TECHNISCHE UNIVERSITÄT DRESDEN

Netzwerkkodierung in Theorie und Praxis

Praktische Anwendungen der Netzwerkkodierung

Professor Dr.-Ing. Dr. h.c. Frank H.P. Fitzek

M.Sc. Juan Cabrera

Deutsche Telekom Chair of Communication Networks (ComNets)

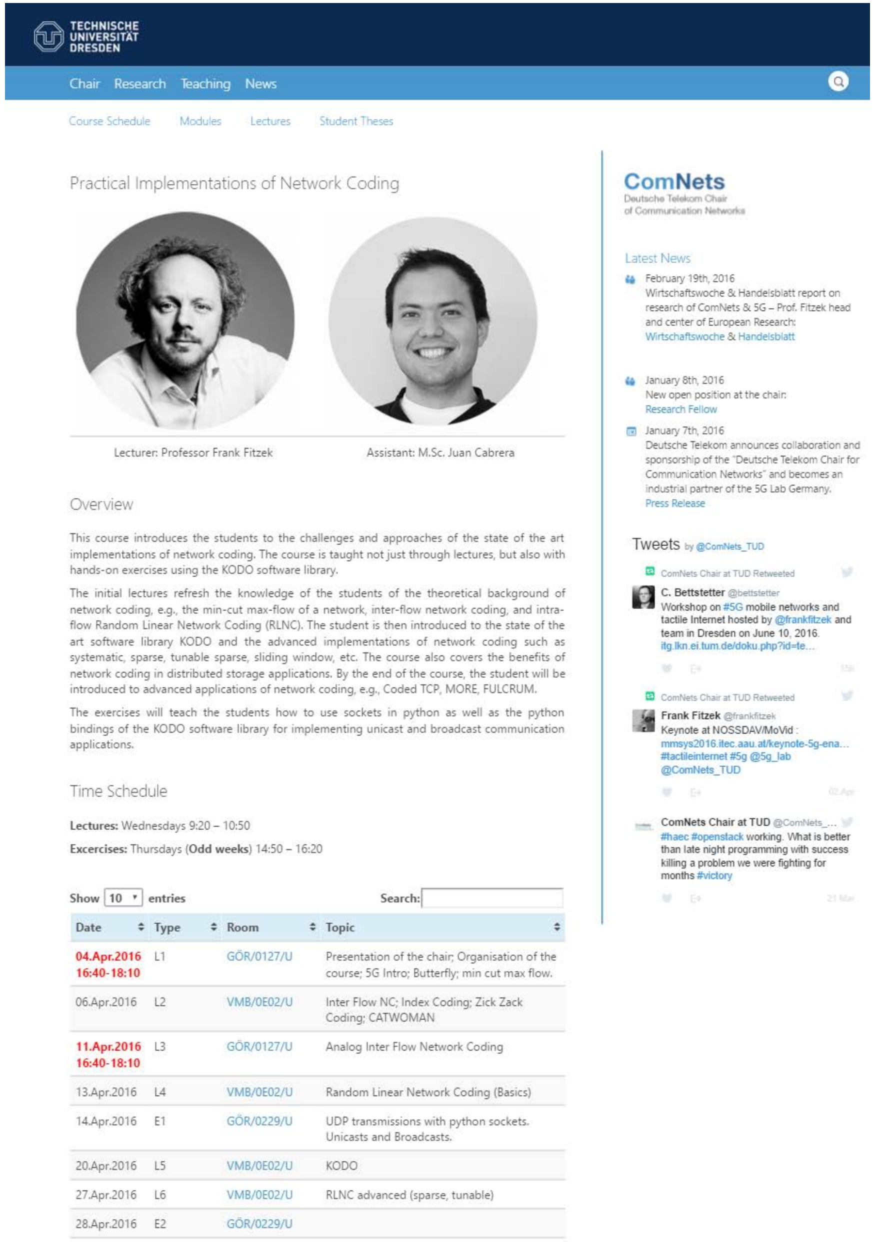
Netzwerkkodierungstheorie

Professor Dr.-Ing. Eduard Jorswieck

Dipl.-Ing. Johannes Richter

Theoretische Nachrichtentechnik

Lecture / Exercise Dates - tinyurl.com/zooafld



Practical Implementations of Network Coding

Lecturer: Professor Frank Fitzek

Assistant: M.Sc. Juan Cabrera

Overview

This course introduces the students to the challenges and approaches of the state of the art implementations of network coding. The course is taught not just through lectures, but also with hands-on exercises using the KODO software library.

The initial lectures refresh the knowledge of the students of the theoretical background of network coding, e.g., the min-cut max-flow of a network, inter-flow network coding, and intra-flow Random Linear Network Coding (RLNC). The student is then introduced to the state of the art software library KODO and the advanced implementations of network coding such as systematic, sparse, tunable sparse, sliding window, etc. The course also covers the benefits of network coding in distributed storage applications. By the end of the course, the student will be introduced to advanced applications of network coding, e.g., Coded TCP, MORE, FULCRUM.

The exercises will teach the students how to use sockets in python as well as the python bindings of the KODO software library for implementing unicast and broadcast communication applications.

Time Schedule

Lectures: Wednesdays 9:20 – 10:50

Excercises: Thursdays (**Odd weeks**) 14:50 – 16:20

Show 10 ▾ entries   Search:

Date	Type	Room	Topic
04.Apr.2016 16:40-18:10	L1	GÖR/0127/U	Presentation of the chair; Organisation of the course; 5G Intro; Butterfly; min cut max flow.
06.Apr.2016	L2	VMB/0E02/U	Inter Flow NC; Index Coding; Zick Zack Coding; CATWOMAN
11.Apr.2016 16:40-18:10	L3	GÖR/0127/U	Analog Inter Flow Network Coding
13.Apr.2016	L4	VMB/0E02/U	Random Linear Network Coding (Basics)
14.Apr.2016	E1	GÖR/0229/U	UDP transmissions with python sockets. Unicasts and Broadcasts.
20.Apr.2016	L5	VMB/0E02/U	KODO
27.Apr.2016	L6	VMB/0E02/U	RLNC advanced (sparse, tunable)
28.Apr.2016	E2	GÖR/0229/U	

ComNets
Deutsche Telekom Chair
of Communication Networks

Latest News

February 19th, 2016
Wirtschaftswoche & Handelsblatt report on research of ComNets & 5G – Prof. Fitzek head and center of European Research:
Wirtschaftswoche & Handelsblatt

January 8th, 2016
New open position at the chair:
Research Fellow

January 7th, 2016
Deutsche Telekom announces collaboration and sponsorship of the "Deutsche Telekom Chair for Communication Networks" and becomes an industrial partner of the 5G Lab Germany.
Press Release

Tweets by @ComNets_TUD

ComNets Chair at TUD Retweeted
C. Bettstetter @bettstetter
Workshop on #5G mobile networks and tactile Internet hosted by @frankfitzek and team in Dresden on June 10, 2016.
itg.lkn.ei.tum.de/doku.php?id=te…

ComNets Chair at TUD Retweeted
Frank Fitzek @frankfitzek
Keynote at NOSSDAV/MoVid :
mmsys2016.itec.aau.at/keynote-5g-ena…
#tactileinternet #5g @5g_lab
@ComNets_TUD
02 Apr

ComNets Chair at TUD @ComNets_…
#haec #openstack working. What is better than late night programming with success killing a problem we were fighting for months #victory
21 Mar

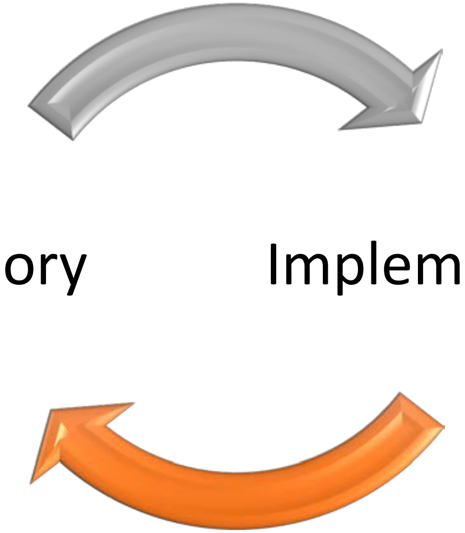
- Here all information for the lecture and the exercise can be found.
- Slides
- Links
  - Steinwurf
  - Python
  - KODOMARK (google play)

Please check every week!

Aim of this lecture module

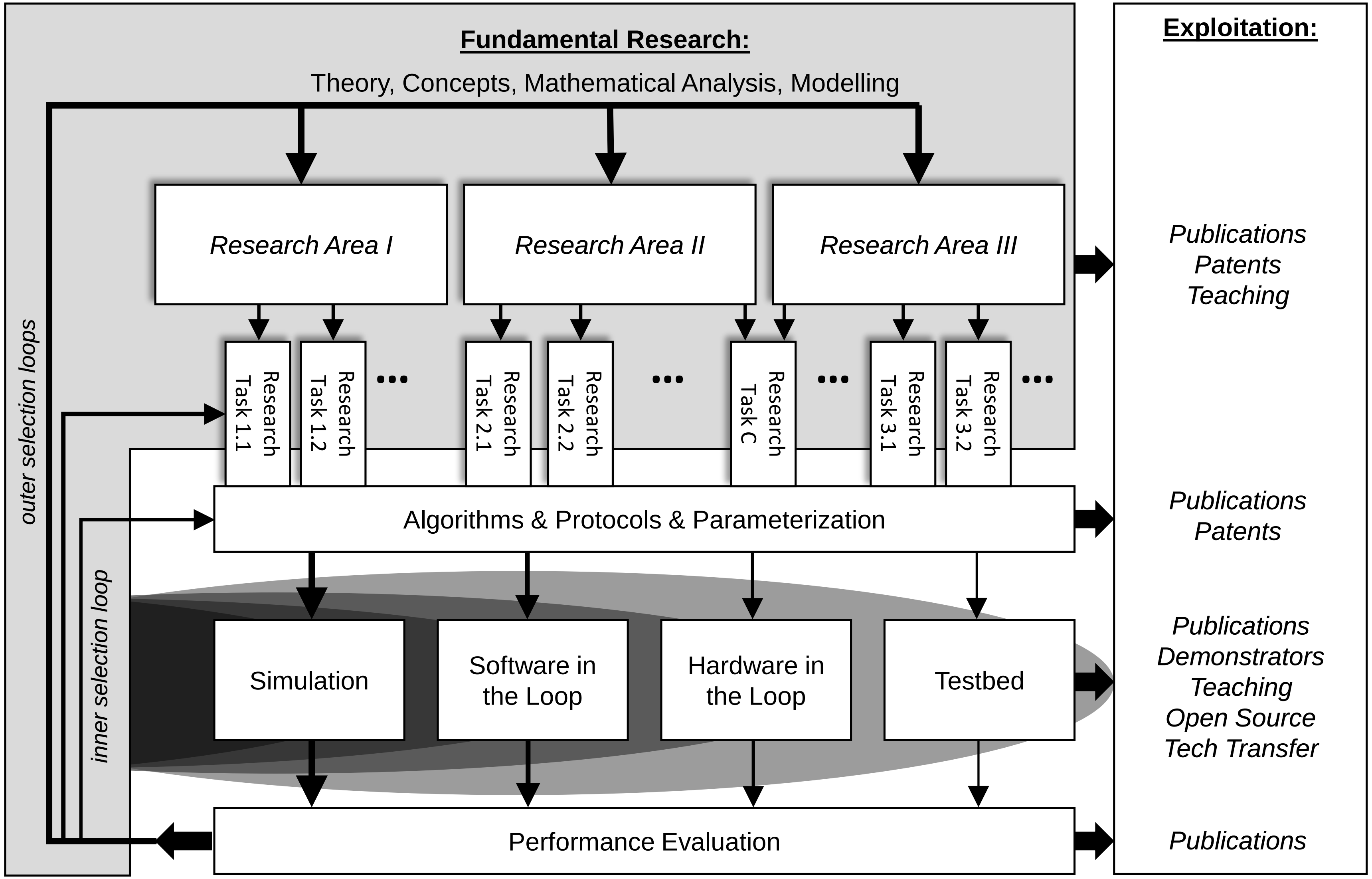
- Explain network coding in theory and practice
- Explain the uniqueness of network coding
- Describe wide range of application of NC in current and future communication systems
  - 5G
  - Storage as well as transportation
- Important is the “hands on” part aligned with theory
  - Please bring your laptop to all lectures and exercises
    - Preinstall software needed
    - Get KODO license from steinwurf.com

Research Methodology: Theory that matters!



Theory

Implementation



Fundamental Research:

Theory, Concepts, Mathematical Analysis, Modelling

Exploitation:

Publications
Patents
Teaching

Publications
Patents

Publications
Demonstrators
Teaching
Open Source
Tech Transfer

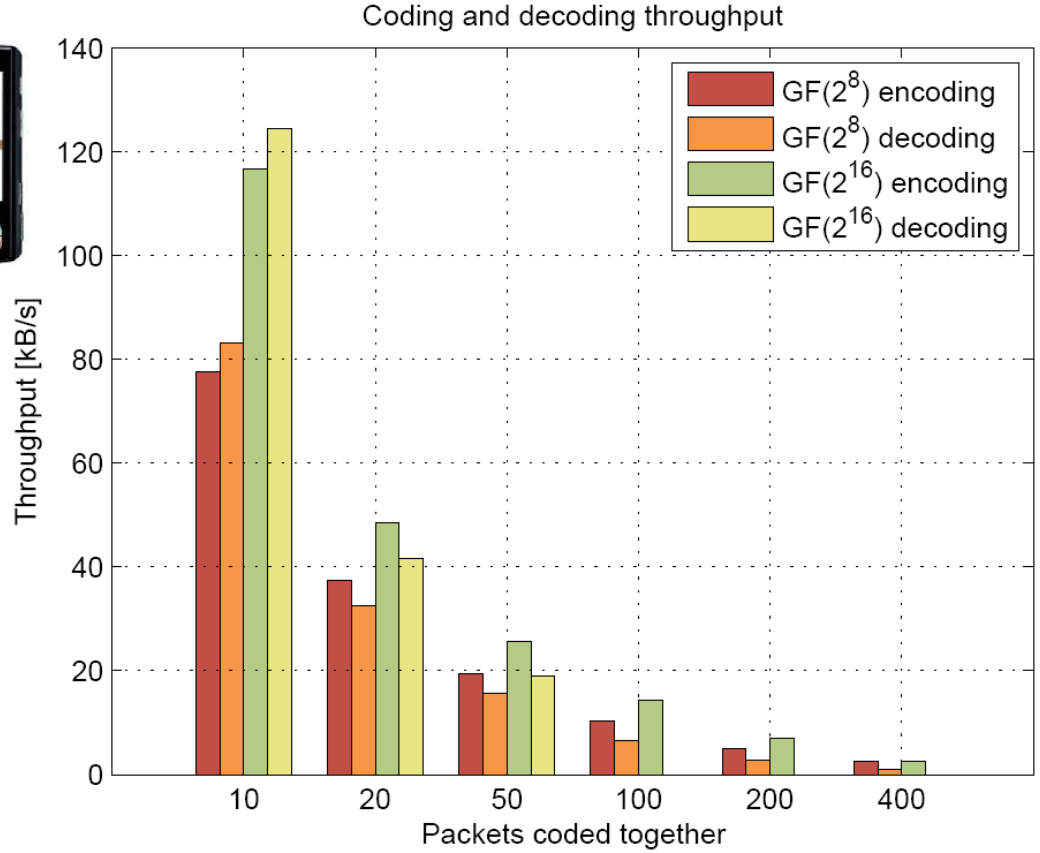
Publications

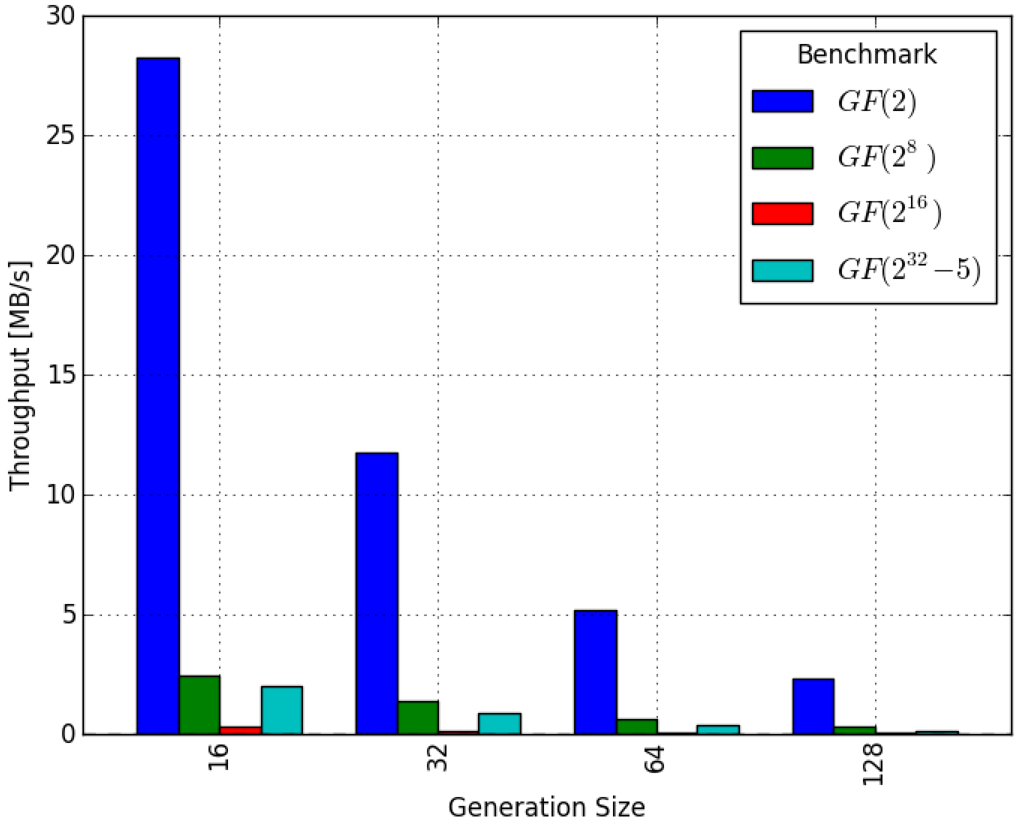
Research Methodology: Example

2007: 120kB/s

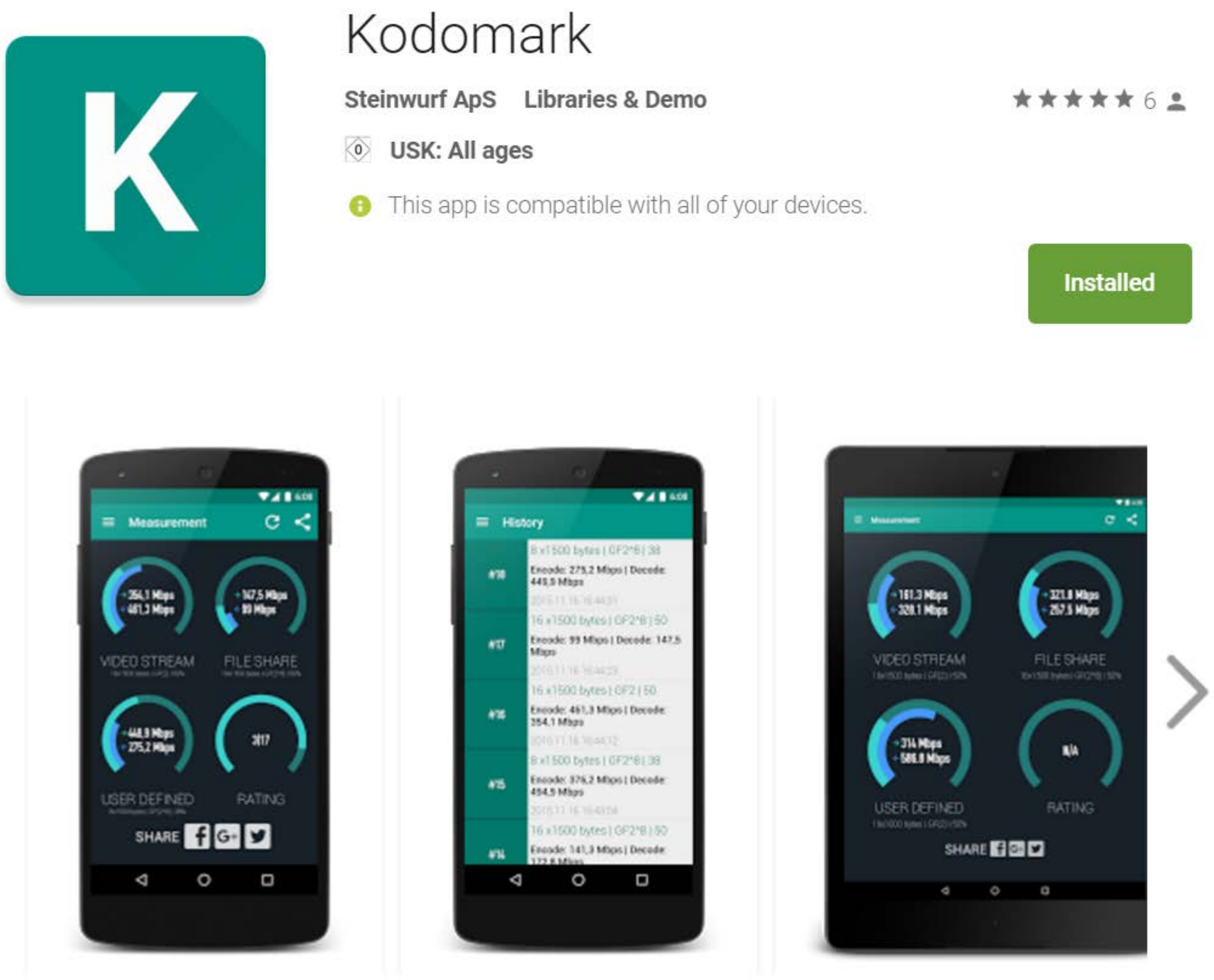
>200x

2012: 27 MB/s





How fast are we now? http://tinyurl.com/z7vsp4c



Kodomark

Steinwurf ApS Libraries & Demo

★★★★★ 6

USK: All ages

This app is compatible with all of your devices.

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.

Preparation

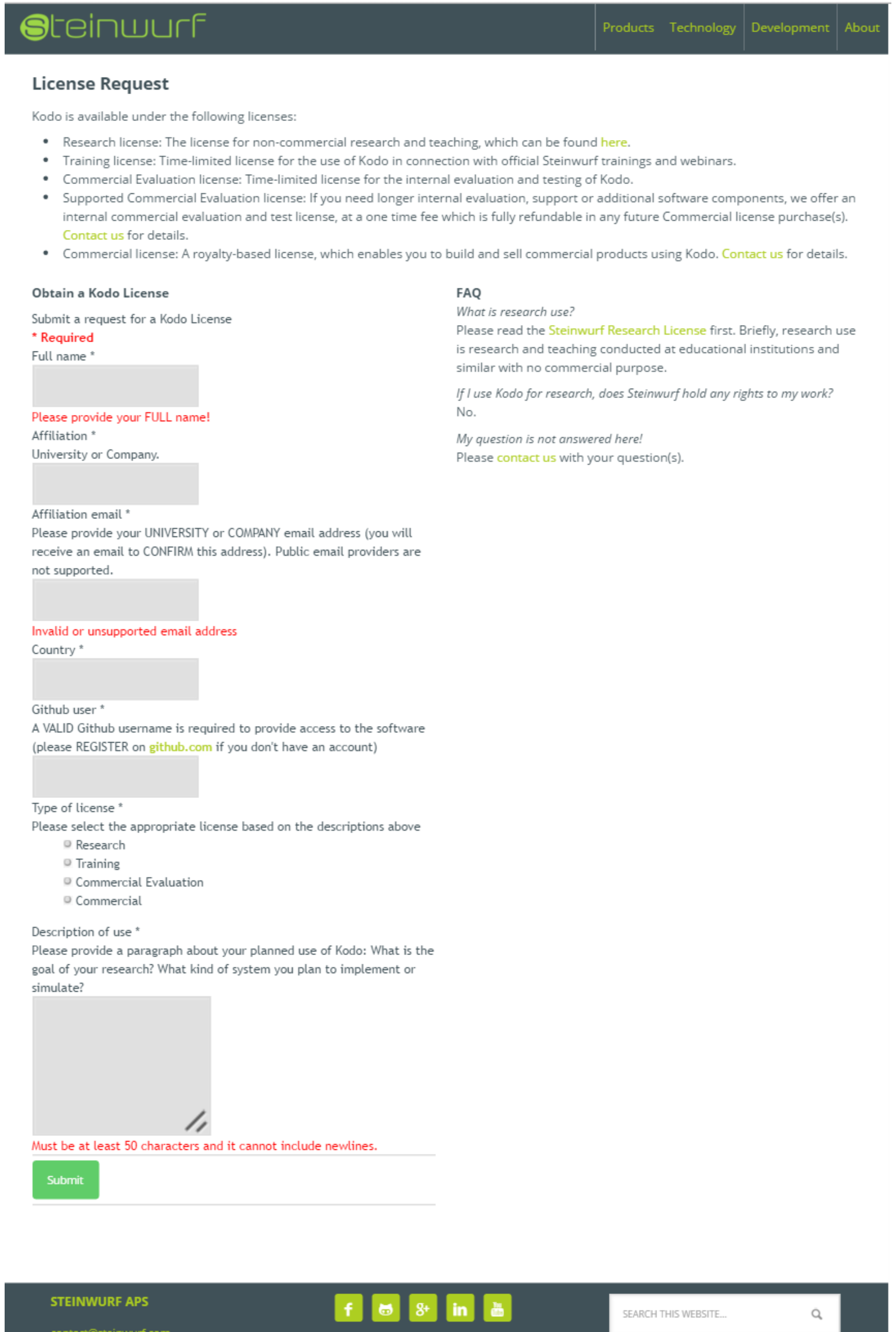
Kodo

Network Coding Software for Mobile Devices
void decode(uint8_t *symbol_data, uint8_t
seed_id id = sak::big_endian::get<seed_id>

Kodo

- Random linear network coding library – Kodo
- Research license for free
- Use your TUD email and you will get access – add “participating lecture of Fitzek/Jorswieck at TUD”
- Use your TUD email account
- You need a github account

http://steinwurf.com/license/



License Request

Kodo is available under the following licenses:

- Research license: The license for non-commercial research and teaching, which can be found here.
- Training license: Time-limited license for the use of Kodo in connection with official Steinwurf trainings and webinars.
- Commercial Evaluation license: Time-limited license for the internal evaluation and testing of Kodo.
- Supported Commercial Evaluation license: If you need longer internal evaluation, support or additional software components, we offer an internal commercial evaluation and test license, at a one time fee which is fully refundable in any future Commercial license purchase(s). Contact us for details.
- Commercial license: A royalty-based license, which enables you to build and sell commercial products using Kodo. Contact us for details.

Obtain a Kodo License

Submit a request for a Kodo License

* Required

Full name *

Please provide your FULL name!

Affiliation *

University or Company.

Affiliation email *

Please provide your UNIVERSITY or COMPANY email address (you will receive an email to CONFIRM this address). Public email providers are not supported.

Invalid or unsupported email address

Country *

Github user *

A VALID Github username is required to provide access to the software (please REGISTER on github.com if you don't have an account)

Type of license *

Please select the appropriate license based on the descriptions above

○ Research
○ Training
○ Commercial Evaluation
○ Commercial

Description of use *

Please provide a paragraph about your planned use of Kodo: What is the goal of your research? What kind of system you plan to implement or simulate?

Must be at least 50 characters and it cannot include newlines.

Submit

FAQ

What is research use?

Please read the Steinwurf Research License first. Briefly, research use is research and teaching conducted at educational institutions and similar with no commercial purpose.

If I use Kodo for research, does Steinwurf hold any rights to my work?

No.

My question is not answered here!

Please contact us with your question(s).

Python

Python

- Python is a “high level” interpreted programming language
- It is known to be easy to learn and use
- It works on many different “platforms”
- Python is free
- It is very versatile
  - Can be used for large scale systems (e.g. youtube)
  - Can be used for small scripts (e.g. run this command on all these files)
- “Batteries included” huge standard library for many common operations.

Setting up Initial Development Environment

- On your computer go to:
  - http://www.python.org/download/
- Download the "Python 2.7 xxx" matching your operating system. If in doubt (and on Windows) used “Python 2.7 Windows x86 MSI Installer”
- Run the installation file
- This will install Python on your computer and a small shell for running Python programs
- We are now ready to start :)

Python – Additional libraries

- Several helpful libraries are available
- Network Coding Library Kode-Python
- We will use those platforms to build our own network coding enabled devices
- Later some slides how to setup the installation.

Python references

- https://docs.python.org/3/tutorial/index.html
- http://www.tutorialspoint.com/python/index.htm

Now let’s get started

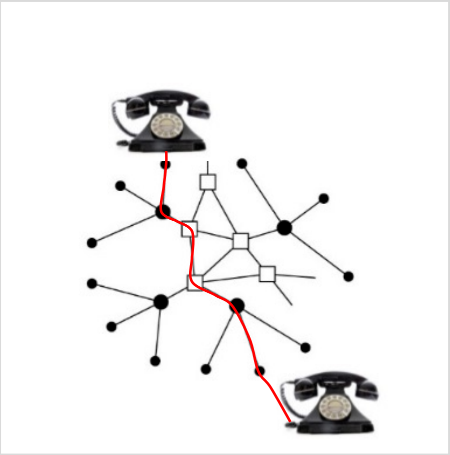
Content

- (R)evolution of communication networks
- Coding in General: Channel and Source Coding: Transport vs. Storage: History
- Butterfly
- Butterfly++
- Index Coding
- Two Way Relay
- X w overhearing
- Cross w and w/o overhearing
- Rate System

(R)evolution of communication networks

Communication Networks

Circuit Switched Networks

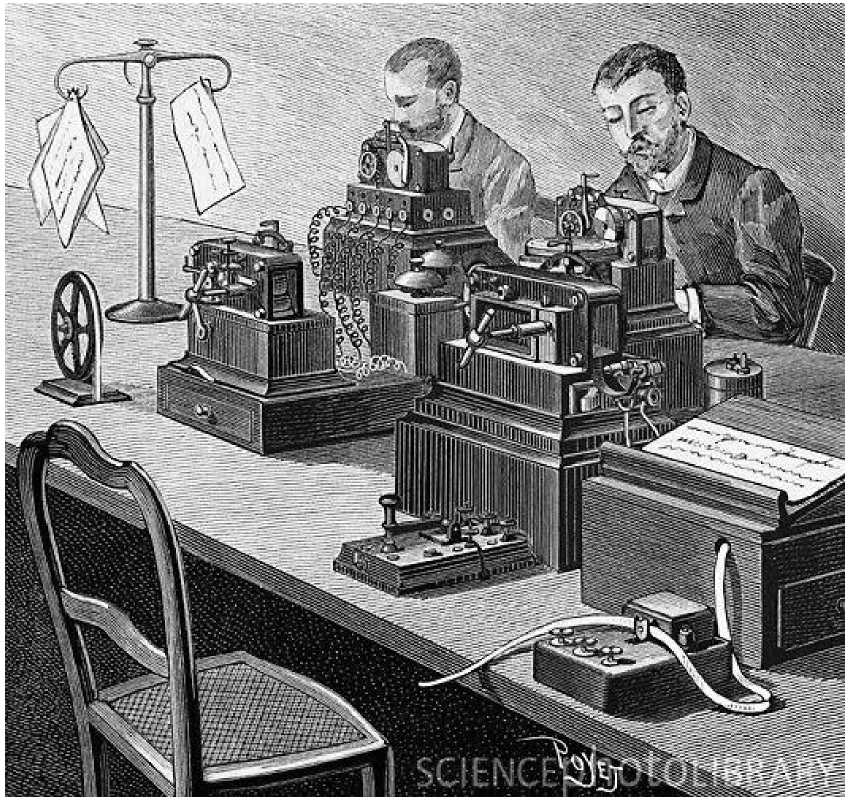


Voice

Places

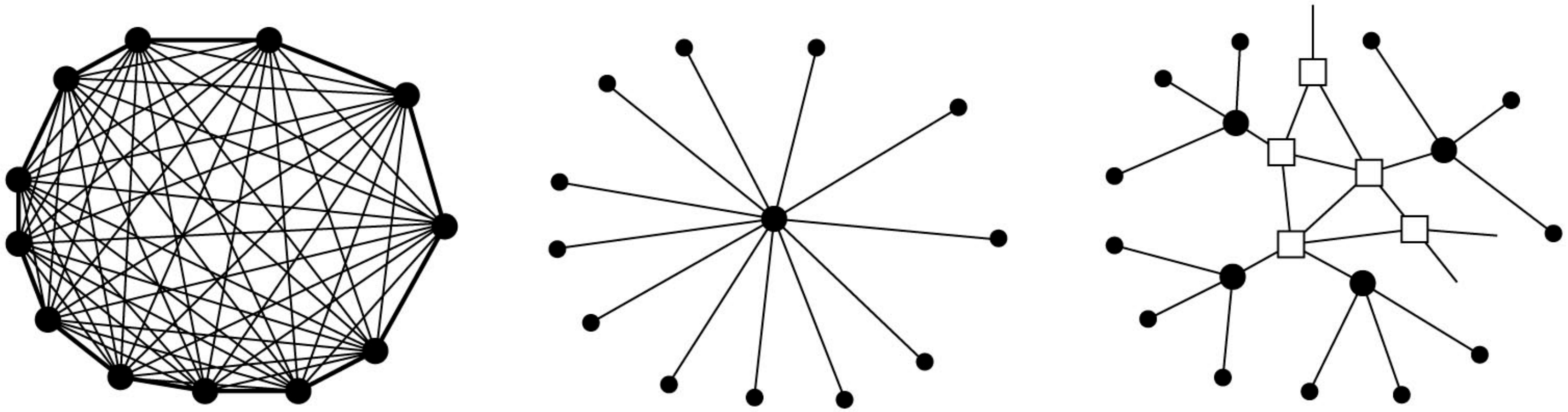
The Telegraph System

- Point to Point links
- Text oriented
- Paddington station to West Drayton in 1839

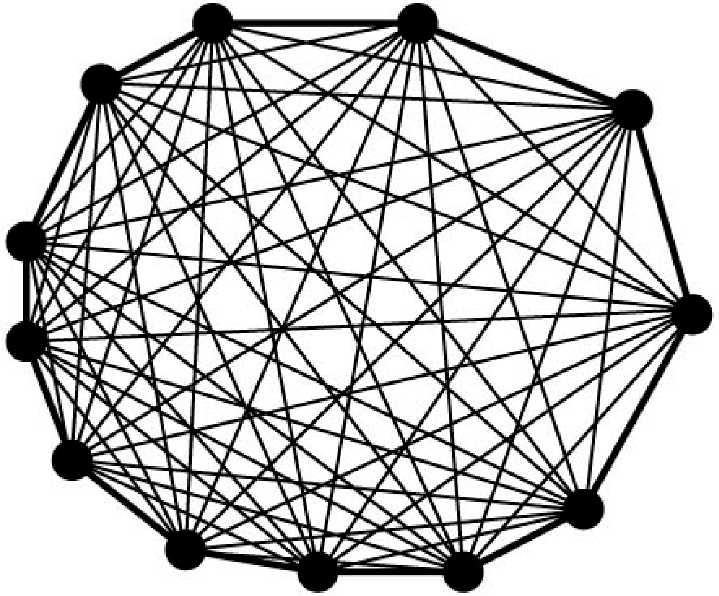


The Telephone System

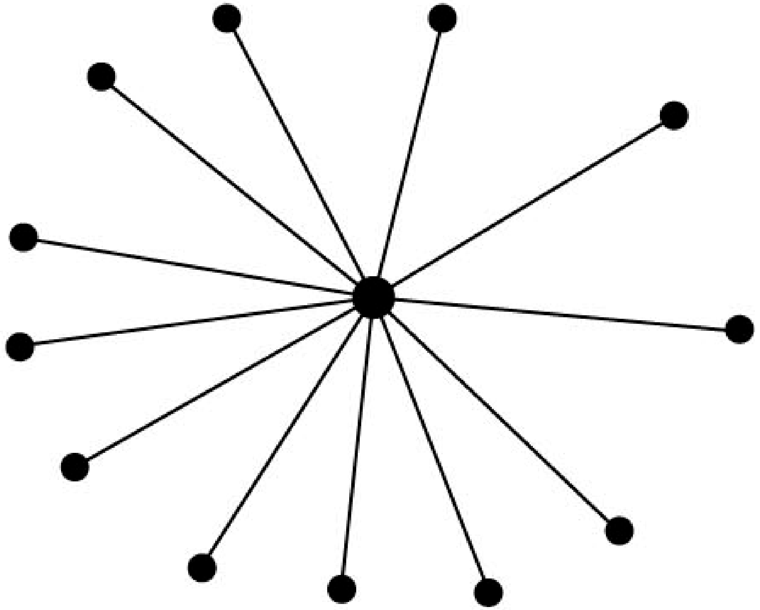
- Voice oriented
- Starting in 1876 and onwards
- One line per communication partner
- Later circuit switched



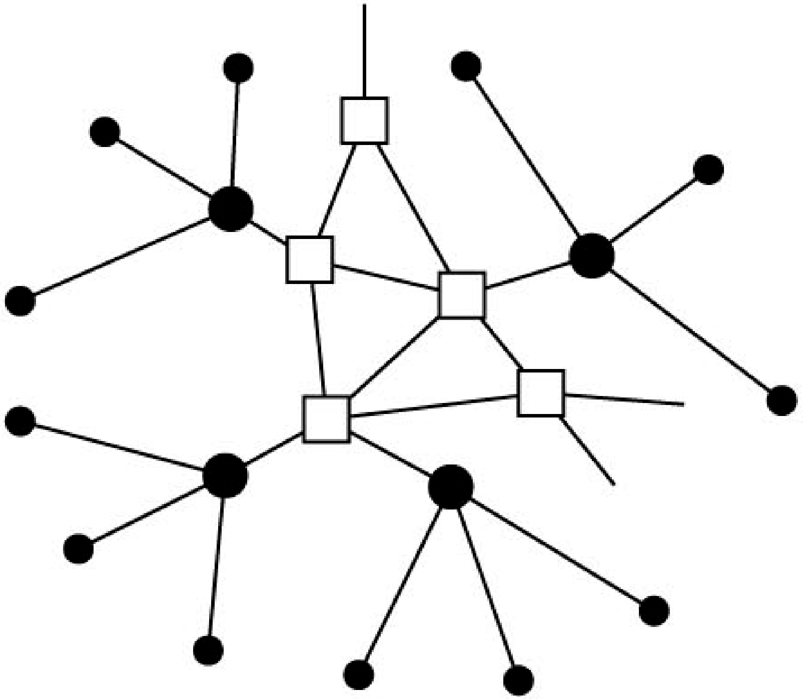
The Telephone System

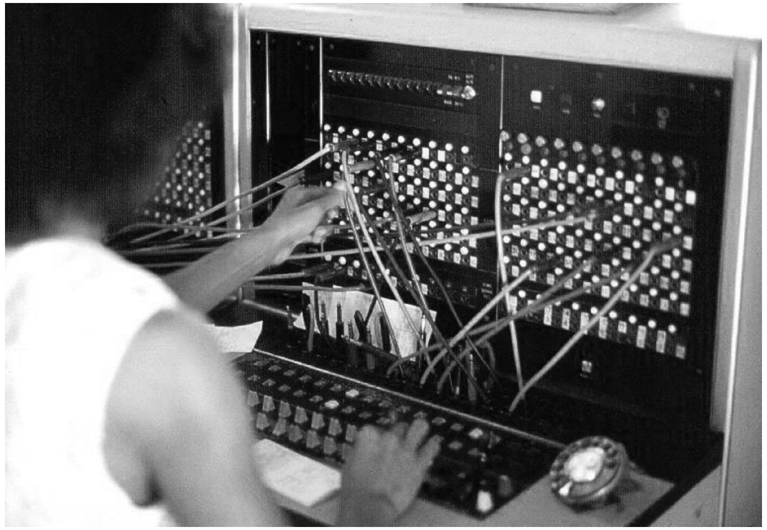


The Telephone System



The Telephone System

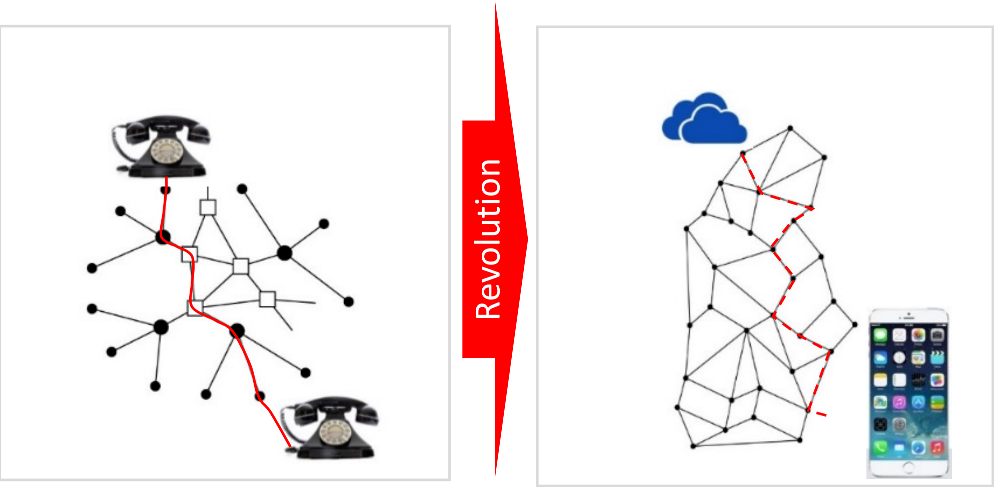




Communication Networks

***Circuit* Switched Networks**

***Packet* Switched Networks**



Voice

Places

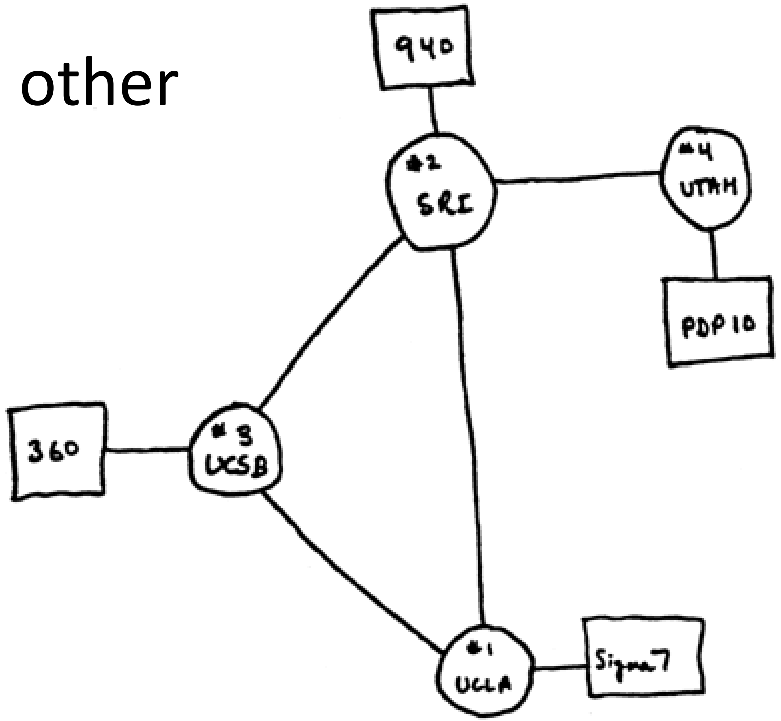
Voice

Data

People

Packet Switched Networks

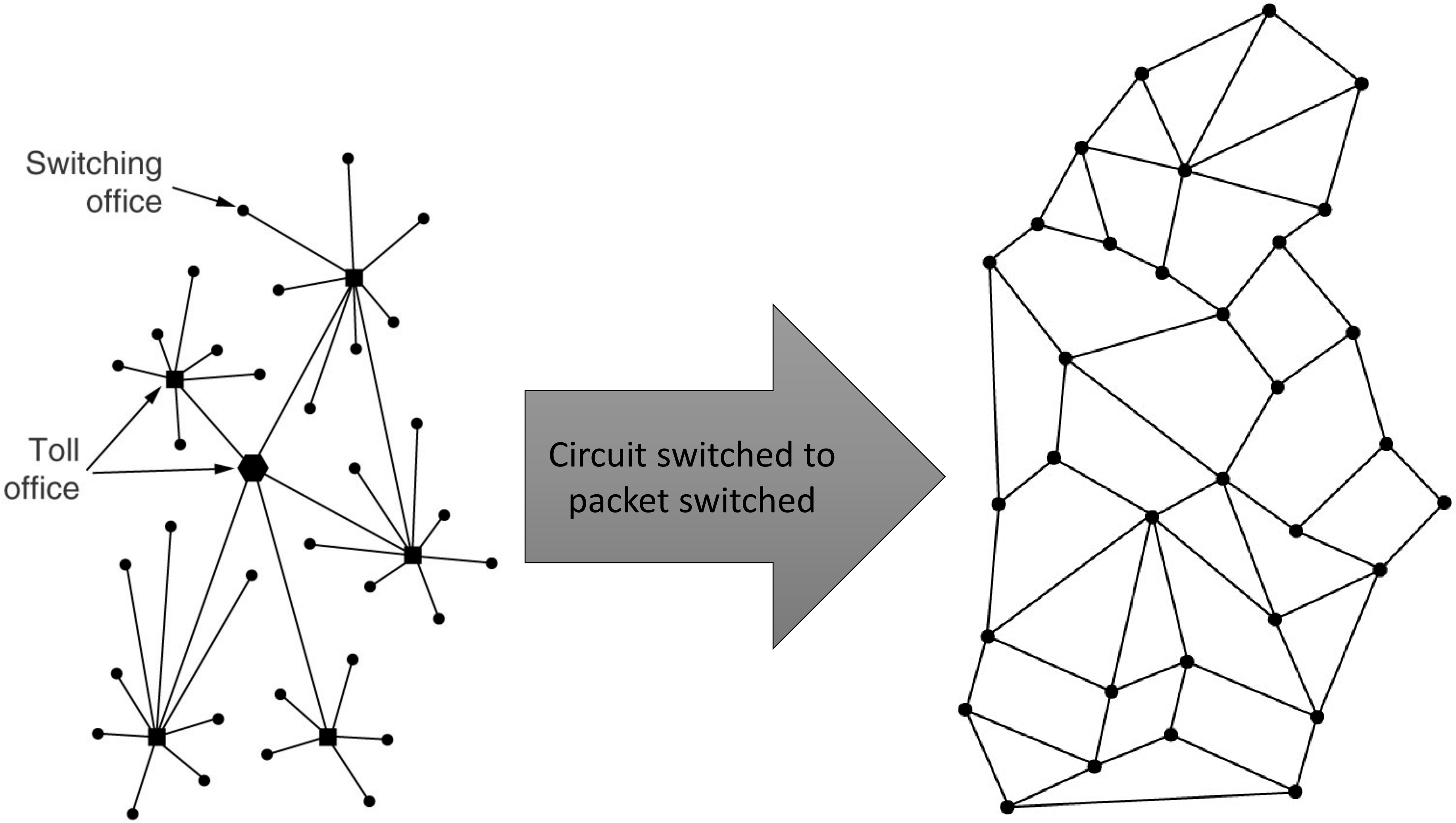
- 1960s: Some experiments with connecting computers at MIT
- 1962: Licklider coins the Intergalactic Computer Network
- 1962: Leonard Kleinrock* completes his doctoral dissertation at MIT on queuing theory in communication networks (now with UCLA)
- 1964: Paul Baran writes 11 chapters on “On Distributed Communications Series”
- 1969: Four institutions selected to connect to each other
  - University of Los Angeles (UCLA)
  - Stanford Research Institute (SRI)
  - University of California, Santa Barbara (UCSB)
  - UTAH
- 1969: First Request For Comments by UCLA team
- 1969: First login on another computer



* forward to our Mobile Cloud book

The Internet

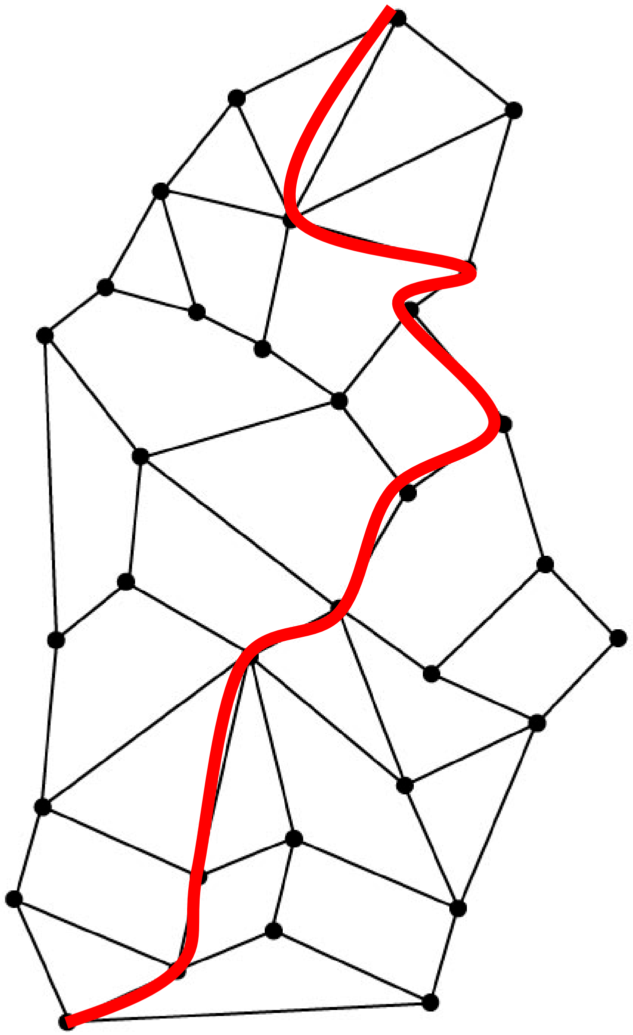
- Paul Baran 1969



Circuit switched to packet switched

The Internet

- Multiple Service
- Packets do not have to follow a given route and can change the route on the fly
- In practise single path communication
- Not good for security
- Not exploiting full potential of the network



Quelle: japantimes.co.jp/news/2014/09/30/asia-pacific/hong-kong-democracy-protesters-set-deadline-for-demands/

7

Billion Devices

2014

Quelle: japantimes.co.jp/news/2014/09/30/asia-pacific/hong-kong-democracy-protesters-set-deadline-for-demands/

7

Billion Devices

2014

500

Billion Devices

2022

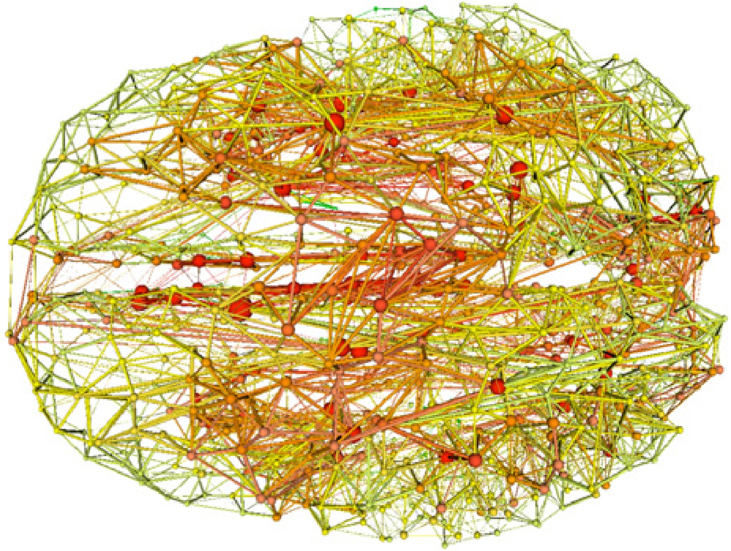
Quelle: japantimes.co.jp/news/2014/09/30/asia-pacific/hong-kong-democracy-protesters-set-deadline-for-demands/

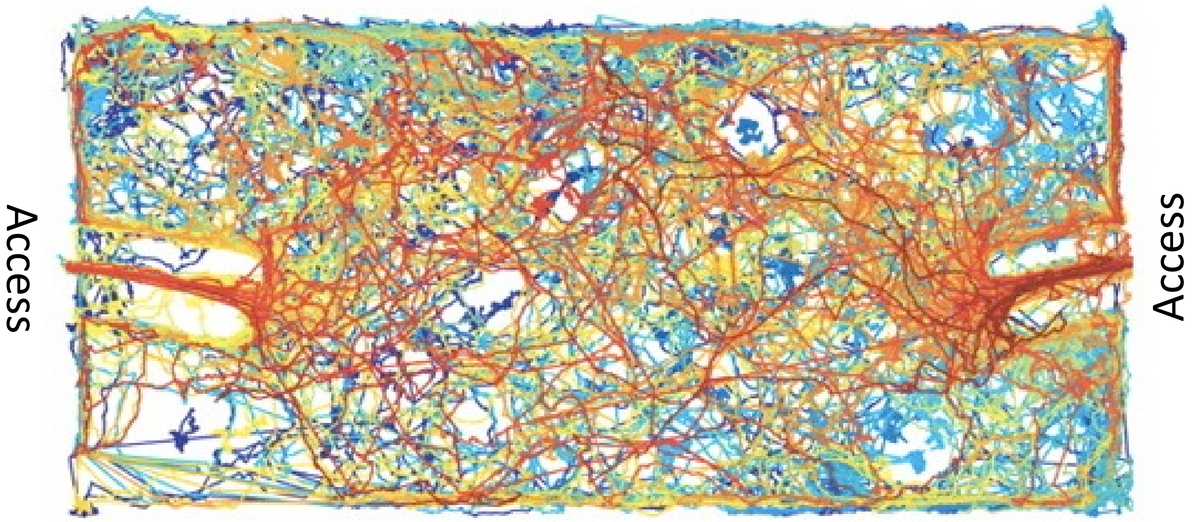
5G Use Cases

Use Cases

Single Path vs. Multi Path

- Comparison with the **brain**
- Our brain uses multi paths
    - Reliability (Pain)

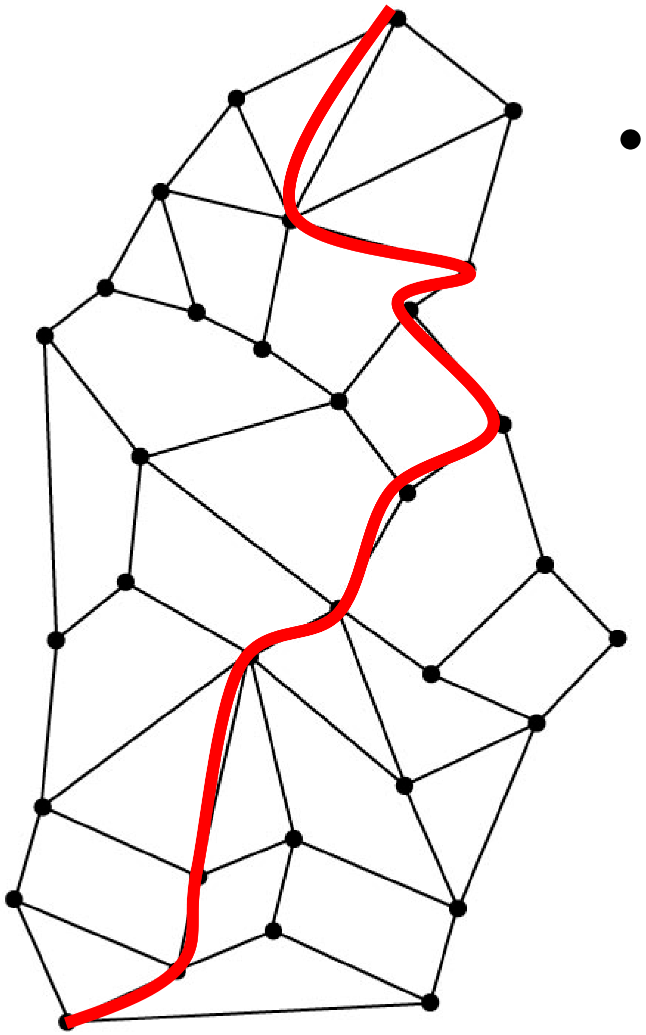




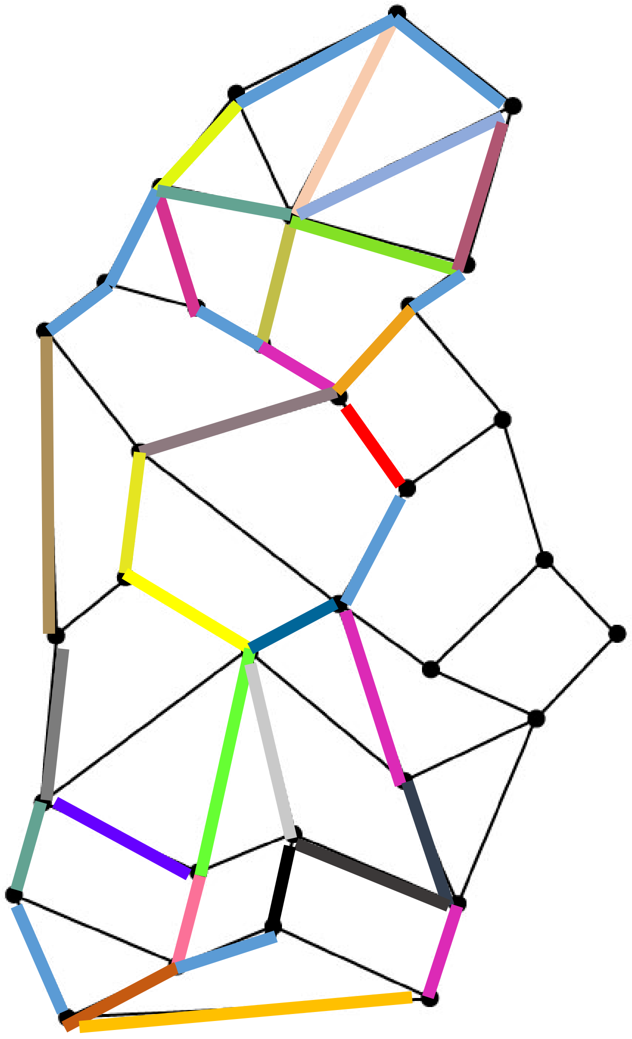
- Comparison with **ants**
- Food retrieval strategies

The Coded Internet

- Wireless meshed networks
  - IoT/M2M/D2D
- Storage and cloud services



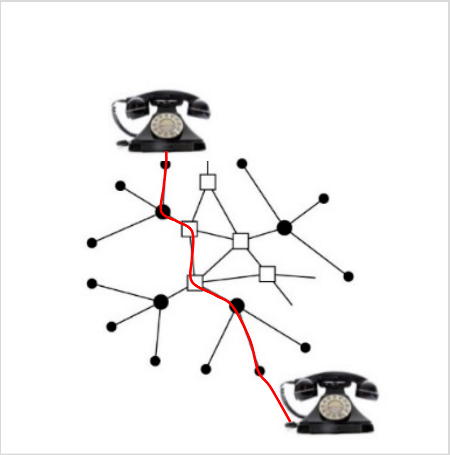
Packet switched to coded ...



- Throughput
- Reliability
- Delay
- Security
- Complexity

Communication Networks

***Circuit* Switched Networks**

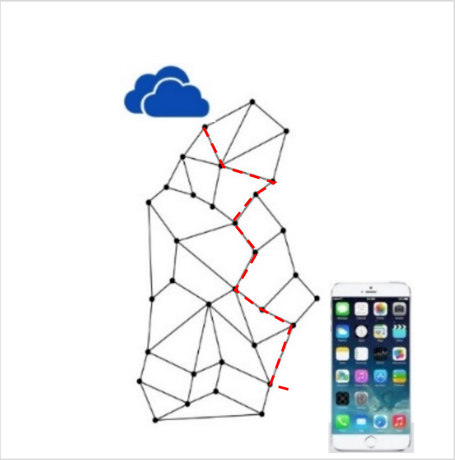


Voice

Places

Revolution

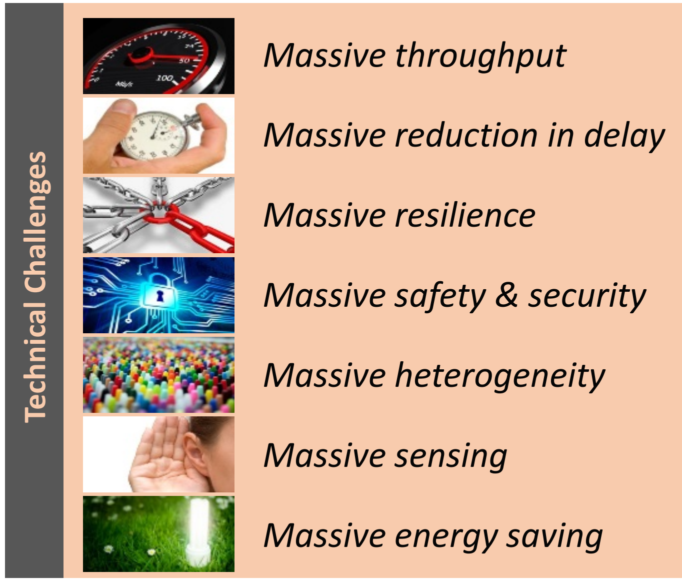
***Packet* Switched Networks**



Voice

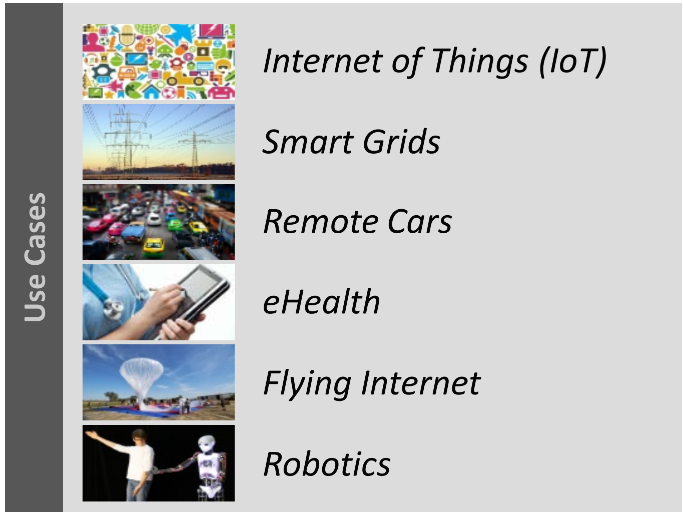
Data

People



Technical Challenges

- *Massive throughput*
- *Massive reduction in delay*
- *Massive resilience*
- *Massive safety & security*
- *Massive heterogeneity*
- *Massive sensing*
- *Massive energy saving*

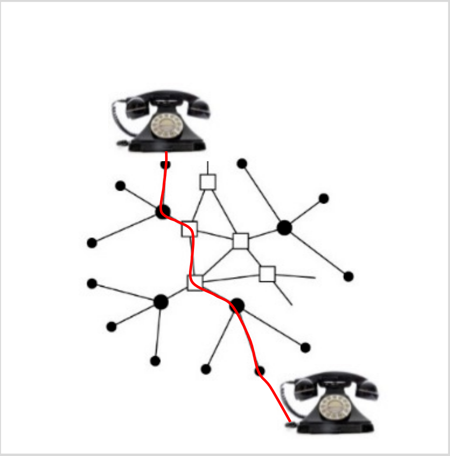


Use Cases

- *Internet of Things (IoT)*
- *Smart Grids*
- *Remote Cars*
- *eHealth*
- *Flying Internet*
- *Robotics*

Communication Networks

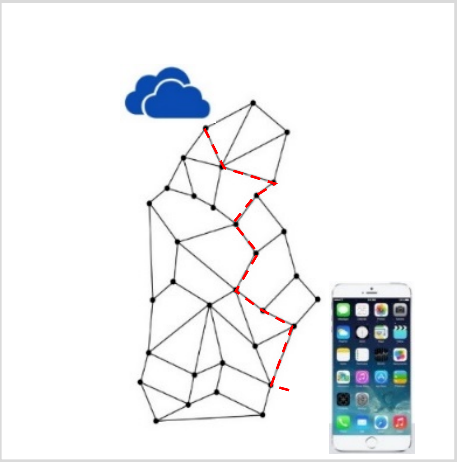
***Circuit** Switched Networks*



Voice

Places

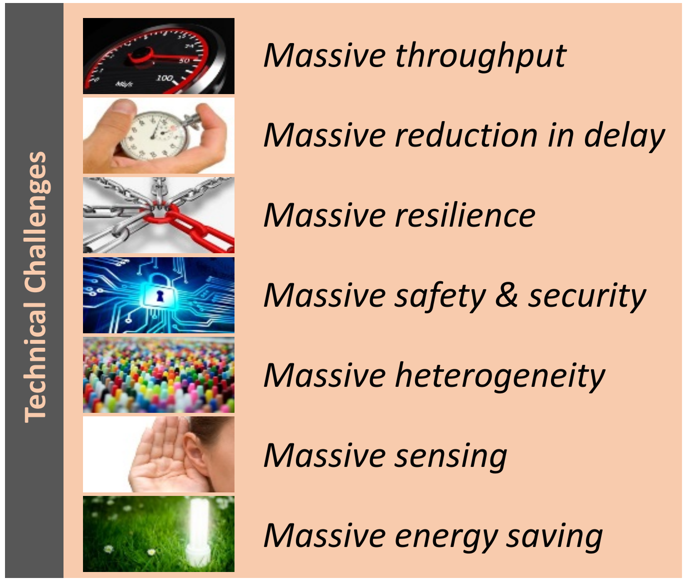
***Packet** Switched Networks*



Voice

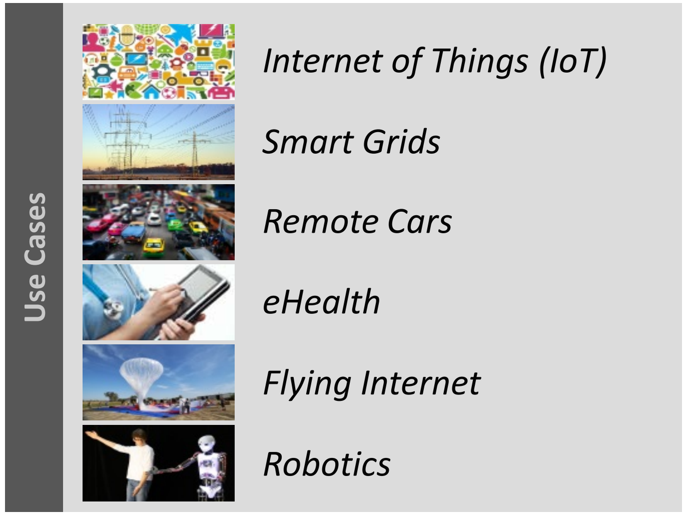
Data

People



Technical Challenges

- *Massive throughput*
- *Massive reduction in delay*
- *Massive resilience*
- *Massive safety & security*
- *Massive heterogeneity*
- *Massive sensing*
- *Massive energy saving*



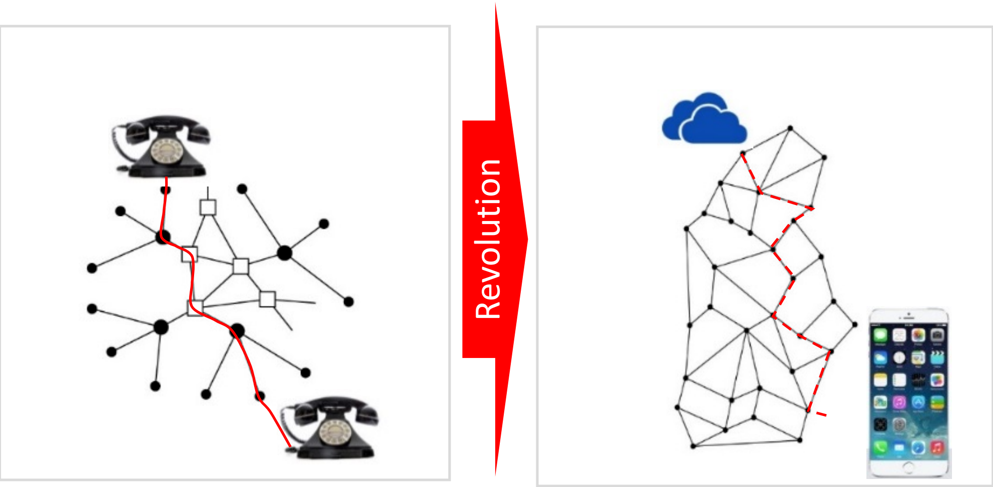
Use Cases

- *Internet of Things (IoT)*
- *Smart Grids*
- *Remote Cars*
- *eHealth*
- *Flying Internet*
- *Robotics*

Communication Networks

***Circuit* Switched Networks**

***Packet* Switched Networks**



Voice

Places

Voice

Data

People

Technical Challenges

Massive throughput

Massive reduction in delay

Massive resilience

Massive safety & security

Massive heterogeneity

Massive sensing

Massive energy saving

Use Cases

Internet of Things (IoT)

Smart Grids

Remote Cars

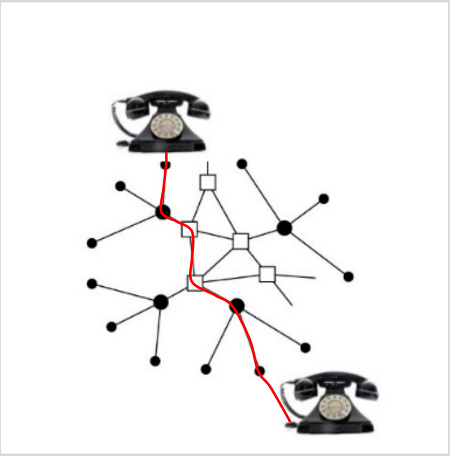
eHealth

Flying Internet

Robotics

Communication Networks

Circuit Switched Networks

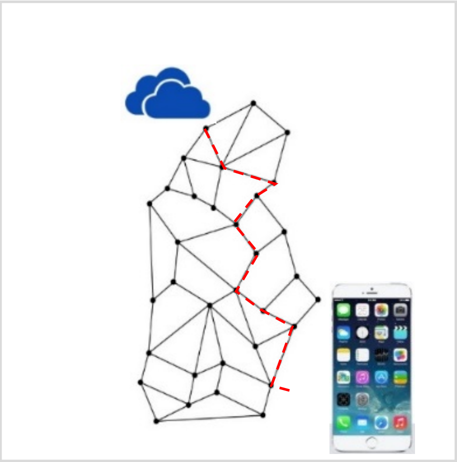


Voice

Places

Revolution

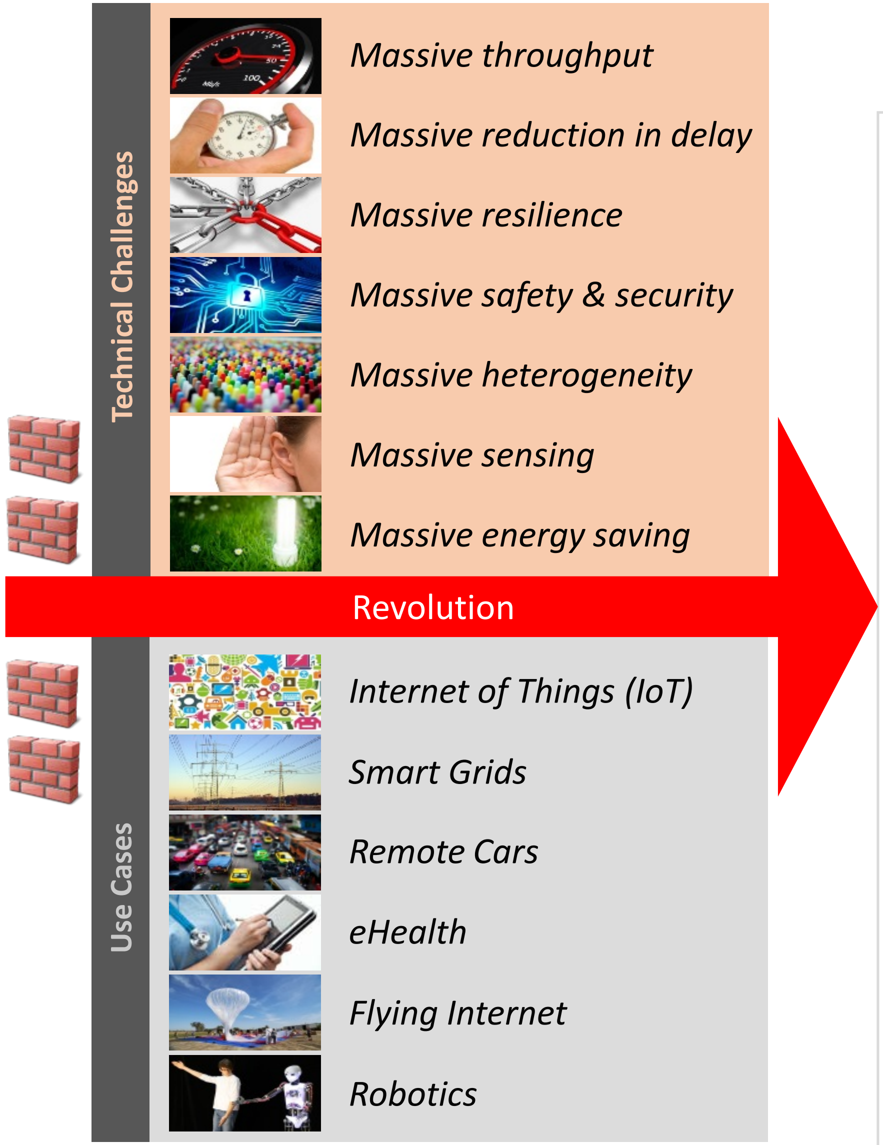
Packet Switched Networks



Voice

Data

People



Technical Challenges

Massive throughput

Massive reduction in delay

Massive resilience

Massive safety & security

Massive heterogeneity

Massive sensing

Massive energy saving

Revolution

Use Cases

Internet of Things (IoT)

Smart Grids

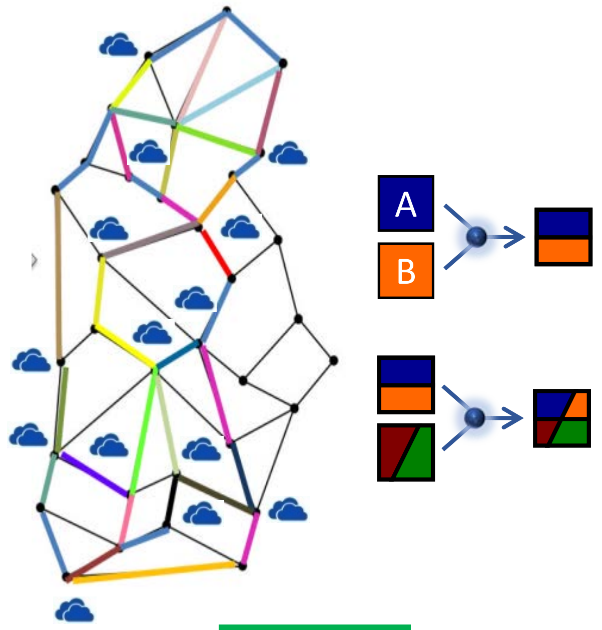
Remote Cars

eHealth

Flying Internet

Robotics

Code Centric Networks



Things

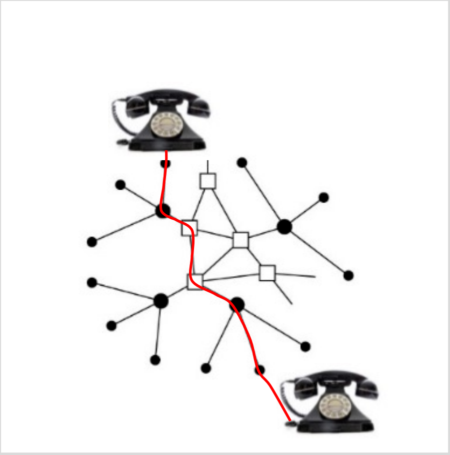
Voice

Data

Control

Communication Networks

***Circuit* Switched Networks**

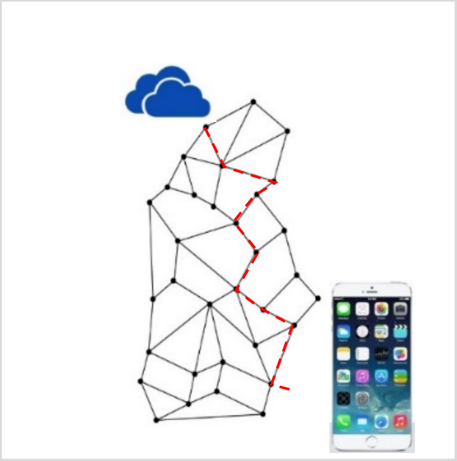


Voice

Places

Revolution

***Packet* Switched Networks**



Voice

Data

People

Technical Challenges

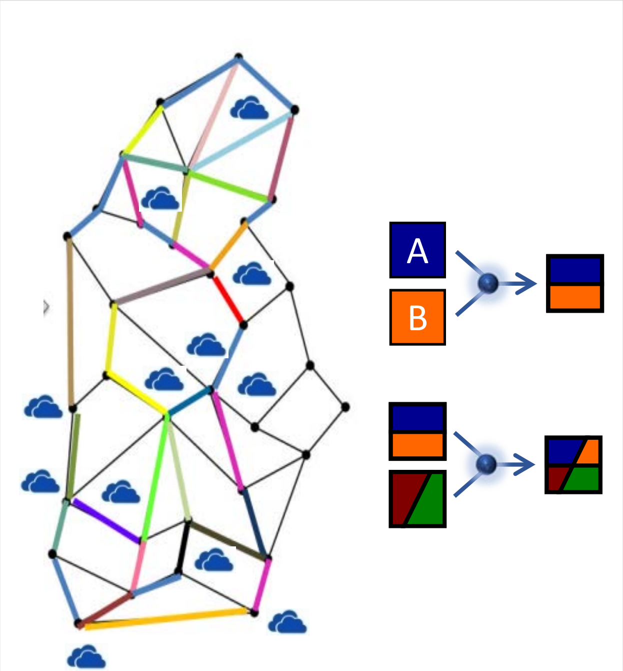
- *Massive throughput*
- *Massive reduction in delay*
- *Massive resilience*
- *Massive safety & security*
- *Massive heterogeneity*
- *Massive sensing*
- *Massive energy saving*

Revolution

Use Cases

- *Internet of Things (IoT)*
- *Smart Grids*
- *Remote Cars*
- *eHealth*
- *Flying Internet*
- *Robotics*

***Code* Centric Networks**



Things

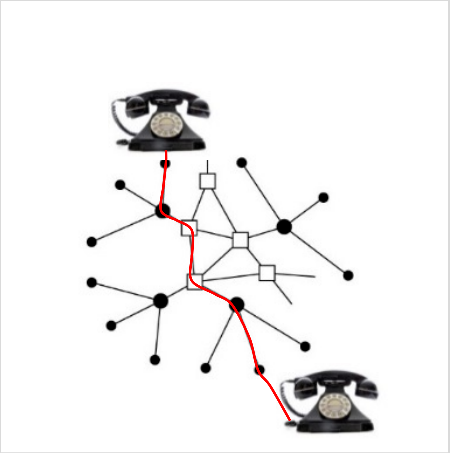
Voice

Data

Control

Communication Networks

***Circuit** Switched Networks*

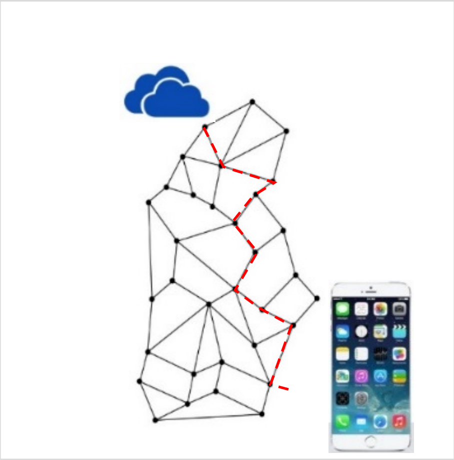


Voice

Places

Revolution

***Packet** Switched Networks*



Voice

Data

People

Technical Challenges

Massive throughput

Massive reduction in delay

Massive resilience

Massive safety & security

Massive heterogeneity

Massive sensing

Massive energy saving

Revolution

Use Cases

Internet of Things (IoT)

Smart Grids

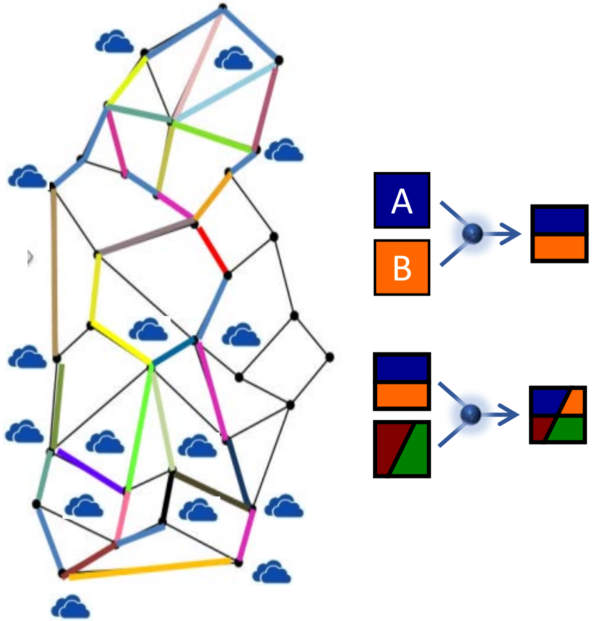
Remote Cars

eHealth

Flying Internet

Robotics

***Code** Centric Networks*



Things

Voice

Data

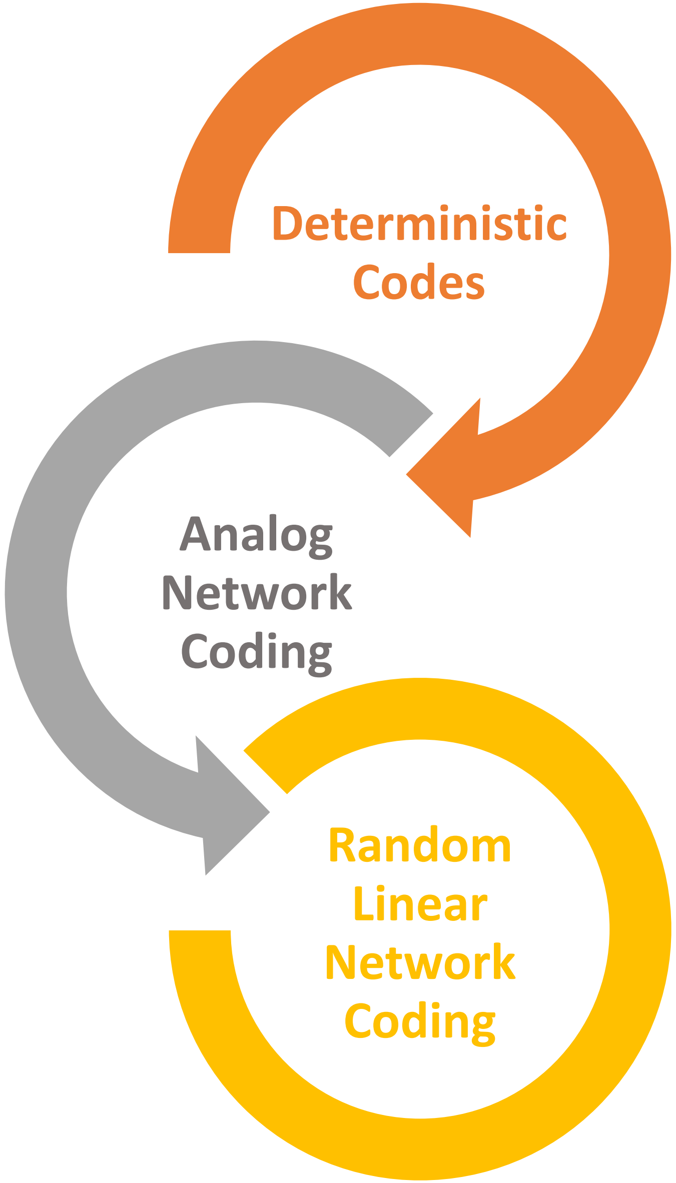
Control

Future Communication Systems

- Meshed networks will play a major role
- Multi-path is core for
  - More throughput
  - Higher reliability
  - Higher security
- Coding will play a major role
  - E2D will not be enough
  - Network coding is the magic juice
- Mobile Edge Cloud
- Fusion of Transport and Storage

One code to rule them all!

How do we approach NC?



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

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

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

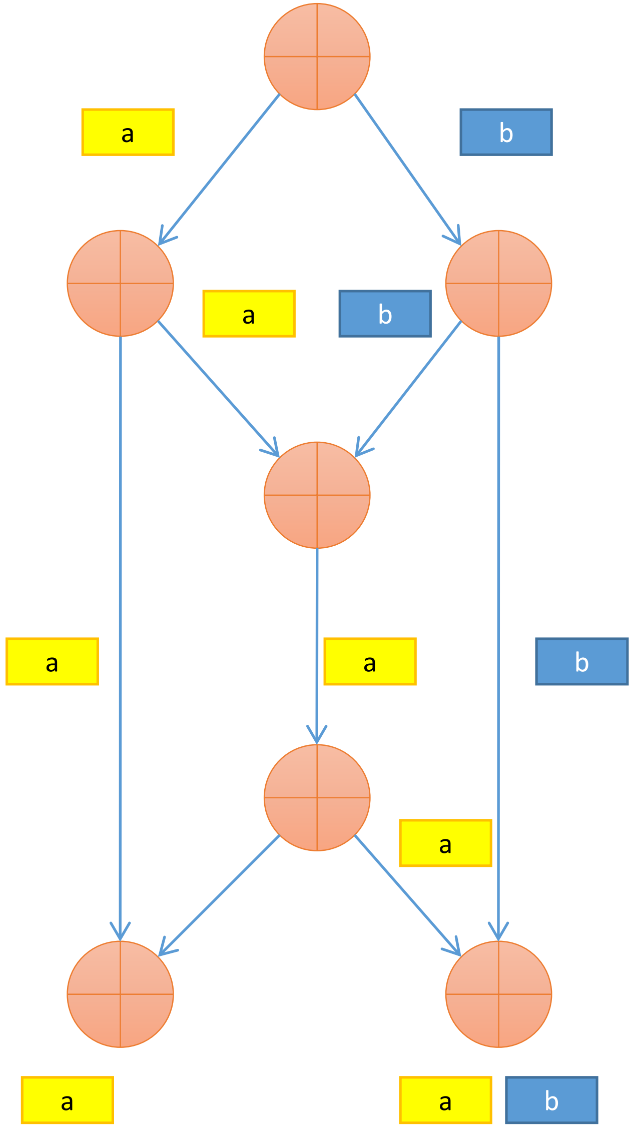
Digital Inter-Flow Network Coding: The Basics

Lecture 1

TECHNISCHE UNIVERSITÄT DRESDEN

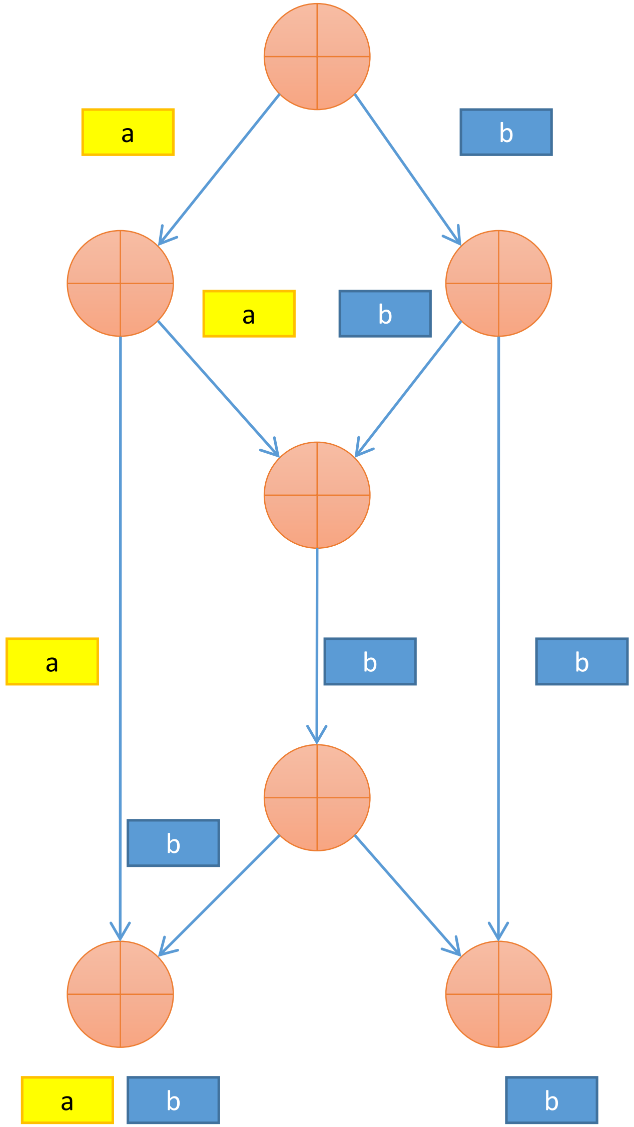
The Butterfly

Network Coding: The Butterfly



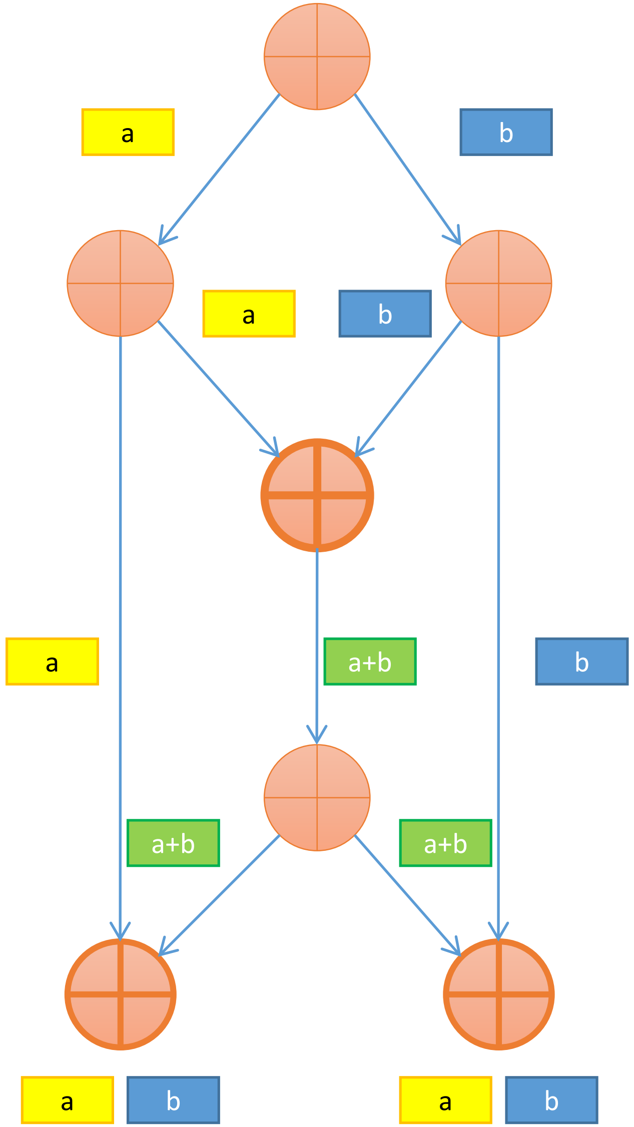
- Two packets a and b should be conveyed to two destinations
- Capacity per link can handle one packet per time slot
- Bottleneck in the middle
- Either packet a or b will path the bottleneck

Network Coding: The Butterfly



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

Network Coding: The Butterfly

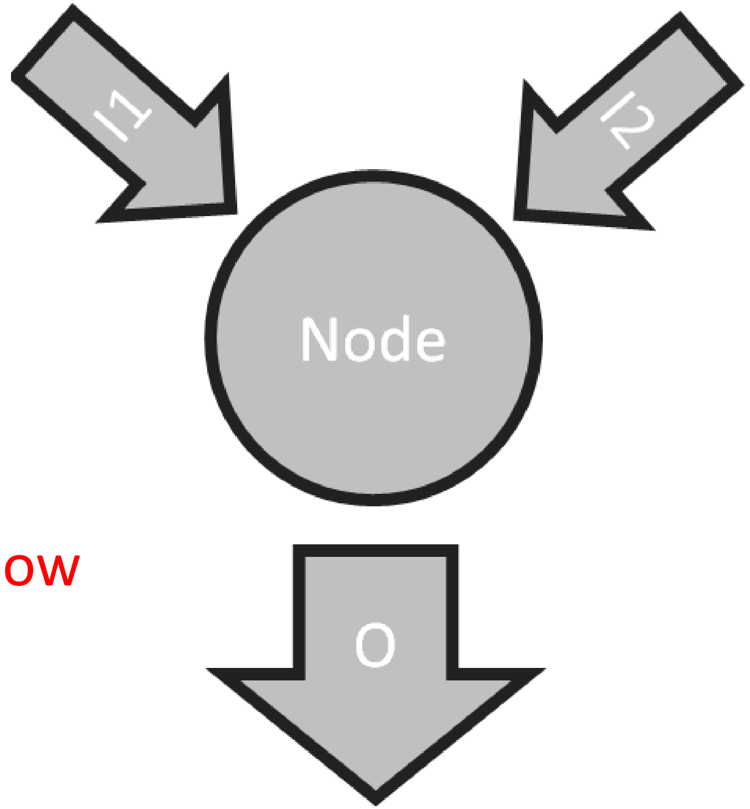


- Ahlswede et. al. In 2000
- Coding the packet
- Other ideas were around
- Max-flow min-cut theorem

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

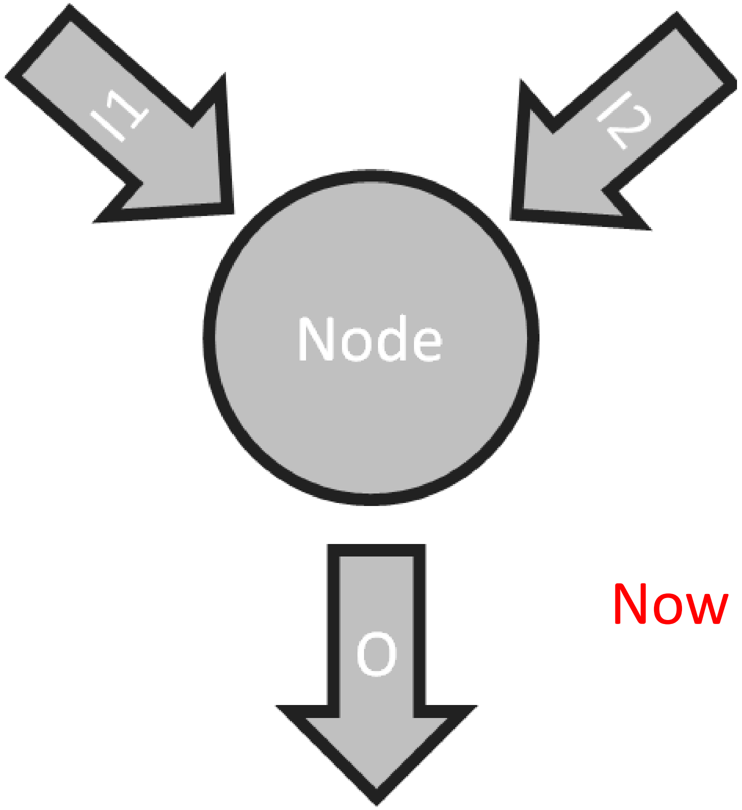
Kirchhoff



All engineers follow this principle!

$O = I1 + I2$

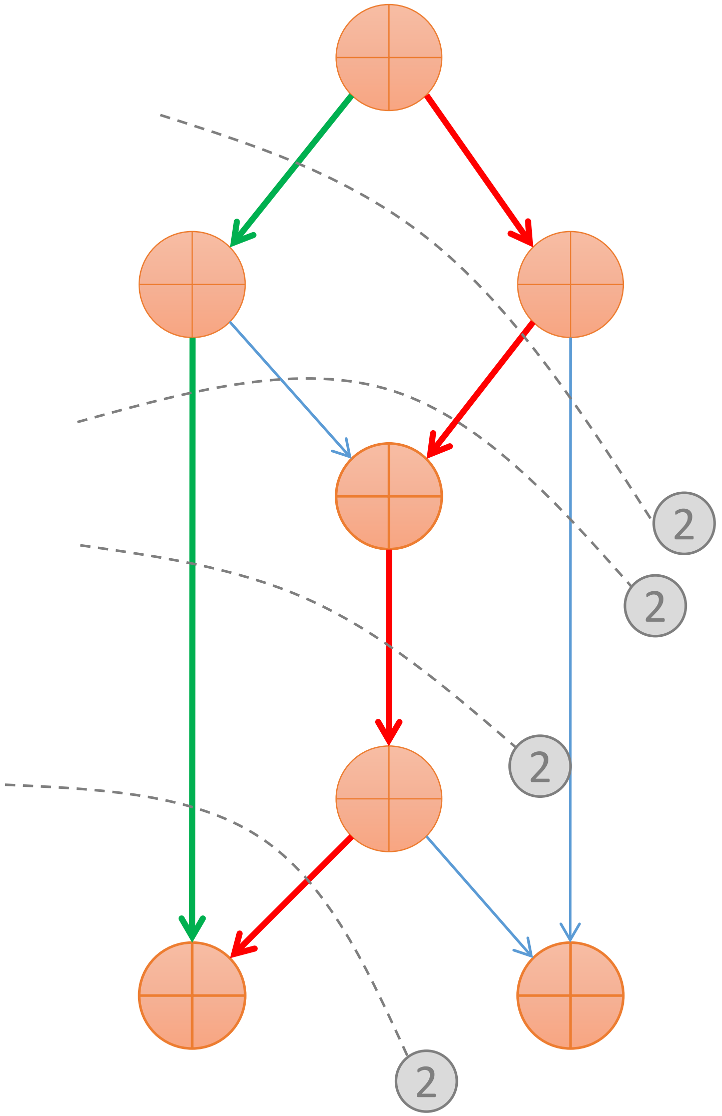
Network Coding

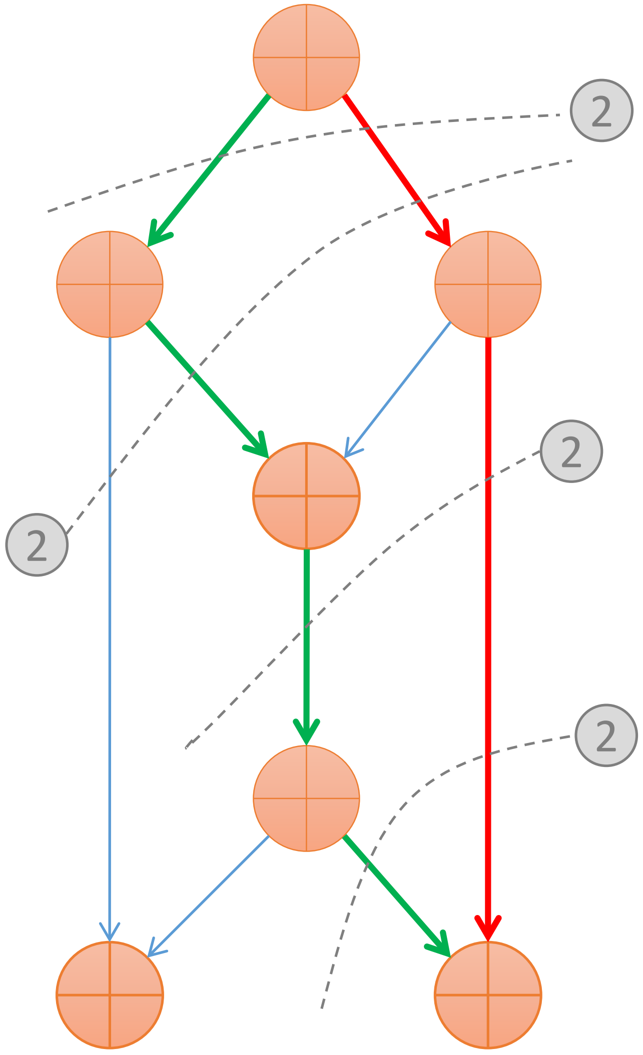


Now we are alone ...!

$O = f(I1,I2)$

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.

In a network, the value of the maximum flow equals the capacity of the minimum cut.

Network Coding: The Butterfly

CODING

0101

XOR 0011

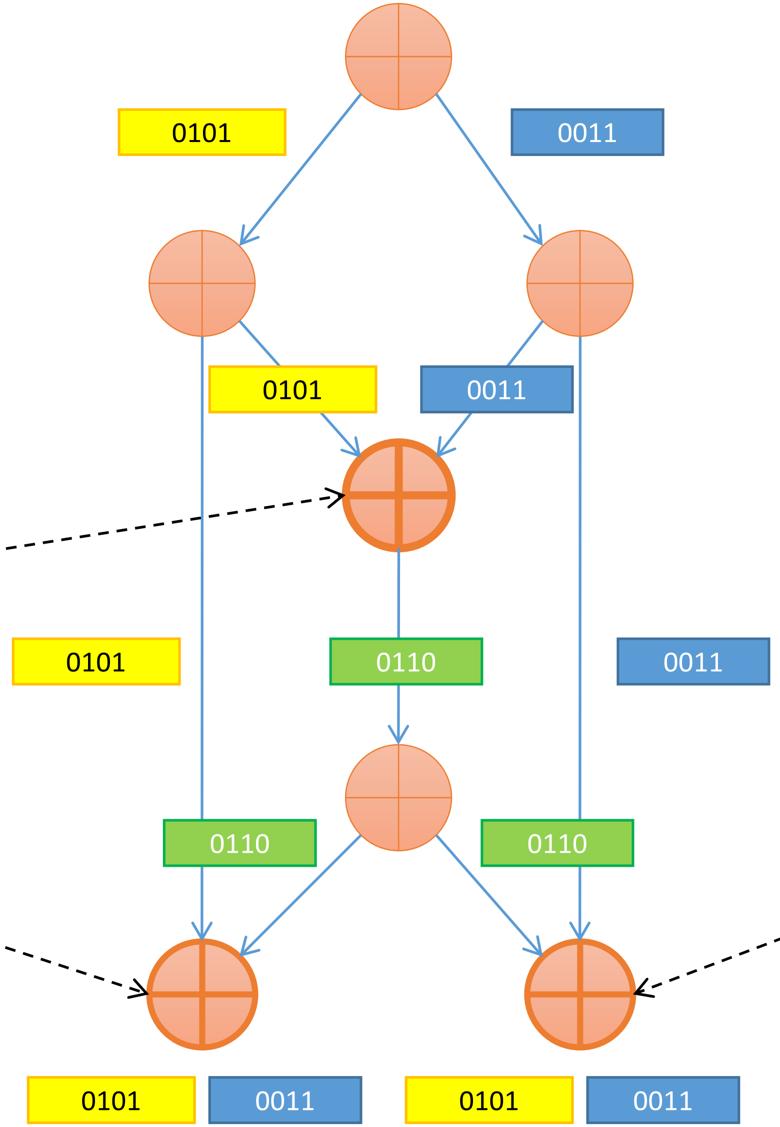
0110

DECODING

0101

XOR 0110

0011



DECODING

0011

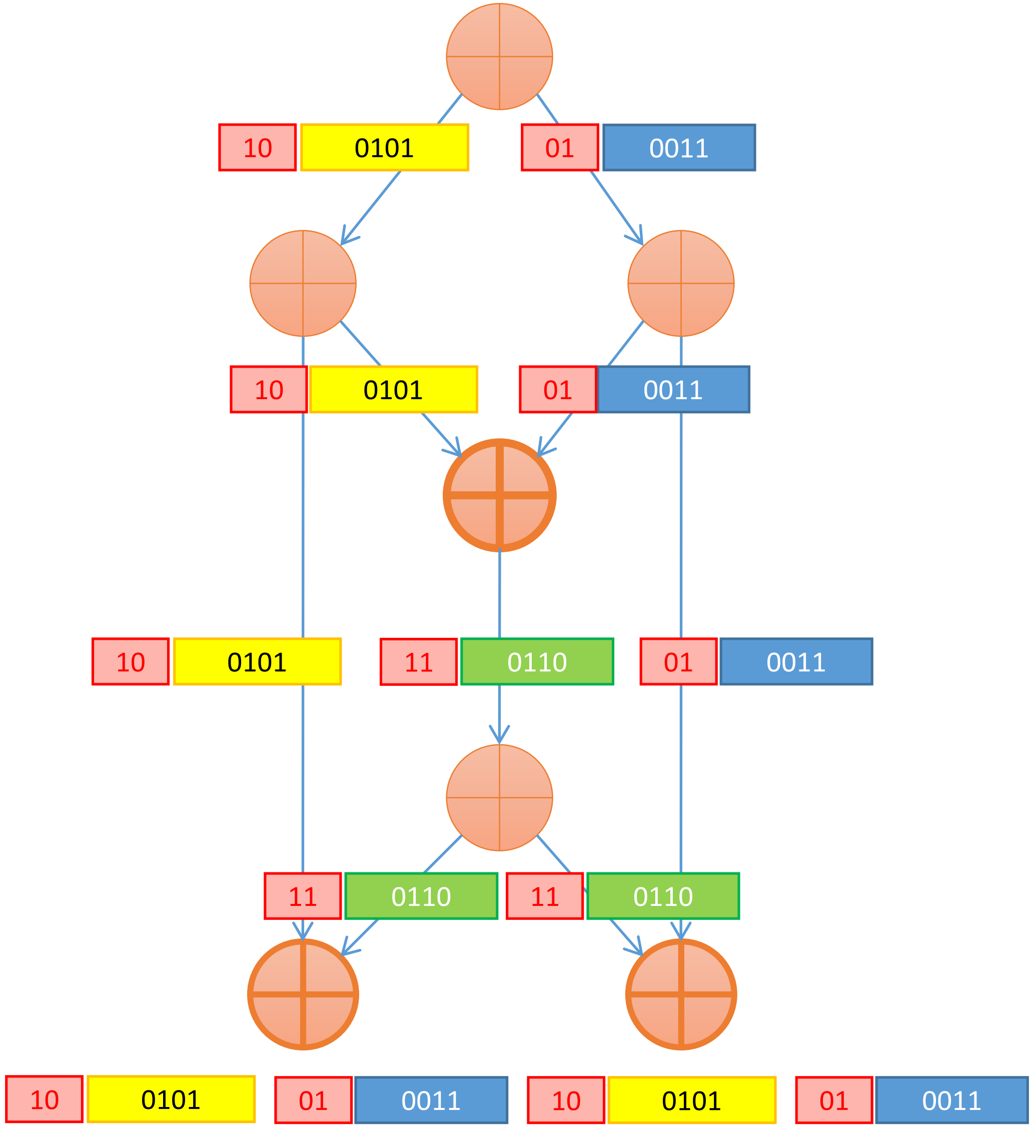
XOR 0110

0101

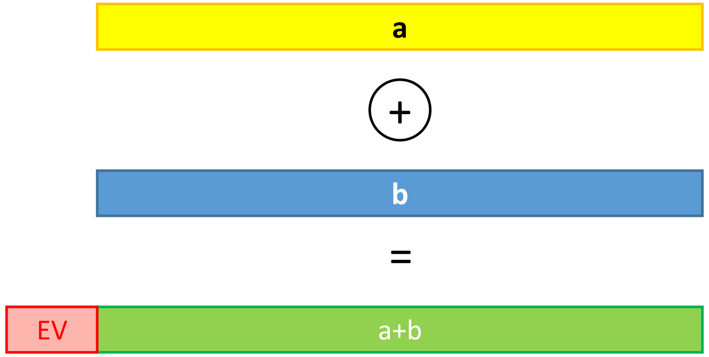
XOR operation

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

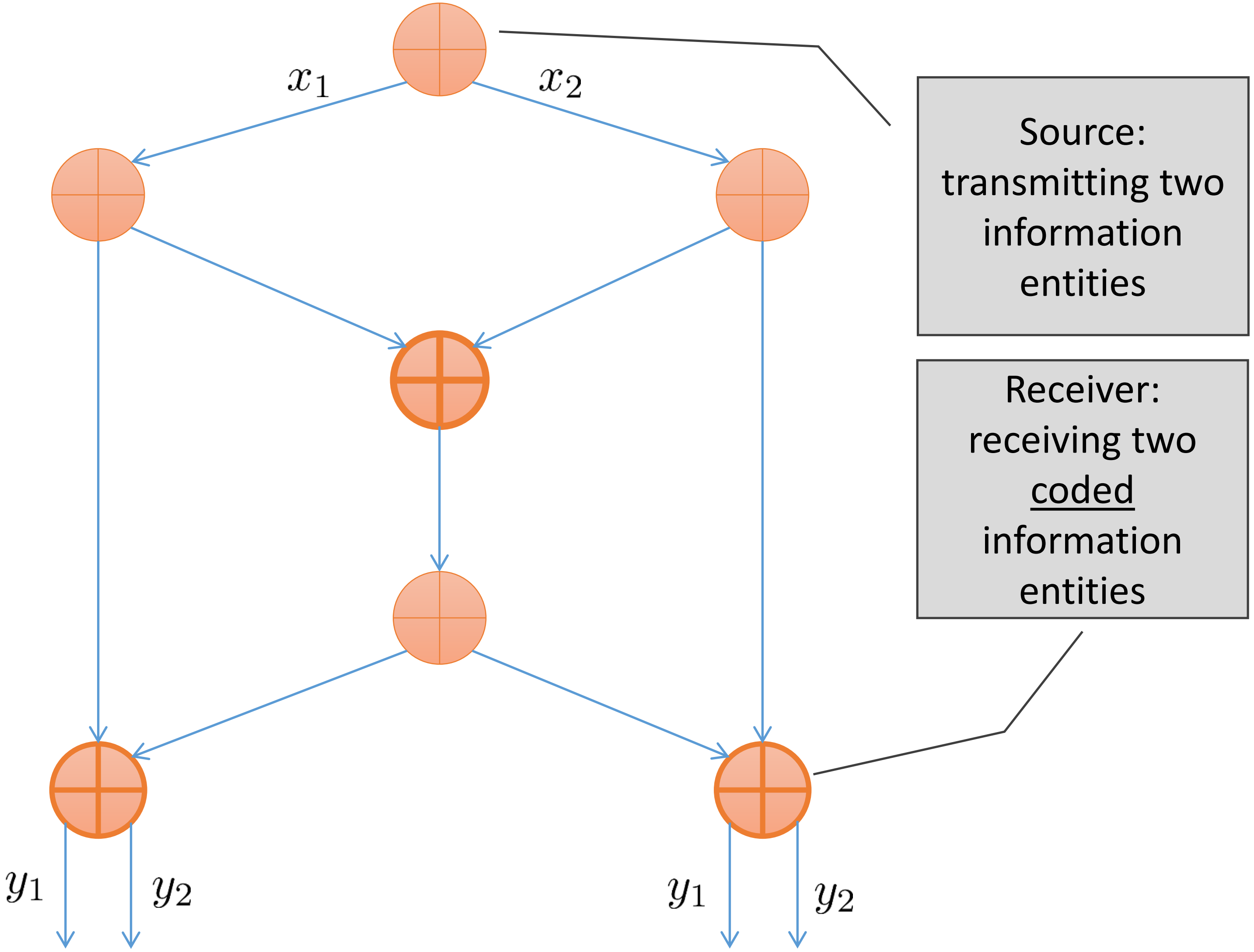
Network Coding: The Butterfly



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



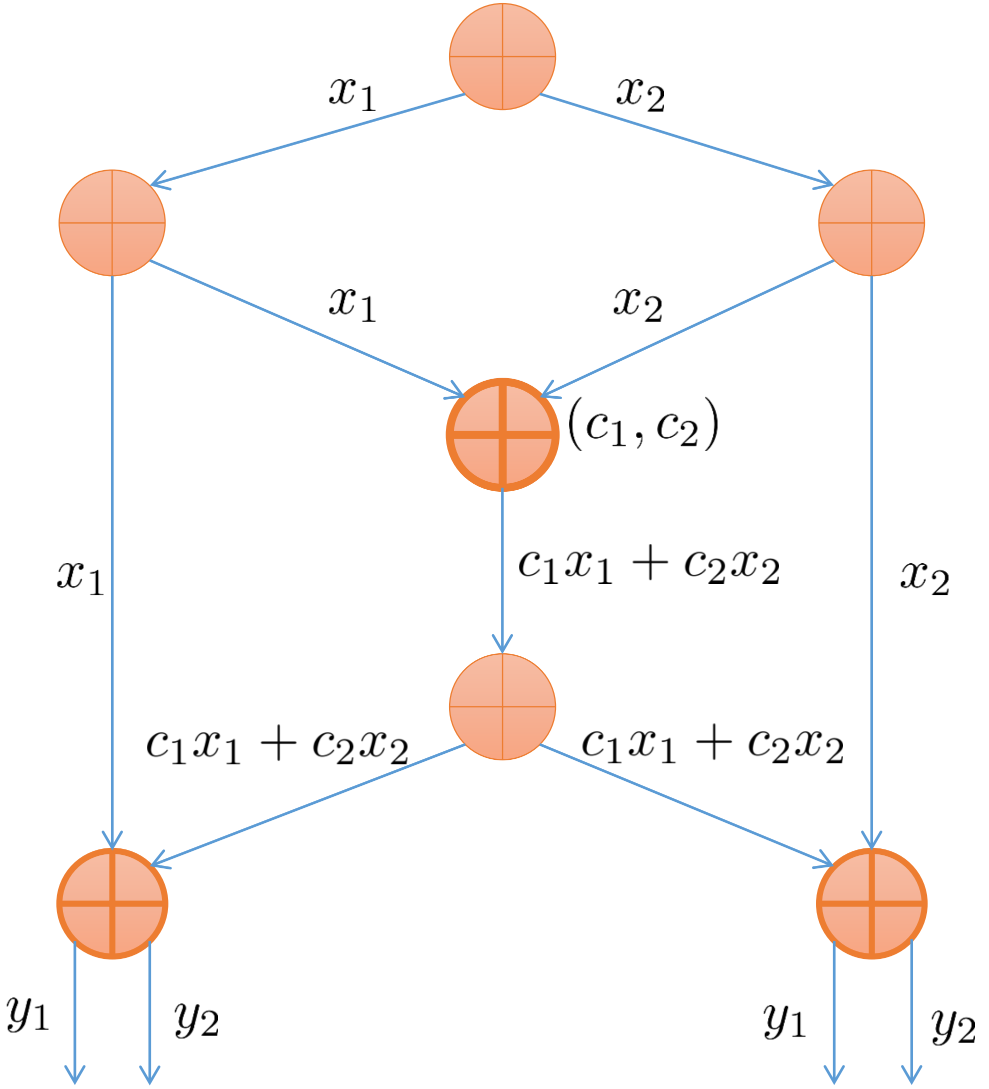
Network Coding: The Butterfly



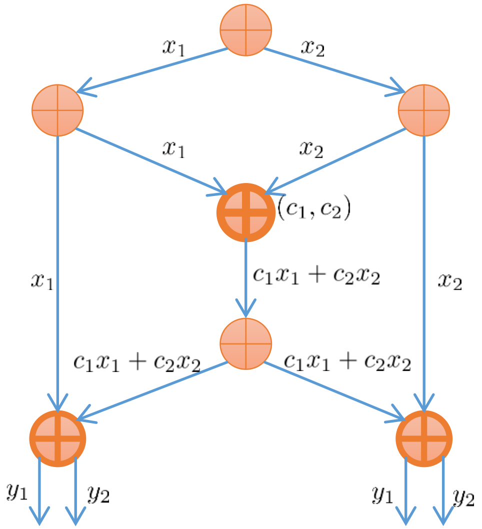
Source: transmitting two information entities

Receiver: receiving two <u>coded</u> information entities

Network Coding: The Butterfly



Network Coding: The Butterfly



$$\begin{aligned} y_1 &= x_1 \\ y_2 &= c_1x_1 + c_2x_2 \end{aligned}$$

$$\begin{aligned} y_1 &= c_1x_1 + c_2x_2 \\ y_2 &= x_2 \end{aligned}$$

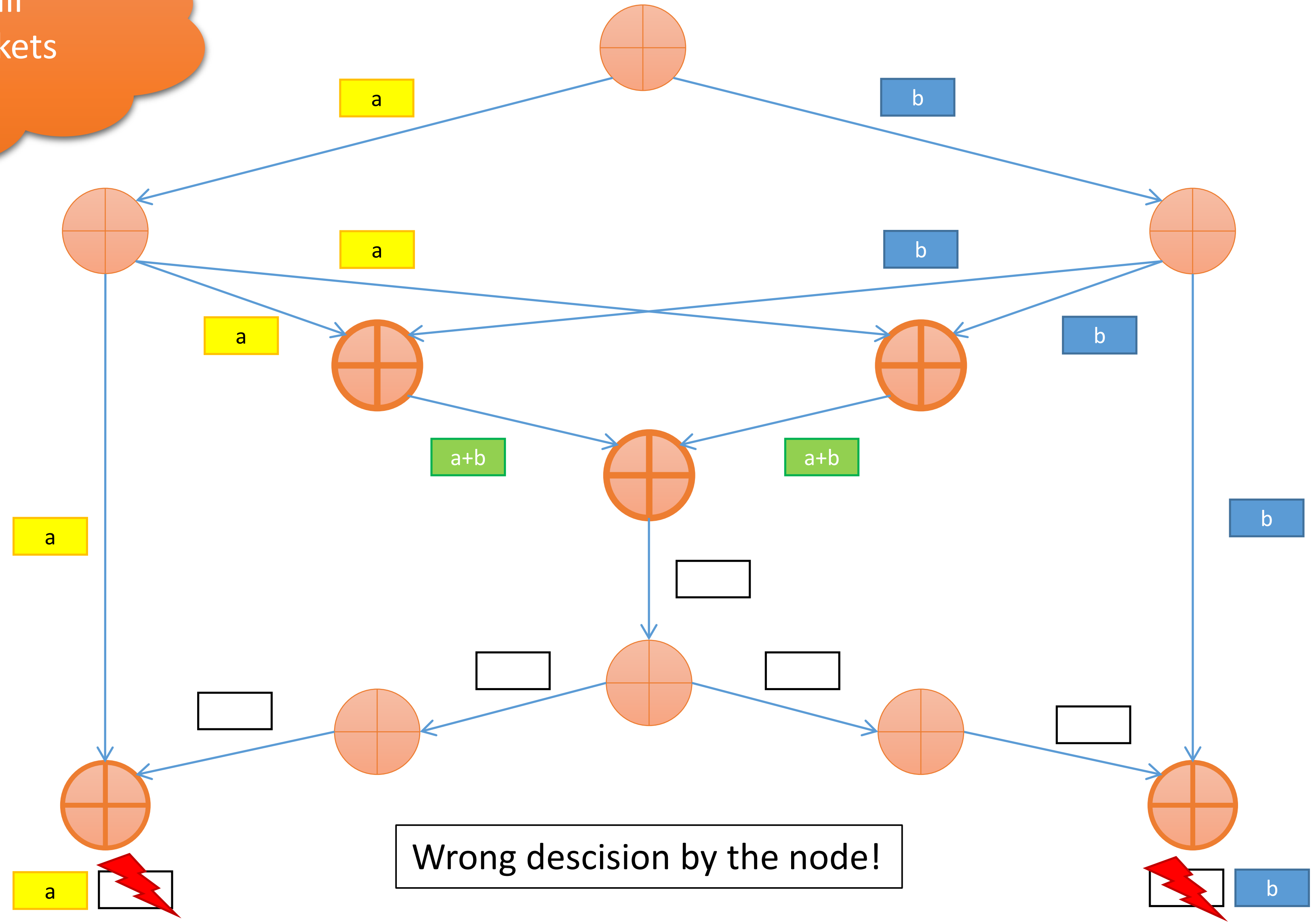
$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ c_1 & c_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

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

The Butterfly++

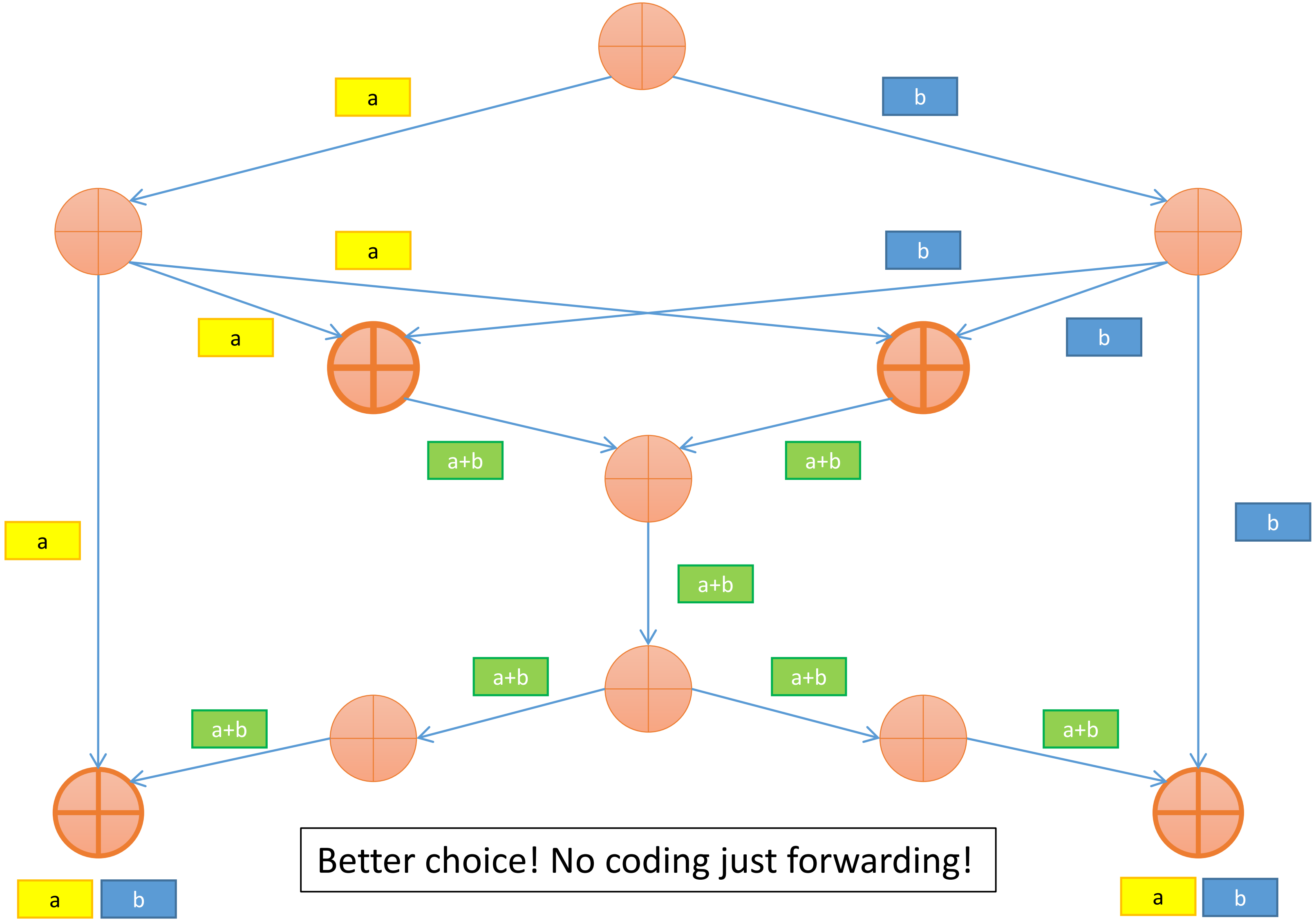
Network Coding: The Butterfly++

Let’s code all incoming packets …. mmmh



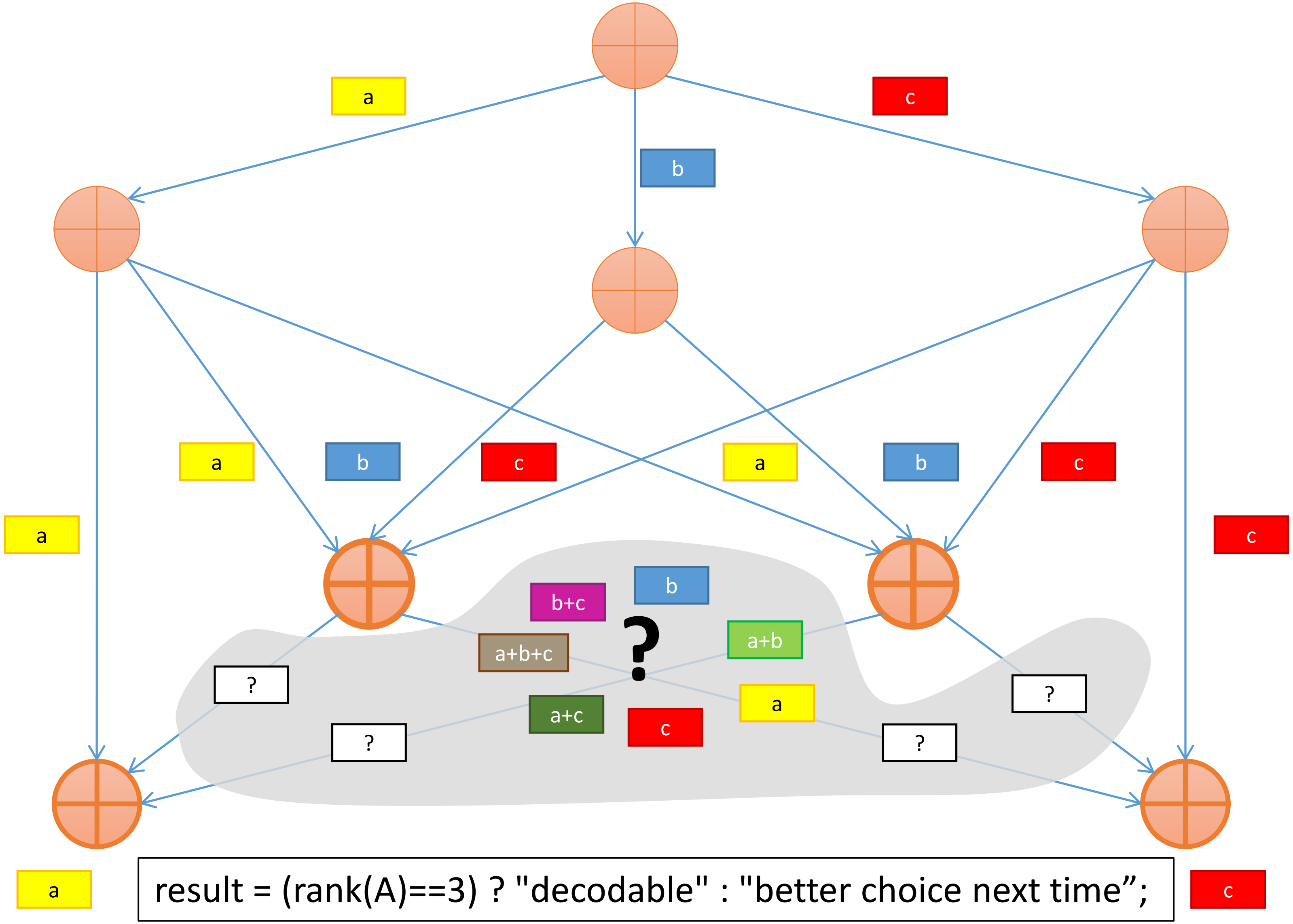
Wrong descision by the node!

Network Coding: The Butterfly++



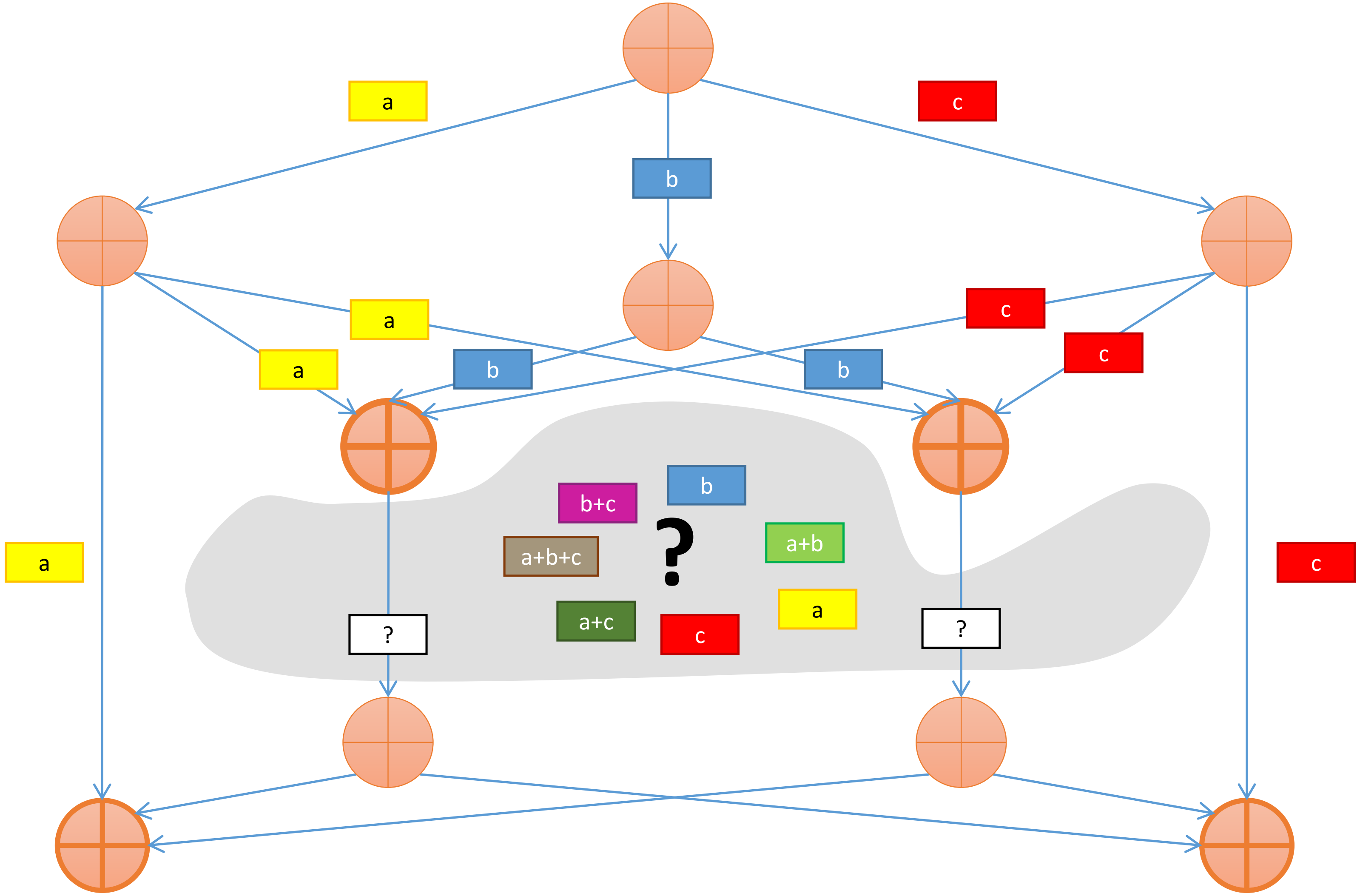
Better choice! No coding just forwarding!

Network Coding: The Butterfly++



result = (rank(A)==3) ? "decodable" : "better choice next time”;

Network Coding: The Butterfly++



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

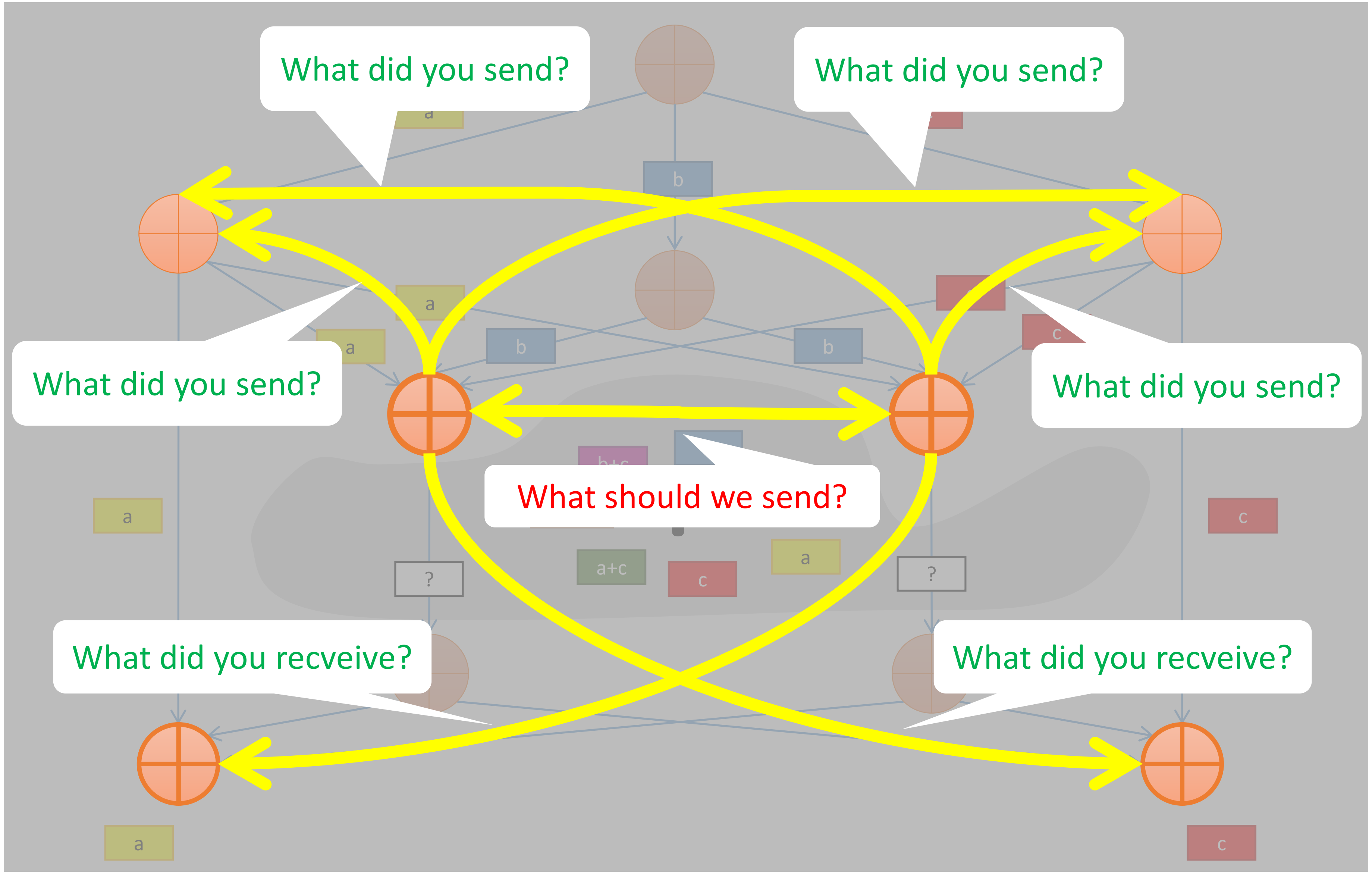
a

result = (rank(A)==3) ? "decodable" : "better choice next time”;

c

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

Network Coding: The Butterfly++



What did you send?

What did you send?

What did you send?

What did you send?

What should we send?

What did you recveive?

What did you recveive?

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.

TECHNISCHE UNIVERSITÄT DRESDEN

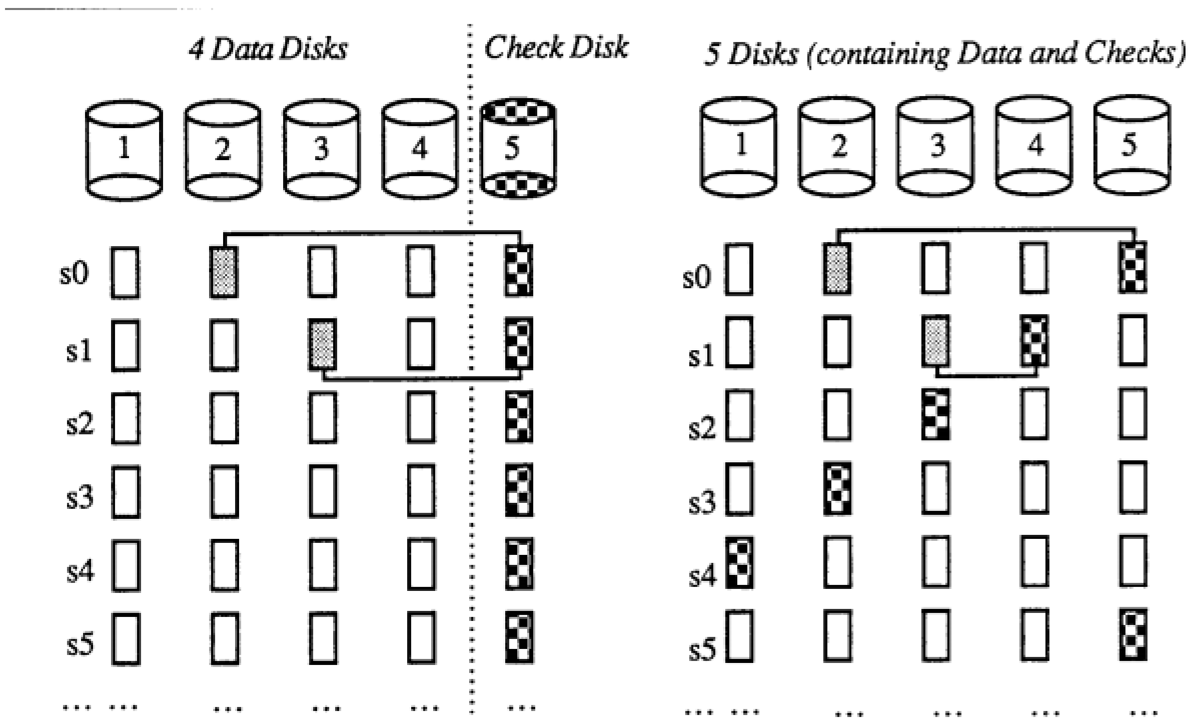
Storage

Later more …

Sounds familiar?

- Redundant Array of Independent Disks (RAID)
- Goal
  - Redundancy in storage
  - Faster information transfer
  - Reduce cost
- RAID 0,1,(2),(3),4,5, ...
- XOR

RAID 4/5



(a) Check information for Level 4 RAID for G=4 and C=1. The sectors are shown below the disks. (The checked areas indicate the check information.) Writes to s0 of disk 2 and s1 of disk 3 imply writes to s0 and s1 of disk 5. The check disk (5) becomes the write bottleneck.

(b) Check information for Level 5 RAID for G=4 and C=1. The sectors are shown below the disks, with the check information and data spread evenly through all the disks. Writes to s0 of disk 2 and s1 of disk 3 still imply 2 writes, but they can be split across 2 disks: to s0 of disk 5 and to s1 of disk 4.

A case for redundant arrays of inexpensive disks (RAID), D. A. Patterson, G. Gibson und R. H. Katz, 1988