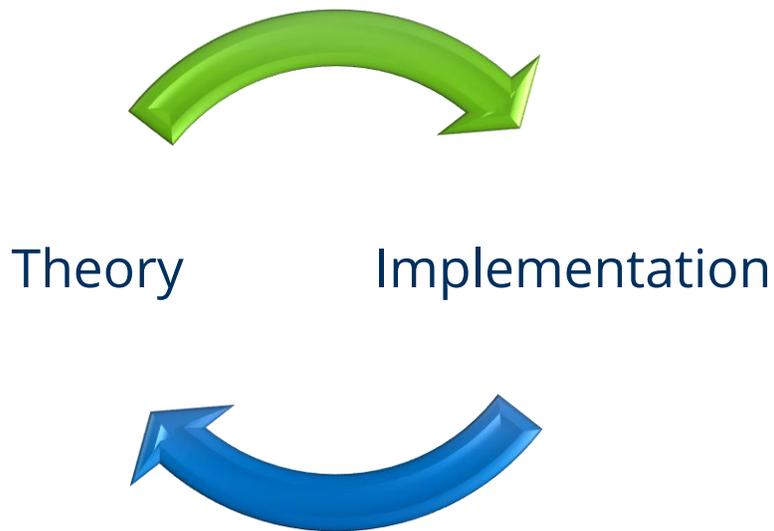
Describe a wide range of applications of NC in current and future communication systems

- 5G
- Storage as well as transportation

Important is the “hands on” parts aligned with the theory

- Please bring your laptop to all lectures and exercises
  - Preinstall software needed
  - Get KODO license from [steinwurf.com/license](https://steinwurf.com/license)

# Research Methodology: Theory that matters!



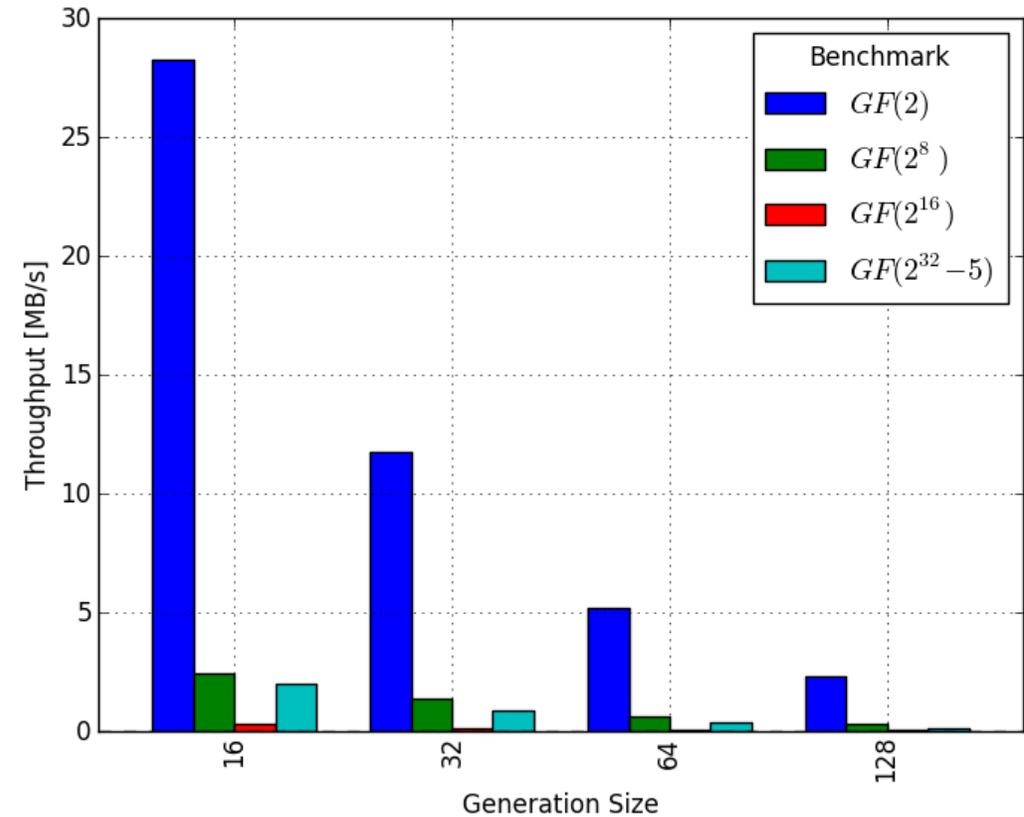
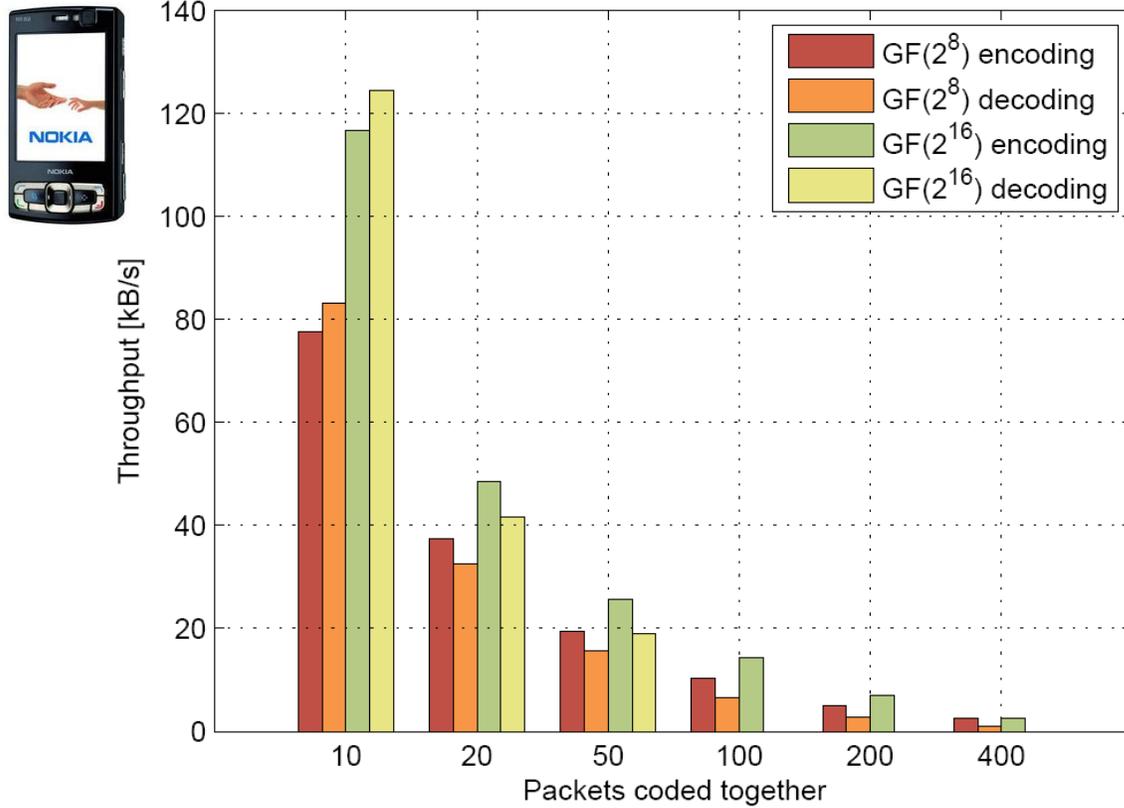
# Research Methodology: Example

2007: 120kB/s



2012: 27 MB/s

Coding and decoding throughput



# How fast are we today?



Kodomark

Steinwurf ApS Libraries & Demo

★★★★★ 6

USK: All ages

This app is compatible with all of your devices.

Installed



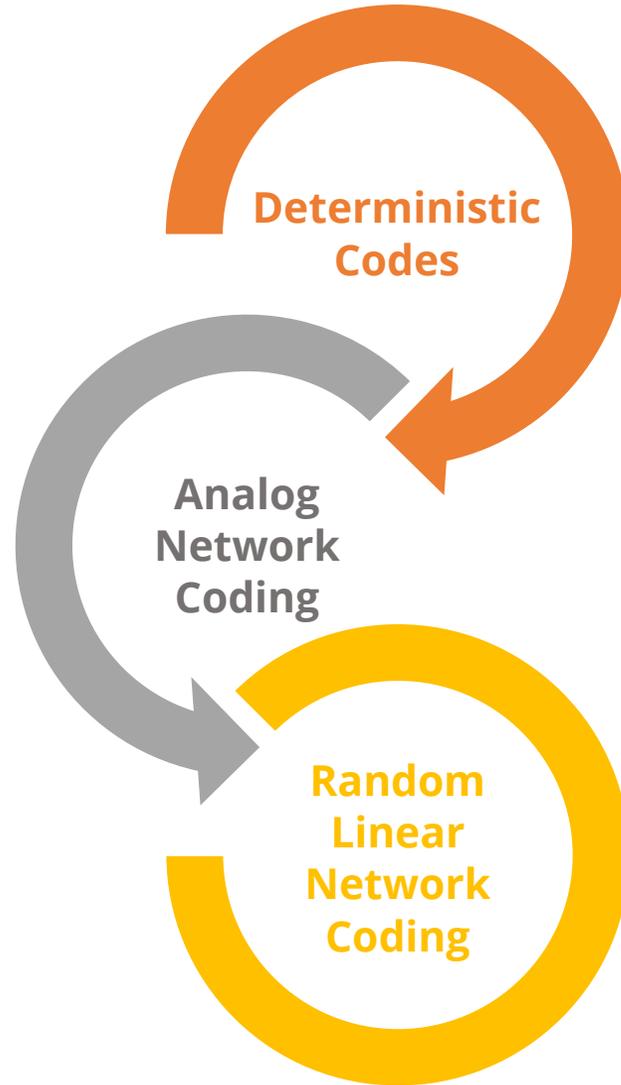
<http://tinyurl.com/z7vsp4c>



Please try it out and support our research! If you have an Android device simply install and press START! Change the parameters to learn about network coding.

# How do we approach NC?

Extreme application of NC! In general the same ideas as before but more gains!



Let's have fun! We play around with some smart ideas!

The real deal! Versatile code for all application fields! Complex but powerful!

# 5G Motivation

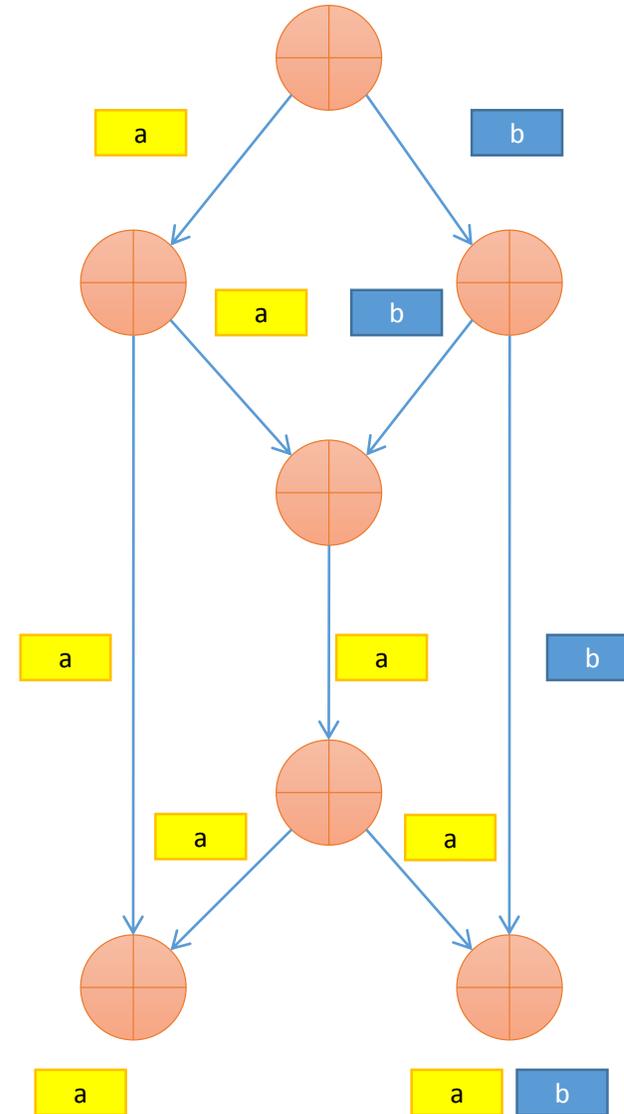
# Digital Inter-Flow Network Coding: The Basics

Lecture 1

# The Butterfly

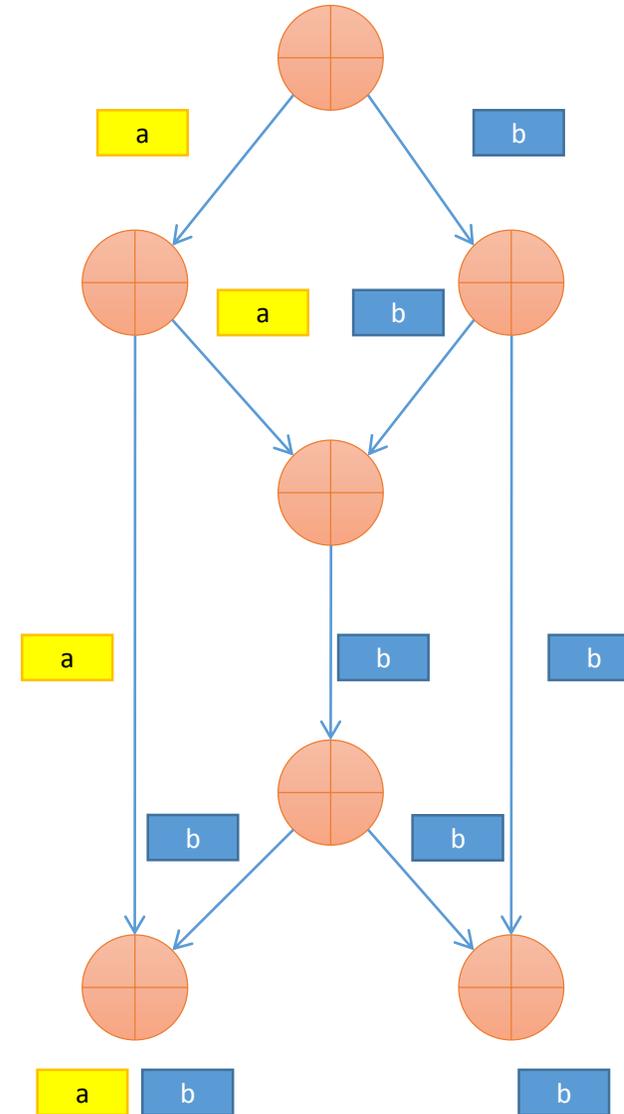
# Network Coding: The Butterfly

- Two packets a and b must be conveyed to two destinations over a given network
- Assumption, capacity per link can handle one packet per time slot
- Bottleneck in the middle
- Either packet a or b will path the bottleneck
- One destination will receive one *unique* packet, the other two packets



# Network Coding: The Butterfly

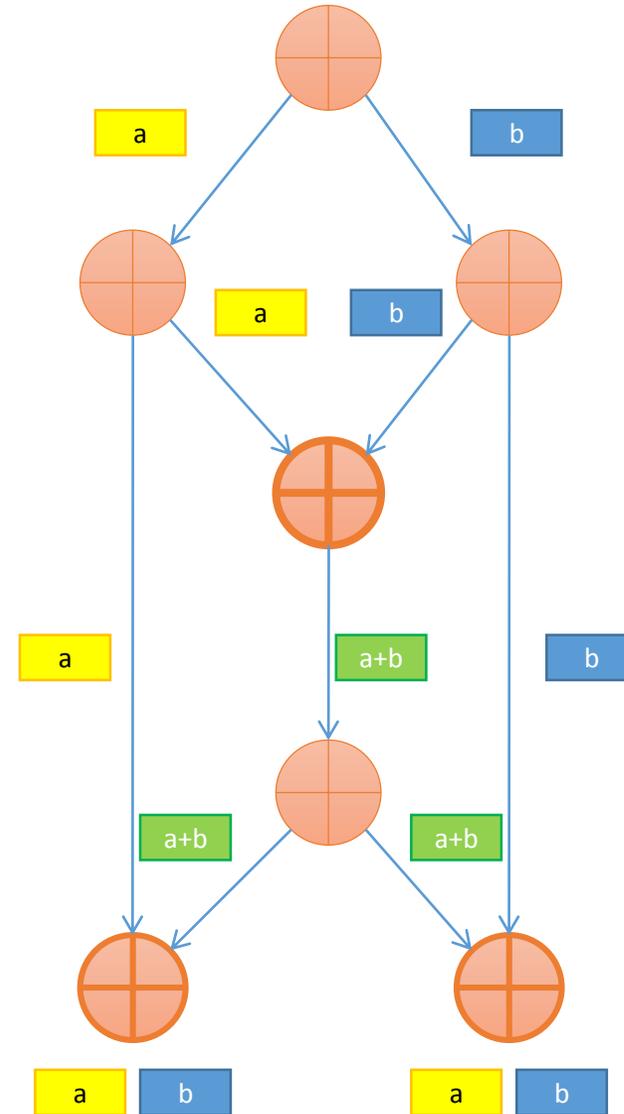
- Let's try b instead of a
- Same old problem



# Network Coding: The Butterfly

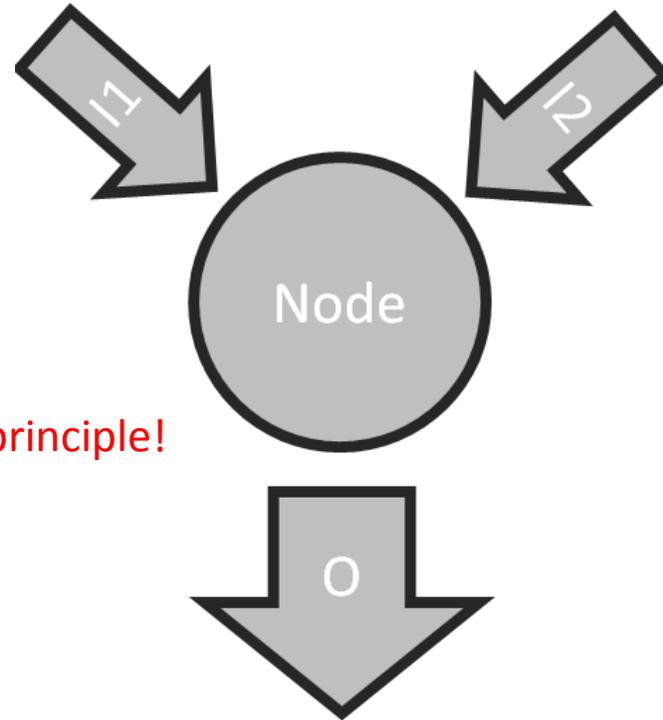
- Ahlswede et. al. In 2000
- Coding the packet
- Other ideas were around
- Max-flow min-cut theorem

Ahlswede, Rudolf; N. Cai, Shuo-Yen Robert Li, and Raymond Wai-Ho Yeung (2000). "Network Information Flow". IEEE Transactions on Information Theory, IT-46 46 (4): 1204–1216.



# Kirchhoff versus Network Coding

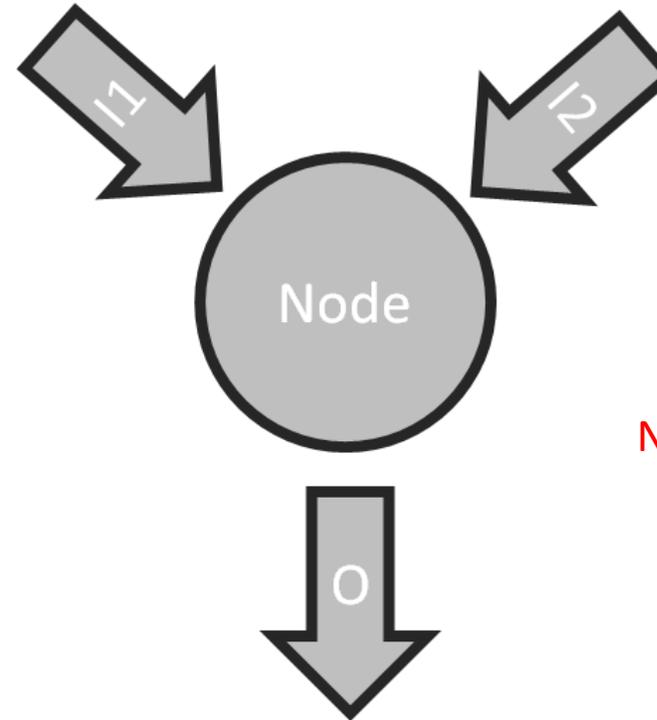
Kirchhoff



All engineers follow this principle!

$$O = I_1 + I_2$$

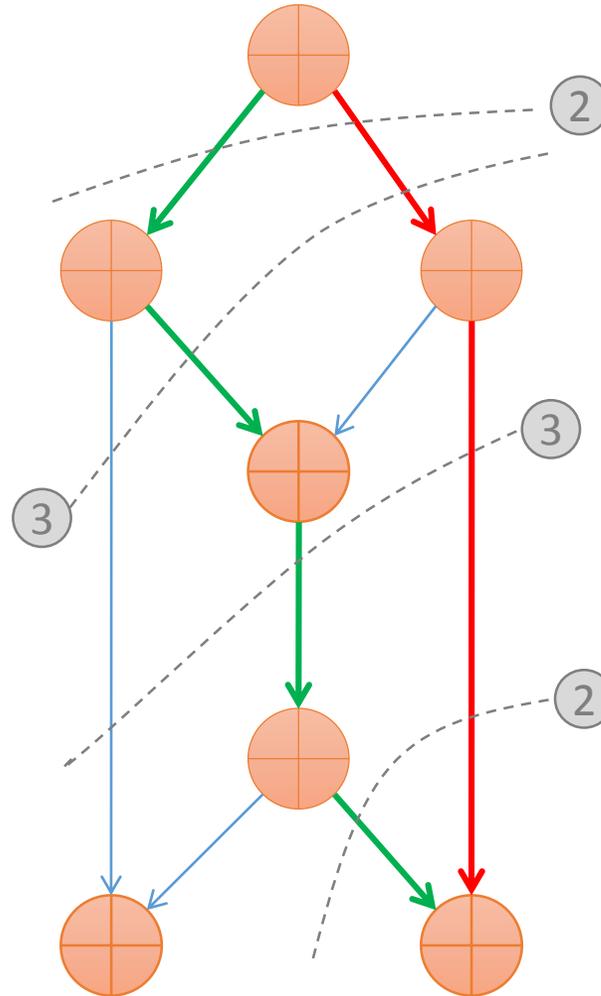
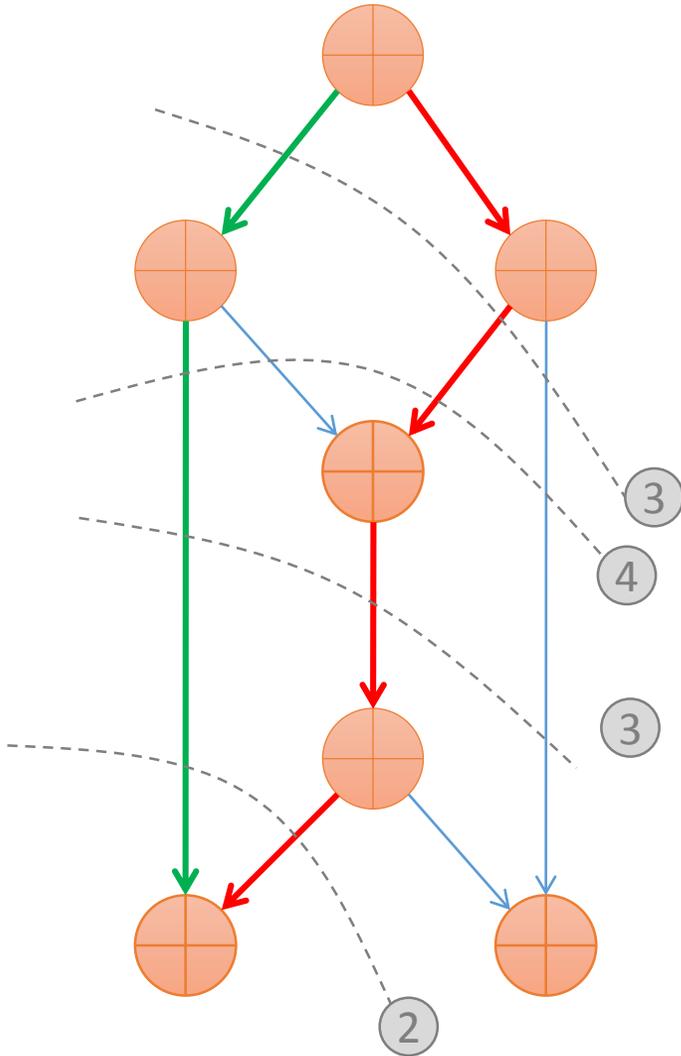
Network Coding



Now we are alone ...!

$$O = f(I_1, I_2)$$

# Max-flow min-cut theorem



*The existence of polynomial time algorithms is remarkable because the maximal rate without coding can be much smaller and finding the routing solution that achieves that maximum is NP-hard.*

Jaggi-Sanders algorithm (2003):  
 Polynomial Time Algorithms for  
 Multicast Network Code Construction:  
 S. Jaggi, P. Sanders, P. A. Chou, M.  
 Effros, S. Egner, K. Jain, and L. M. G. M.  
 Tolhuizen, "Polynomial time algorithms  
 for multicast network code  
 construction," IEEE Trans. Inf. Theory,  
 vol. 51, no. 6, pp. 1973–1982, Jun. 2005.



# Network Coding: The Butterfly

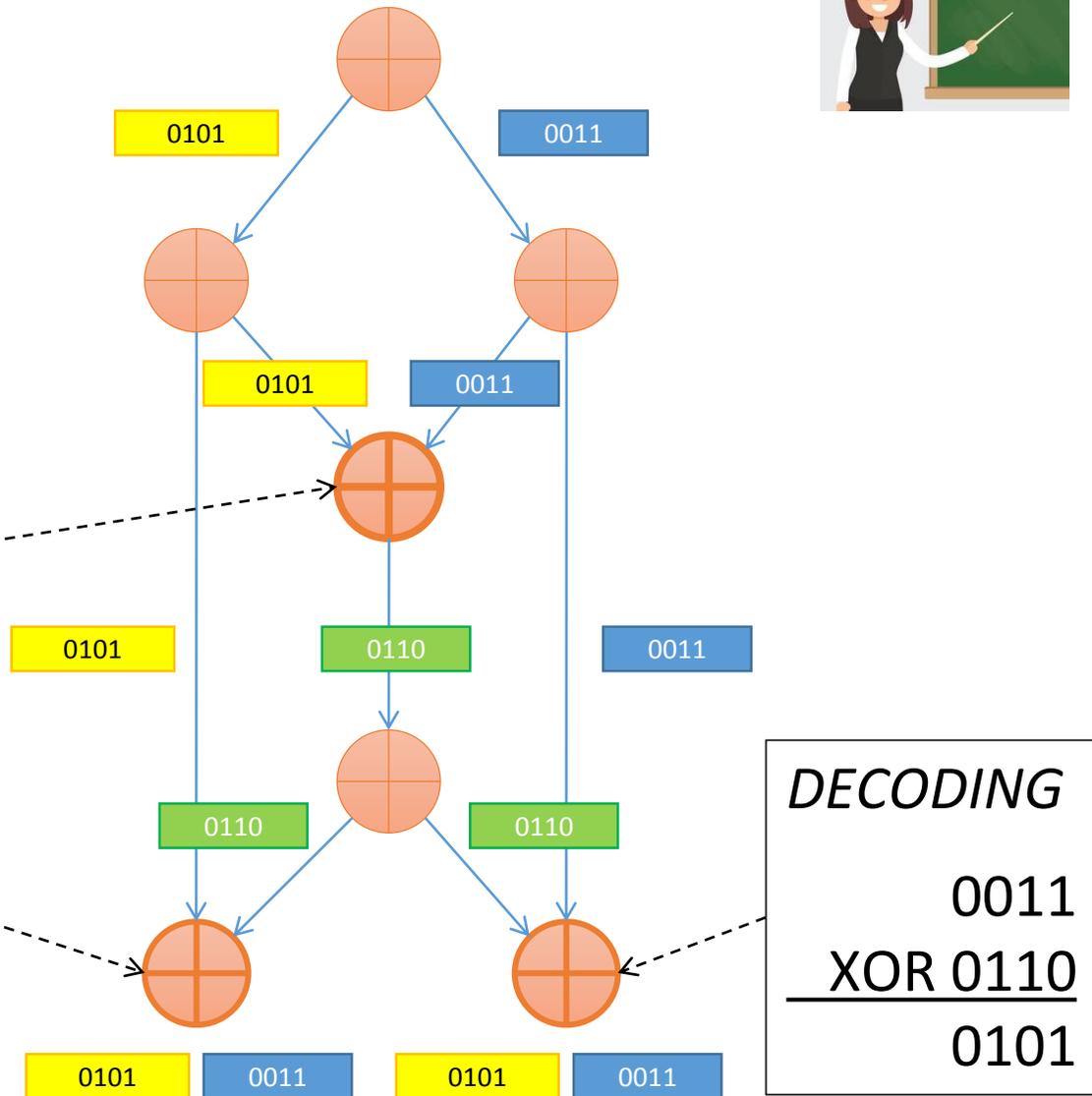
- XOR operation
- Bitwise operation
- Same bit value results in „0“
- Different bit value results in „1“

Input 1	0	1	0	1
Input 2	0	0	1	1
<b>Result</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>

*CODING*

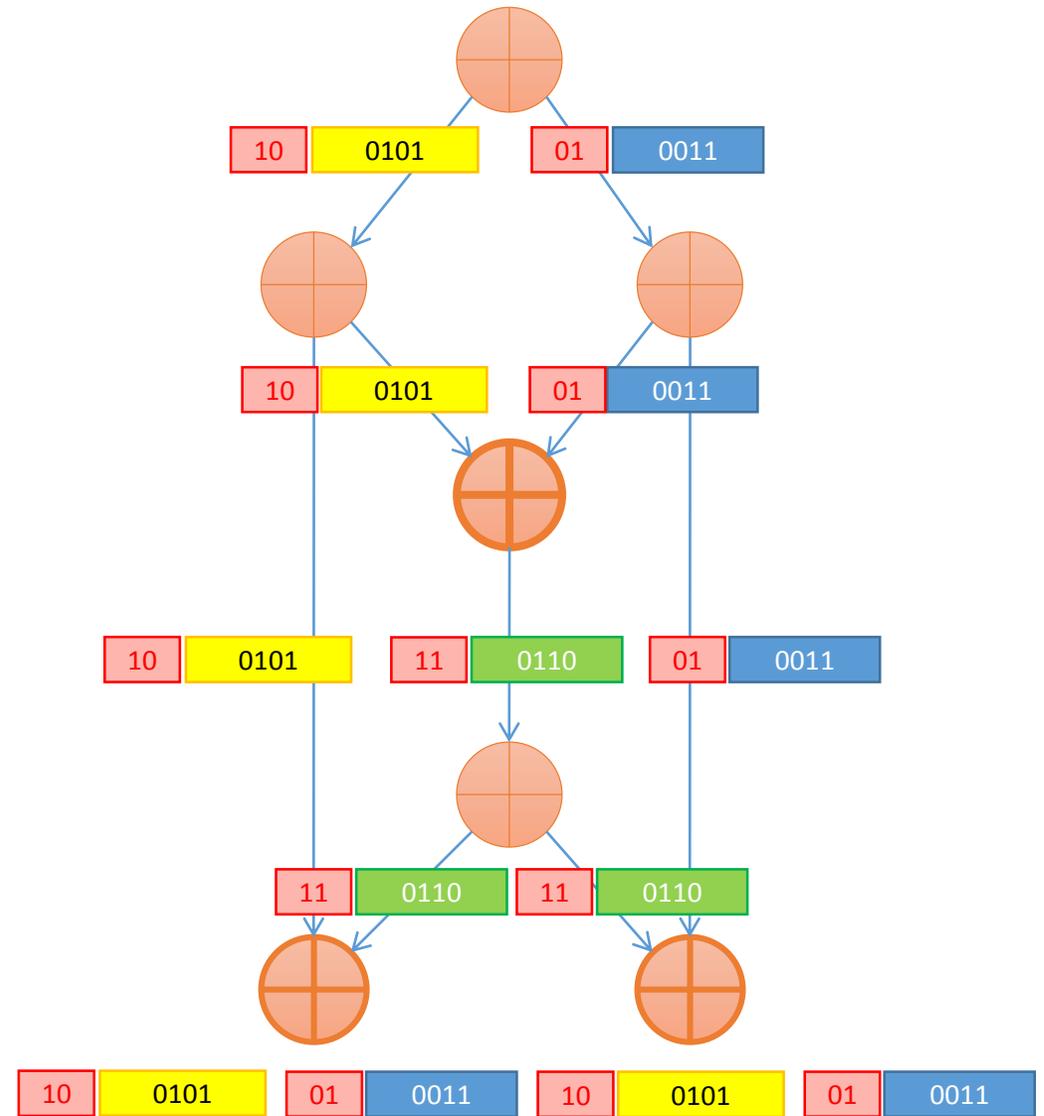
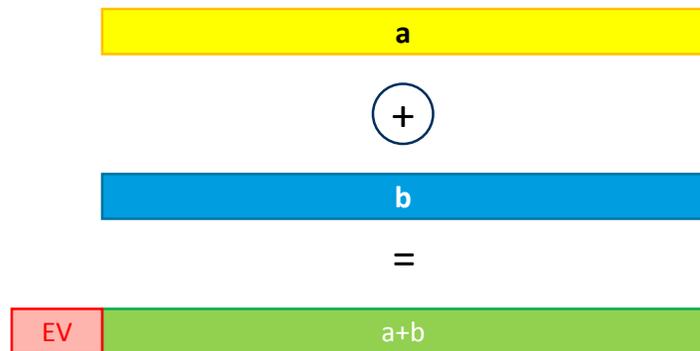
$$\begin{array}{r} 0101 \\ \text{XOR } 0011 \\ \hline 0110 \end{array}$$

*DECODING*

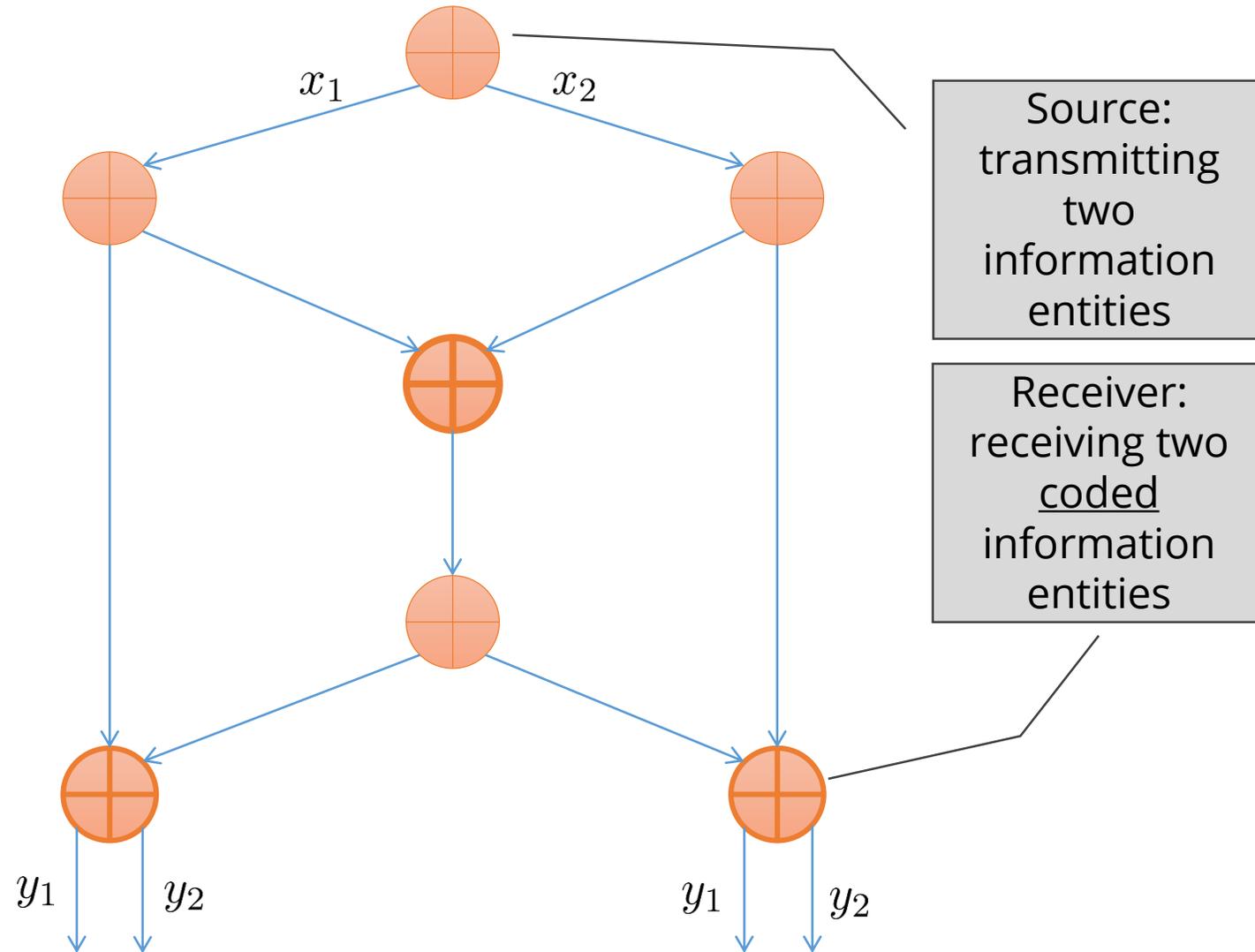
$$\begin{array}{r} 0101 \\ \text{XOR } 0110 \\ \hline 0011 \end{array}$$


# Network Coding: The Butterfly

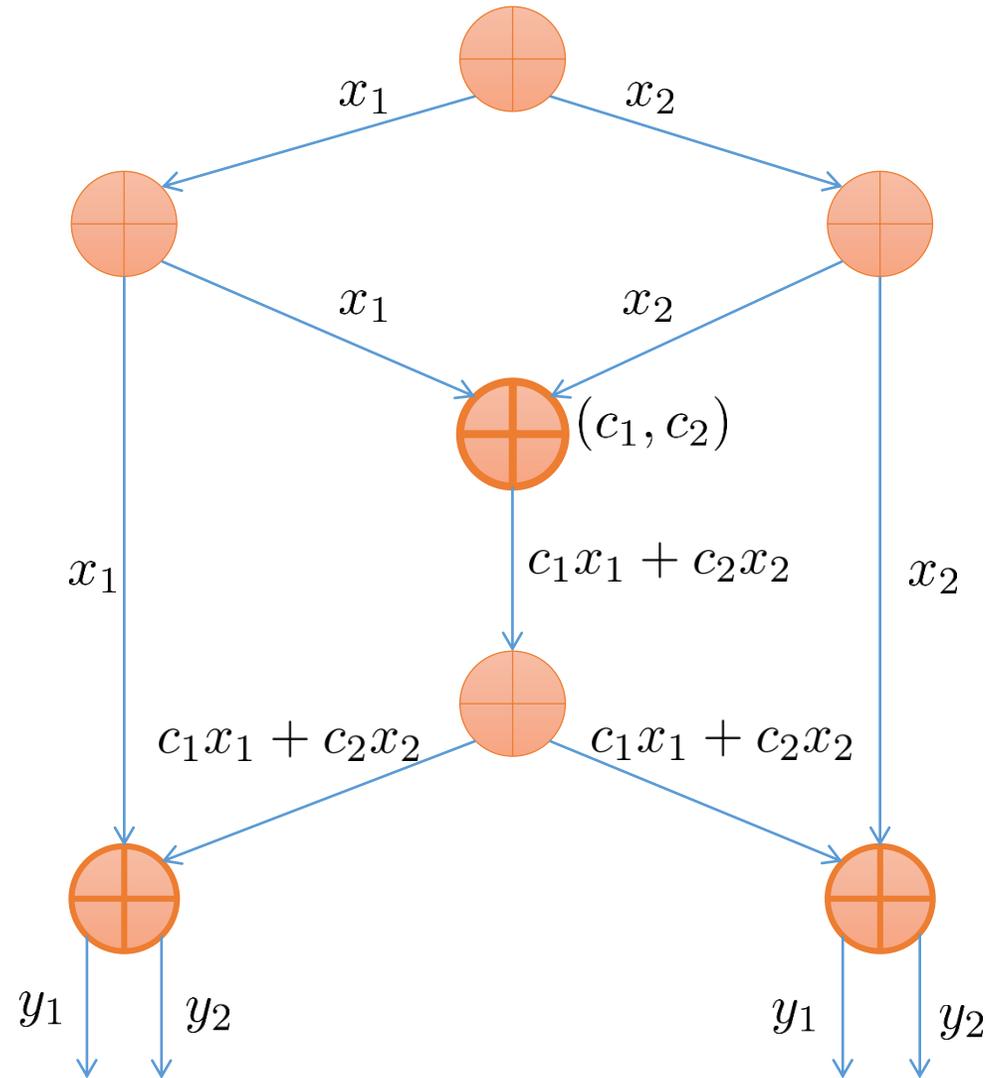
- Adding complexity at some nodes of the network
- Adding overhead in order to know what was coded together (encoding vector)



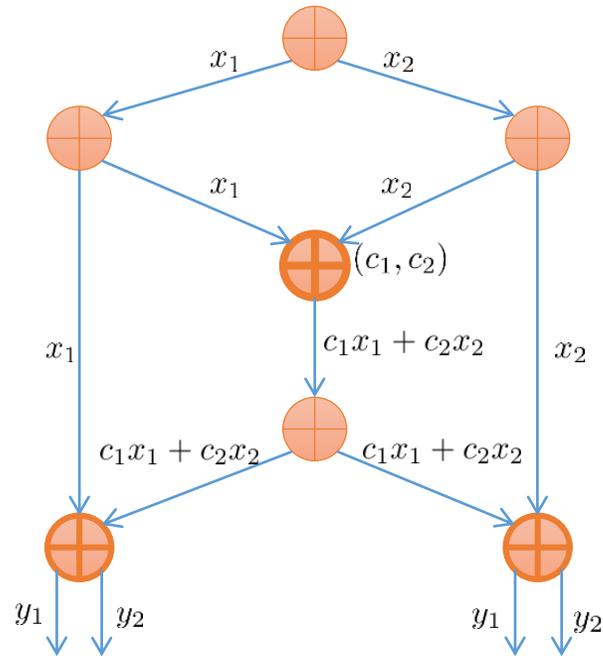
# Network Coding: The Butterfly



# Network Coding: The Butterfly



# Network Coding: The Butterfly



$$y_1 = x_1$$

$$y_2 = c_1x_1 + c_2x_2$$

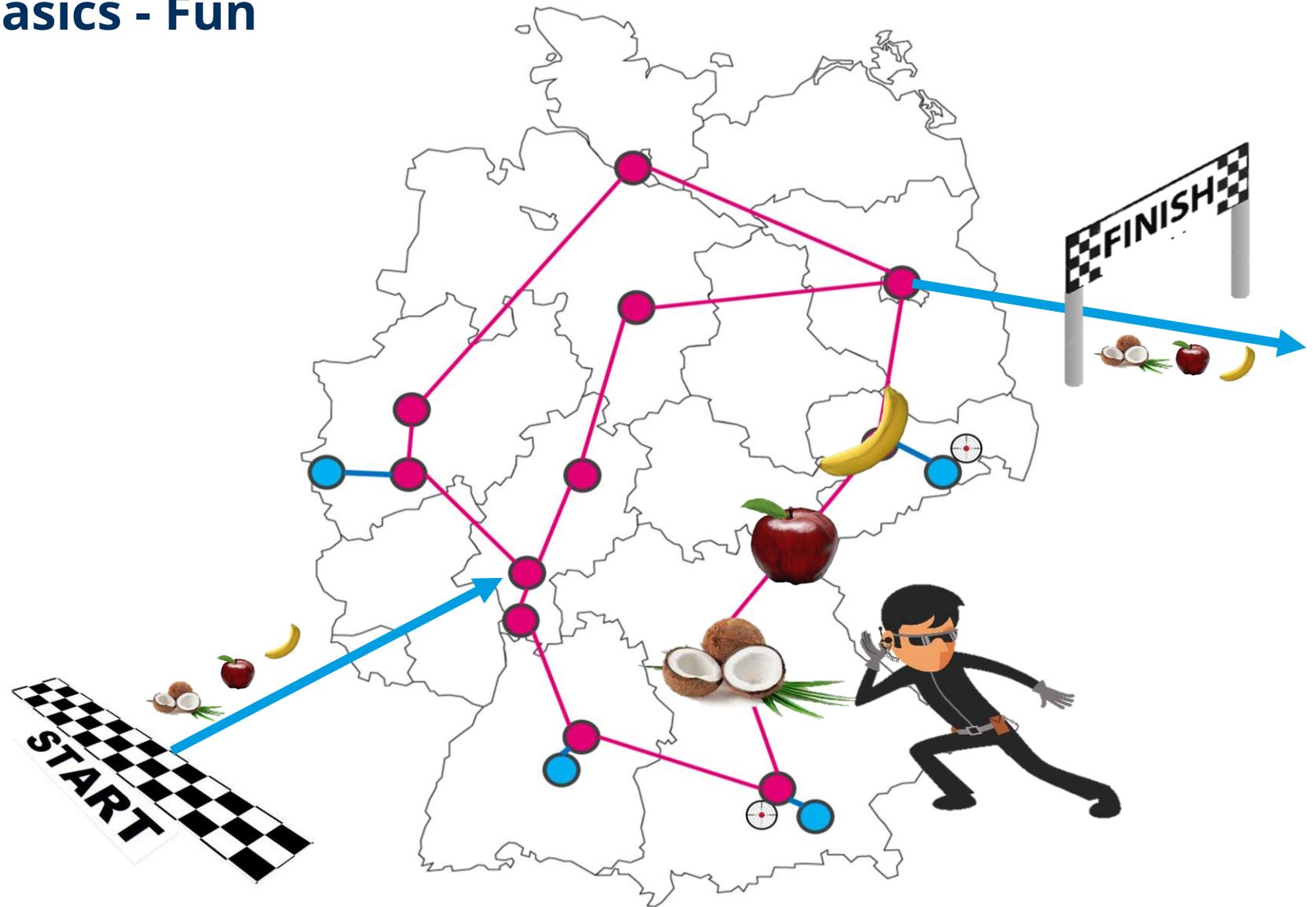
$$y_1 = c_1x_1 + c_2x_2$$

$$y_2 = x_2$$

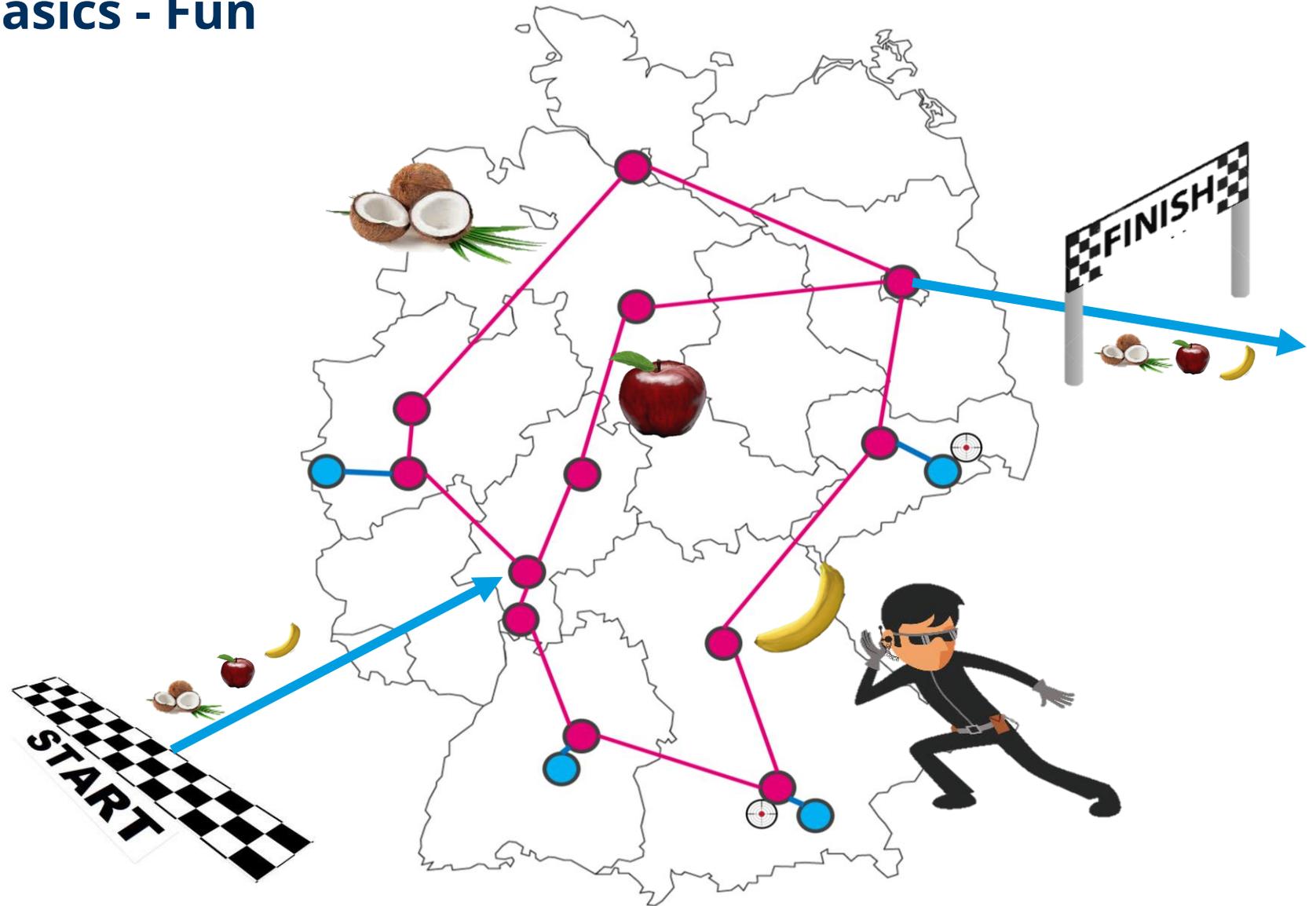
$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ c_1 & c_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} c_1 & c_2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

# Network Coding Basics - Fun



# Network Coding Basics - Fun



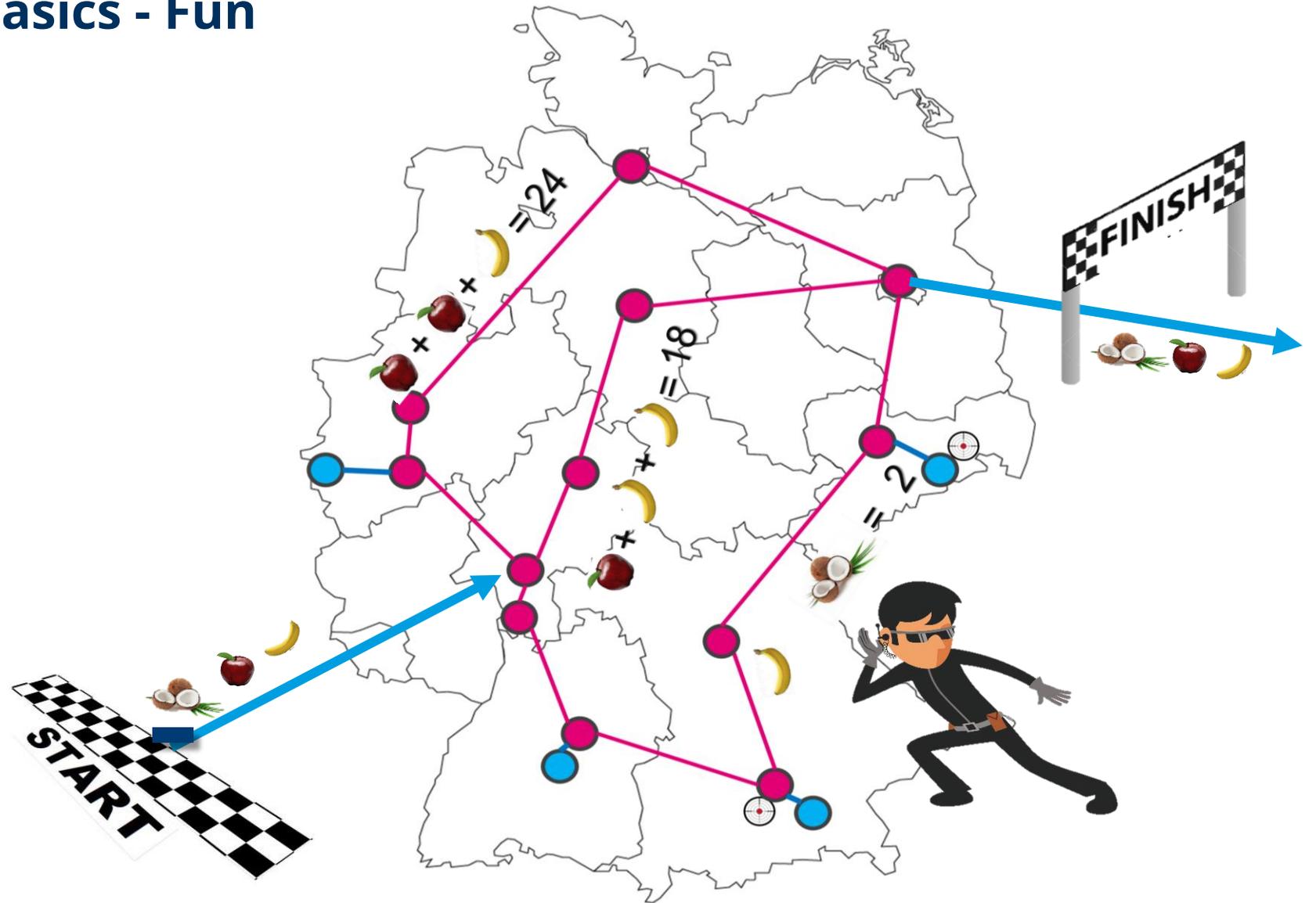
# Network Coding Basics - Fun

$$\text{Apple} + \text{Apple} + \text{Banana} = 24$$

$$\text{Apple} + \text{Banana} + \text{Banana} = 18$$

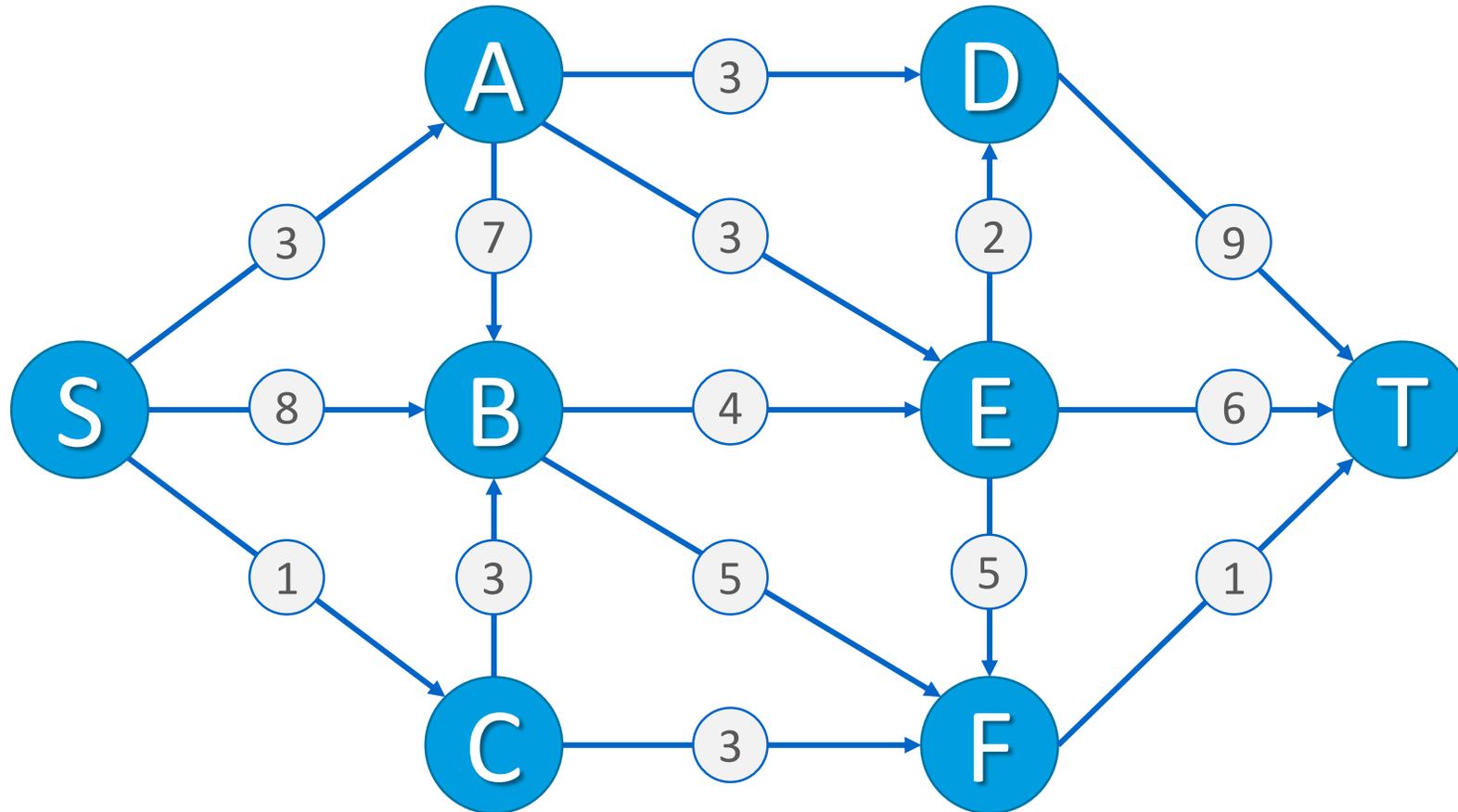
$$\text{Banana} - \text{Coconut} = 2$$

# Network Coding Basics - Fun

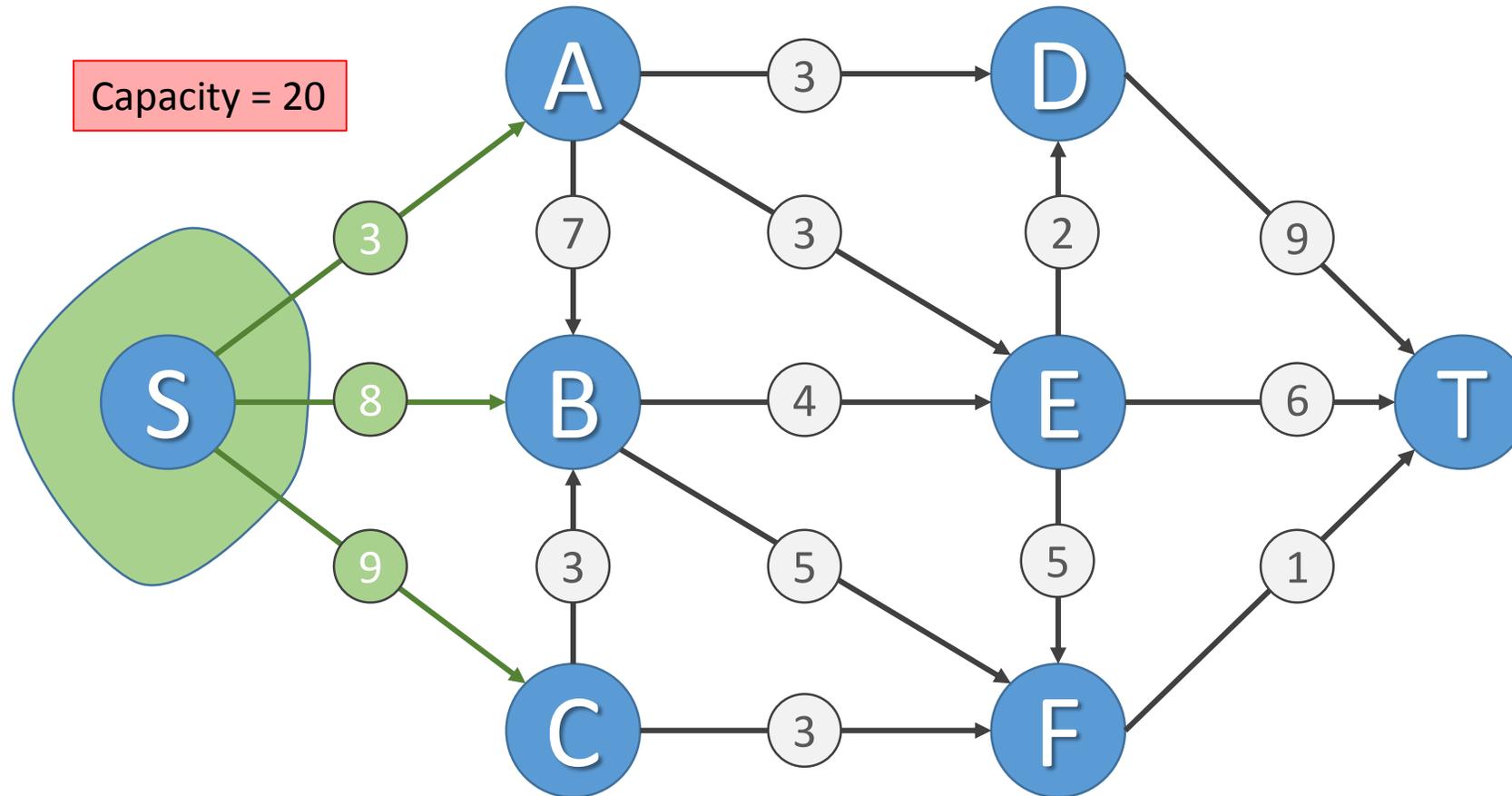


# Min-Cut

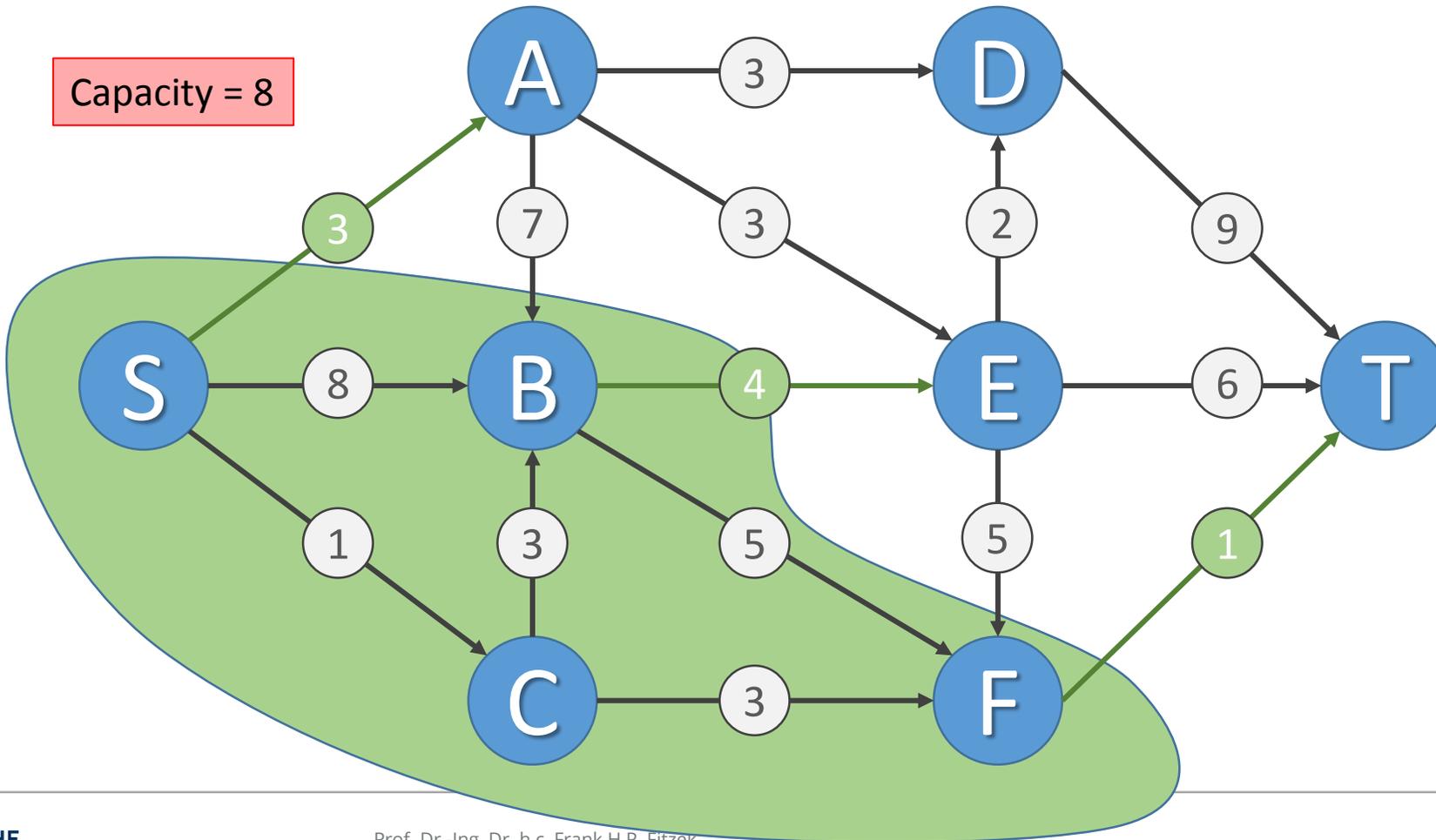
# Cuts of Flow Networks



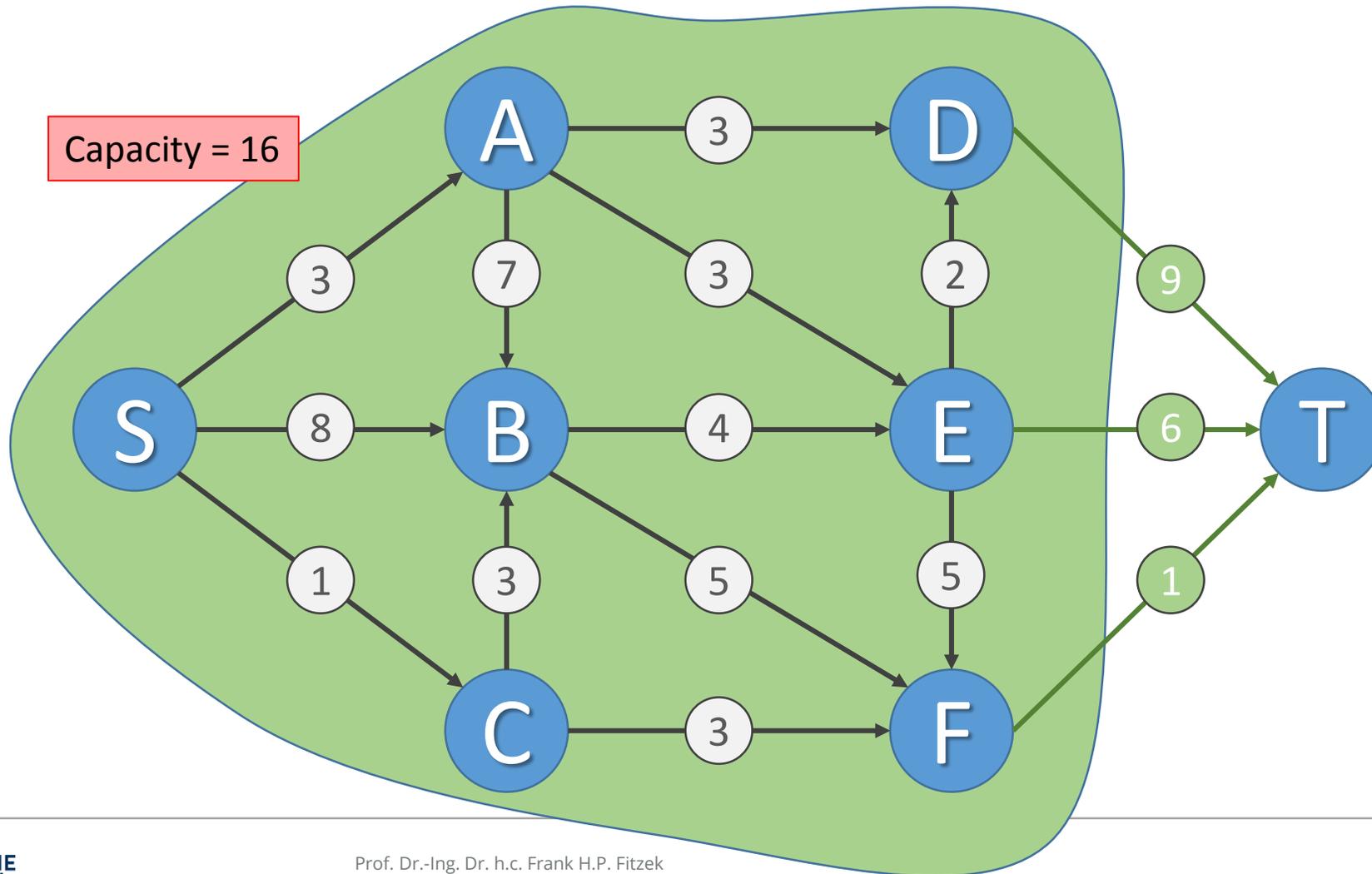
# Cuts of Flow Networks



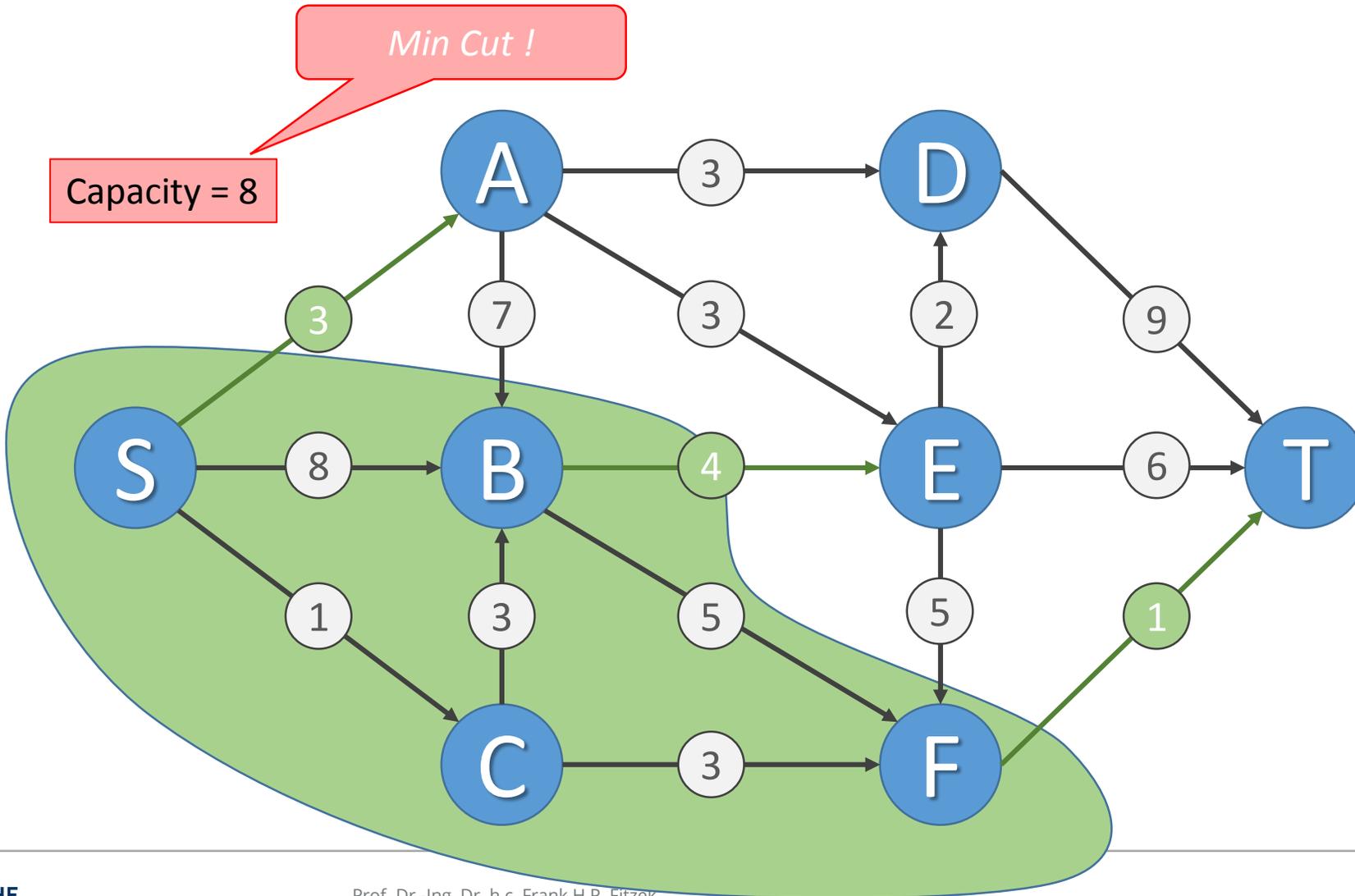
# Cuts of Flow Networks



# Cuts of Flow Networks

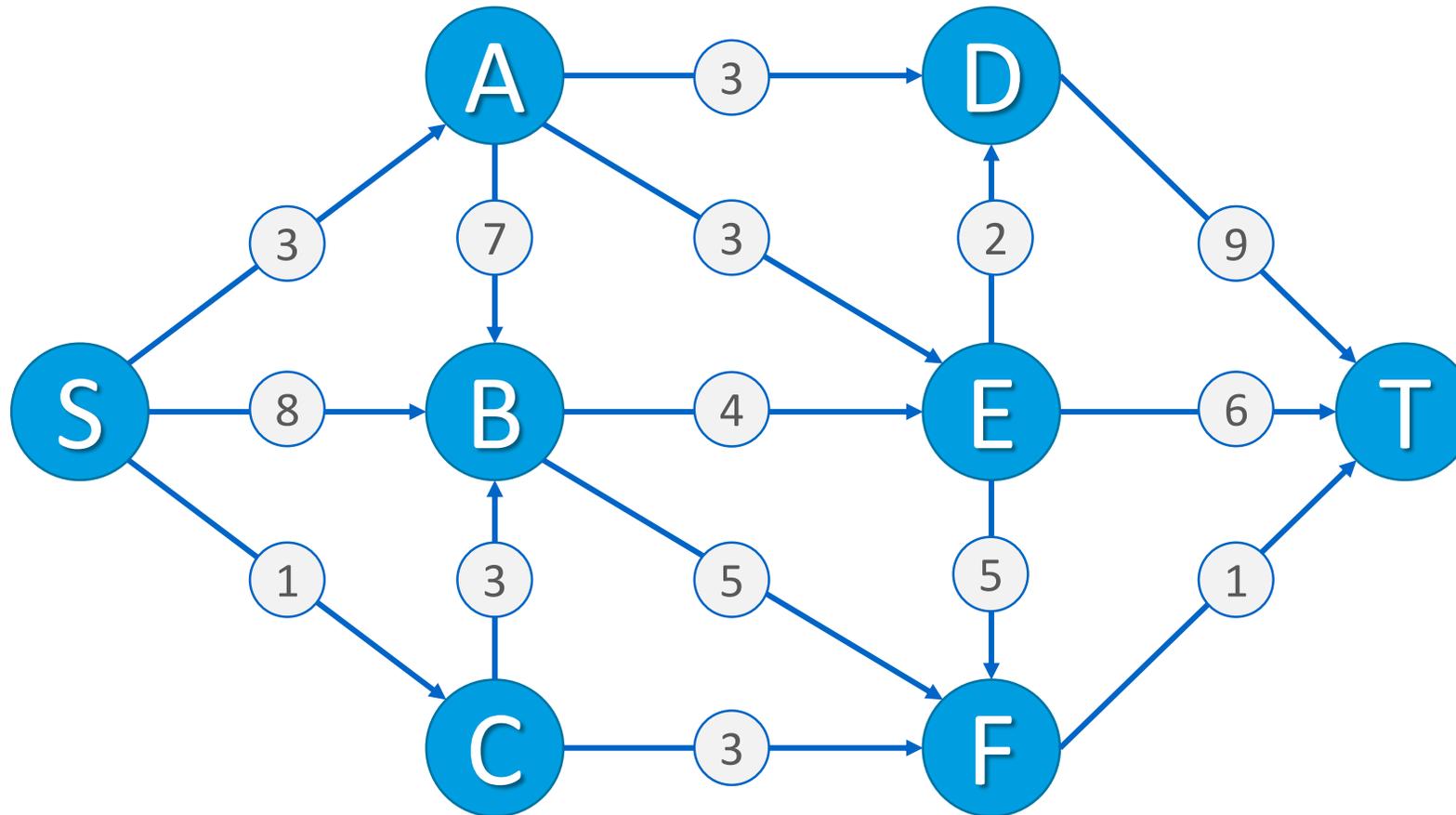


# Cuts of Flow Networks

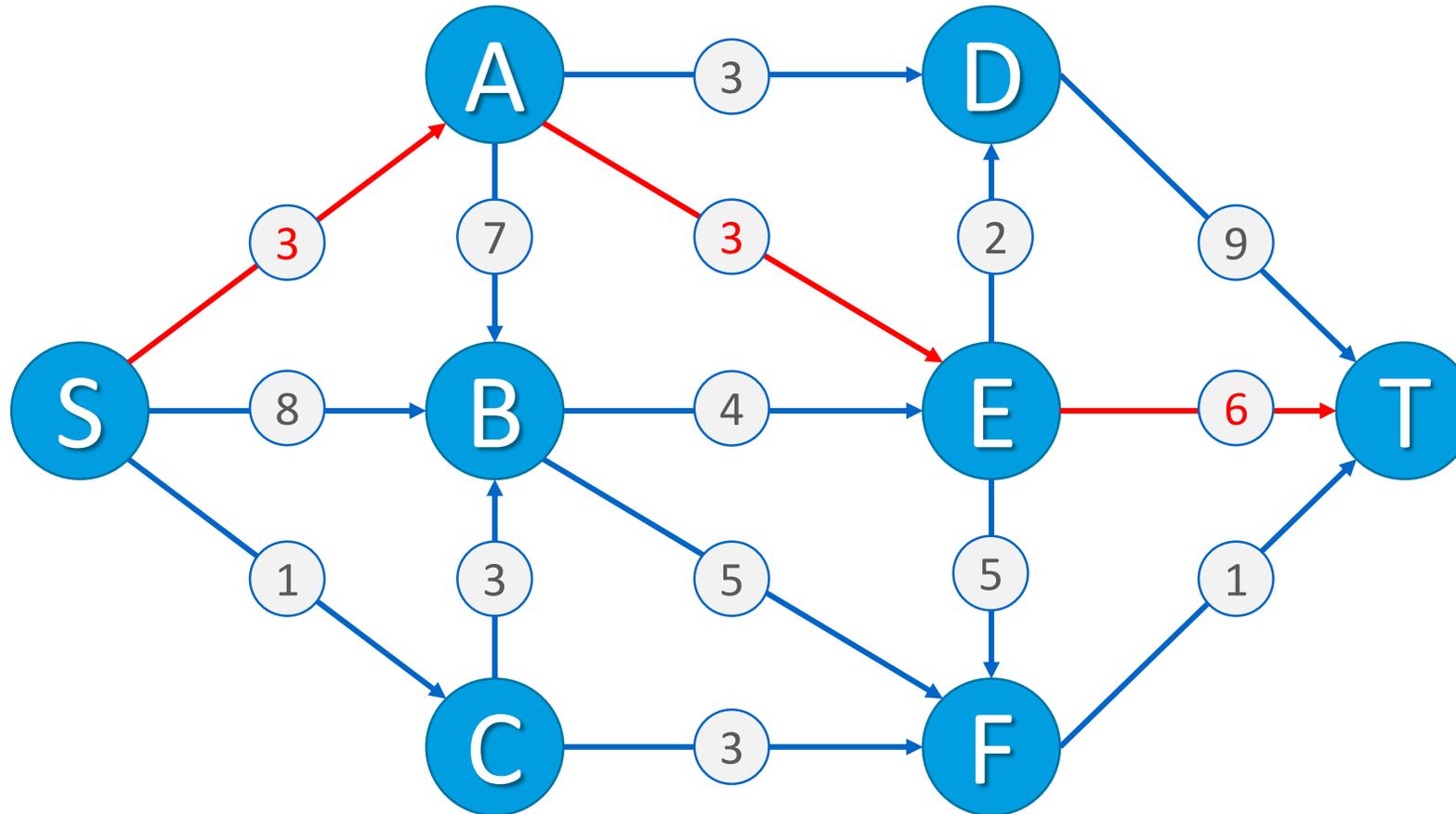


# Ford-Fulkerson

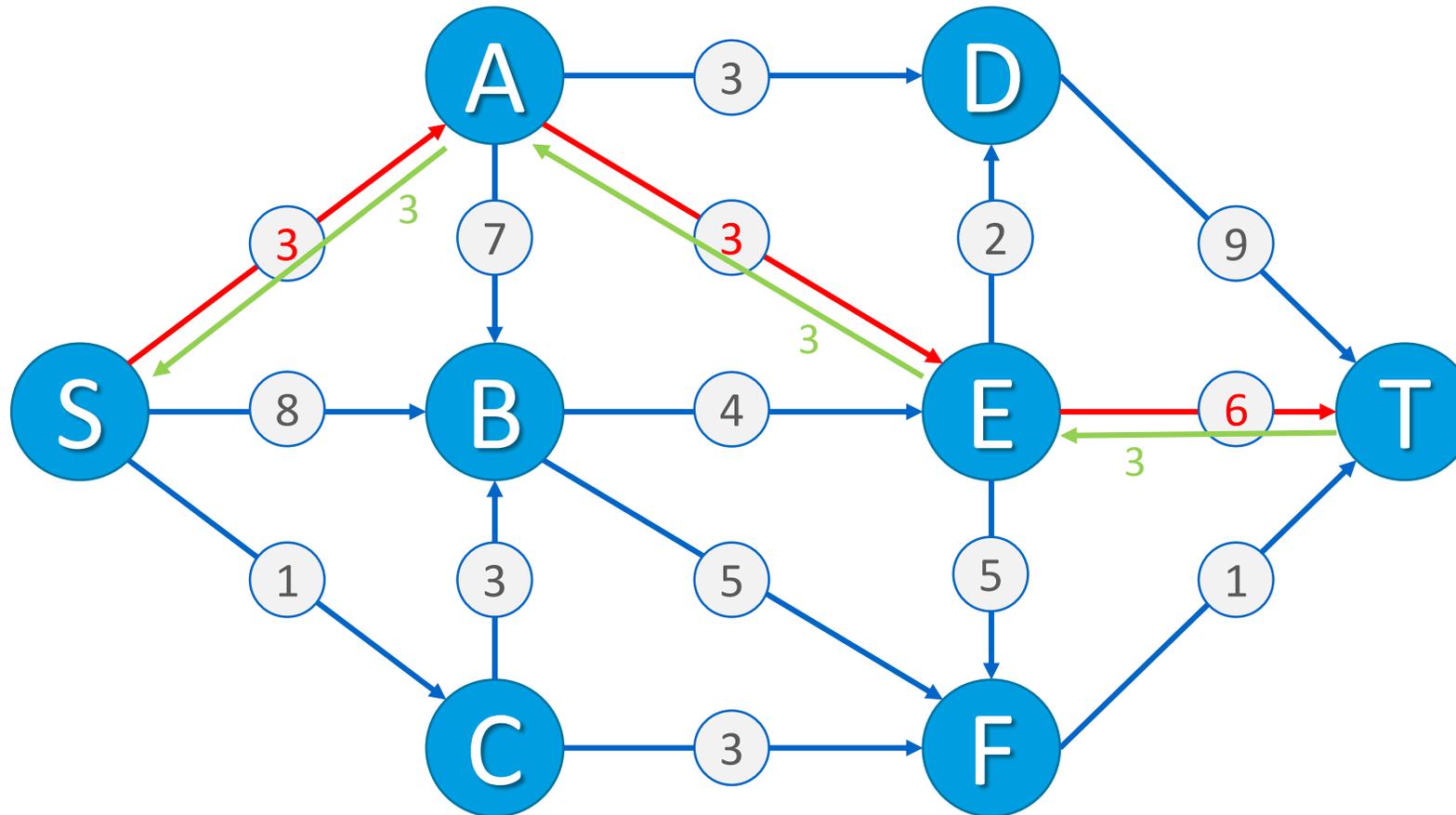
# Ford-Fulkerson Max Flow



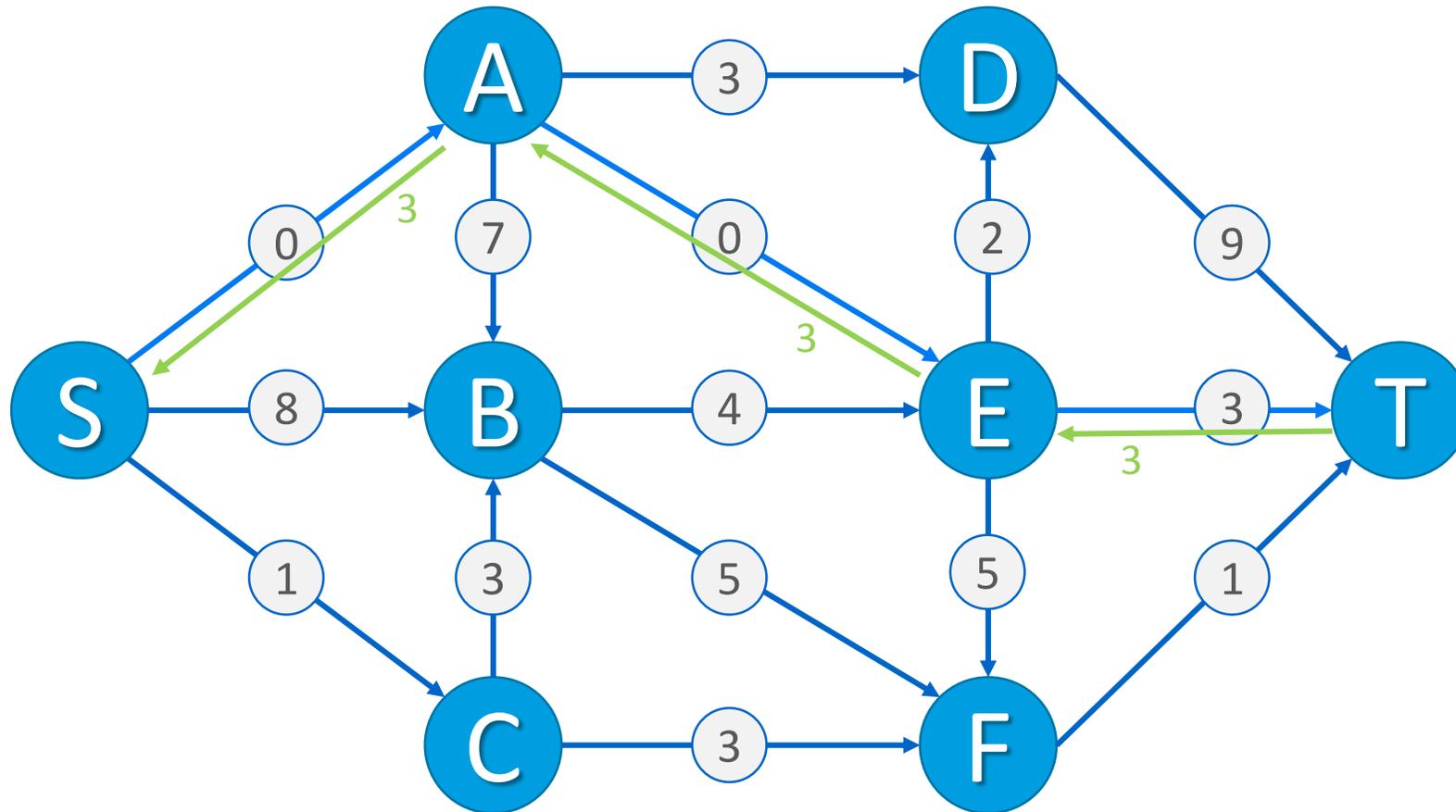
# Ford-Fulkerson Max Flow



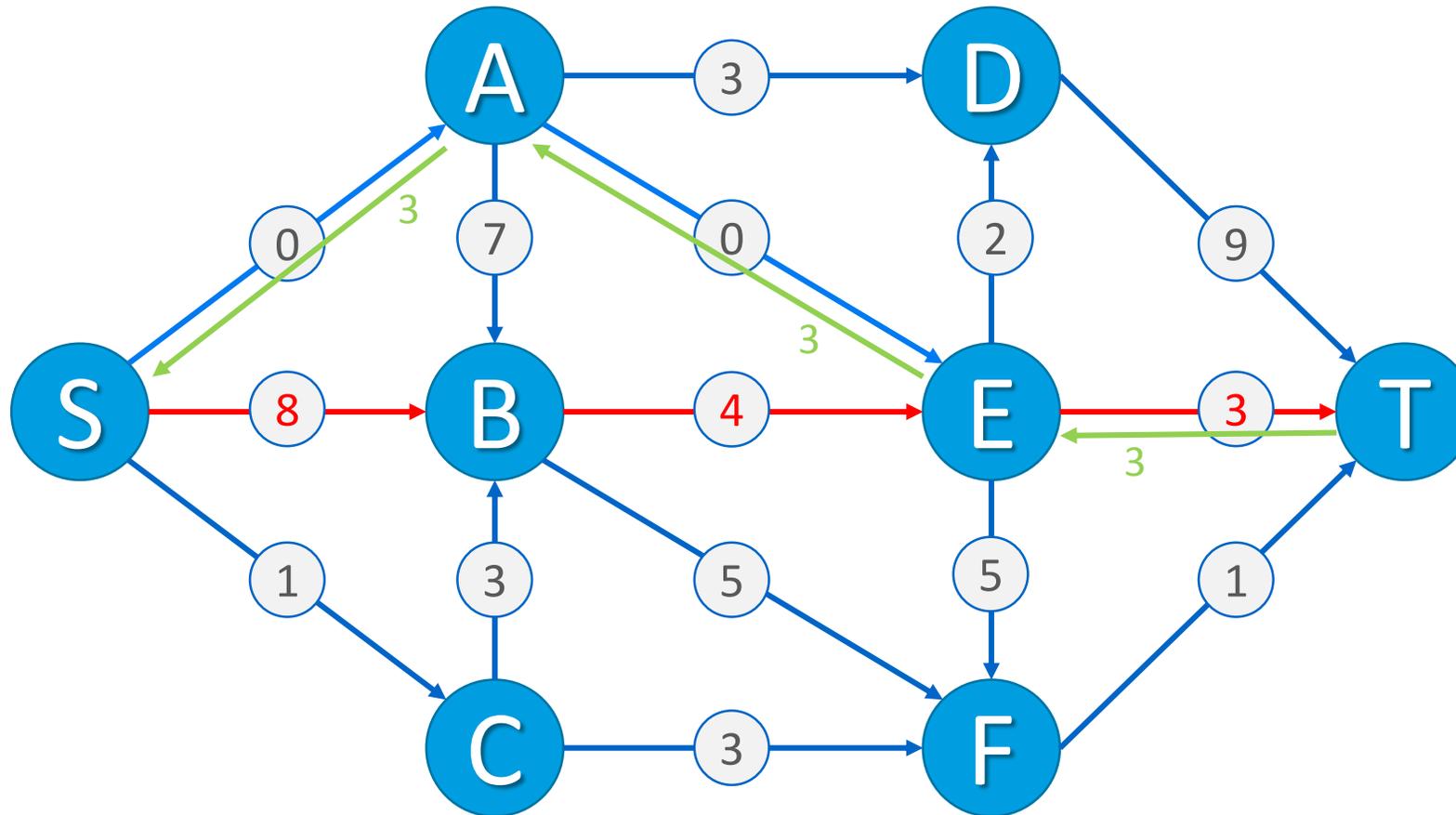
# Ford-Fulkerson Max Flow



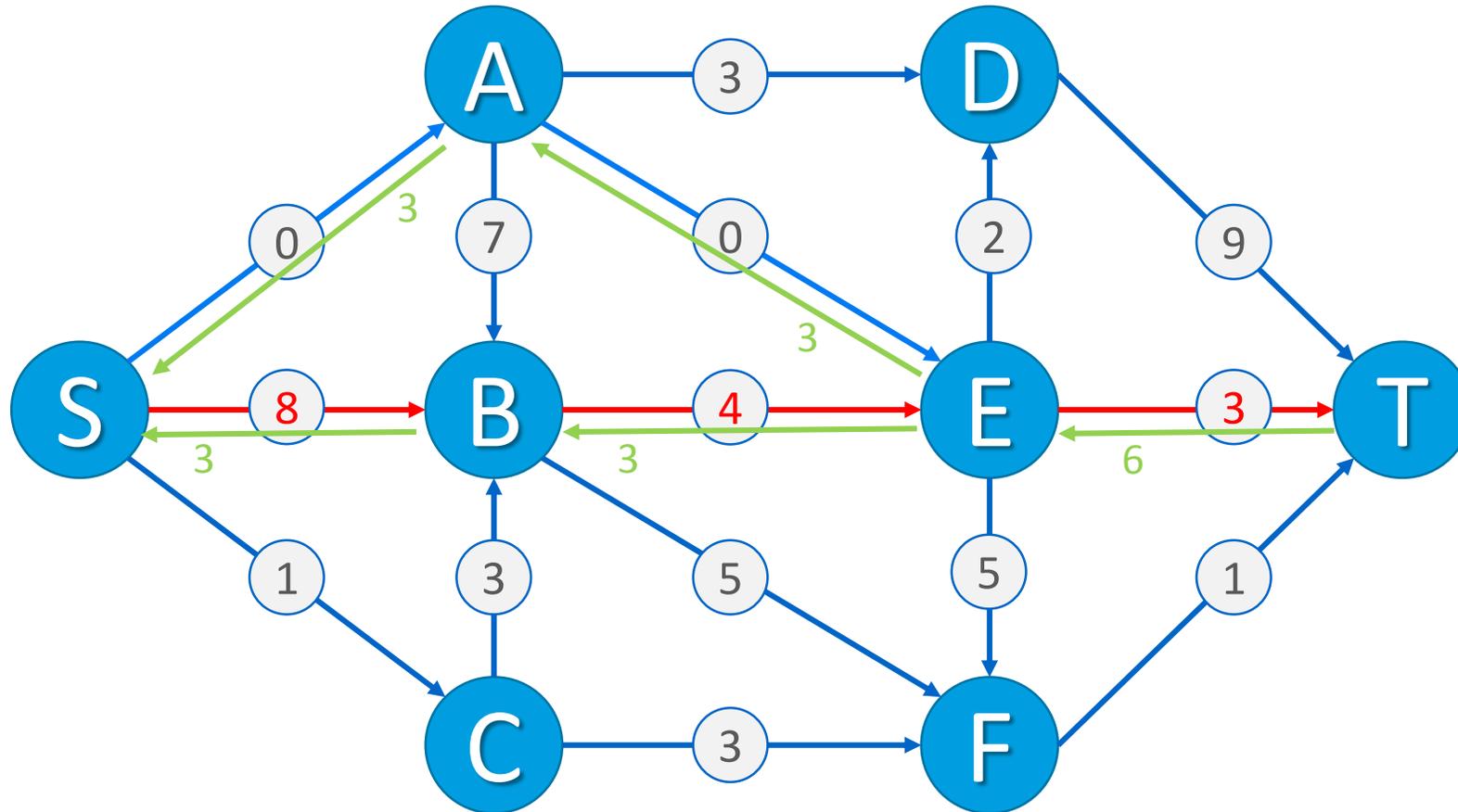
# Ford-Fulkerson Max Flow



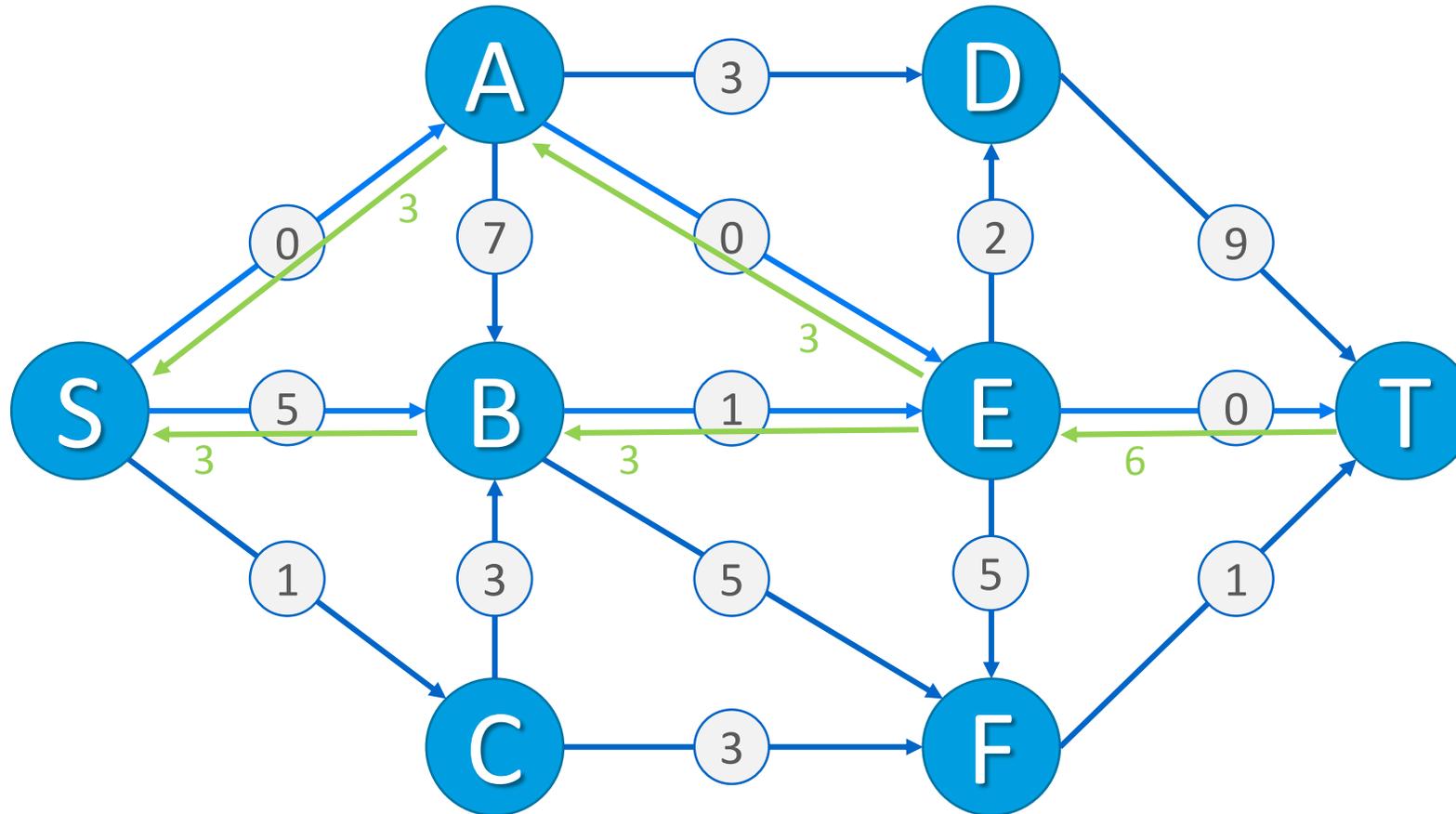
# Ford-Fulkerson Max Flow



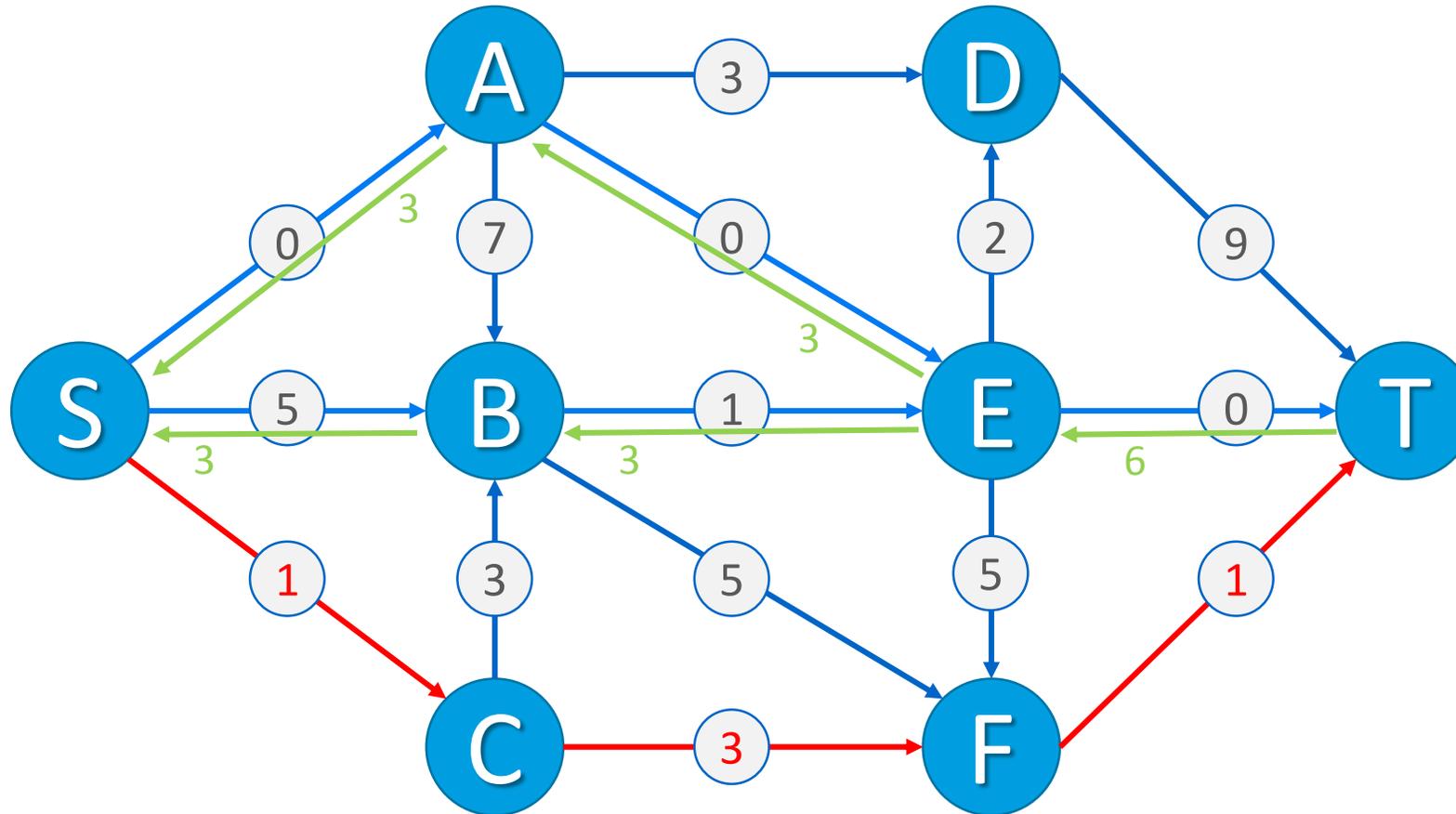
# Ford-Fulkerson Max Flow



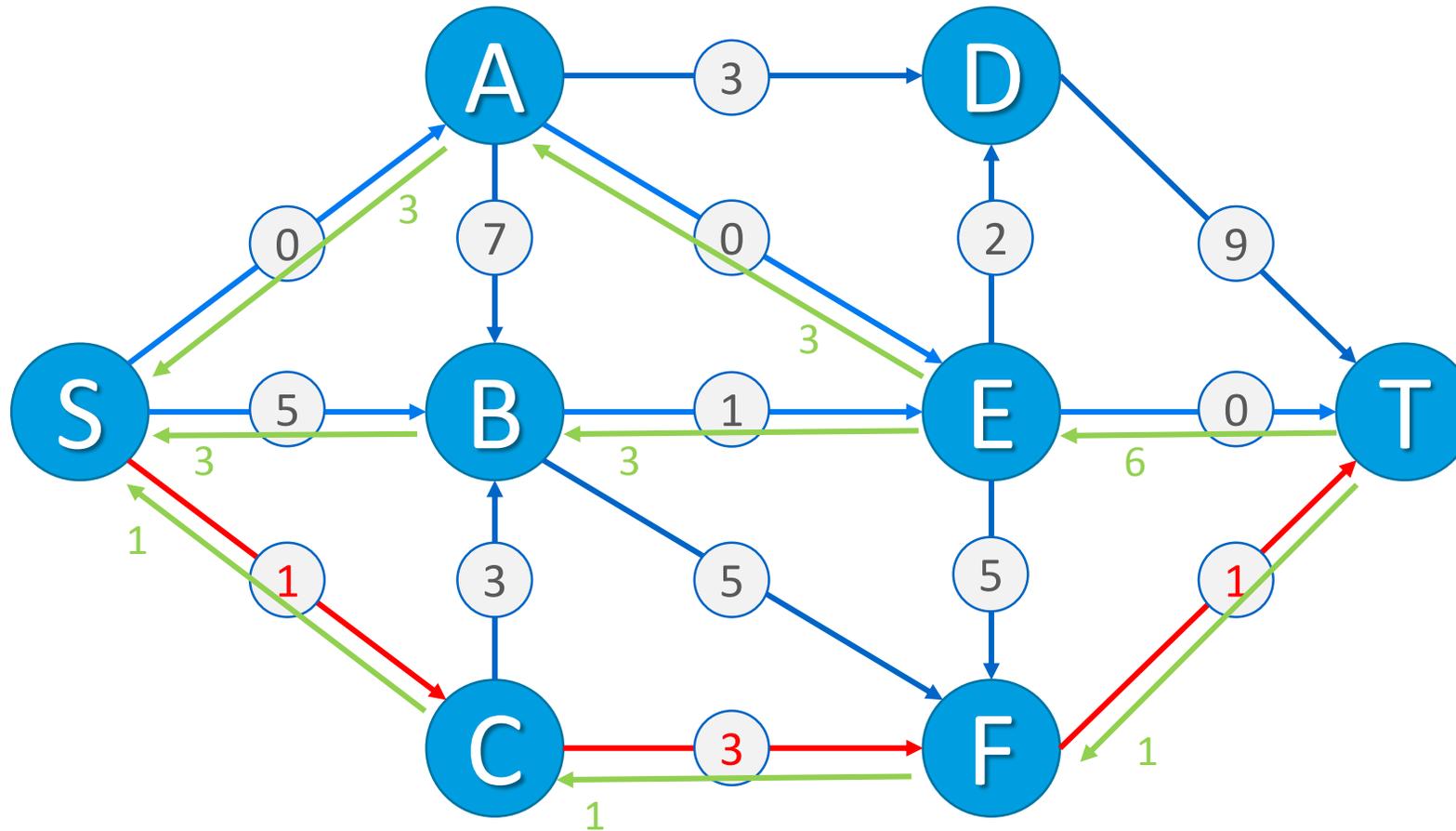
# Ford-Fulkerson Max Flow



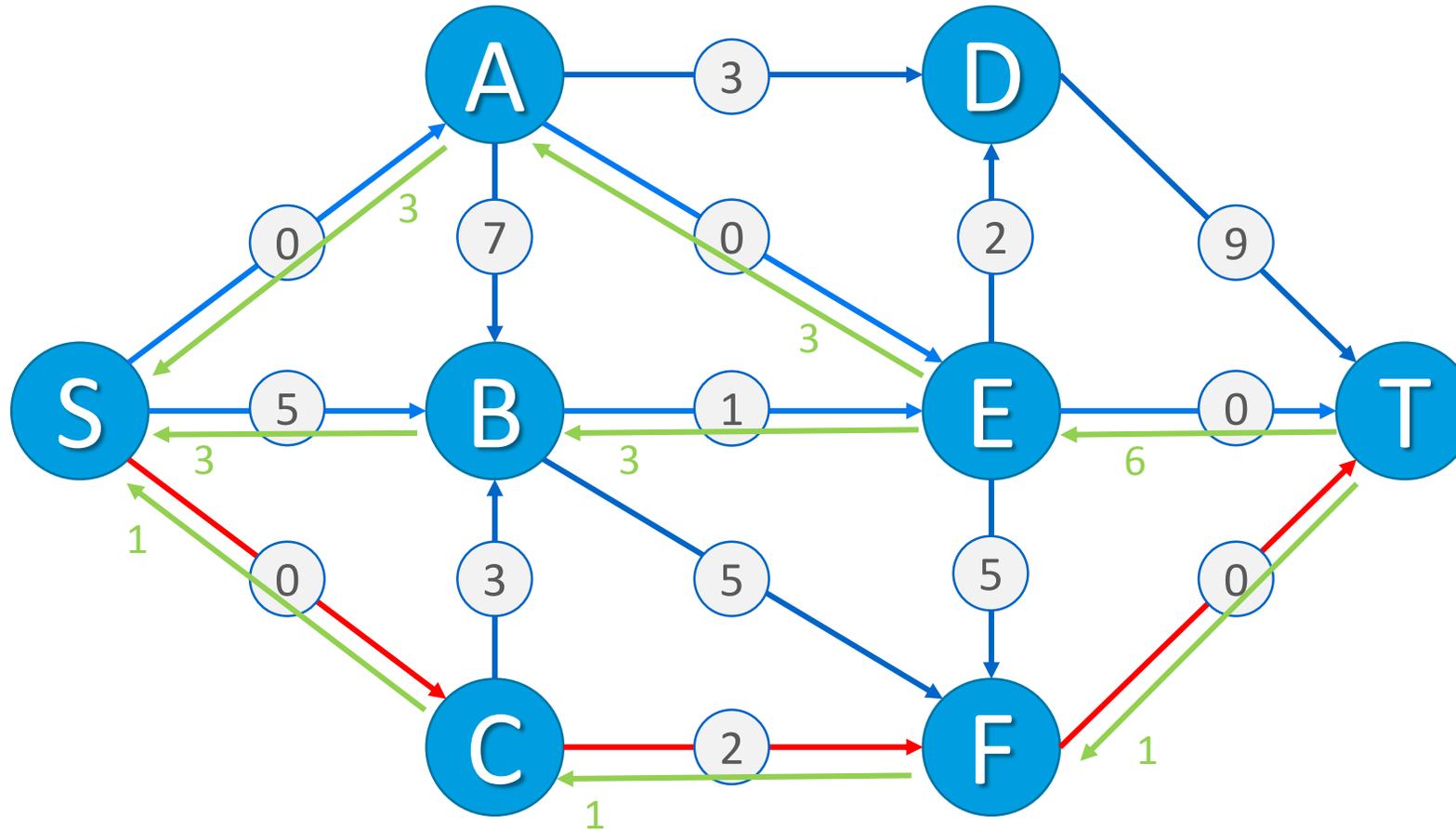
# Ford-Fulkerson Max Flow



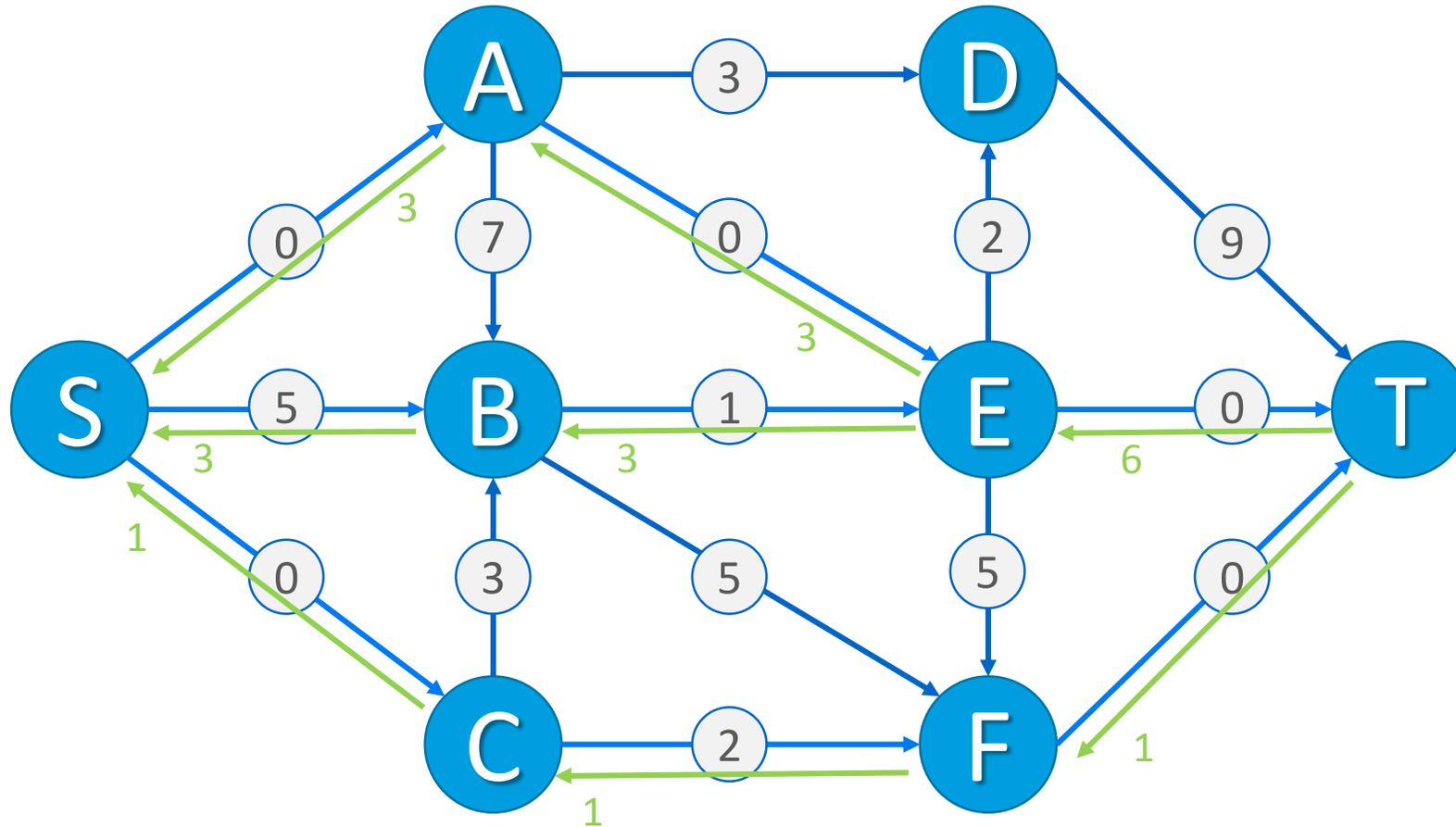
# Ford-Fulkerson Max Flow



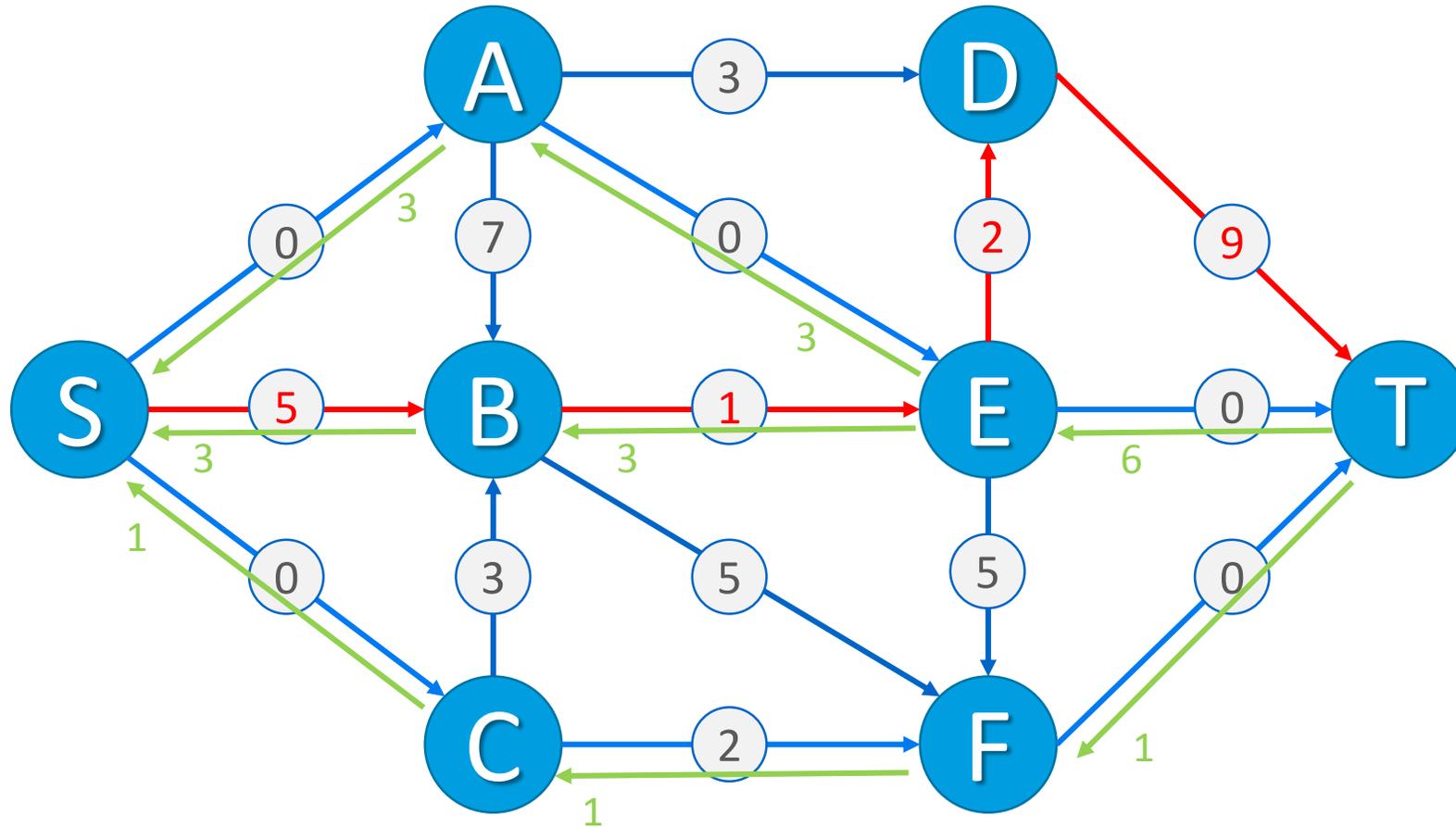
# Ford-Fulkerson Max Flow



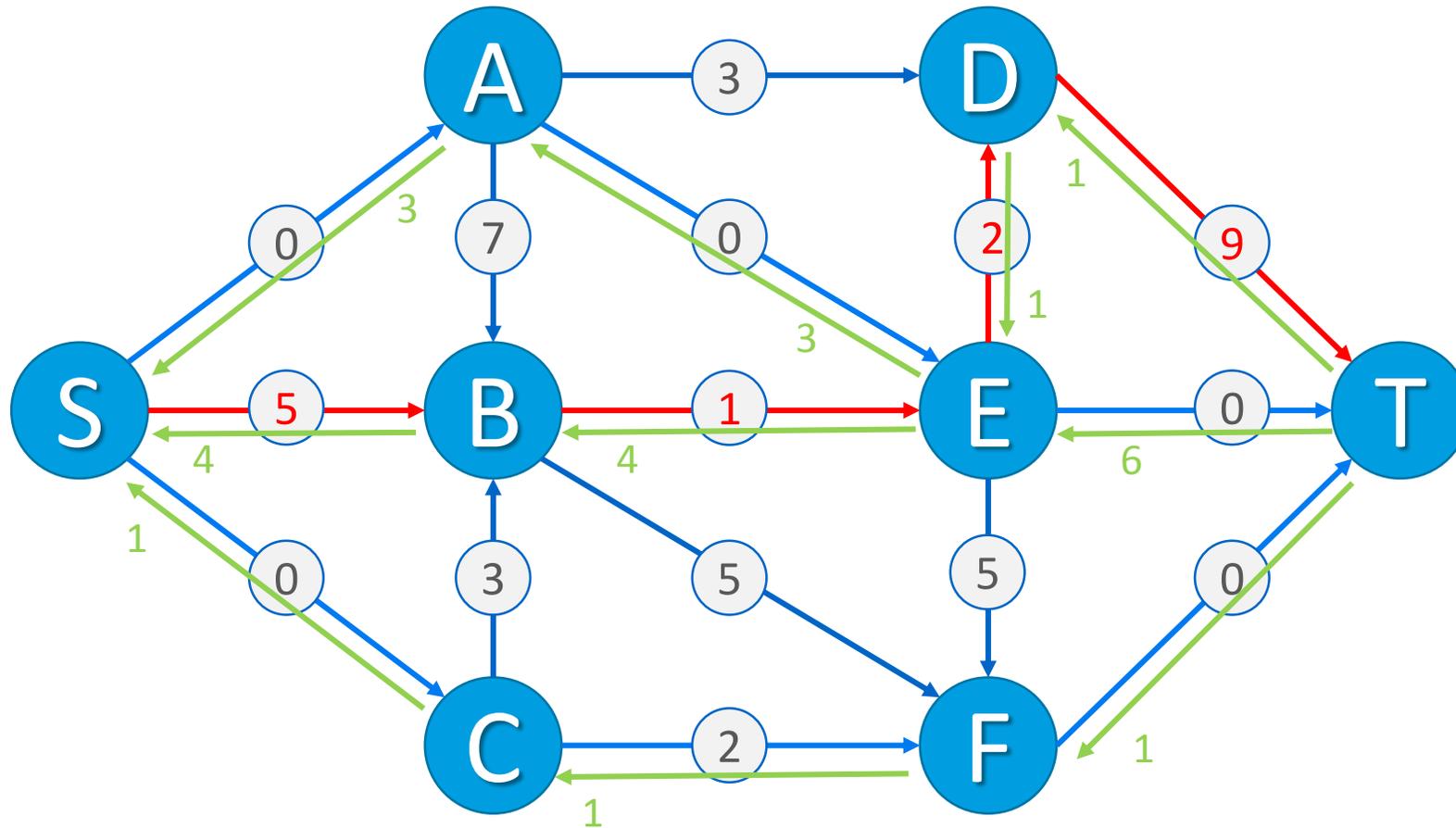
# Ford-Fulkerson Max Flow



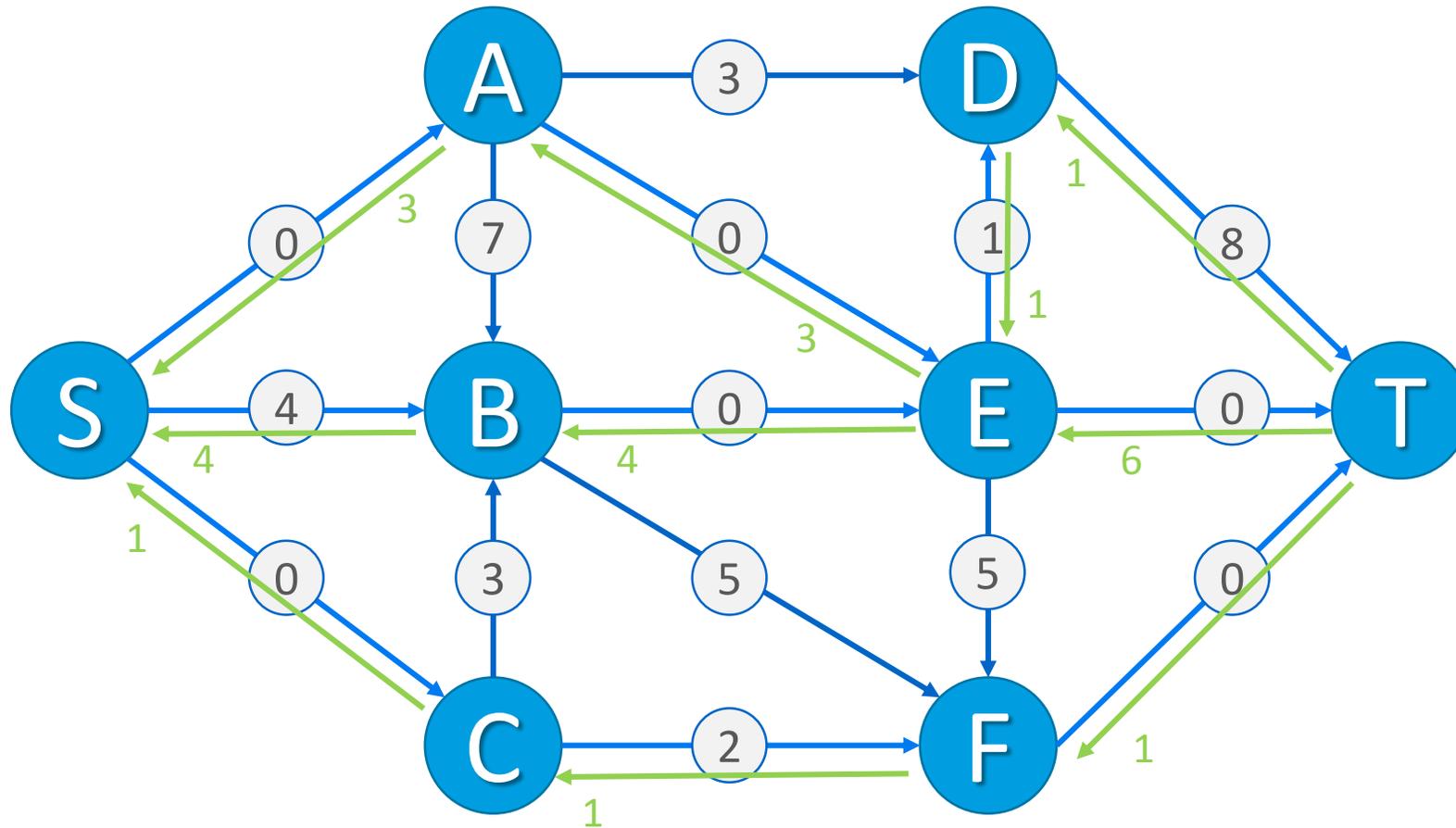
# Ford-Fulkerson Max Flow



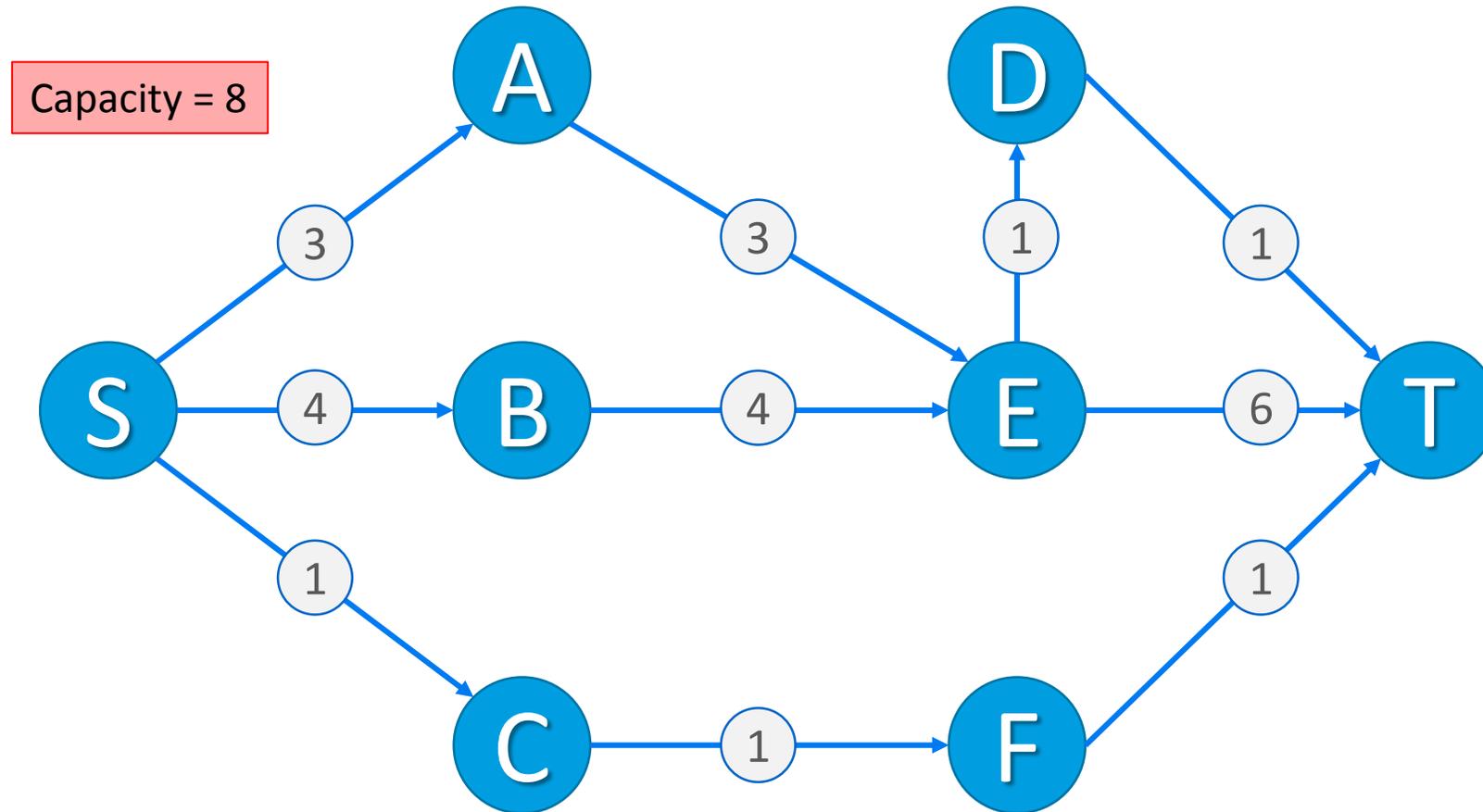
# Ford-Fulkerson Max Flow



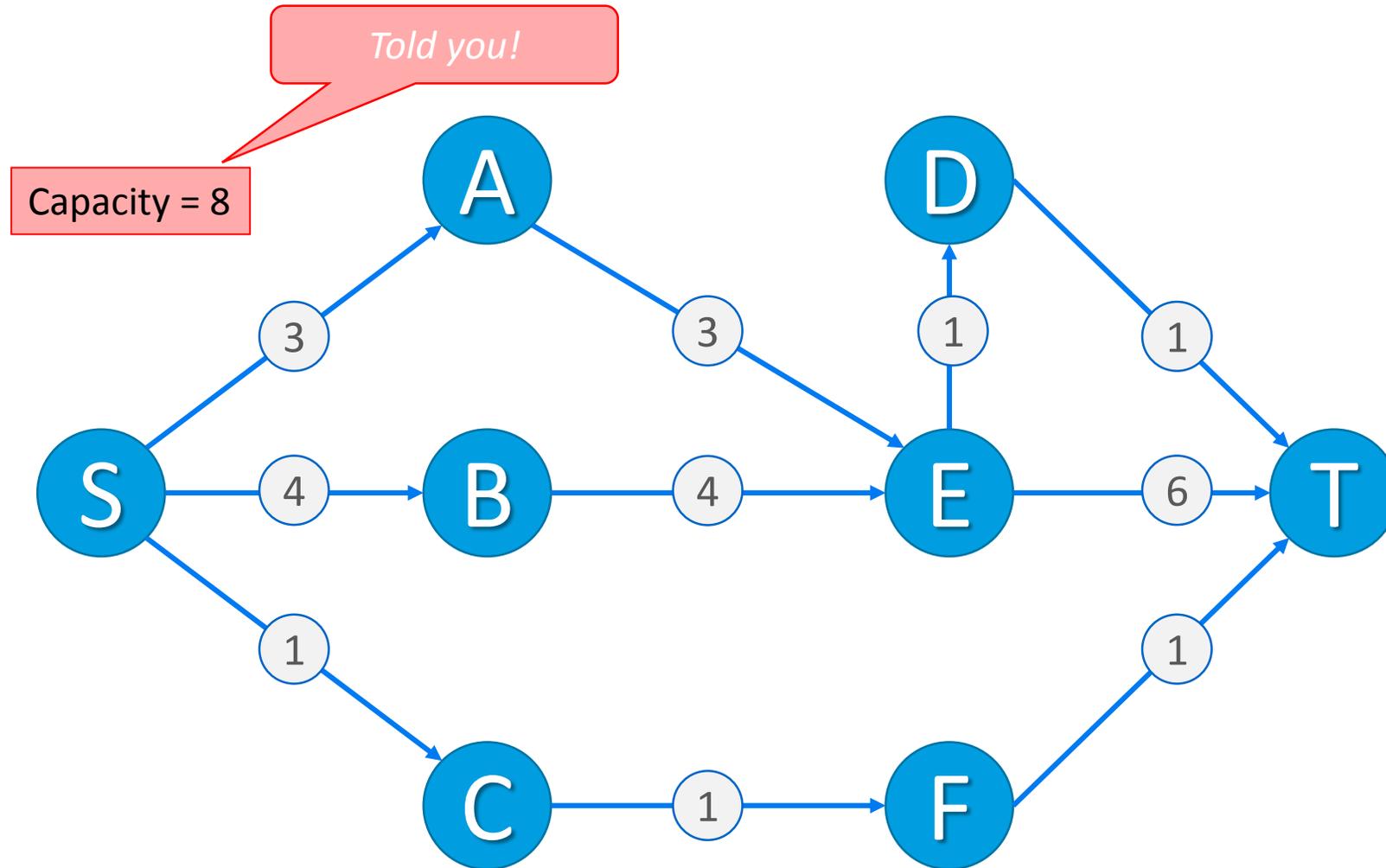
# Ford-Fulkerson Max Flow



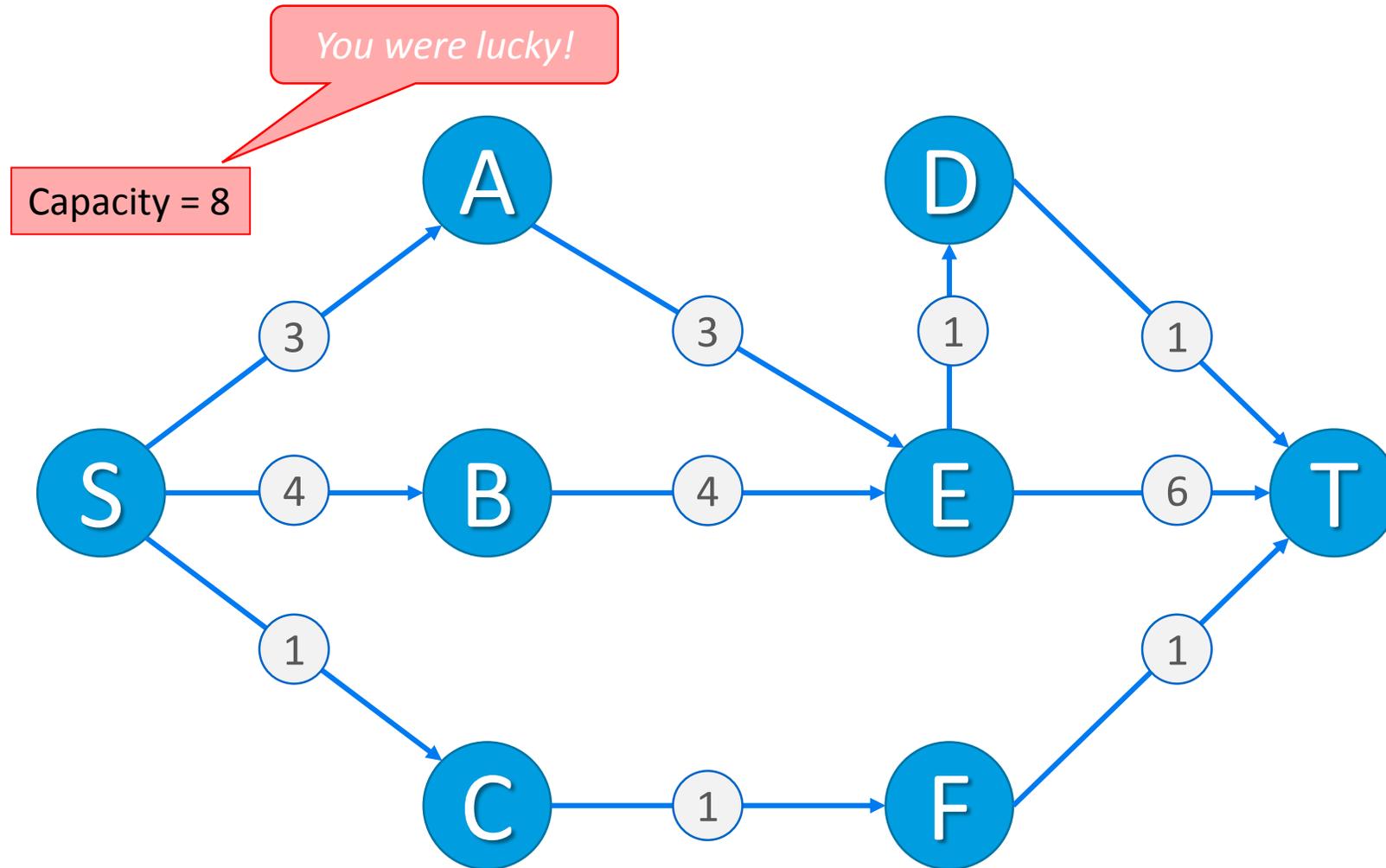
# Ford-Fulkerson Max Flow



# Ford-Fulkerson Max Flow

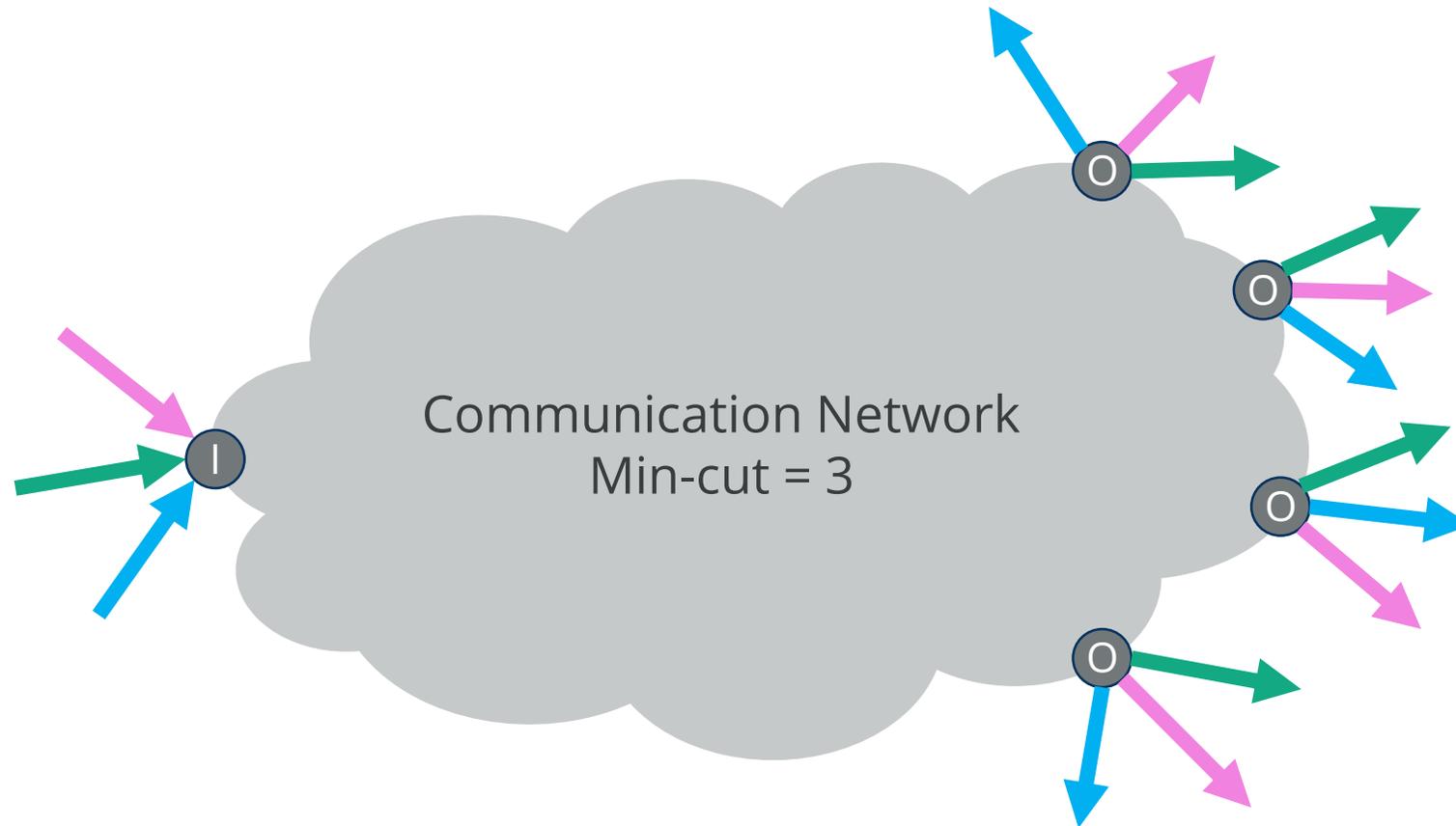


# Ford-Fulkerson Max Flow



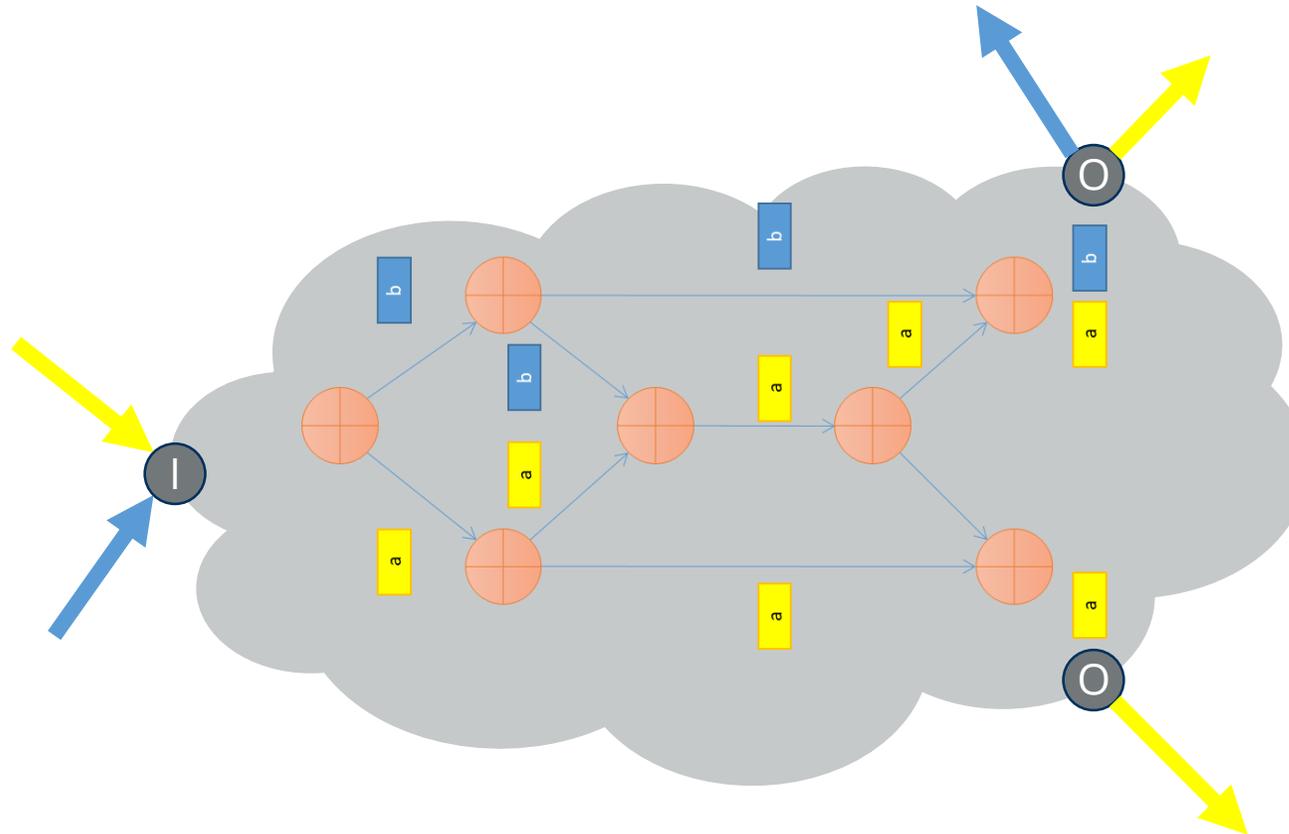
# Multi-Cast Example

# Multi-Cast Example



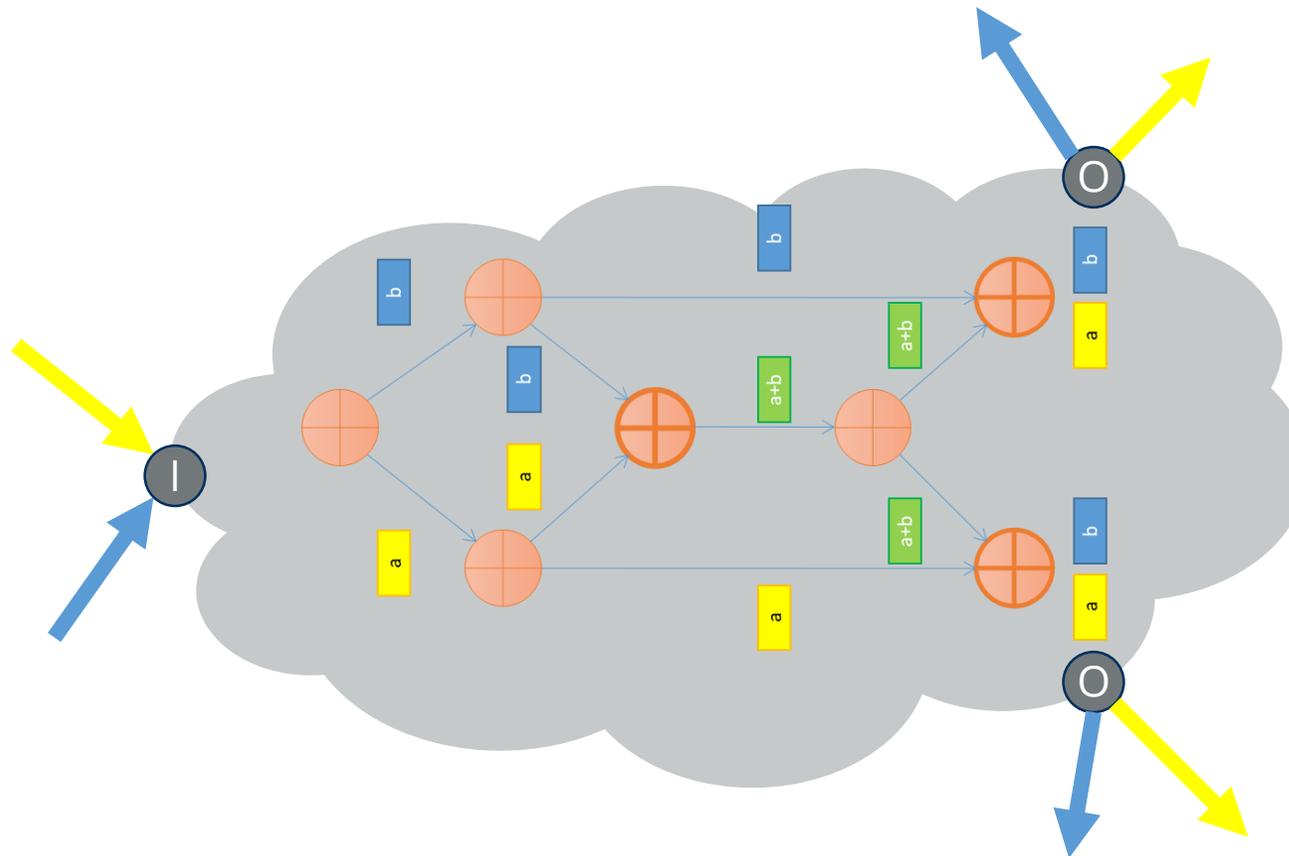
Compare Routing with Network Coding

# Multi-Cast Example



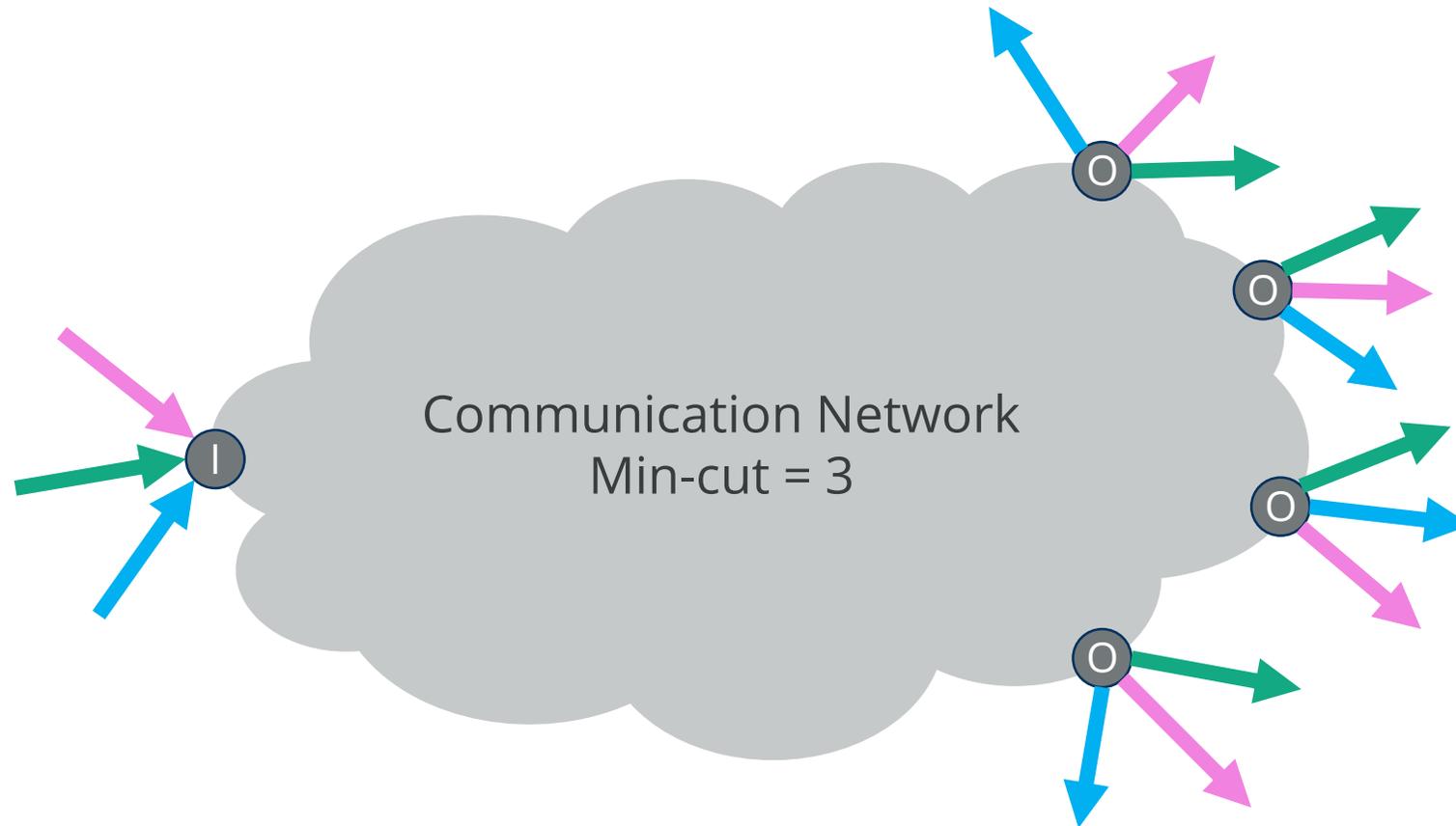
Compare Routing with Network Coding  
Routing: np-hard; not optimal with respect to capacity

# Multi-Cast Example



Compare Routing with Network Coding  
Network Coding: optimal solution can be found and can be found fast

# Multi-Cast Example

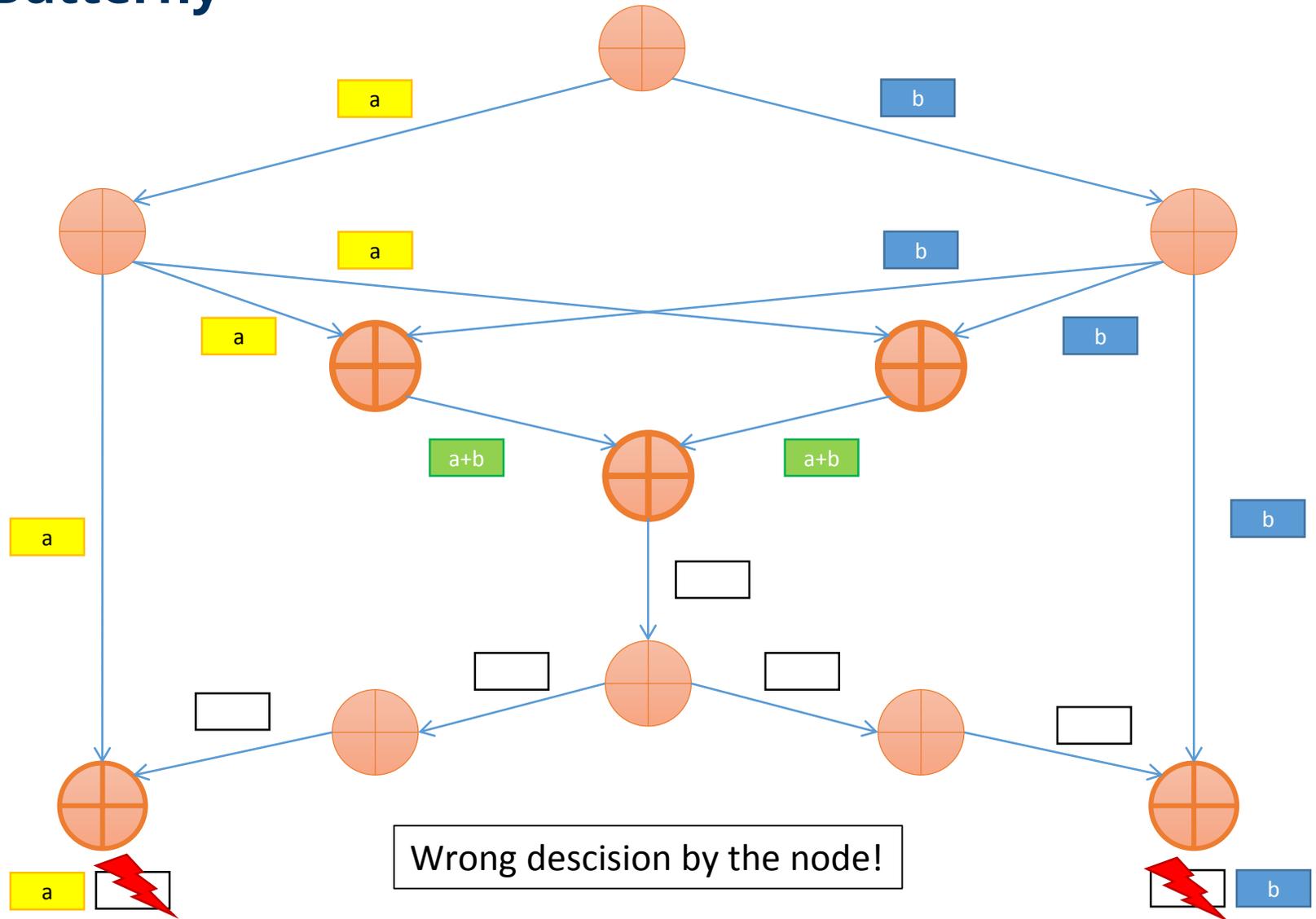


Compare Routing with Network Coding

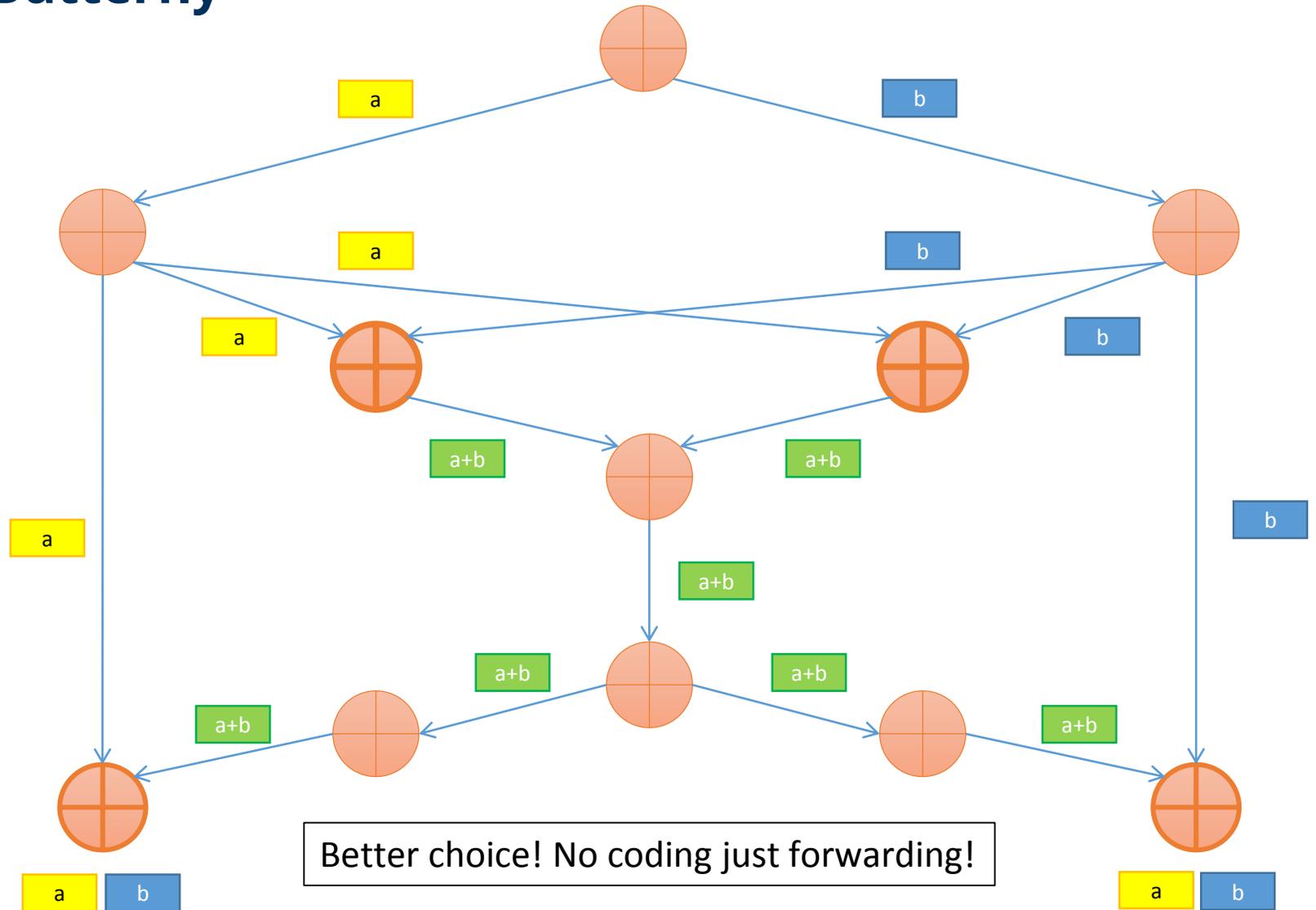
# The Butterfly++

# Network Coding: The Butterfly++

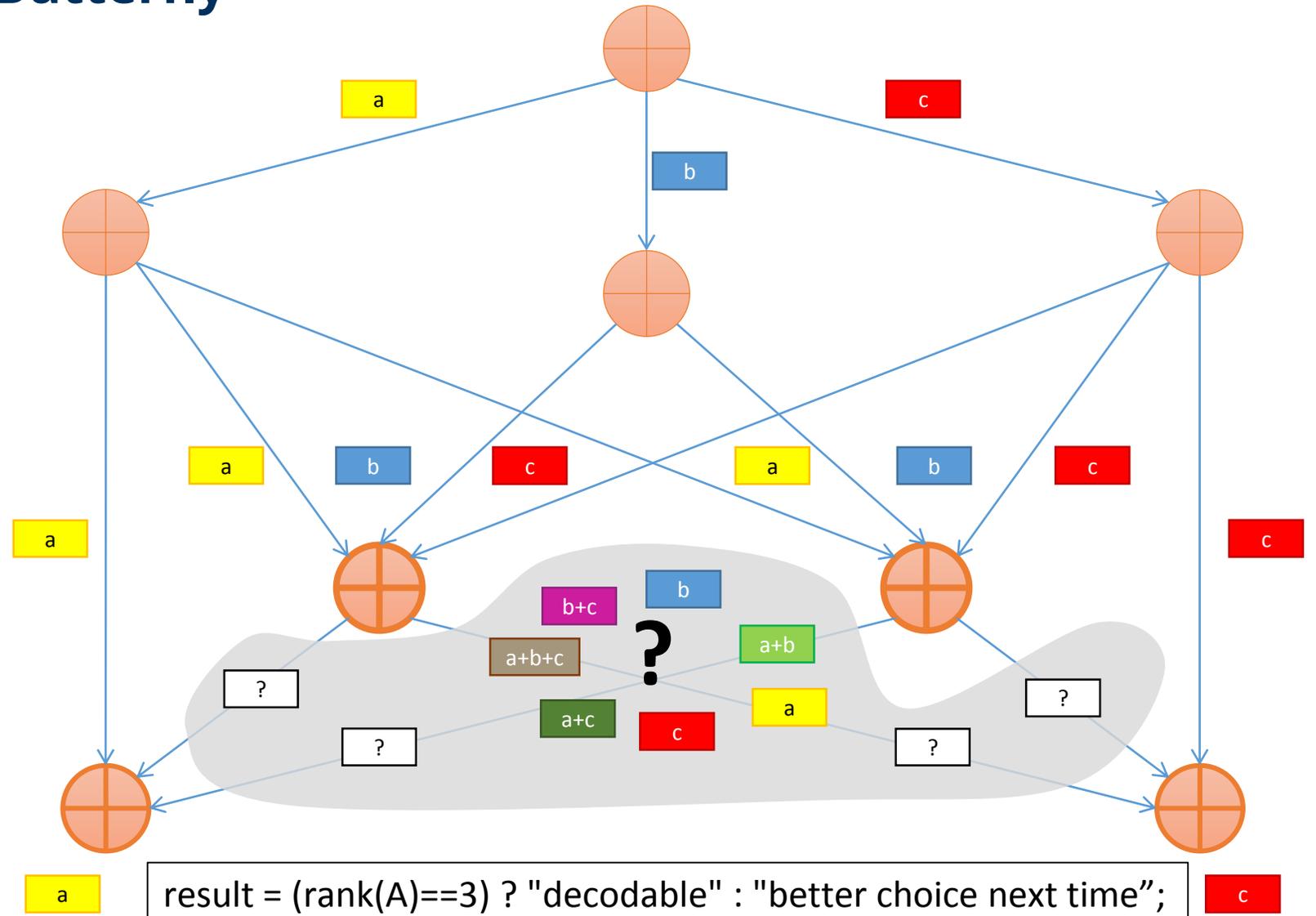
Let's code all incoming packets ... mmmh



# Network Coding: The Butterfly++



# Network Coding: The Butterfly++

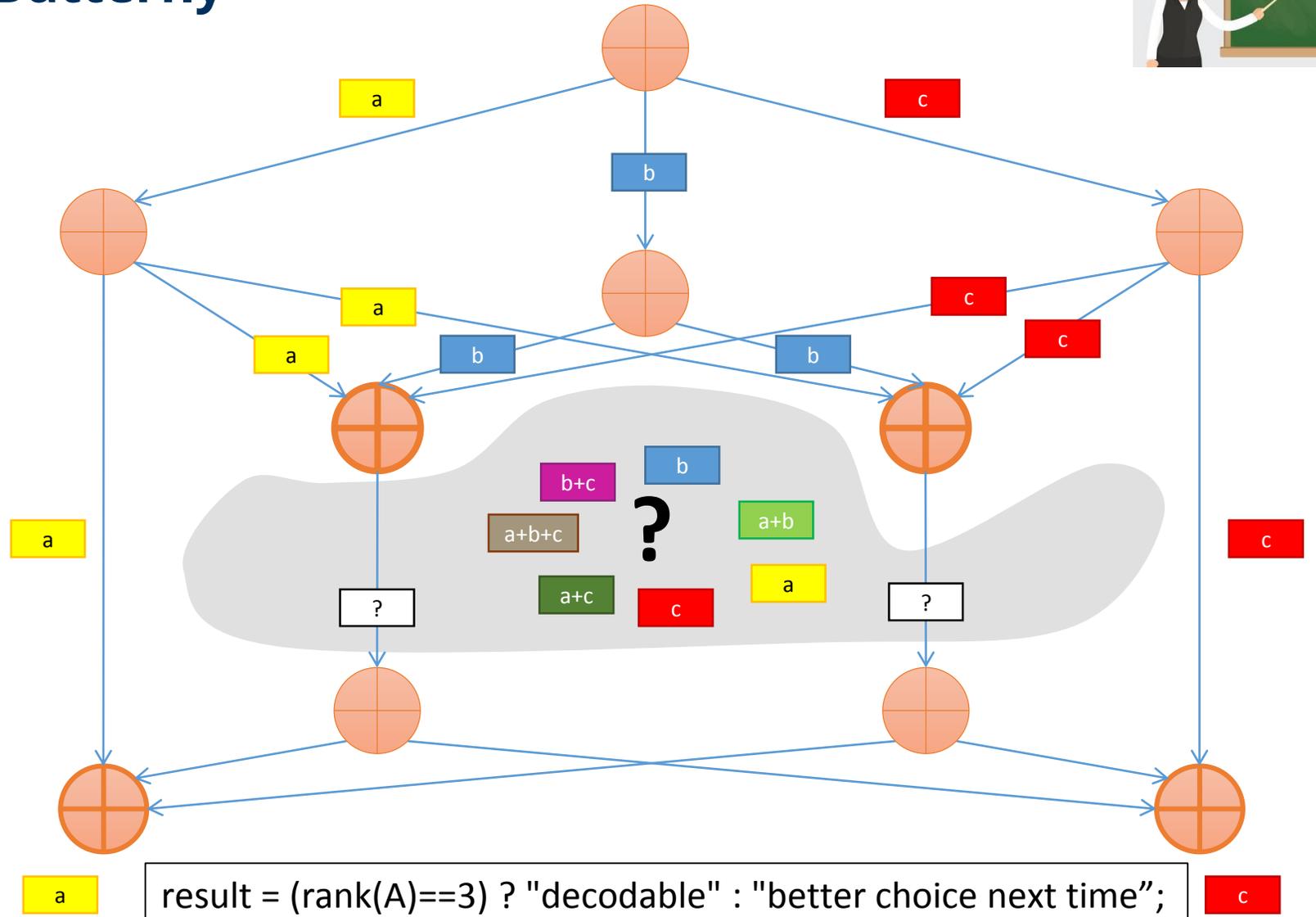




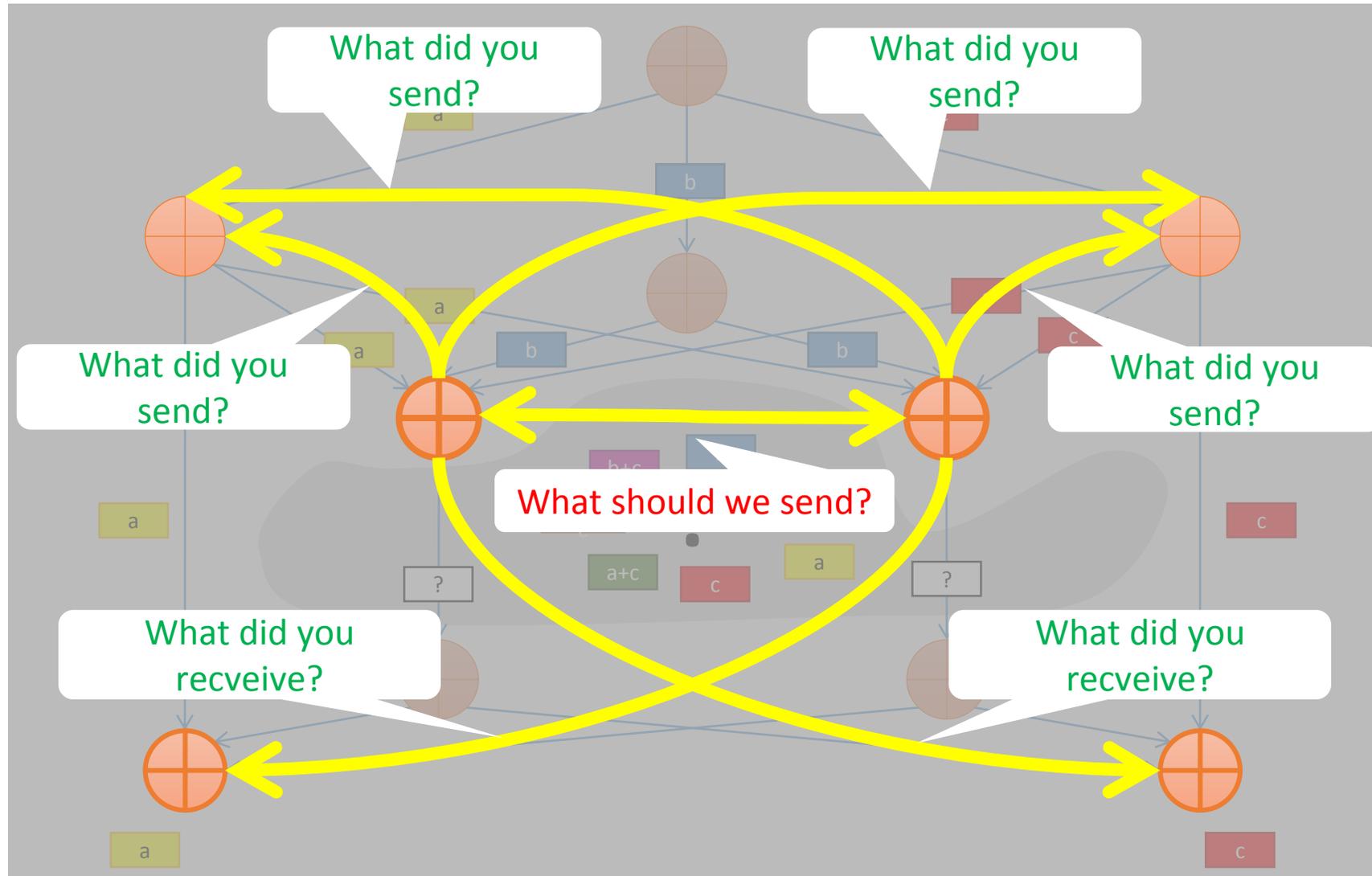
# Network Coding: The Butterfly++

$$\begin{pmatrix} 1 & 0 & 0 \\ c_4 & c_5 & c_6 \\ c_1 & c_2 & c_3 \end{pmatrix}$$

$$\begin{pmatrix} c_1 & c_2 & c_3 \\ c_4 & c_5 & c_6 \\ 0 & 0 & 1 \end{pmatrix}$$



# Network Coding: The Butterfly++



# Deterministic Network Coding

Deterministic Network Coding refers to a specific method for network code design. I.e. exactly specifying how in-put data is mapped to output data for all nodes in a network. This is in contrast to Random Network Coding.

## Advantages

- Coding coefficients are known and therefore not required to be explicitly communicated.

## Drawbacks

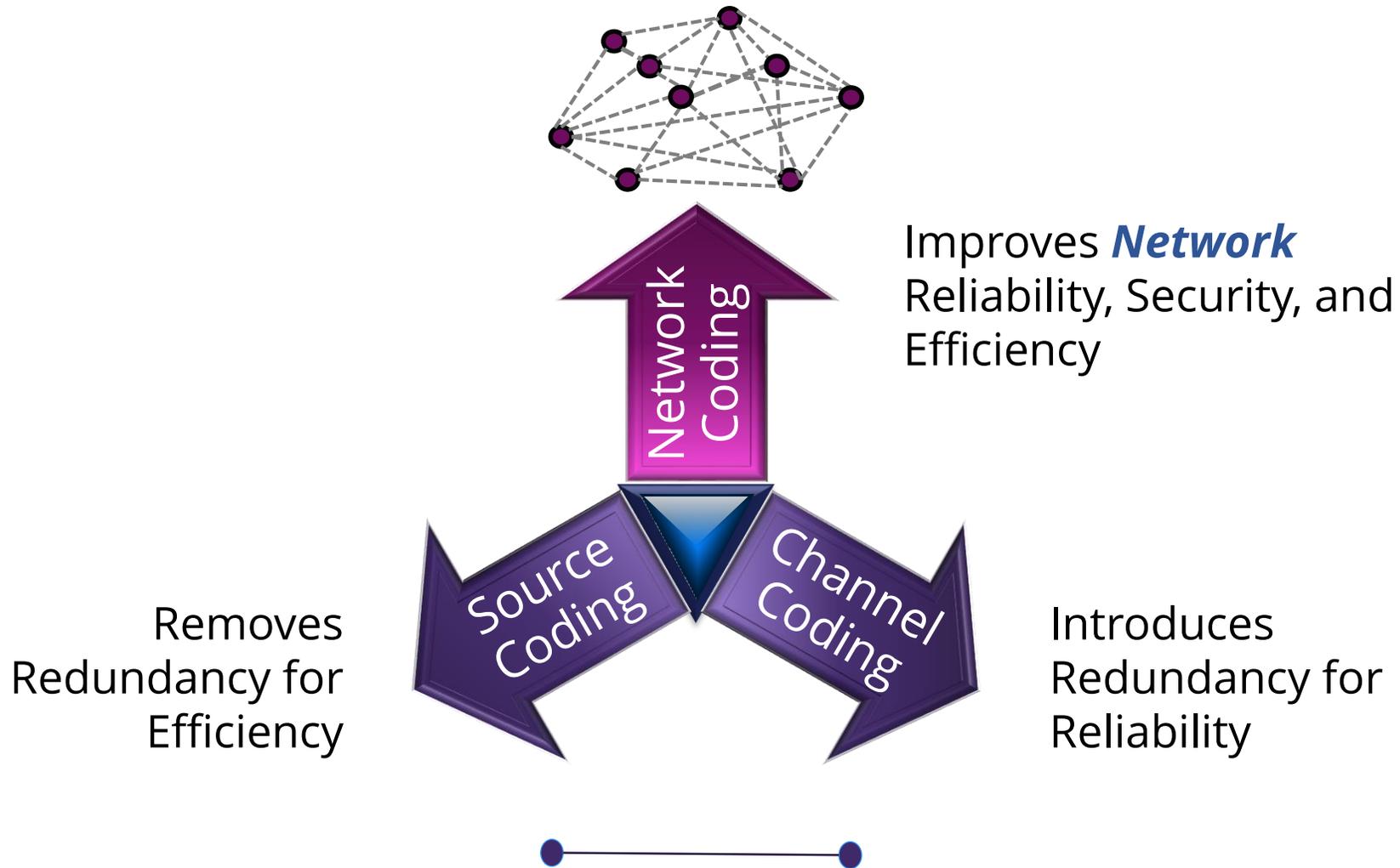
- Algorithms often require that the exact and full topology as input.
- Dynamic networks will require frequent updates, to reflect current state of the network.

# Digital Inter-Flow Network Coding: The Basics

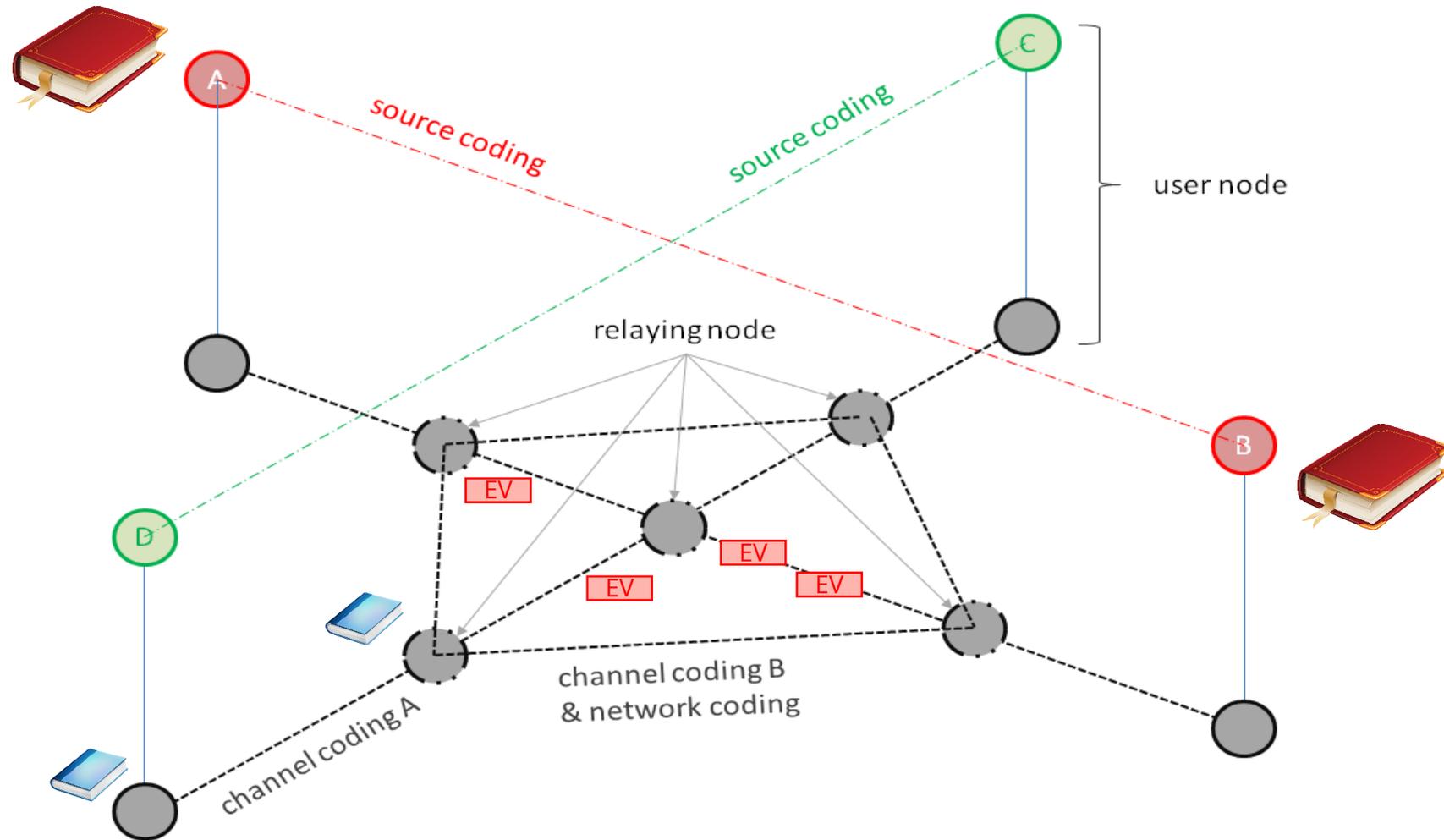
Lecture 2

# Channel & Source Coding vs. Network Coding

# Comparison of Coding Approaches



# Comparison of Coding Approaches



# Wireless Network Examples

Exploiting the broadcast nature of the wireless medium

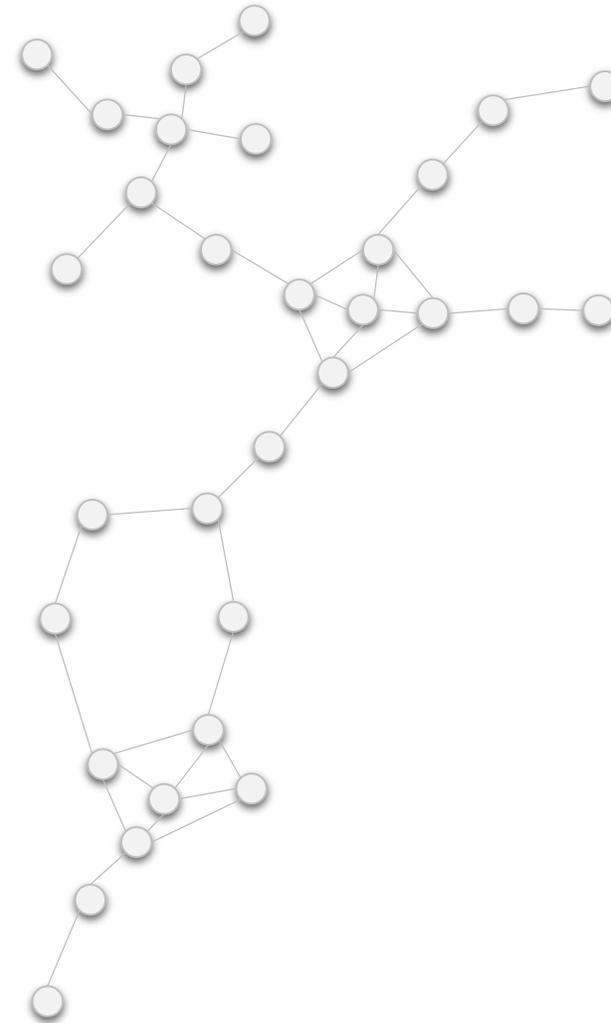
## It is not about butterflies ...



M. Medard and F.H.P. Fitzek and M.J. Montpetit and C. Rosenberg. Network coding mythbusting: why it is not about butterflies anymore. 2014. IEEE Communications Magazine, 52(7):177-183.

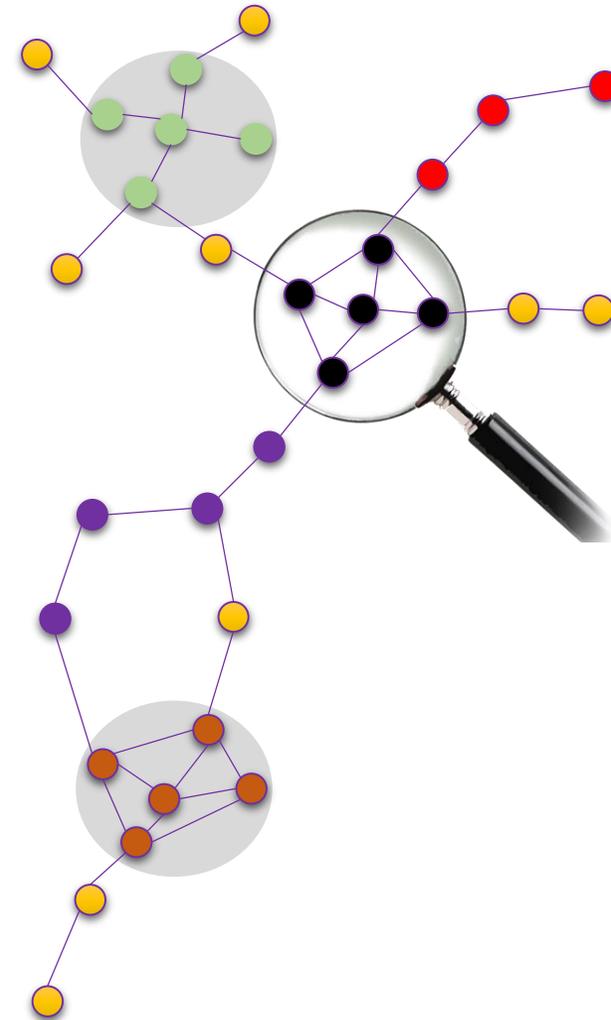
# General Network Topologies and Sub-Nets

- Two way relay (Alice and Bob)
- Chain
- X- Topology
- Cross
- Cross with Overhearing



# General Network Topologies and Sub-Nets

- Two way relay (Alice and Bob)
- Chain
- X- Topology
- Cross
- Cross with Overhearing



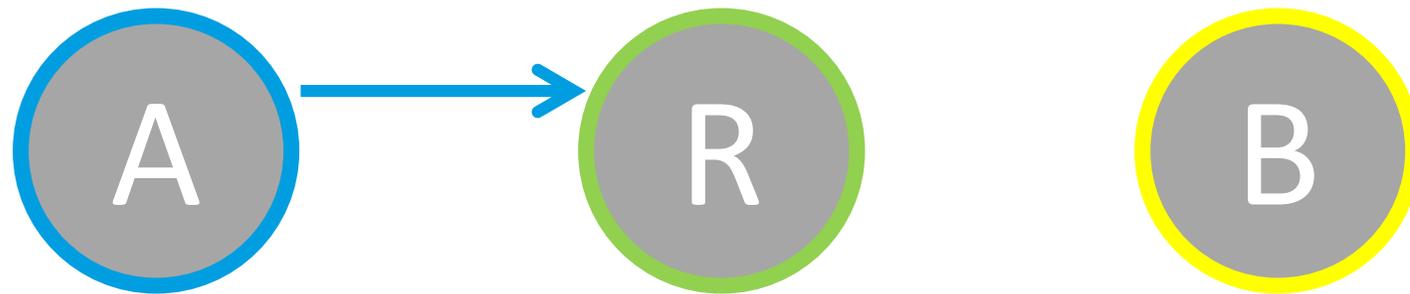
# **Alice and Bob –The two way relay**

Early VTC paper by Ericsson

# Alice and Bob



# Alice and Bob: Forwarding



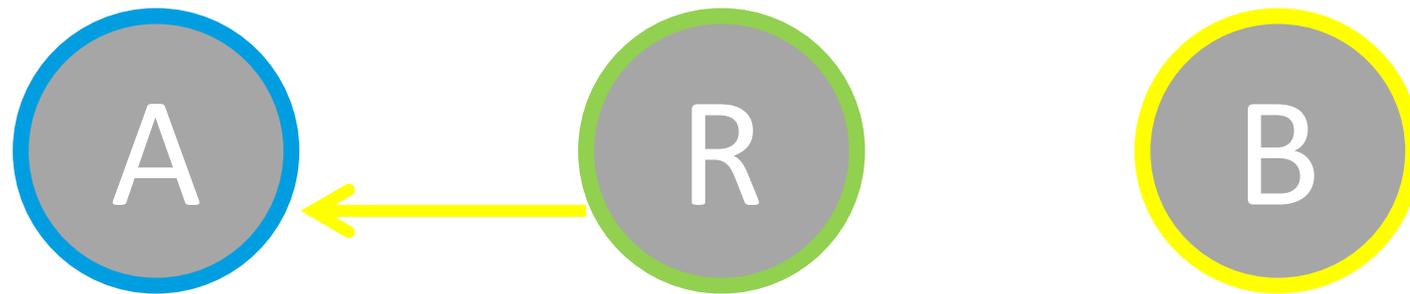
# Alice and Bob: Forwarding



# Alice and Bob: Forwarding

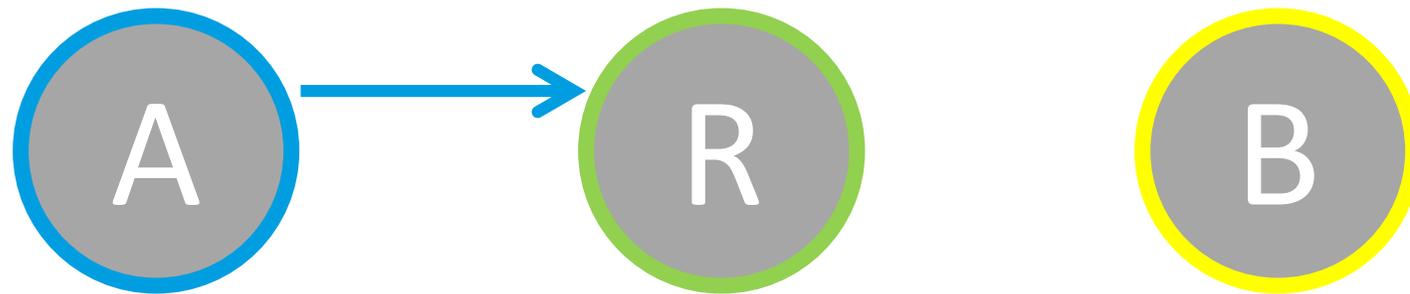


# Alice and Bob: Forwarding



4 time slots for exchange

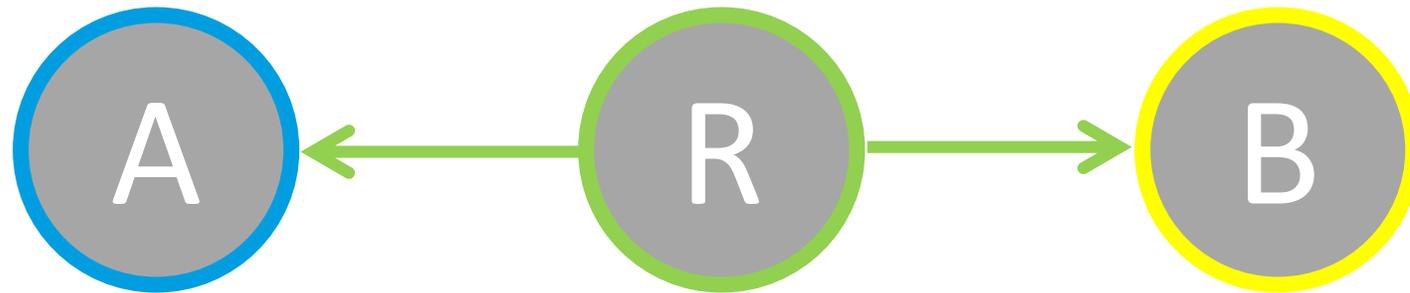
# Alice and Bob: Network Coding



# Alice and Bob: Network Coding



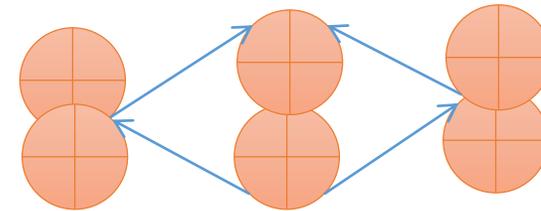
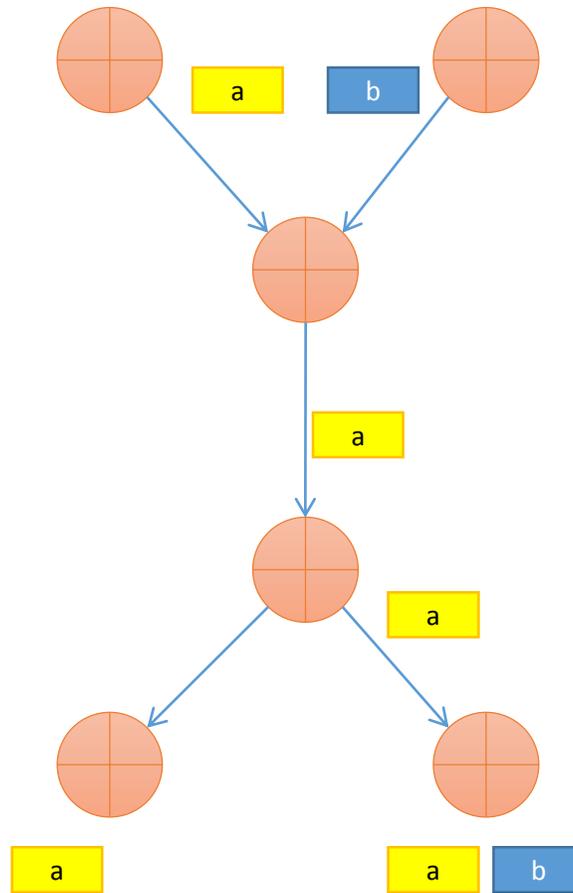
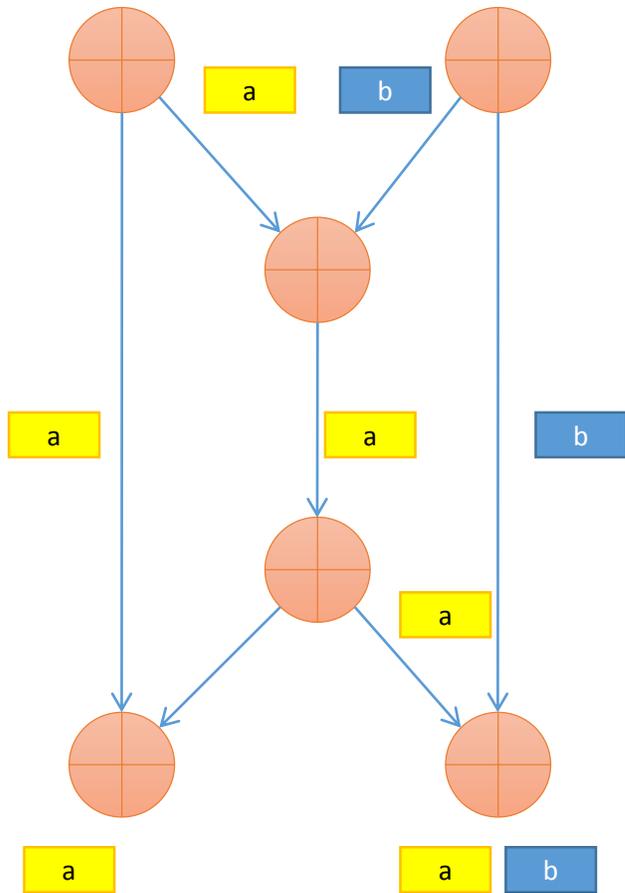
# Alice and Bob: Network Coding



3 time slots for exchange

**Gain = 33%**

# Two-Way Relay (Alice and Bob) vs. Butterfly



# Network Coding: Magic XOR

M.V. Pedersen and F.H.P. Fitzek and T. Larsen. **Implementation and Performance Evaluation of Network Coding for Cooperative Mobile Devices**. 2008. in *IEEE International Conference on Communications (ICC 2008) - CoCoNet Workshop*.

# Alice and Bob: Symbian60

Our starting point

Simple scenario

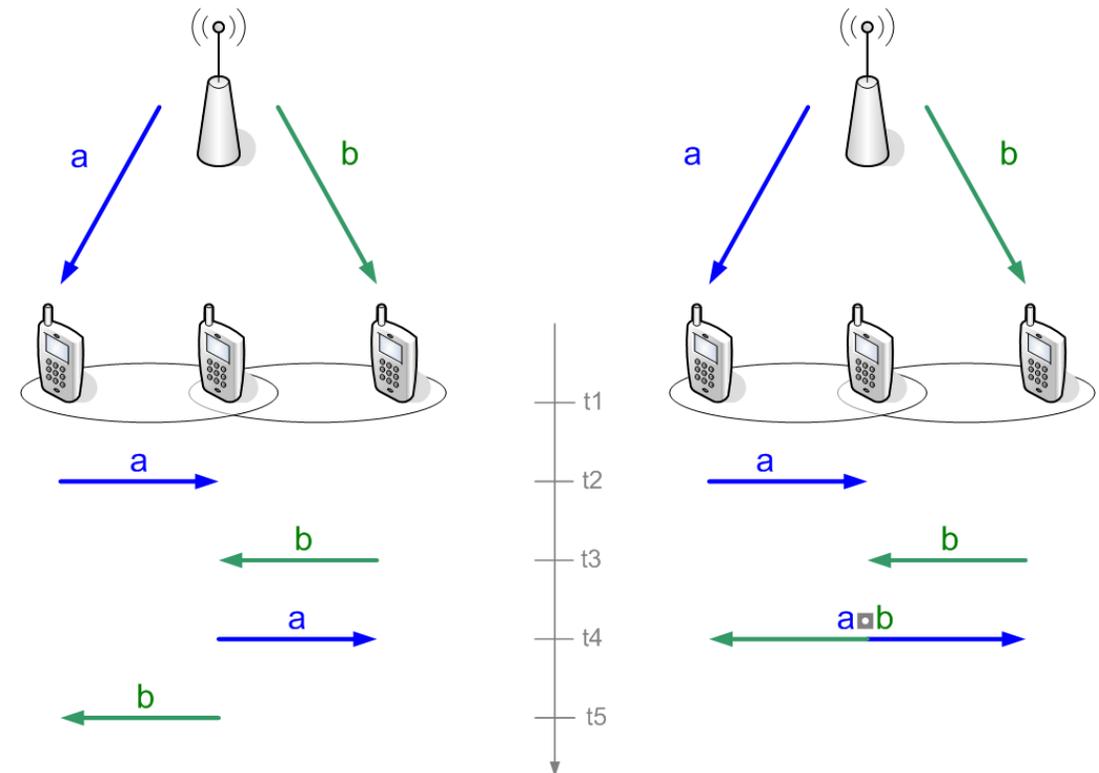
Seeding of packet a and b is crucial

- Fairness
- Performance

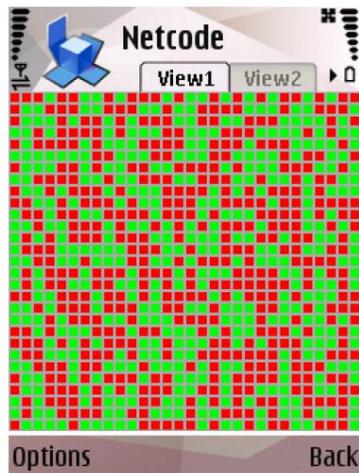
Forms of NC

- XOR in the air (COPE)

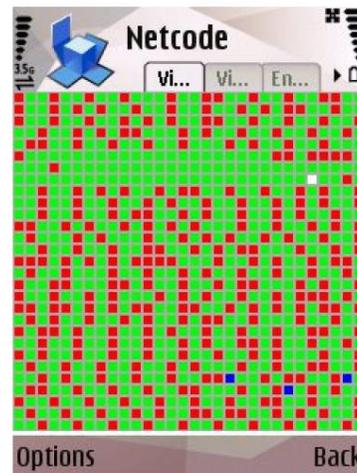
Implementation on S60



# Alice and Bob: Symbian60

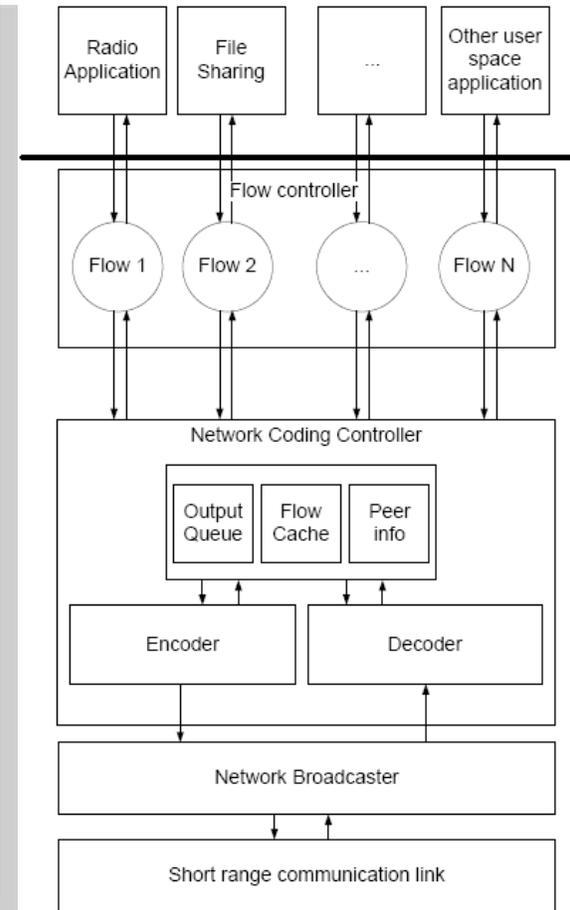


(a)

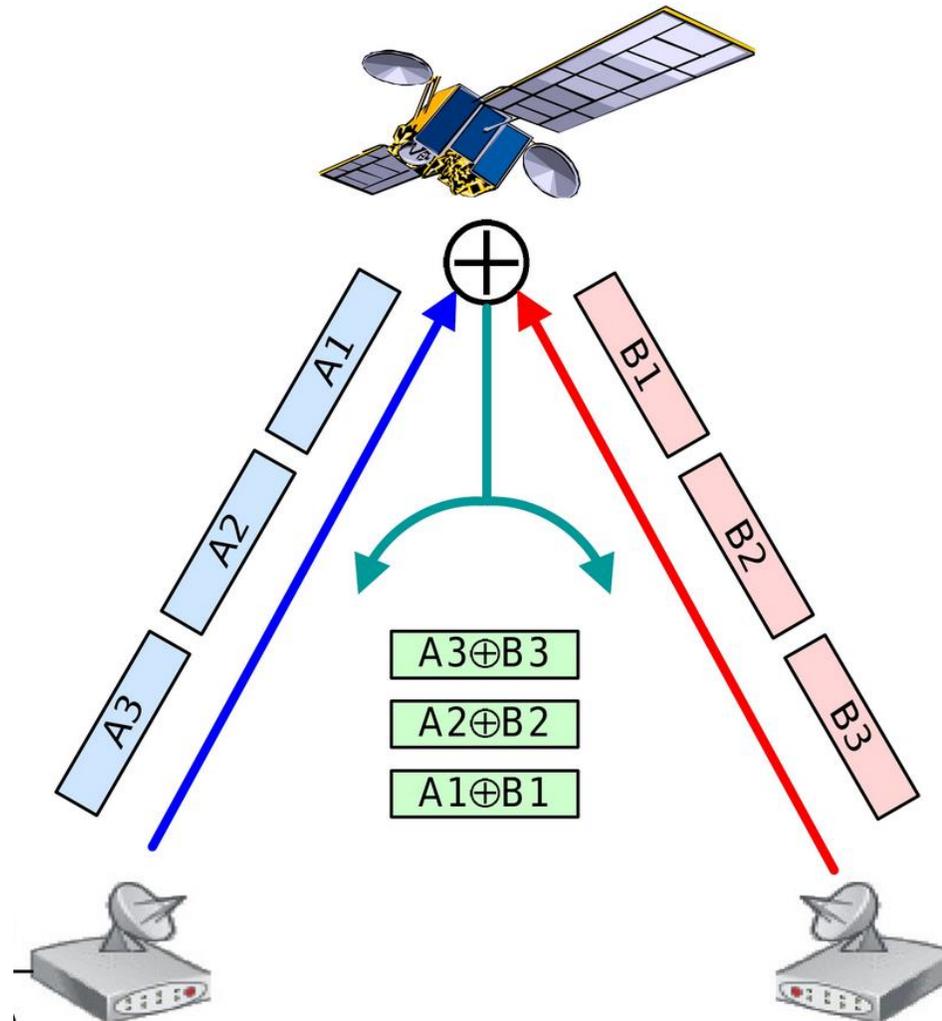


(b)

Lessons learned: It is important to create coding potential!



# Alice and Bob: Satellite Communication



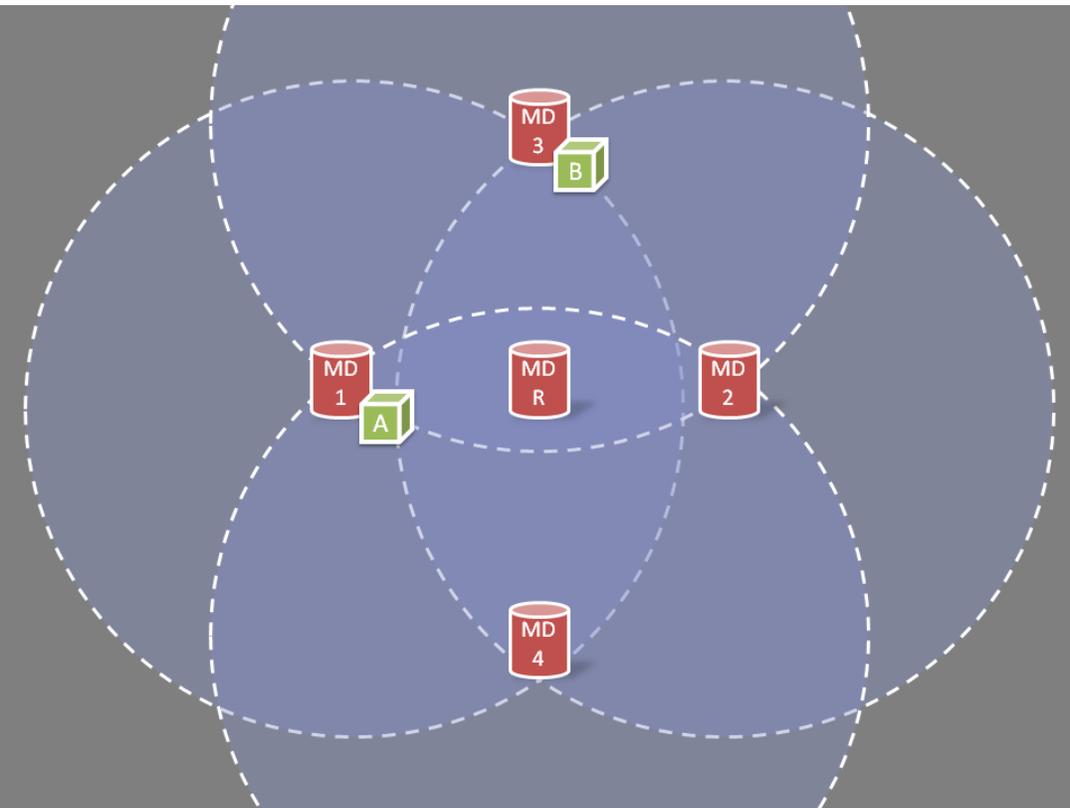
[http://www.dlr.de/kn/Portaldata/27/Resources/images/projekte/NEXT-Network\\_coded\\_interconnection\\_of\\_computer\\_networks\\_via\\_satellite.jpg](http://www.dlr.de/kn/Portaldata/27/Resources/images/projekte/NEXT-Network_coded_interconnection_of_computer_networks_via_satellite.jpg)

# The X-Topology

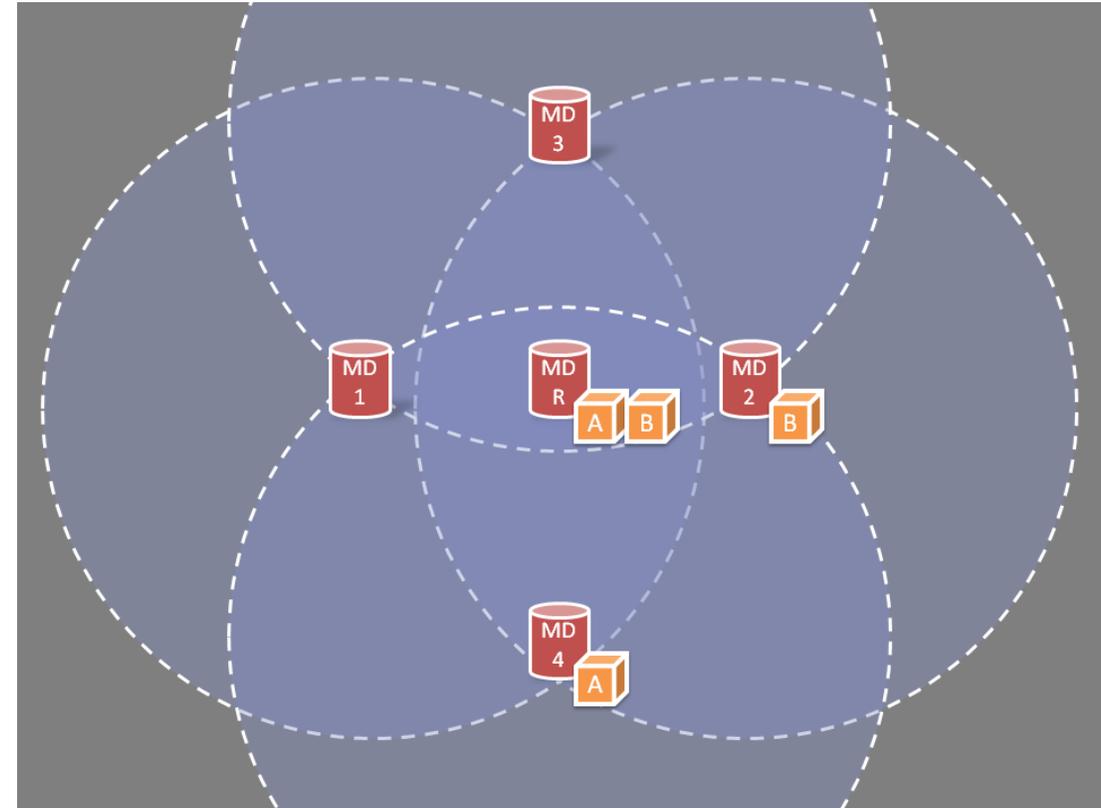
Introducing the concept of overhearing

# X Topology with Overhearing

Start

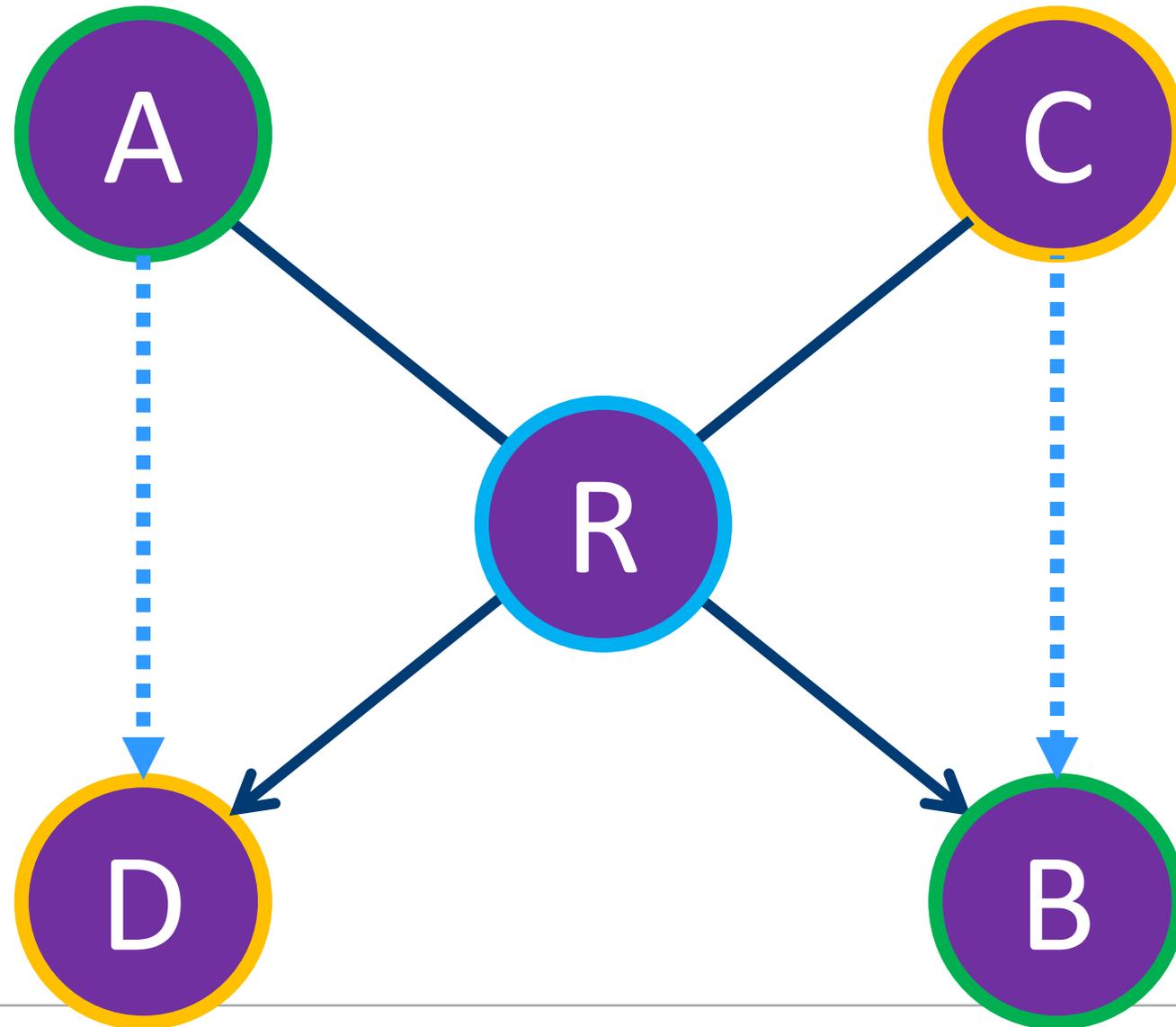


Broadcast

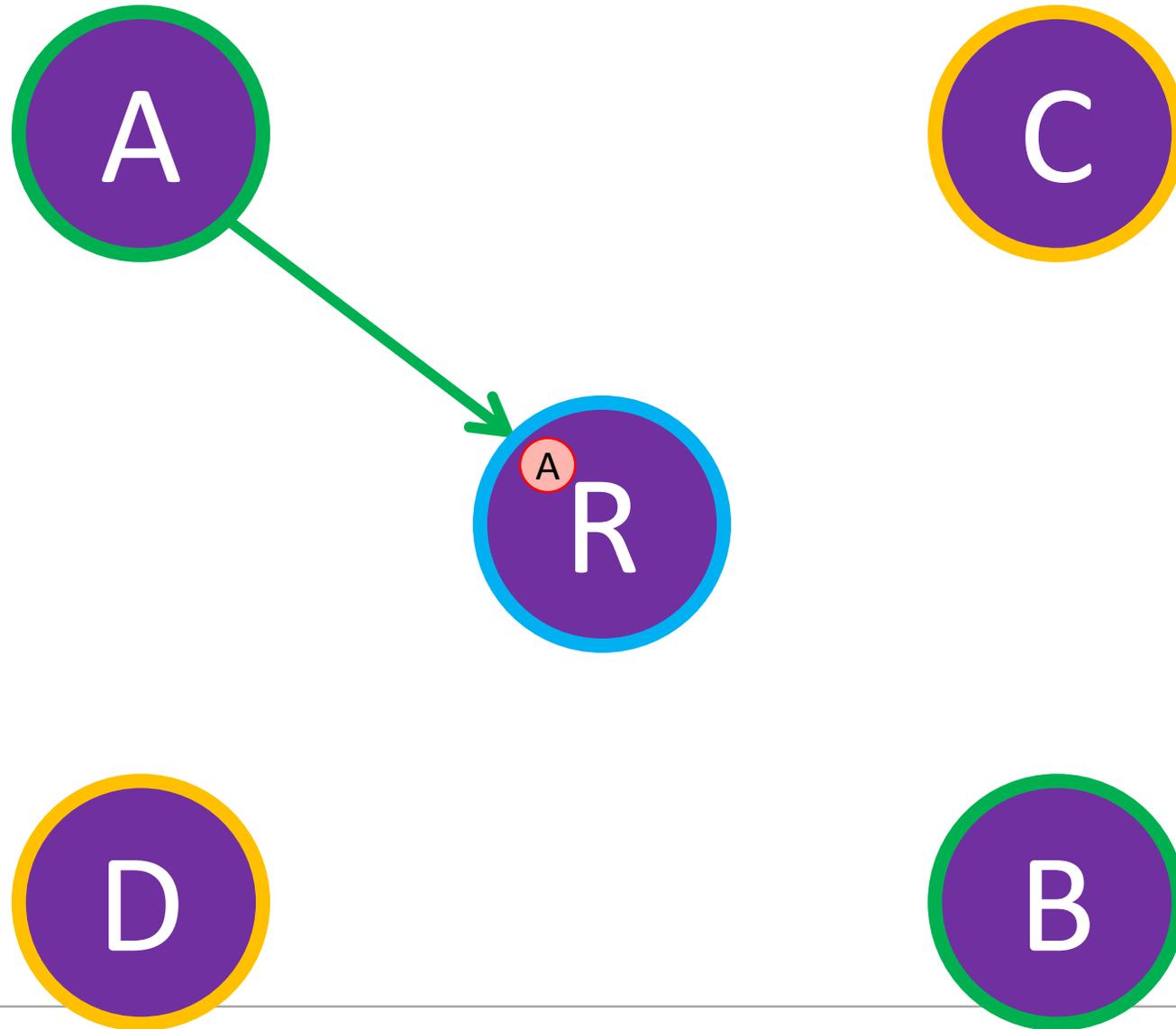




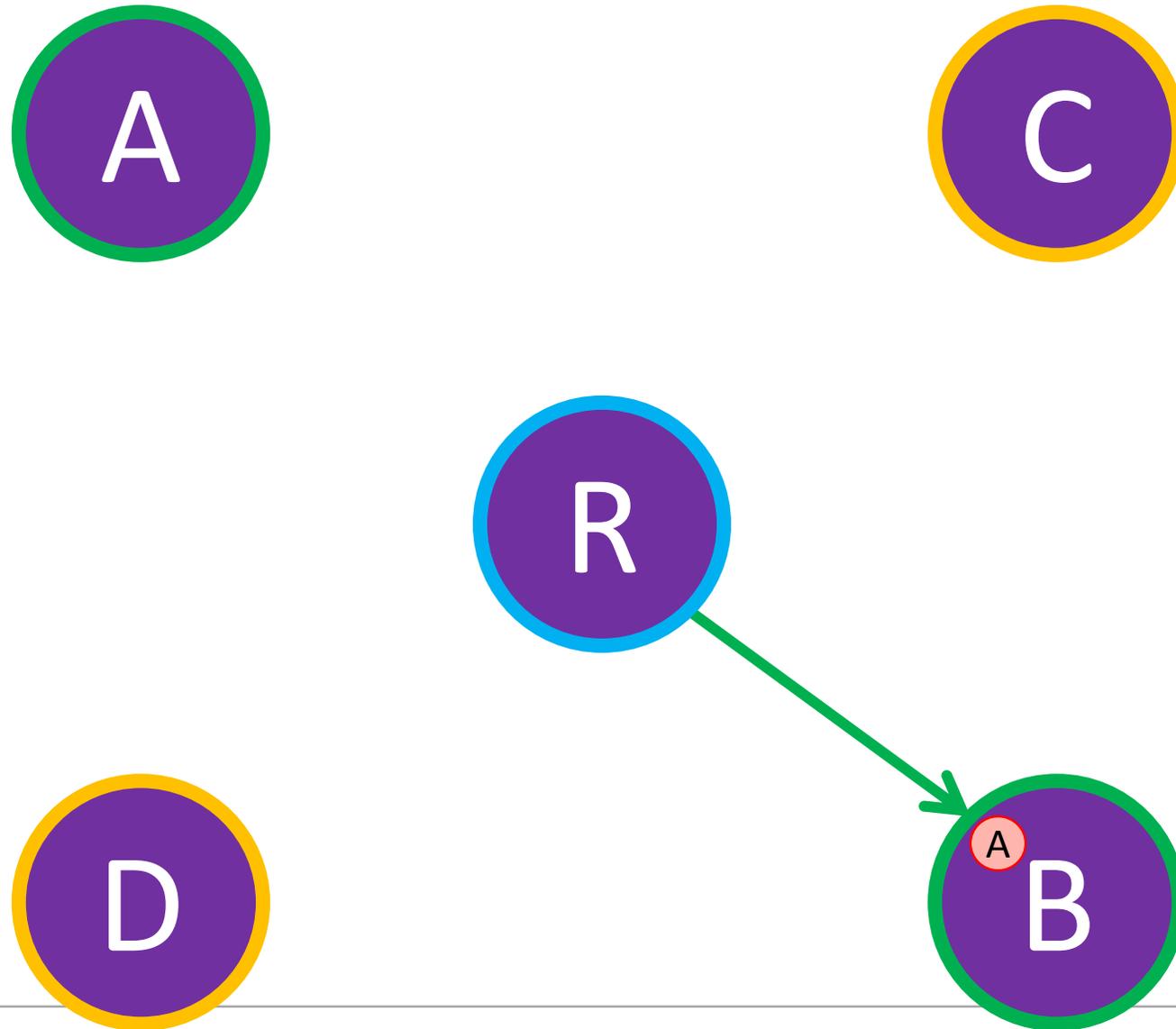
# X Topology with Overhearing



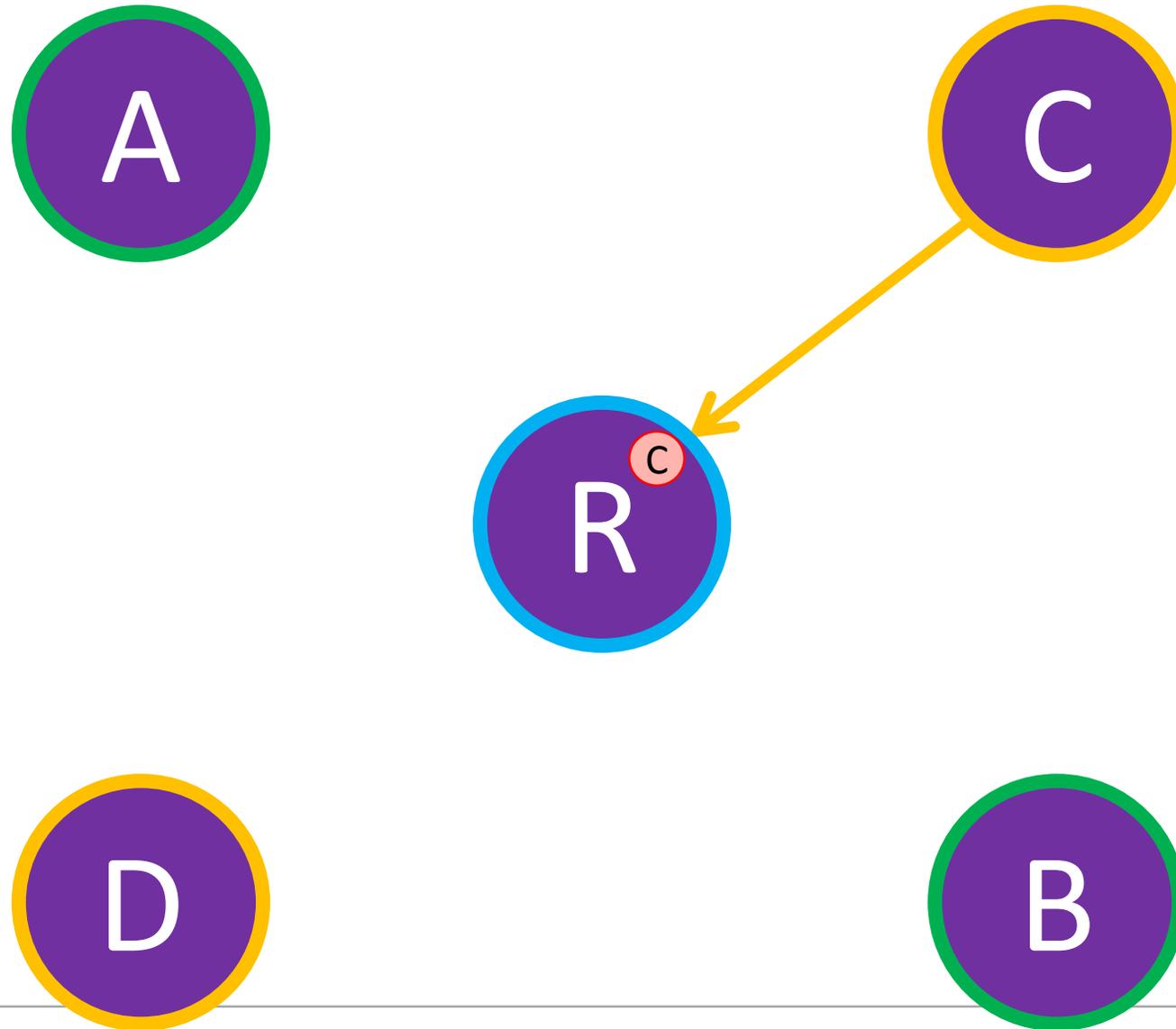
# X Topology: SoA



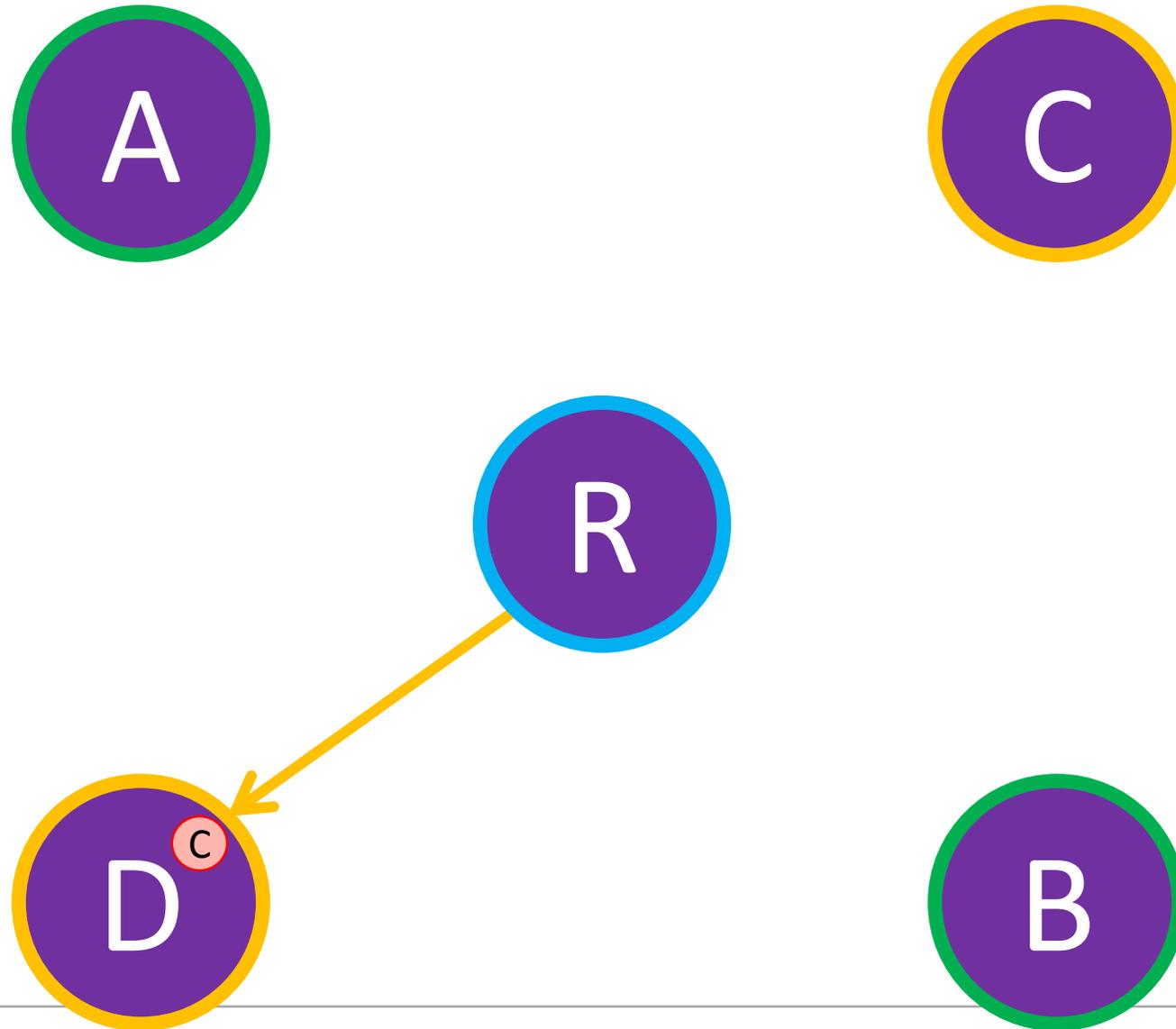
# X Topology: SoA



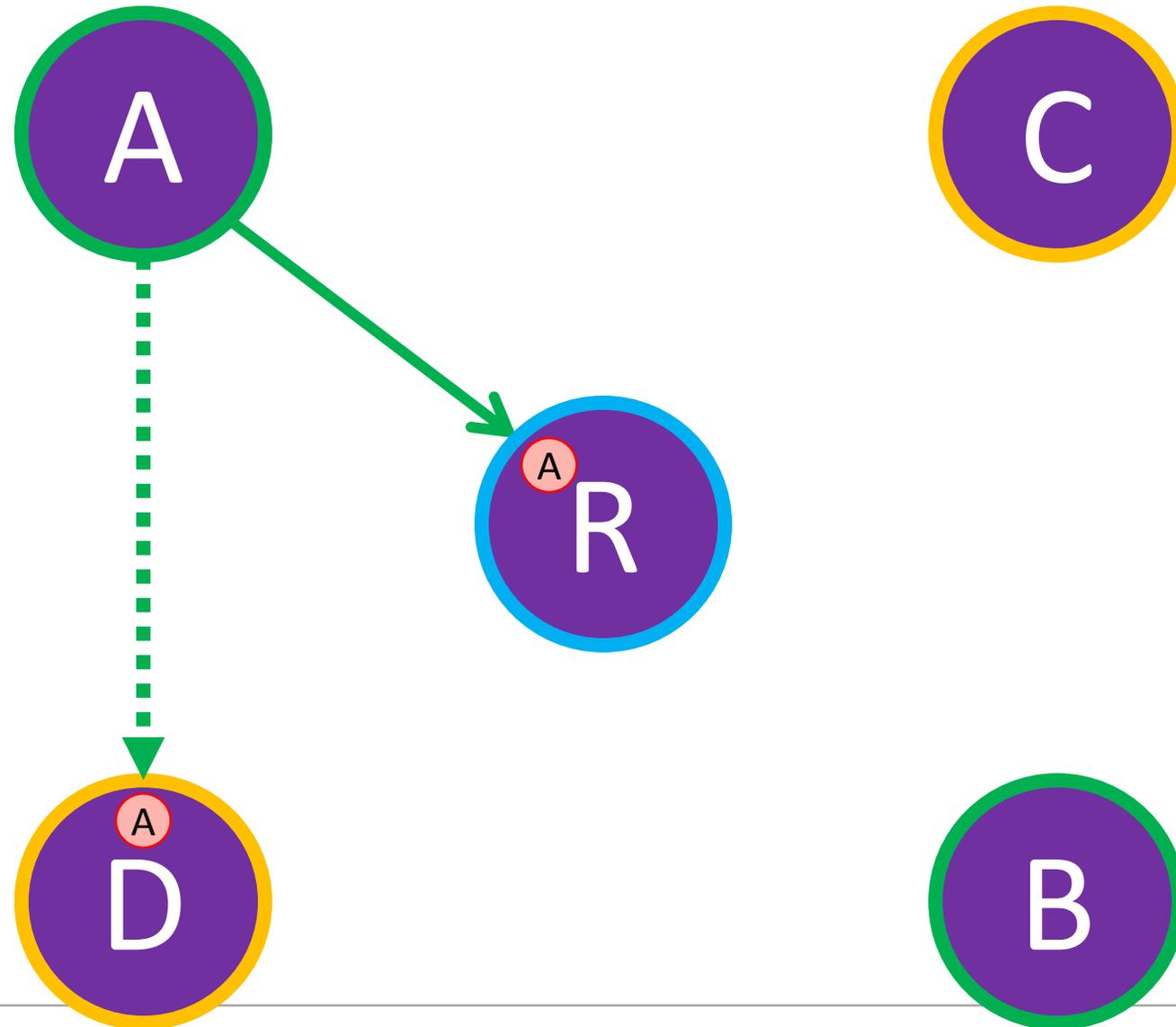
# X Topology: SoA



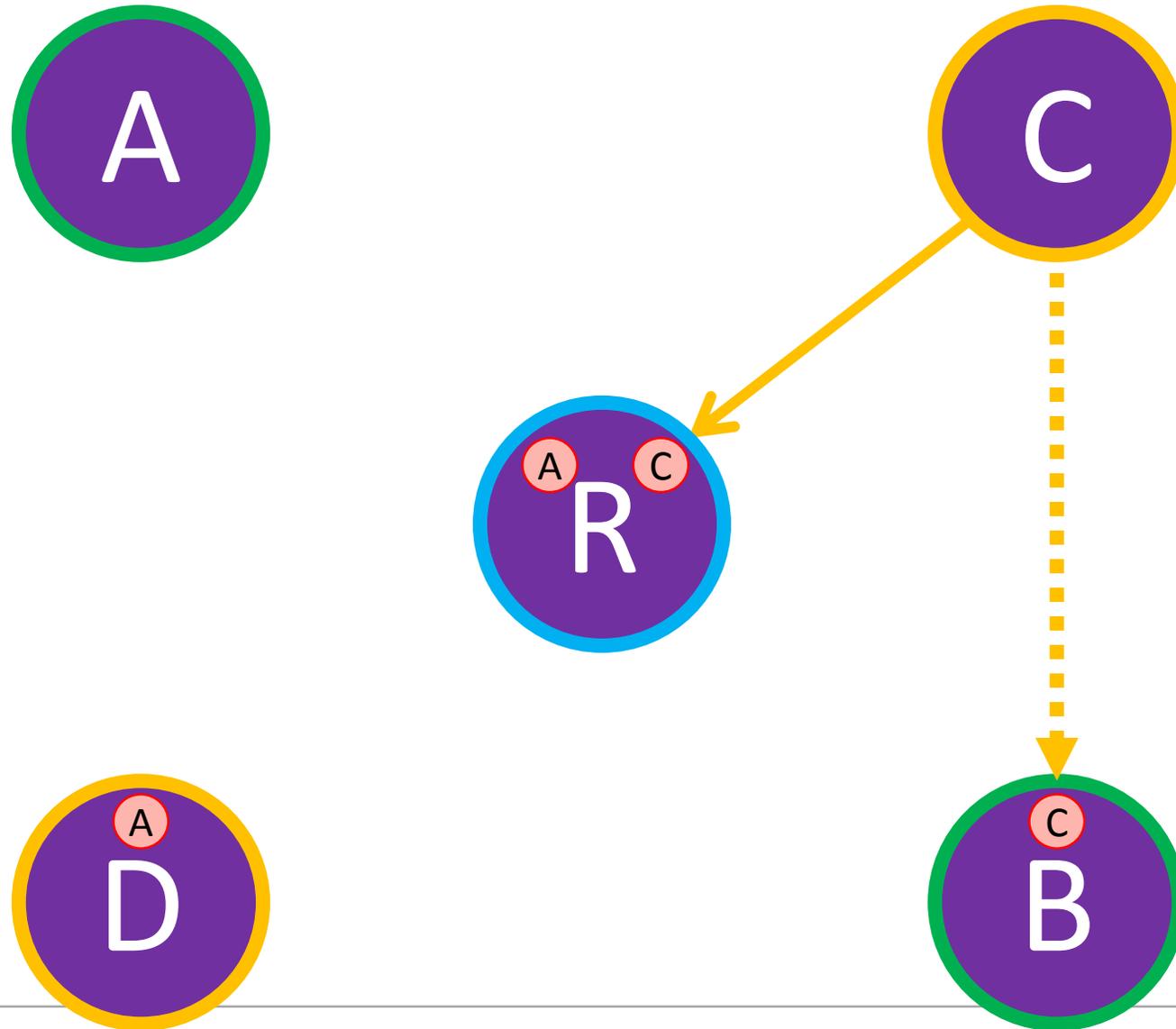
# X Topology: SoA



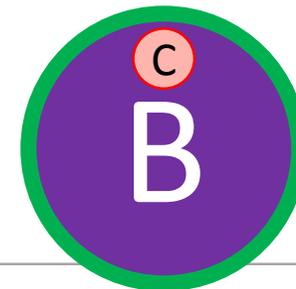
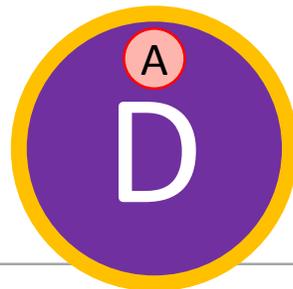
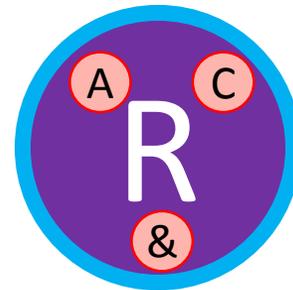
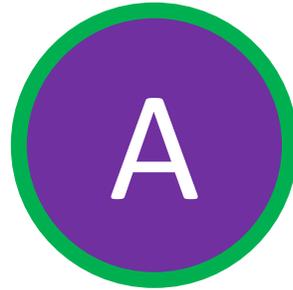
# X Topology with Overhearing



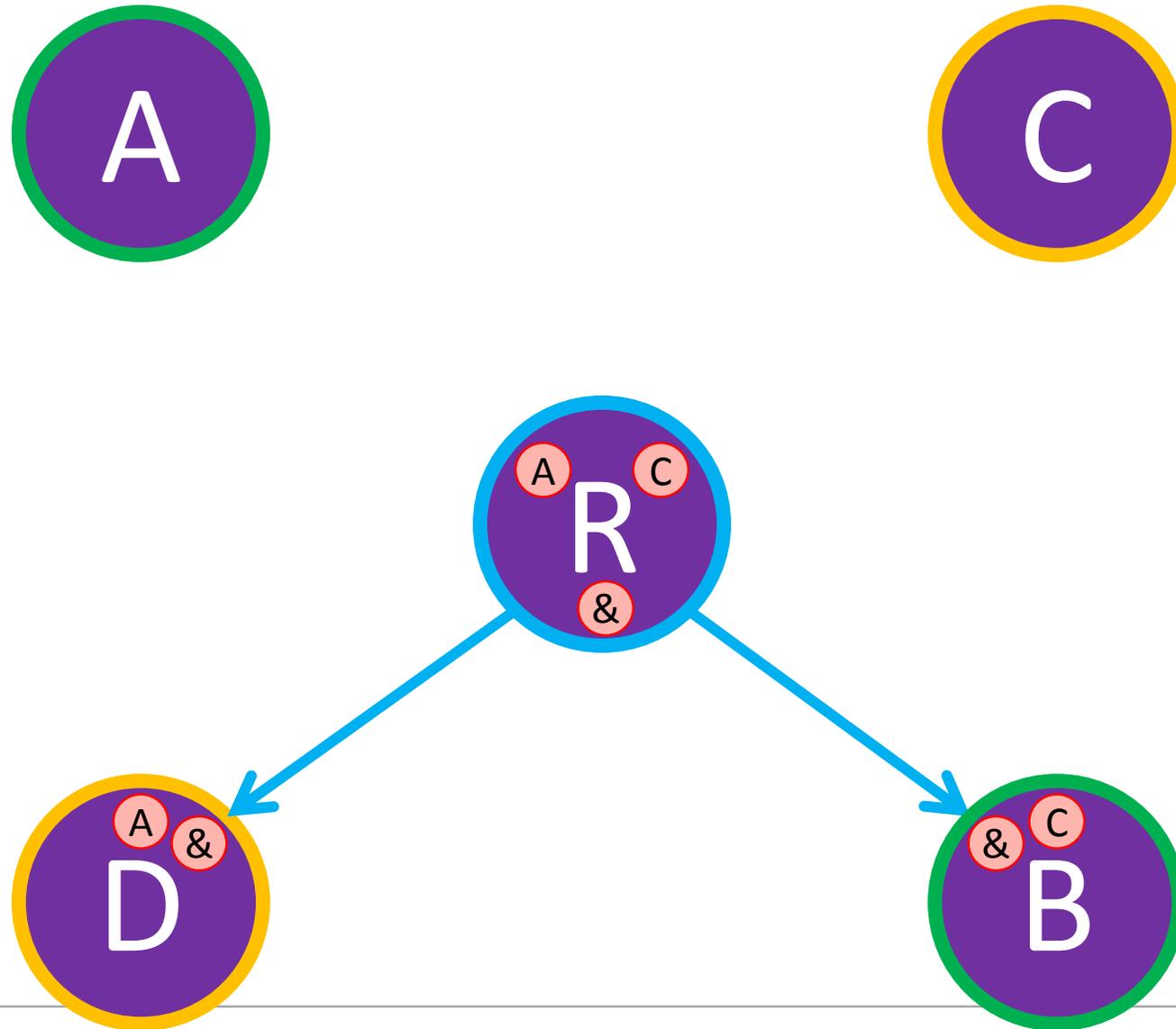
# X Topology with Overhearing



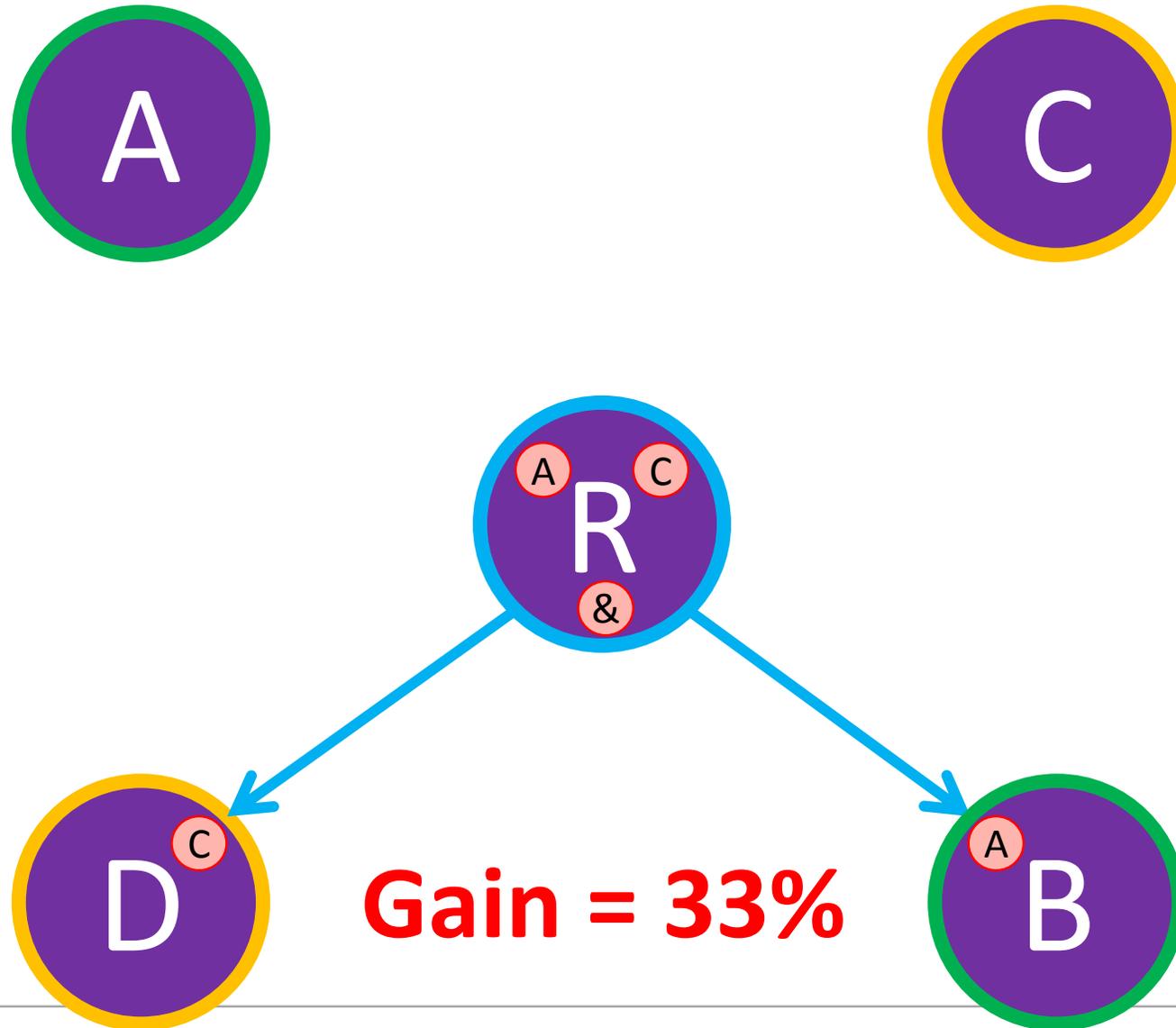
# X Topology with Overhearing



# X Topology with Overhearing

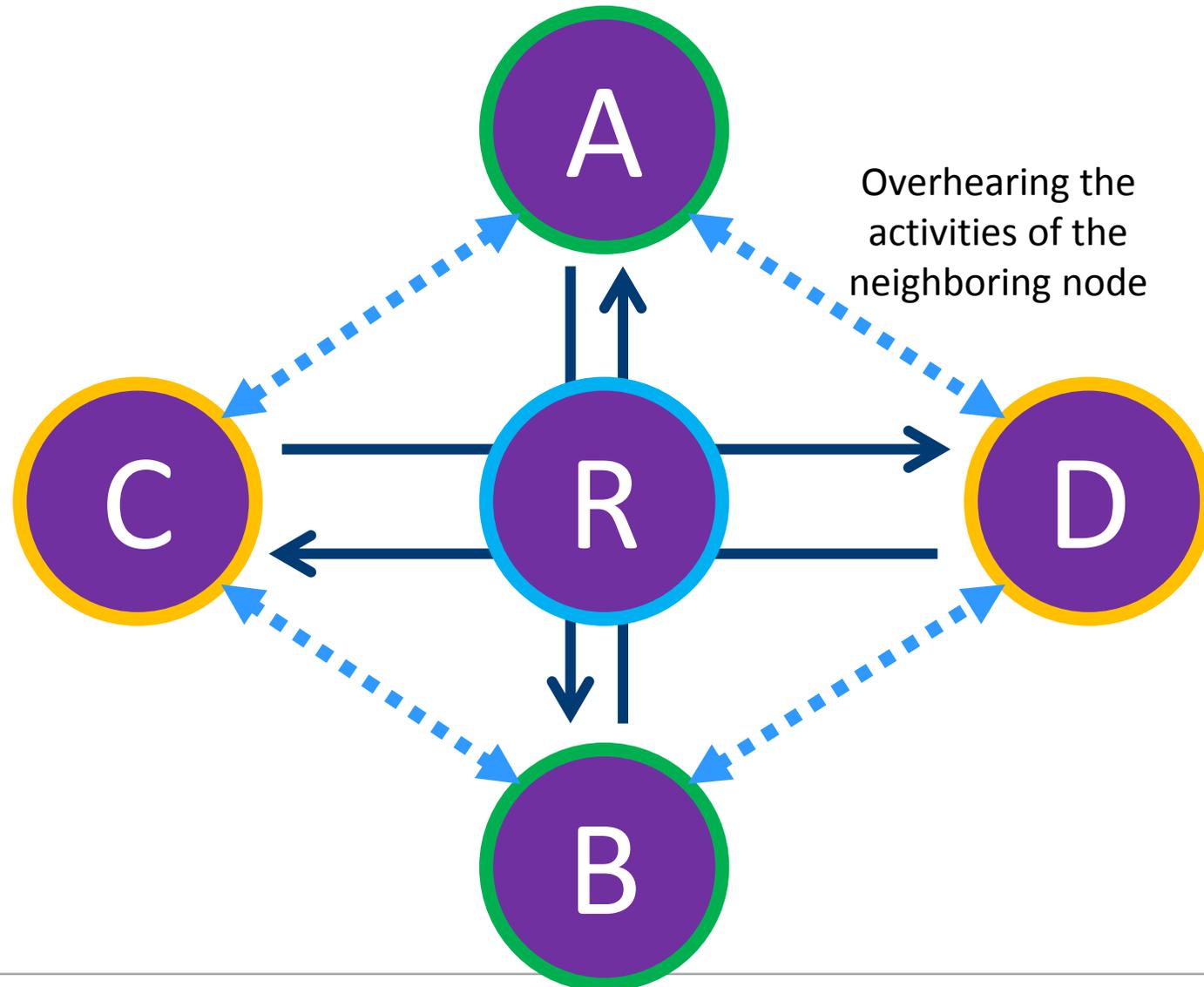


# X Topology with Overhearing

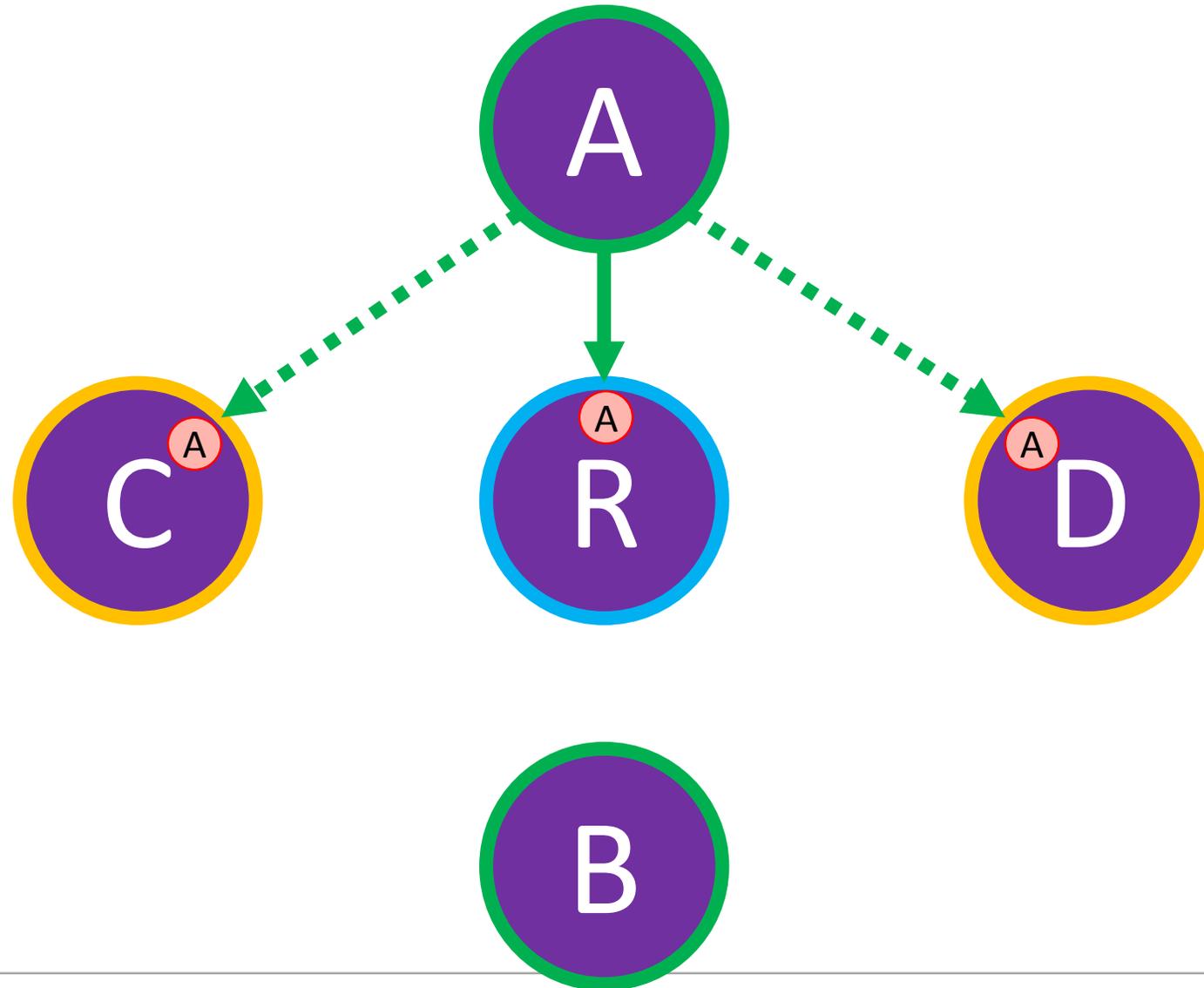


# The Cross Topology

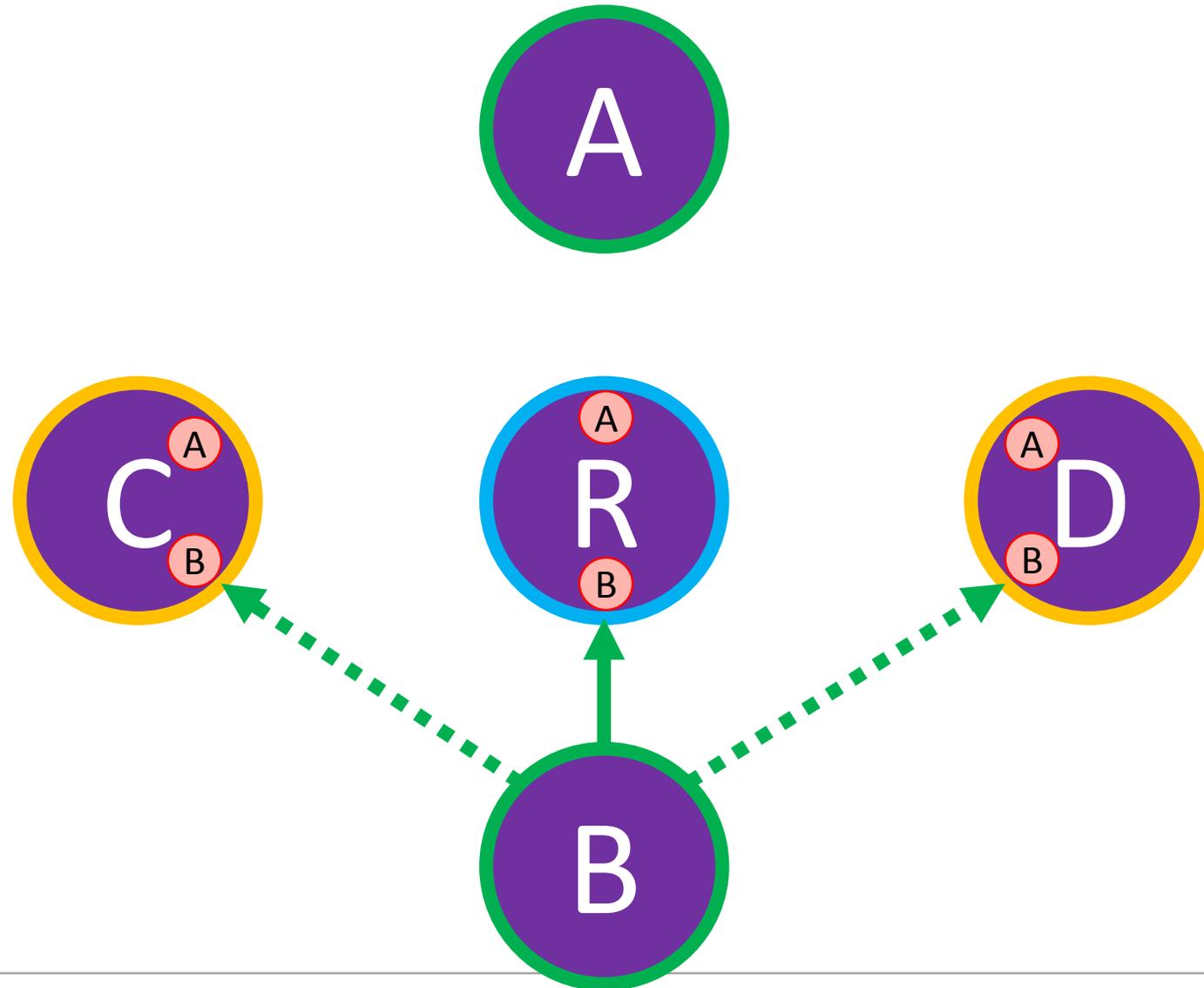
# Cross Topology with Overhearing



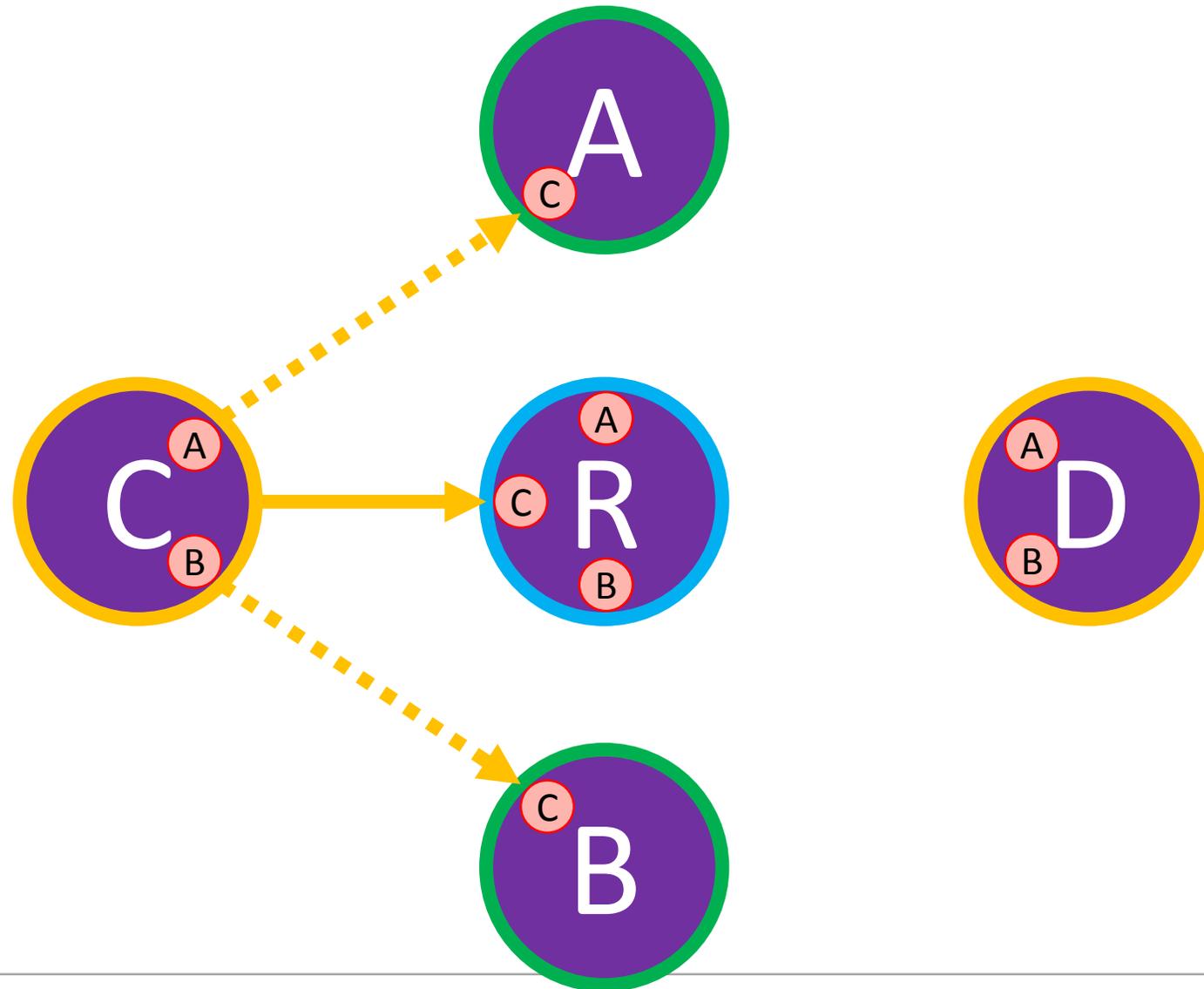
# Cross Topology with Overhearing 1/5



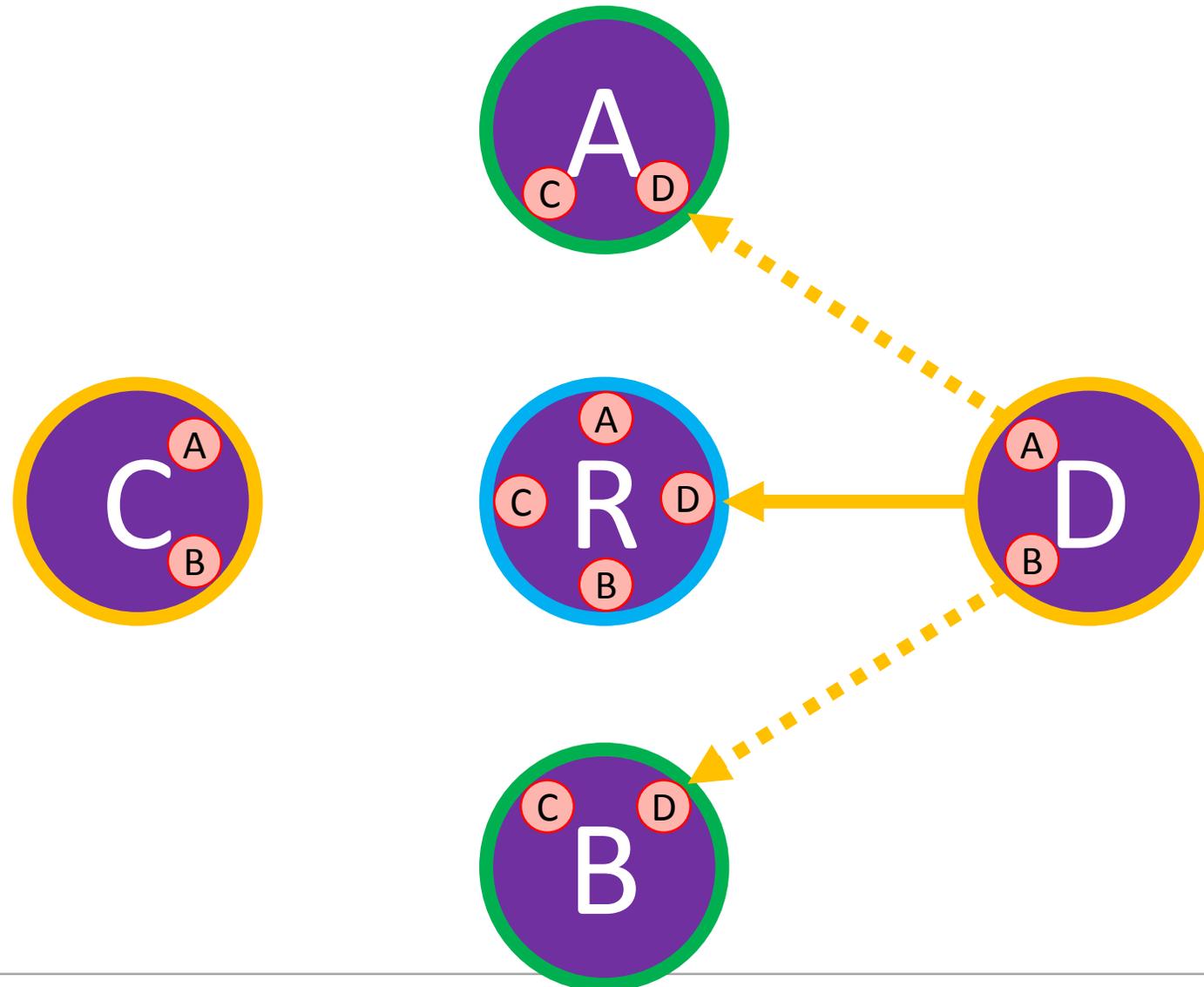
# Cross Topology with Overhearing 2/5



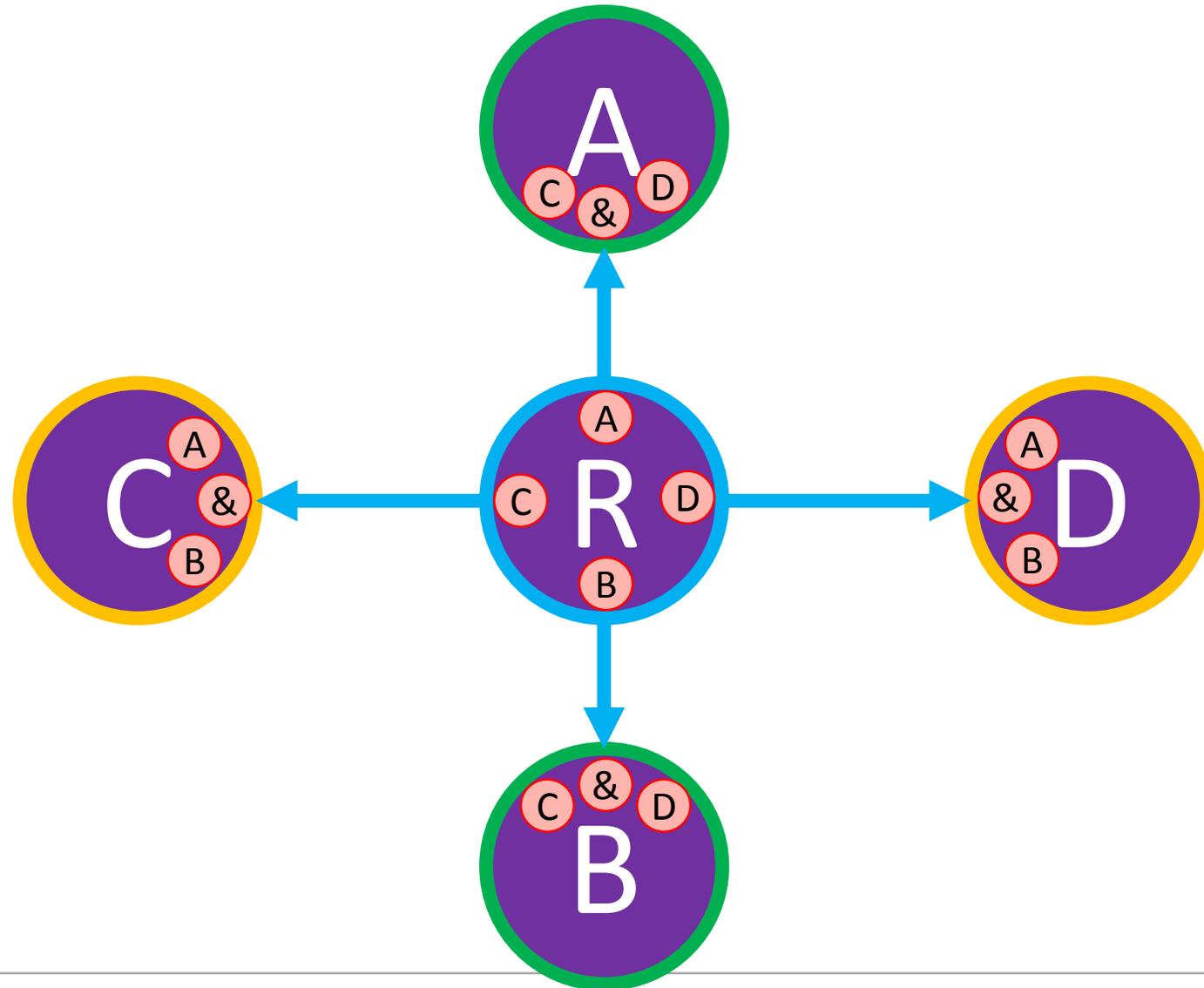
# Cross Topology with Overhearing 3/5



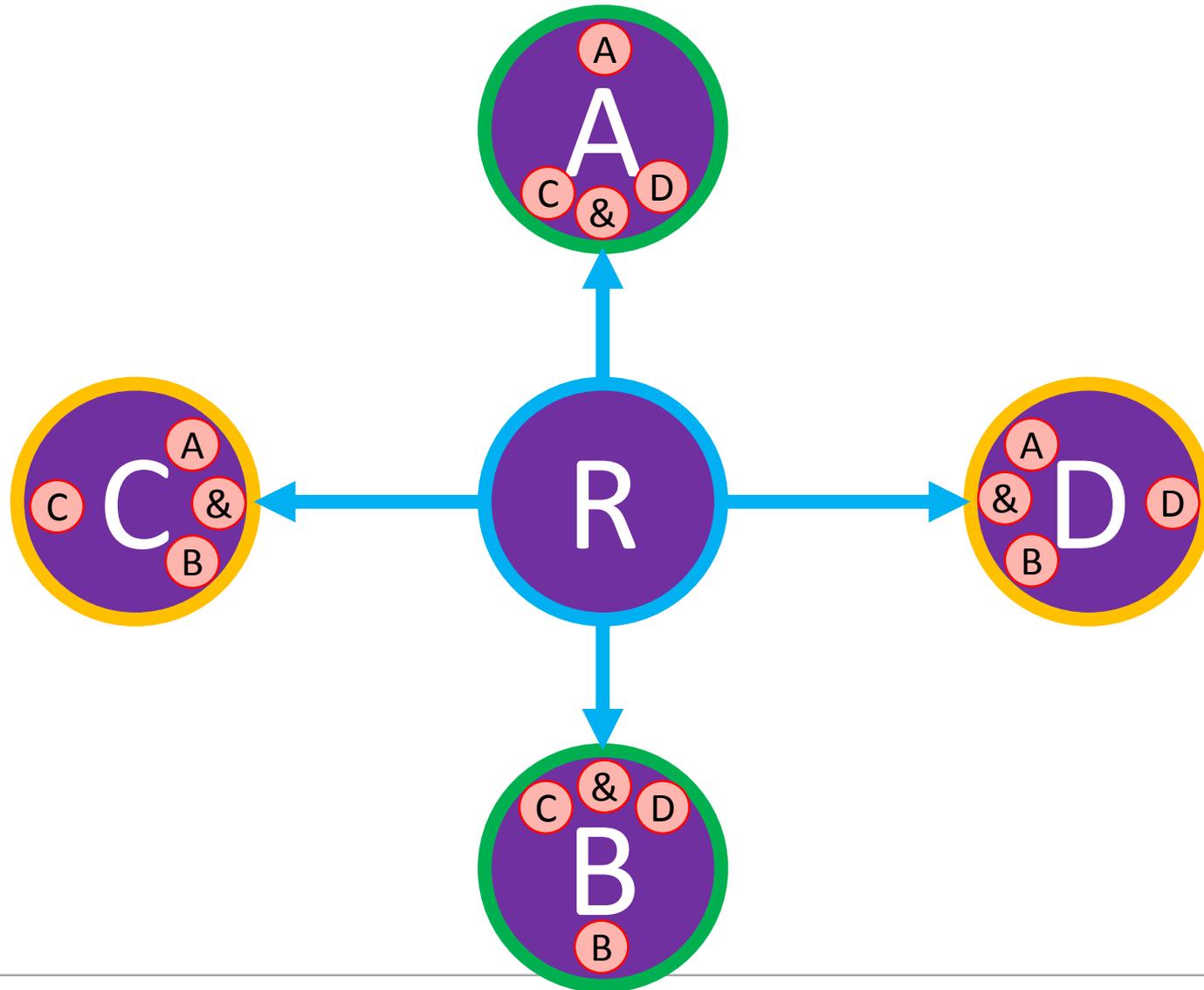
# Cross Topology with Overhearing 4/5



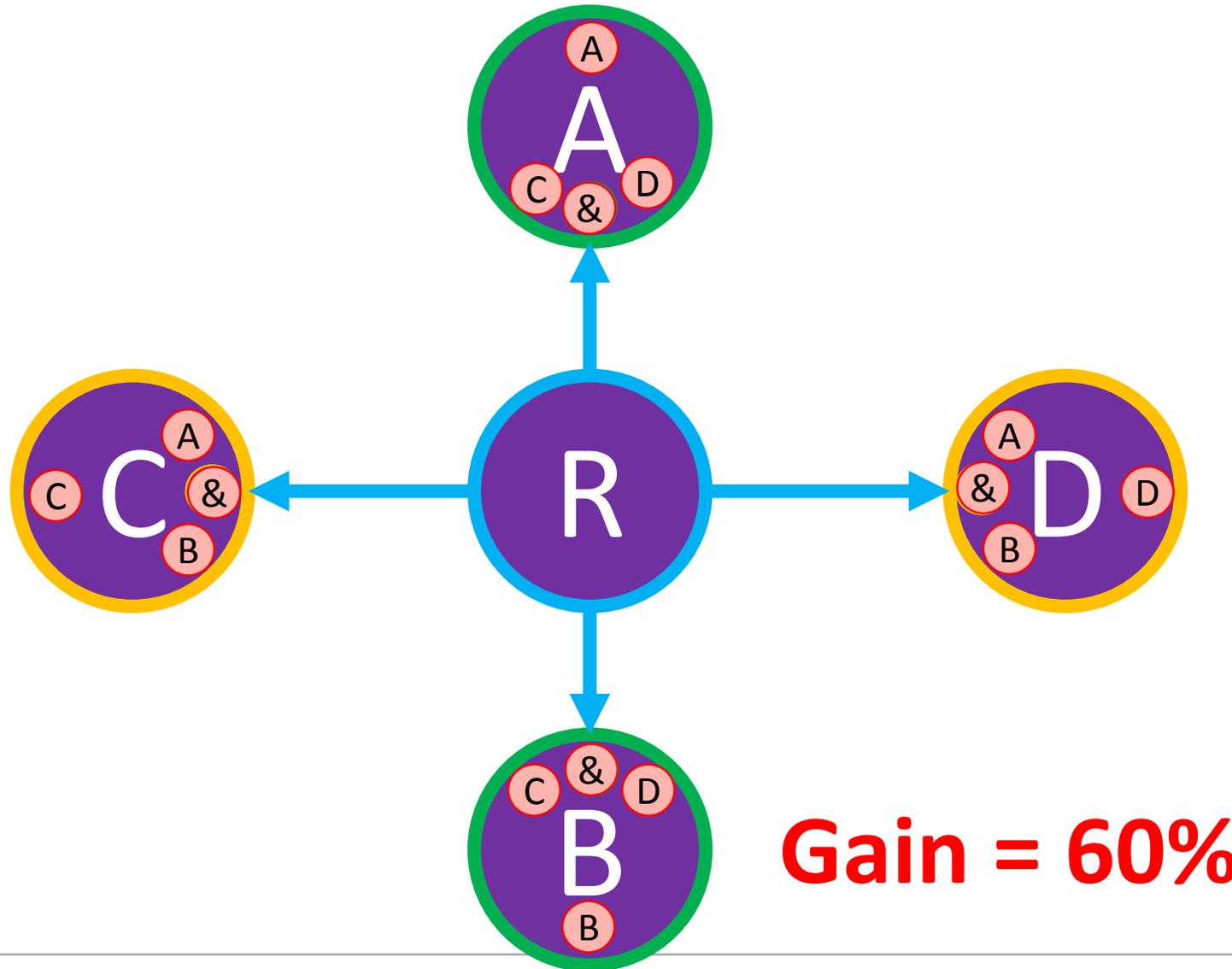
# Cross Topology with Overhearing 5/5



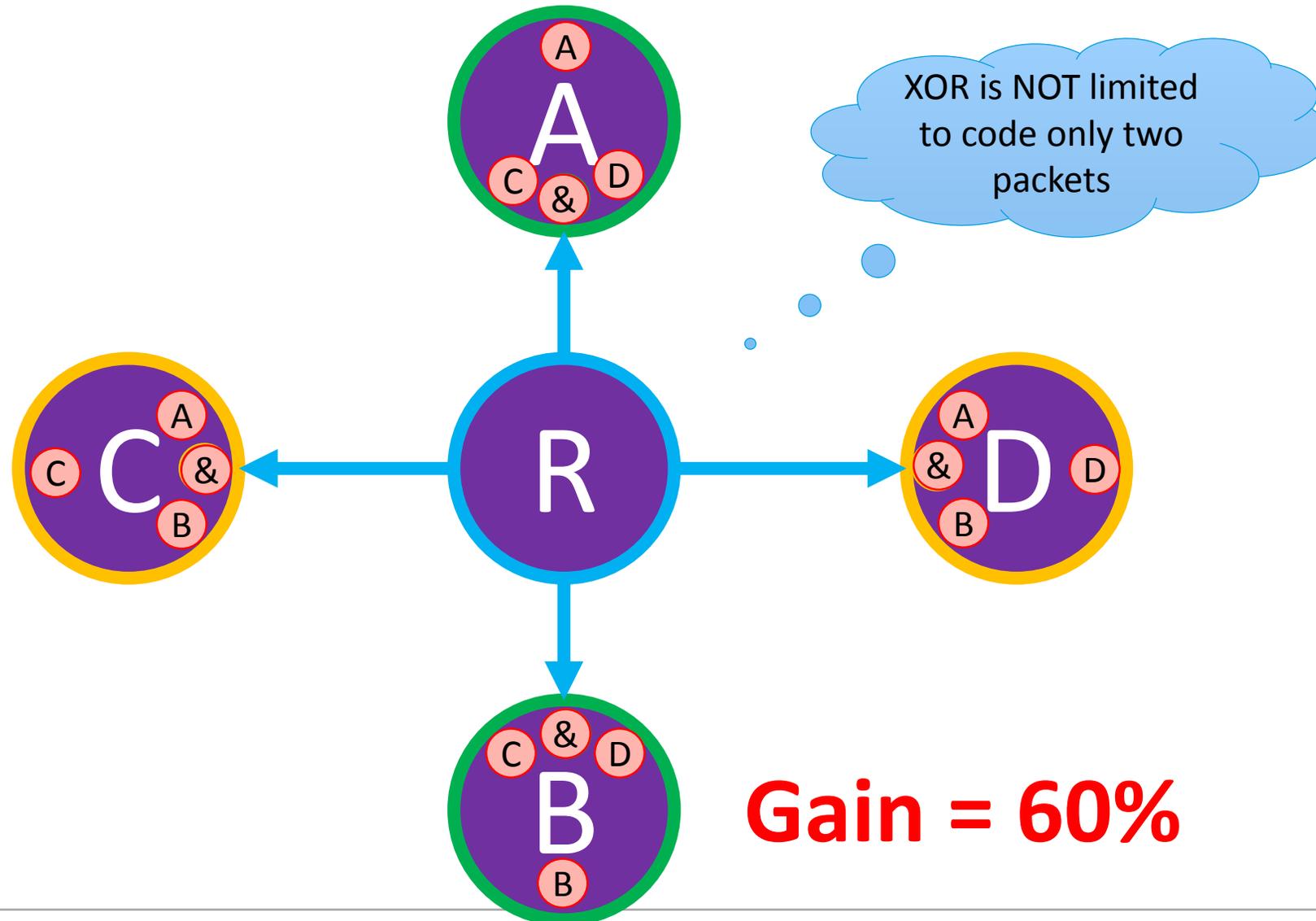
# Cross Topology with Overhearing 1/2



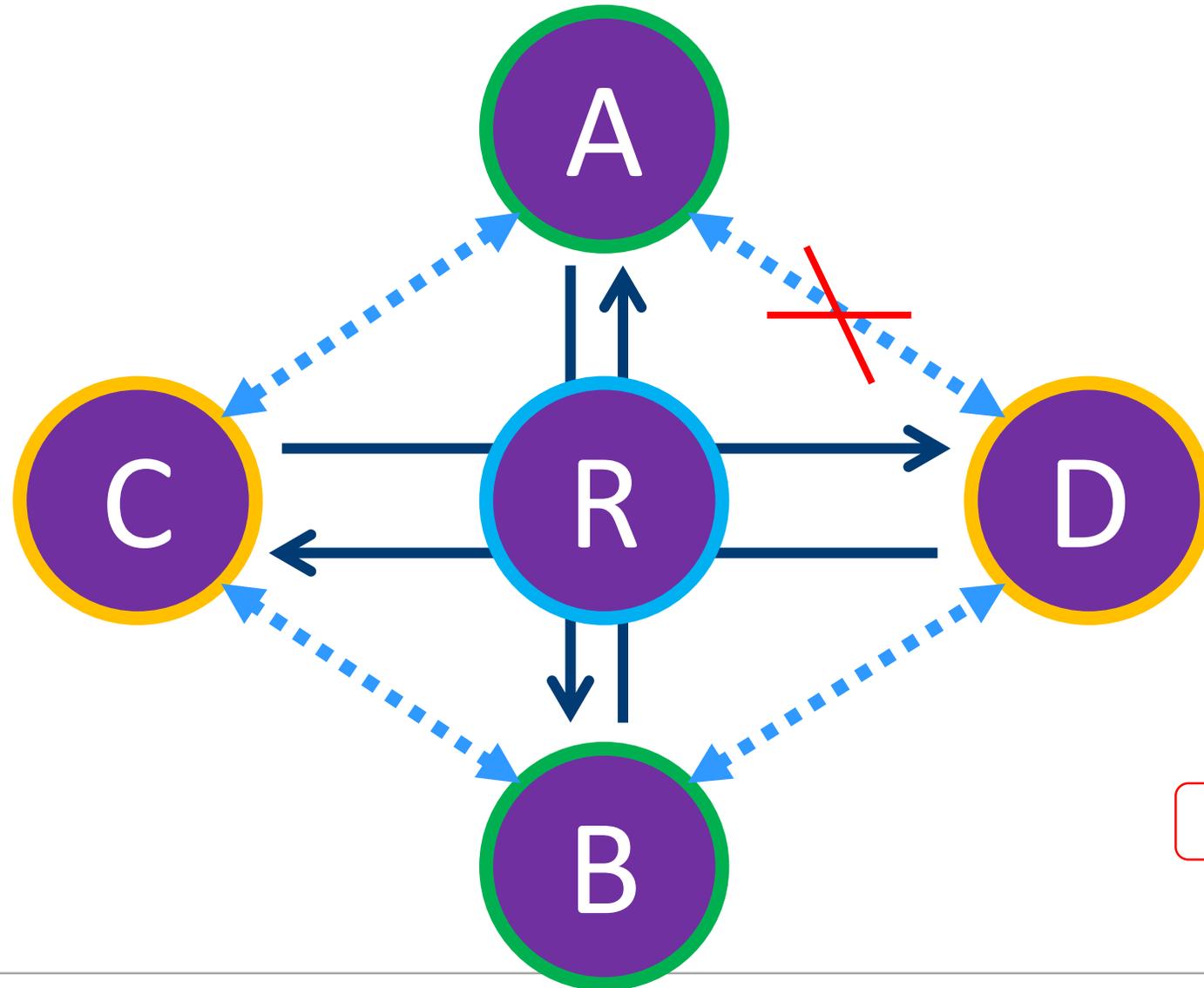
# Cross Topology with Overhearing 2/2



# Cross Topology with Overhearing 2/2



# Cross Topology with Overhearing Problems



Hint: CORE protocol

# Inter Flow Network Coding

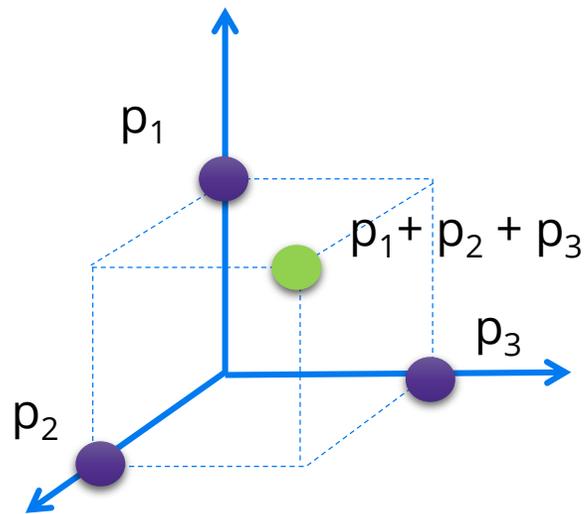
- No delay
- Low complexity
- Easy integration with commercial solutions
- Some ideas are used in ANALOG network coding
- Planning and book keeping needed
- Certain traffic characteristics are beneficial in mesh networks
- Symmetric traffic allows for more coding potential

# Index Coding

# Index Coding

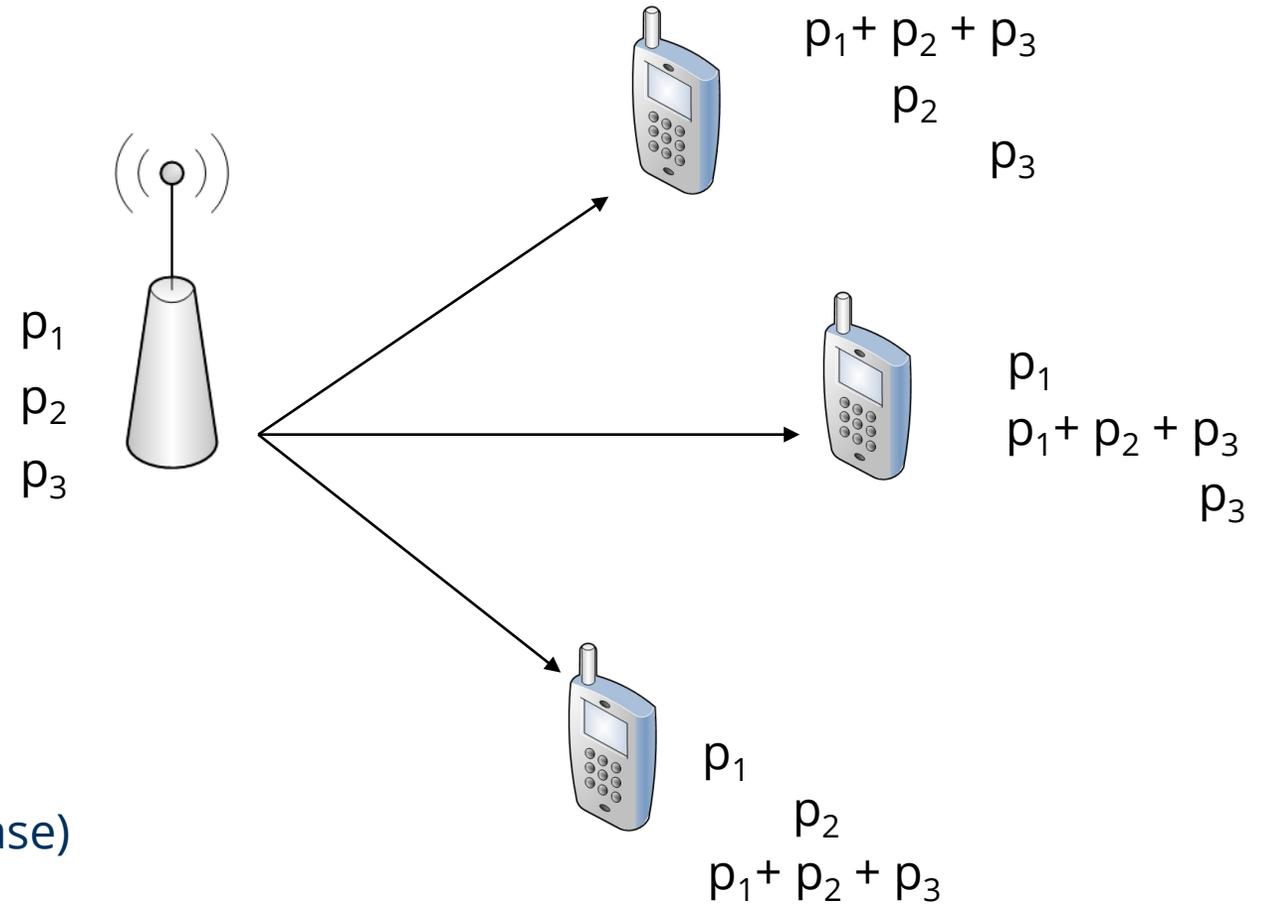


# Index Coding



In wireless networks

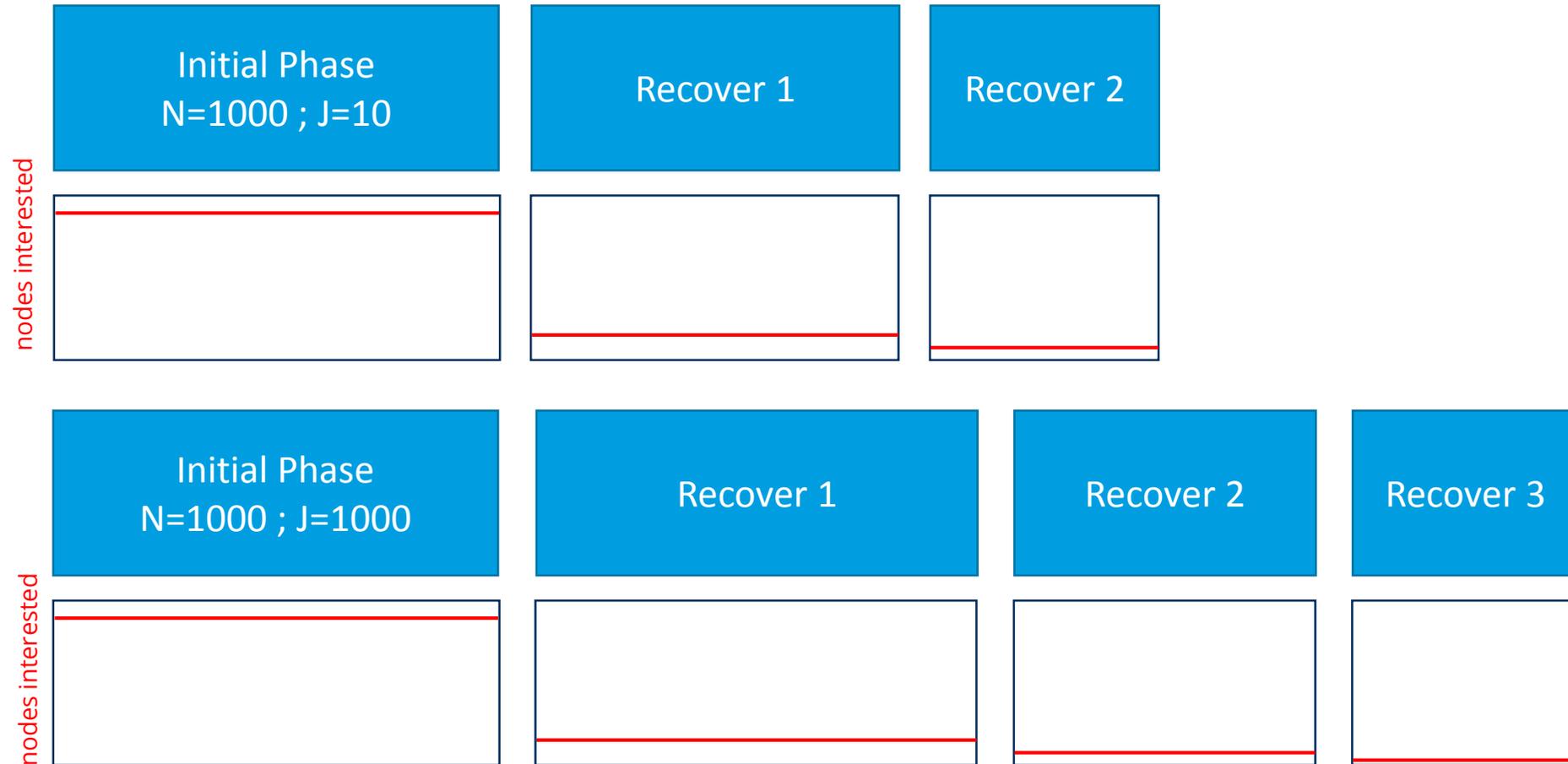
- Broadcast advantage
- Generation of packets (three in this case)
- Packet erasures (losses)



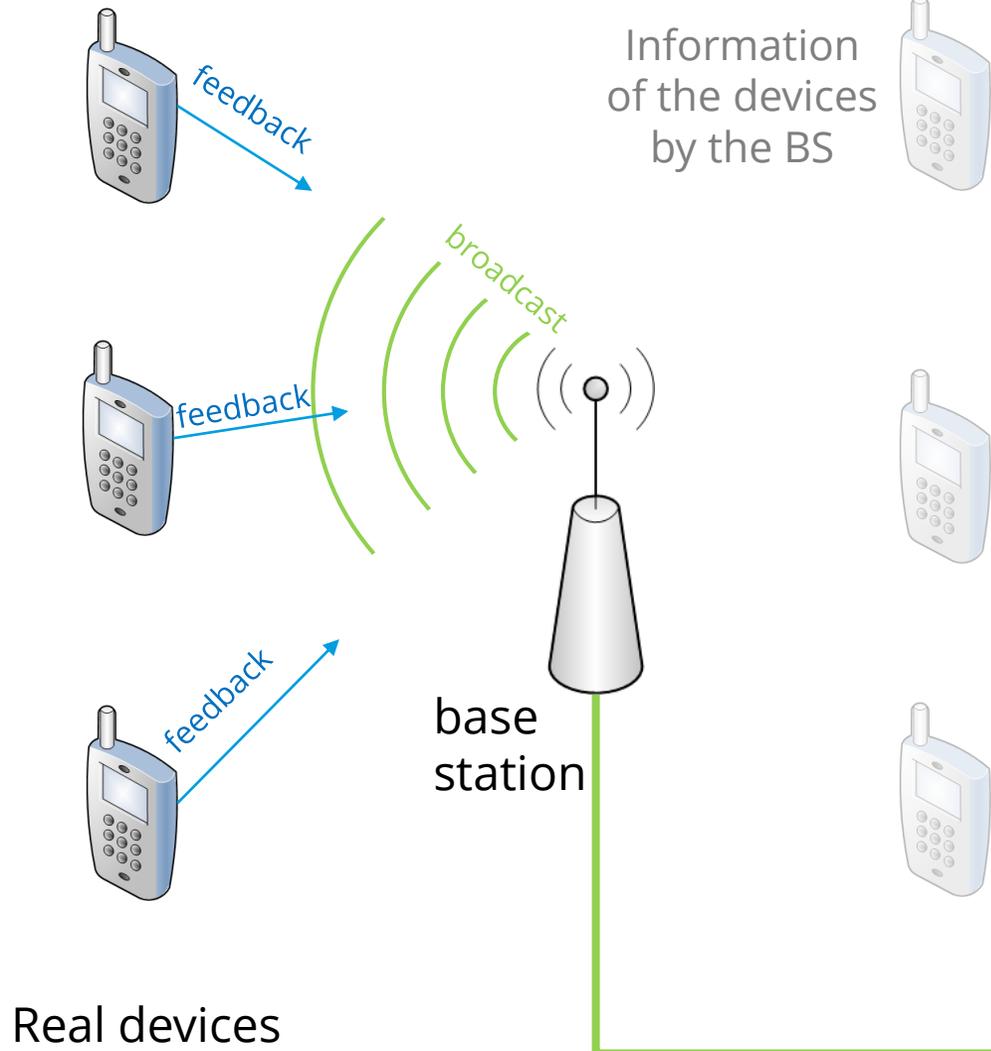
# Reliable Multicast: Motivation

- Wireless has the advantage of inherent broadcast
- Wireless is error prone
- Coding versus Retransmissions
- $N$  = number of packets
- $J$  = number of users

# Reliable Multicast: Motivation



# Index Coding



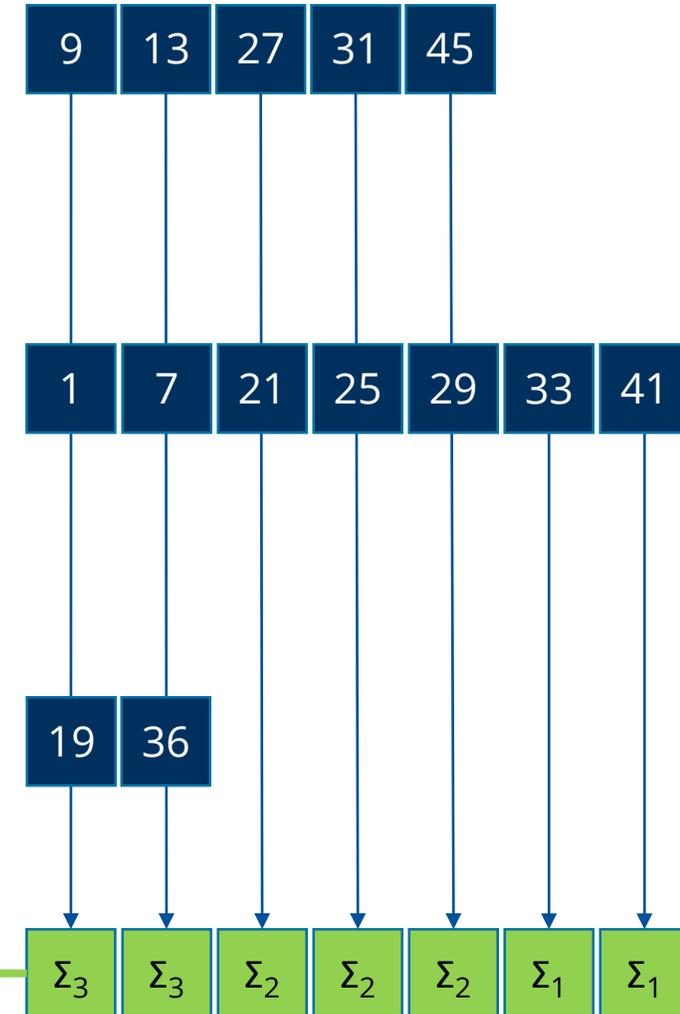
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

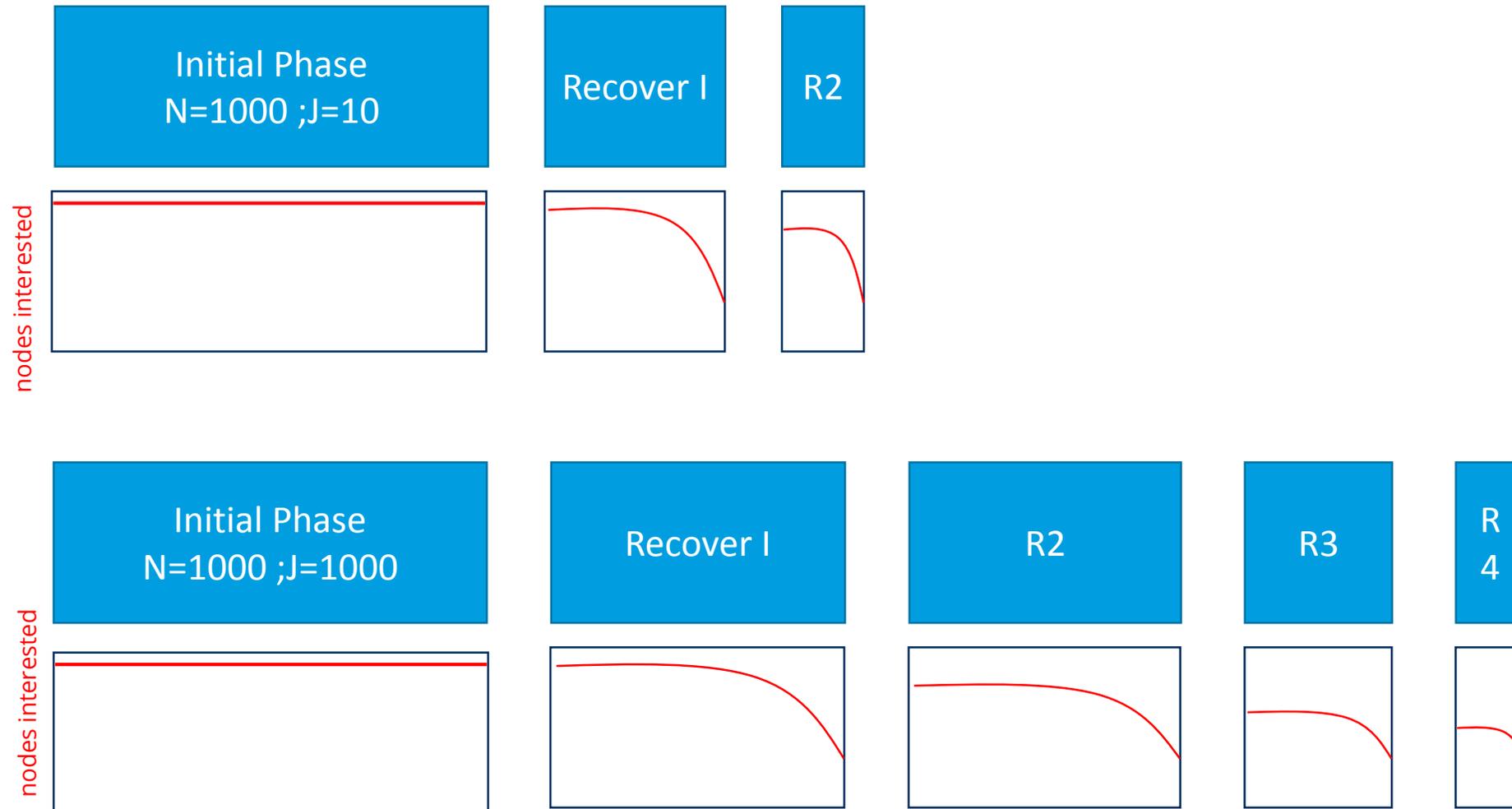
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

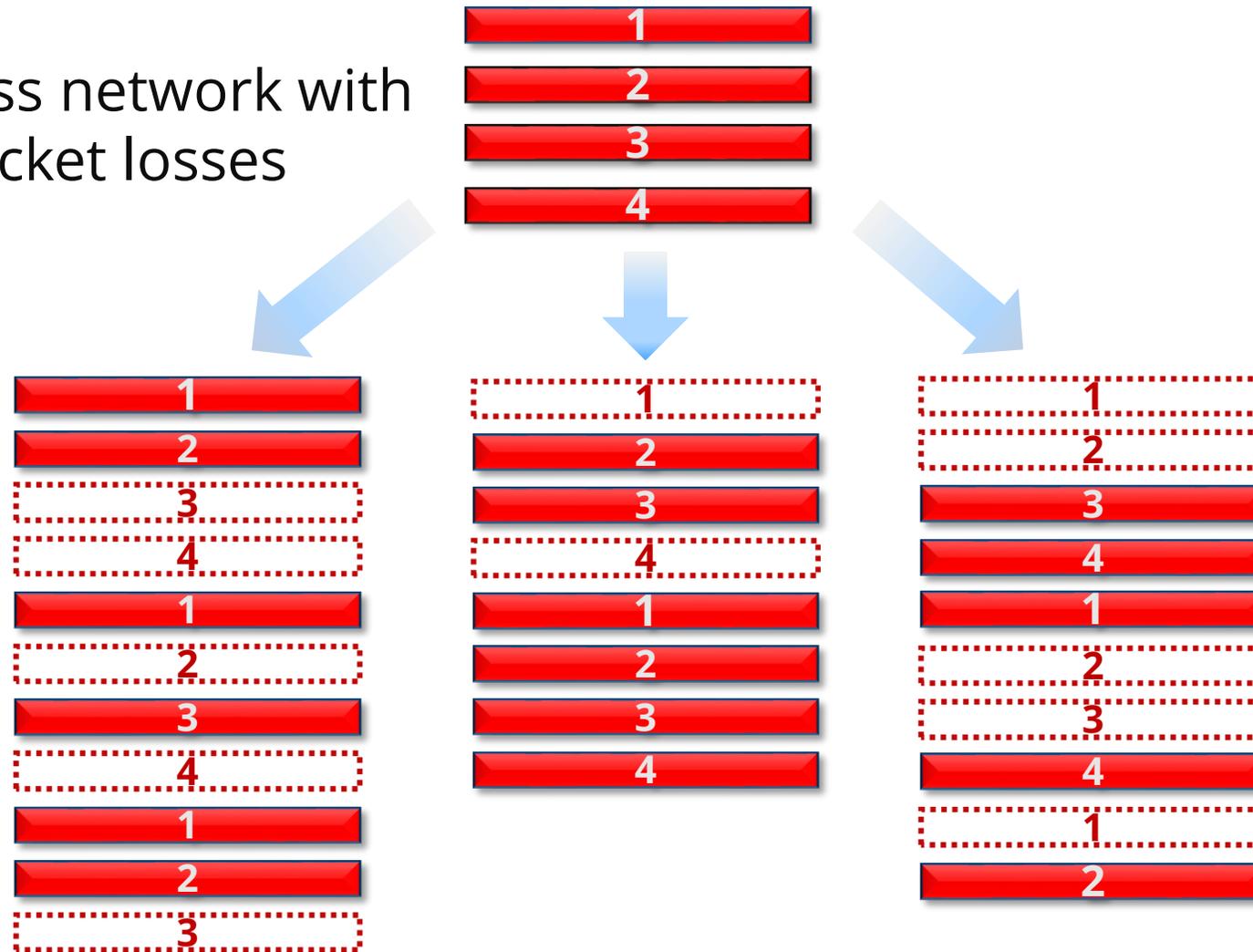


# Index Coding

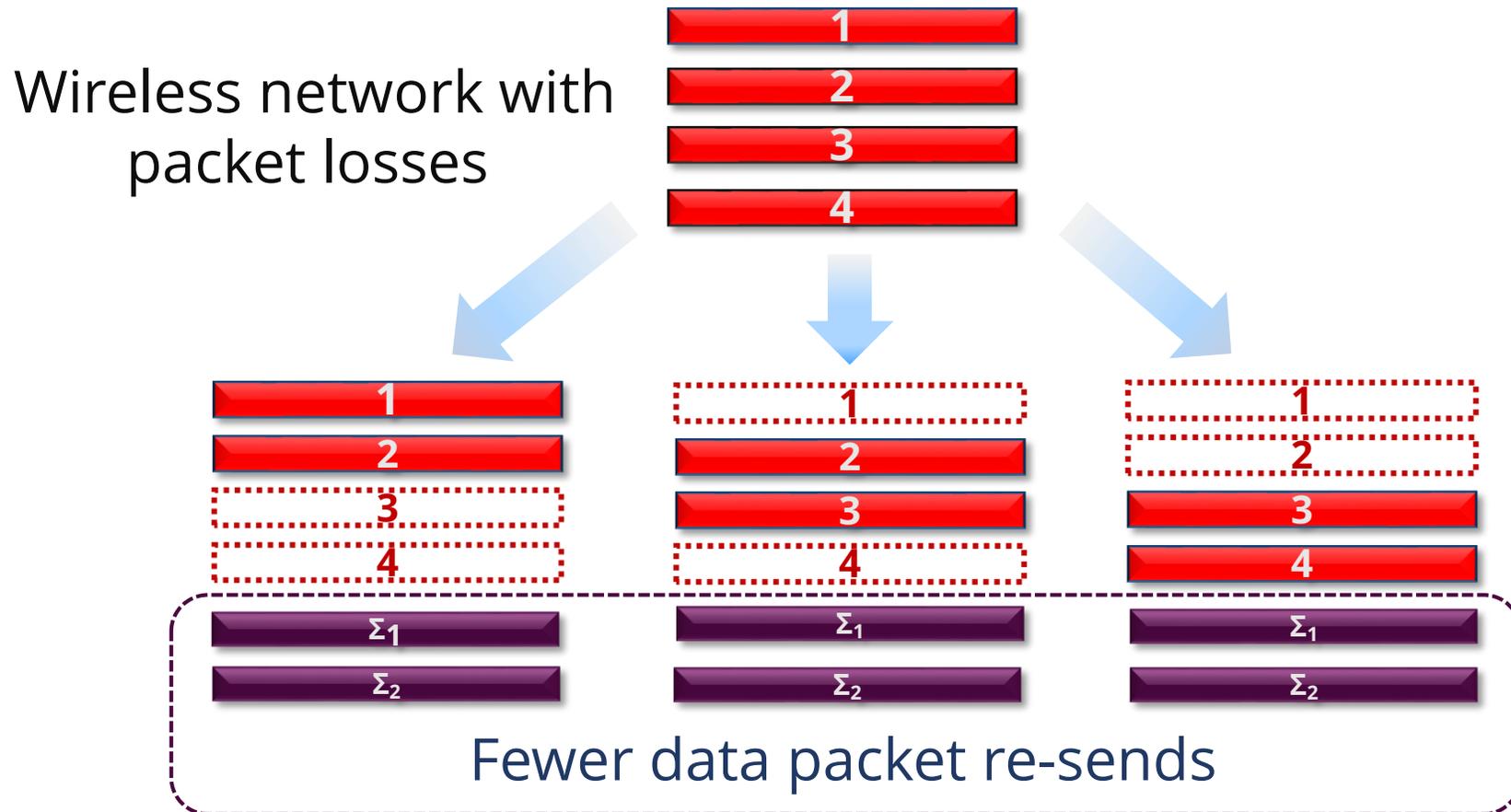


# Broadcast Example

Wireless network with packet losses

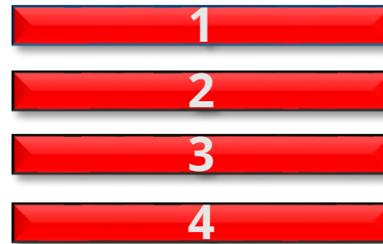


# Broadcast Example

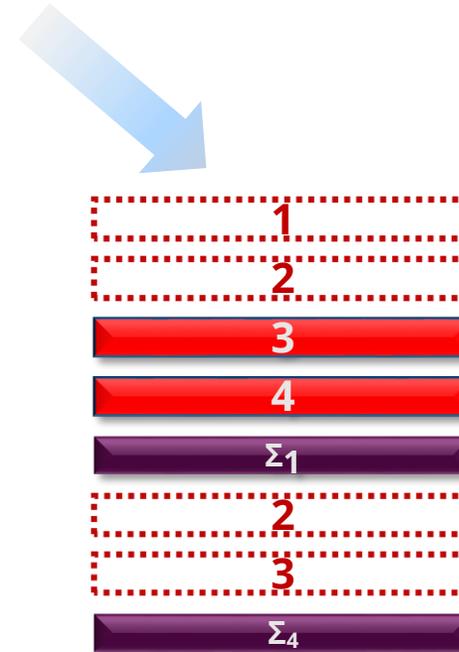
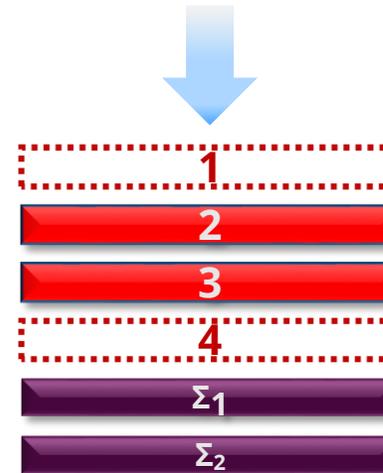
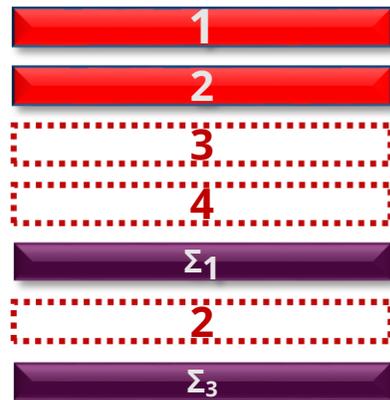


# Broadcast Example

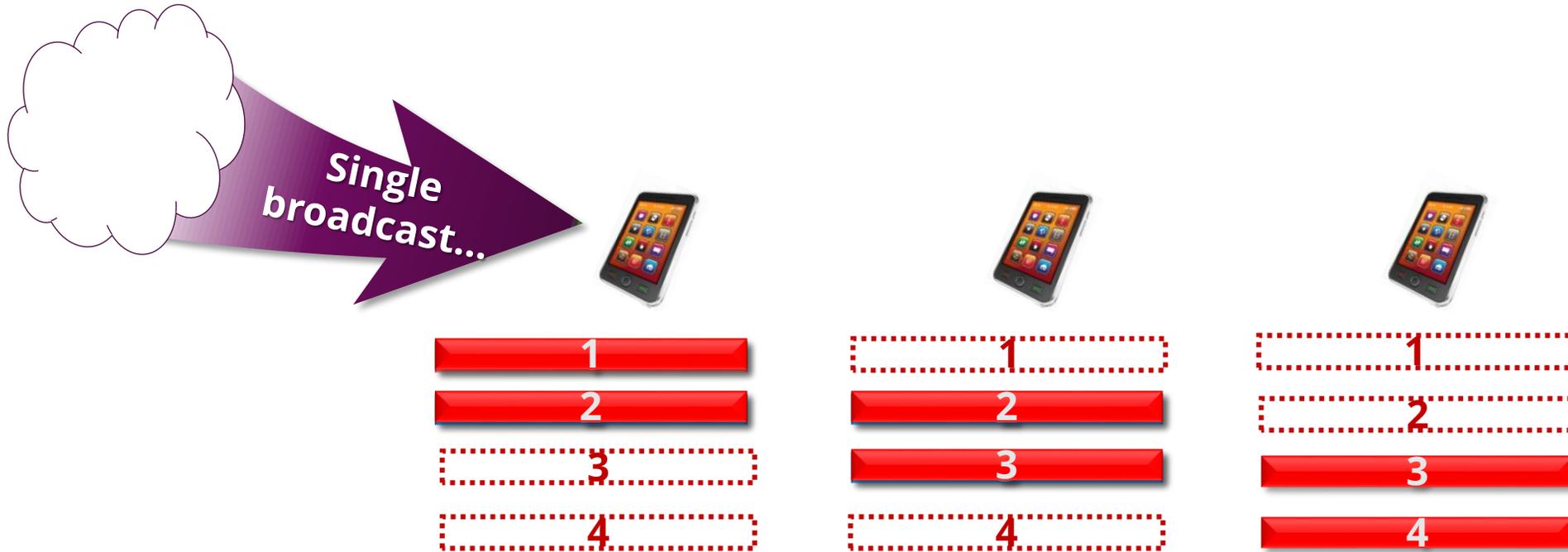
Wireless network with packet losses



Same errors as before ...



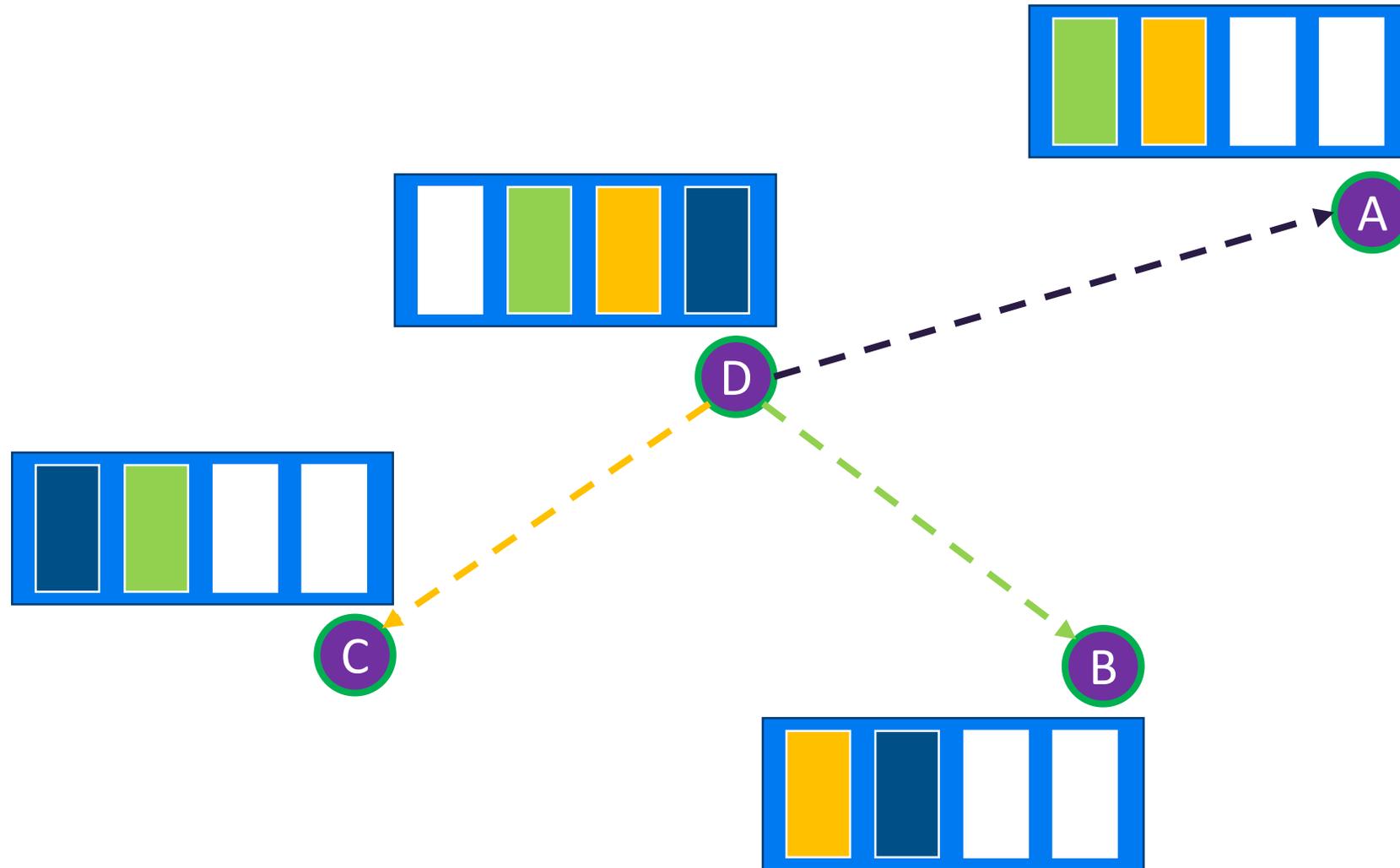
# Broadcast Example: Cooperative Advantage



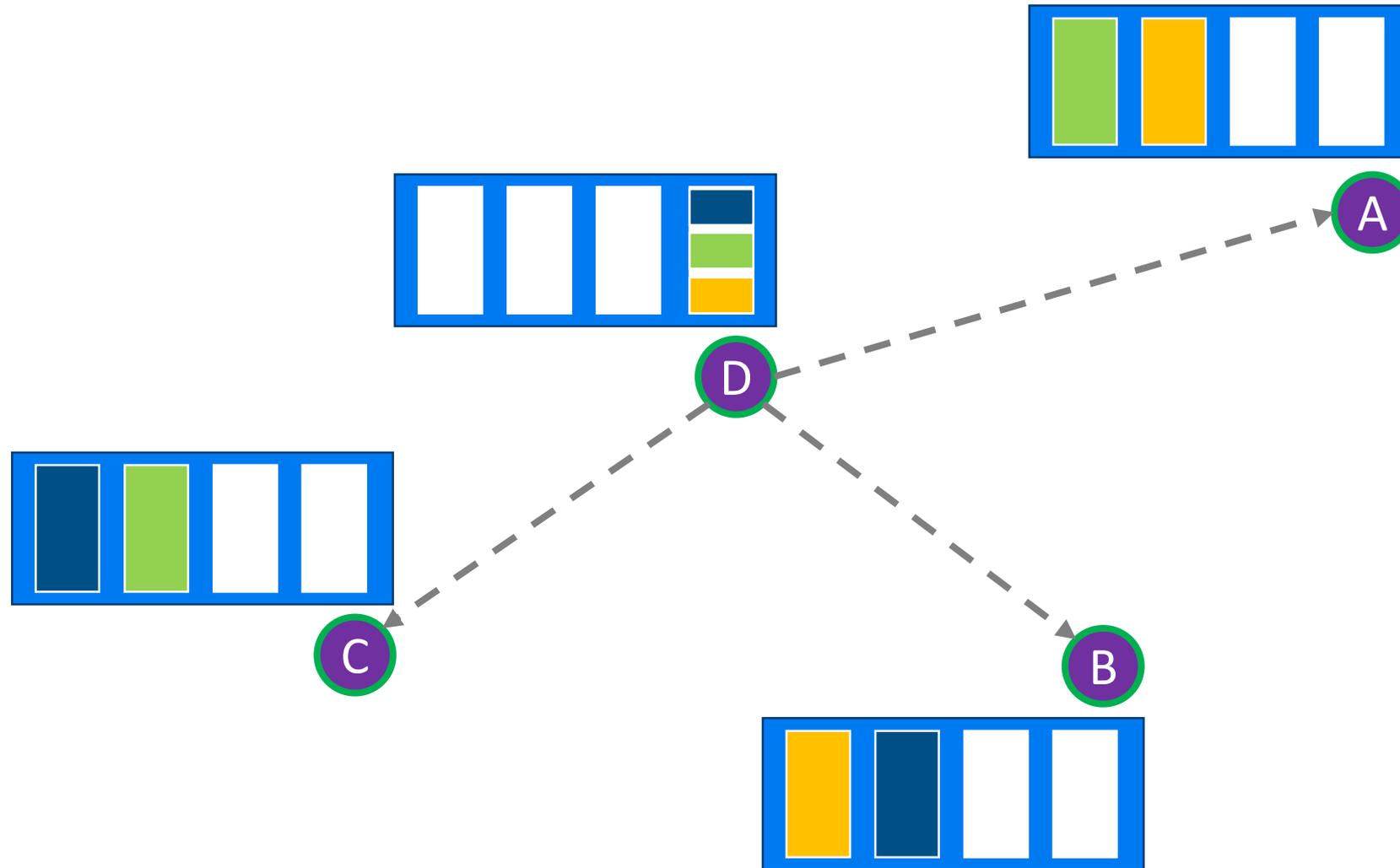
...and peers share missing data



# w/o Coding



# w Coding



# First implementations

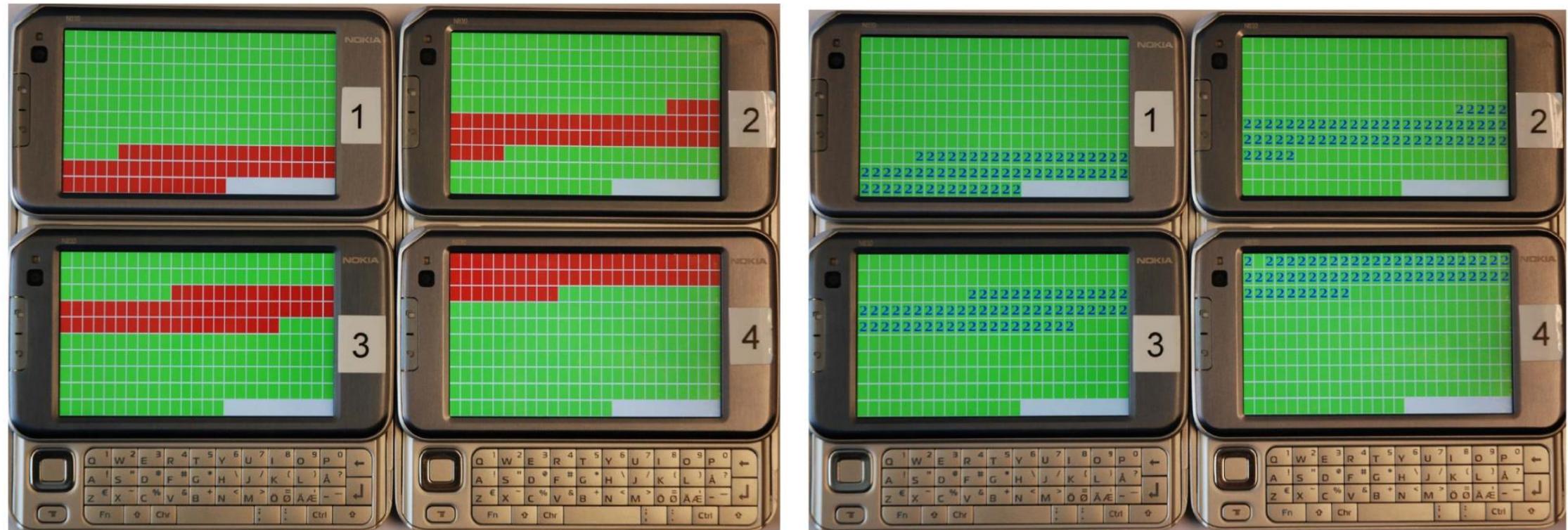
K.F. Nielsen, T.K. Madsen, and F.H.P. Fitzek, “**Network coding opportunities for wireless grids formed by mobile devices,**” in The Second International Conference on Networks for Grid Applications, Springer in the ICST Lecture Notes (LNICST) series, Ed. ICST, Oct. 2008.

# Inter-Flow NC on N810 (2008)

- XOR coding on N810 (linux device by NOKIA)
- Remote setup
  - Predefined set
  - Random set
- Constantly exchanging reception updates (who needs what)
- Overall goal is to let all devices have all information



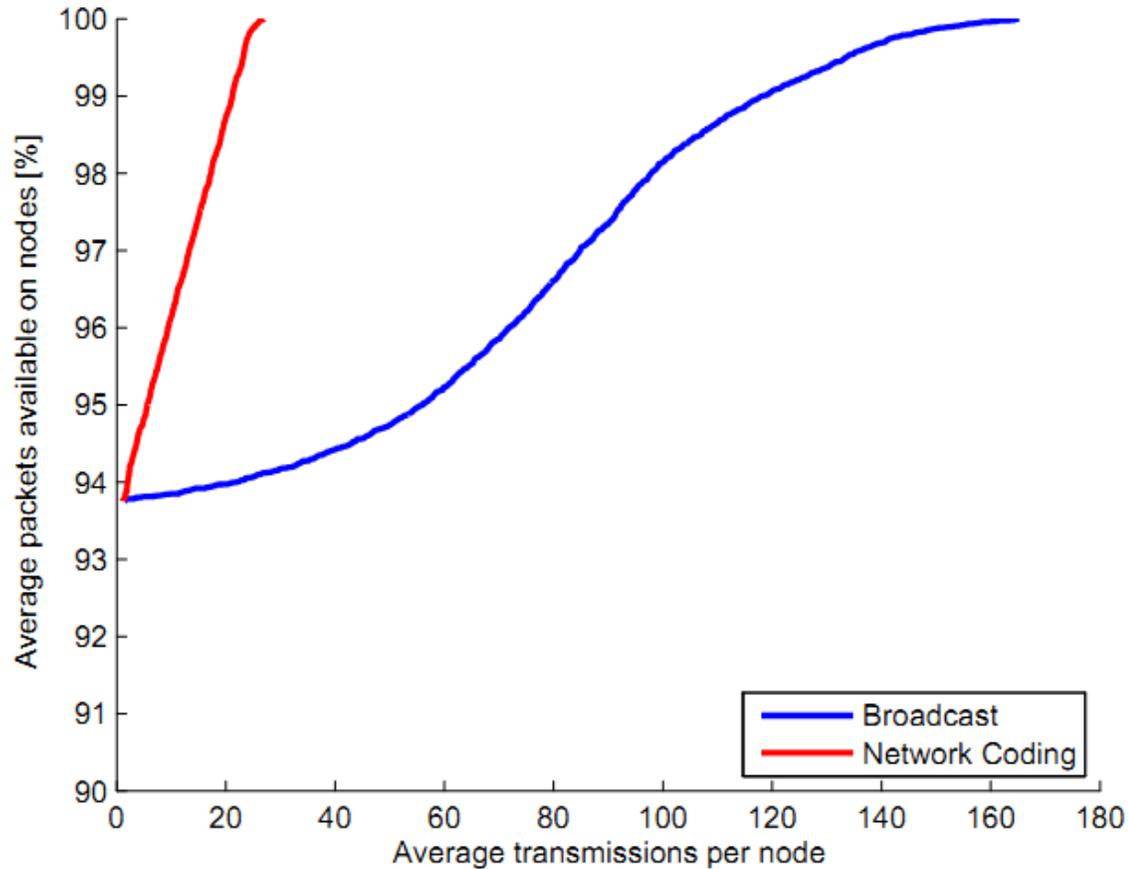
# N810 Implementation COPE



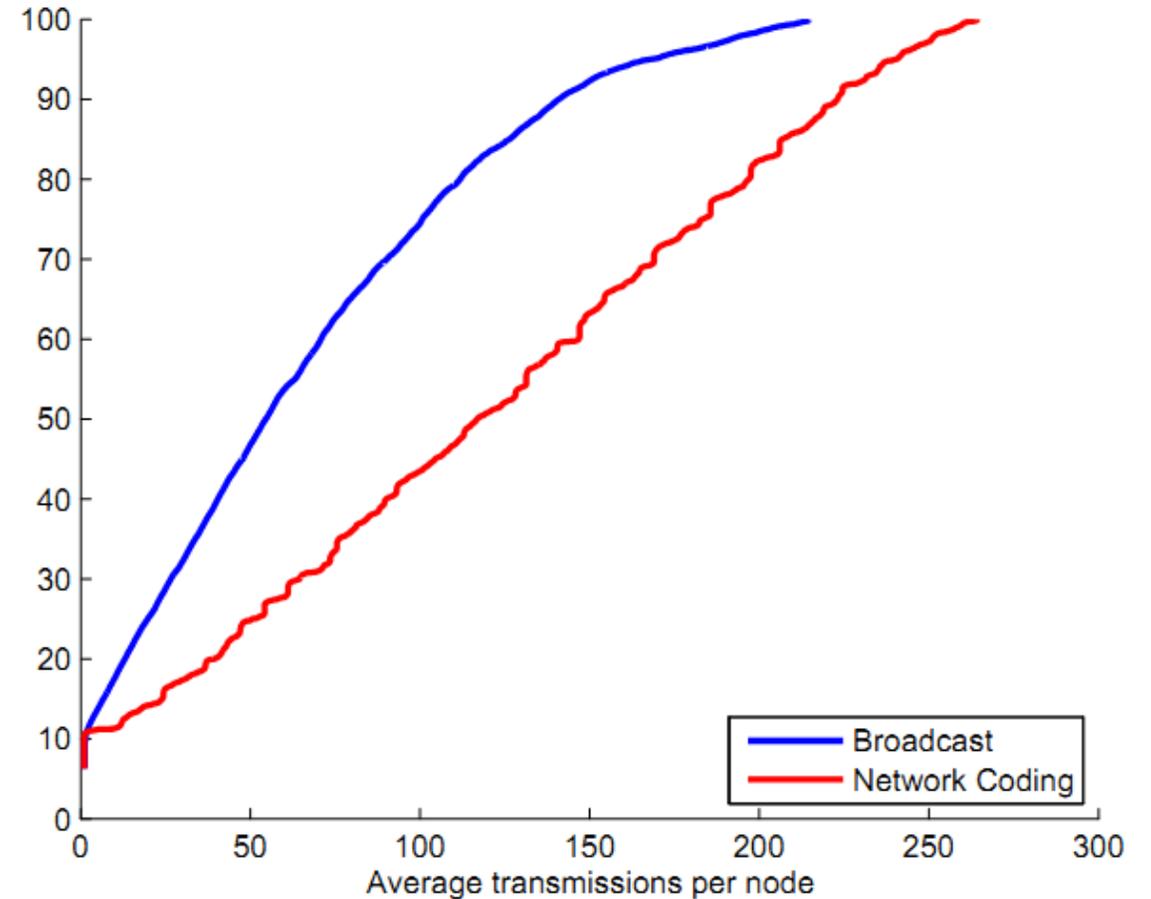
<https://www.youtube.com/watch?v=VZYLsyZaEO8>

# Results: N810

Random Starting phase:  
(duplicates are possible)



Disjoint Starting Phase:  
(no duplicates on beginning)



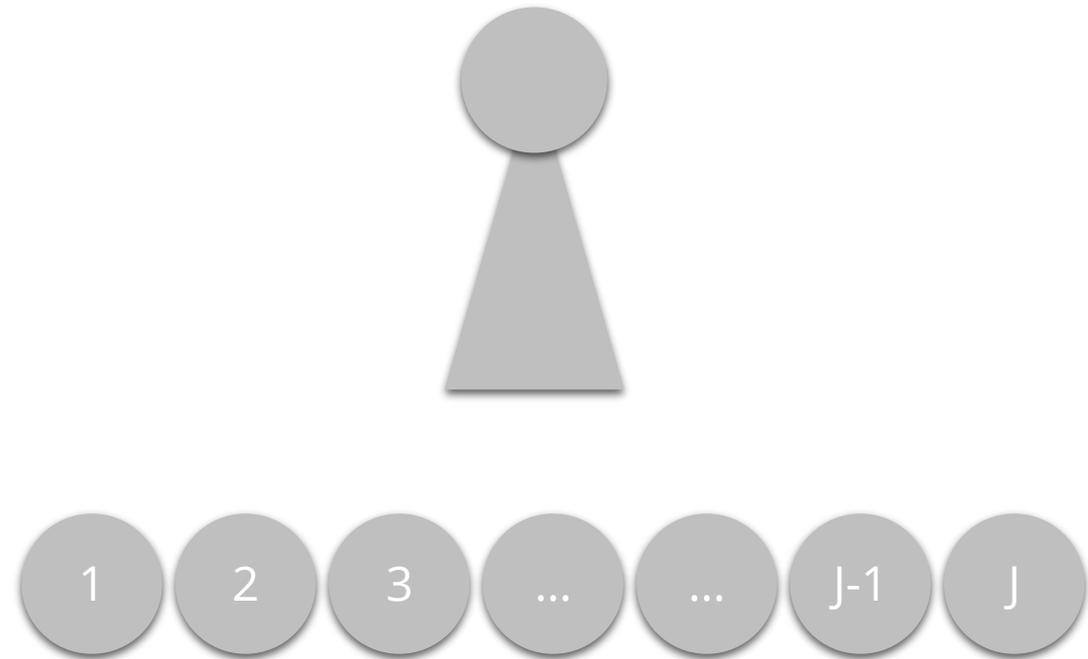
# Digital Zig Zag Coding

*Sagt es ihnen! Zick, zick, zick, nein zack ... die falsche Richtung ...  
Meister ... meine Bilder ... meine Leinwand ...  
Zickzack falsch ... sagt es ihnen ... falsch...*



# Architecture and Example

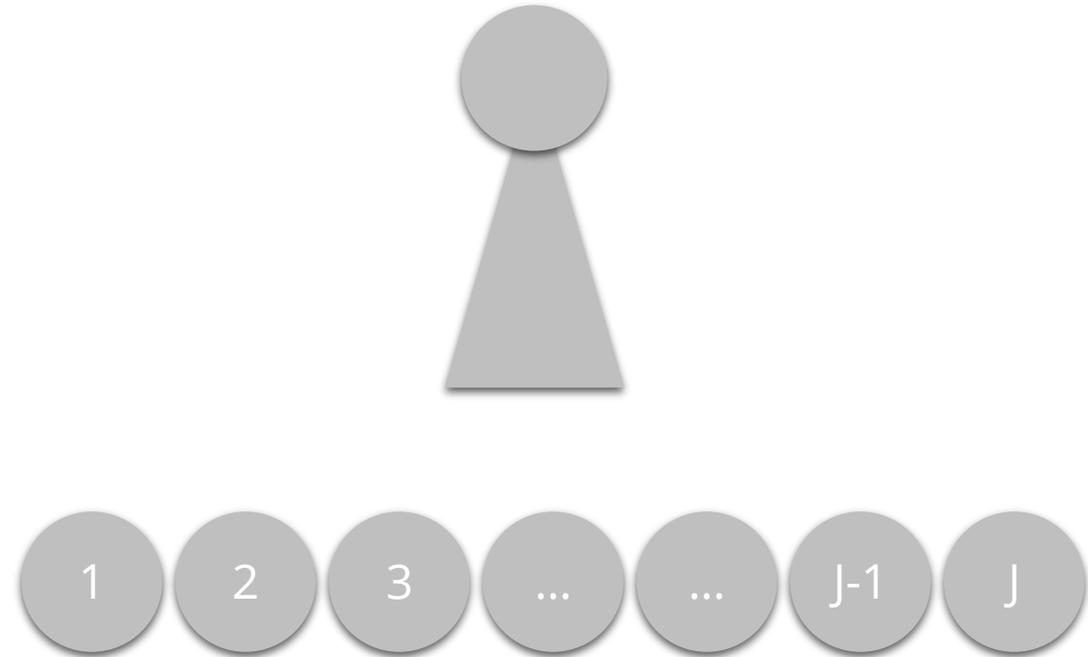
- Assuming one node broadcasts information to several nodes over error-prone medium
- All nodes are interested in the same content



# Architecture and Example

- Assuming one node broadcasts information to several nodes over error-prone medium
- All nodes are interested in the same content
- Assuming we have only two messages A and B with three bits of information each (to make it easy)

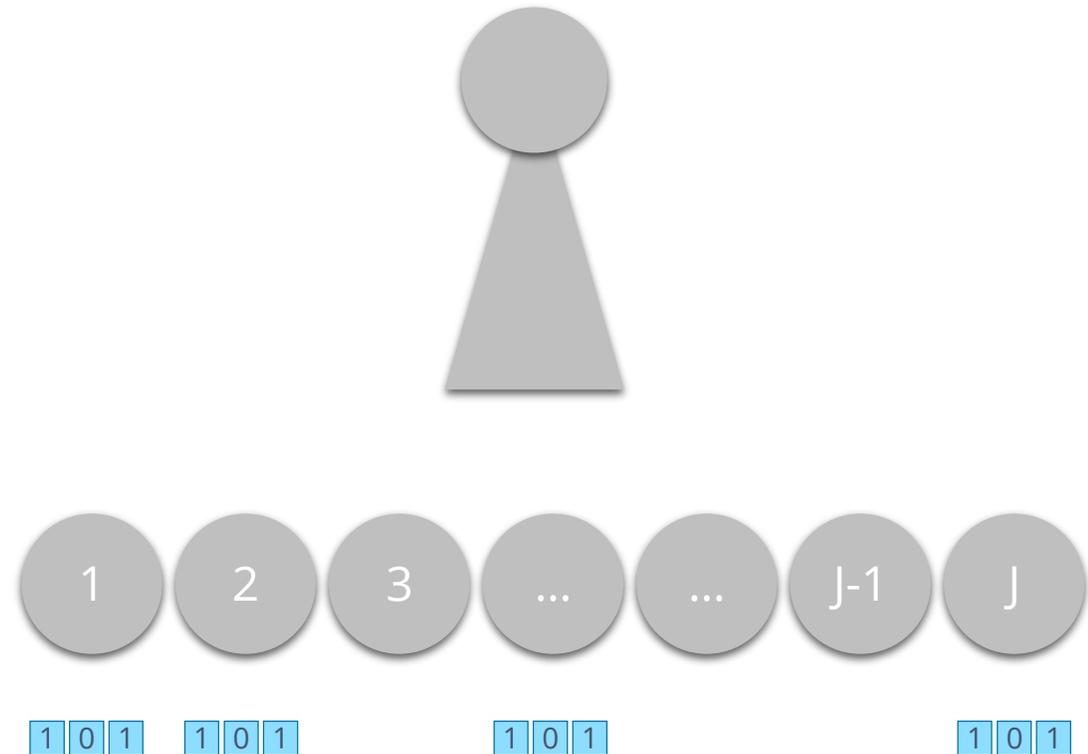
1	0	1	A
0	0	1	B



# Architecture and Example

- Assuming one node broadcasts information to several nodes over error-prone medium
- All nodes are interested in the same content
- Assuming we have only two messages A and B with three bits of information each (to make it easy)

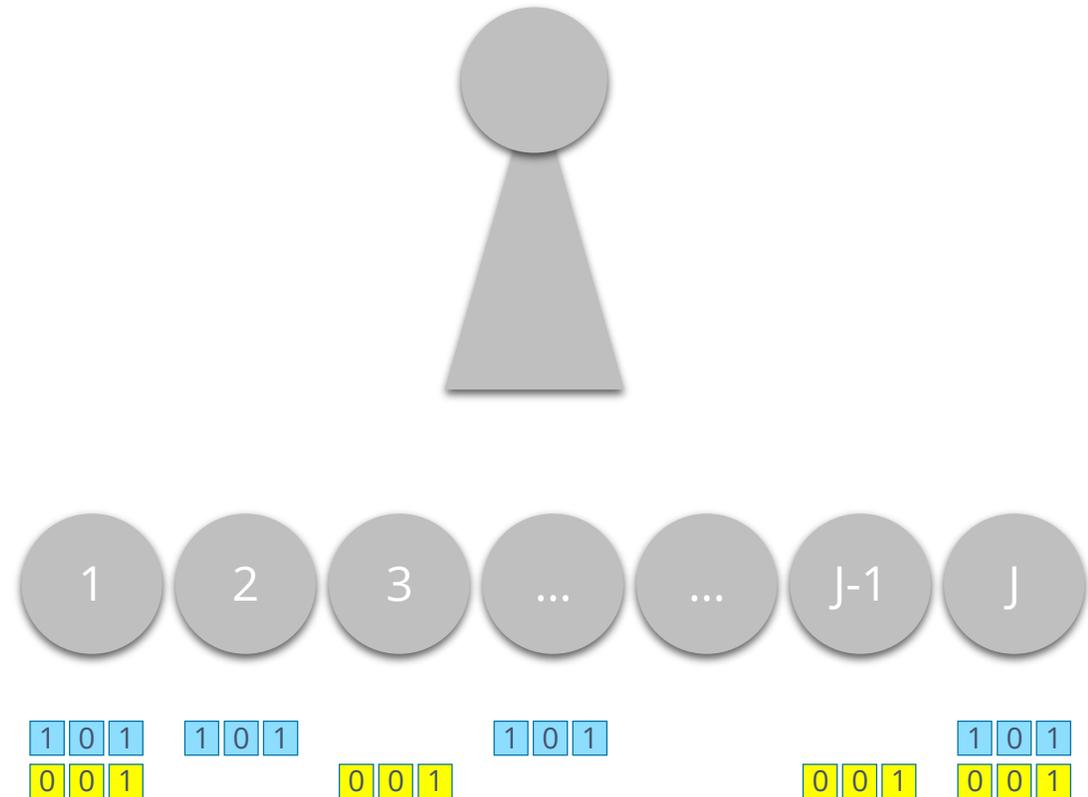
1	0	1	A
0	0	1	B



# Architecture and Example

- Assuming one node broadcasts information to several nodes over error-prone medium
- All nodes are interested in the same content
- Assuming we have only two messages A and B with three bits of information each (to make it easy)

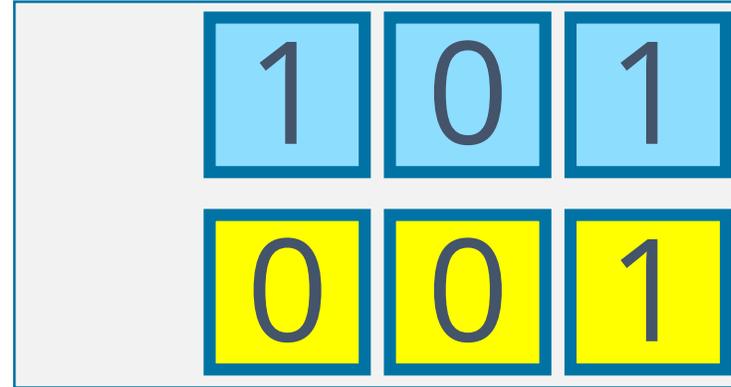
1 0 1 A  
0 0 1 B



# Decoding 1

Original packet A

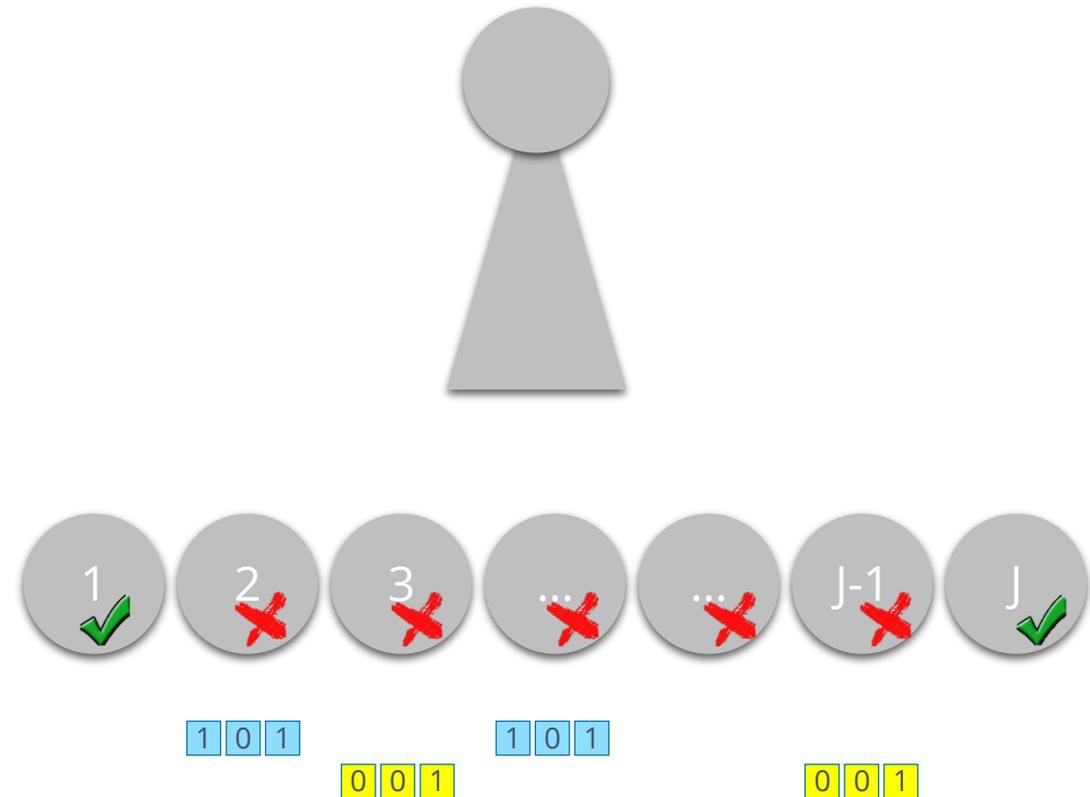
Original packet B



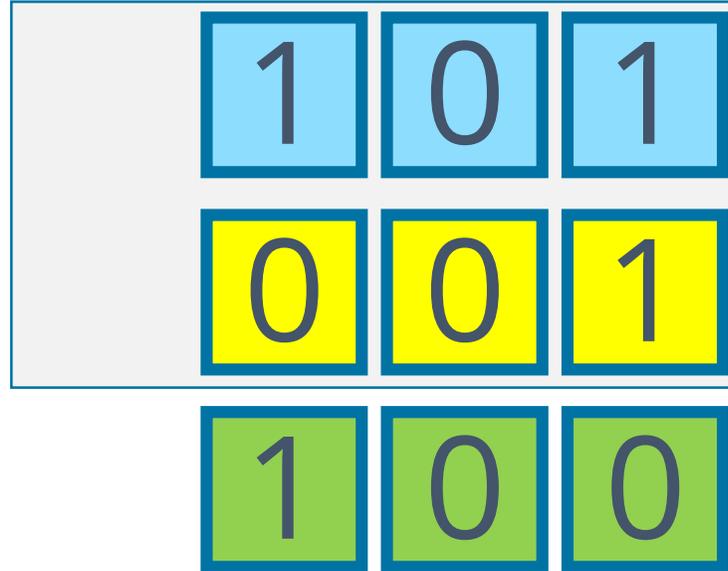
# Architecture and Example

- Assuming one node broadcasts information to several nodes over error-prone medium
- All nodes are interested in the same content
- Assuming we have only two messages A and B with three bits of information each (to make it easy)
- And now? Don't say repetition coding!

1	0	1	A
0	0	1	B



# Encoding - XOR



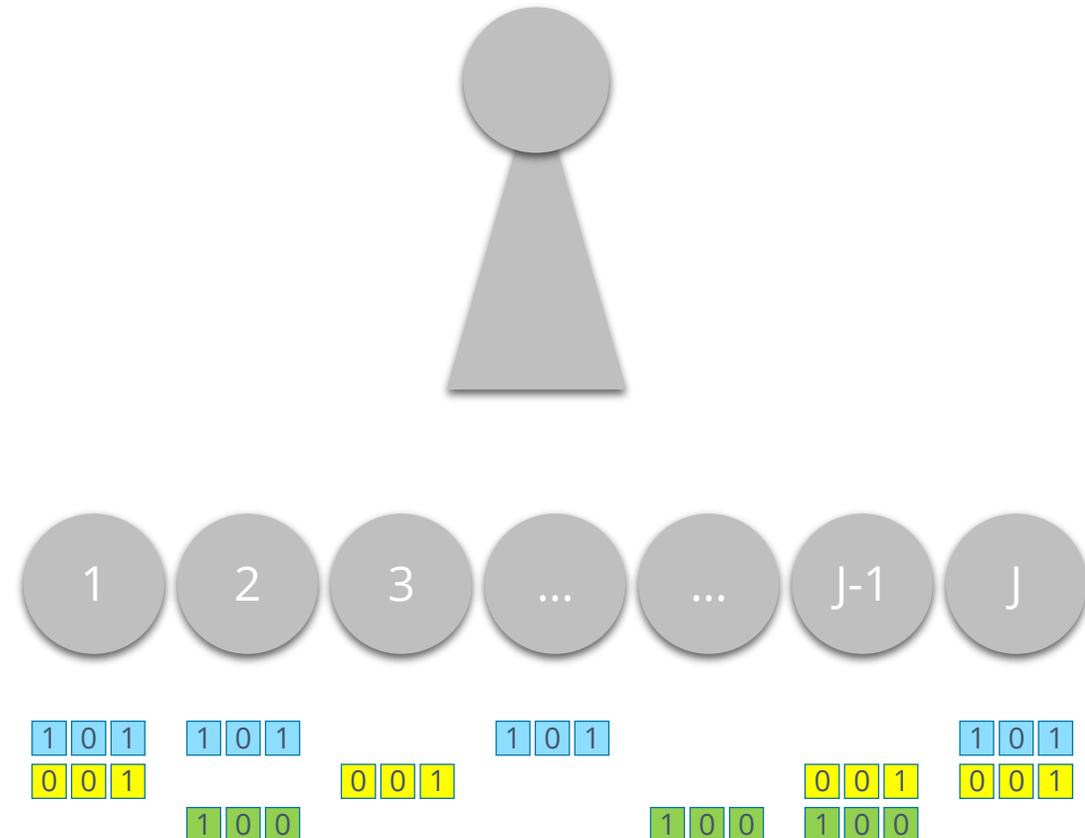
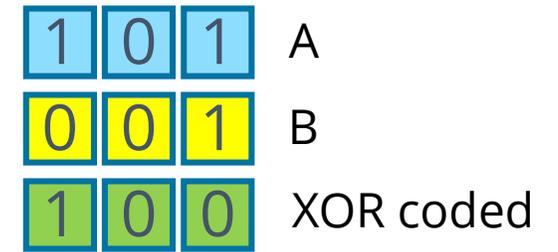
Original packet A

Original packet B

Coded packet  $A \oplus B$

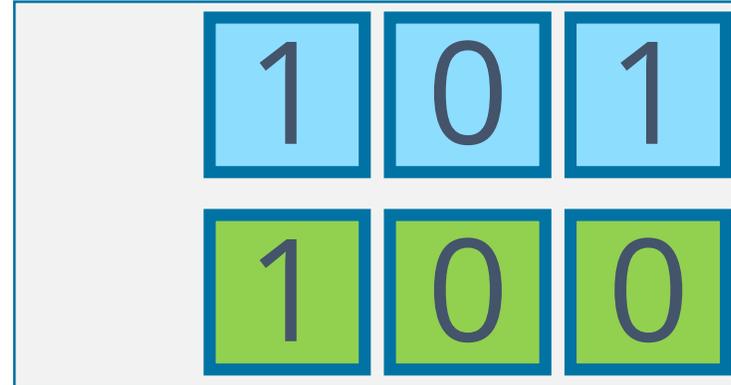
# Architecture and Example

- Assuming one node broadcasting information to several nodes over error-prone medium
- All nodes are interested in the same content
- Assuming we have only two messages A and B with three bits of information (to make it easy)
- Still not good enough now? Don't say repetition coding!

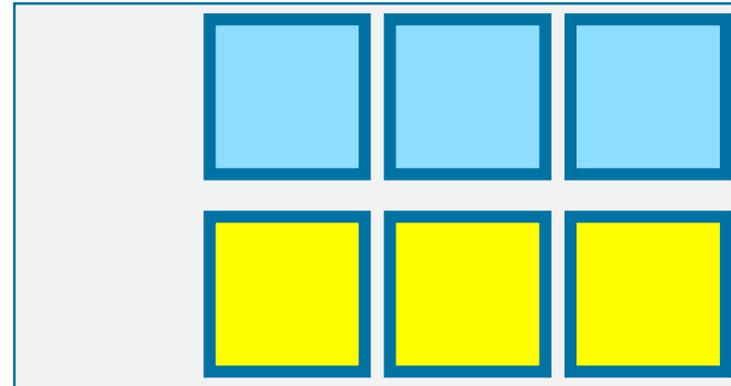


# Decoding 2

Original packet A

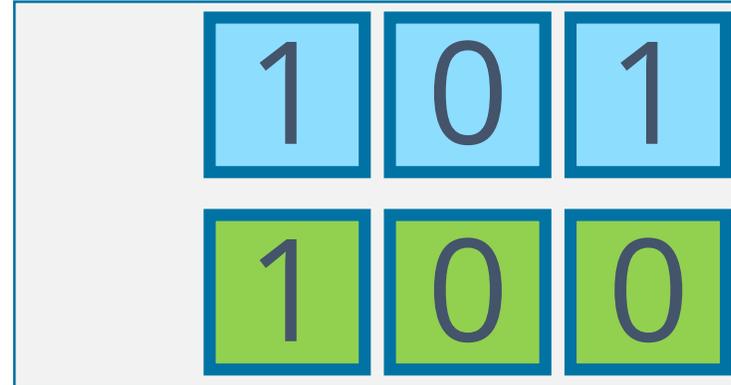


Coded packet  $A \oplus B$

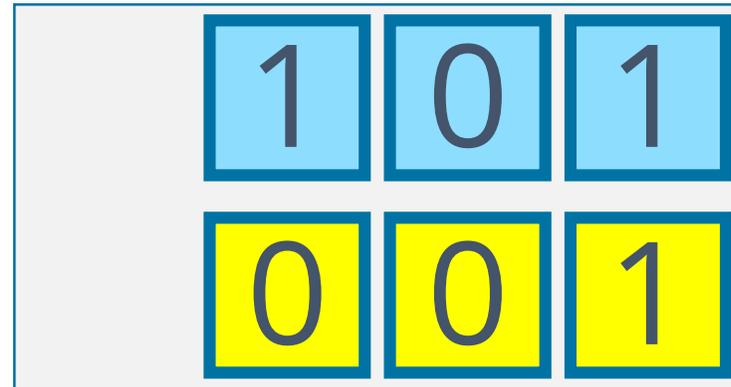


## Decoding 2

Original packet A



Coded packet  $A \oplus B$



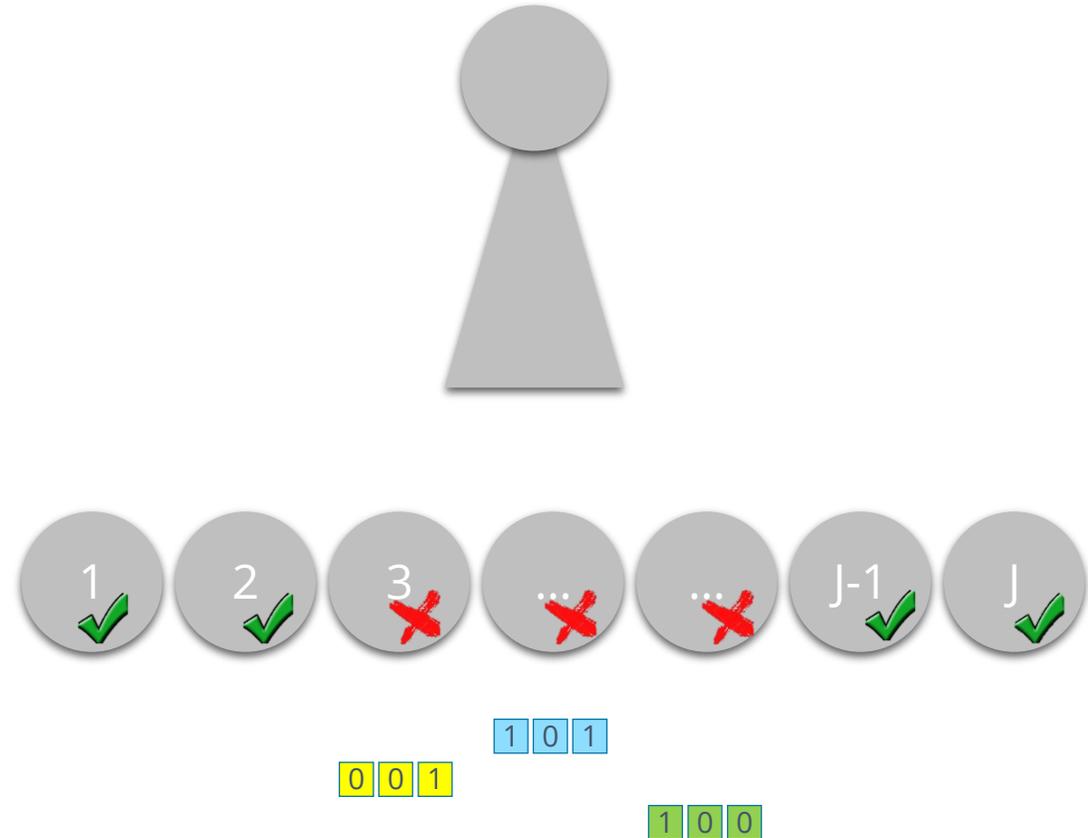
Simple reading & XOR



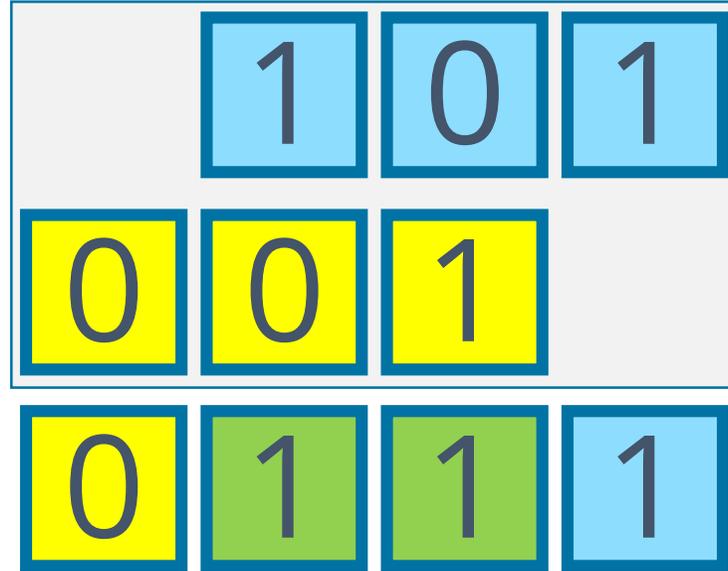
# Architecture and Example

- Assuming one node broadcasting information to several nodes over error-prone medium
- All nodes are interested in the same content
- Assuming we have only two messages A and B with three bits of information (to make it easy)
- Still not good enough now? Don't say repetition coding!

1	0	1	A
0	0	1	B
1	0	0	XOR coded



# Encoding - Shifted XOR



Original packet A

Original packet B\*2

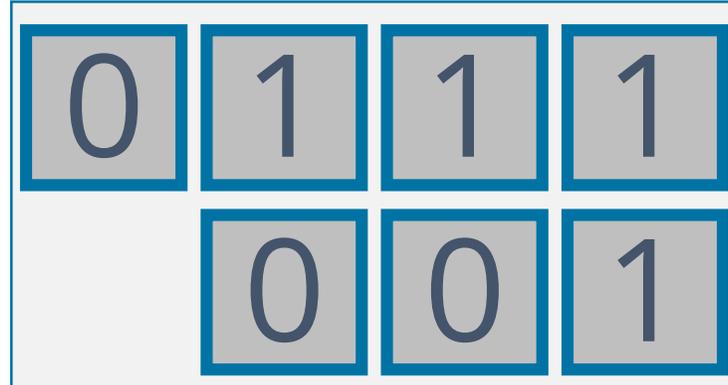
Coded packet  $A \oplus B^*2$

Legend:

-  Bit with value x from packet A
-  Bit with value x from packet B
-  Coded bit with value x from packet A and packet B

# Decoding 3

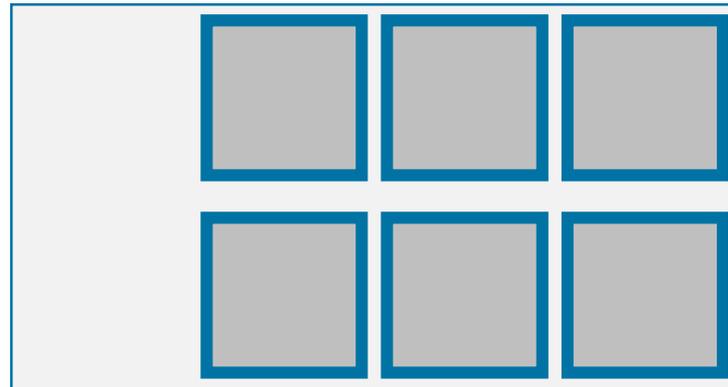
Coded packet  $A \oplus B^*2$



Original packet B

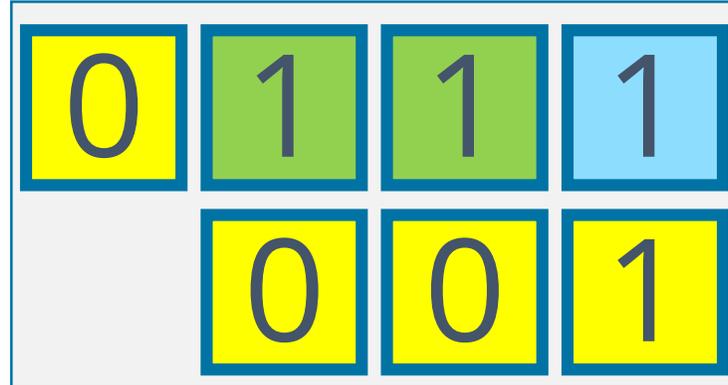
Original packet A

Original packet B



# Decoding 3

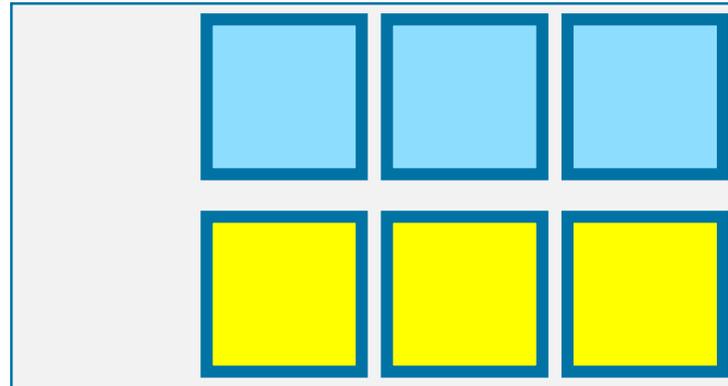
Coded packet  $A \oplus B^*2$



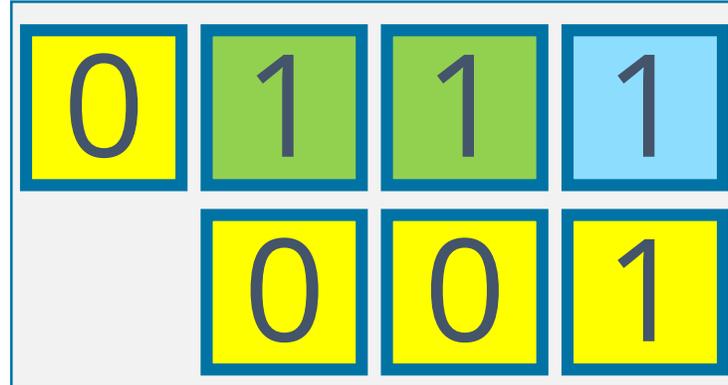
Original packet B

Original packet A

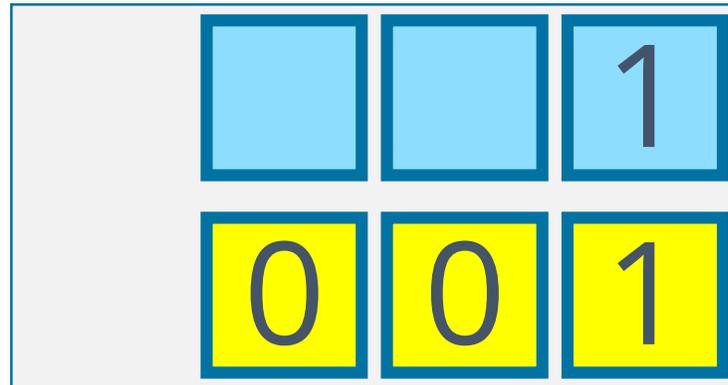
Original packet B



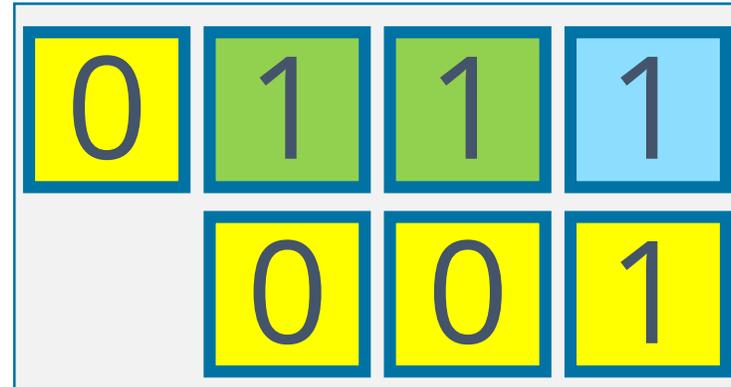
# Decoding 3



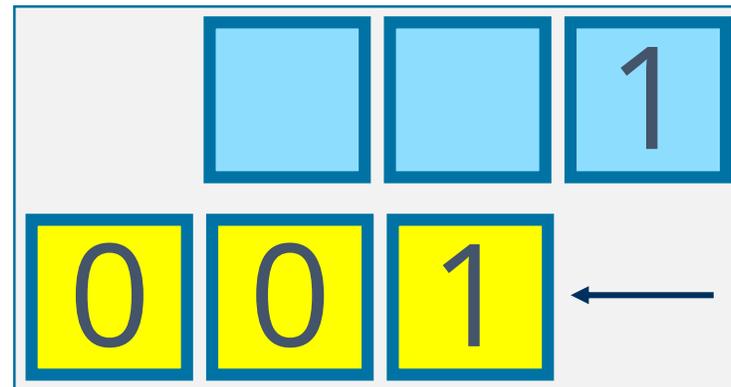
Simple reading



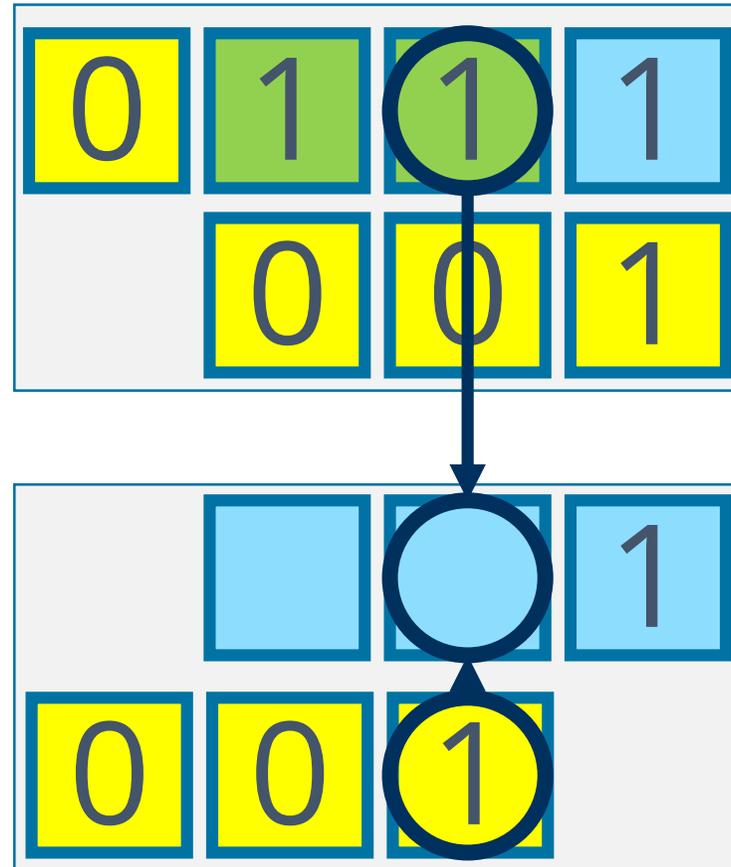
# Decoding 3



Restore old shift

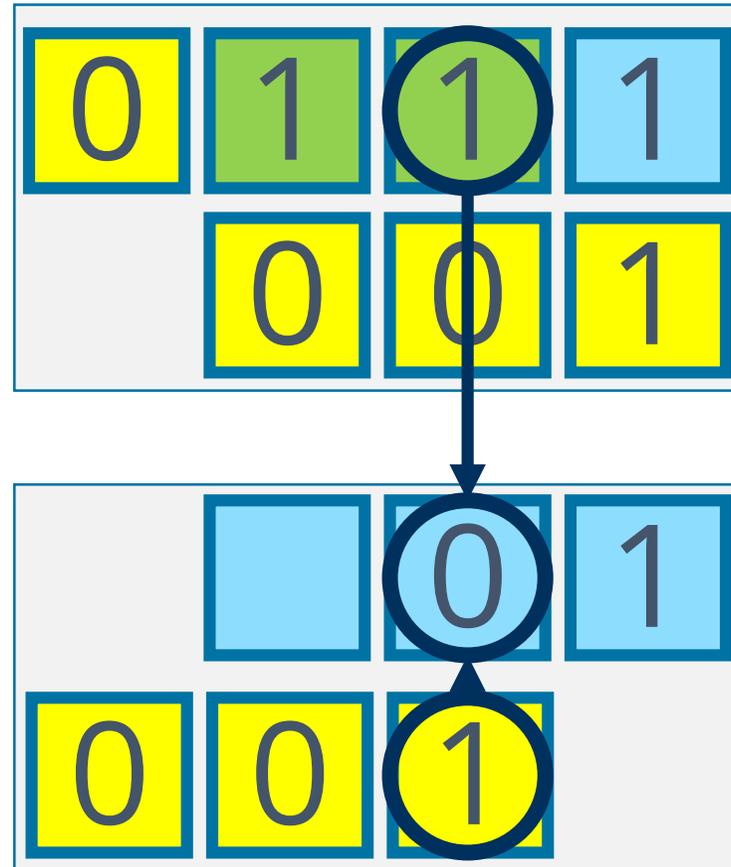


# Decoding 3



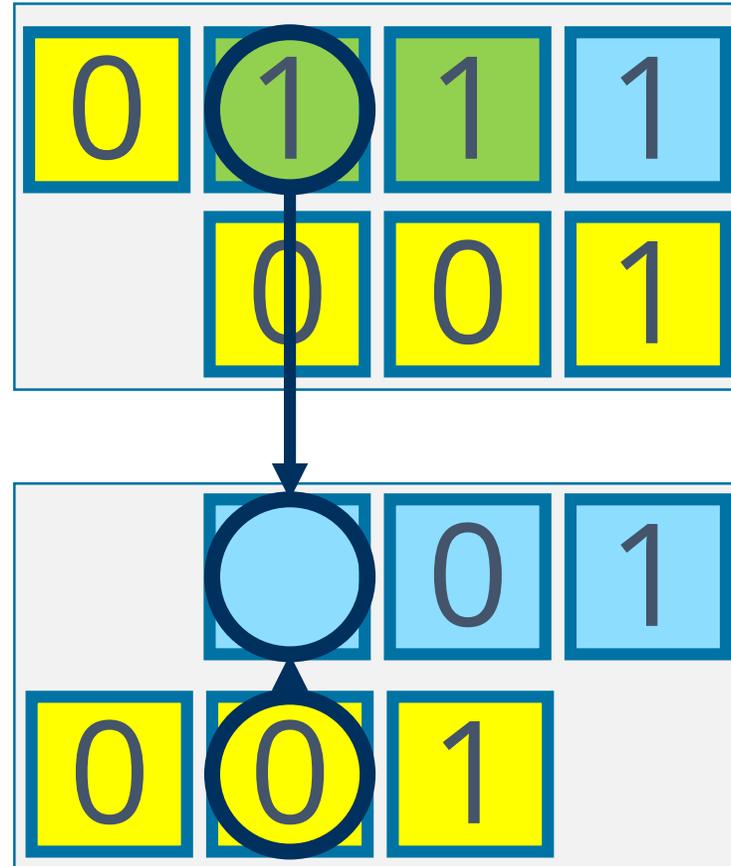
XOR:  $1 \oplus 1$

# Decoding 3



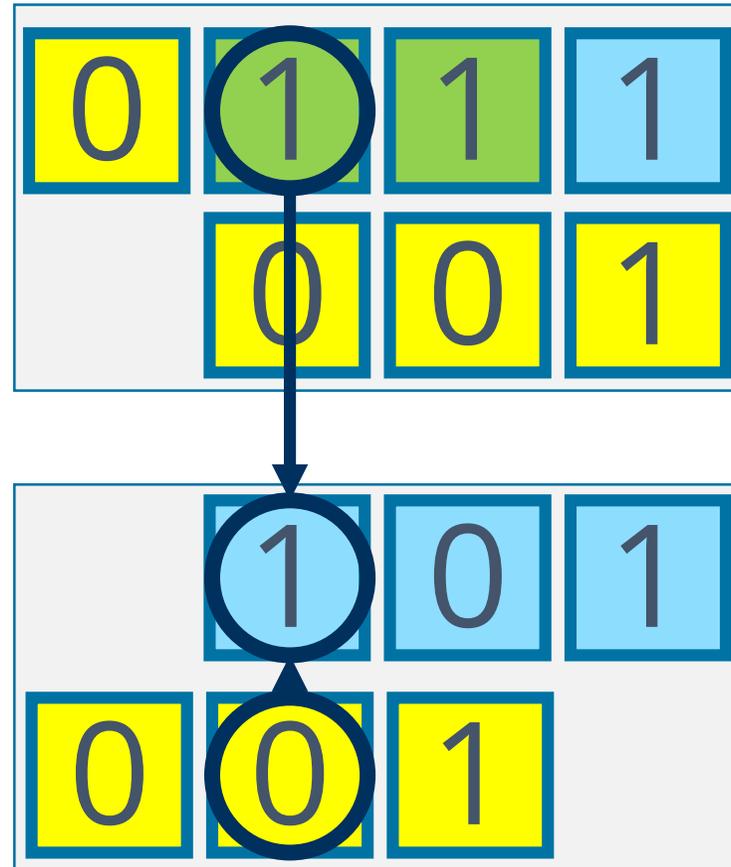
$$\text{XOR: } 1 \oplus 1 = 0$$

# Decoding 3



XOR:  $1 \oplus 0$

# Decoding 3

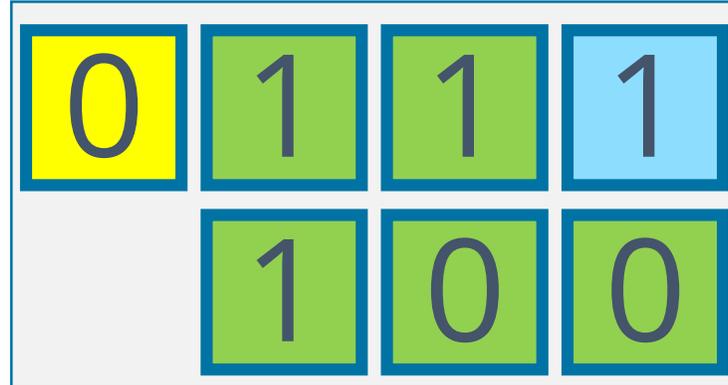


XOR:  $1 \oplus 0 = 1$

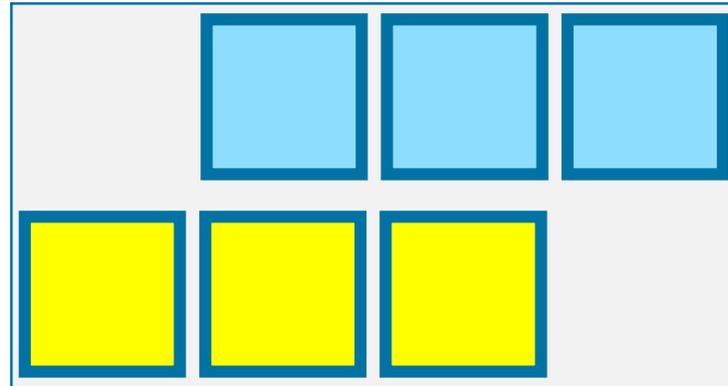


# Decoding 4

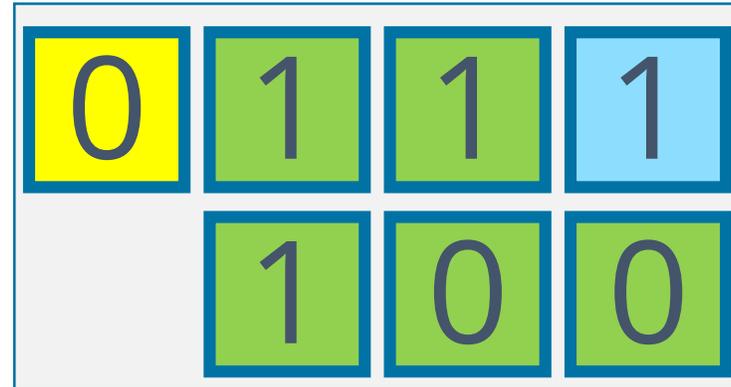
Coded packet  $A \oplus B^*2$



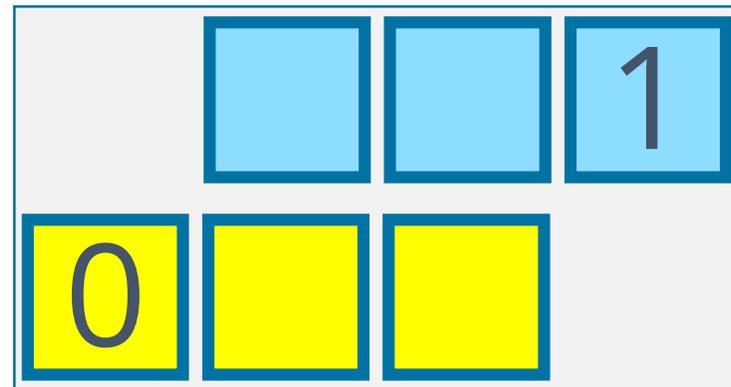
Coded packet  $A \oplus B$



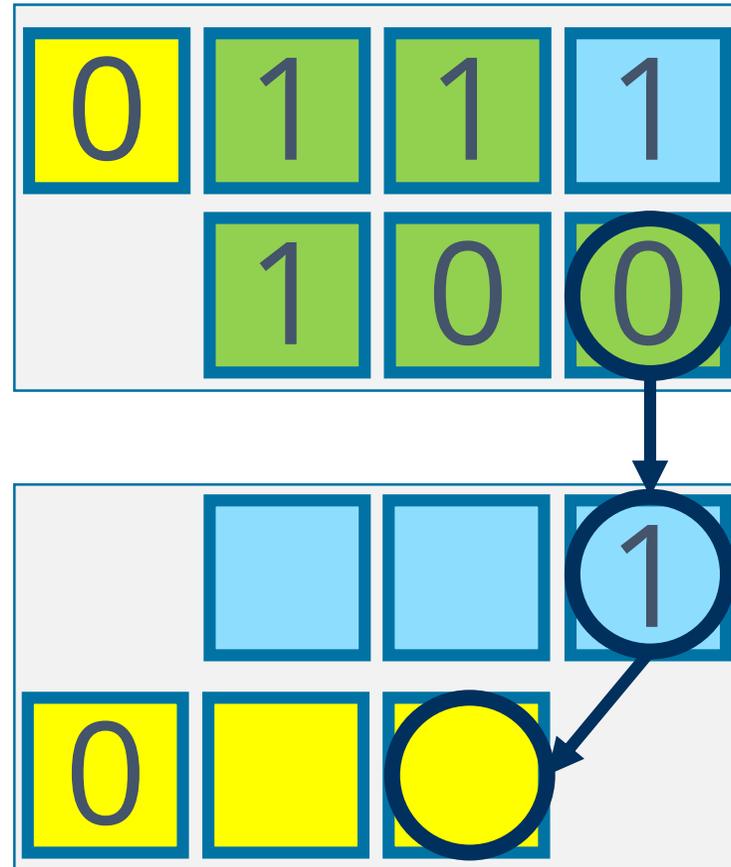
# Decoding 4



Simple reading

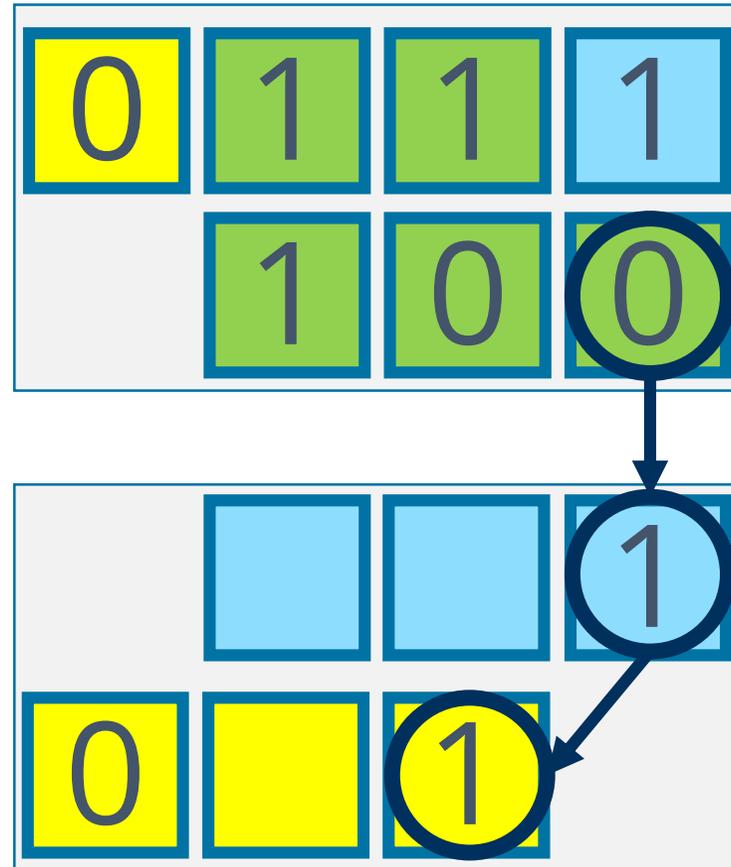


# Decoding 4



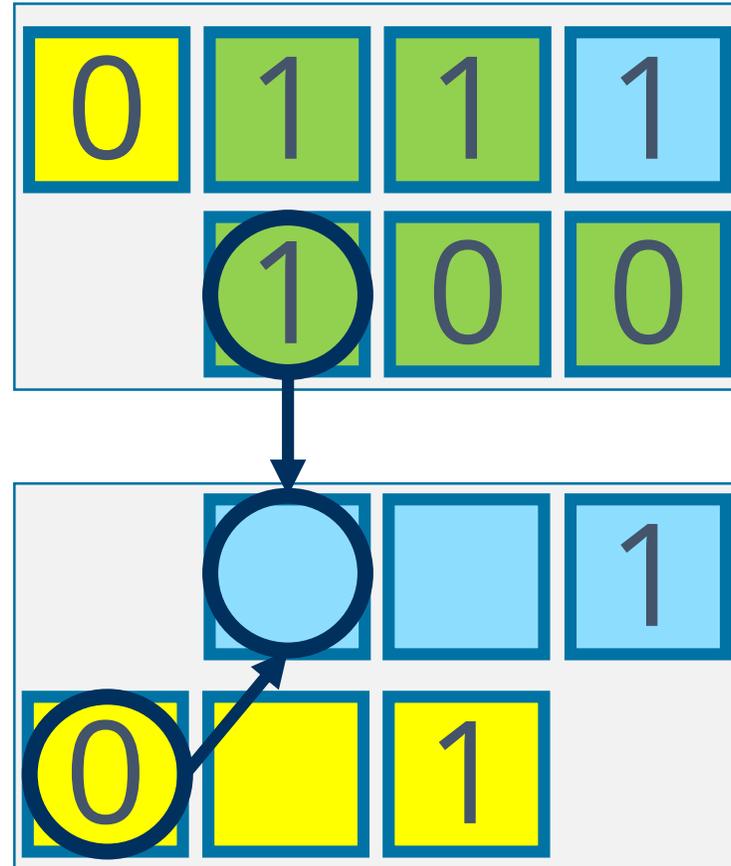
XOR:  $0 \oplus 1$

# Decoding 4



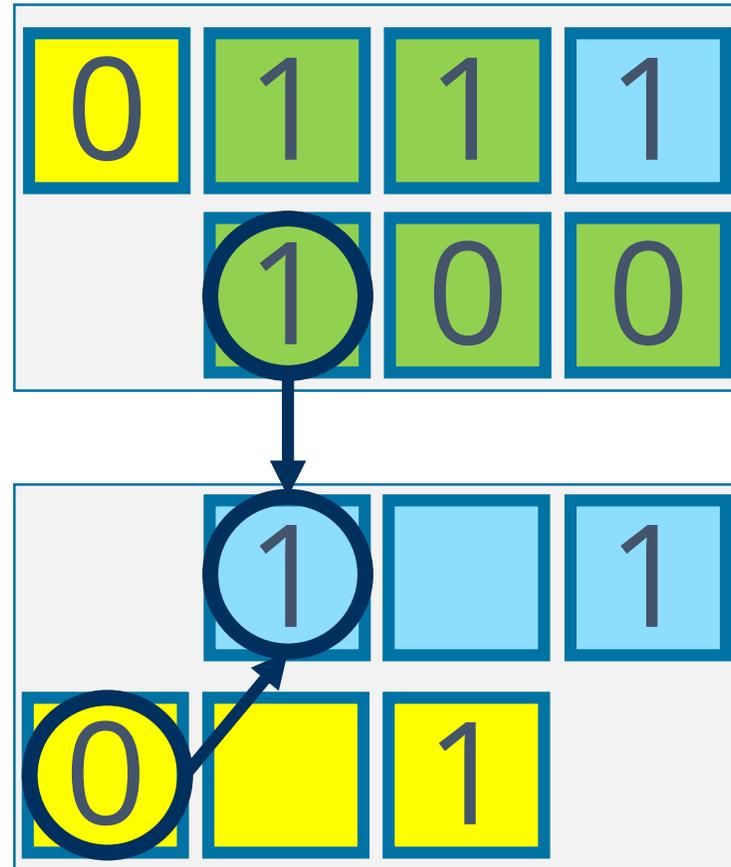
XOR:  $0 \oplus 1$

# Decoding 4



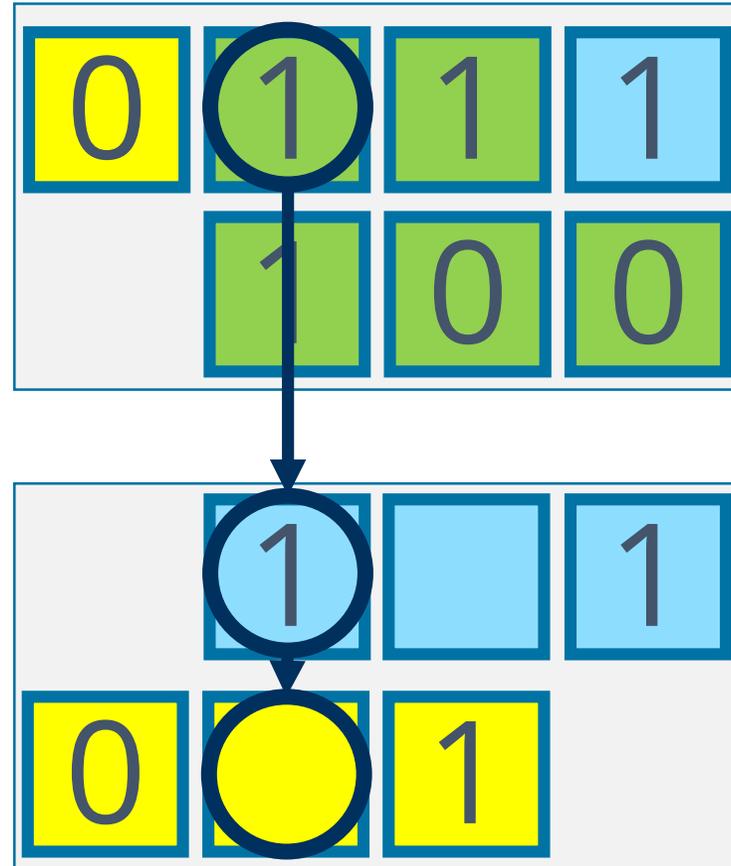
XOR:  $1 \oplus 0$

# Decoding 4



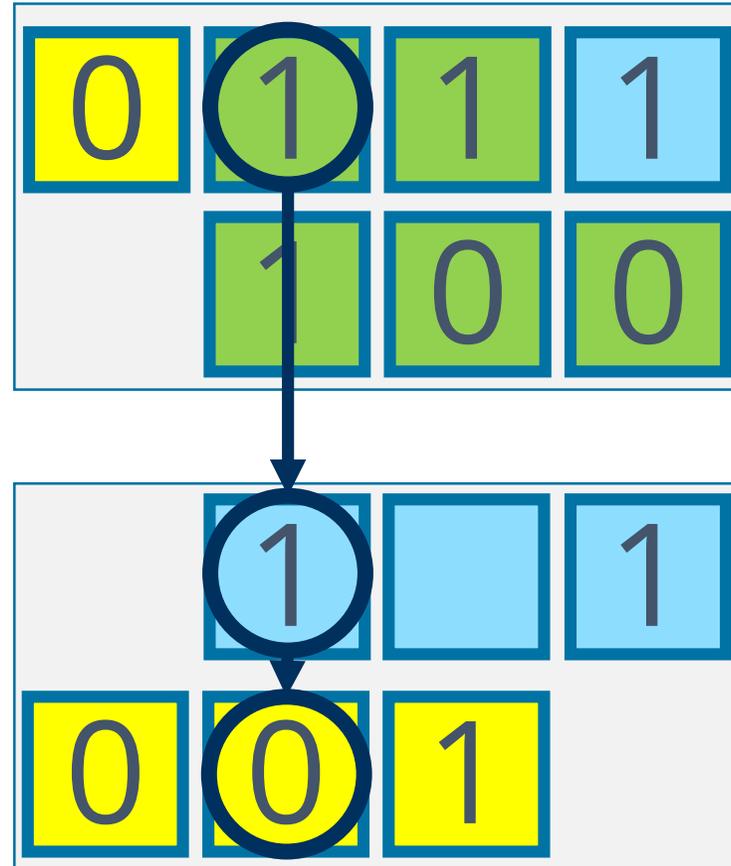
$$\text{XOR: } 1 \oplus 0 = 1$$

# Decoding 4



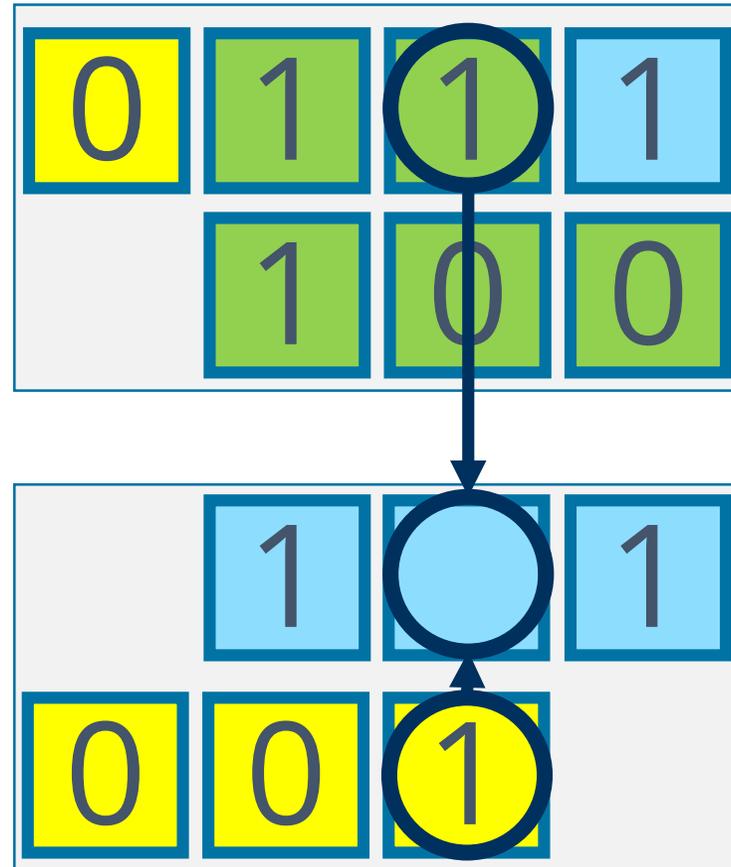
XOR:  $1 \oplus 1$

# Decoding 4



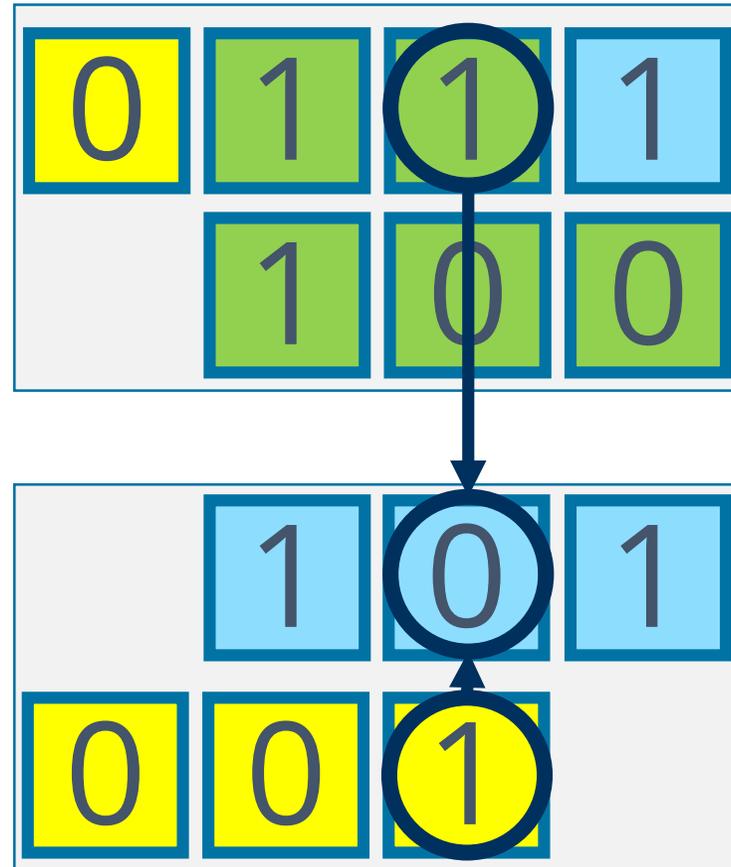
$$\text{XOR: } 1 \oplus 1 = 0$$

# Decoding 4



XOR:  $1 \oplus 1$

# Decoding 4



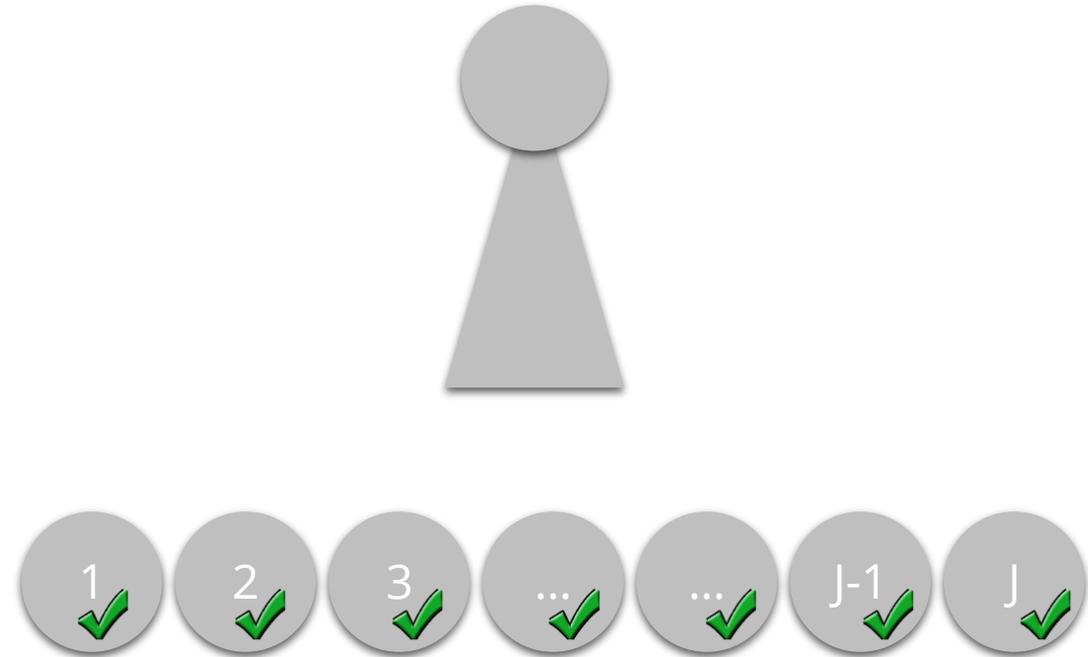
XOR:  $1 \oplus 1 = 0$



# Architecture and Example

- Assuming one node broadcasting information to several nodes over error-prone medium
- All nodes are interested in the same content
- Assuming we have only two messages A and B with three bits of information (to make it easy)
- Still not good enough now? Don't say repetition coding!

1	0	1	A
0	0	1	B
1	0	0	XOR coded

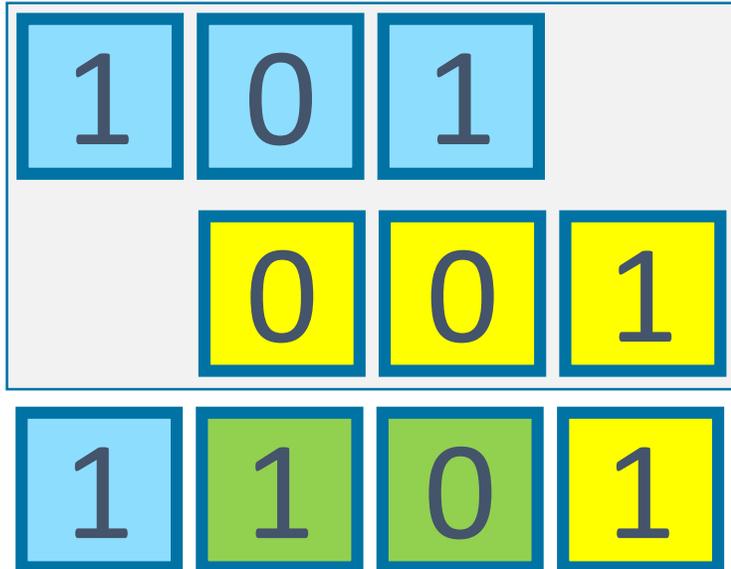


				1	0	1					
				0	0	1					
								1	0	0	
0	1	1	1	0	1	1	1	0	1	1	1

# Overhead consideration

- For packets with  $B$  bits and  $S$  shift bits, the overhead  $O$  is simply  $S/B$ .
- IP packets of size 1500 byte and 2 bits shift  $\rightarrow$  overhead is marginal.
- How to code multiple packets with bit shifting?
- How many combinations can be achieved with  $G$  packets and  $S$  shifts?
- What else would be needed?

# Encoding - Shifted XOR



Different direction



More shift

# Zig Zag Summary

- Very simple extension to the XOR world
- Overhead due to shifting (in our example from 3 to 4 bits)
- Very low complexity due to simple XOR operations
- Same can be achieved with Reed-Solomon and Random Linear Network Coding without overhead but more complexity
- Maybe interesting for IoT device (simple sensor networks)

# Digital Inter-Flow Network Coding: The Medium Access

# COPE

Katti, Rahul, Hu, Katabi, Medard, Crowcroft, **“XORs in the air: practical wireless network coding”**, SIGCOMM '06 Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications

# Network Coding in Wireless Networks

Core contribution by Katti et. al. „XOR in the Air“  
applying XOR coding to WIFI enabled meshed  
networks



Figure 7—Node locations for one floor of the testbed.

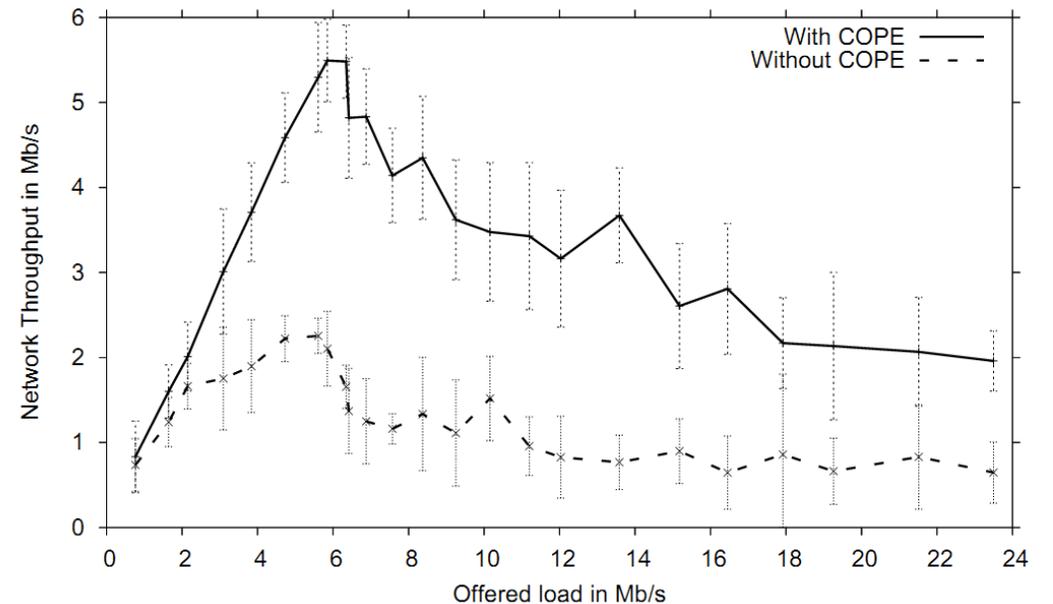


Figure 12—COPE can provide a several-fold (3-4x) increase in the throughput of wireless Ad hoc networks. Results are for UDP flows with randomly picked source-destination pairs, Poisson arrivals, and heavy-tail size distribution.

# Network Coding in Wireless Networks

COPE characteristics:

- Desktops
- Click software (user space)
- SRCR as routing algorithm
- Ad hoc mode
- RTS/CTS disabled
- Netgear hardware chips
- 802.11 a/g

# CATWOMAN

M. Hundeboll, J. Leddet-Pedersen, J. Heide, M.V. Pedersen, S.A. Rein, and F.H.P. Fitzek, **“Catwoman: Implementation and performance evaluation of ieee 802.11 based multi-hop networks using network coding,”** in I E E E V T S Vehicular Technology Conference. Proceedings. 2012, IEEE.

F. Zjao, M. Medard, M. Hundeboll, J. Ledet-Pedersen, S.A. Rein, and F.H.P. Fitzek, **“Comparison of analytical and measured performance results on network coding in ieee 802.11 ad-hoc networks,”** in The 2012 International Symposium on Network Coding, June 2012.

# CATWOMAN (2011)



Foto: Torsten Proß, Jeibmann Photographik

# CATWOMAN (2011)

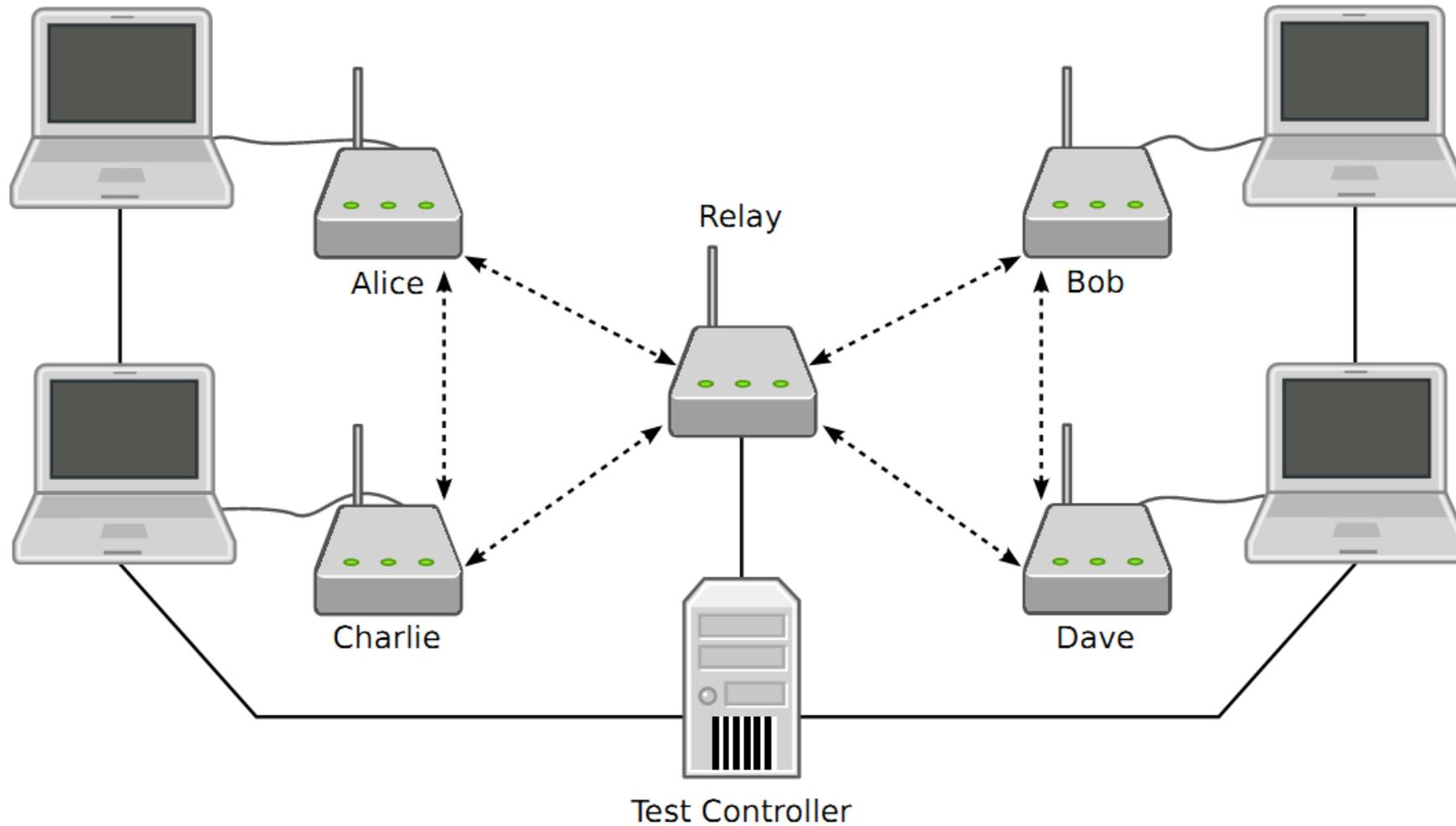
- Multihop network based on BATMAN routing (draft RFC)
- Implementation of network coding on real WiFi access points
- Multi hop
- Part of Linux Kernel 3.10



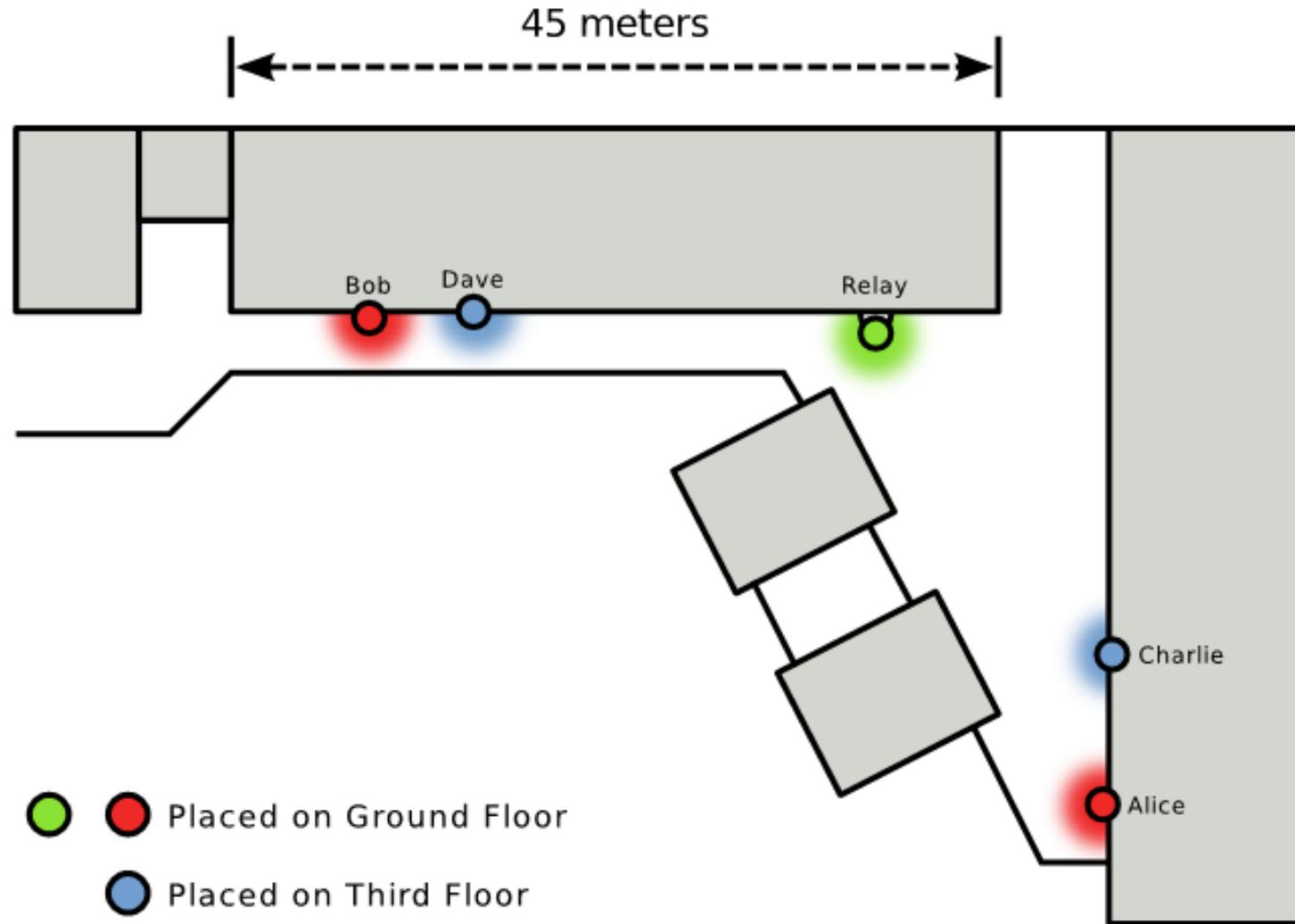
# CATWOMAN: Scenarios under Investigation



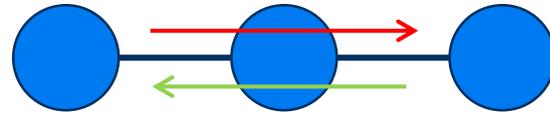
# CATWOMAN: Testbed



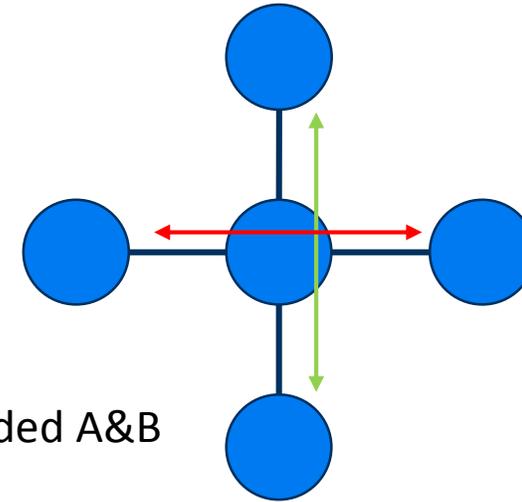
# CATWOMAN: Testbed Placement



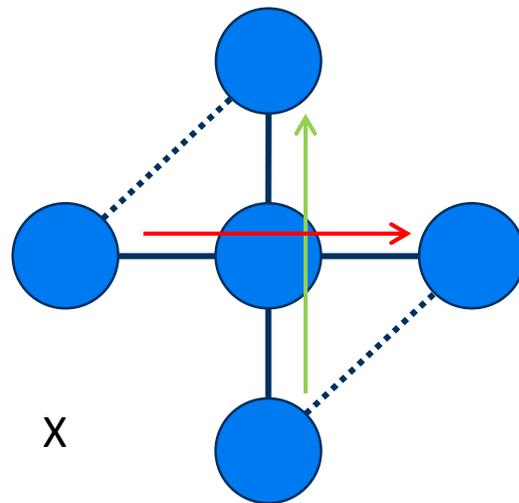
# CATWOMAN Topologies



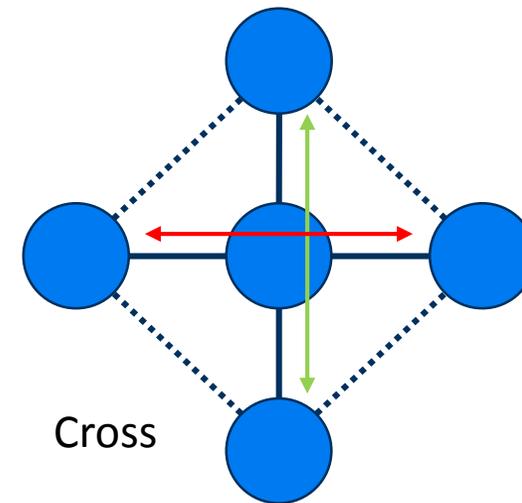
Alice and Bob



Extended A&B



X



Cross

# CATWOMAN Metric

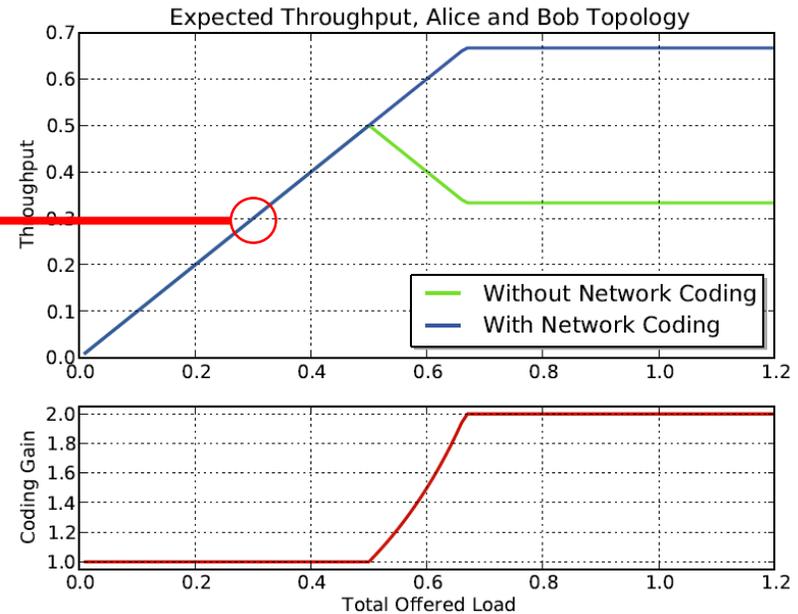
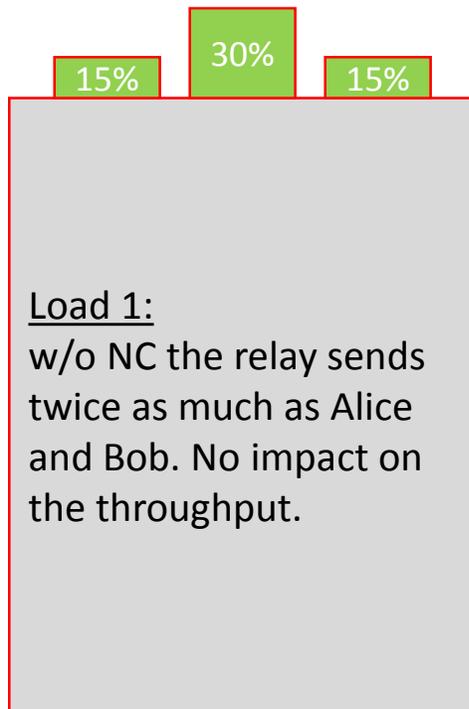
- Throughput [bit/s]
  - Bandwidth used
- Losses [%]
- Energy [J]
  - Activity level of send, receive, idle
  - CPU load
- Delay [s]

# Model for Alice and Bob (symmetric traffic)

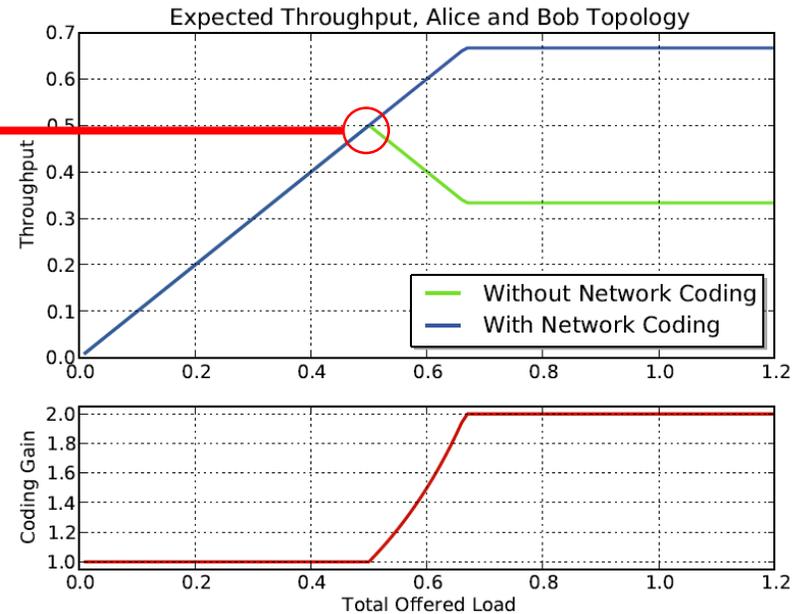
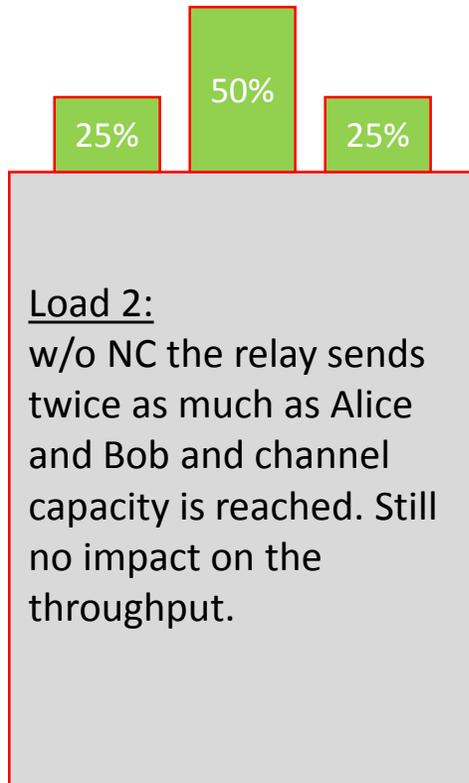
# Assumption

- Alice and Bob have no direct connection
- Same amount of traffic is generated by Alice and Bob
- Medium Access Control (MAC) is based on IEEE802.11, i.e. CSMA/CA

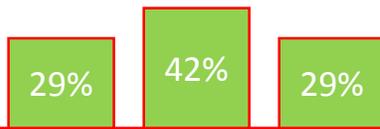
# IEEE802.11 Alice & Bob Throughput Model



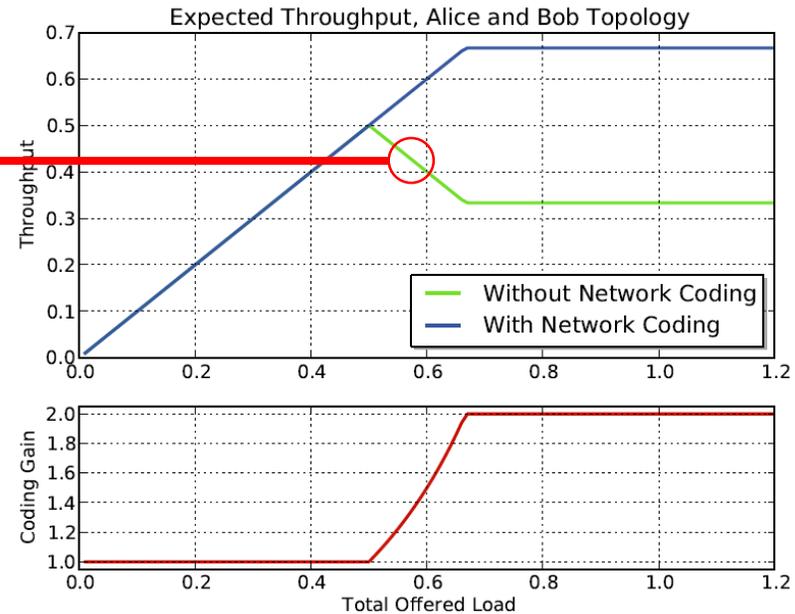
# IEEE802.11 Alice & Bob Throughput Model



# IEEE802.11 Alice & Bob Throughput Model



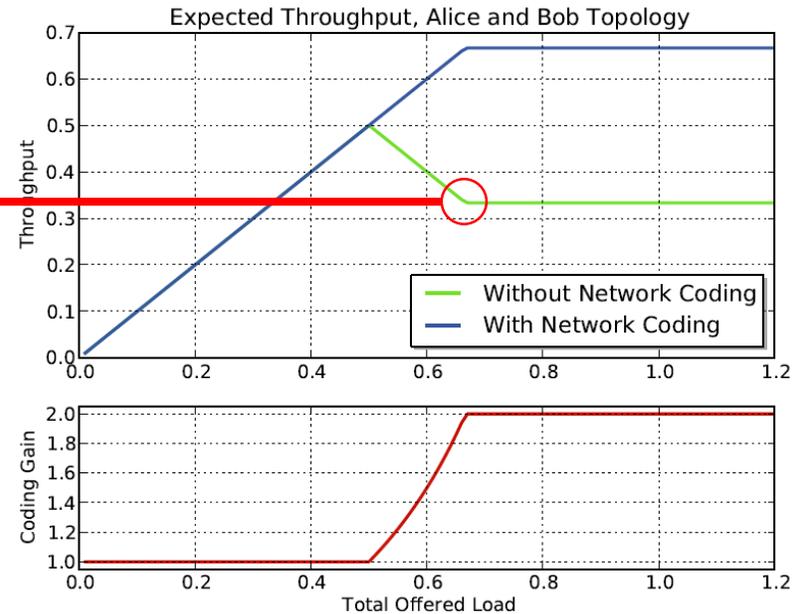
Load 3:  
w/o NC Alice and Bob are „stealing“ the capacity from the relay. 802.11 fairness destroys the performance of the system.



# IEEE802.11 Alice & Bob Throughput Model



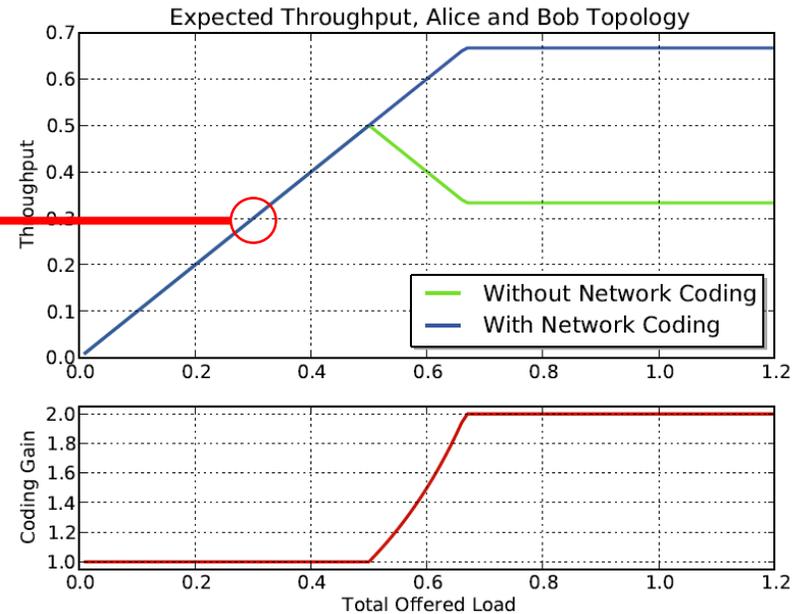
Load 3:  
w/o NC Alice and Bob are „stealing“ the capacity from the relay. 802.11 fairness destroys the performance of the system.



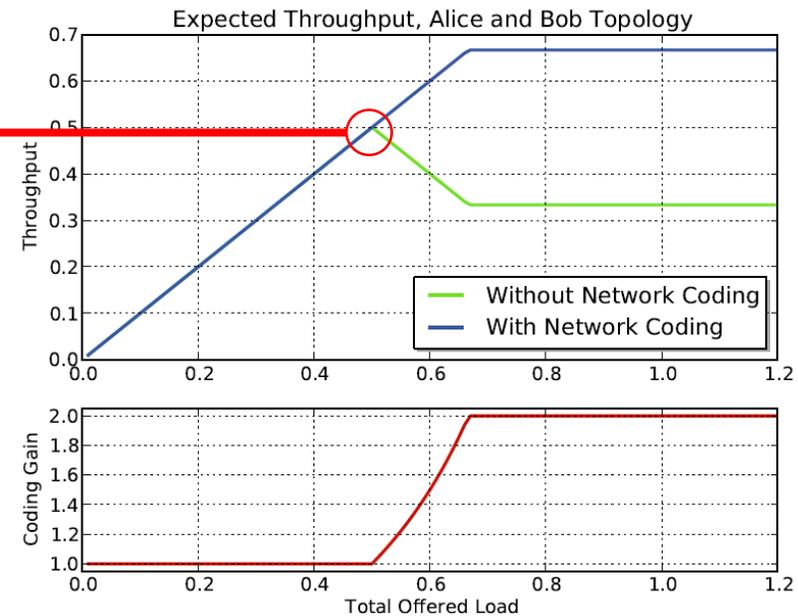
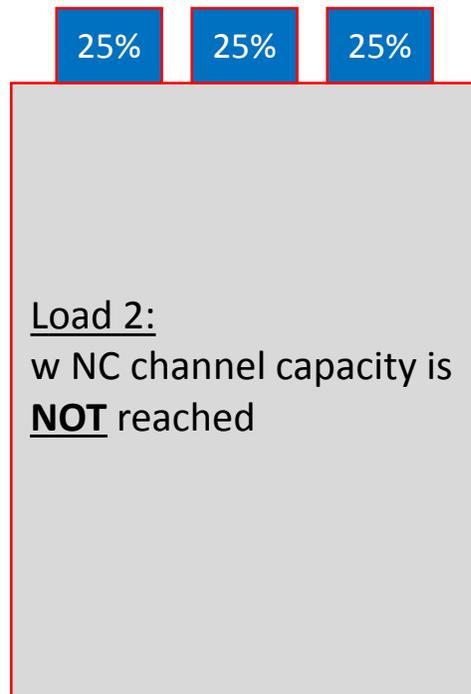
# IEEE802.11 Alice & Bob Throughput Model

15% 15% 15%

Load 1:  
w NC the relay sends the same amount of data as Alice or Bob.



# IEEE802.11 Alice & Bob Throughput Model



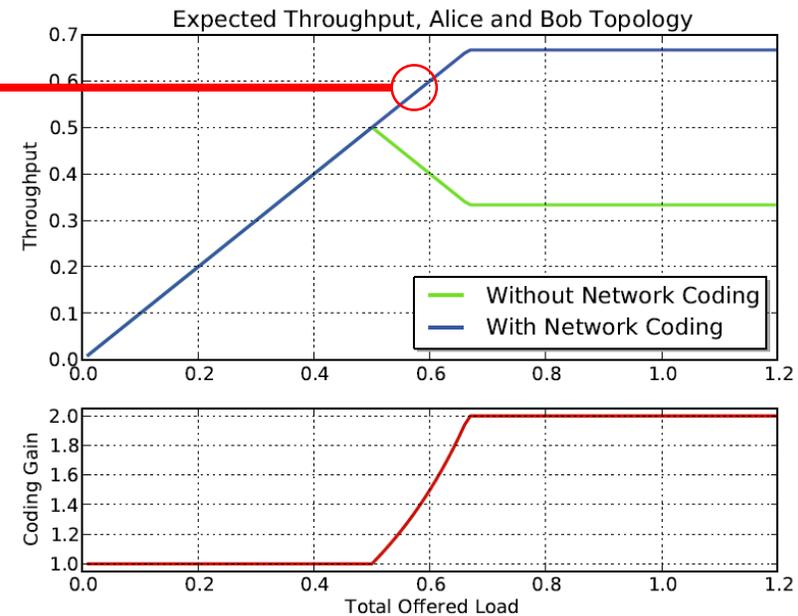
# IEEE802.11 Alice & Bob Throughput Model

29%

29%

29%

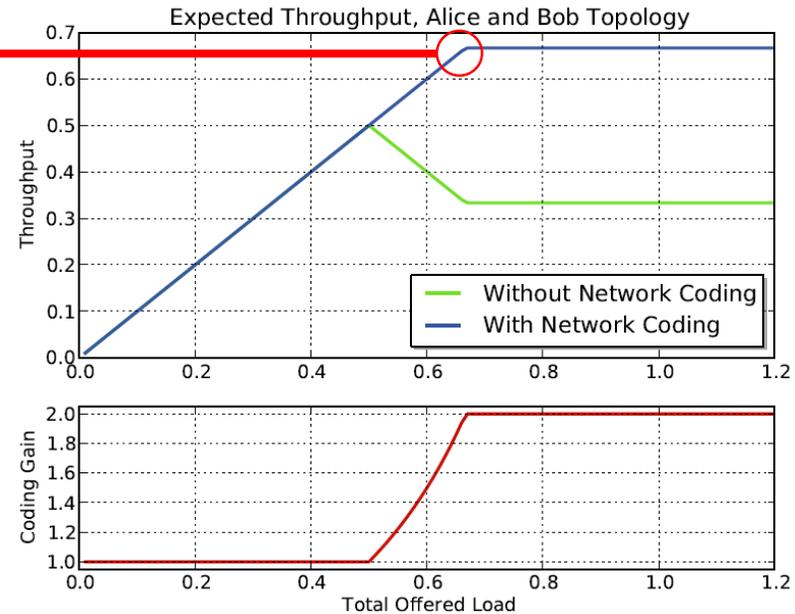
Load 3:  
w NC Alice, Bob and the relay live in perfect harmony, each of the entities requests one third of the capacity



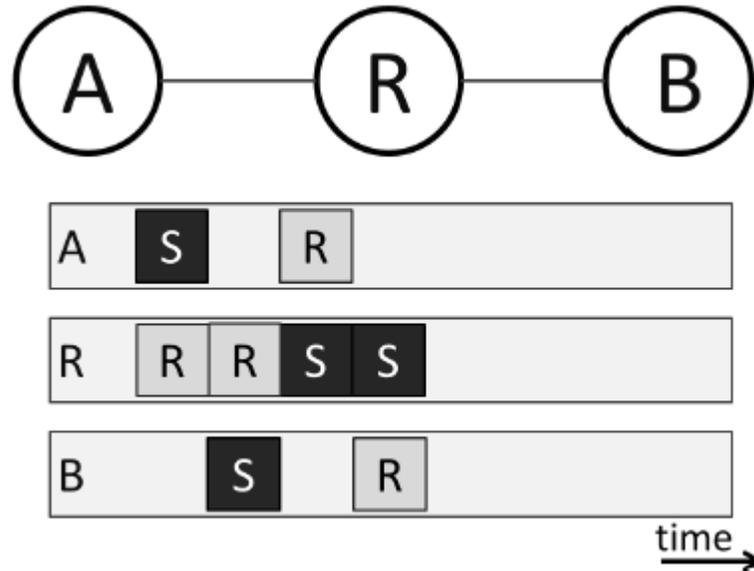
# IEEE802.11 Alice & Bob Throughput Model

33% 33% 33%

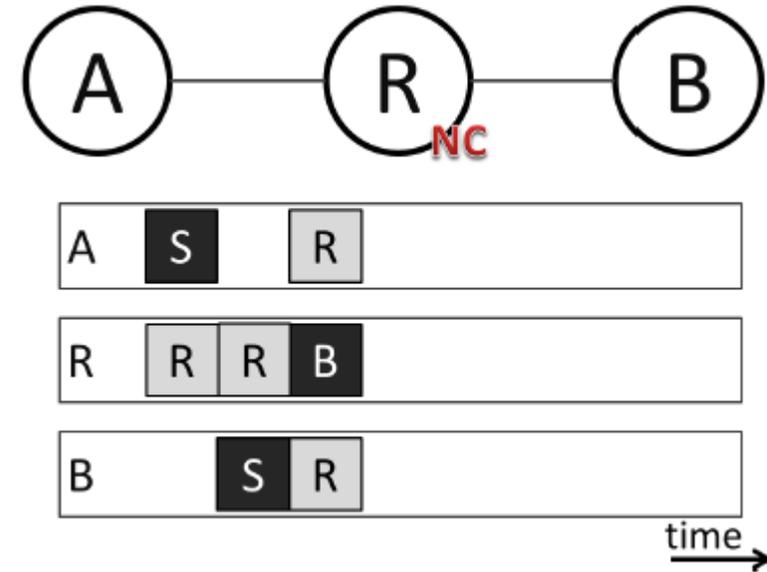
Load 4:  
w NC Alice, Bob and the relay live in perfect harmony now the channel capacity is reach and therefore the throughput remains constant.



# Alice and Bob: Activity Chart

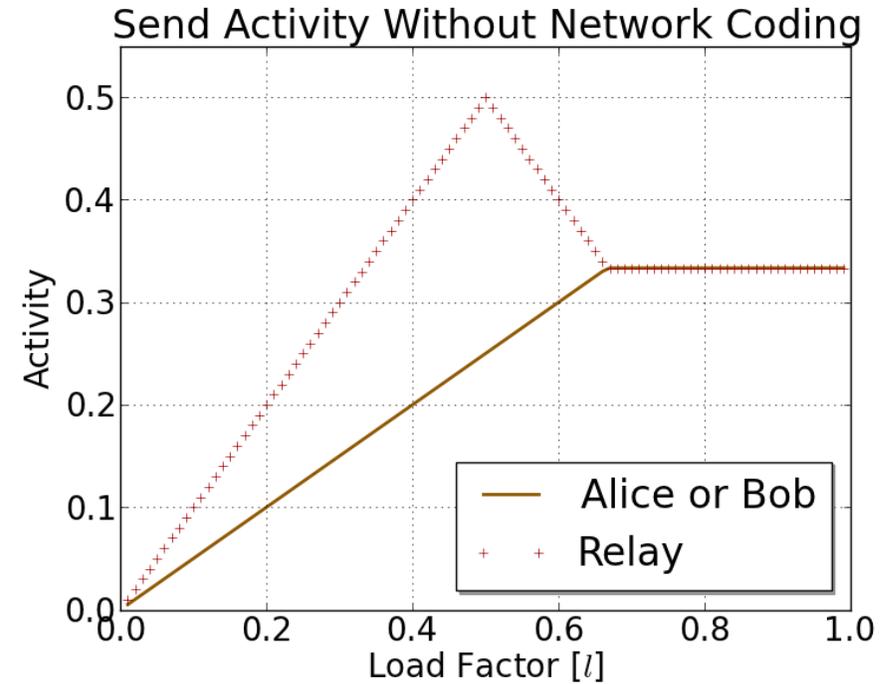
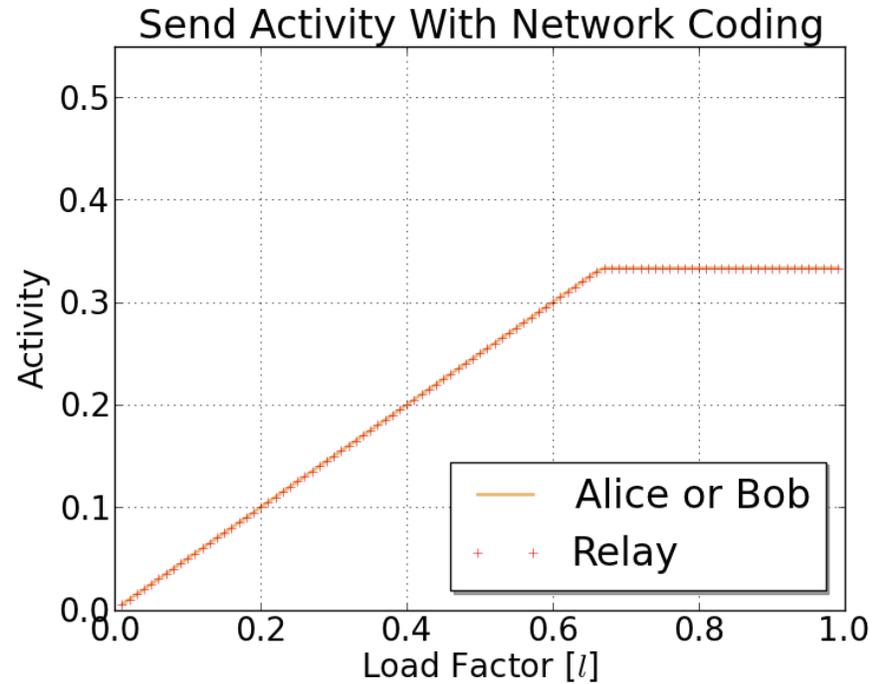


4 send  
4 receive  
rest idle

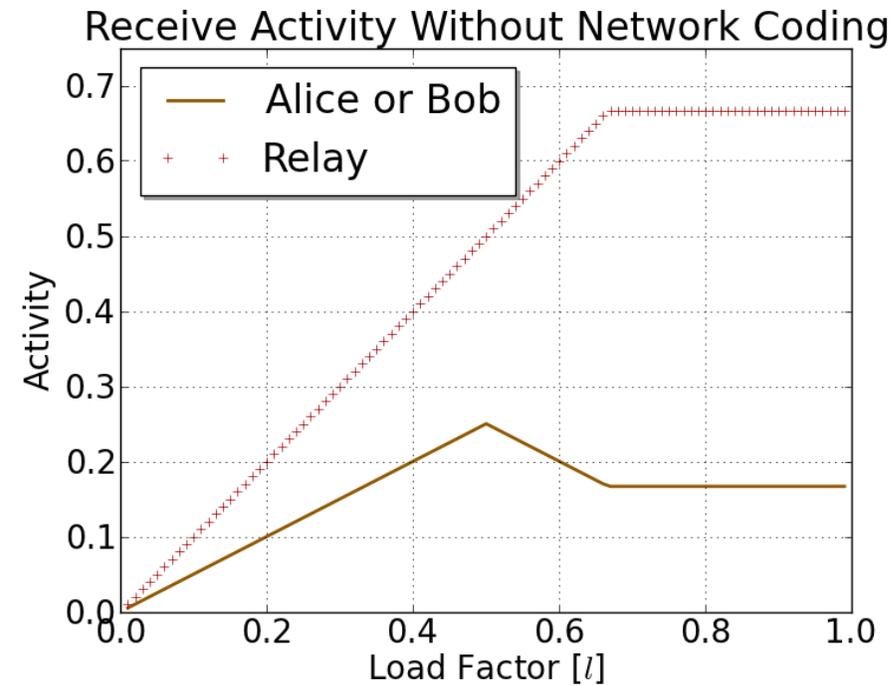
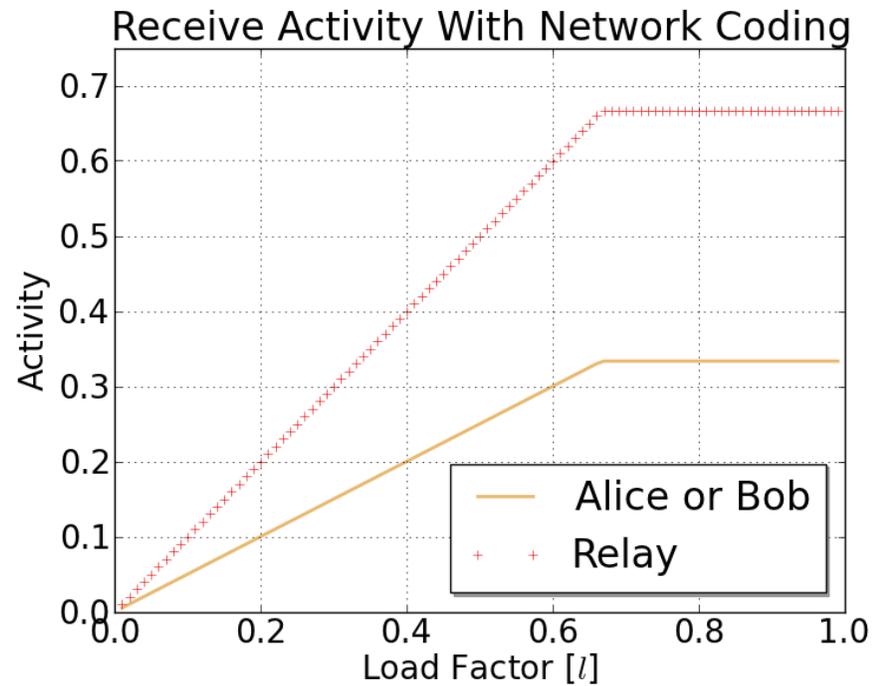


3 send  
4 receive  
rest idle

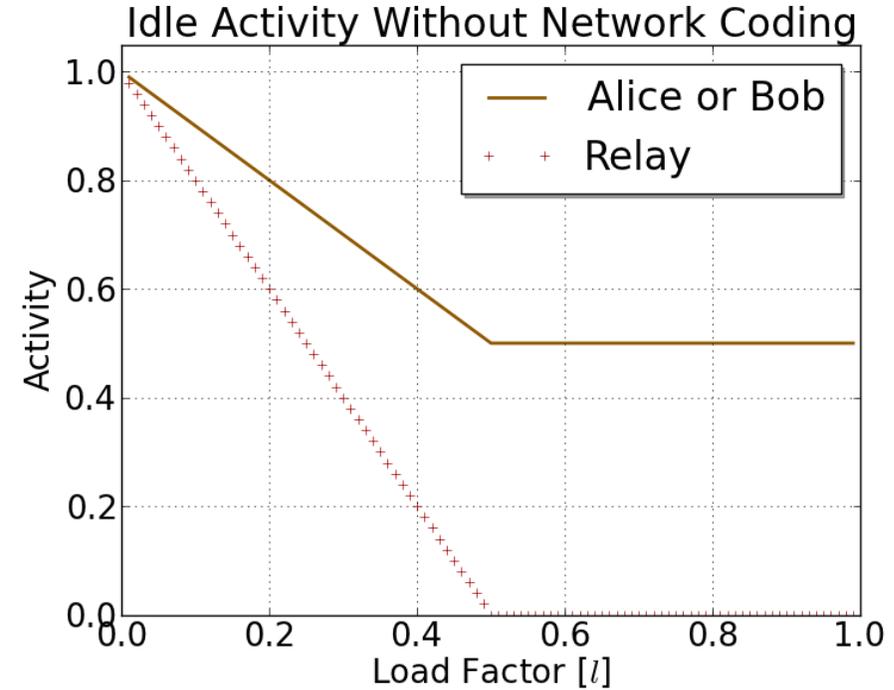
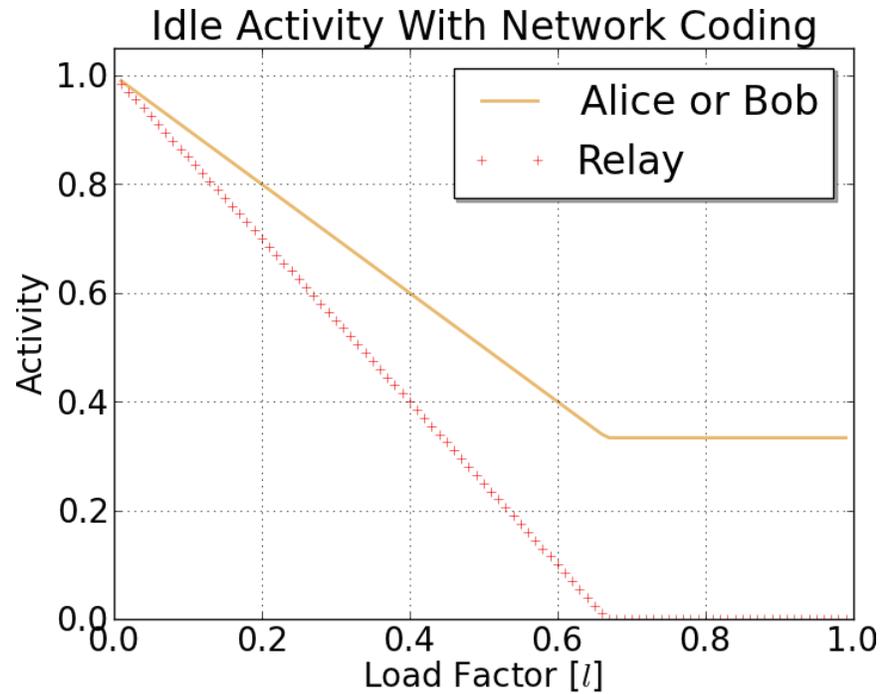
# Send Activity Model



# Receive Activity Model



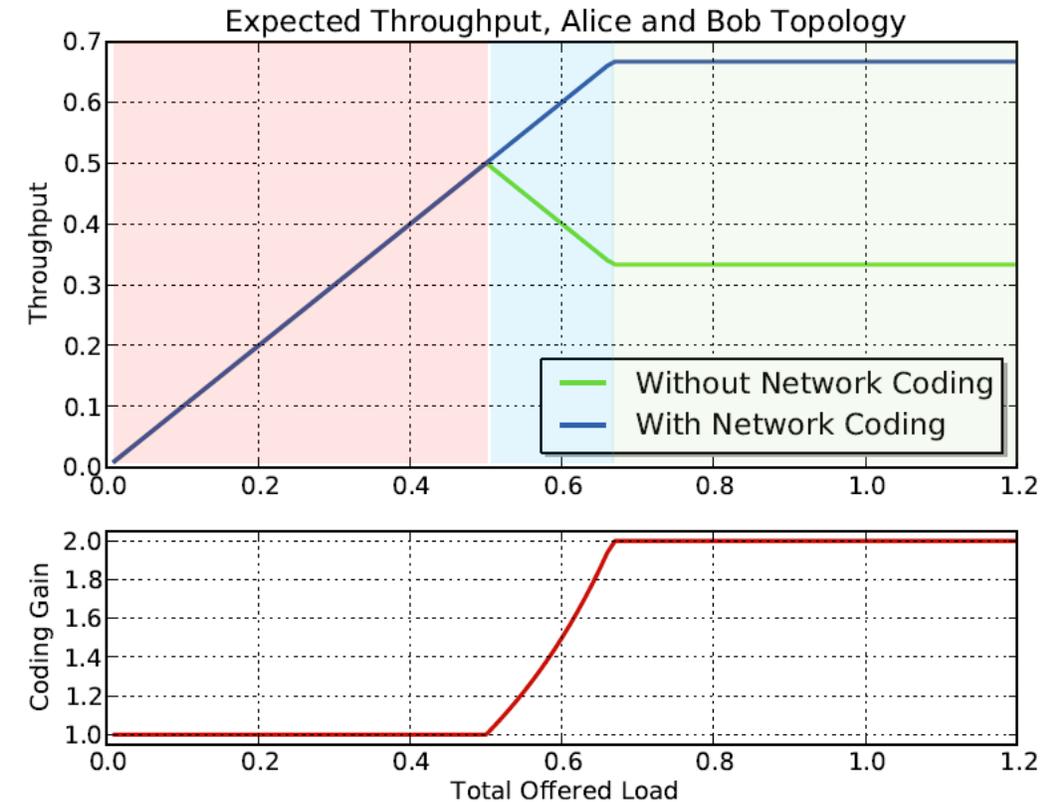
# Idle Activity Model



# Activity Model



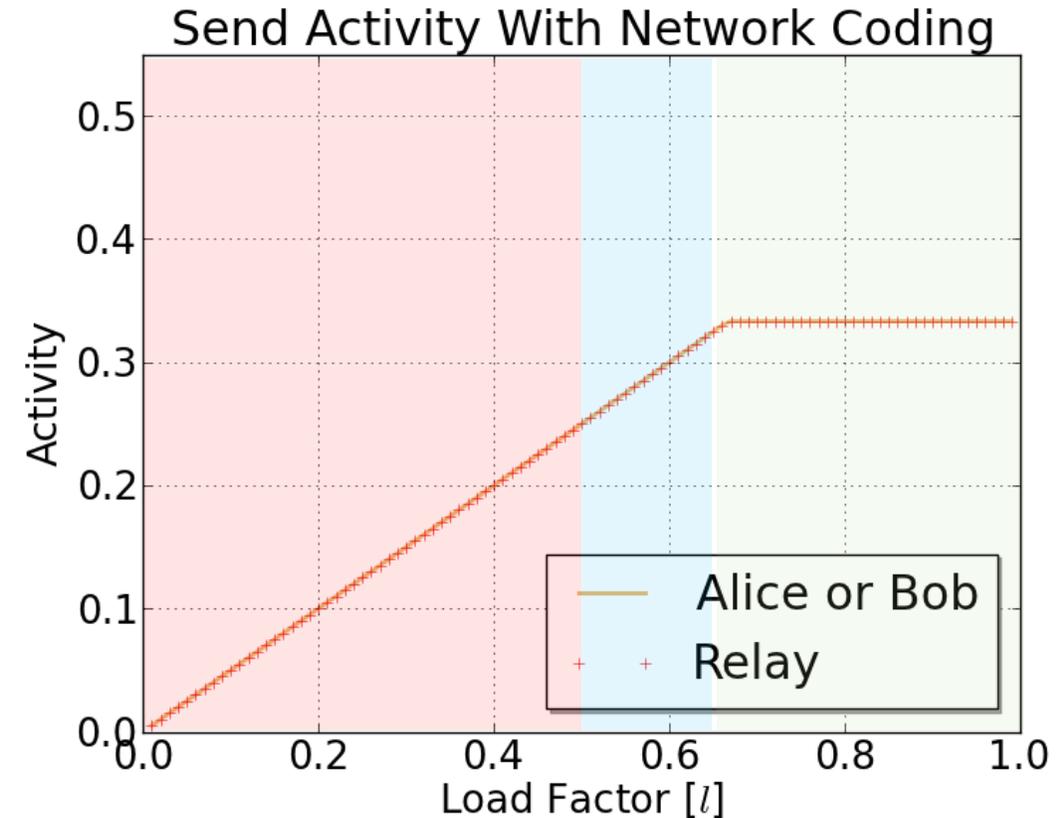
		Phase	I	II	III
		Load [ $l$ ]	0-0.5	0.5-0.6	0.6-1.0
Send [ $\alpha_s$ ]	WoNC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$l$	$1-l$	$\frac{1}{3}$
	NC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
Receive [ $\alpha_r$ ]	WoNC	A&B	$\frac{1}{2}l$	$\frac{1-l}{2}$	$\frac{1}{6}$
		R	$l$	$l$	$\frac{2}{3}$
	NC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$l$	$l$	$\frac{2}{3}$



# Activity Model



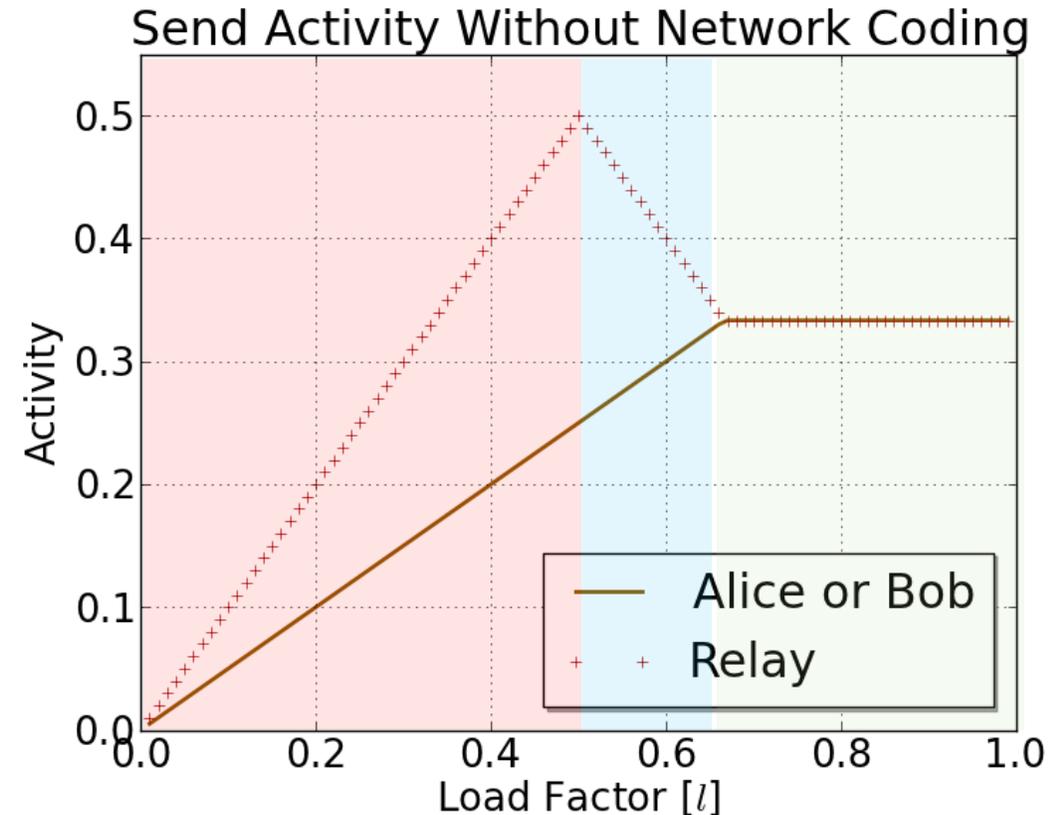
		Phase	I	II	III
		Load [ $l$ ]	0-0.5	0.5-0.6	0.6-1.0
Send [ $\alpha_s$ ]	WoNC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$l$	$1-l$	$\frac{1}{3}$
	NC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
Receive [ $\alpha_r$ ]	WoNC	A&B	$\frac{1}{2}l$	$\frac{1-l}{2}$	$\frac{1}{6}$
		R	$l$	$l$	$\frac{2}{3}$
	NC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$l$	$l$	$\frac{2}{3}$



# Activity Model



		Phase	I	II	III
		Load [ $l$ ]	0-0.5	0.5-0.6	0.6-1.0
Send [ $\alpha_s$ ]	WoNC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$l$	$1-l$	$\frac{1}{3}$
	NC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
Receive [ $\alpha_r$ ]	WoNC	A&B	$\frac{1}{2}l$	$\frac{1-l}{2}$	$\frac{1}{6}$
		R	$l$	$l$	$\frac{2}{3}$
	NC	A&B	$\frac{1}{2}l$	$\frac{1}{2}l$	$\frac{1}{3}$
		R	$l$	$l$	$\frac{2}{3}$



# Power Model

In order to derive total power we sum up the product of the activity level  $a$  and power level  $P$  of the individual states.

$$P_{total} = P_r * a_r + P_s * a_s + P_i * a_i$$

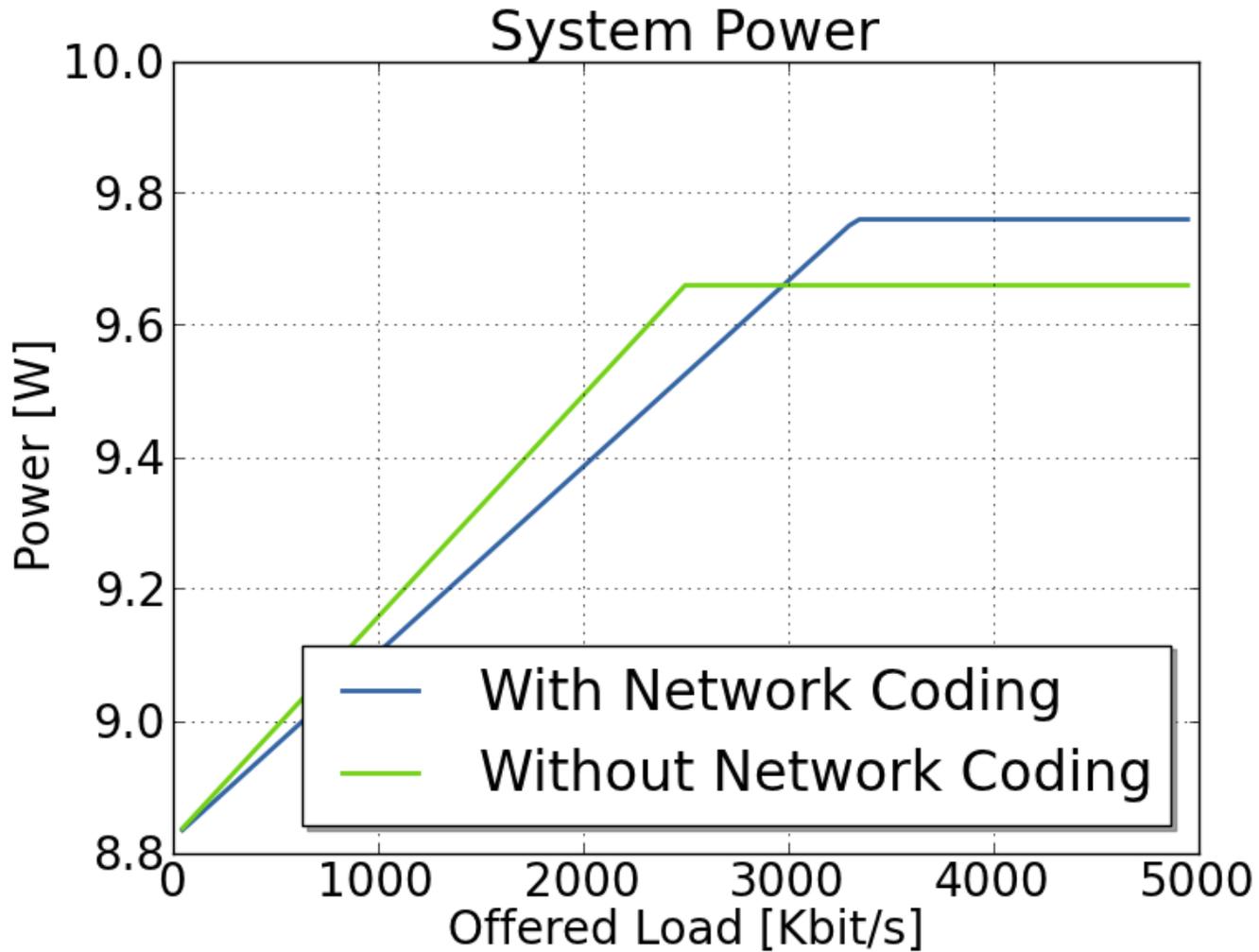
# Power Model

Out of our measurements for WHITEBOX we derive:

- Total power level sending:  $P_{\text{send}} = 3.48 \text{ W}$
- Total power level receiving:  $P_{\text{receive}} = 3.24 \text{ W}$
- Total power level idle:  $P_{\text{idle}} = 2.94 \text{ W}$

Later we assume a maximum capacity of the channel of 4.9 Mbit/s

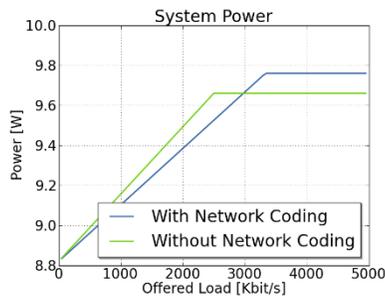
# Power Rate Model



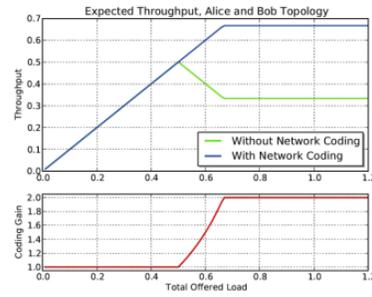
# Energy Model

$$Energy = Power * Time$$

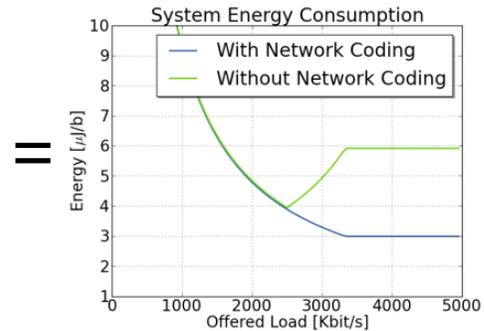
$$Energy \text{ per bit} = \frac{Power}{Throughput} = \frac{Joule}{bit}$$



power



throughput



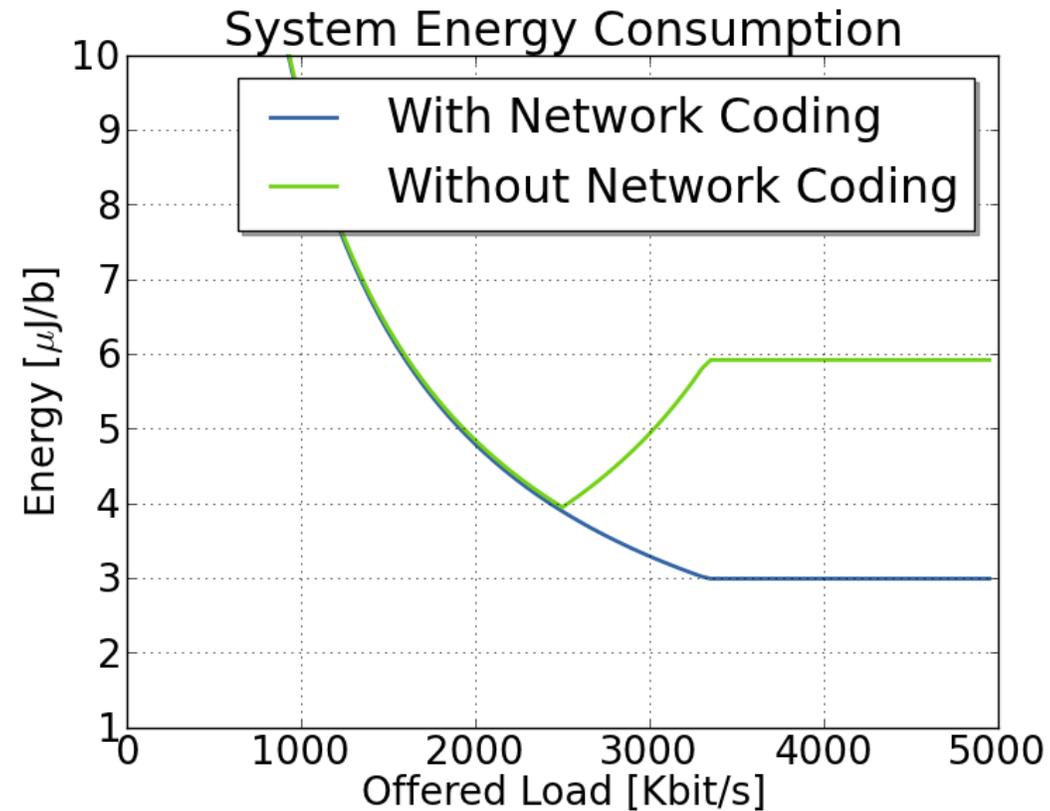
energy/bit

•

=

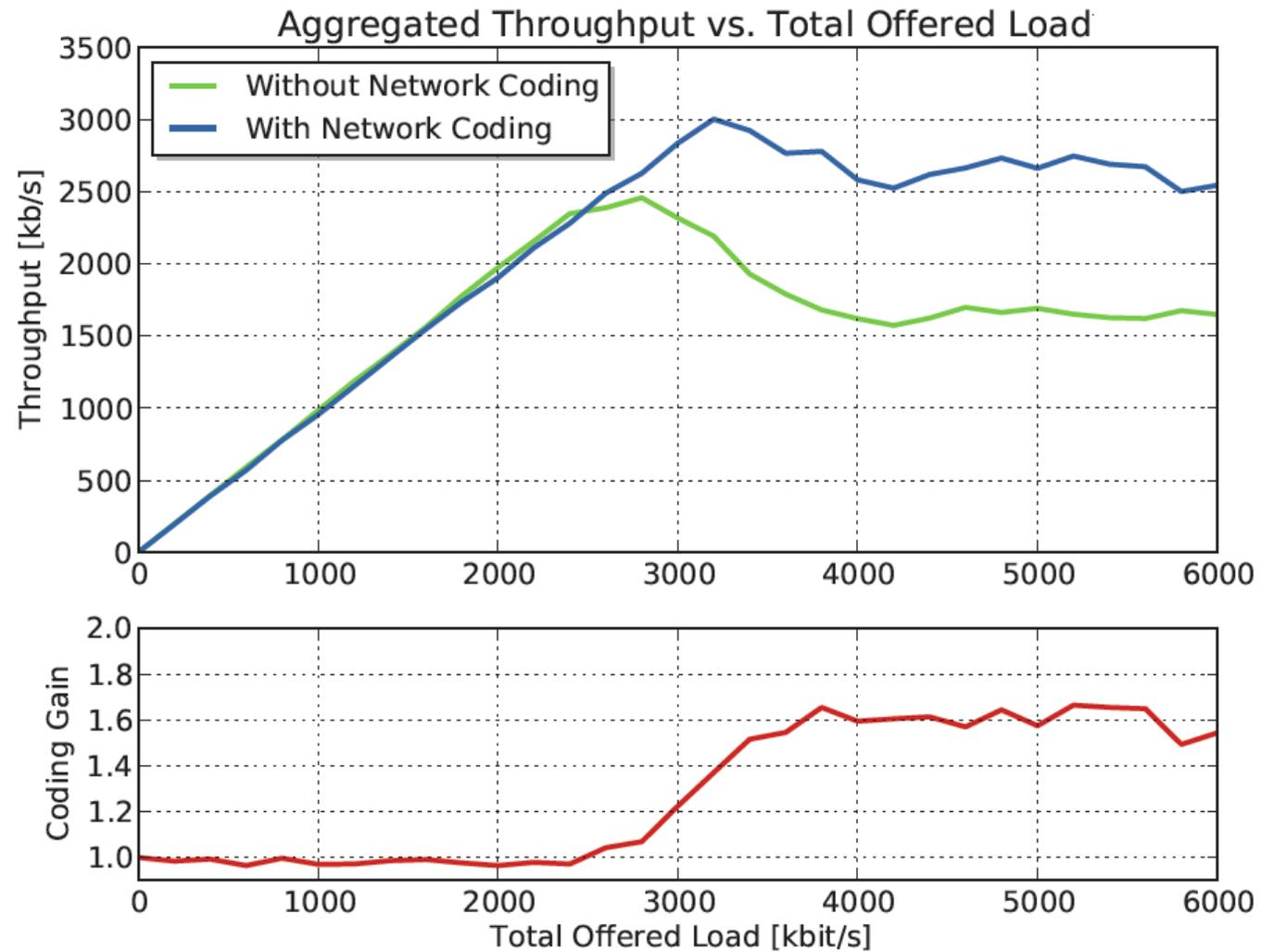
# Energy Model

$$\text{Energy per bit} = \frac{\text{Power}}{\text{Throughput}} = \frac{\text{Joule}}{\text{bit}}$$

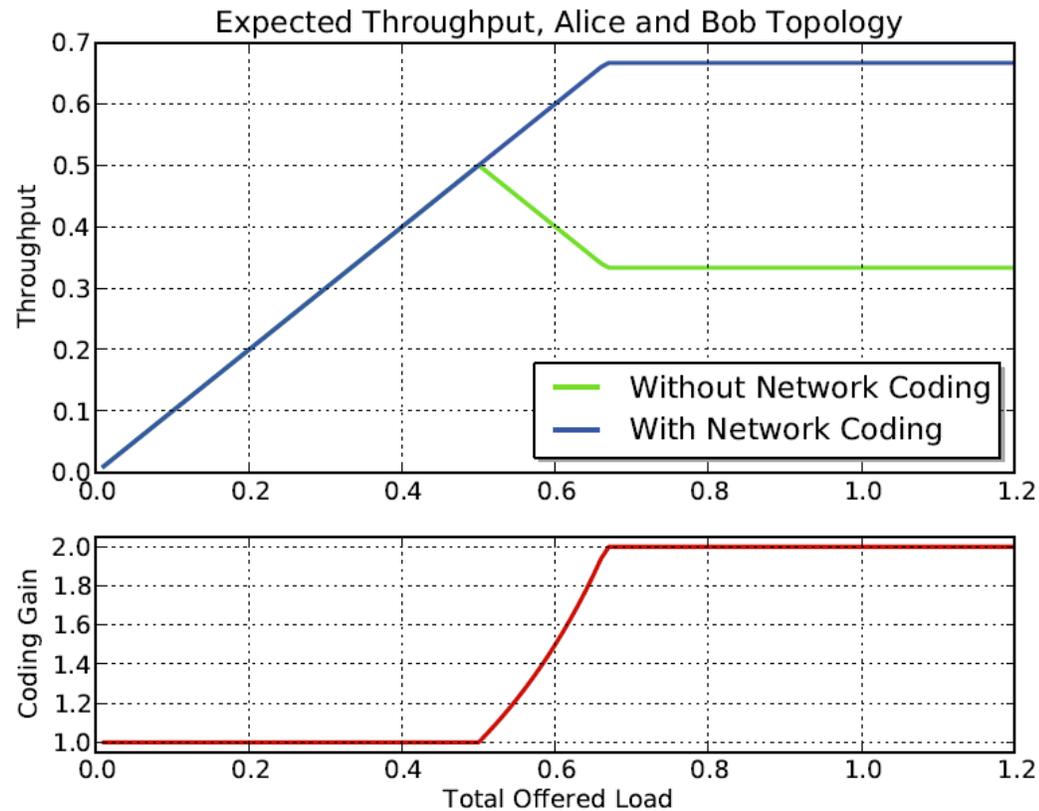


# CATWOMAN Measurement Results

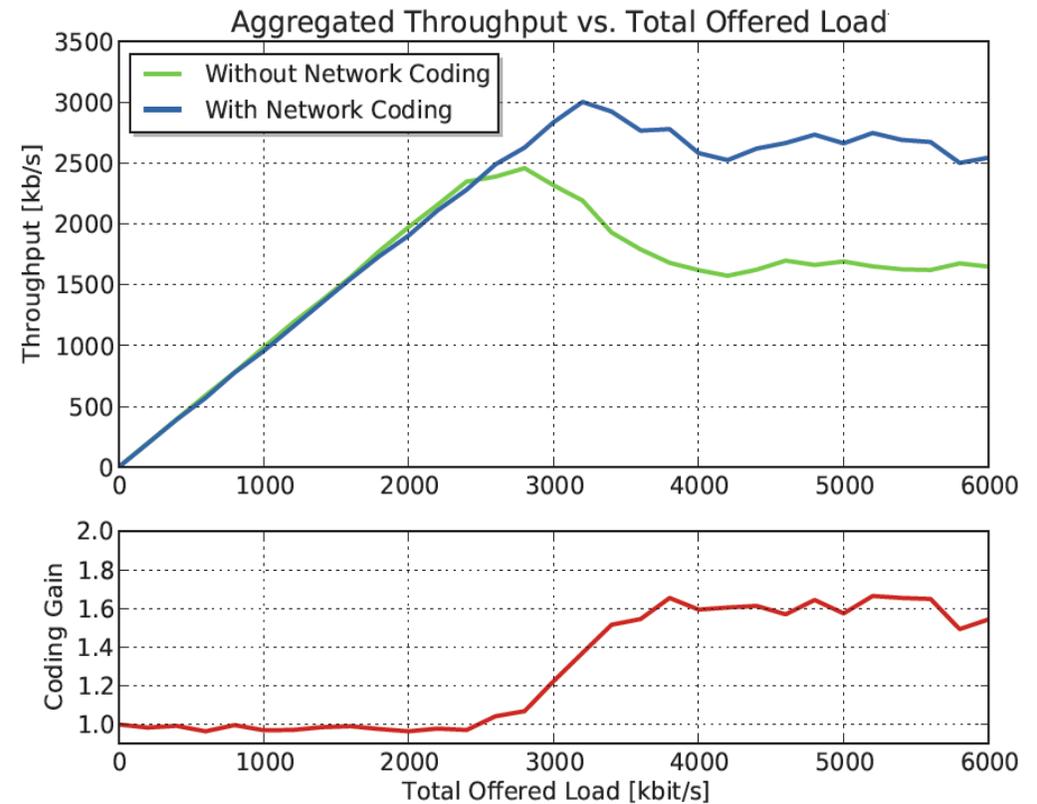
# CATWOMAN: Results



## Analytical results



## Measurement results



# Discussion

Results fit nicely

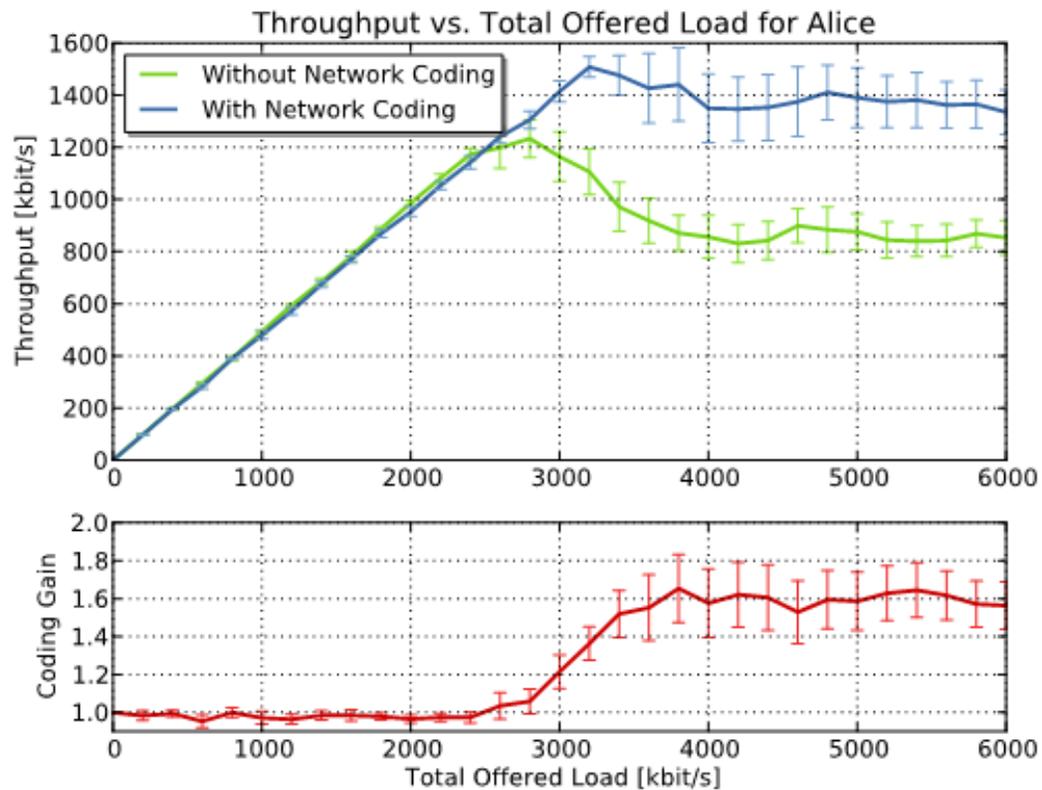
But

- The coding gain is with 1.6 lower than expected with 2.0
- The throughput of NC is not stable after reaching its maximum

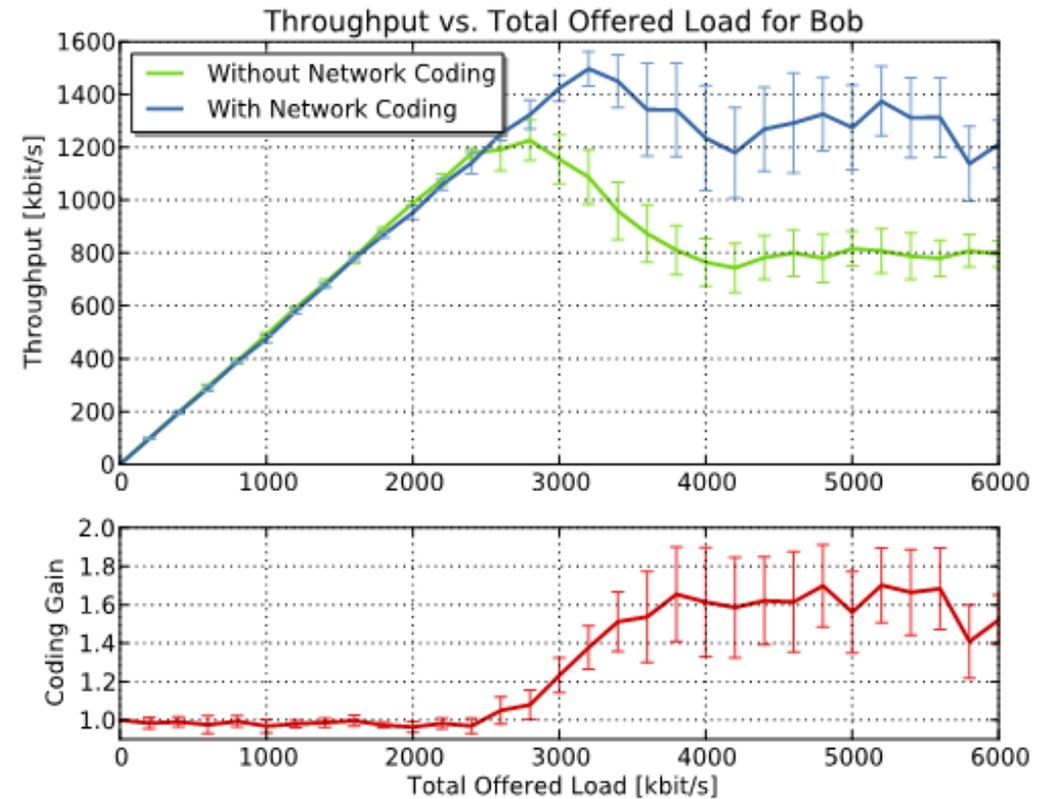
Why?

# CATWOMAN: Results

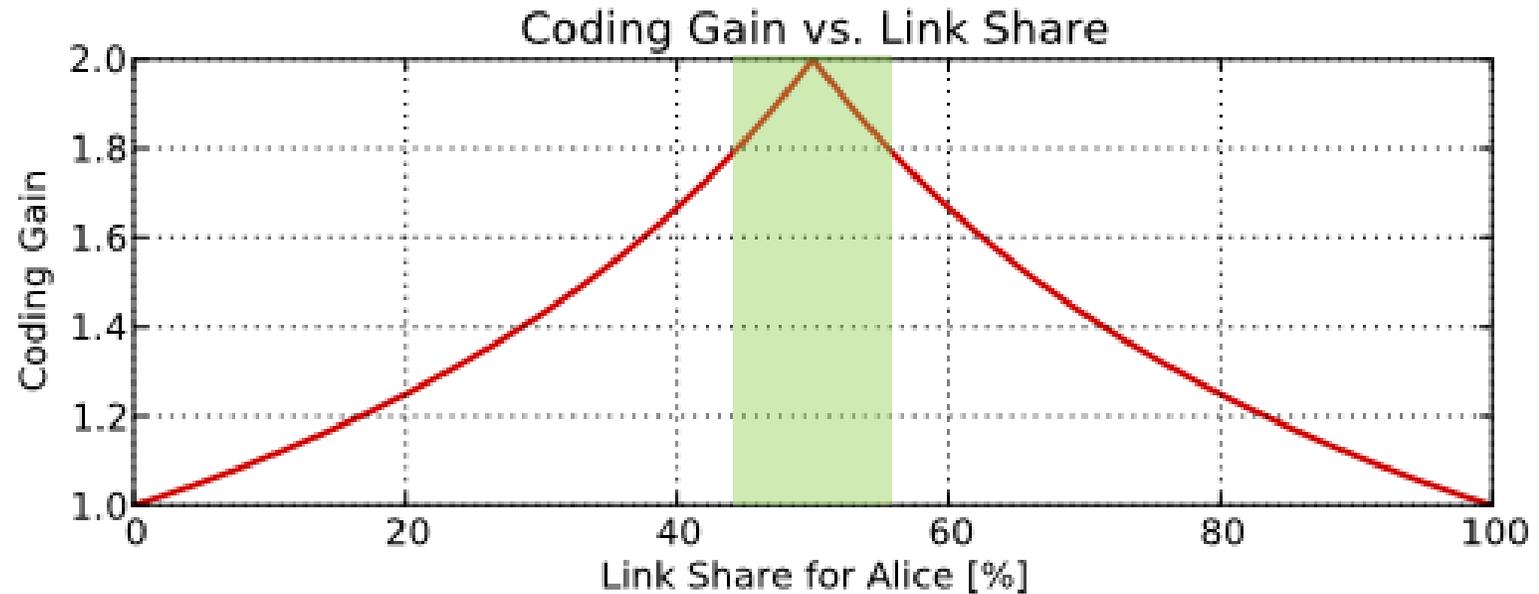
## Alice



## Bob

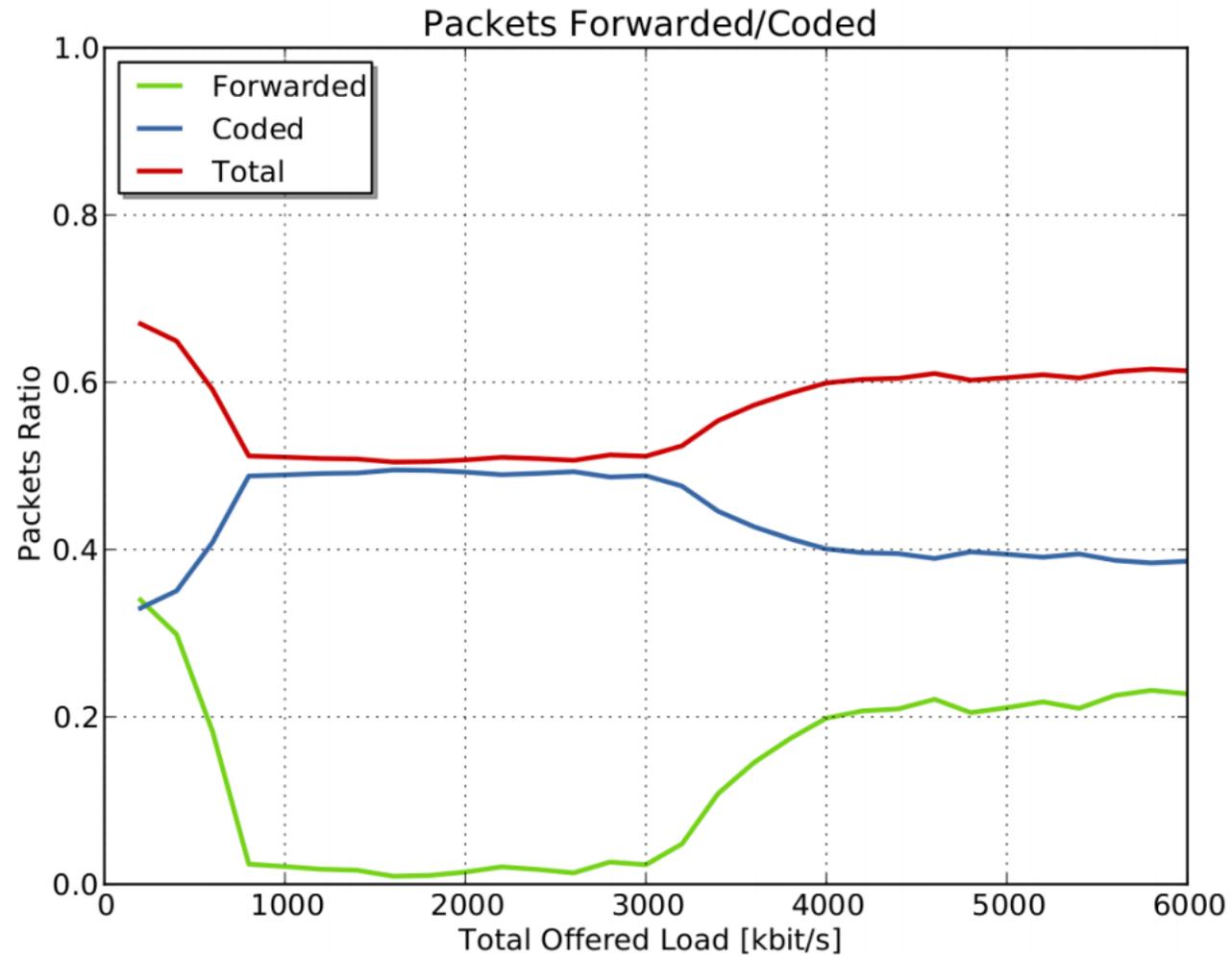


# CATWOMAN: Results

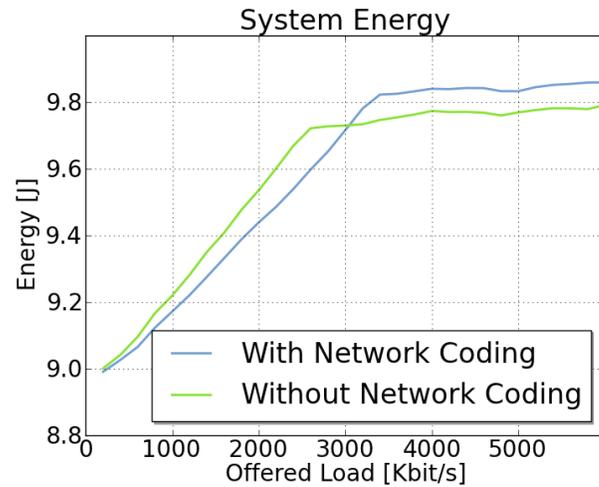


$$g_C = \frac{N}{N \cdot x + N((1-x) - x)} = \frac{1}{1-x}, \quad 0 \leq x \leq 0.5.$$

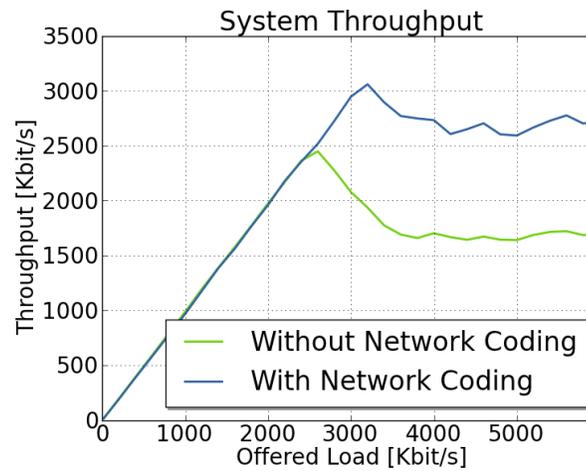
# CATWOMAN: Results



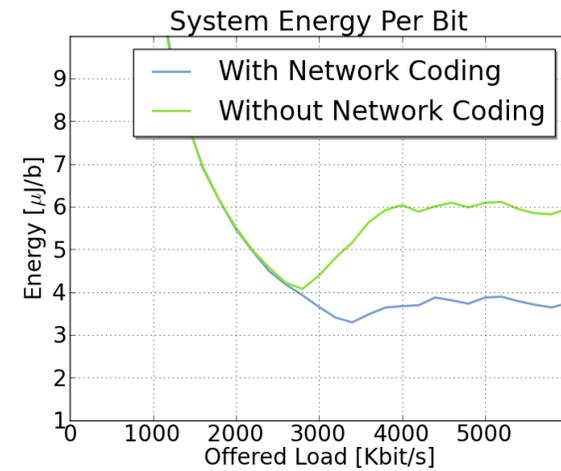
# First Measurement Result



power

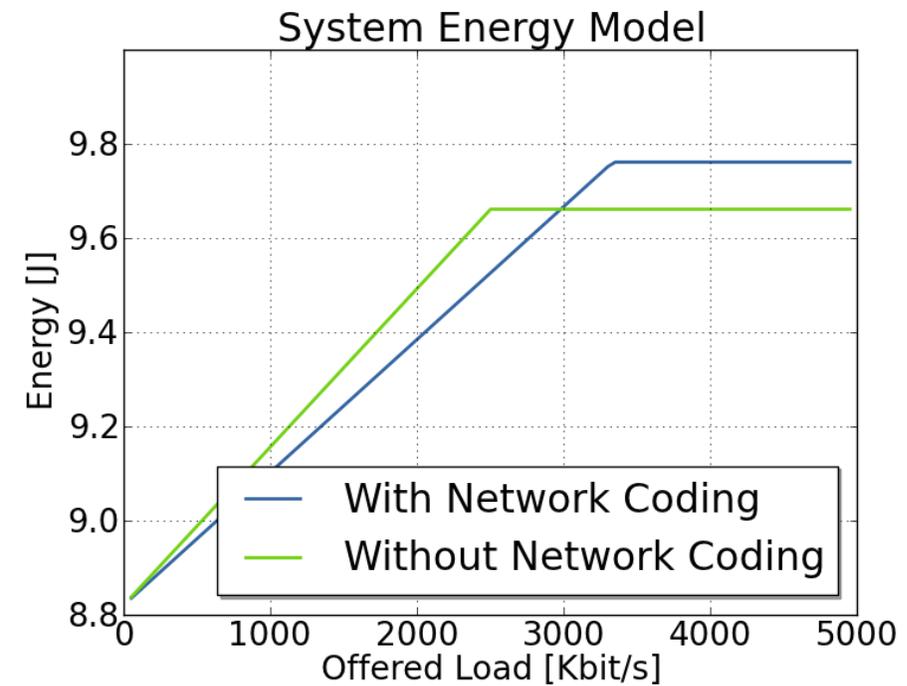
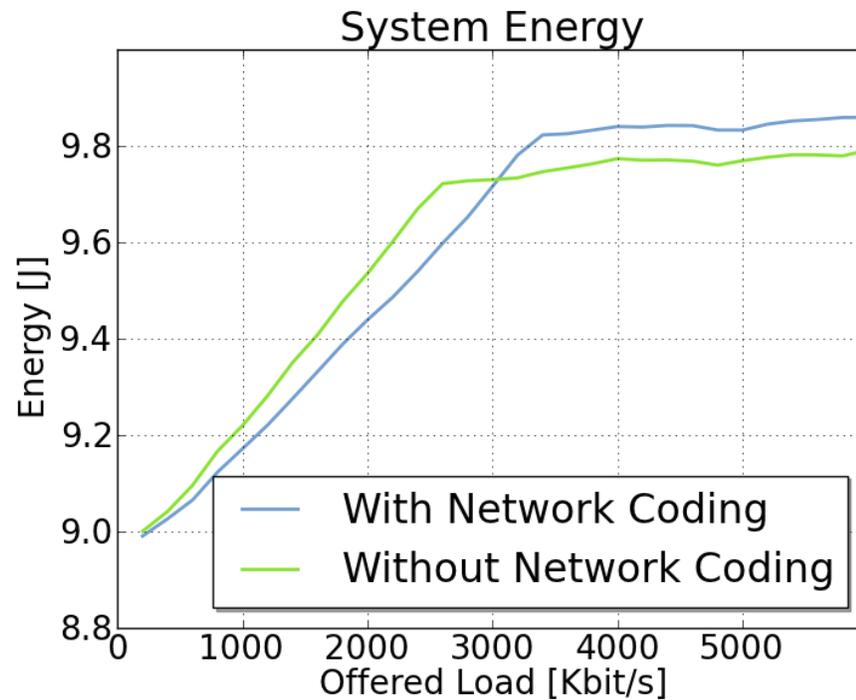


throughput

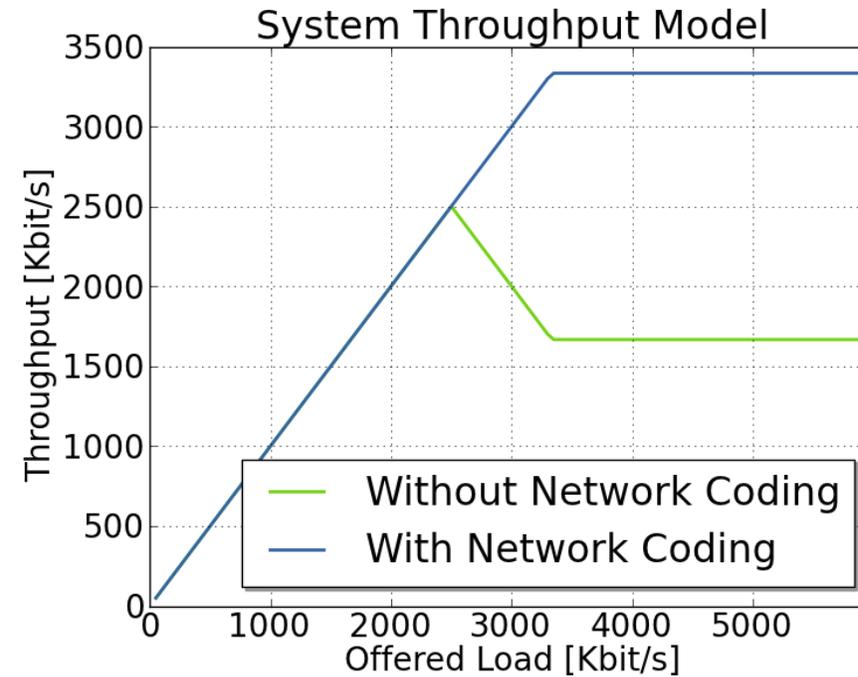
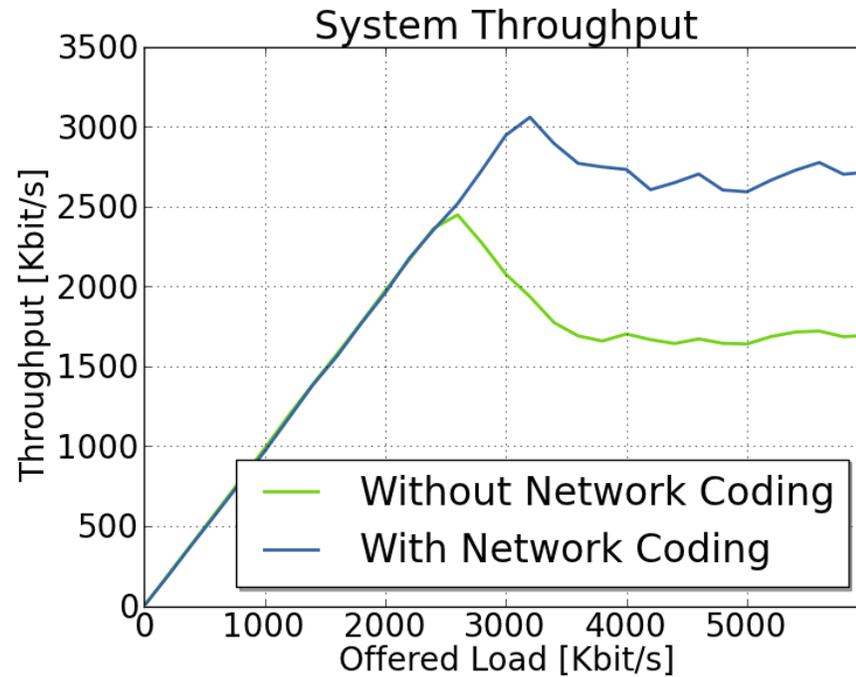


energy/bit

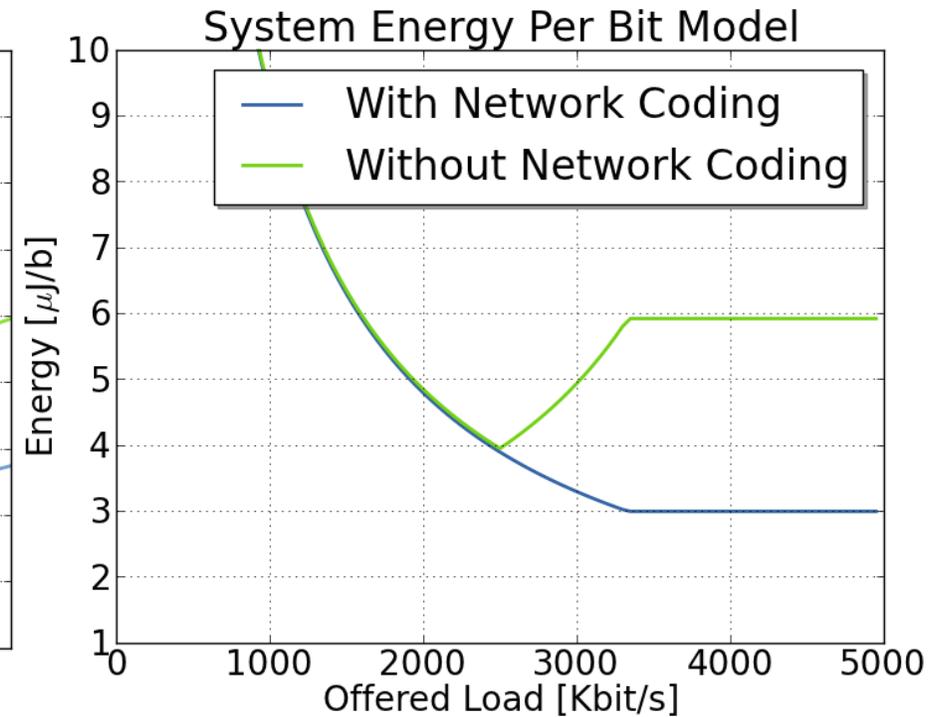
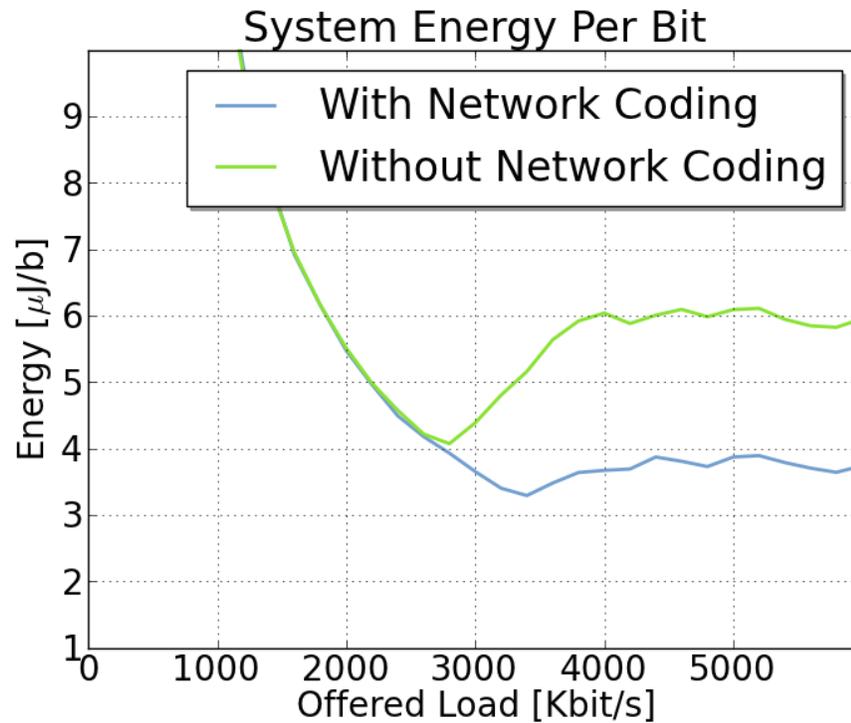
# Comparison: Energy



# Comparison: Throughput



# Comparison: Energy Per Bit



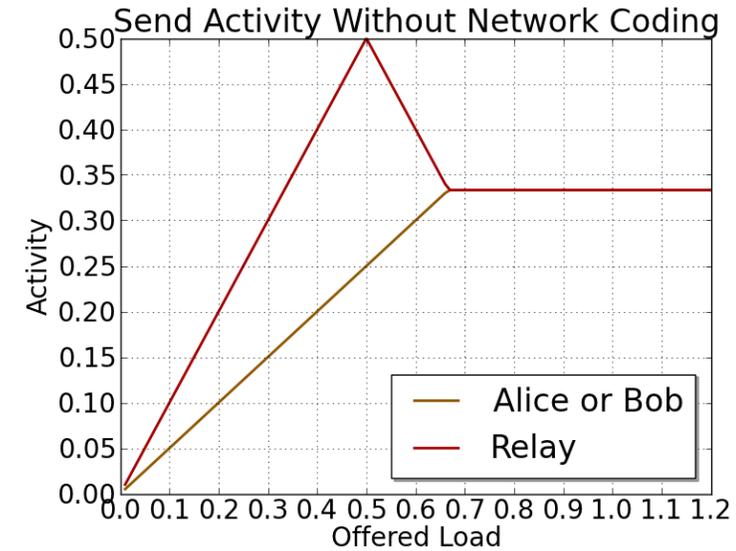
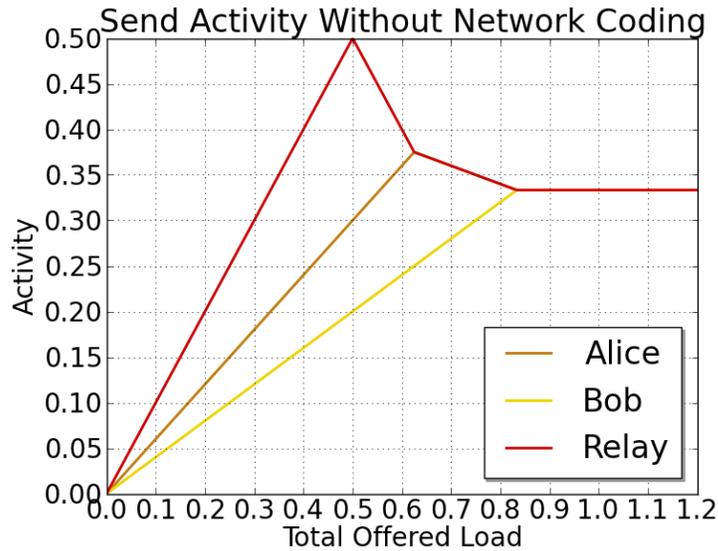
# Model for Alice and Bob (asymmetric traffic)

A. Paramanathan, J. Heide, P. Pahlavani, M. Hundeboll, S.A. Rein, F.H.P. Fitzek, and G. Ertli, **“Energy and data throughput for asymmetric inter-session network coding,”** in IEEE International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD), Barcelona, Spain.

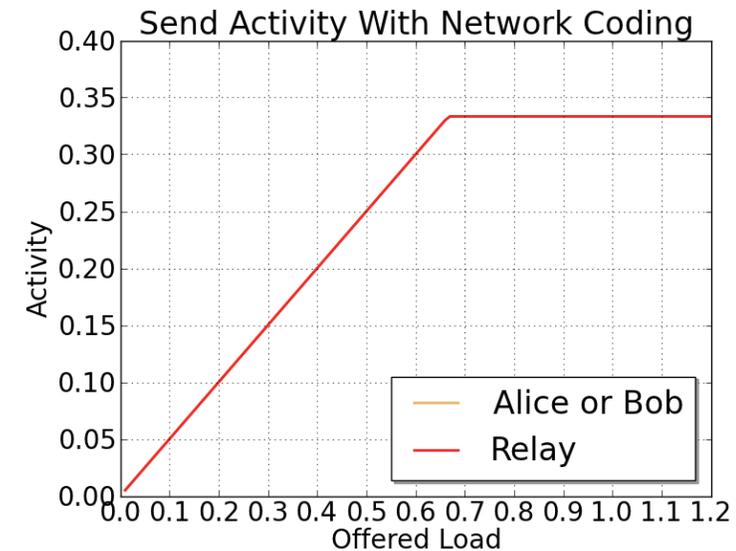
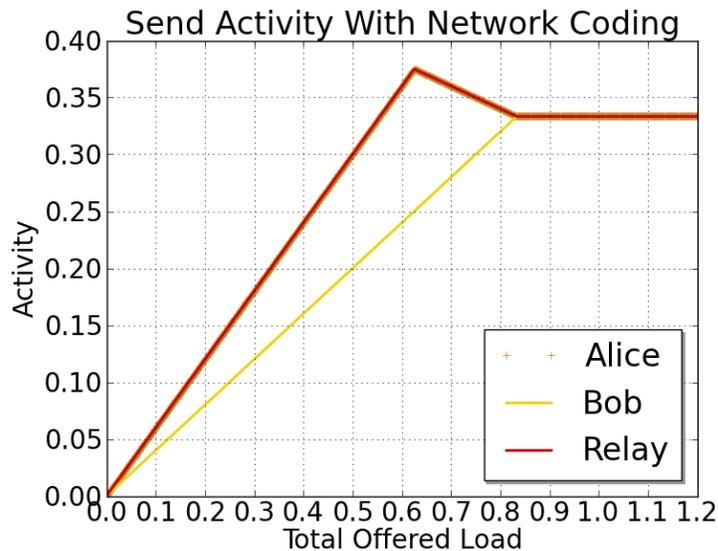


# Model for asymmetric traffic (vs. symmetric traffic): SEND

w/o NC



w NC

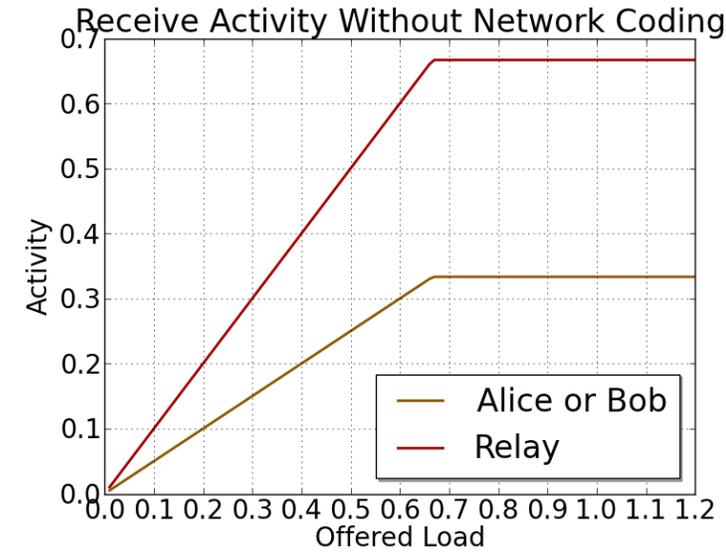
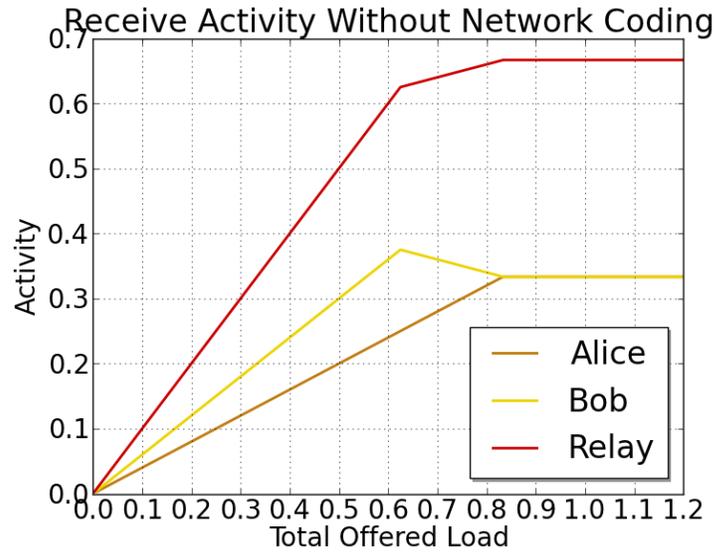


Asymmetric with link share of 60%(A)/40%(B)

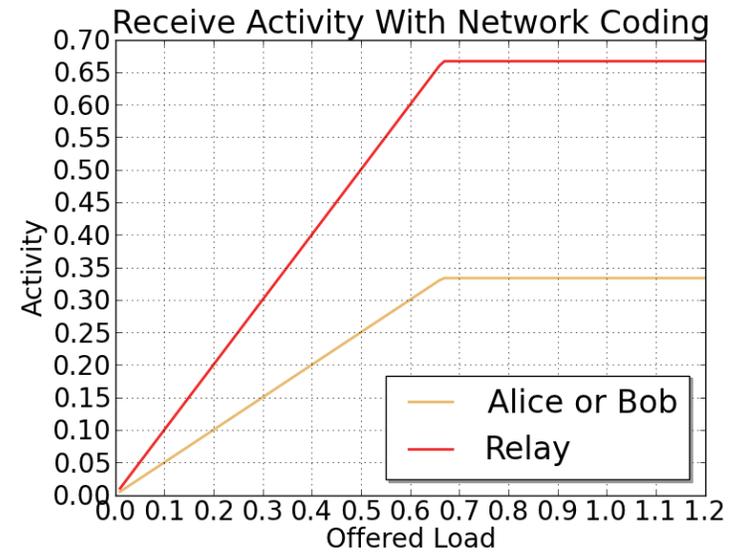
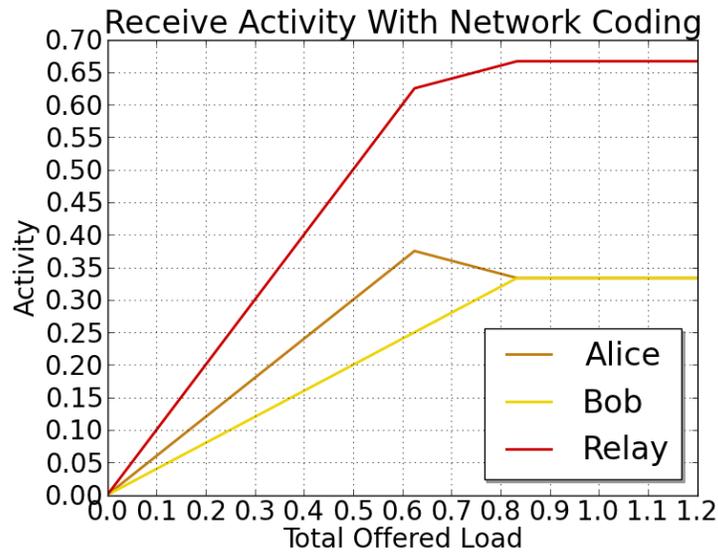


# Model for asymmetric traffic (vs. symmetric traffic): RECEIVE

w/o NC



w NC

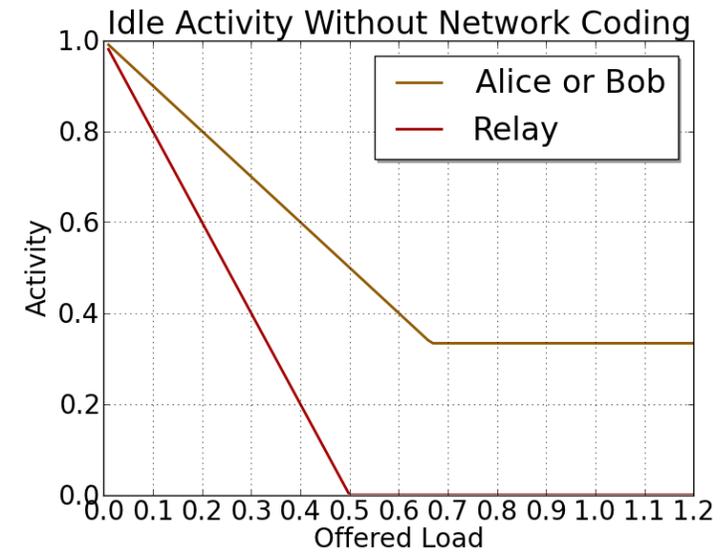
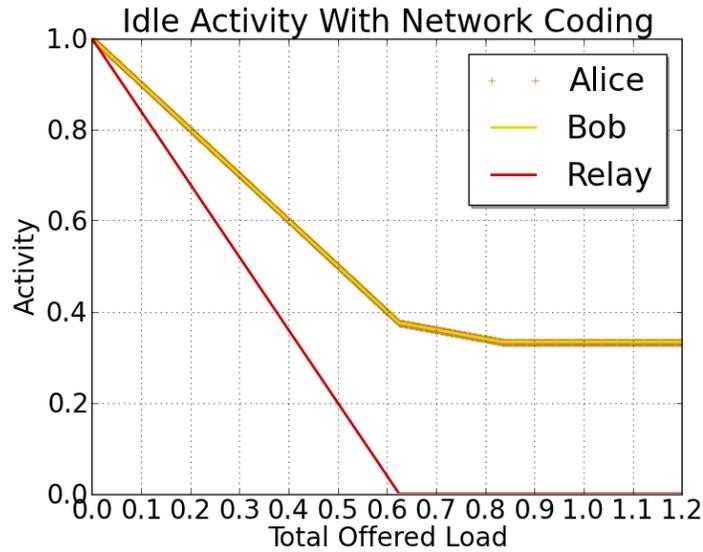


Asymmetric with link share of 60%(A)/40%(B)

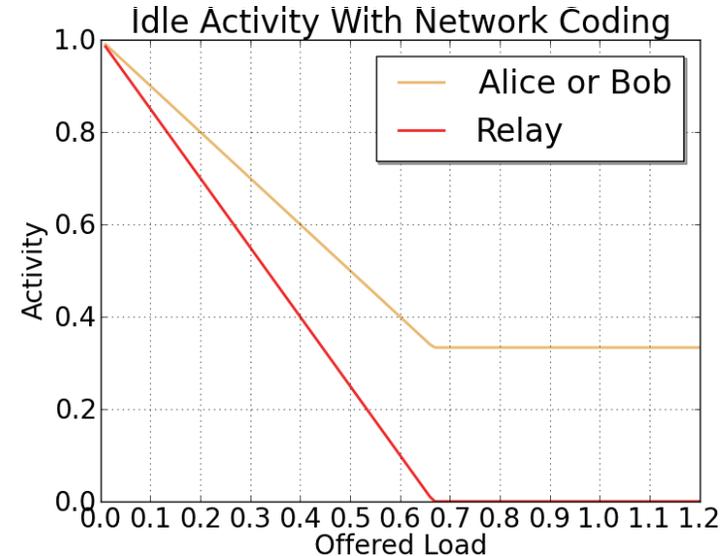
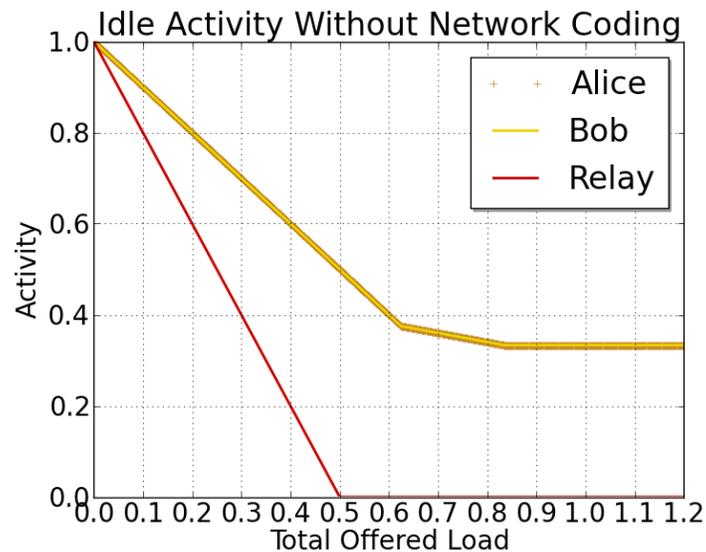


# Model for asymmetric traffic (vs. symmetric traffic): IDLE

w/o NC



w NC

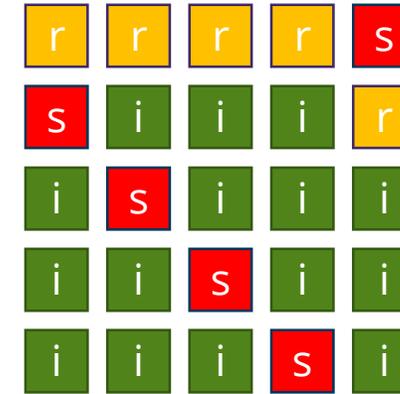
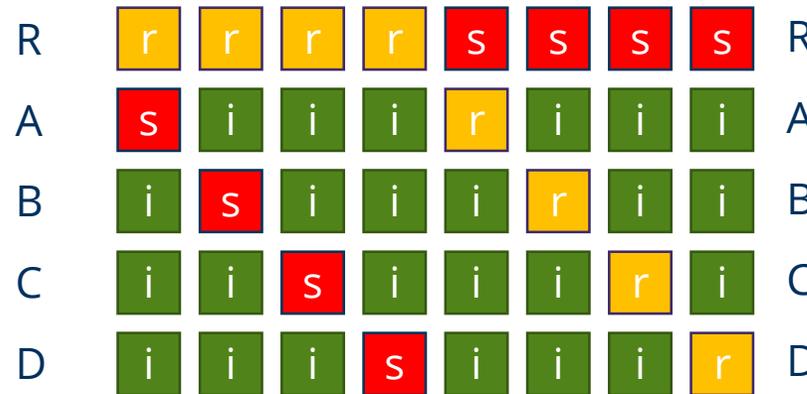
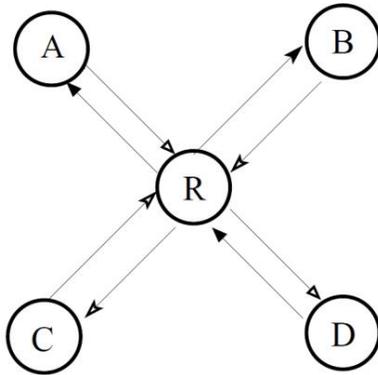


Asymmetric with link share of 60%(A)/40%(B)

# Model for THE CROSS (symmetric traffic)

# Cross Forwarding

pure relaying

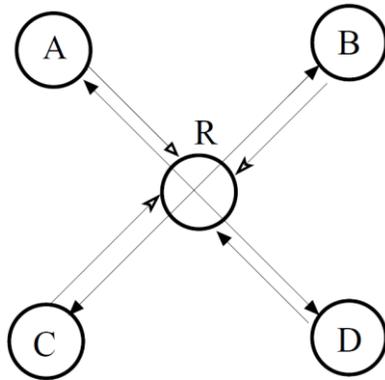


Whatever goes into the relay has to be forwarded.

What to do if there is not enough capacity?

# XOR Network Coding

NC w/o overhearing



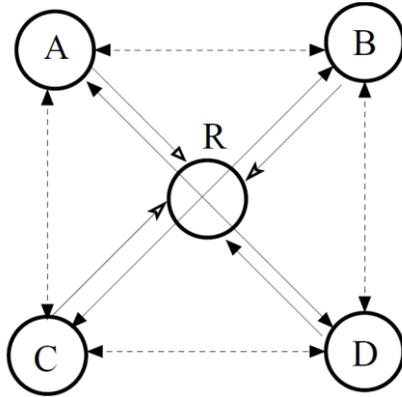
R	r	r	r	r	s	s
A	s	i	i	i	r	i
B	i	s	i	i	i	r
C	i	i	s	i	i	r
D	i	i	i	s	r	i

R	r	r	r	r	s
A	s	i	i	i	r
B	i	s	i	i	i
C	i	i	s	i	i
D	i	i	i	s	r

- Each out node sends to the relay
- And for each pair the relay sends out one coded packet

# XOR Network Coding with overhearing

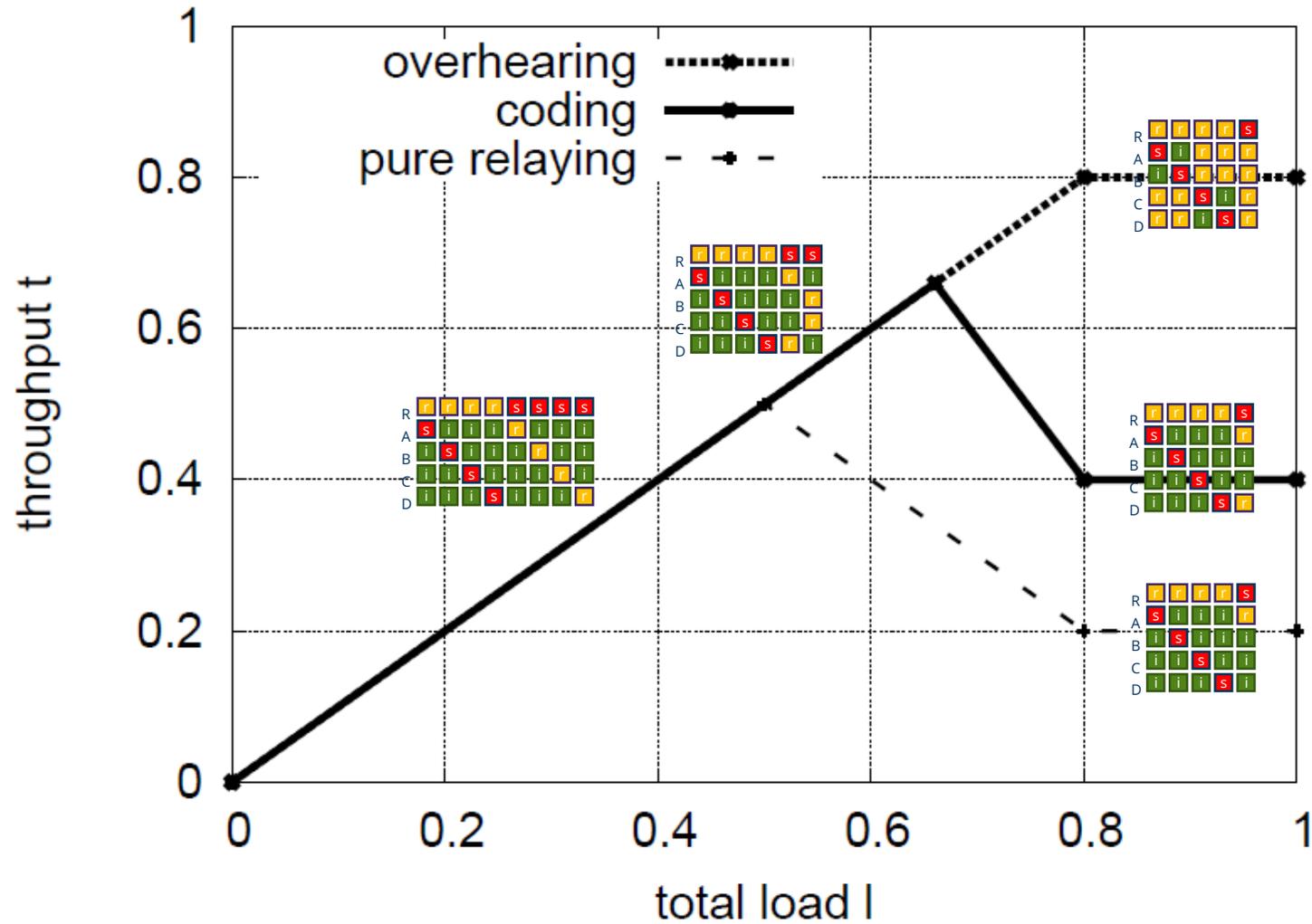
NC with overhearing



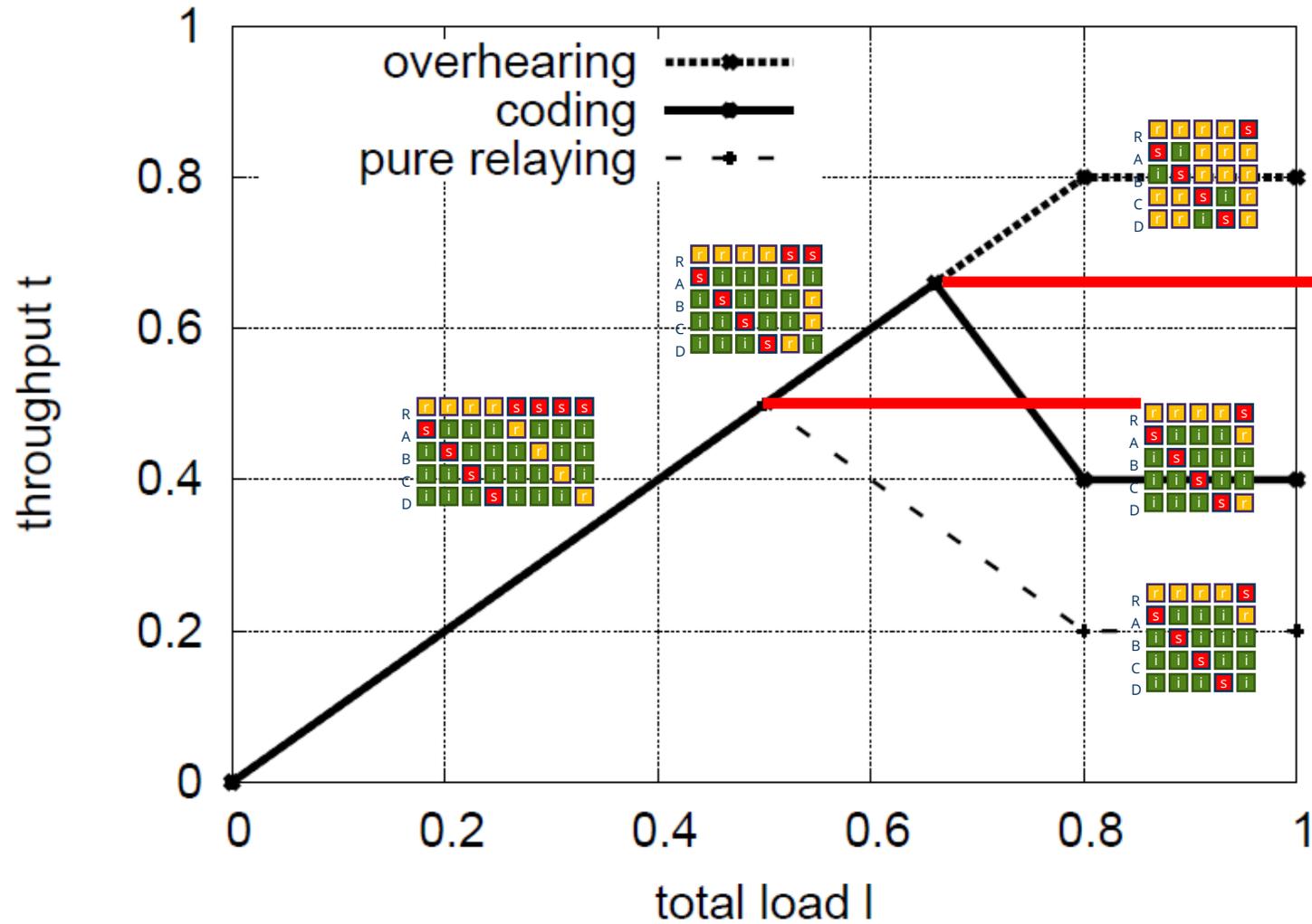
R	r	r	r	r	s
A	s	r	r	i	r
B	r	s	i	r	r
C	r	i	s	r	r
D	i	r	r	s	r

- Each outer node sends a packet to the relay
- Each outer node will overhear two packets from neighboring nodes
- Relay sends out one full coded packet

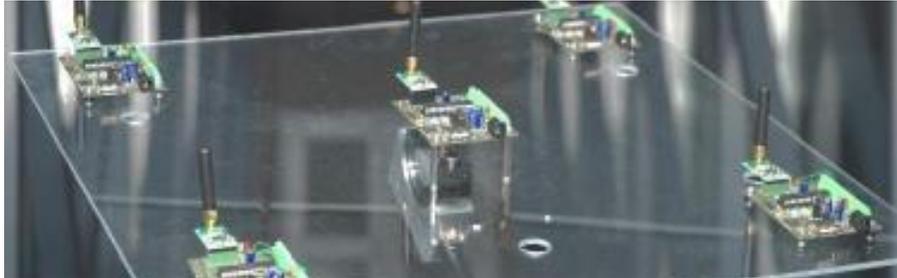
# Cross Throughput



# Cross Throughput



# The Cross (MAC)



## Hardware

16bit PIC24 microprocessor

nRF905 transceiver (433 MHz, 50 kbps)

## Software

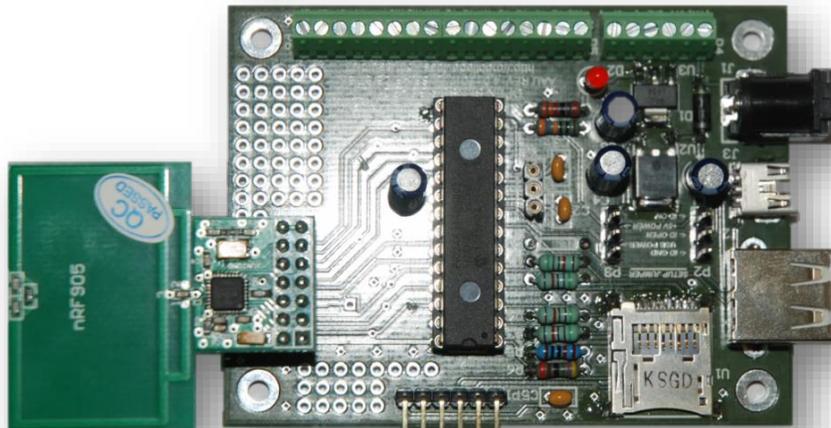
MAC-Protocol: A CSMA/CA design

Network Coding: A simple XOR design [COPE06]

## Capability

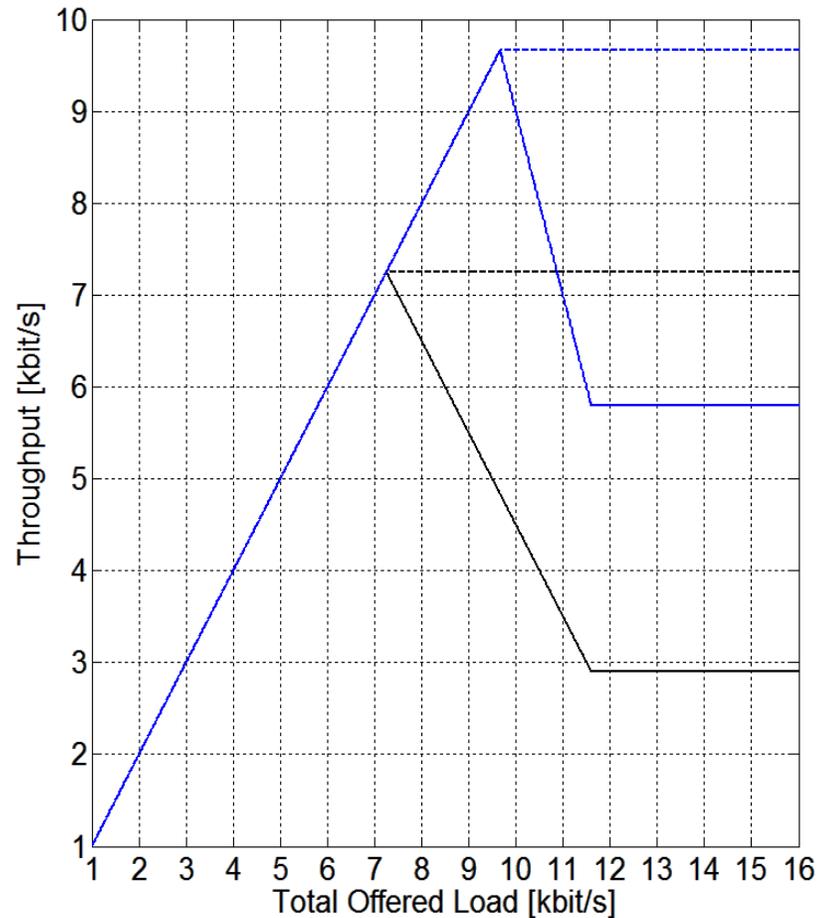
Easy access to the software

Full control of both HW and SW

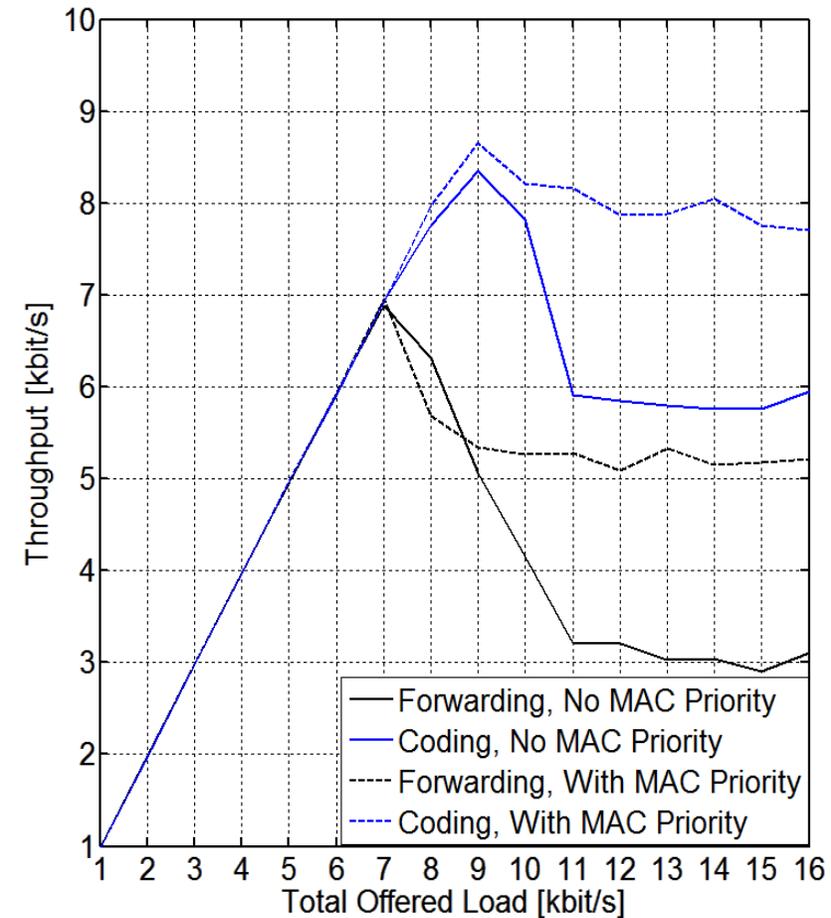


# The Cross (MAC)

## Expected



## Measurement

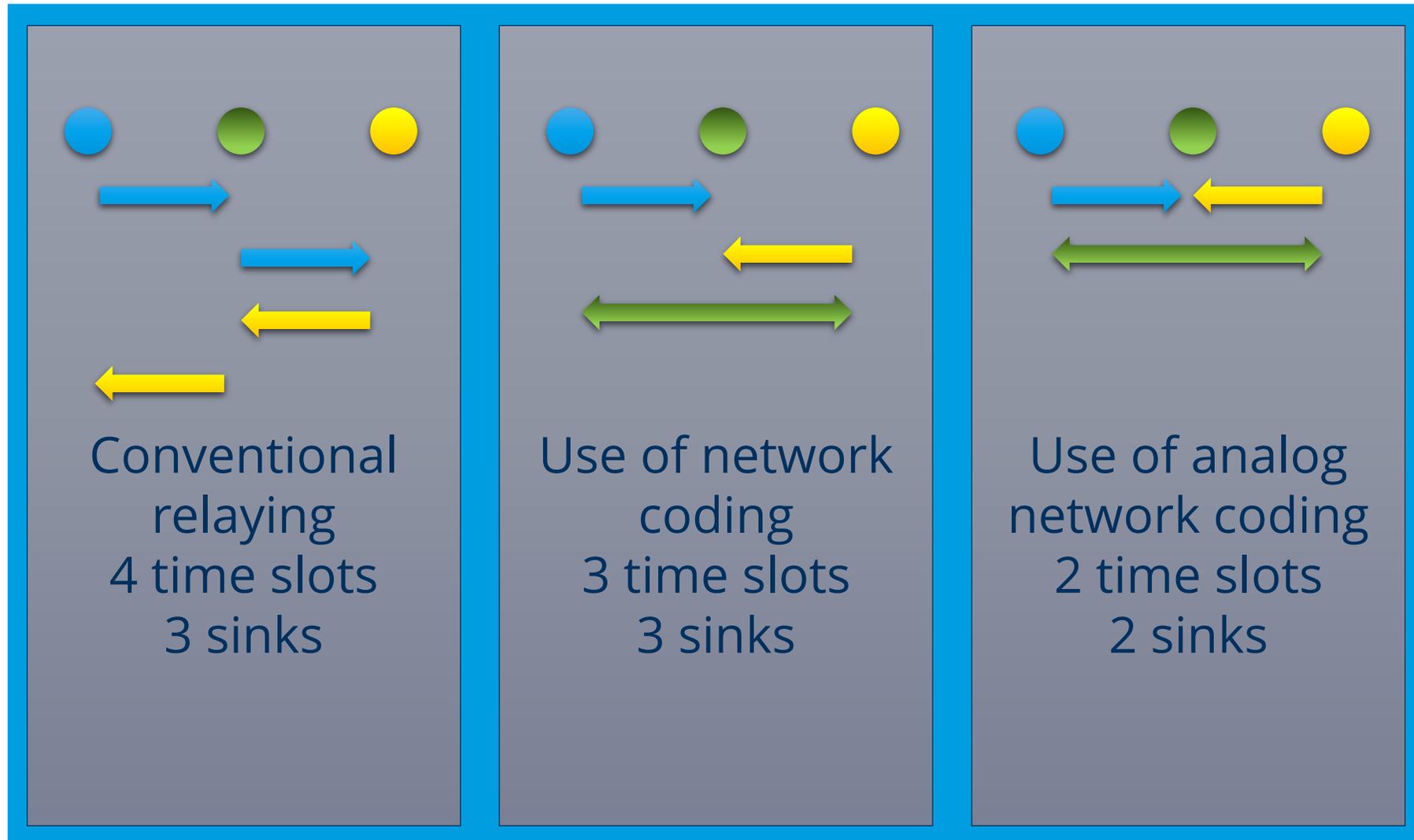


# Analog Network Coding

# Overview

- So far network coding was done in the packet domain
- Network coding can be applied in any ISO/OSI layer
- Analog and physical network coding is a special case at the physical layer (lowest ISO/OSI layer) coding “symbols”
- Coding symbols is nothing else than a superposition of signals. Analog network coding breaks with the paradigm to separate signals in time and force “collision” to achieve higher coding gains

# Analog Network Coding for Wireless Networks



# Analog Network Coding for Wireless Networks

Analog network coding seems to be more efficient than digital network coding

For the two way relay it reduces the number of necessary transmissions to two (three for digital network coding and four for store and forward)

## Advantages

- Throughput
- Energy
- Security (role of the relay differs)

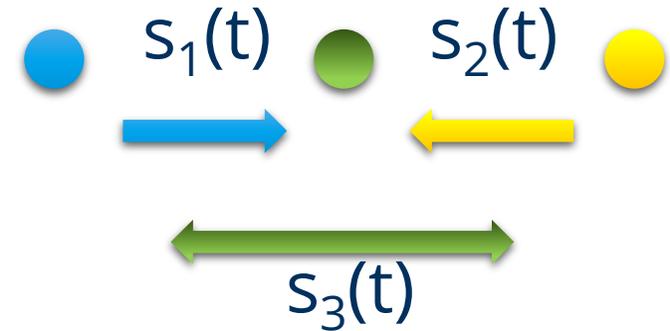
# Physical Layer Network Coding



Presented by [Zhang et al 2006]

First, simple example: no fading

Let us look at bandpass signals



$$\begin{aligned} r_3(t) &= s_1(t) + s_2(t) + n(t) \\ &= [a_1 \cos(\omega t) + b_1 \sin(\omega t)] + [a_2 \cos(\omega t) + b_2 \sin(\omega t)] + n(t) \\ &= (a_1 + a_2) \cos(\omega t) + (b_1 + b_2) \sin(\omega t) + n(t) \end{aligned}$$

How to generate  $s_3(t)$  ?

# Physical Layer Network Coding

How to generate  $s_3(t)$  ?

Amplify and forward?

Decode and forward?

Initial approach: decode and forward

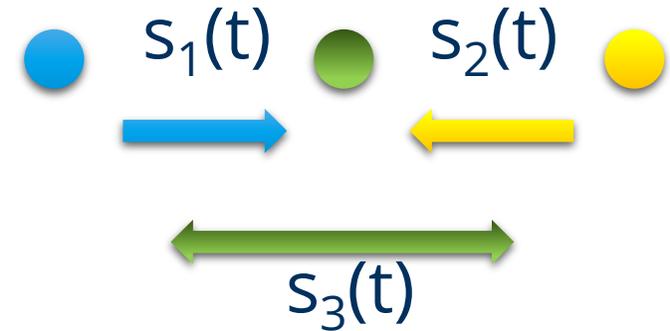
**Example with BPSK:** say  $b_i = 0$ ,  $a_i \in \{-1, 1\}$   $i = 1, 2, 3$

$$r_3(t) = s_1(t) + s_2(t) + n(t) = (a_1 + a_2) \cos(\omega t) + n(t)$$

Note that there are 3 possible values of  $a_1 + a_2$ :

"-2" and "2" correspond to  $a_1 = a_2$

"0" corresponds to  $a_1 = -a_2$



# Physical Layer Network Coding

## Example with BPSK:

Let us generate  $s_3(t)$

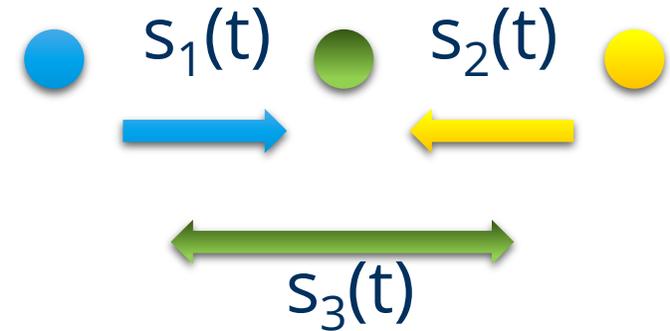
(hint: XOR-like operation)

If  $a_1 = a_2$  then  $a_3 = 1$  (logical "0")

If  $a_1 = -a_2$  then  $a_3 = -1$  (logical "1")

Alice and Bob receive as standard BPSK modulation

Then, *XOR bit by bit* with the sent packet

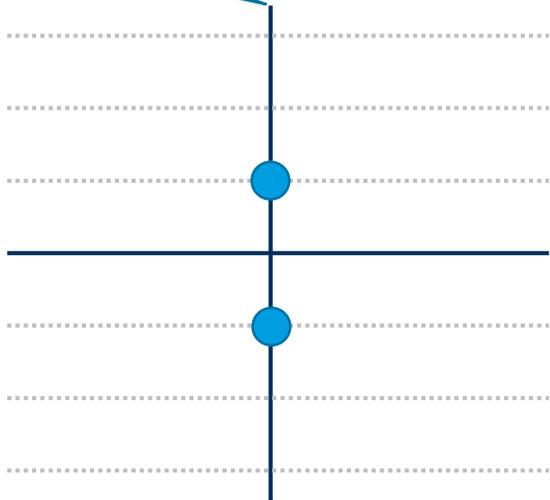


# Physical Layer Network Coding

What if we A&F?

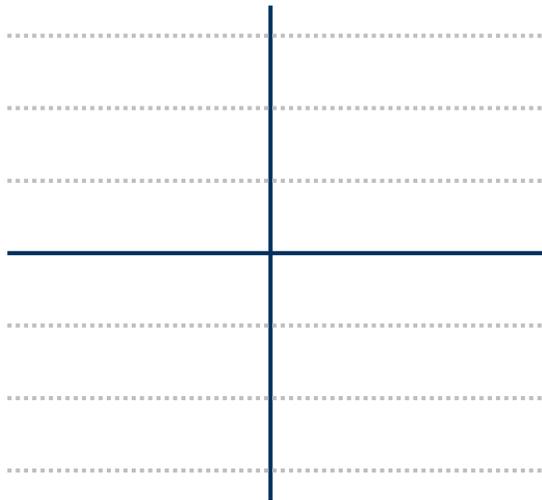
1st step – coding in the air

Alice may send two symbols

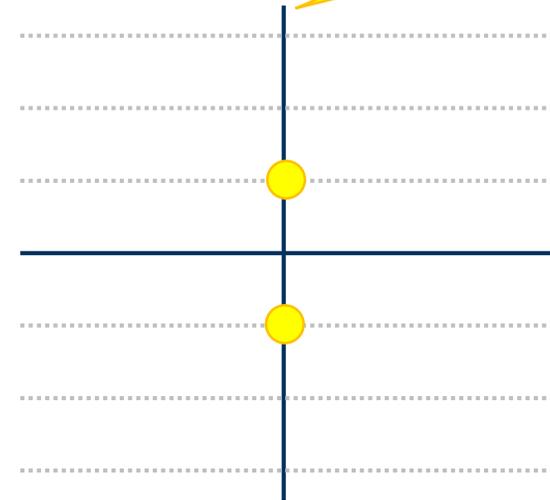


Alice

Bob may send two symbols



Relay



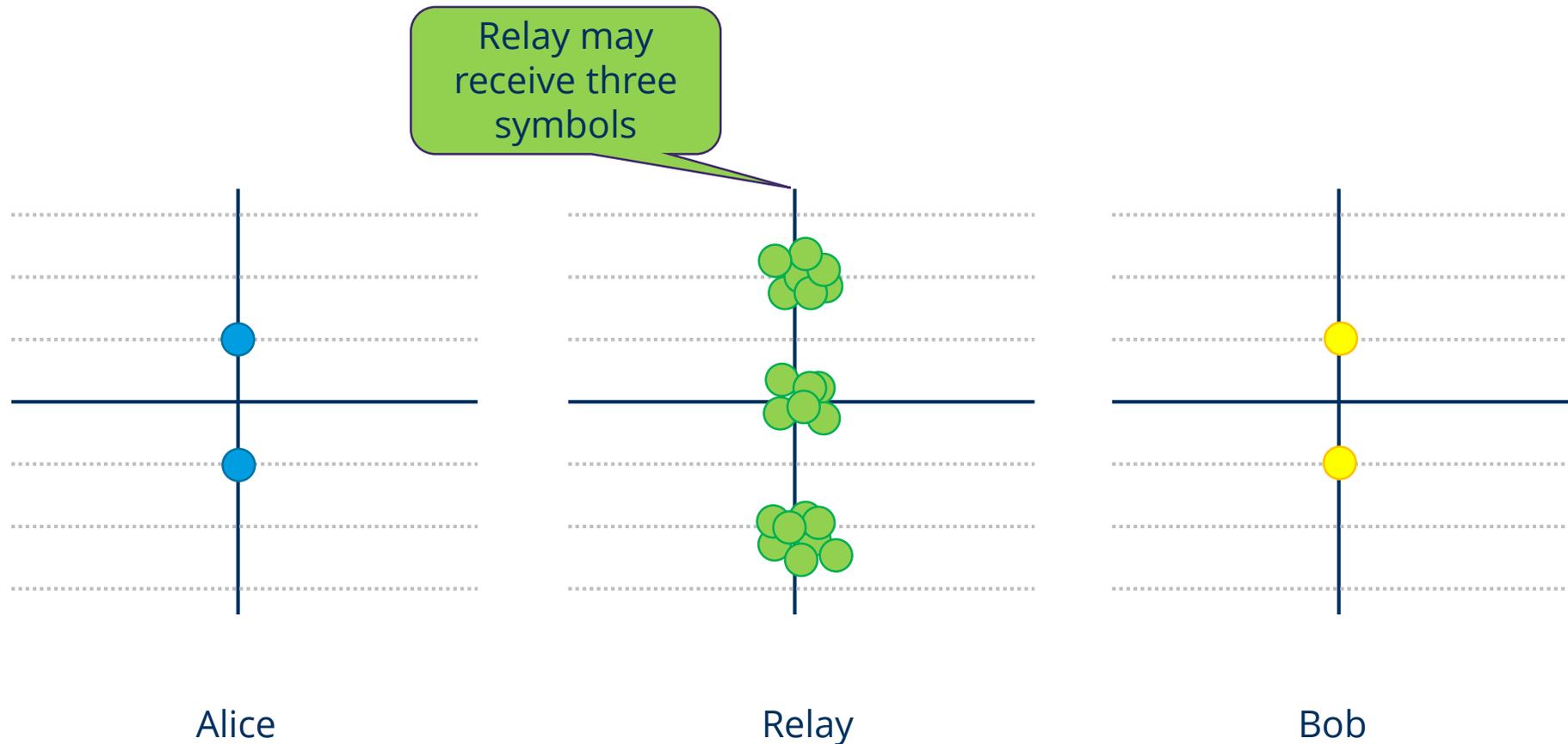
Bob

BPSK Example

# Physical Layer Network Coding

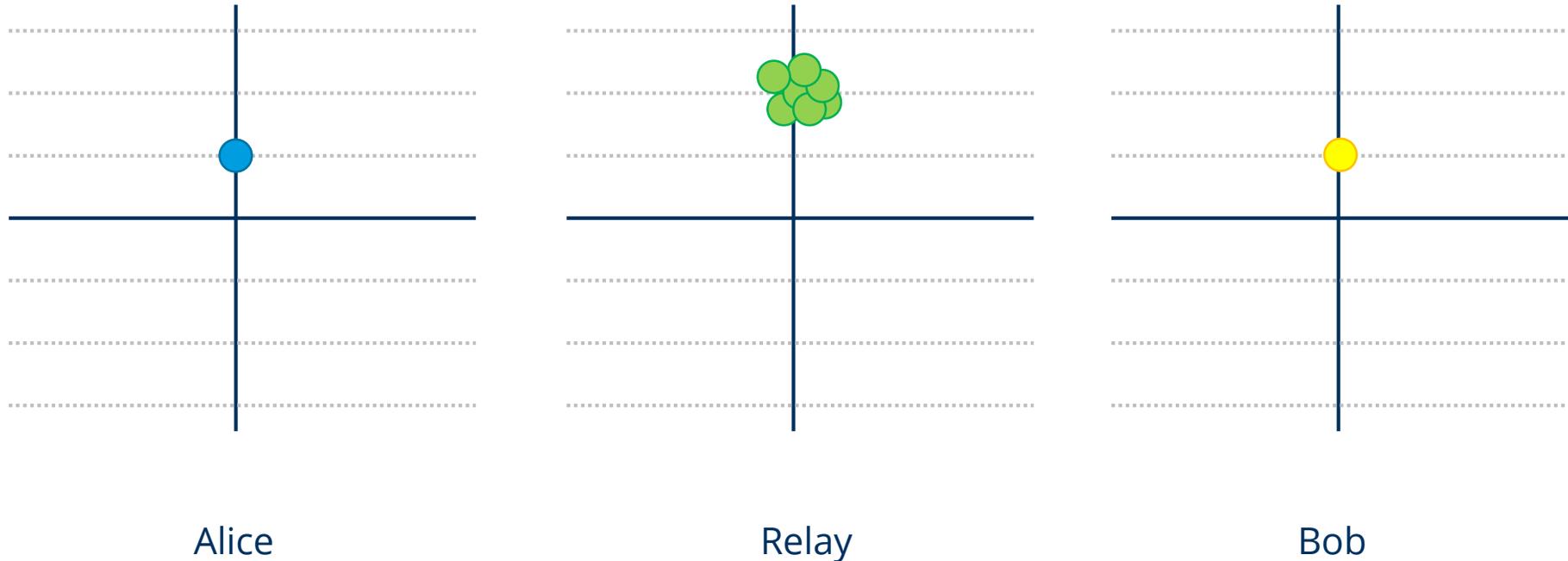
What if we A&F?

1st step – coding in the air



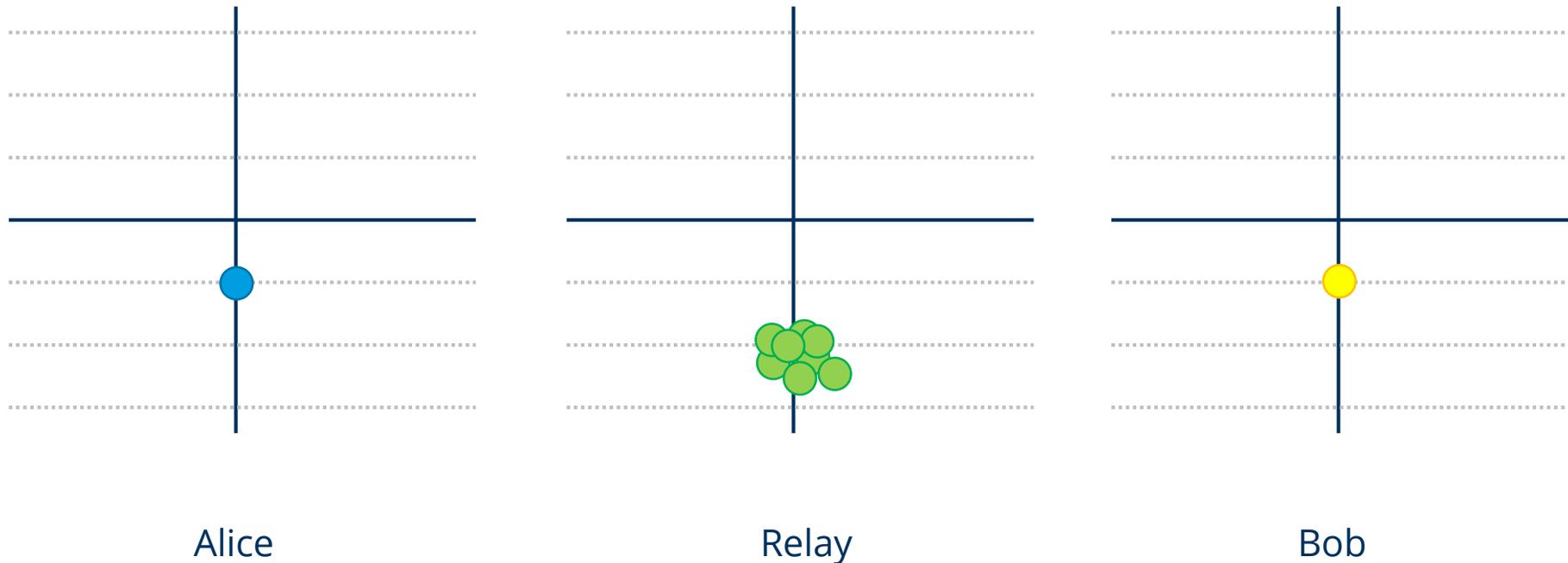
# Physical Layer Network Coding

1st step – coding in the air – e.g. 1/1



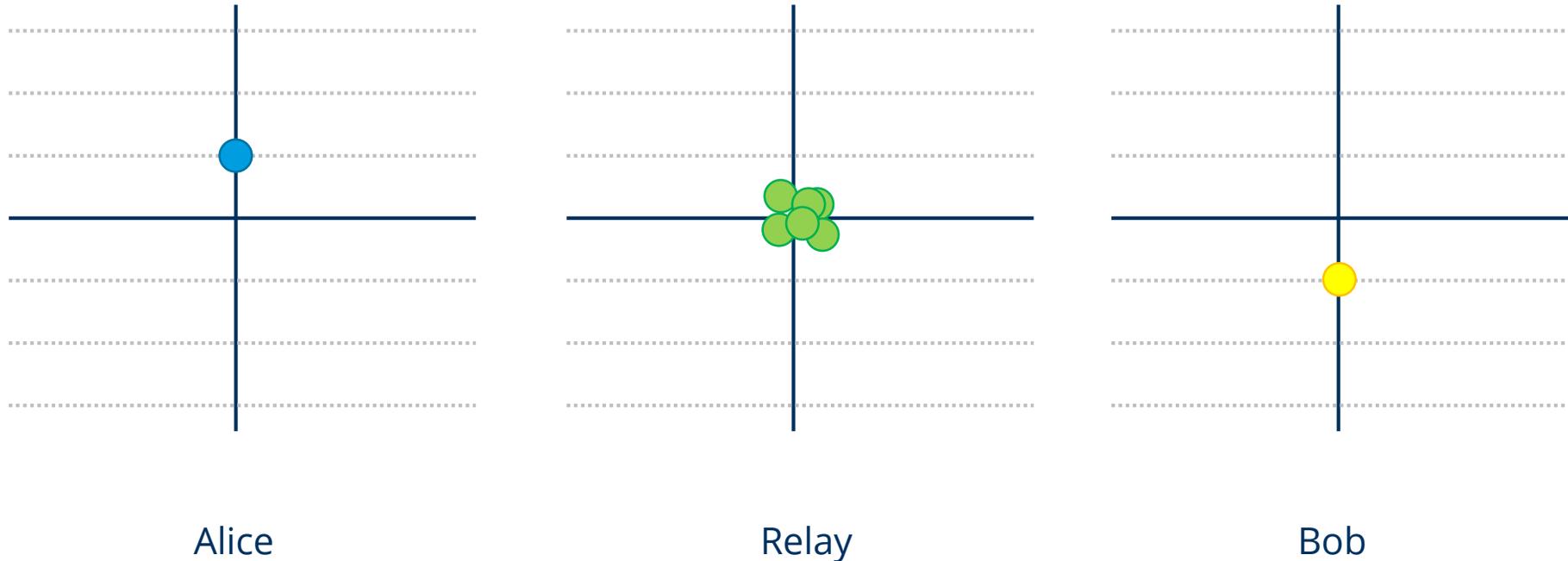
# Physical Layer Network Coding

1st step – coding in the air – e.g. 0/0



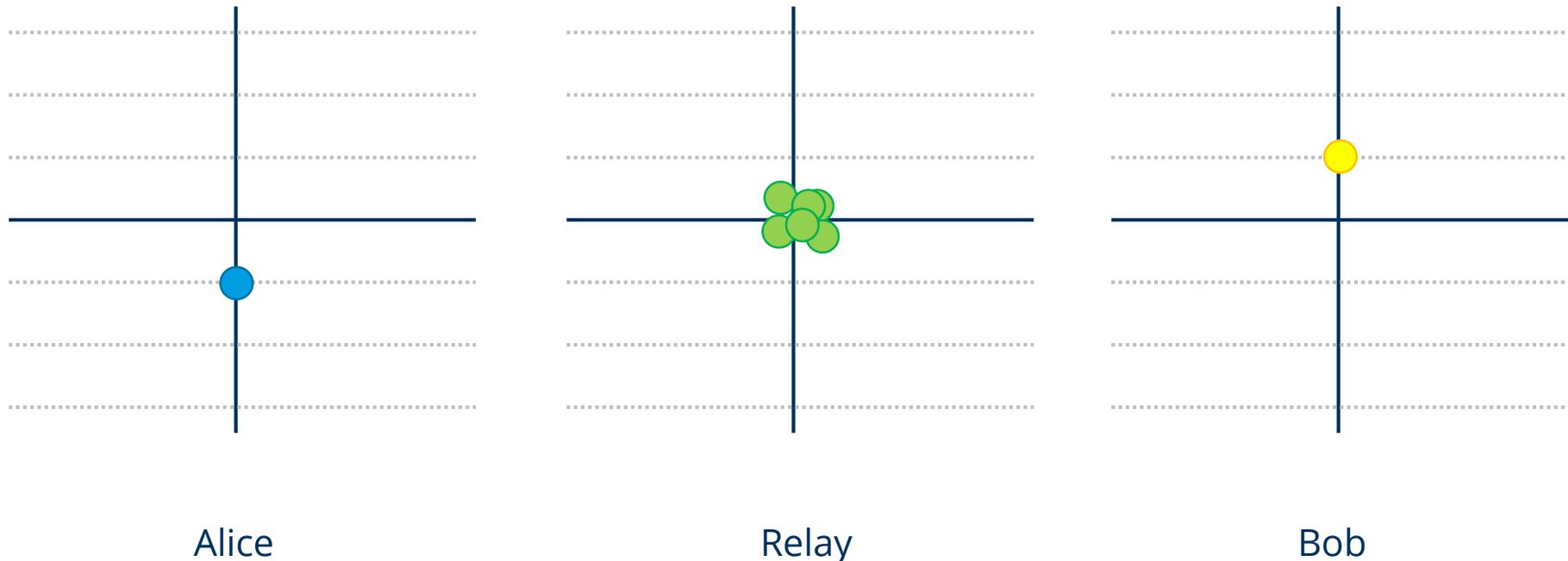
# Physical Layer Network Coding

1st step – coding in the air – e.g. 1/0



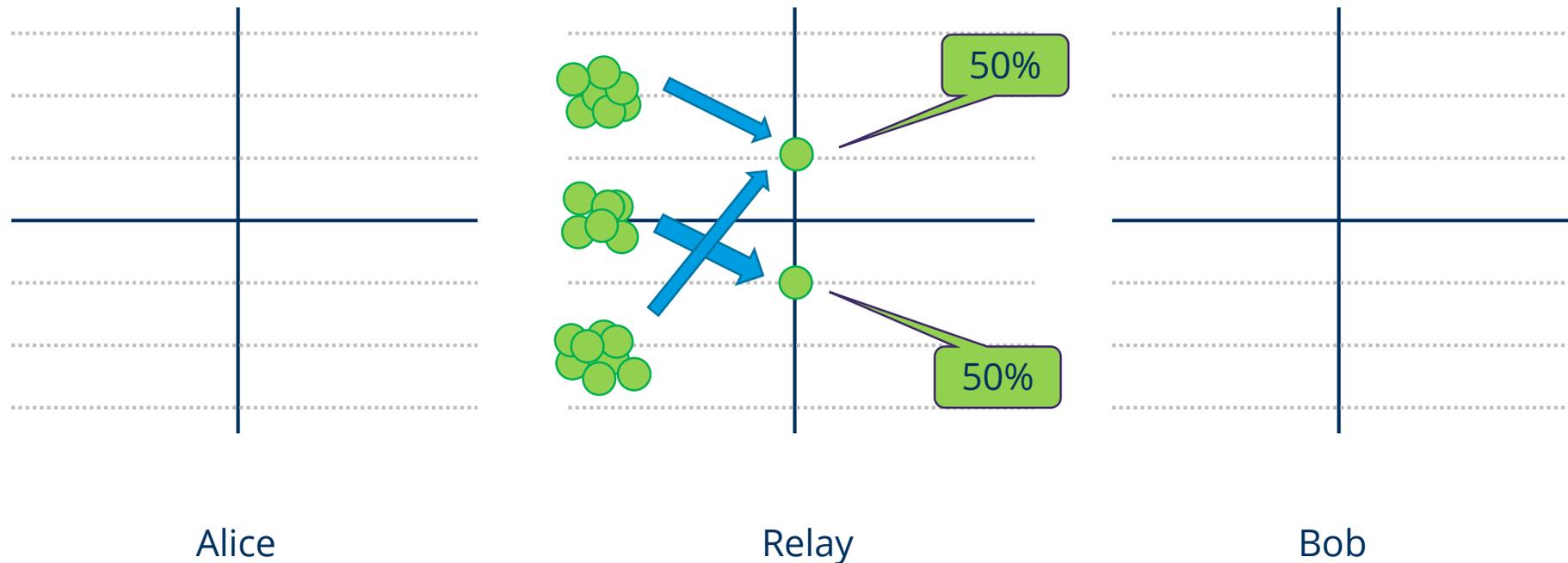
# Physical Layer Network Coding

1st step – coding in the air – e.g. 0/1



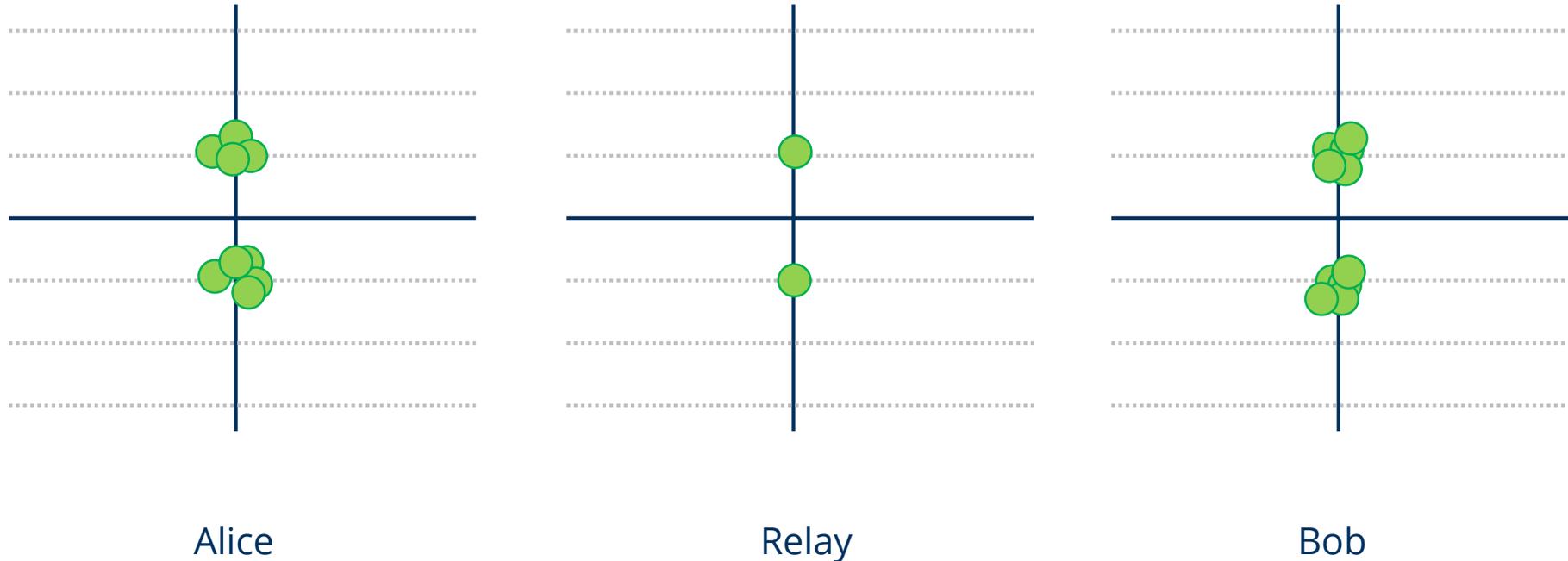
# Physical Layer Network Coding

2nd step - relay



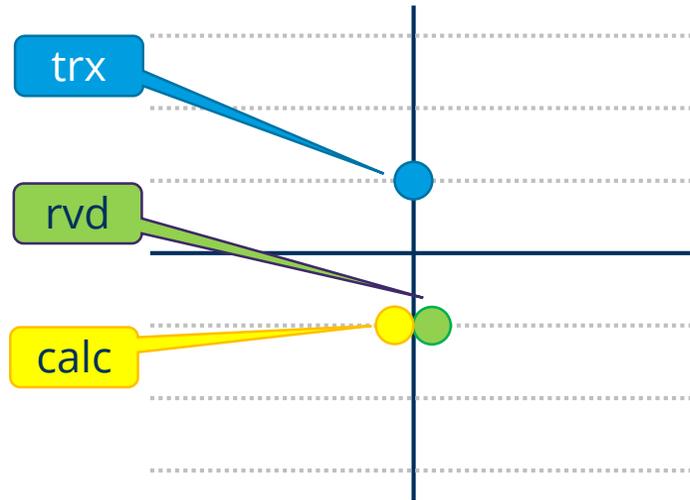
# Physical Layer Network Coding

2nd step - relay

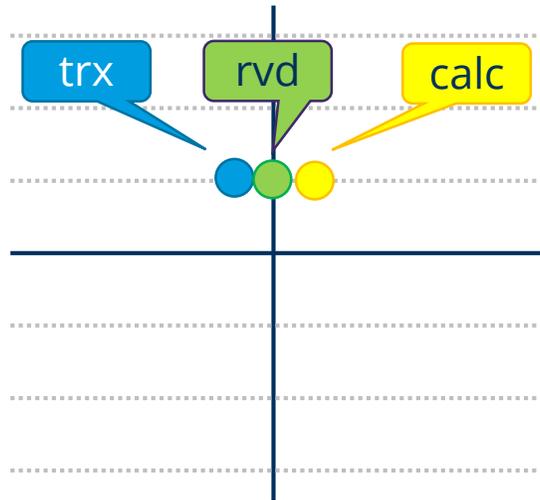


# Physical Layer Network Coding

## 3rd step decoding



Alice



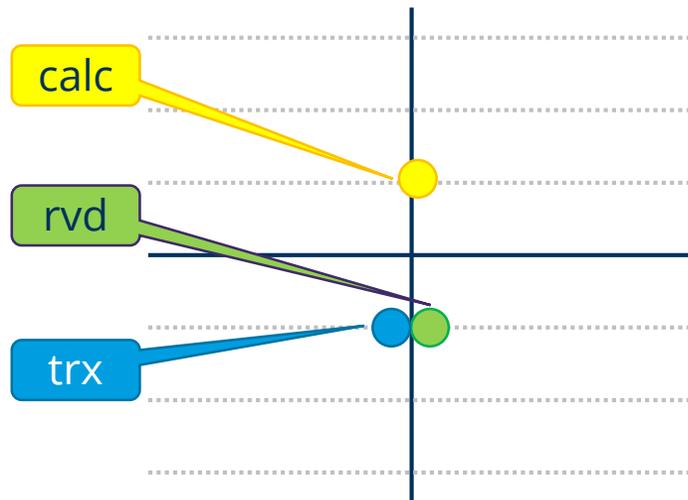
Alice



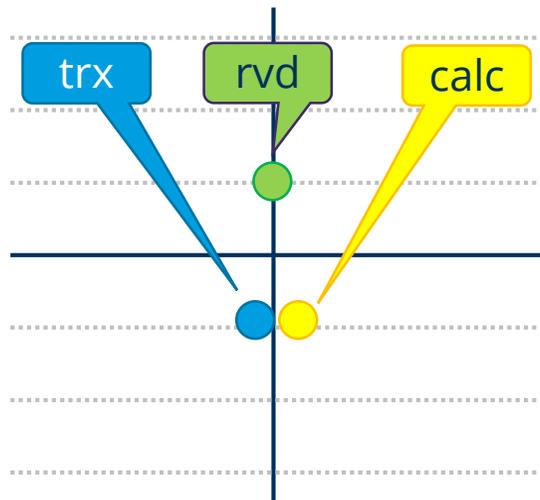
Alice

# Physical Layer Network Coding

## 3rd step decoding



Alice



Alice



Alice

# Analog Network Coding

What were our assumptions so far?

- No fading → there is amplitude + phase distortion
- Perfect sync
- Perfect detection of a collision
- Perfect knowledge of packet used for decoding at Alice and Bob
- The "right" packets interfere (MAC / Network impact)

How to make it practical?

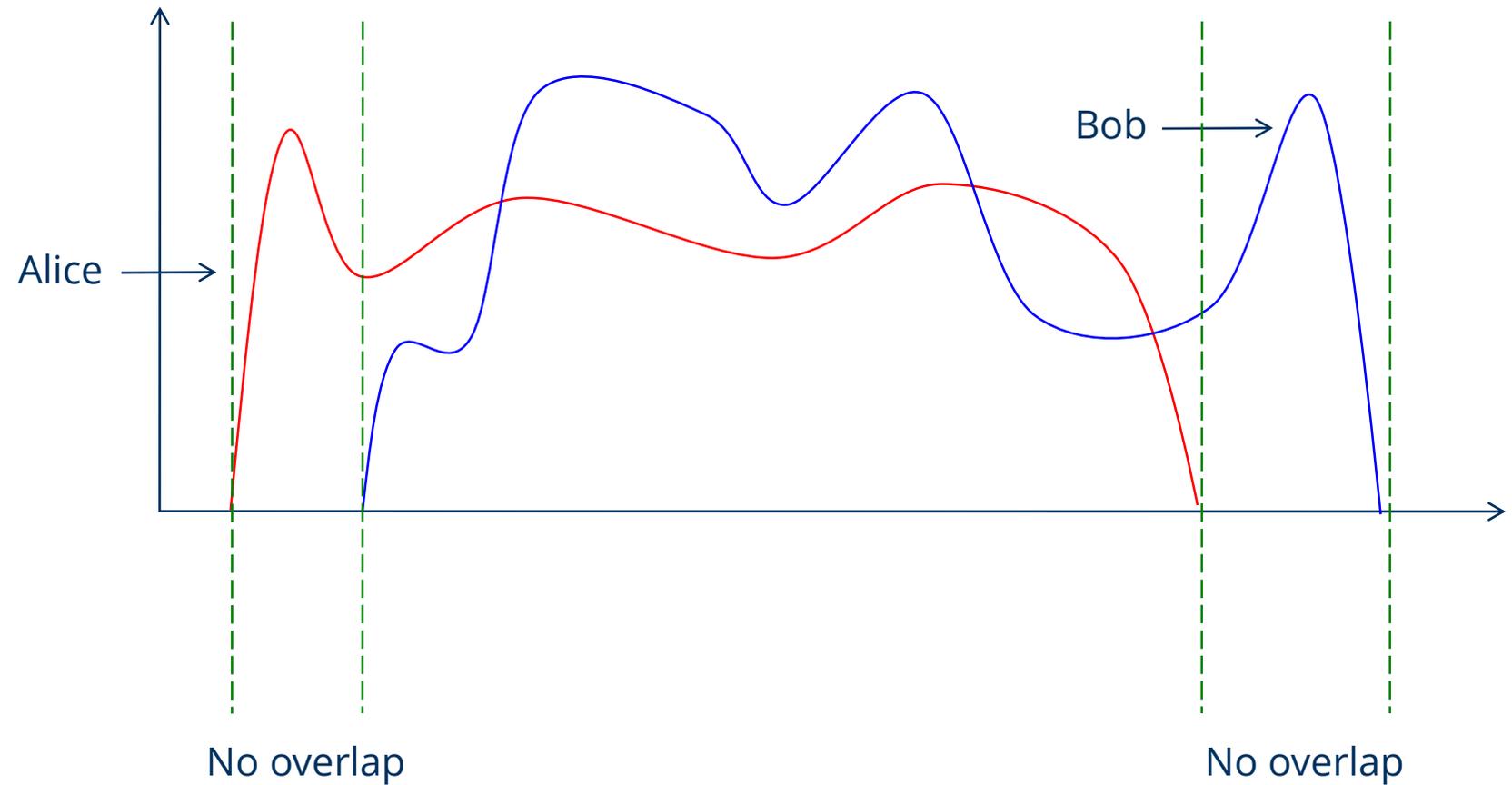
[Katti et al 2007] Analog network coding

[Gollakota et al 2008] ZigZag decoding

(different problem, similar intuition)

# Analog Network Coding

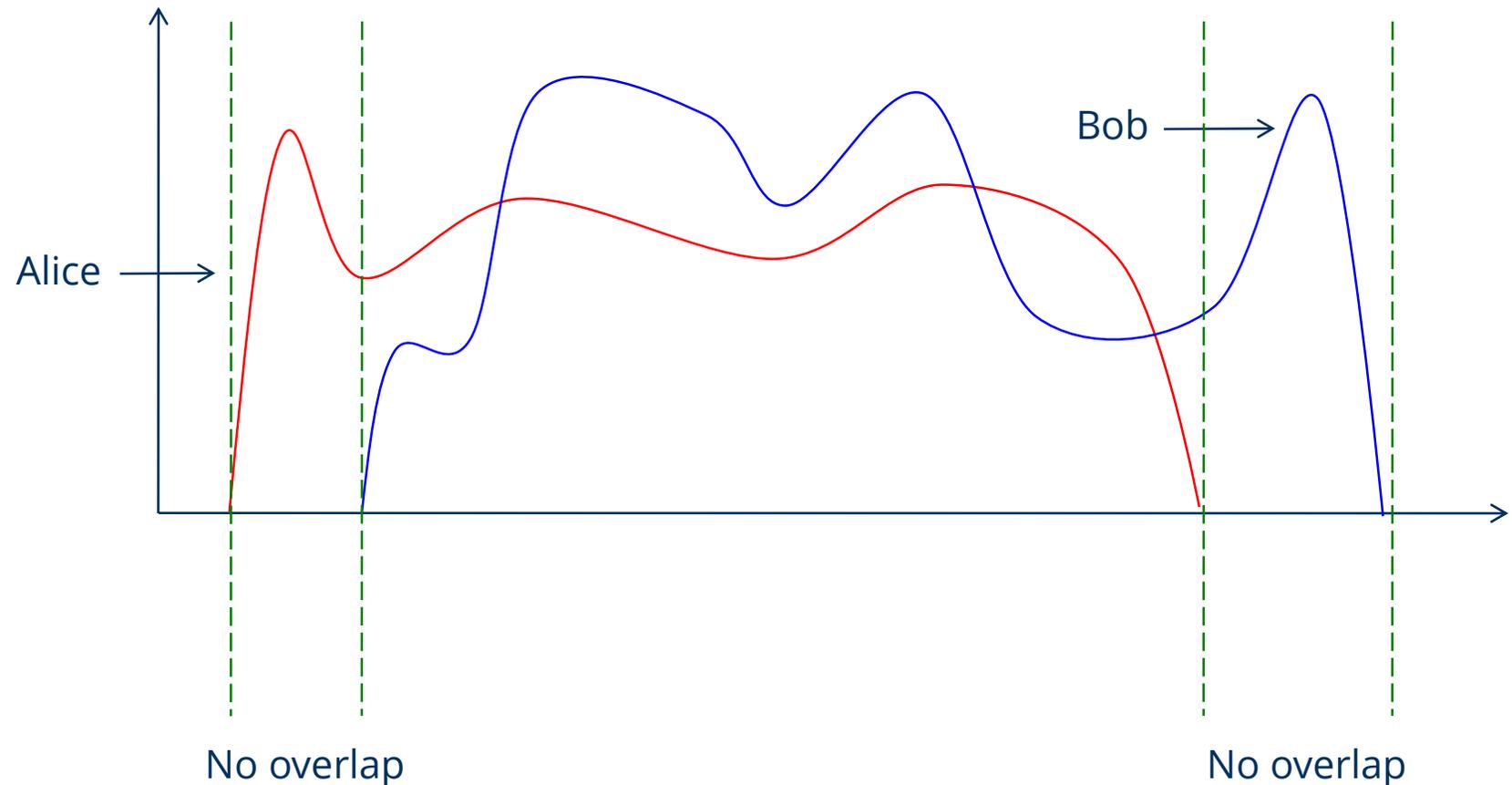
Key intuition: exploit asynchrony [Katti et al 2007]



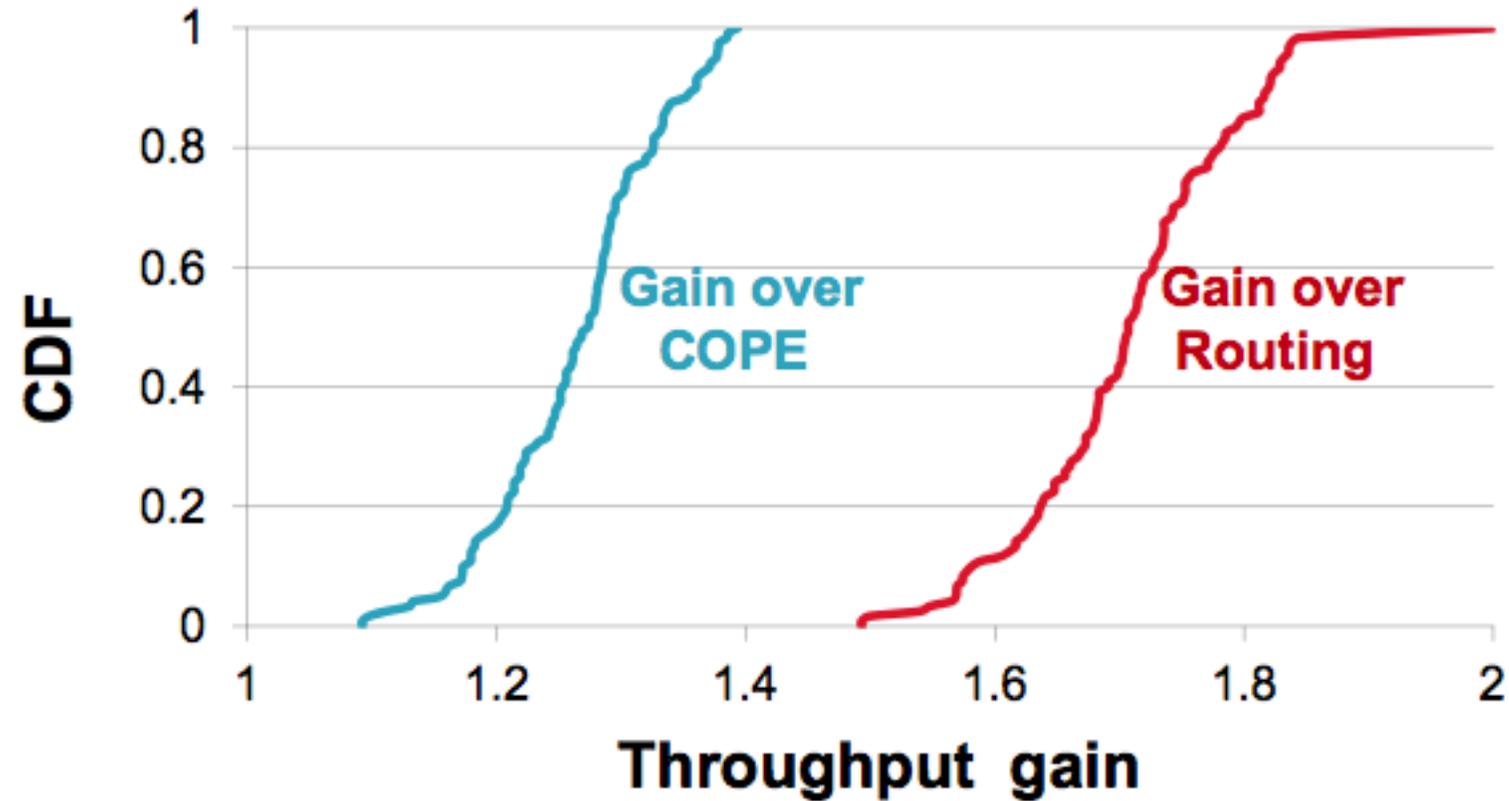
# Analog Network Coding

Pilot sequence: channel estimation ID of sender + destination + sequence number of the packet: active session and to determine which packet was used

Construct „header“ and „footer“ for each packet



# Analog Network Coding



# ZigZag Decoding

Draws from the same intuition as the above problem

Difference:

- More general setting
- A node can use it to recover several interfering signals (no knowledge required on its end)
- We need to receive  $n$  collisions of  $n$  packets to recover

Where is it useful?

- Hidden terminal problem
- In high SNR, to boost overall data rate from multiple sources to a single receiver [ParandehGheibi et al 2010]

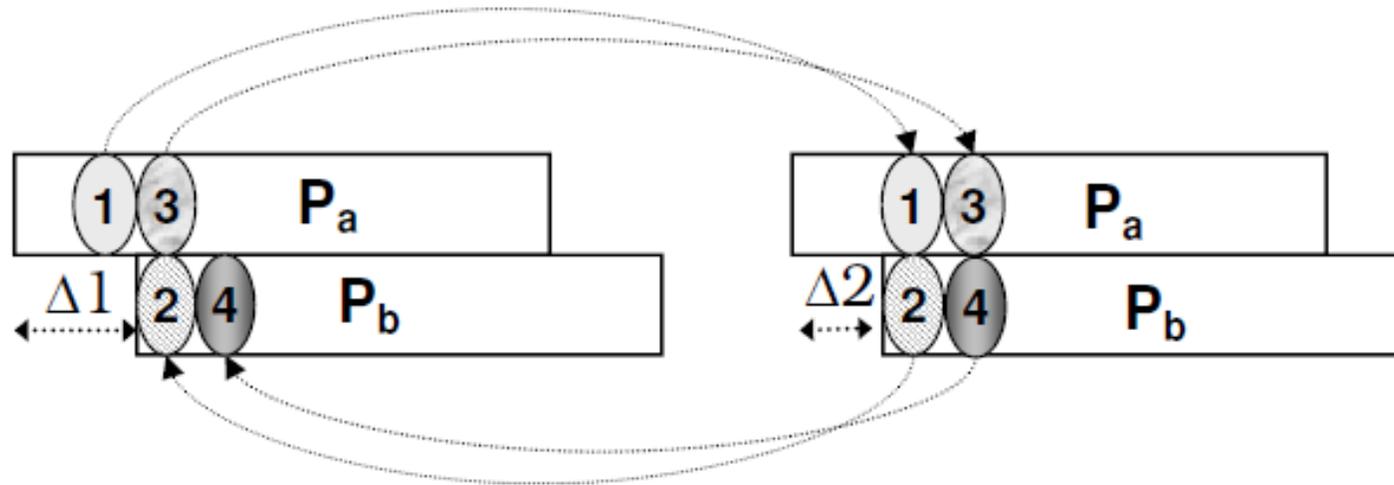
# ZigZag Decoding: Basic Idea

Again: asynchrony

Chunk 1 of bits from user A from 1st collision is decoded successfully

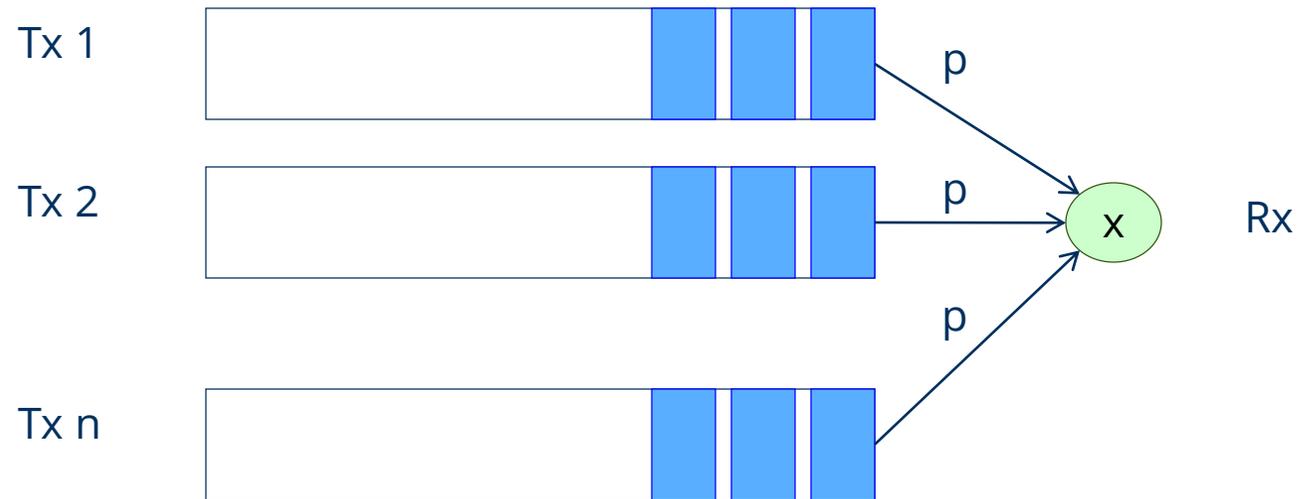
Thus, can subtract it from 2nd collision to decode Chunk 2 of bits of user B

Once Chunk 2 is free, can use to free Chunk 3, and so on



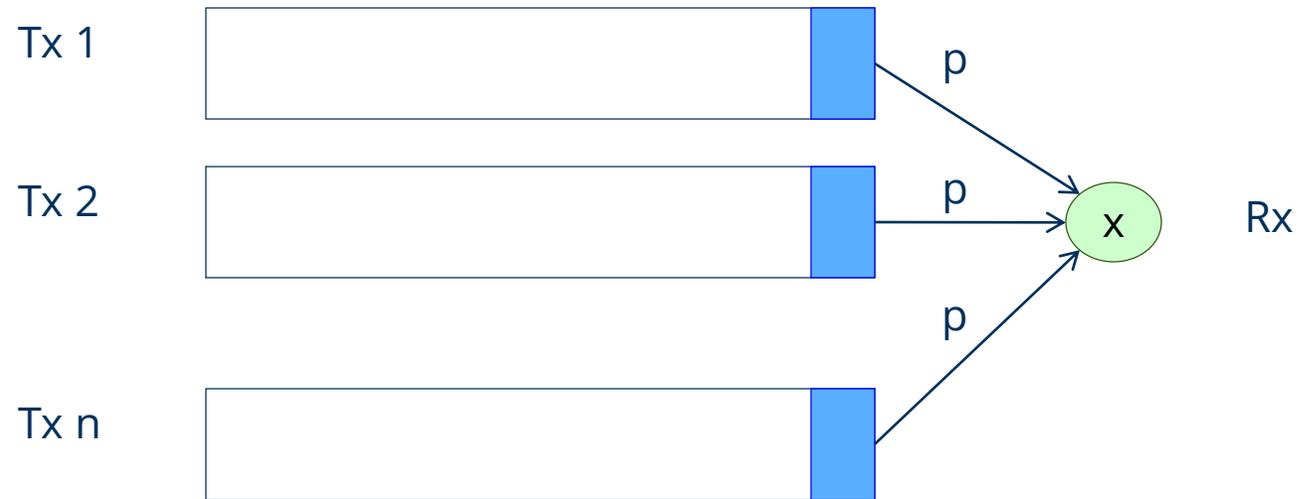
# ZigZag Decoding: Single Hop Analysis

Work in [ParandehGheibi et al 2010]



# ZigZag Decoding: Single Hop Analysis

Work in [ParandehGheibi et al 2010]



First result: Mean time to deliver one packet each

With zigzag: 
$$\frac{n}{1-p^n} \leq E[T_D] \leq \sum_{i=1}^n \frac{1}{1-p^{n-i+1}}$$

Perfect scheduler (no collisions): 
$$E[T_D] = \frac{n}{1-p}$$

**p = 1/2, n = 3**  
**ZZ: 4+**  
**10/21**  
**PS: 6**

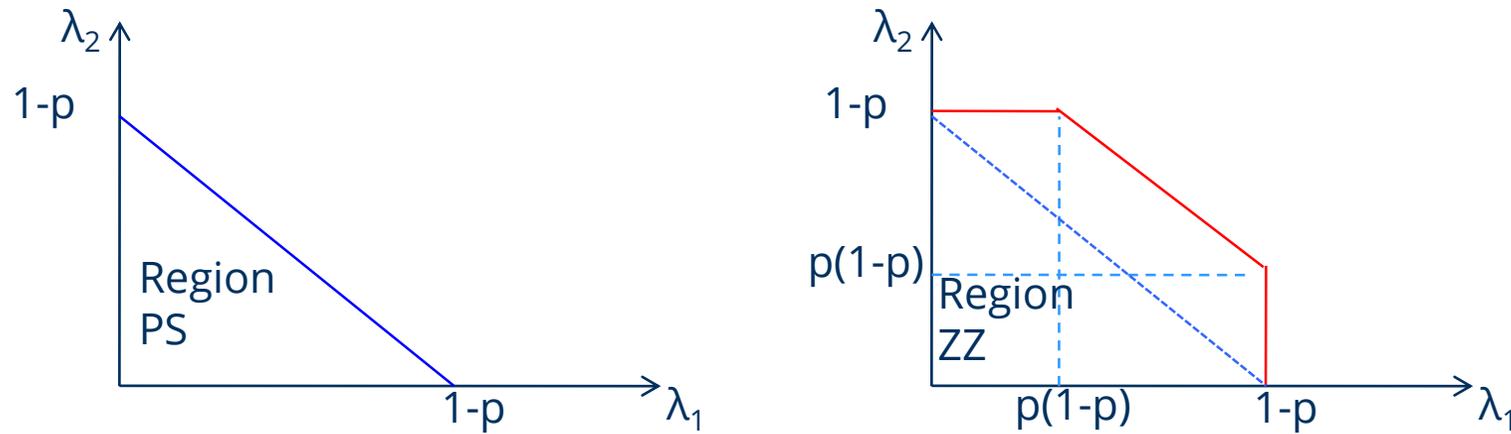
# ZigZag Decoding: Single Hop Analysis

<http://arxiv.org/pdf/1001.1948v1.pdf>

Work in [ParandehGheibi et al 2010]



Second result: Stable throughput increases



# Coded Access

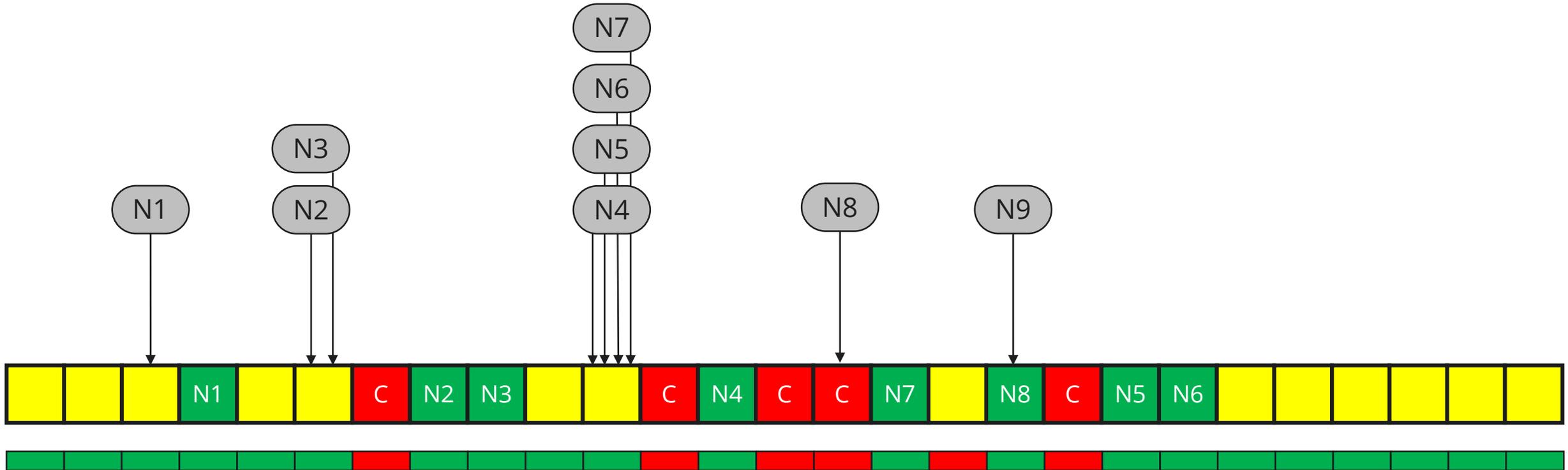
# Multiple Access SoA

Random access was the key to get access to the wireless channel

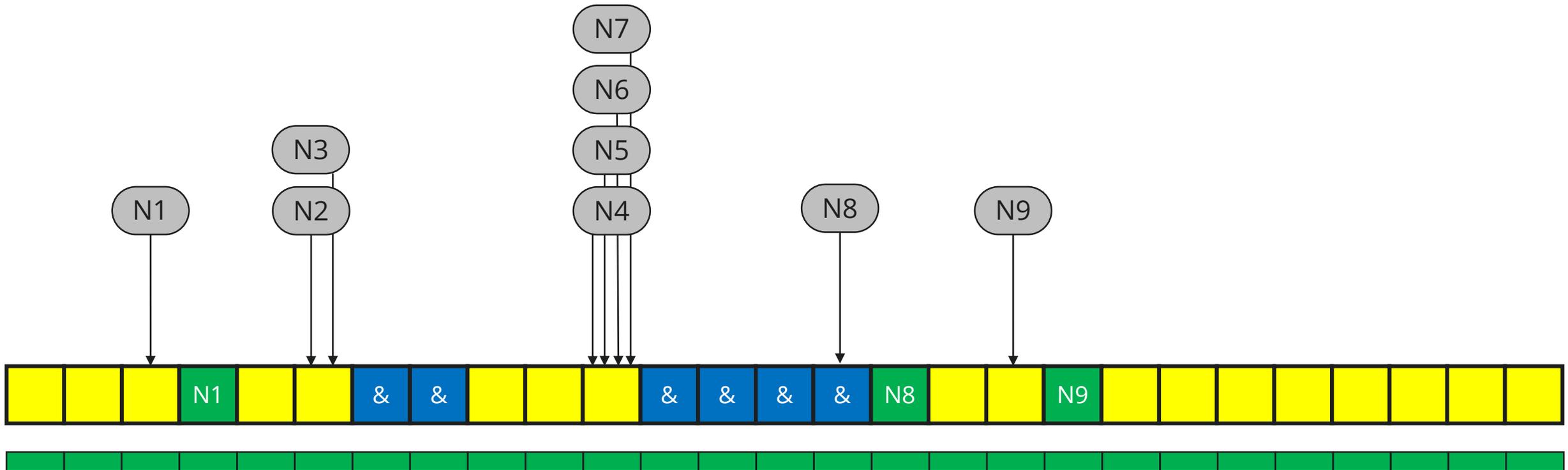
In time slotted systems

- Idle slots: nobody was transmitting
- One transmission: successful access to the resources
- More than one transmission: Collision

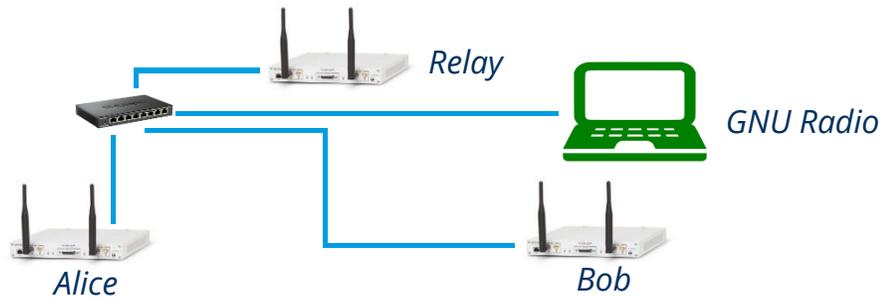
# SoA Multiple Access



# Coded Access



# Our Analog Network Coding Testbed



GNU Radio, from design to deployment

- With GNU Radio you can simulate, prototype, and deploy, all from the same workflow

Active Community

- <http://www.cgran.org/>
- <http://list.gnu.org/archive/html/discuss-gnuradio>

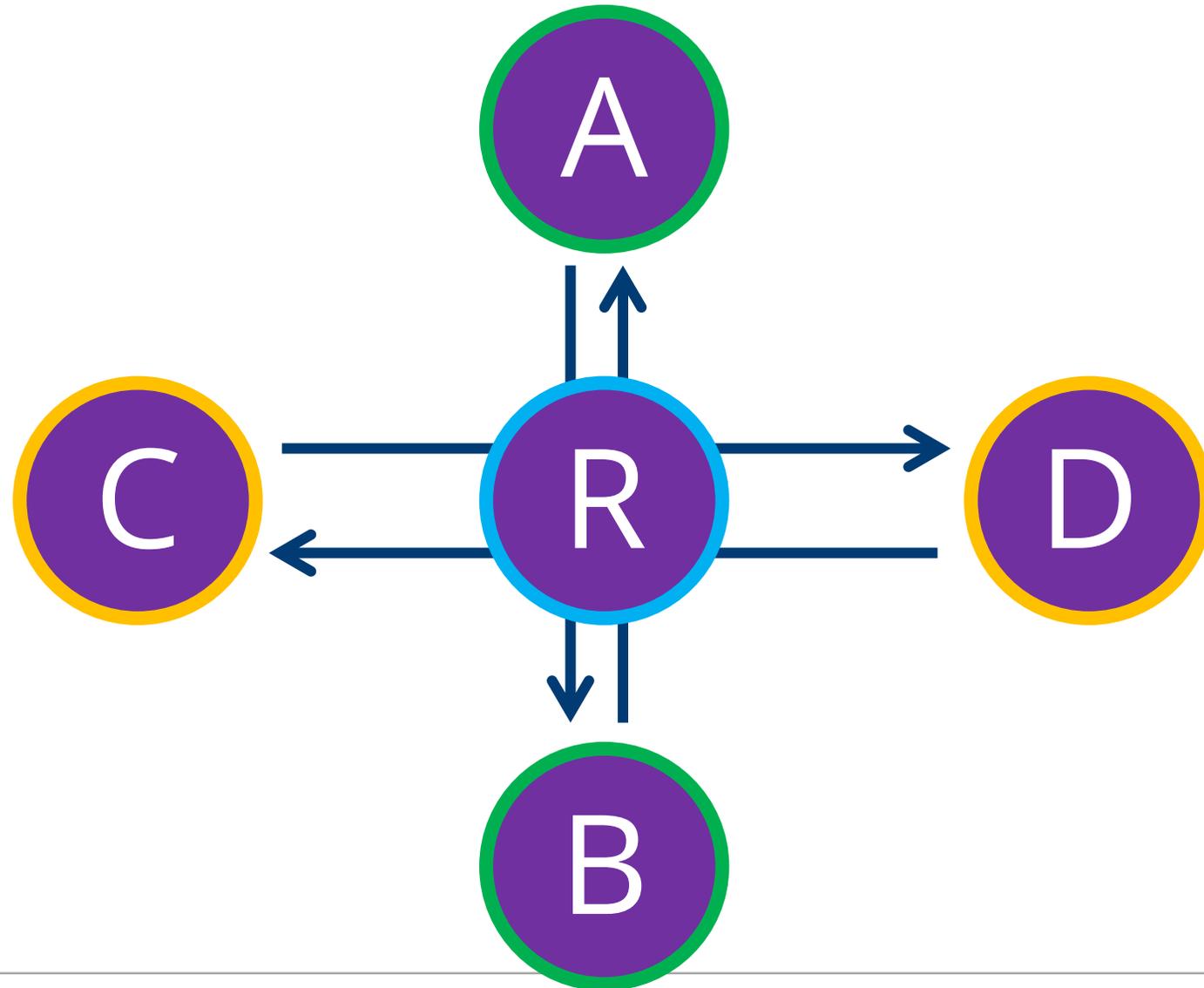
Free Software

- GNU Radio is free software. That means you have the liberty to use it and modify it as you wish
- <https://www.gnuradio.org/>

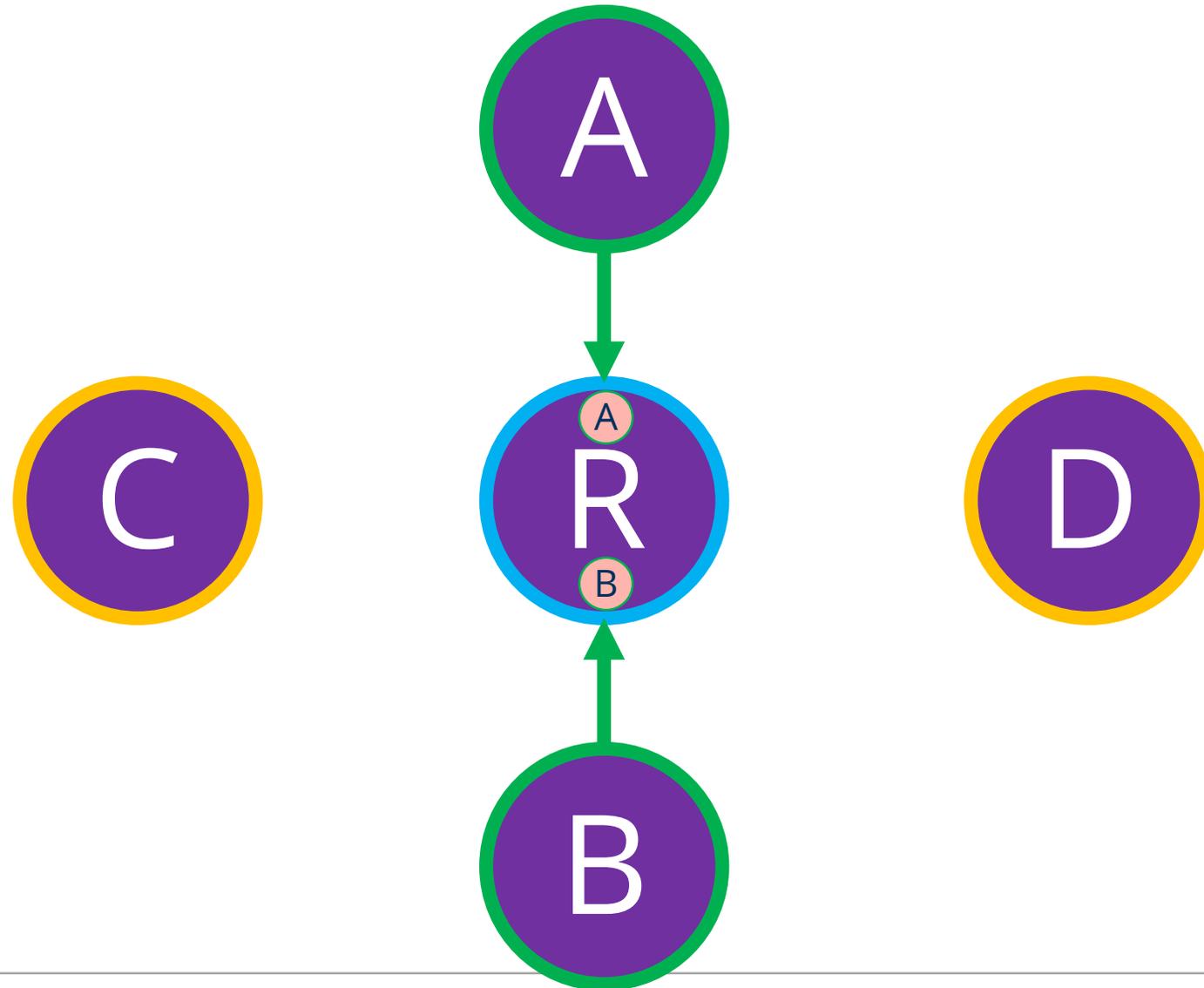


# ANC for cross topologies

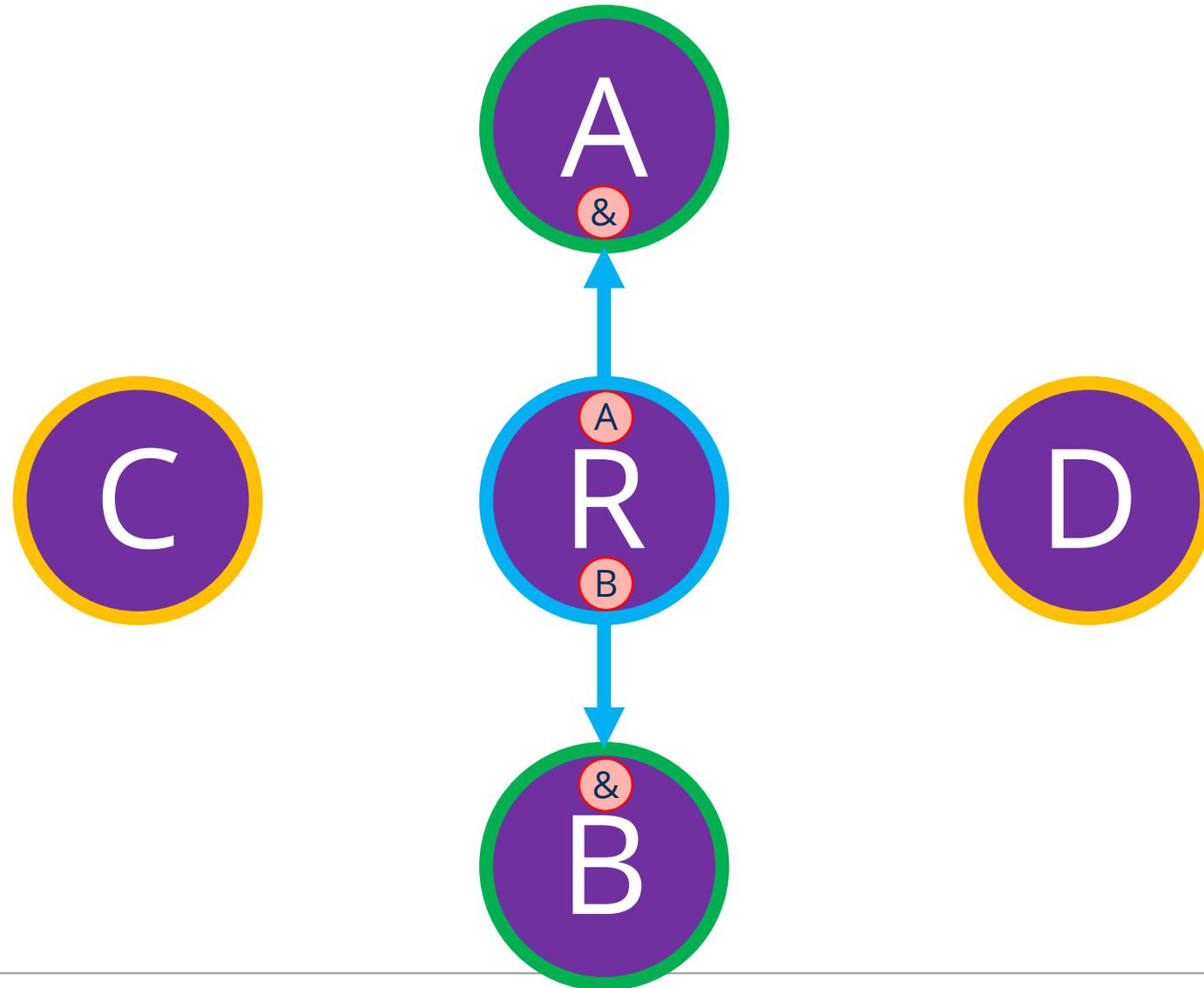
# Cross Topology ANC without Overhearing



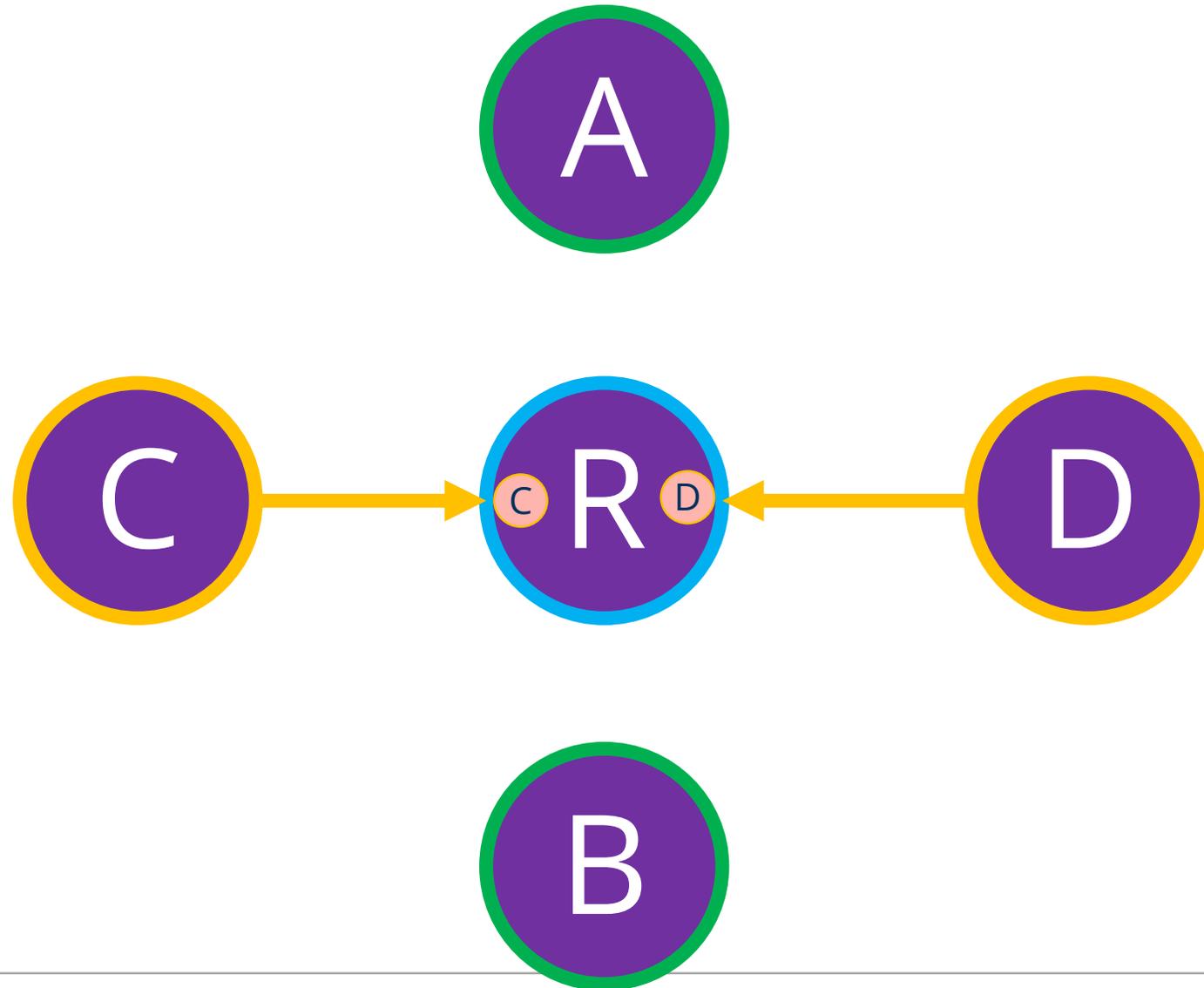
# Cross Topology ANC without Overhearing - Slot 1



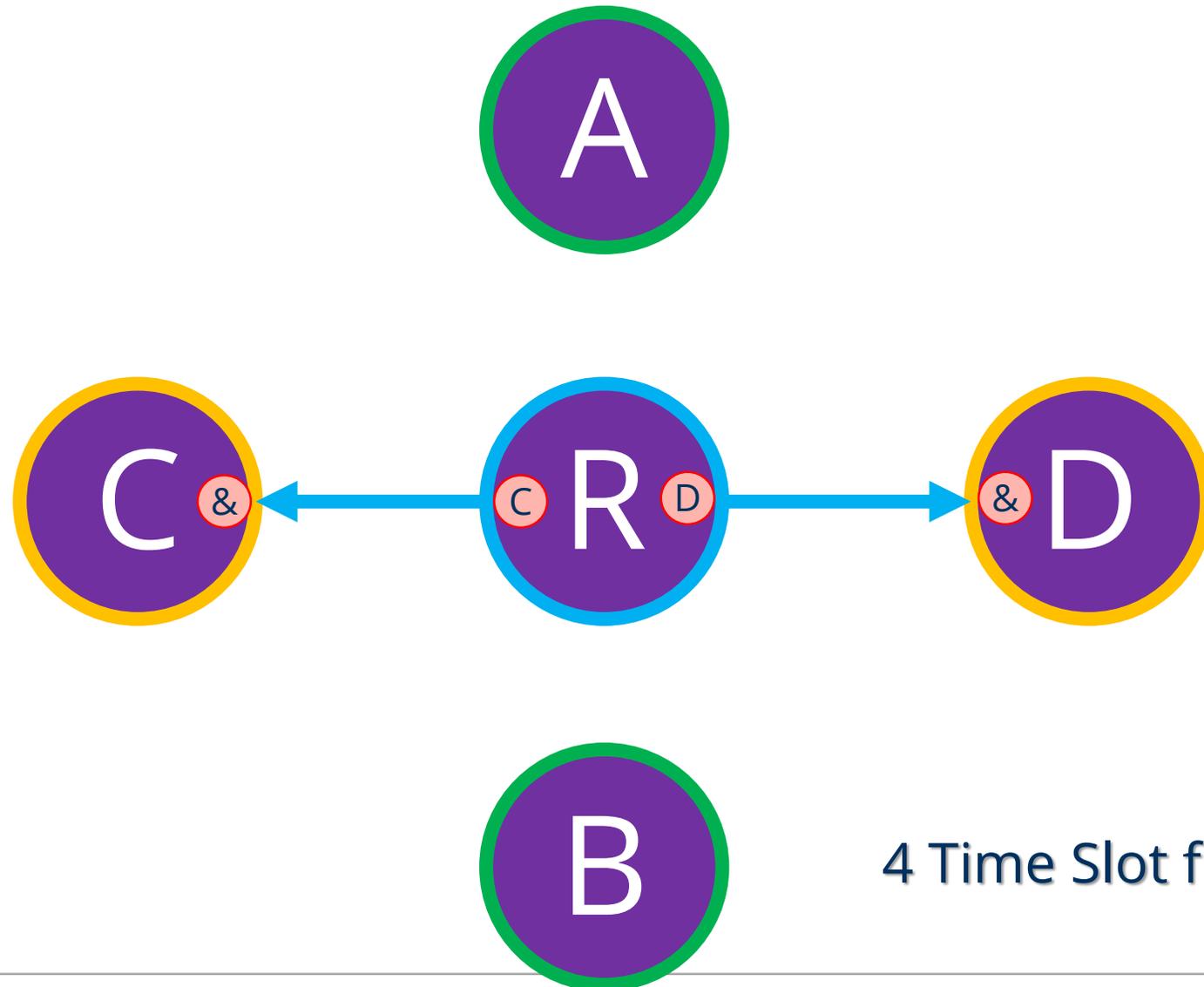
# Cross Topology ANC without Overhearing - Slot 2



# Cross Topology ANC without Overhearing - Slot 3

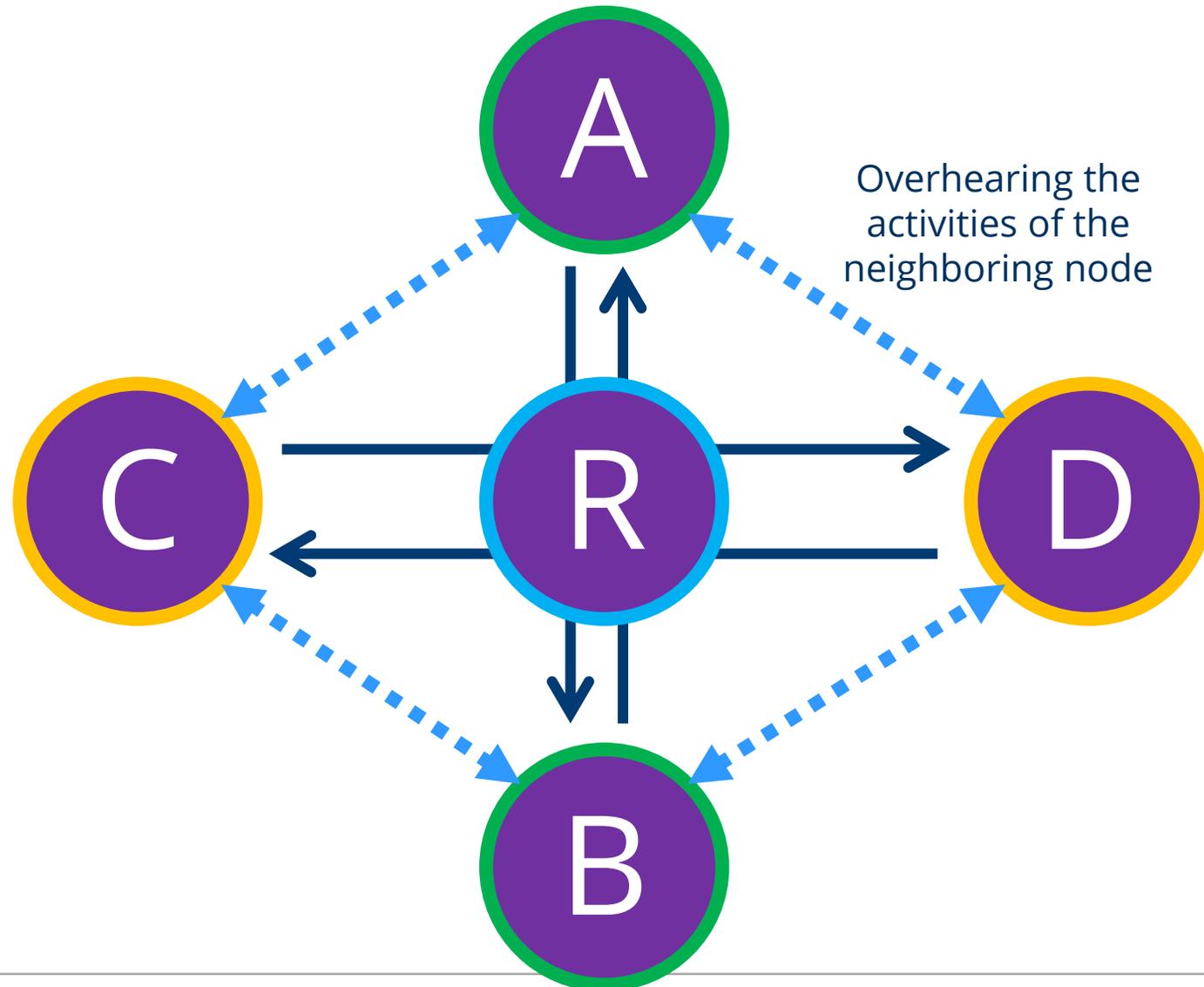


# Cross Topology ANC without Overhearing - Slot 4

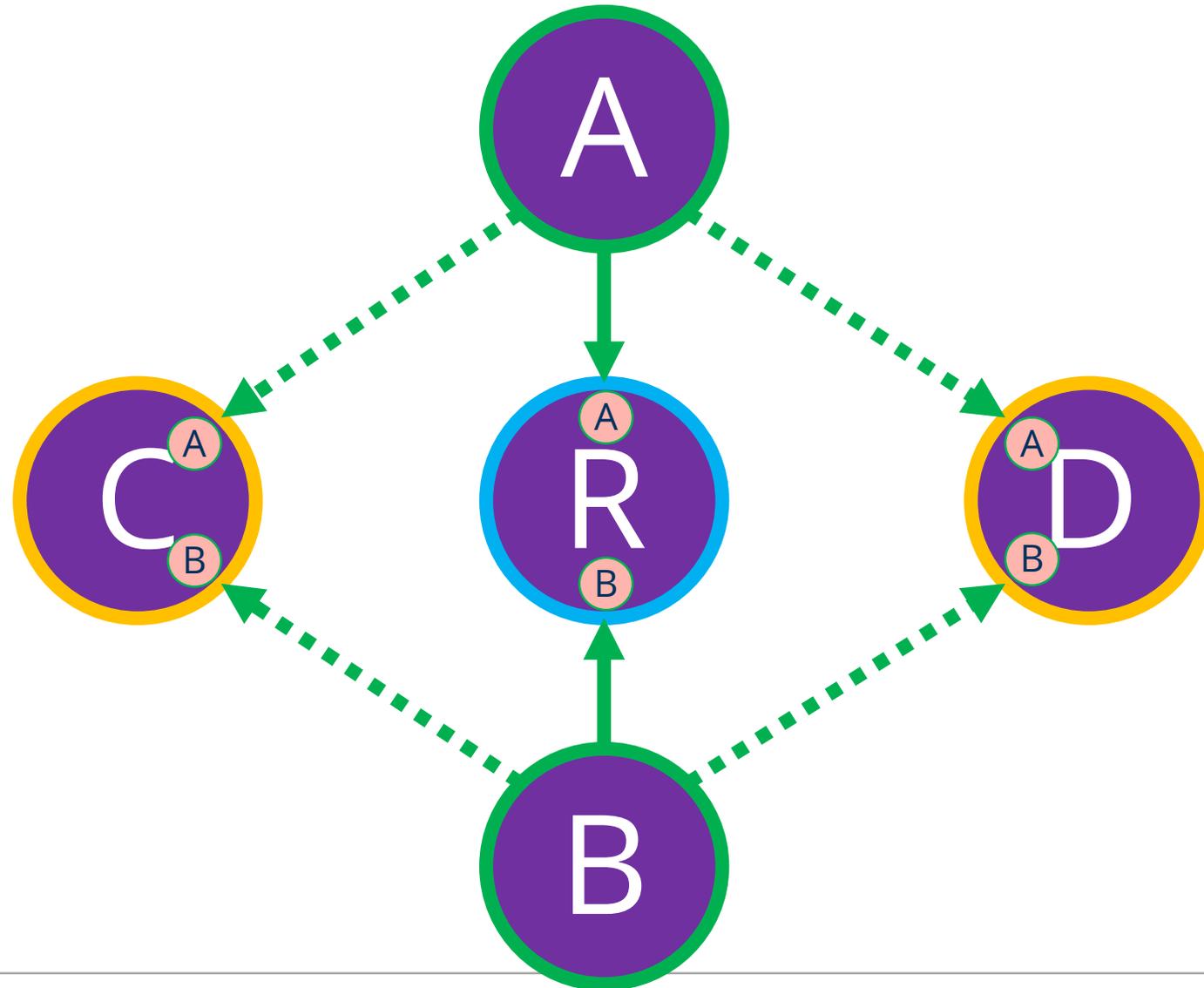


4 Time Slot for full exchange

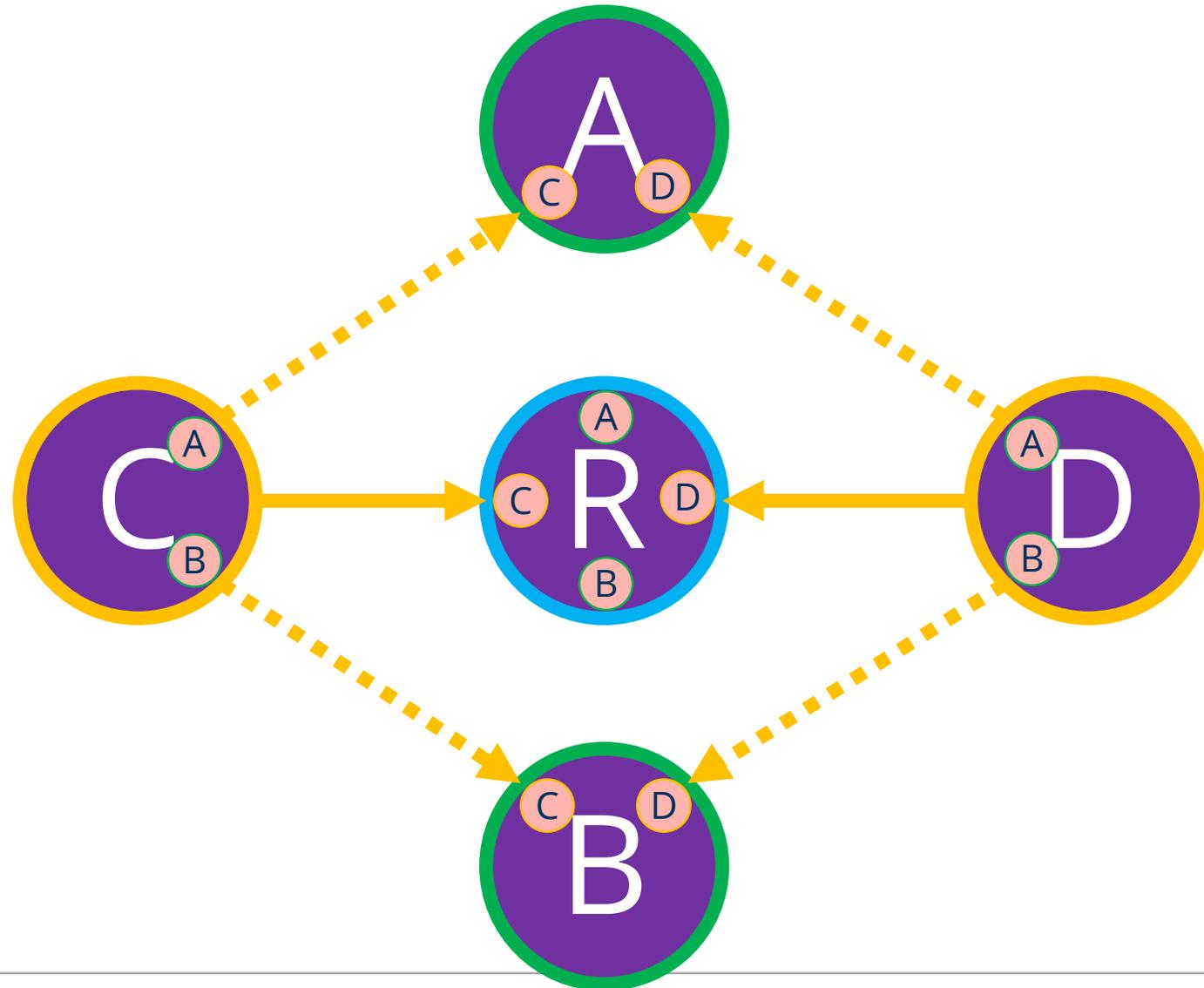
# Cross Topology ANC with Overhearing



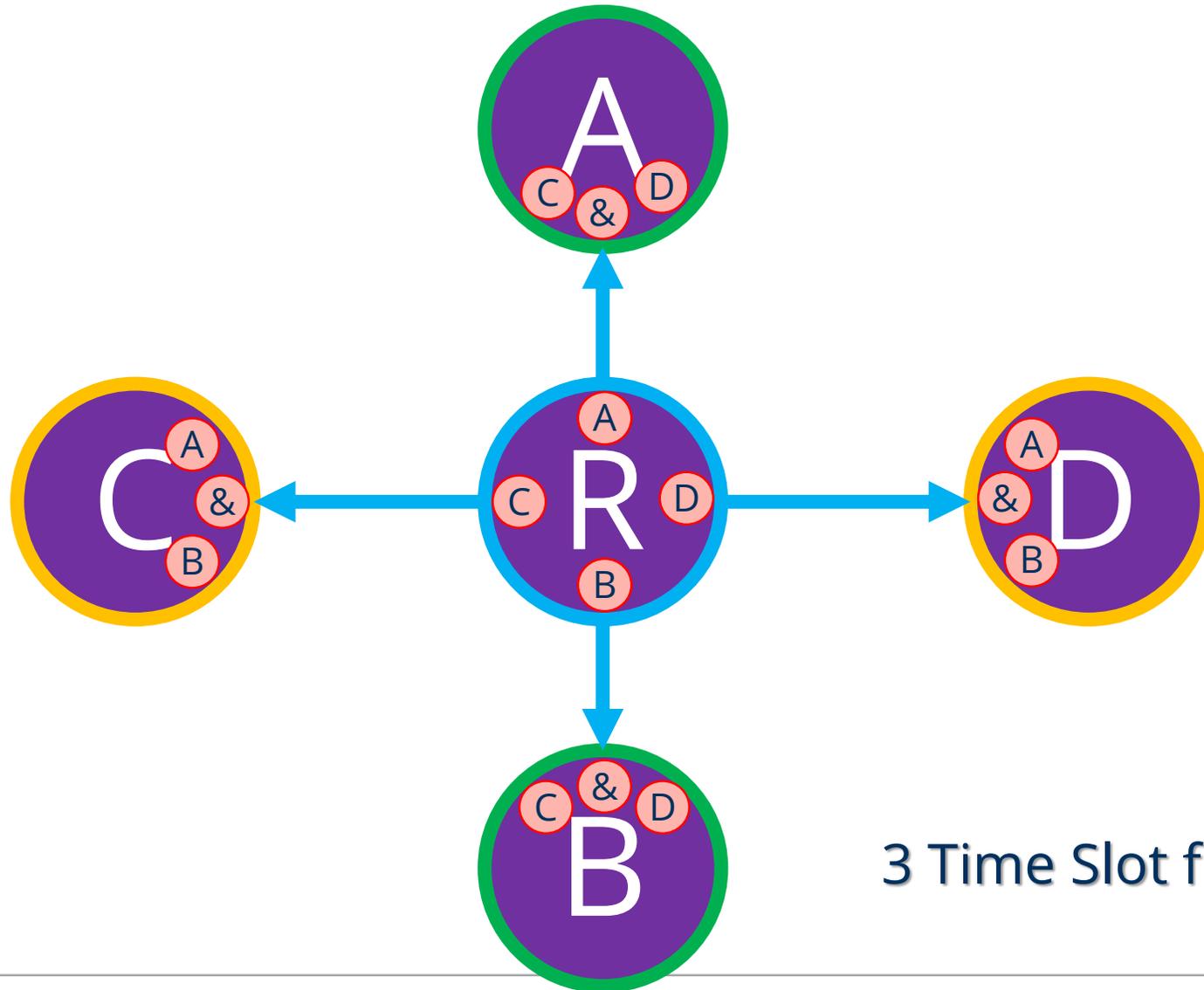
# Cross Topology ANC with Overhearing - Slot 1



# Cross Topology ANC with Overhearing - Slot 2

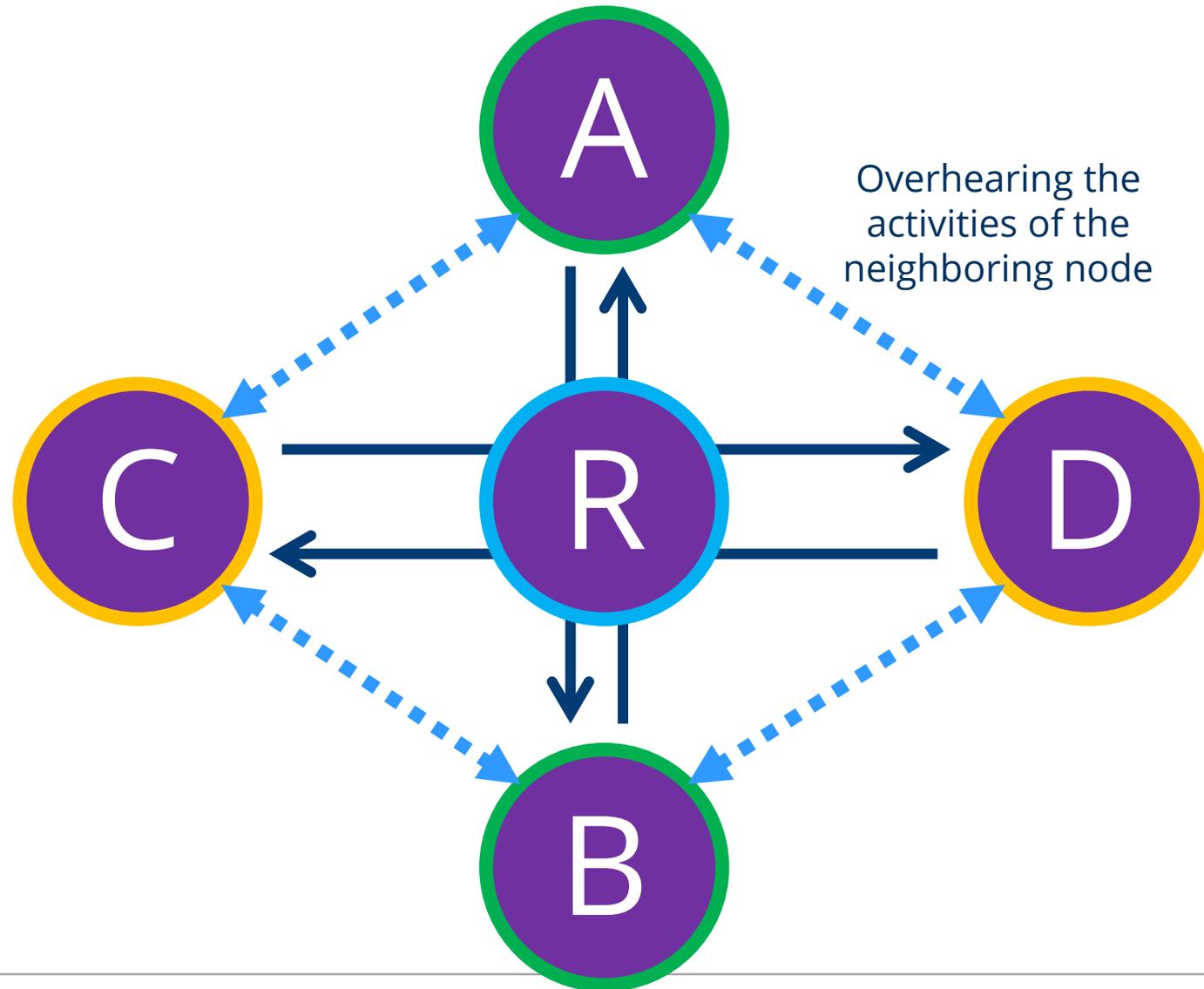


# Cross Topology ANC with Overhearing - Slot 3

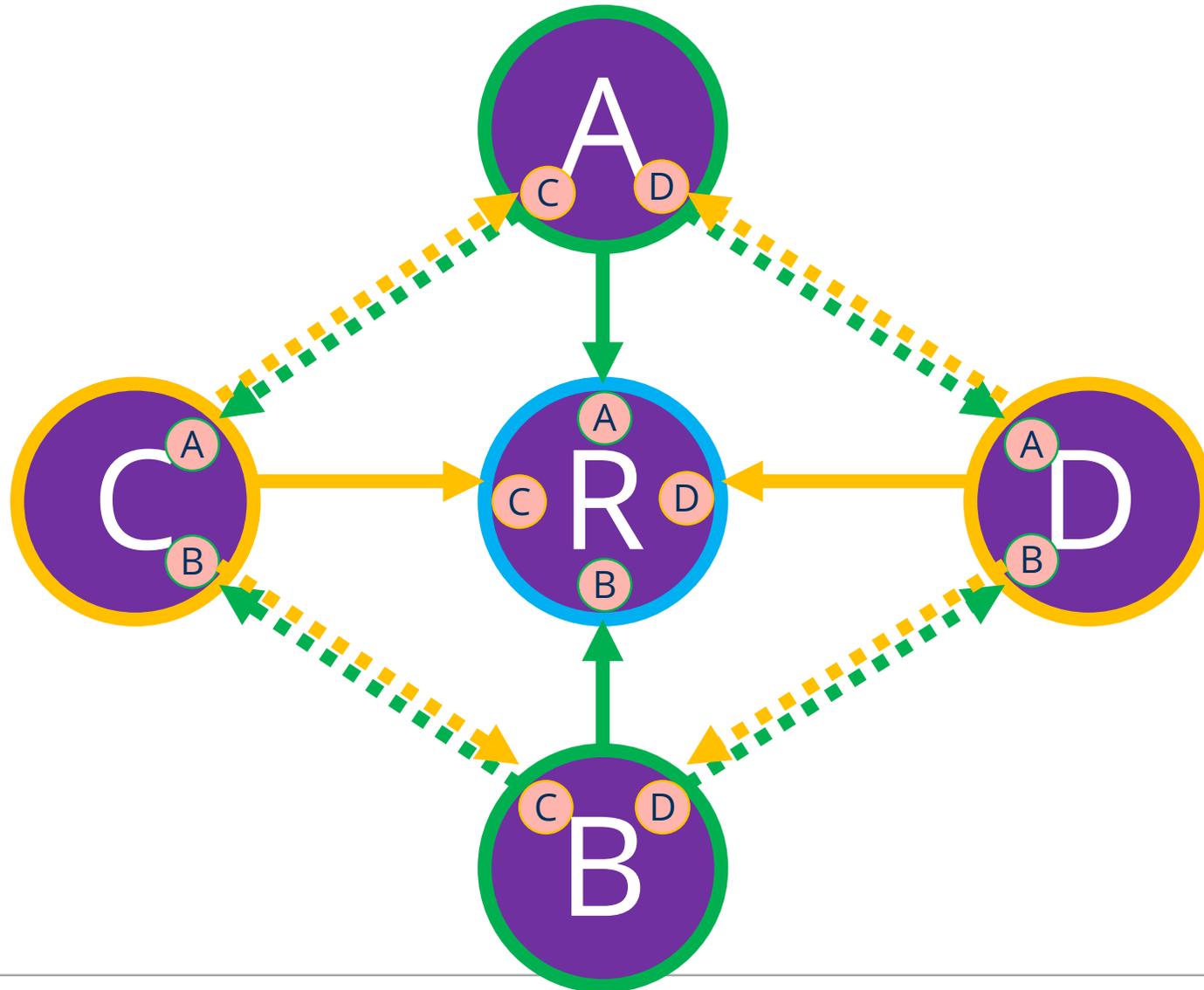


3 Time Slot for full exchange

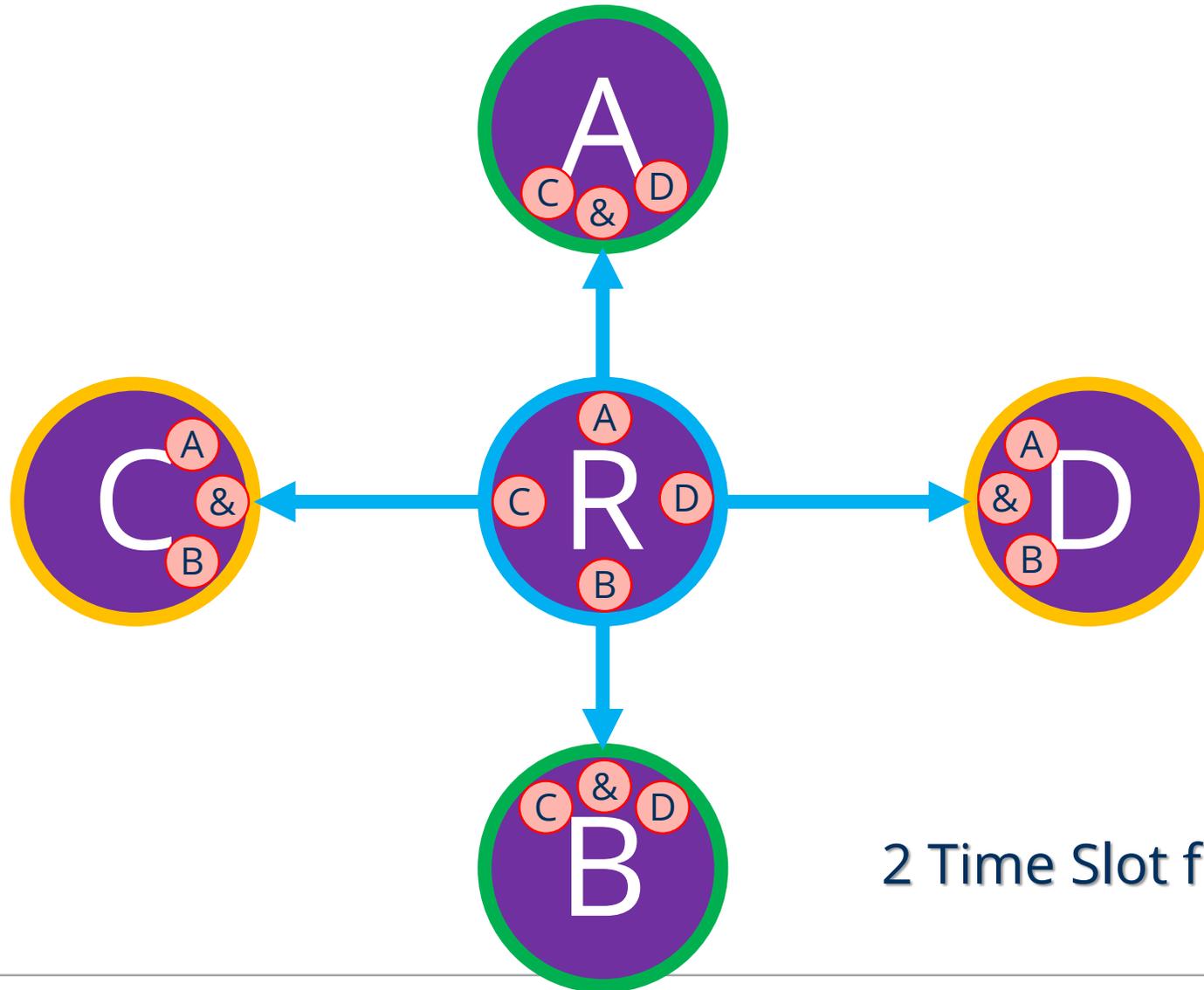
# Cross Topology ANC with Overhearing Duplex



# Cross Topology ANC with Overhearing Duplex – Slot 1

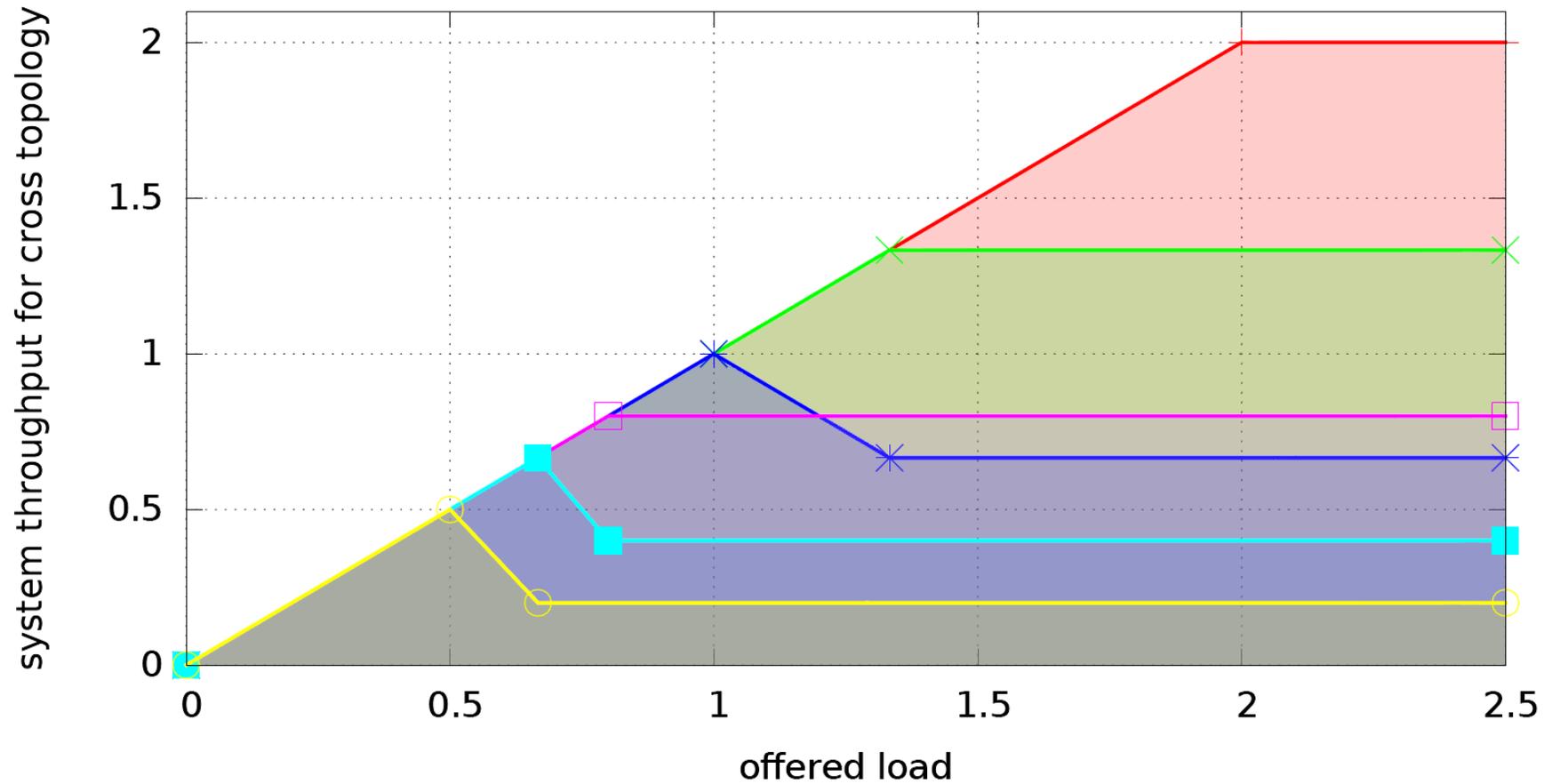


# Cross Topology ANC with Overhearing Duplex – Slot 2



2 Time Slot for full exchange

# Throughput for the Cross Topology



ANC duplex +    ANC w/o OH \*    DNC w/o OH ■  
 ANC w OH x    DNC w OH □    forwarding ○

# Inter-Flow Network Coding on CAN Bus

*Courtesy of BOSCH*

## Plug-and-Secure Communication for CAN



**Andreas Müller & Timo Lothspeich**

Robert Bosch GmbH, Germany

1

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

*Courtesy of BOSCH*

## Plug-and-Secure Communication for CAN

### Outline

- 1 Motivation
- 2 Fundamental Idea
- 3 Implementation Aspects
- 4 Security Considerations
- 5 Conclusion



2

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

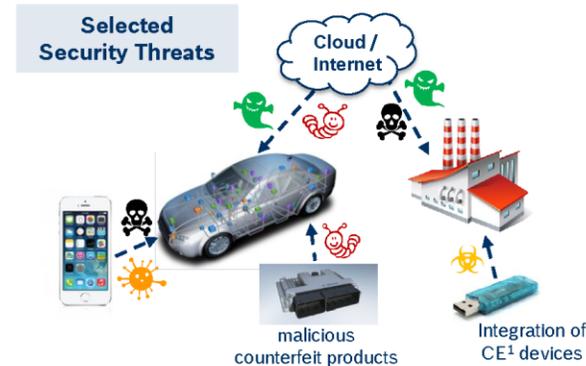
## Plug-and-Secure Communication for CAN

### Motivation

#### Motivation

- ❑ Current trends (e.g., Cloud/Internet connectivity) lead to novel & serious security threats
- ❑ Today's CAN networks are often hardly secured
- ❑ Cryptographic methods may help (e.g., message / entity authentication)

**?** **But:** Distribution of cryptographic keys between devices as a major challenge



**Our Idea:** Novel approach for completely automated & secure key distribution of very low complexity for CAN networks (“plug-and-secure”)

- ➔ Potential building block and enabler for future secure CAN networks
- ➔ Especially suitable against software-based & remote attack scenarios
- ➔ Basic Idea: Exploit special properties of CAN bus (dominant / recessive bits)

<sup>1</sup>Consumer Electronics

3

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



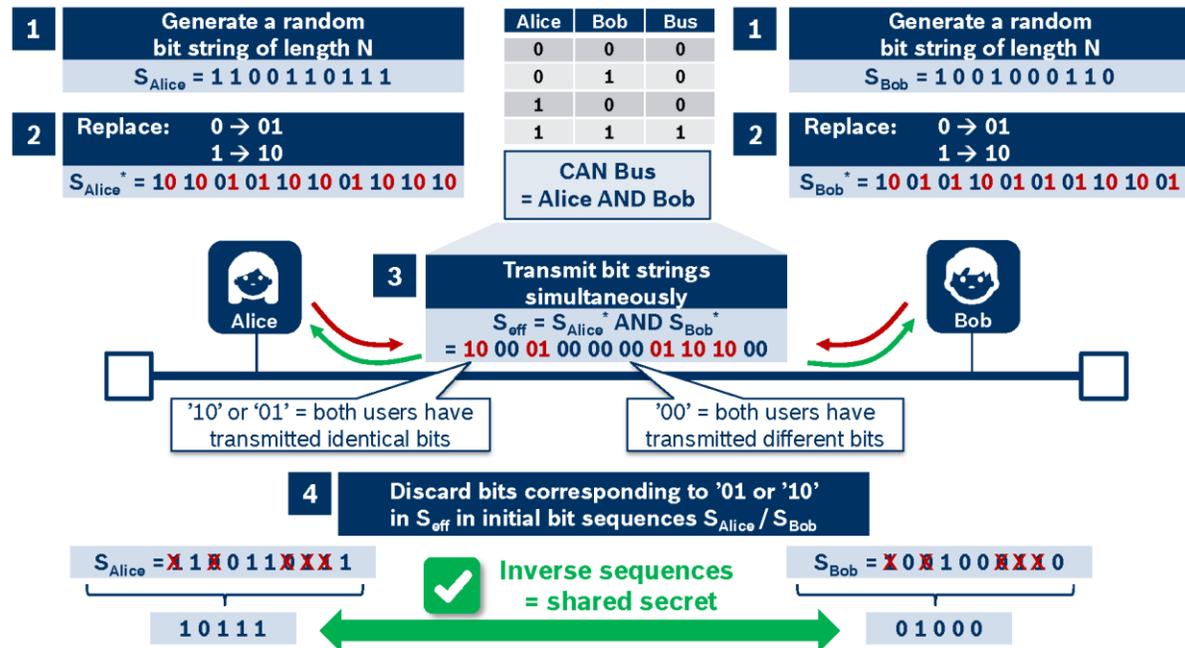
**BOSCH**

# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Basic Idea



4

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

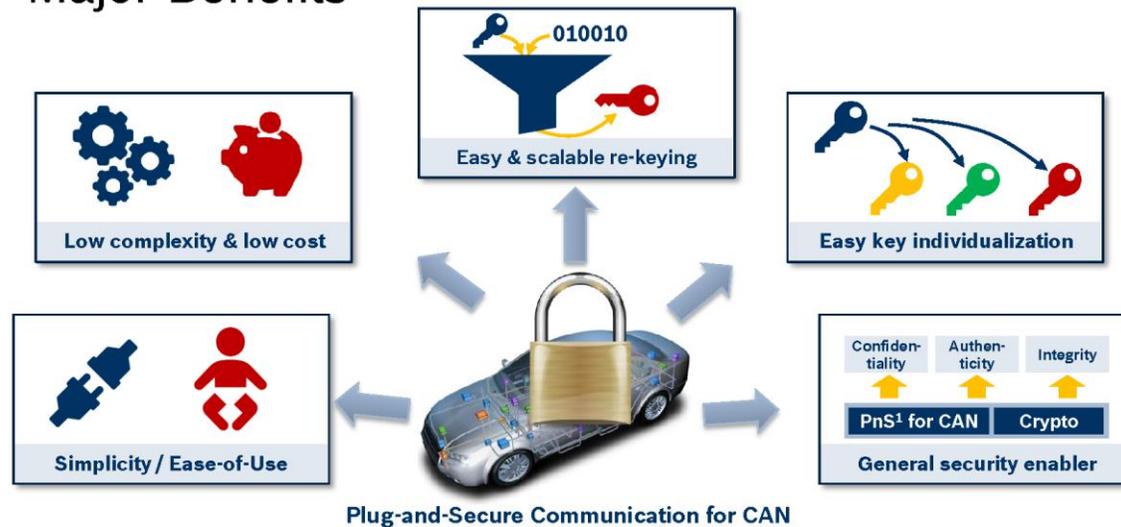


# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Major Benefits



➔ Seamless integration in CAN ecosystem, simple add-on to existing CAN controllers sufficient

➔ Approach may be readily extended to other bus systems, such as LIN<sup>2</sup>, I<sup>2</sup>C<sup>3</sup>, etc.

<sup>1</sup>Plug-and-Secure  
<sup>2</sup>Local Area Interconnect

<sup>3</sup>Inter-Integrated Circuit



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

*Courtesy of BOSCH*

Plug-and-Secure Communication for CAN

## Implementation Aspects

6

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

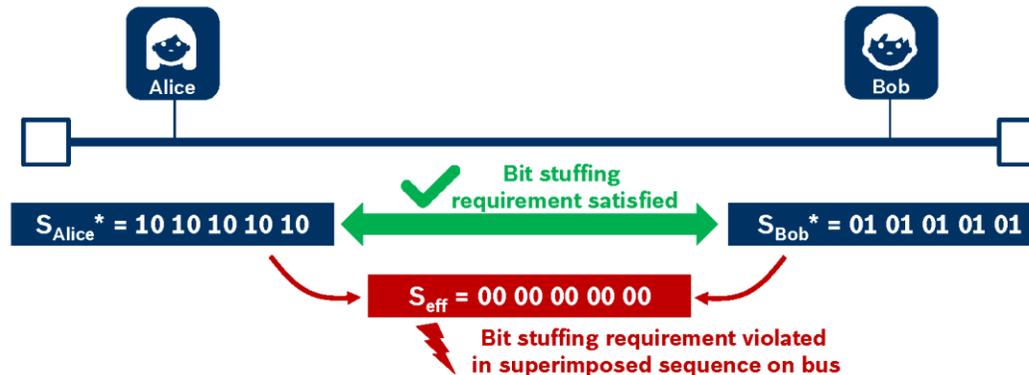
# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Bit Stuffing

⚠ After 5 equal bits in a frame, a stuff bit has to be introduced



💡 Insert stuffing bits on-the-fly based on effective bit sequence on the bus

<sup>1</sup>End of Frame

7

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

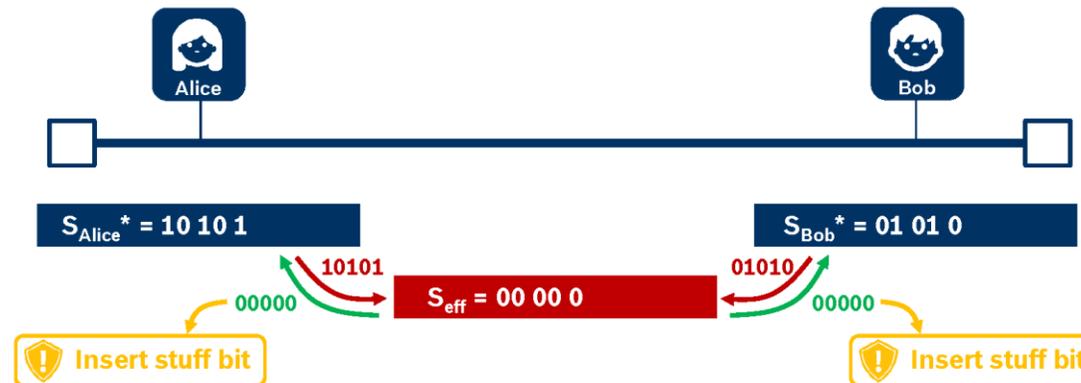
Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Bit Stuffing



After 5 equal bits in a frame, a stuff bit has to be introduced



Insert stuffing bits on-the-fly based on effective bit sequence on the bus

<sup>1</sup>End of Frame

8

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

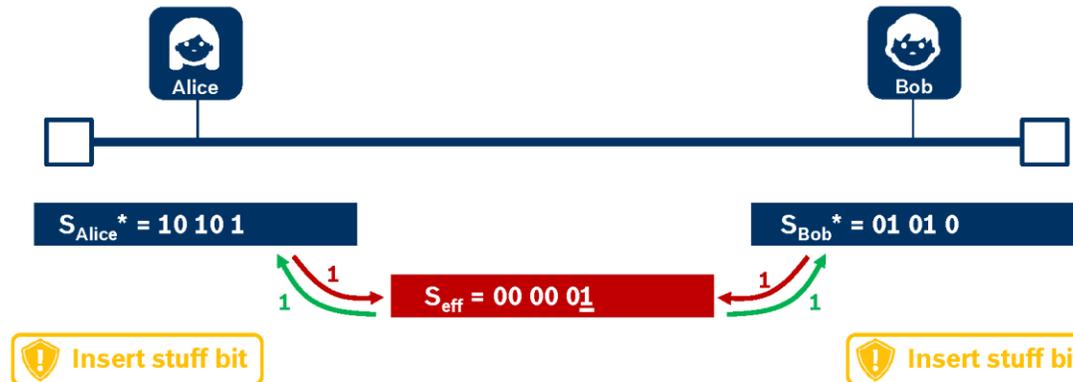
Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Bit Stuffing



After 5 equal bits in a frame, a stuff bit has to be introduced



Insert stuffing bits on-the-fly based on effective bit sequence on the bus

<sup>1</sup>End of Frame

9

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Bit Stuffing



After 5 equal bits in a frame, a stuff bit has to be introduced



Insert stuffing bits on-the-fly based on effective bit sequence on the bus

<sup>1</sup>End of Frame

10

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

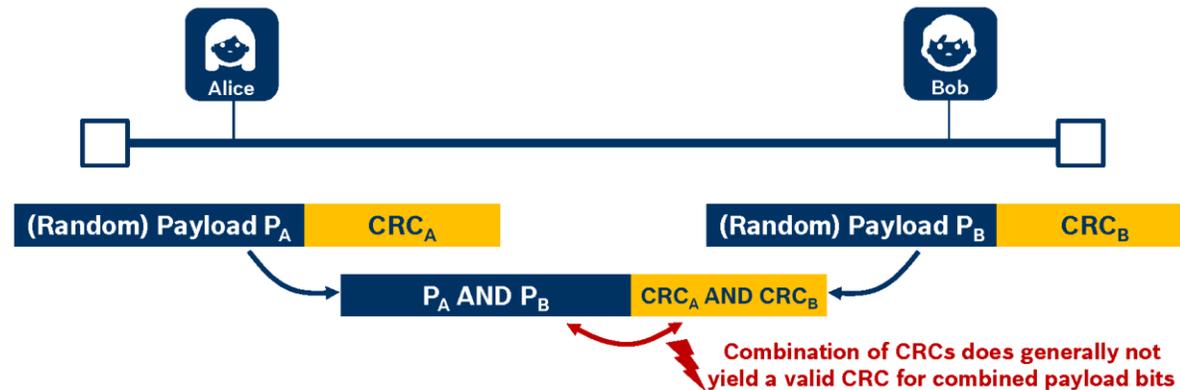
Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### CRC Check



Cyclic redundancy check field enables detection of transmission errors  
→ Superimposed CRC fields ≠ correct CRC for superimposed payloads



Calculate CRC on-the-fly based on effective payload field on the bus

11

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

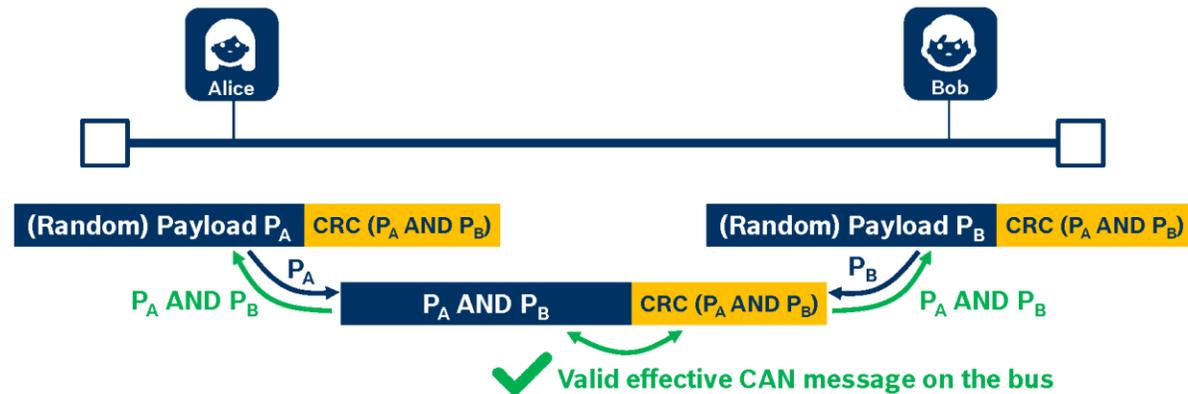
Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### CRC Check



Cyclic redundancy check field enables detection of transmission errors  
→ Superimposed CRC fields ≠ correct CRC for superimposed payloads



Provides automatic verification that Alice and Bob have received the same superimposed bit sequence for free 😊

12

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



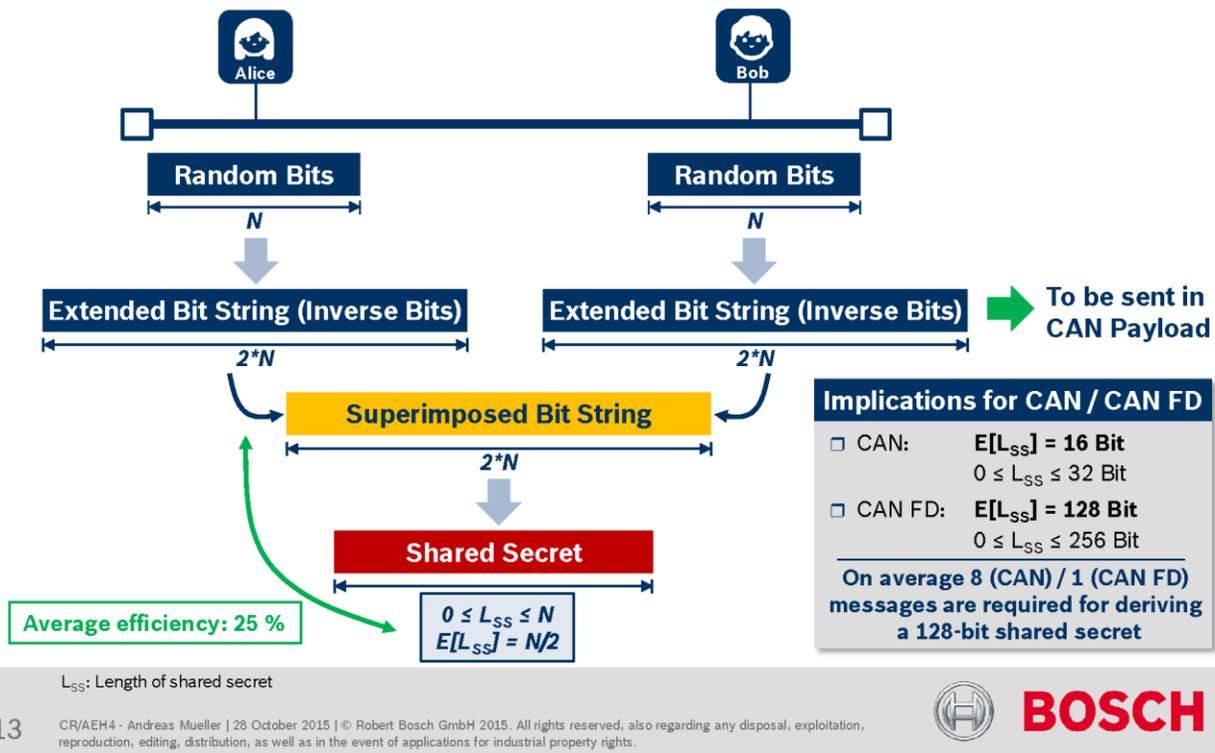
**BOSCH**

# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Performance / Efficiency



13

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

*Courtesy of BOSCH*

Plug-and-Secure Communication for CAN

## Security Considerations

15

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



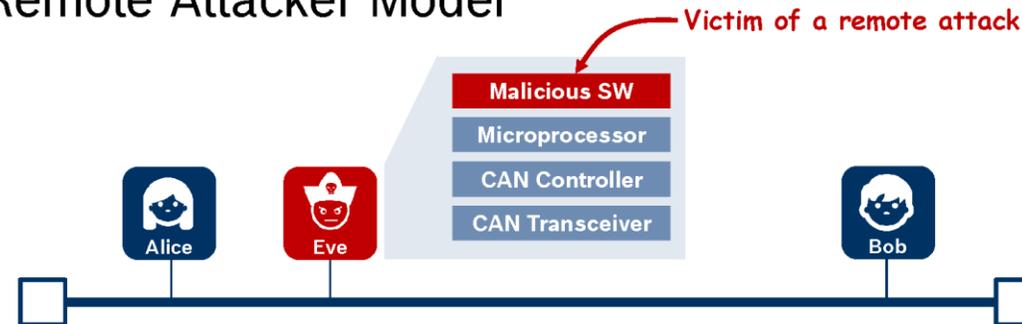
**BOSCH**

# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Remote Attacker Model



#### Assumptions

- 1 Eve is using standard HW with modified (malicious) SW
- 2 Eve may eavesdrop on all messages exchanged on the CAN bus
- 3 Eve may inject arbitrary bits on the CAN bus (via the CAN transceiver)

➔ Highly relevant attacker model due to easy scalability of attacks!

16

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Passive Eavesdropping



Idea: Passively eavesdrop on the channel during key setup

Transmitted Bits		Observed Bits
Alice	Bob	Eve
01	01	01
01	10	00
10	01	00
10	10	10

Transmission of tuples (random bit + inverse bit)      Observation of regenerated bits only (→ through CAN transceiver)

Eve has full knowledge of bits transmitted by Alice & Bob, but bits will be discarded anyway

Eve is not able to tell who has transmitted '01' and who has transmitted '10' → secret 😊

Eve has full knowledge of bits transmitted by Alice & Bob, but bits will be discarded anyway

➔ A passive Eve cannot determine the established secret bits 😊

17

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Active Attacks (I)



Idea: Actively interfere with key establishment procedure

#### By superimposing recessive bits

Transmitted Bits		Eve	Bus
			CAN
0	0	1	0
0	1	1	0
1	1	1	1

No different from not transmitting at all → no impact 😊

#### By superimposing dominant bits

Transmitted Bits		Eve	Bus
			CAN
01	01	00	00
01	10	00	00
?1	?0	01	00

Requires a closer look



# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Active Attacks (II)



**Idea: Actively interfere with key establishment procedure**

#### By superimposing dominant bits

Transmitted Bits	Eve	Bus
Alice	Bob	Eve
01	01	00
01	10	00
?1	?0	01

- Manageable**  
Alice and Bob wrongfully assume that they have transmitted different bits, even though it is not true  
→ Derived keys are not identical  
→ Solution: Perform final key verification
- No problem**  
No impact → same situation as w/o interference
- Good for us**  
If Eve sends a dominant bit, she cannot tell the status on the bus would have been w/o her

➔ **An active Eve can prevent a successful key establishment, but cannot determine or influence the established keys 😊**

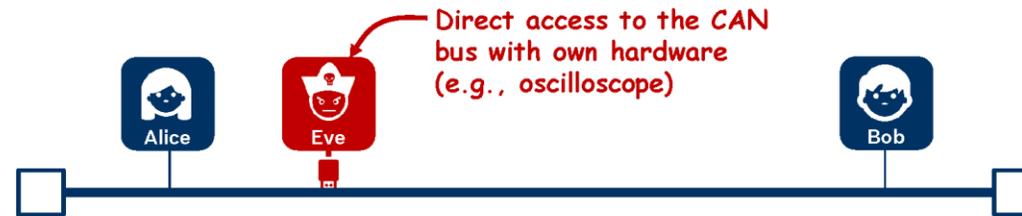


# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Attacker Model w/ Physical Access to CAN Bus



#### Principle Threats



#### BUT:

- With physical access, an attacker could compromise a vehicle much easier (e.g., cutting a cable)
- Attacks requiring physical access do not scale; threat w/ physical access has existed ever since
- Countermeasures are possible → e.g., artificial (random) jitter in bit timing

20

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

*Courtesy of BOSCH*

Plug-and-Secure Communication for CAN

## Proof-of-Concept Demonstration

21

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

*Courtesy of BOSCH*

## Plug-and-Secure Communication for CAN

### Proof-of-Concept Demonstrator (I)



Portable demonstrator  
including Alice & Bob



Bus analyzer  
emulating attacker Eve

22

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



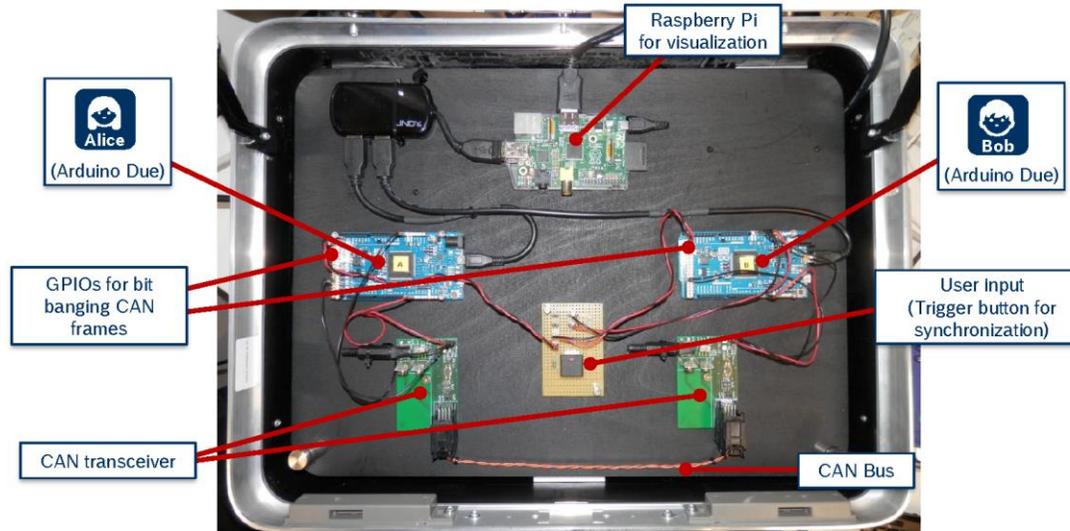
**BOSCH**

# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Proof-of-Concept Demonstrator (II)



➡ **Basic idea successfully demonstrated using off-the-shelf (maker) hardware**

➡ **100% standard compliant CAN frames on the bus → full backwards-compatibility**

23

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

Courtesy of BOSCH

## Plug-and-Secure Communication for CAN

### Conclusion

- 1 Security is getting more and more important in many domains
- 2 Distribution of cryptographic keys as a major challenge
- 3 PnS<sup>1</sup> for CAN as an innovative approach of very low complexity
- 4 Basic idea successfully demonstrated, fully backwards compatible
- 5 Major strengths: Remote / SW-based attacks, re-keying, wide applicability



Possible Extensions	Many Applications	Your Input
  <p>Extension to other bus systems with similar PHY rather straightforward</p>	  <p>Numerous applications in automotive, industrial &amp; other CAN networks</p>	 <p>Interested? → Get in touch with us 😊</p>

<sup>1</sup>Plug-and-Secure

24

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



**BOSCH**

# Inter-Flow Network Coding on CAN Bus

*Courtesy of BOSCH*

## Plug-and-Secure Communication for CAN

Thank you for your attention.



Dipl.-Ing. (FH)  
**Timo Lothspeich**

Automotive Electronics  
Product Security (AE-BE/EKE)

timo.lothspeich@de.bosch.com  
Tel.: +49-711-811-34016



Dipl.-Ing., M.Sc.  
**Dr.-Ing. Andreas Müller**

Corporate Sector Research and Advance Engineering  
Communication Technology (CR/AEH4)

andreas.mueller21@de.bosch.com  
Tel.: +49-711-811-20836



25

CR/AEH4 - Andreas Mueller | 28 October 2015 | © Robert Bosch GmbH 2015. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

# Summary Inter-Flow Network Coding

- Advantages
  - Easy to understand
  - Huge gains in theory
- Disadvantages
  - Requires signaling/planning
    - Not a huge problem in static networks, e.g. SDN controller
    - Huge problem for dynamic networks, more load and inefficient if not in time
- Real implementation (more than lab test run)
  - CATWOMAN in LINUX Kernel 3.10 onwards
  - BOSCH CAN Bus

# Backup

# Cautious View

Janus Heide, Morten V. Pedersen, Frank H.P. Fitzek, and Torben Larsen, **“Cautious view on network coding - from theory to practice,”** Journal of Communications and Networks (JCN), 2009

# XOR operations

Simple XOR seems to be no problem with respect to computation

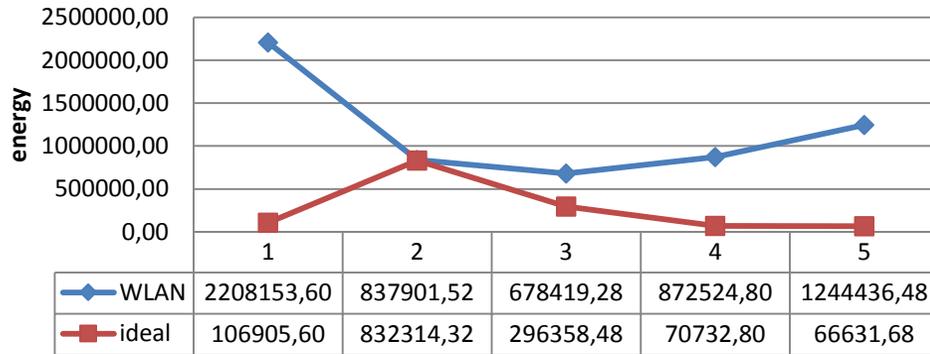
More energy used for the encoding vector to be transmitted than in the actual coding

POWER AND ENERGY LEVELS USED FOR NETWORK CODING ON THE  
MOBILE PHONE

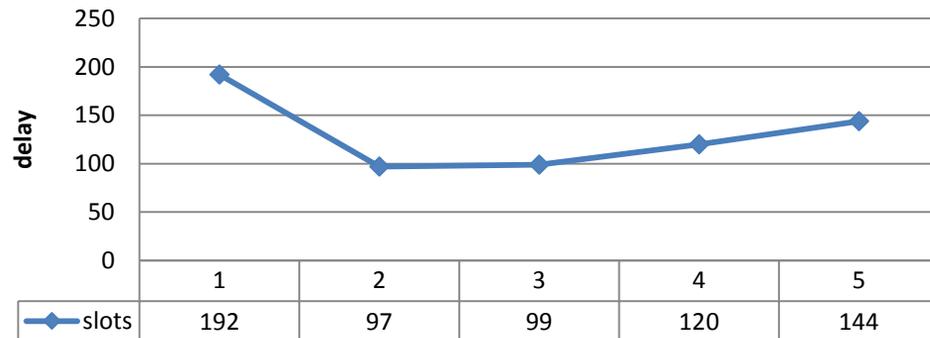
state	power value [W]	energy value [J]	number of operations
coding	0,593	4,789	25000000
coding	0,593	$0,191 \cdot 10^{-6}$	1
idle	0,041	—	—

# Wheel Example: Multi-dimension Alice and Bob

## Energy Plot



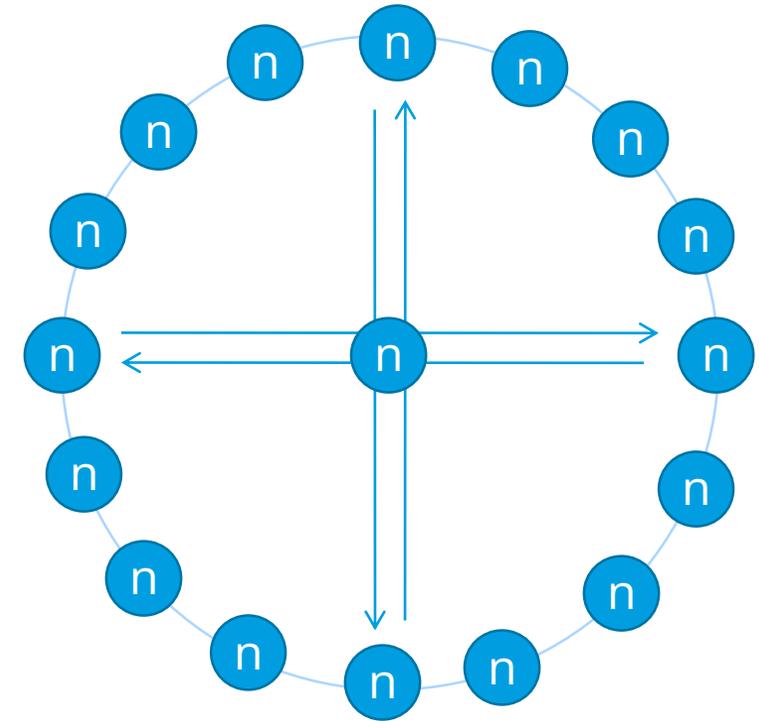
## Delay Plot



Scenario: Wheel

- 97 mobile devices in total
- 96 flows (one per cluster)
- Just relaying to exchange

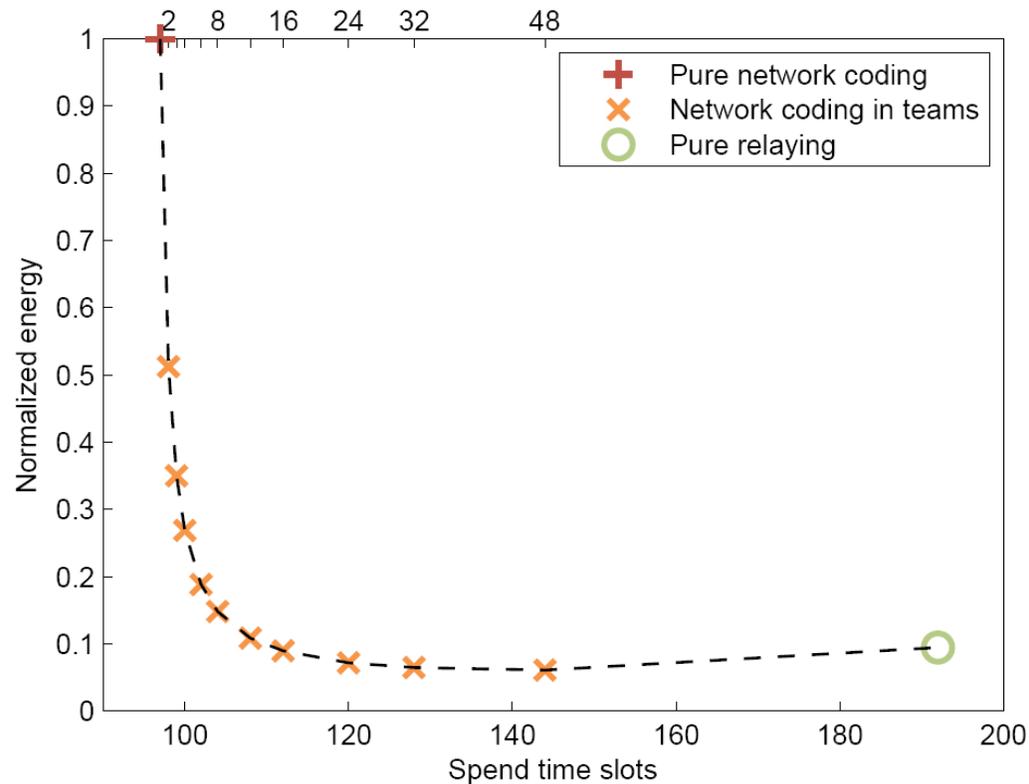
F number of flows  
T number of teams



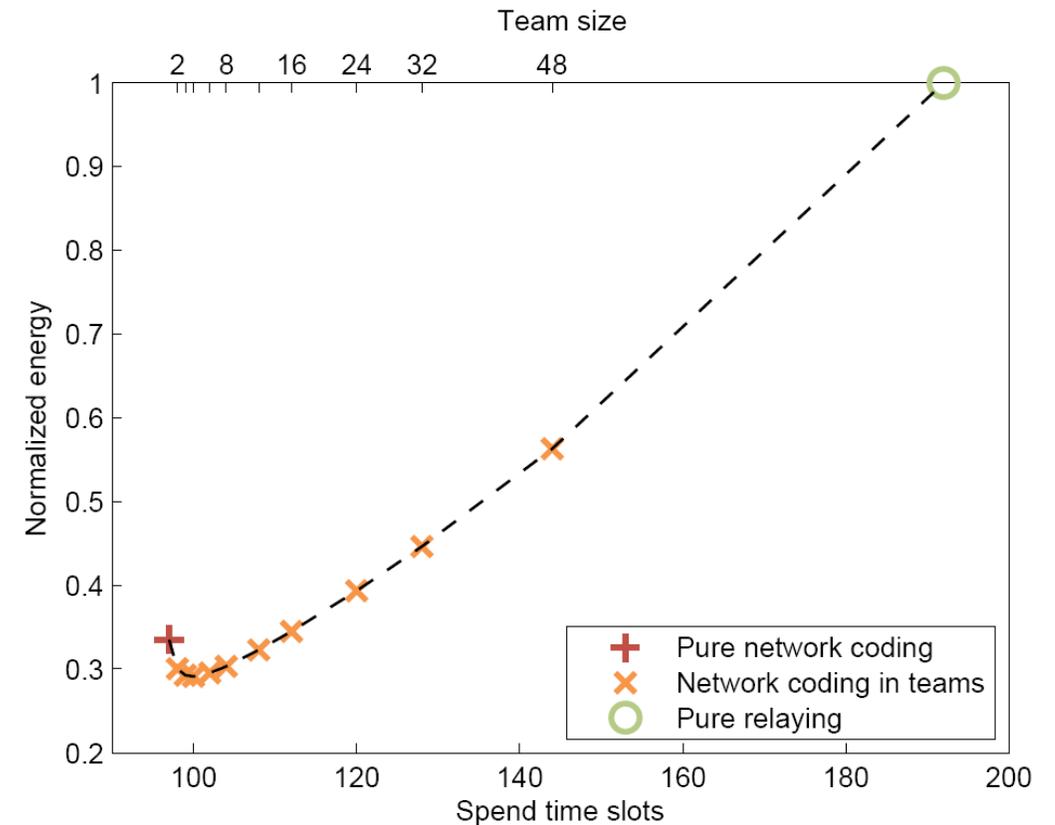
	sending	receiving	idle
Single node	1	$F/T-1$	$F*(T-1)/T + T$
Outer nodes	F	$(F/T-1)F$	$(F(T-1)/T + T)F$
Inner node	T	F	0
<b>SUM netcod</b>	<b>F+T</b>	<b><math>F^2/T</math></b>	<b><math>(F(T-1)/T + T)F</math></b>
<b>SUM relay</b>	<b>2F</b>	<b>2F</b>	<b>2F(F-1)</b>

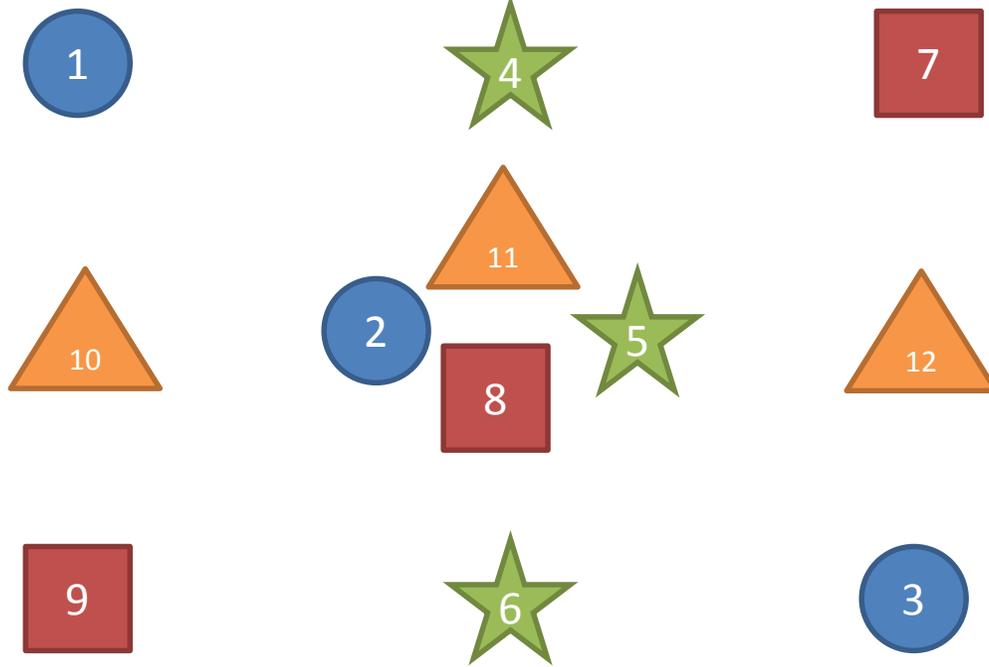
# Wheel Example: Multi-dimension Alice and Bob

*Without considering IDLE power value*



*With considering IDLE power value*

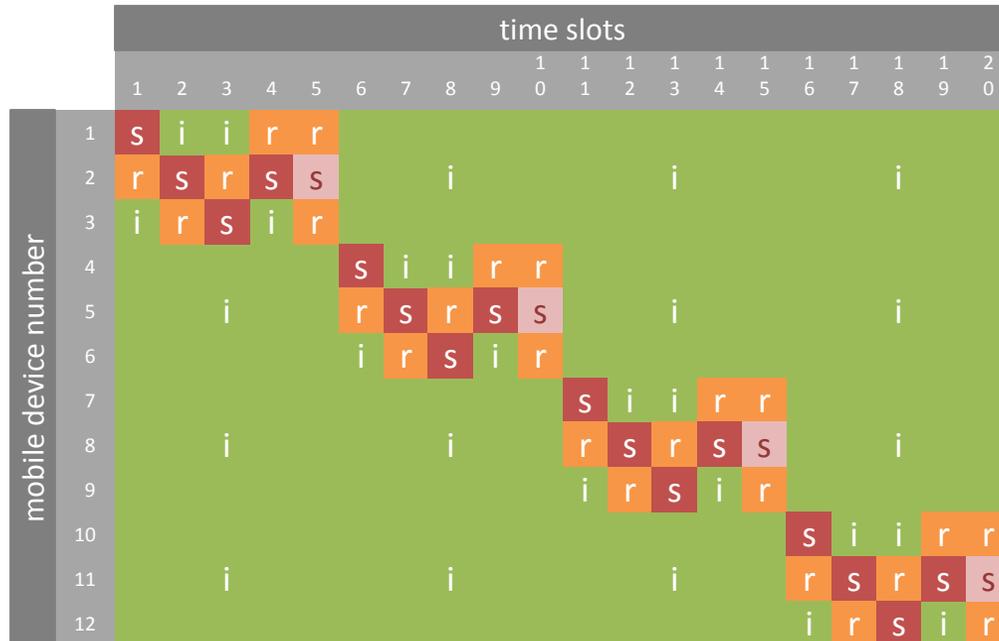




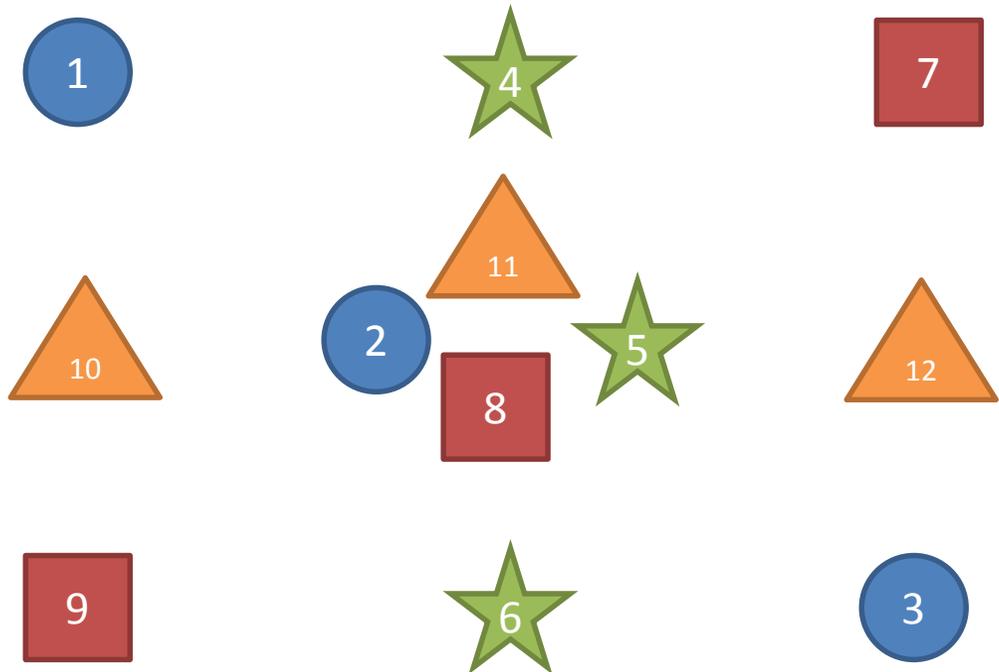
### Scenario: Wheel++

- 12 mobile devices in total
- Four flows (one per cluster)
- Each device receives cellular input
- Just relaying to exchange

i idle slot  
r receiving slot  
s sending unicast slot  
s broadcasting slot  
s broadcasting coded slot



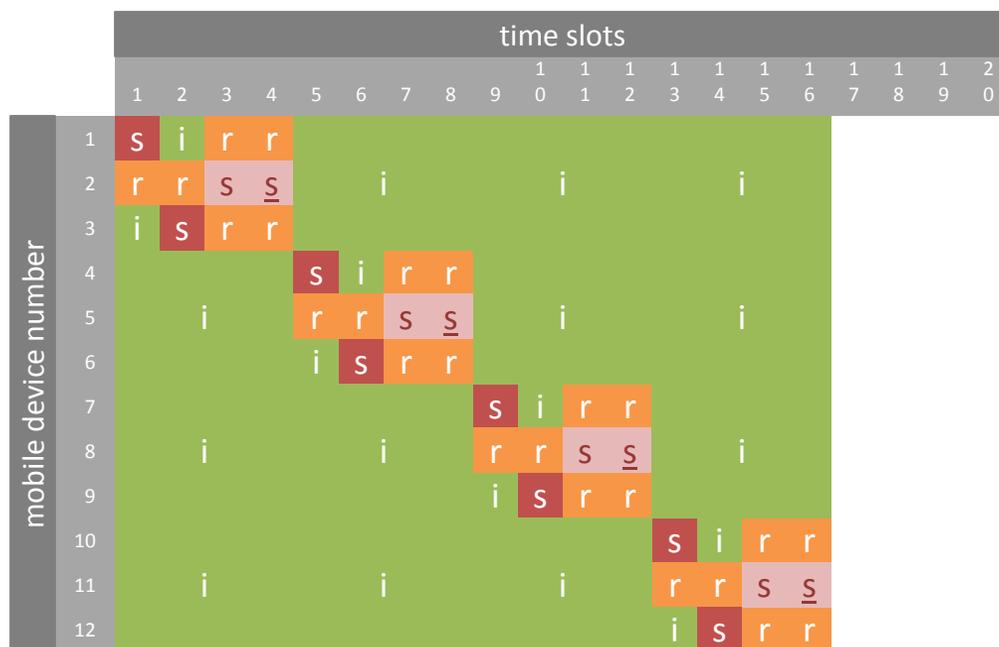
	sending	receiving	idle
One activity matrix	5	6	4
All activity matrixes	20	24	16
All idle matrixes	0	0	180
<b>SUM</b>	<b>20</b>	<b>24</b>	<b>196</b>



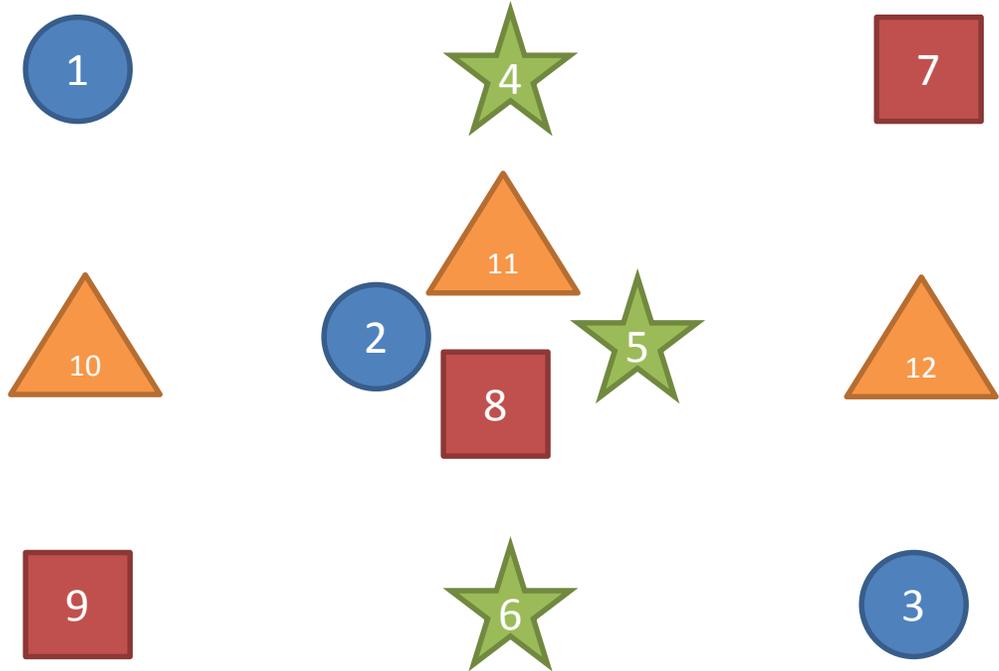
### Scenario: Wheel++

- 12 mobile devices in total
- Four flows (one per cluster)
- Each device receives cellular input
- Network coding within each cluster

i idle slot  
r receiving slot  
s sending unicast slot  
s broadcasting slot  
s broadcasting coded slot



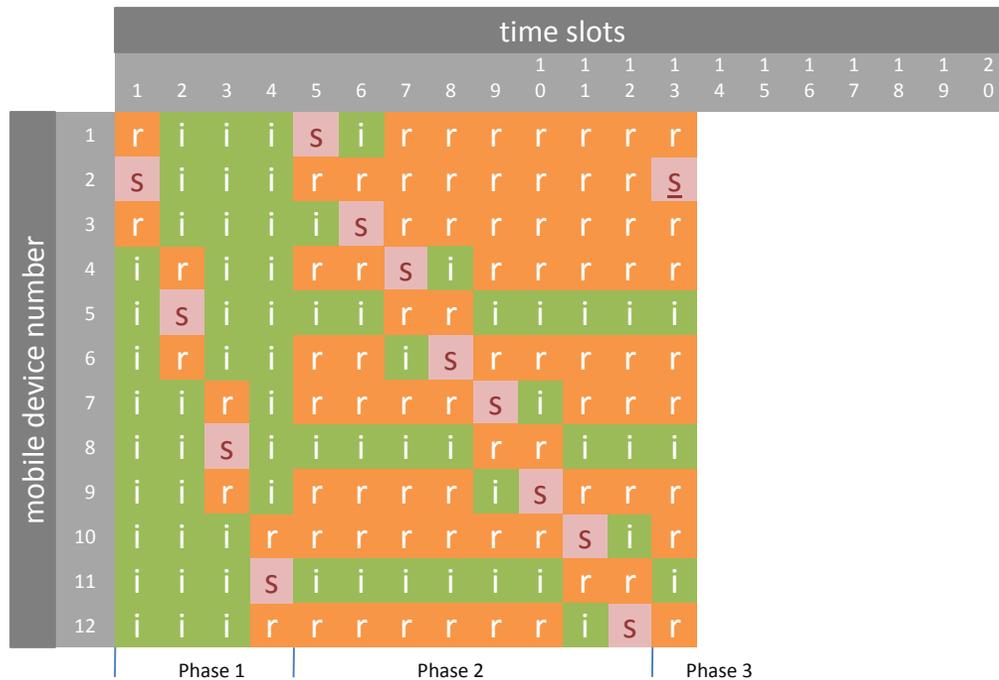
	sending	receiving	idle
One activity matrix	4	6	2
All activity matrixes	16	24	8
All idle matrixes	0	0	144
<b>SUM</b>	<b>16</b>	<b>24</b>	<b>152</b>



### Scenario: Wheel++

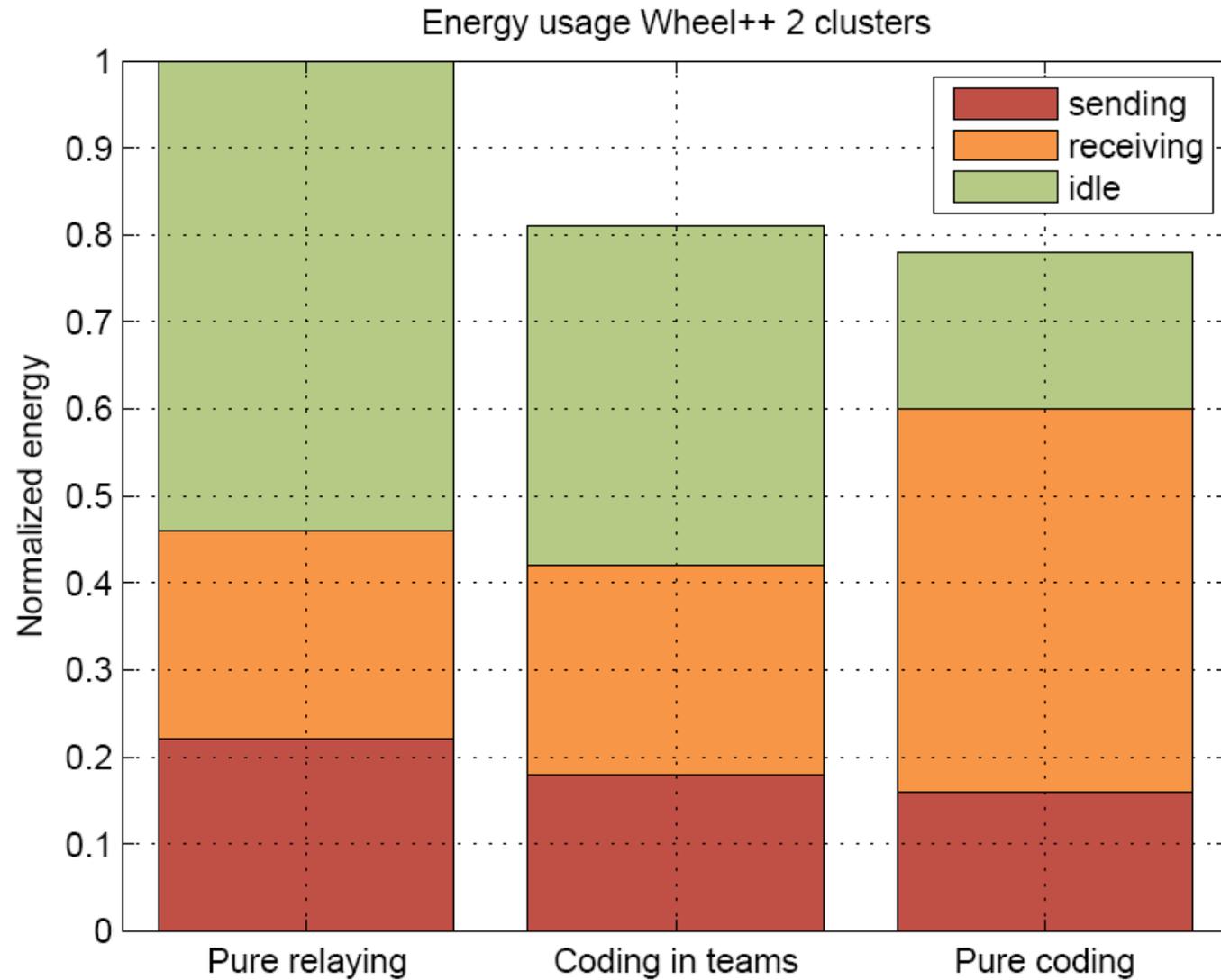
- 12 mobile devices in total
- Four flows (one per cluster)
- Each device receives cellular input
- Network coding over all cluster
- Device 2 is doing the most work

- i idle slot
- r receiving slot
- s sending unicast slot
- s broadcasting slot
- s broadcasting coded slot



	sending	receiving	idle
Phase 1	4	8	36
Phase 2	8	62	26
Phase 3	1	8	3
<b>SUM</b>	<b>13</b>	<b>78</b>	<b>65</b>

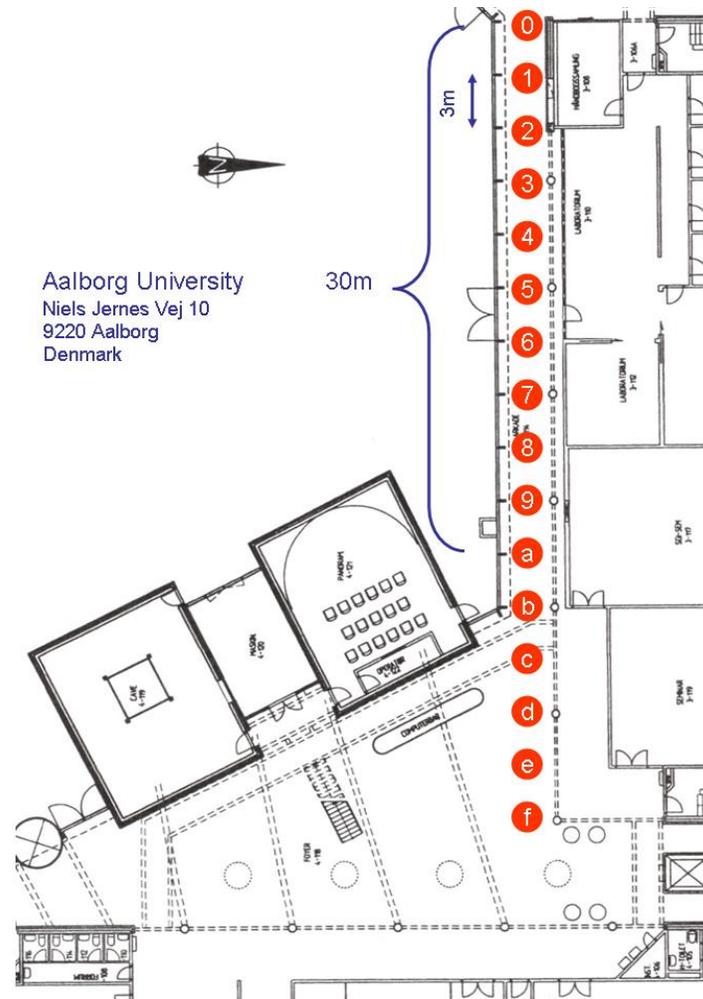
# Results



# Energy and Channel Measurements: Point to point

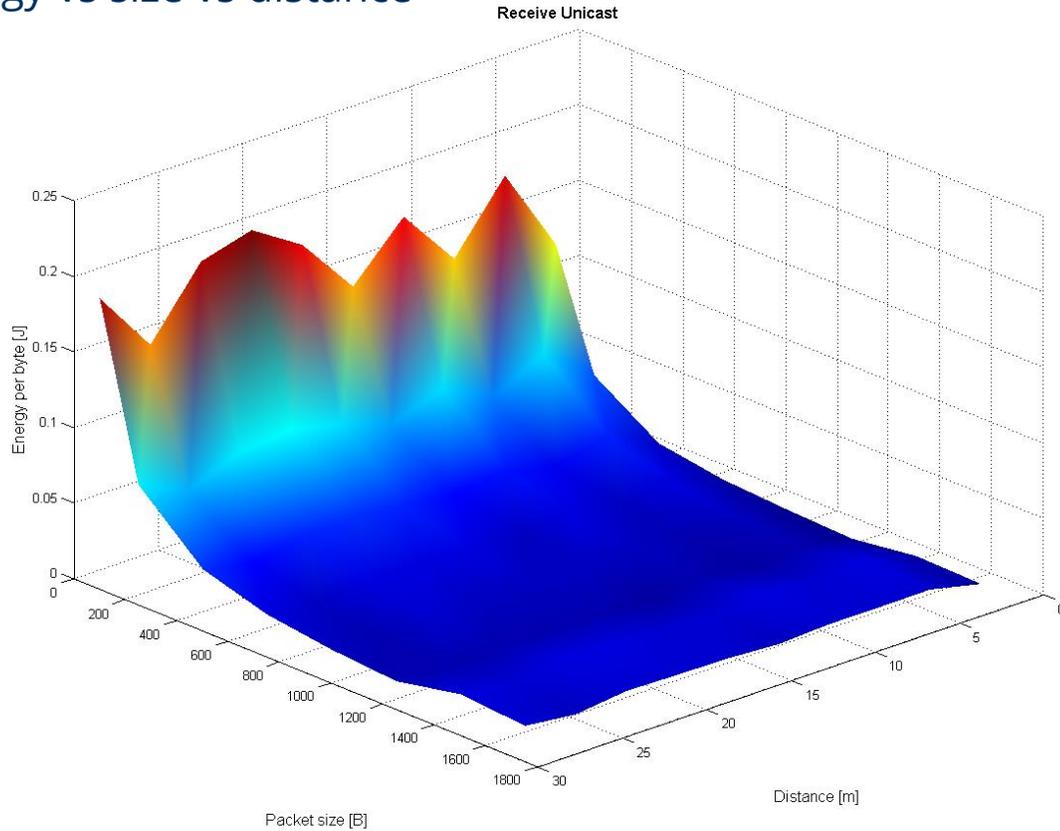
M.V. Petersen and G.P. Perrucci and F.H.P. Fitzek. **Energy and Link Measurements for Mobile Phones using IEEE802.11b/g**. 2008. in *The 4th International Workshop on Wireless Network Measurements (WiNMEE 2008) - in conjunction with WiOpt 2008*. Berlin, Germany.

# Measurement Campaign for NC

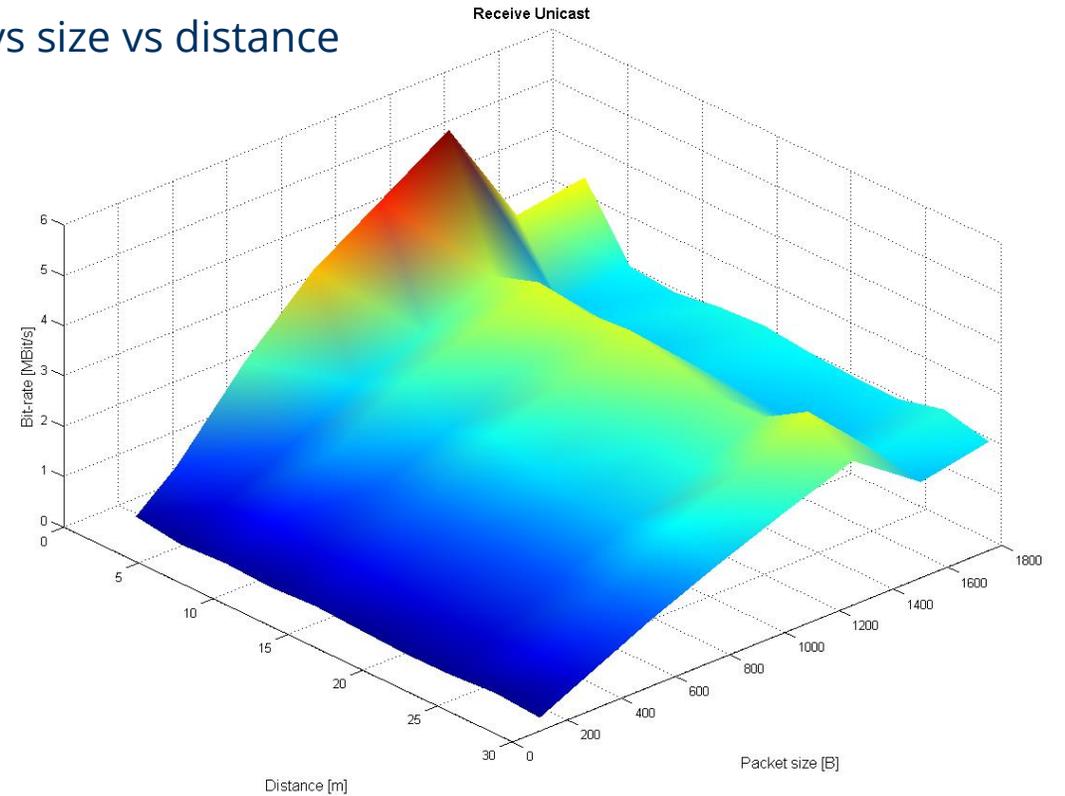


# Unicast Results

Energy vs size vs distance



Data rate vs size vs distance



- Small packets have higher energy per bit ratios due to the MAC overhead
- Small increase for packets larger than MTU size

- Larger data rates for small distances
- Small decrease after MTU sizes