Network Coding (NC)

CITHN2002 – Summer 2024

Prof. Dr.-Ing. Stephan Günther

Chair of Distributed Systems and Security
School of Computation, Information and Technology
Technical University of Munich

Organizational stuff

- 6 ECTS / 5 VU (lecture with integrated exercises)
- Module number CITHN2002, lecture times:
  - Wednesday 10:00 – 12:00
  - Wednesday 14:00 – 16:00

Organizational stuff

- 6 ECTS / 5 VU (lecture with integrated exercises)
- Module number CITHN2002, lecture times:
  - Wednesday 10:00 – 12:00
  - Wednesday 14:00 – 16:00

Exercises

- Lecture with integrated exercises
- Exercises are done on demand
- Problem sheets with solutions are provided
- Regular participation in lecture strongly recommended

Organizational stuff

Hardware provided for practical part / demos

- PC Engines APU2C4 [5]
- AMD Embedded G series GX-412TC (1 GHz AMD “Jaguar” quadcore)
- 4 GB DDR3-1333 ECC-RAM
- Intel I210AT Gigabit Ethernet
- 2.4 GHz and 5 GHz WLAN via distinct Atheros mini-PCIe cards
- Optional dual-band Ralink WLAN via USB
- Coreboot
- Serial console
- Running Debian “Stretch” with patched kernel for frame injection

Organizational stuff

Endterm / Retake

- Written, supervised exam at the end of the lecture period:
  - 75 minutes / 60 credits
  - 1 sheet of paper (A4), hand-written / printed (cheatsheet)
  - non-programmable pocket calculator
  - closed-book otherwise
- We do no “programming on paper”, promised
- Date and time *tba*

Organizational stuff

Endterm / Retake

- Written, supervised exam at the end of the lecture period:
  - 75 minutes / 60 credits
  - 1 sheet of paper (A4), hand-written / printed (cheatsheet)
  - non-programmable pocket calculator
  - closed-book otherwise
- We do no “programming on paper”, promised
- Date and time *tba*

Grading and Bonus

- There will be three homeworks consisting of old exam problems
- The homeworks give a total of 15 possible bonus credits
- Bonus credits are added to the final result of Endterm or Retake if the respective final exam is graded with 4.0 or better (“passed”) withouth the bonus.

Organizational stuff

Lecture material

⇒ `https://nchn.net.in.tum.de`

Some requests

- The course requires your continuous attendance:
  - not everything may be on slides and there will be discussions in class and presentations on the whiteboard
  - missing lectures without learning the stuff on your own results in fragmentation of the group
  - passing the exam does not become easier, either
- Please make contact with your fellows
  - helps you in learning: try explain problems to your team member
    ⇒ you will quickly recognize that you did not understand it in detail yourself
- If you want to quit, please:
  - unenroll from the course in TUMonline
  - a short email why you quit is appreciated

Chair of Distributed Systems and Security
School of Computation, Information and Technology
Technical University of Munich

NC can be considered as a generalization of routing and forwarding:

- Routing determines best-paths from source to destination.
- Forwarding switches packets along one of these paths.
- Forwarding merely creates replicas of incoming packets, i. e., a packet's payload remains unaltered.

Introduction

NC can be considered as a generalization of routing and forwarding:

- Routing determines best-paths from source to destination.
- Forwarding switches packets along one of these paths.
- Forwarding merely creates replicas of incoming packets, i. e., a packet's payload remains unaltered.

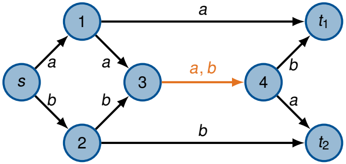
NC drops this restriction:

- Outgoing packets are arbitrary combinations of previously received packets.
- The process of combining packets in such a way is referred to as coding.
- Since coding does not only happen at the source but on any node in the network, one sais that "the network codes on packets".

Introduction

Example 1: the famous butterfly network

Source s transmits 2 packets a, b to both t_1, t_2 (multicast):

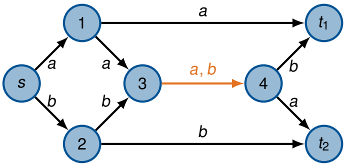


(a) Routing (with multicast)

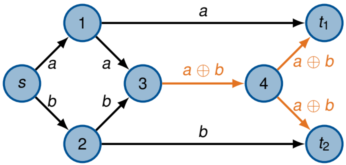
- The link $(3, 4)$ poses a bottleneck and must be used twice.

Example 1: the famous butterfly network

Source s transmits 2 packets a, b to both t_1, t_2 (multicast):



(a) Routing (with multicast)



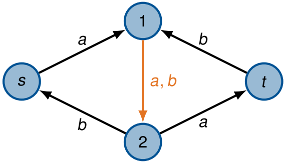
(b) Network Coding

- The link $(3, 4)$ poses a bottleneck and must be used twice.

- NC saves one transmission on the critical link $(3, 4)$.
- t_1, t_2 can decode the missing packet by XORing the coded packet with a and b, respectively.

Example 2: diamond network

Nodes s, t want to communicate with each other (bidirectional unicasts):

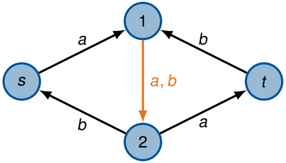


(a) Routing

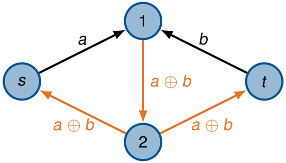
- The link $(1, 2)$ poses a bottleneck and must be used twice.

Example 2: diamond network

Nodes s, t want to communicate with each other (bidirectional unicasts):



(a) Routing



(b) Network Coding

- The link $(1, 2)$ poses a bottleneck and must be used twice.

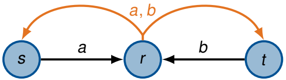
- NC saves again one transmission on the critical link $(1, 2)$.
- s, t know what they have sent and are thus able to decode.

Example 3: wireless relay network

Nodes s, t want to communicate with each other (bidirectional unicasts):

Note:

- Only 1 node can transmit at any time (otherwise transmissions would collide).
- A transmission by r is seen by both s, t (broadcast-nature of wireless networks).



(a) Routing

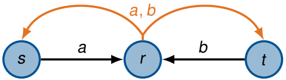
- The relay has to transmit a, b individually using 2 distinct broadcasts.
- Although s, t might overhear both transmissions, only one transmission is interesting for each node.

Example 3: wireless relay network

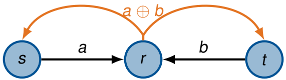
Nodes s, t want to communicate with each other (bidirectional unicasts):

Note:

- Only 1 node can transmit at any time (otherwise transmissions would collide).
- A transmission by r is seen by both s, t (broadcast-nature of wireless networks).



(a) Routing



(b) Network Coding

- The relay has to transmit a, b individually using 2 distinct broadcasts.
- Although s, t might overhear both transmissions, only one transmission is interesting for each node.

- With NC, the relay transmits $a \oplus b$.
- Both s, t know what they have sent and are thus able to decode the missing packet.

Introduction

Throughput gain and reduced complexity

- Examples 1–3 already demonstrated the potential gain in throughput.
- May be even more interesting: in certain situations NC allows for a reduction in complexity:
  - The problem to find an optimal solution for Example 1 with routing results in the Steiner Tree problem, which is $\mathcal{NP}$.
  - With NC, a solution is found efficiently in polynomial time.

Introduction

Throughput gain and reduced complexity

- Examples 1–3 already demonstrated the potential gain in throughput.
- May be even more interesting: in certain situations NC allows for a reduction in complexity:
  - The problem to find an optimal solution for Example 1 with routing results in the Steiner Tree problem, which is $\mathcal{NP}$.
  - With NC, a solution is found efficiently in polynomial time.

Robustness and adaptability

During the course of this class we will see that NC not only allows for

- more efficient channel usage but also
- reduces the cost of acknowledging and retransmitting packets.

Introduction

Peer-to-peer content distribution (see Avalanche [2, 1])

Imagine a peer-to-peer network:

- A file is split into $n = 3$ blocks and spread over multiple nodes.
- Some node i has a set of $N(i) = \{1, 2, 3\}$ neighbors.
- For simplicity assume that each $j \in N(i)$ posseses the whole file.
- i asks each $j \in N(i)$ to send 1 of its blocks.
- Each $j \in N(i)$ chooses a packet independently and uniformely distributed.
- What is the probability that i gets the whole file?

Introduction

Peer-to-peer content distribution (see Avalanche [2, 1])

Imagine a peer-to-peer network:

- A file is split into $n = 3$ blocks and spread over multiple nodes.
- Some node i has a set of $N(i) = \{1, 2, 3\}$ neighbors.
- For simplicity assume that each $j \in N(i)$ posseses the whole file.
- i asks each $j \in N(i)$ to send 1 of its blocks.
- Each $j \in N(i)$ chooses a packet independently and uniformely distributed.
- What is the probability that i gets the whole file?

$$p = 1 \cdot \frac{2}{3} \cdot \frac{1}{3} \approx 22\,\%$$

Introduction

Peer-to-peer content distribution (see Avalanche [2, 1])

Imagine a peer-to-peer network:

- A file is split into $n = 3$ blocks and spread over multiple nodes.
- Some node i has a set of $N(i) = \{1, 2, 3\}$ neighbors.
- For simplicity assume that each $j \in N(i)$ posseses the whole file.
- i asks each $j \in N(i)$ to send 1 of its blocks.
- Each $j \in N(i)$ chooses a packet independently and uniformely distributed.
- What is the probability that i gets the whole file?

$$p = 1 \cdot \frac{2}{3} \cdot \frac{1}{3} \approx 22\,\%$$

Now assume the following:

- $j \in N(i)$ sends the XOR of a random number of blocks.
- To decide whether or not each of the blocks should be XORed, j flips a coin.
- The outcome of those trials is sent along with the XOR to i.
- i can obviously decode if those trials are linear independent.

Introduction

Peer-to-peer content distribution (see Avalanche [2, 1])

Imagine a peer-to-peer network:

- A file is split into $n = 3$ blocks and spread over multiple nodes.
- Some node i has a set of $N(i) = \{1, 2, 3\}$ neighbors.
- For simplicity assume that each $j \in N(i)$ posseses the whole file.
- i asks each $j \in N(i)$ to send 1 of its blocks.
- Each $j \in N(i)$ chooses a packet independently and uniformely distributed.
- What is the probability that i gets the whole file?

$$p = 1 \cdot \frac{2}{3} \cdot \frac{1}{3} \approx 22\,\%$$

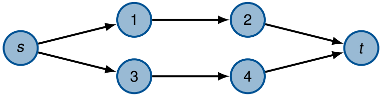
Now assume the following:

- $j \in N(i)$ sends the XOR of a random number of blocks.
- To decide whether or not each of the blocks should be XORed, j flips a coin.
- The outcome of those trials is sent along with the XOR to i.
- i can obviously decode if those trials are linear independent.

$$p' = \left(1 - \frac{1}{2^3}\right)\left(1 - \frac{1}{2^2}\right)\left(1 - \frac{1}{2}\right) \approx 33\,\%$$

Network security

- s wants to send messages to t.
- s knows that one of the four relay nodes is operated by an eavesdropper.



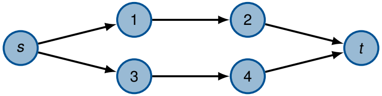
Routing:

- Since s does not know the eavesdropper, it has an odd by $1/2$ to choose the wrong path.
- Sending packets alternating over both paths might still yield information to the eavesdropper.

Introduction

Network security

- s wants to send messages to t.
- s knows that one of the four relay nodes is operated by an eavesdropper.



Routing:

- Since s does not know the eavesdropper, it has an odd by $1/2$ to choose the wrong path.
- Sending packets alternating over both paths might still yield information to the eavesdropper.

Network Coding:

- s splits every message to be sent into four packets p_i, $1 \leq i \leq 4$ of equal size.
- s then calculates

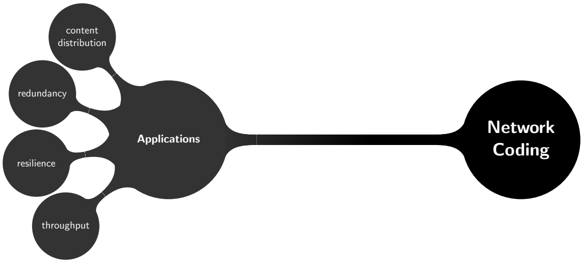
$$c_1 = p_1 \oplus p_2, \; c_2 = p_3 \oplus p_4, \; c_3 = p_2 \oplus p_3, \; c_4 = p_1 \oplus p_3 \oplus p_4$$

  and sends c_1, c_2 over one path and c_3, c_4 over the other one.
- As long as the eavesdropper is unable to guess the contents of at least one packet, decoding is impossible.

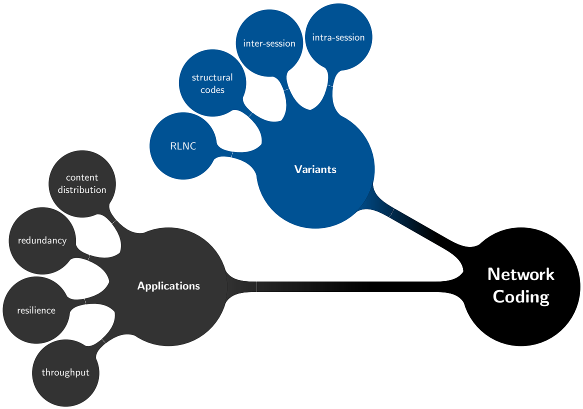
Introduction

Network Coding

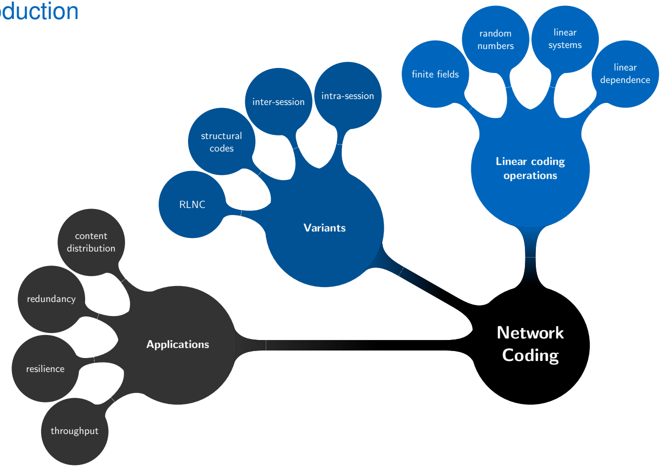
Introduction

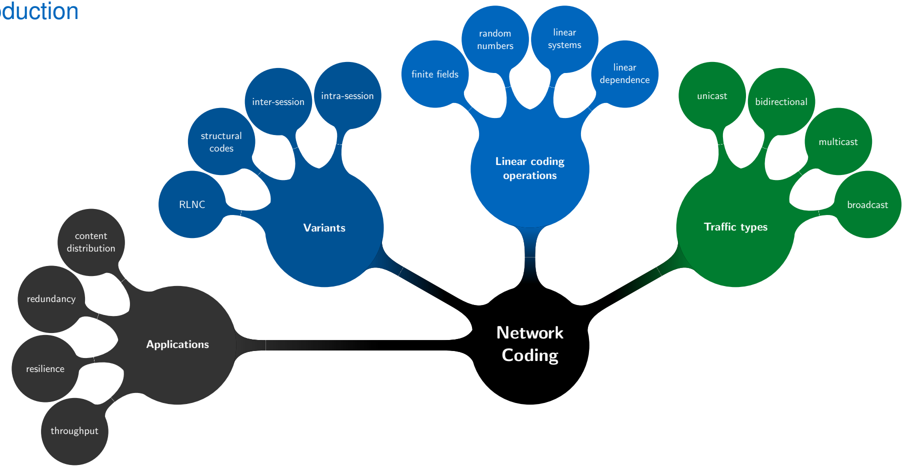


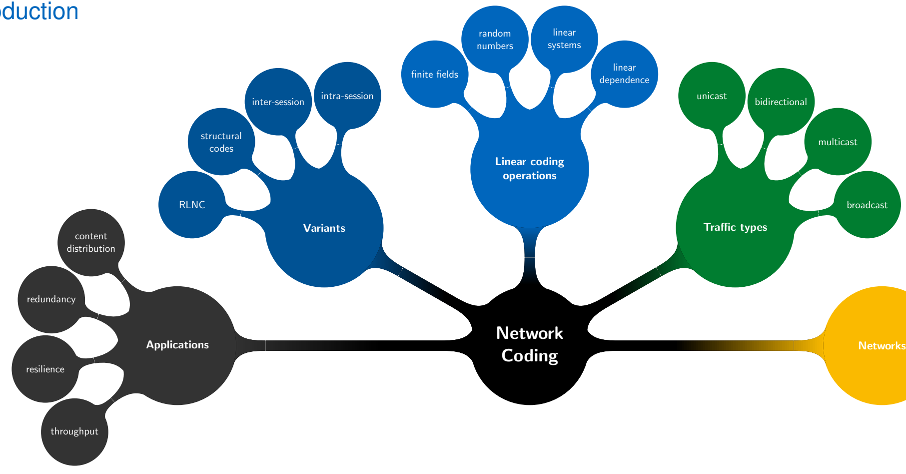
Introduction

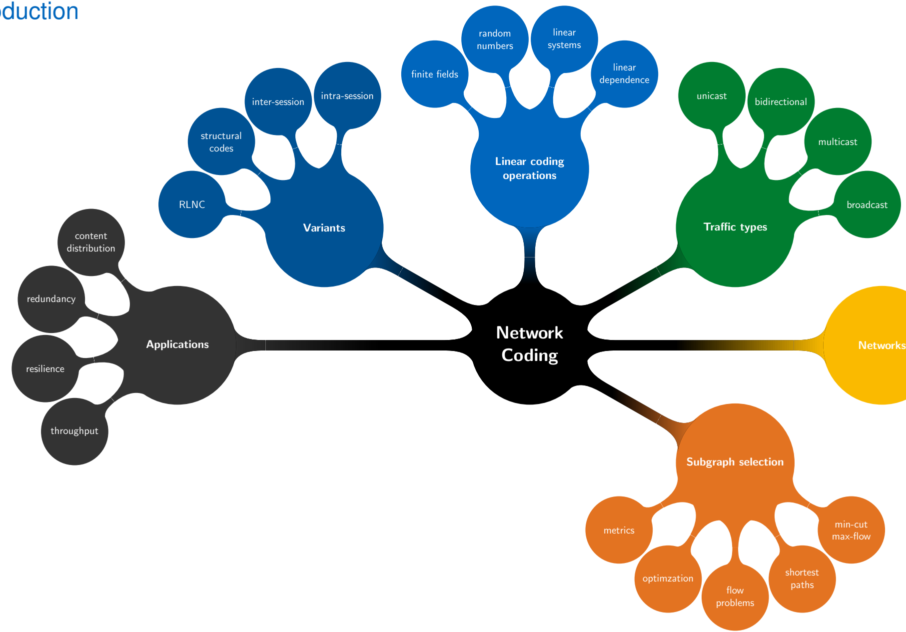


Introduction



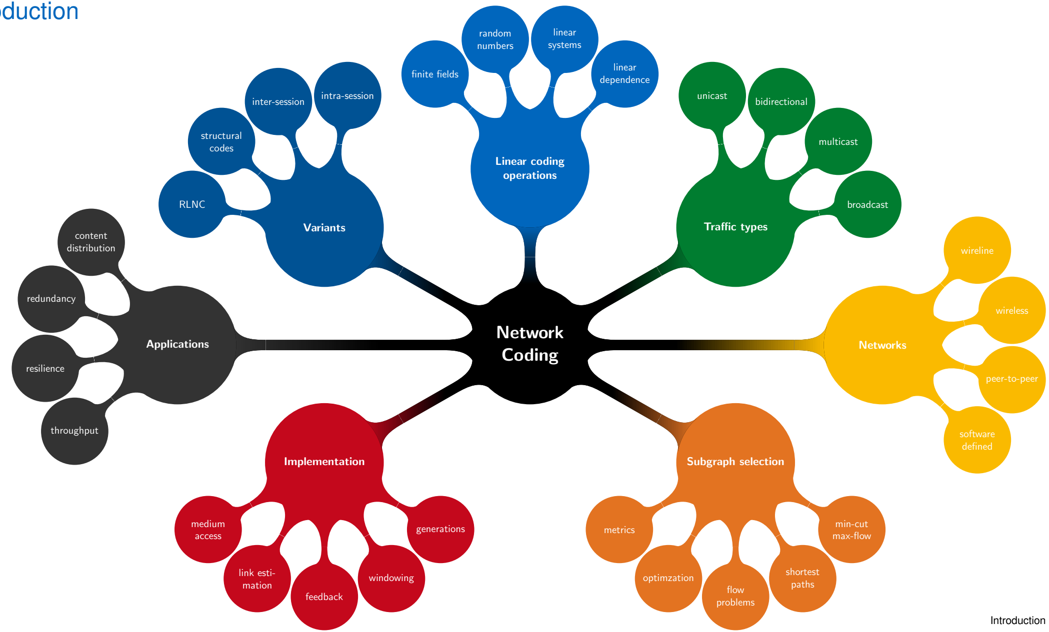

Network Coding
Applications
content distribution
redundancy
resilience
throughput
Variants
RLNC
structural codes
inter-session
intra-session
Linear coding operations
finite fields
random numbers
linear systems
linear dependence
Traffic types
unicast
bidirectional
multicast
broadcast


Network Coding
Applications
content distribution
redundancy
resilience
throughput
Variants
RLNC
structural codes
inter-session
intra-session
Linear coding operations
finite fields
random numbers
linear systems
linear dependence
Traffic types
unicast
bidirectional
multicast
broadcast
Networks
wireline
wireless
peer-to-peer
software defined



Network Coding
Linear coding operations
finite fields
random numbers
linear systems
linear dependence
Variants
RLNC
structural codes
inter-session
intra-session
Traffic types
unicast
bidirectional
multicast
broadcast
Networks
wireline
wireless
peer-to-peer
software defined
Subgraph selection
metrics
optimzation
flow problems
shortest paths
min-cut max-flow
Applications
content distribution
redundancy
resilience
throughput

Introduction



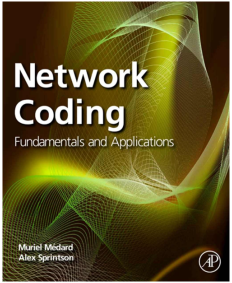
Organizational stuff

Introduction

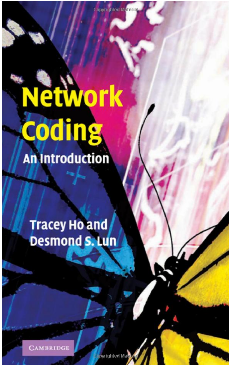
Literature

Bibliography

Literature



(a) Network Coding: Fundamentals and Applications [4]



(b) Network Coding: An Introduction [3]

And don’t forget to study the Linux Kernel Coding Style [6]!

Organizational stuff

Introduction

Literature

Bibliography

Bibliography

[1] C. Gkantsidis and M. Goldberg.
Avalanche: File Swarming with Network Coding.
http://research.microsoft.com/en-us/projects/avalanche/.

[2] C. Gkantsidis and P. Rodriguez.
Network Coding for Large Scale Content Distribution.
In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOMM).*, volume 4, pages 2235–2245, March 2005.

[3] T. Ho and D. Lun.
Network Coding: An Introduction.
Cambridge University Press, 2008.

[4] M. Médard and A. Sprintson.
Network Coding: Fundamentals and Applications.
Academic Press, 2011.

[5] PCengines.
APU2C4.
http://www.pcengines.ch/apu2c4.htm.

[6] L. Torvalds et al.
Linux Kernel Coding Syle.
https://www.kernel.org/doc/Documentation/process/coding-style.rst.