HS19

DISTRIBUTED SYSTEMS ADVANCED & BLOCKCHAIN

Repetition VSS

T. Bocek

thomas.bocek@hsr.ch

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

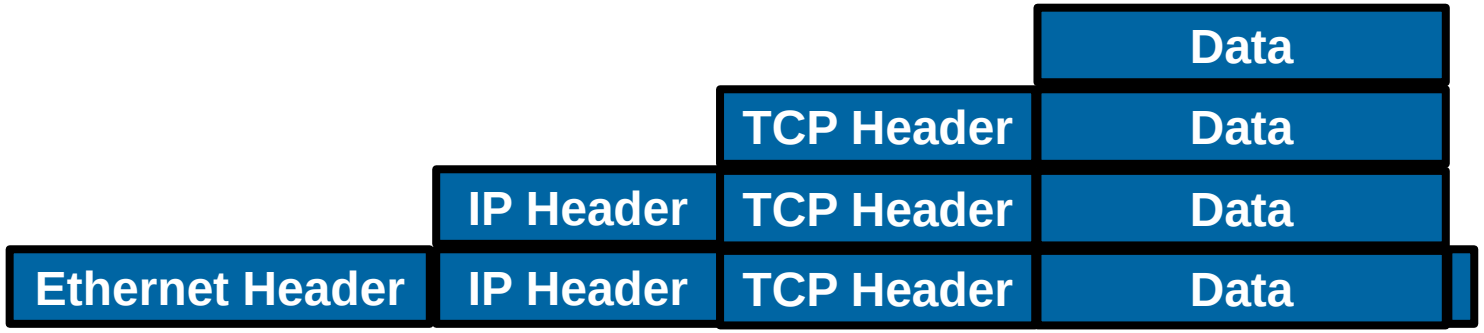
Distributed Systems - In the *News*

- **02.09.2019: Hong Kong Protestors Using Mesh Messaging App China Can't Block: Usage Up 3685%**
  - Bad Internet connectivity: use a meshed P2P network
  - "On Sunday, thousands of protesters gathered in front of the US embassy in central Hong Kong to urge US officials and politicians to take diplomatic action against the city's government. As expected, mobile internet speeds ground to a halt as the area filled up with people." [source]
  - Bridgefy app: bluetooth, hop-by-hop
    - + it worked
    - - nobody used it, Telegram more active
    - - arrival time of messages mixed up
- **17.09.2019: The First Yearlong ICO for EOS Raised $4 Billion. The Second? Just $2.8 Million**
  - ICO – difficult to raise money
  - 2018, raised 4.1b USD (1y)
  - 2019, 2.8m USD (1/2y)
- **16.09.2019: Facebook's David Marcus Responds to Critics Over Libra 'Threat'**
  - Threat to nations' "monetary sovereignty."
  - 12.09.2019: France Says It Will Block Facebook Libra in Europe
  - Libra backed 1:1 by a basket of strong currencies
- **22.08.2019: Æternity State Channels: A Peer-to-peer Browser Game**
  - Idea for challenge task: Reversi
  - Uses aeternity blockchain, state channels

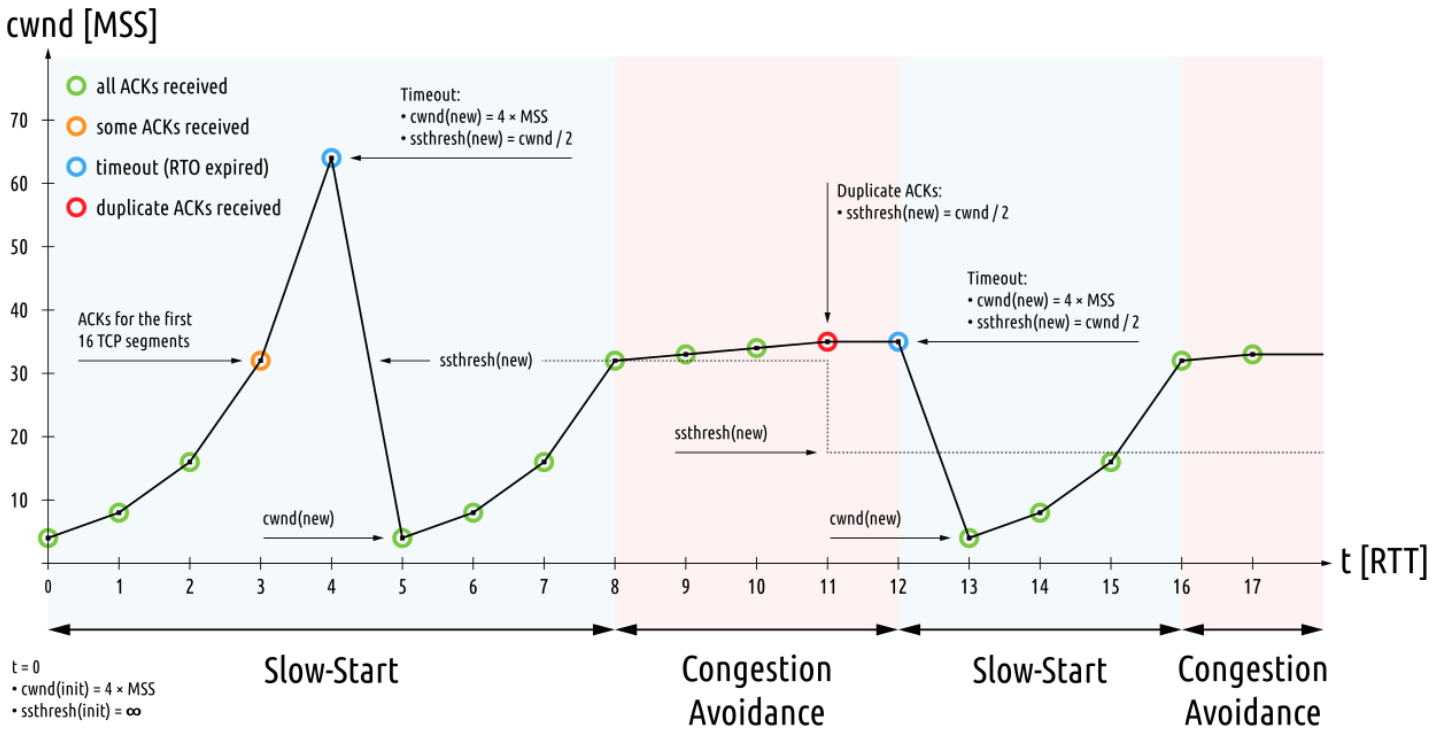
VSS W07: Networking: Layers

- **Networking: Each vendor had its own proprietary solution - not compatible with another solution**
- **Nowadays most vendors build compatible networks hardware/software from different vendors**
- **Goal of layers: interoperability**
- **1984: ISO 7498 - The Basic Reference Model for Open Systems Interconnection**

OSI model	"Internet model"
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data link	Link
Pysical	

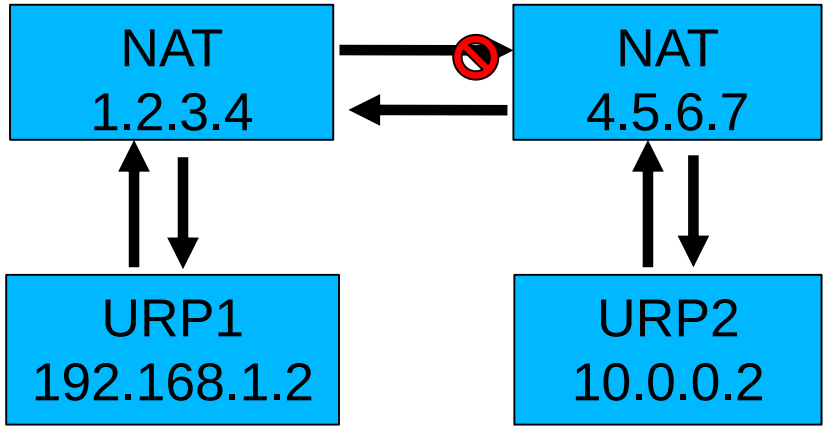


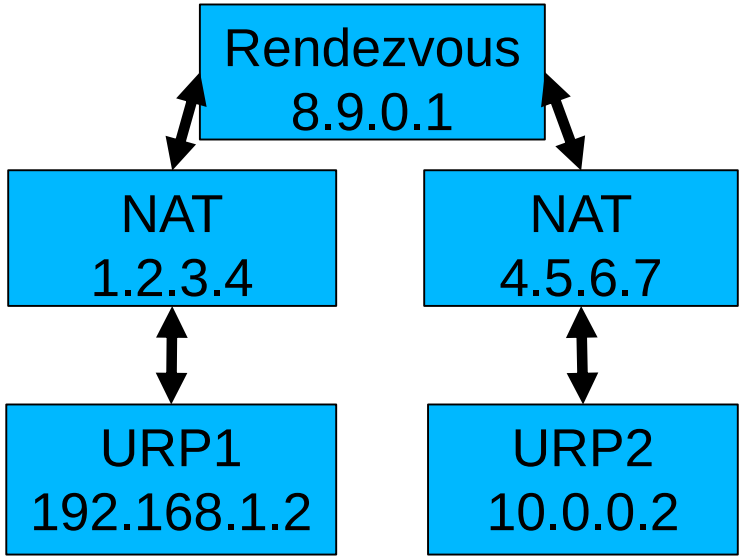
- **Flow control**
  - Sender is not overwhelming a receiver
  - Back pressure
  - Sliding window:
    - Receiver specifies the amount of additionally received data in bytes that can be buffered
    - Sender up to that amount of data before ACK
- **Congestion control**
  - slow-start
  - congestion avoidance
- **Difference flow/congestion control**
- **Wireshark - example**



https://upload.wikimedia.org/wikipedia/commons/thumb/2/24/TCP_Slow-Start_and_Congestion_Avoidance.svg/1280px-TCP_Slow-Start_and_Congestion_Avoidance.svg.png

- **Hole punching**
  - URP1 got 4.5.6.7:5000, URP2 got 1.2.3.4:4000
  - Unreachable peer 1 request to NAT 4.5.6.7, will fail – no mapping, however, unreachable peer 1 creates mapping with that request
  - Unreachable peer 2 sends request to unreachable peer 1 (1.2.3.4:4000) success!





Mapping for NAT 1.2.3.4 (Unreachable peer 1)			
192.168.1.2:4000	4.5.6.7:5000	4.5.6.7:5000	1.2.3.4:4000

Mapping for NAT 4.5.6.7 (Unreachable peer 2)			
10.0.0.2:5000	1.2.3.4:4000	1.2.3.4:4000	4.5.6.7:5000

URL: b.socrative.com/student/

Room: HSR

VSS W08: Comparison of Lookup Concepts

System	Per Node State	Communi-cation Overhead	Fuzzy Queries	No false negatives	Robust-ness
Central Server	O(N)	O(1)	✓	✓	✗
Flooding Search	O(1)	O(N)	✓	✗	✓
Distributed Hash Tables	O(log N)	O(log N)	✗	✓	✓

VSS W08: Kademlia Example

- **2^3, max size 8, #6 searches for 3**
  - Neighbors of 6, if k=1

1	2	3
7	4 (or 5)	0 (or 1, 2)

Routing Table of #6
6 xor 3 = 101b

  - Search for 3, ask 0, neighbors of 0

1	2	3
1	2	4 (or 5, 6, 7)

Routing Table of #0
0 xor 3 = 11b

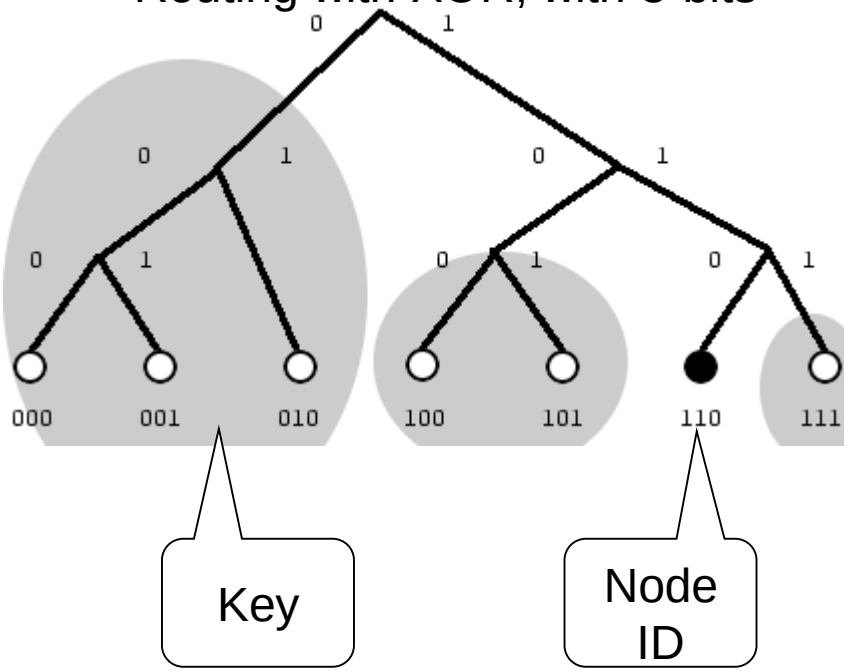
  - Ask 2, neighbors of 2

1	2	3
-	0 (or 1)	4 (or 5, 6, 7)

Routing Table of #2
2 xor 3 = 1b

  - Ask 2, 2 replies 0. 6 figures that there is no closer node, 2 is the closest one (2 xor 3 =1)

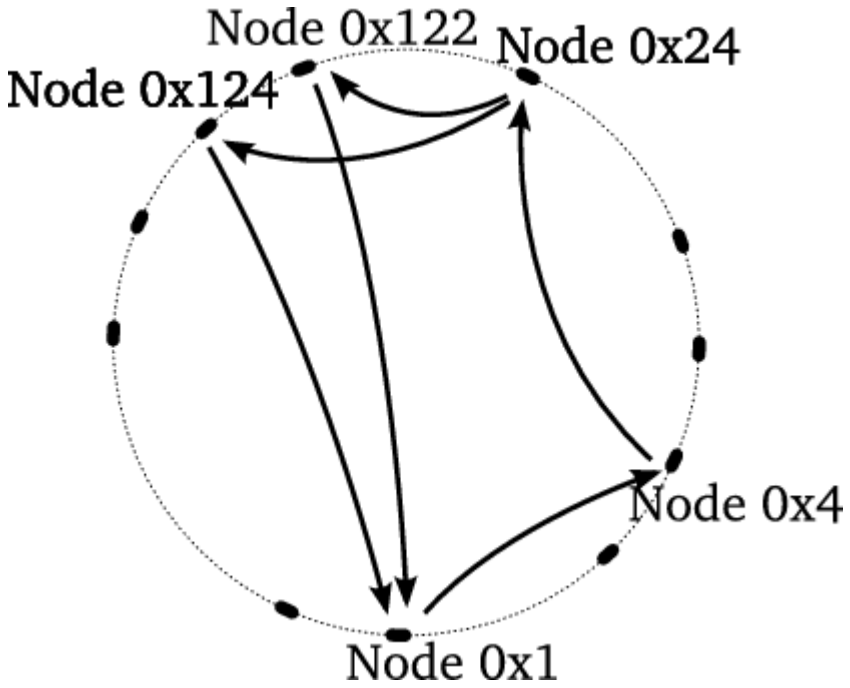
Routing with XOR, with 3-bits



URL: b.socrative.com/student/

Room: HSR

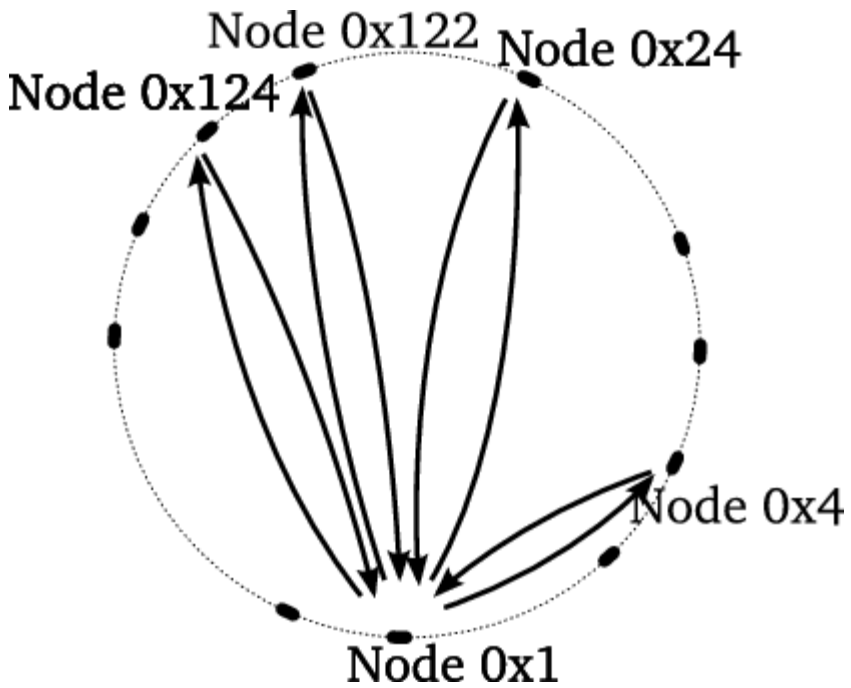
- **Recursive routing**



+ online status update

- faulty peers cause delay

- **Iterative routing**

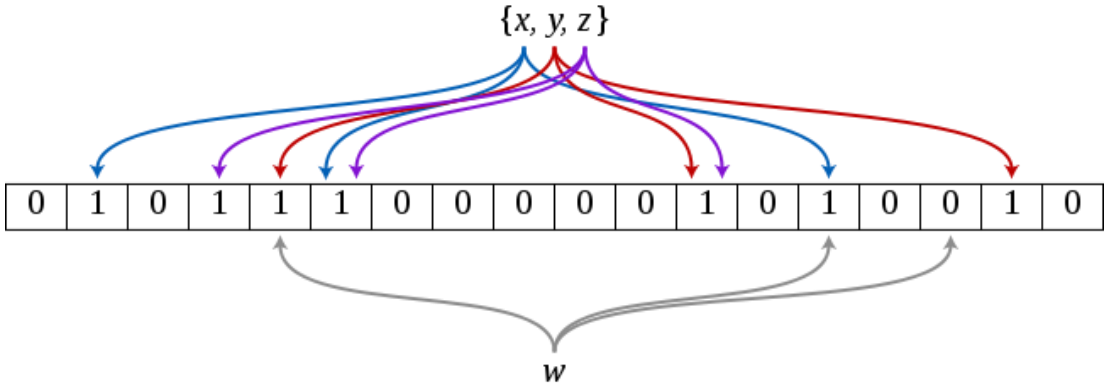


+ control

- neighbor maintenance

VSS W09: Query of an Element, m=18, k=3

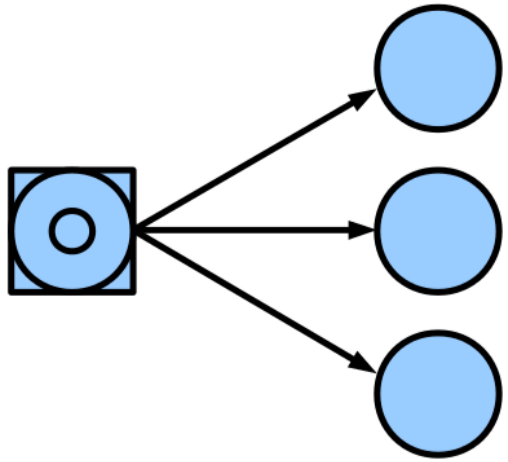
- **Insert x, y, z**
- **Query w**

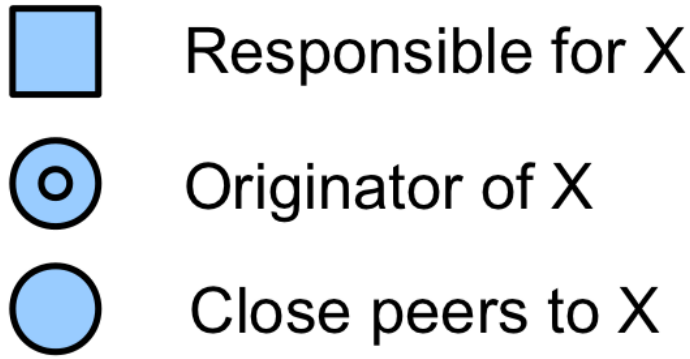


http://en.wikipedia.org/wiki/Bloom_filter

- **Example for False-positives**
  - Insertions
    - Hash („color printer“) => (1,4,6)
    - Hash („digital camera“) => (3,4,5)
    - Bloom filter (1,3,4,5,6)
  - Query
    - Hash („heat sensor“) => (3,4,6)
    - Matches since bits 3,4,6 are all set to 1
  - Online
- **False-negative**
  - Query
    - Hash (“color printer”) => (1,4,6) , matches (1,3,4,5,6) → no false-negative

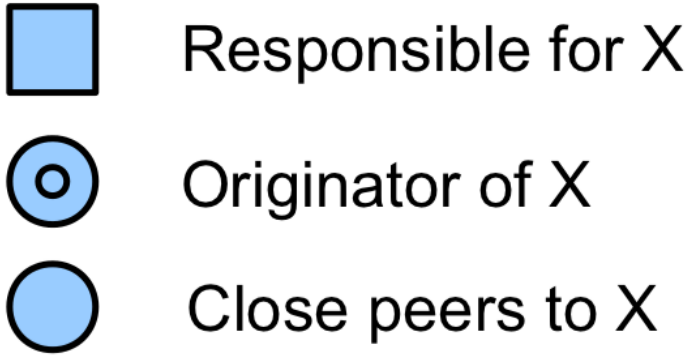
- **Direct Replication**
  - Enough replicas
    - Originator peer is responsible
    - Periodically refresh replicas
    - Example: tracker that announces its data

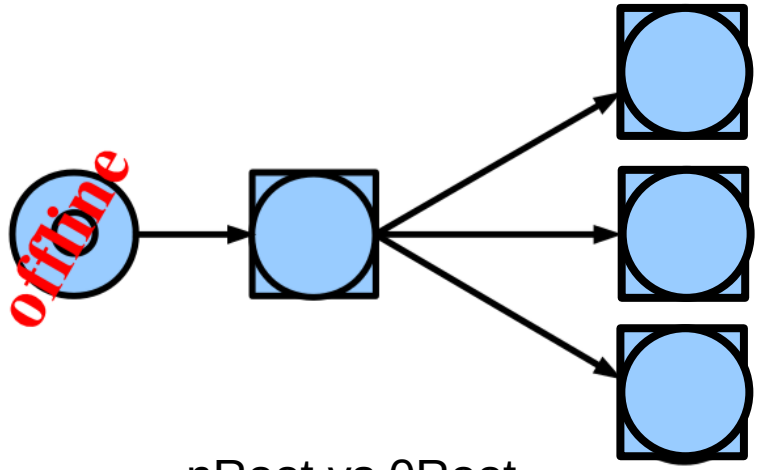




- **Problem**
  - Originator offline → replicas disappear. Content has TTL

- **Indirect Replication**
  - The closest peer is responsible, originator may go offline (0Root)
    - Periodically checks if enough replicas exist
    - Detects if responsibility changes





nRoot vs 0Root

- **Problem**
  - Requires cooperation between responsible peer and originator
  - Multiple peers may think they are responsible for different versions → eventually solved

URL: b.socrative.com/student/

Room: HSR

Questions?

Dr. Thomas Bocek thomas.bocek@hsr.ch