INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING
ICPP/2021/CHICAGO/USA
acm In-Cooperation
sighpc
AUGUST 9-12, 2021

Multi-level Forwarding and Scheduling Recovery Technique in Heterogeneous Network for Erasure-coded Clusters

Hai Zhou, **Dan Feng**, Yuchong Hu

Huazhong University of Science and Technology

ICPP 2021

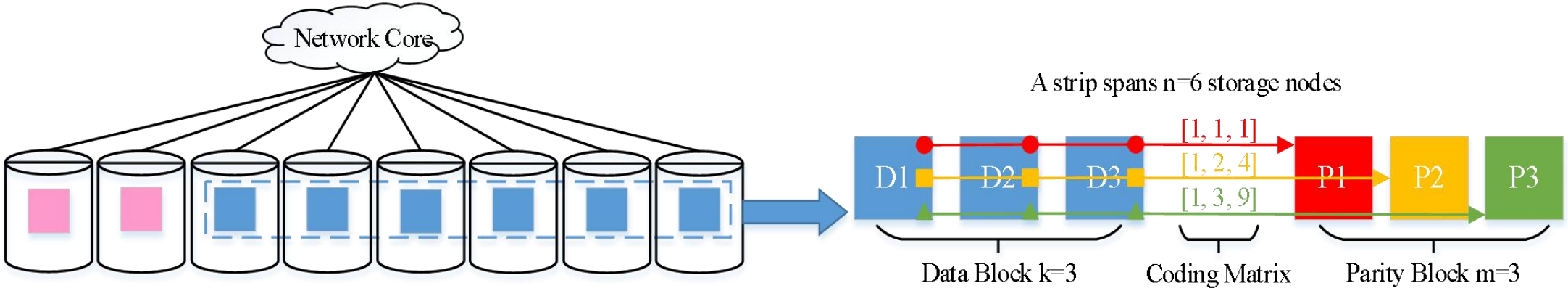
Speaker: Hai Zhou

Introduction

- Fault tolerance for distributed storage is critical
  - Failures are common
  - Data remains accessible under failures
  - No data loss even under failures
- **Erasure coding** is a promising redundancy technique
  - Erasure coding is a promising redundancy technique
  - Higher reliability with same storage redundancy than replication
  - Reportedly deployed in Google, Azure, Facebook

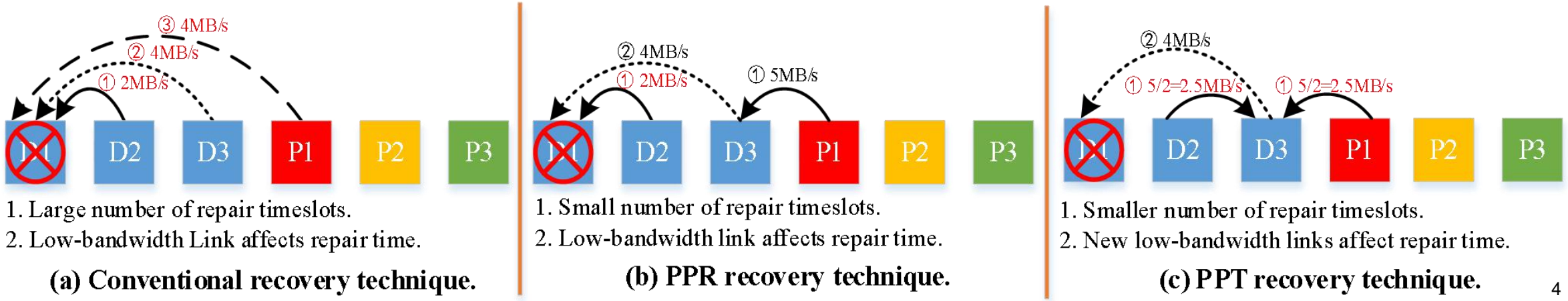
Erasure Coding

- Erasure-coded Clusters
  - Modern data centers deploy erasure codes in clusters composed of multiple nodes
  - The cluster's nodes are interconnected via the network core
- Construction of erasure coding:
  - **Encode**: Generate n-k=**m** parity blocks from **k** data blocks
  - **Decode**: Any k out of **n** coded blocks can recover original file



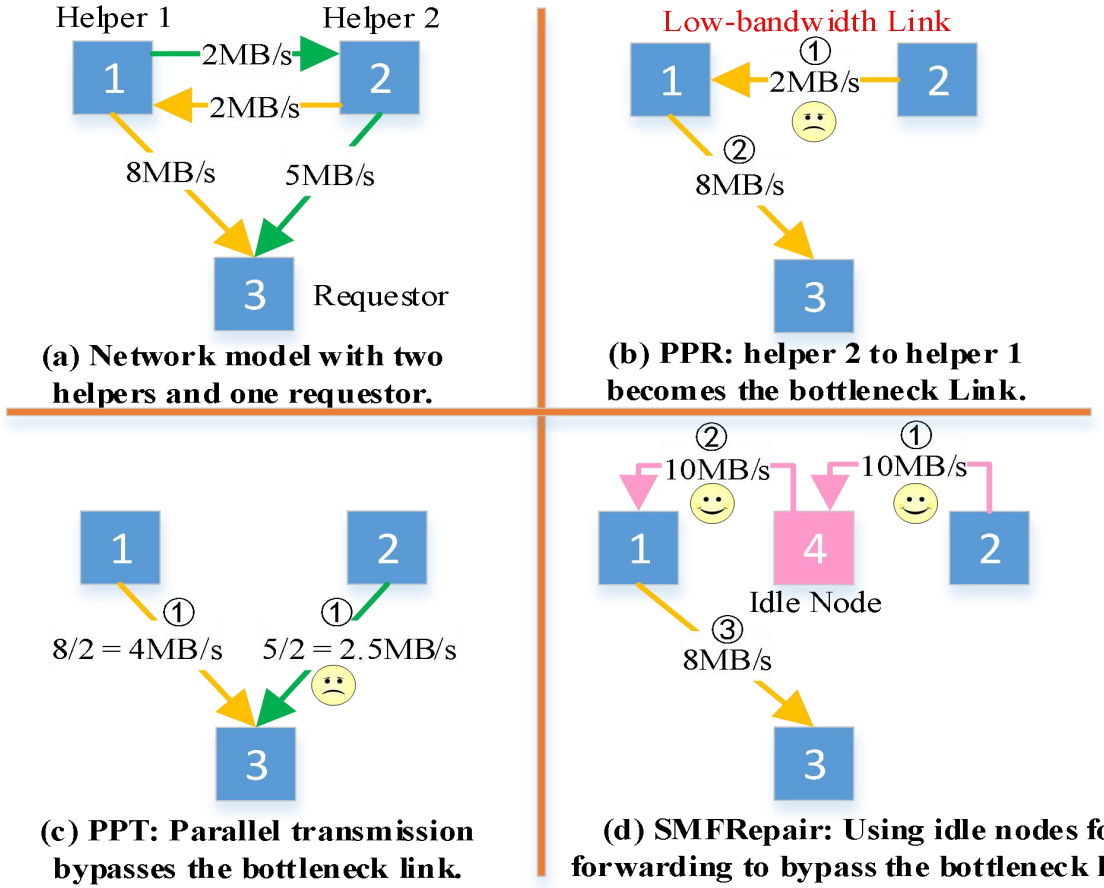
Repair in Clusters

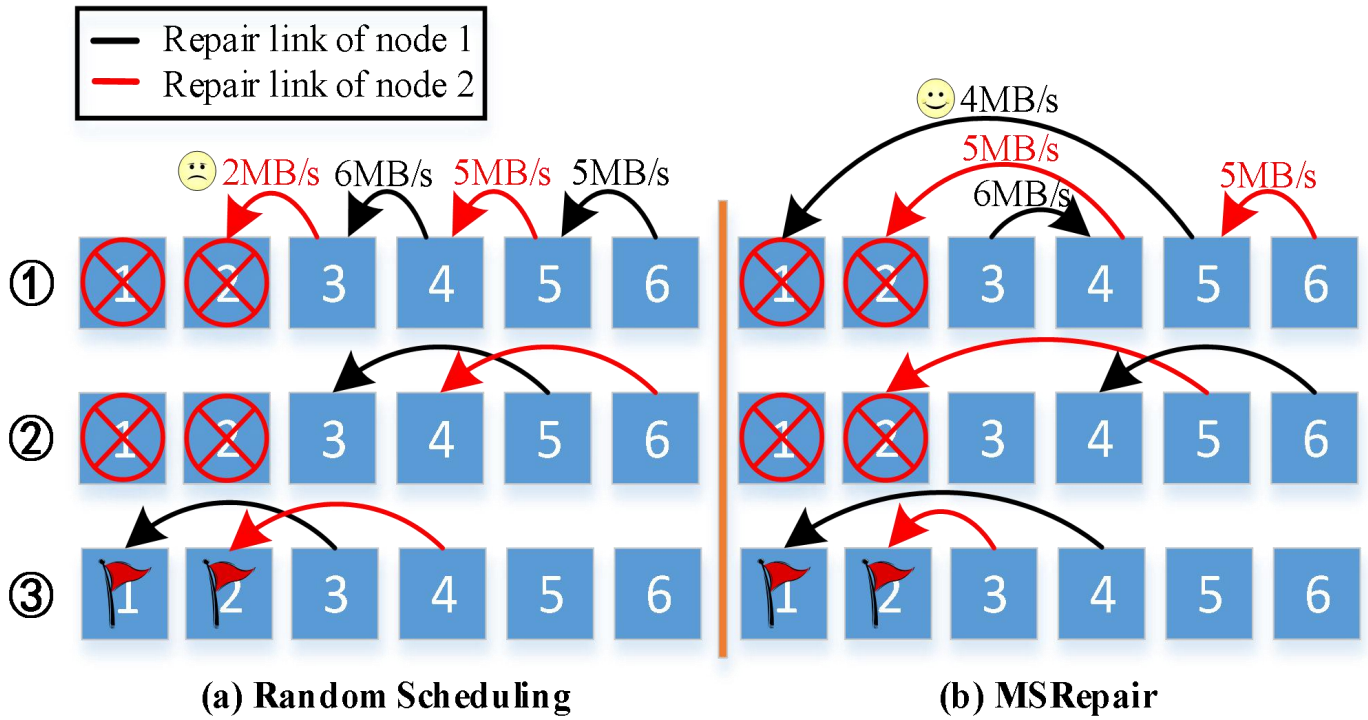
- Erasure codes incur high network traffic consumption and recovery time in failure repair
  - The bandwidth between each node is heterogeneous where link bandwidths vary in a wide range
- Recovery time is always limited by the **lowest-bandwidth link**
  - **Conventional Recovery**: retrieves k blocks from k nodes
  - **PPR**[EurSys'16]: decomposes a repair operation into multiple partial operations
  - **PPT**[ICPP'19]: utilizes a special bandwidth gap to bypass the low-bandwidth link



Observation

- **Observation 1**: **Idle nodes** can bypass the low-bandwidth links.
- **Observation 2**: Reasonably scheduled repair links can avoid the low-bandwidth link for multi-node recovery.





Our Contributions

- Propose **Single-node Multi-level Forwarding Repair** (SMFRepair):
  - Uses idle nodes outside the stripe to bypass the lowest-bandwidth link.
  - Occupys the bandwidth resources of a single link without incurring network congestion and competition.
- Propose **Multi-node Scheduling Repair** (MSRepair):
  - Finds a recovery solution that schedules the parallel repair of multi-node and transfers data from as large bandwidth links as possible.
  - Analyze theoretically that MSRepair can minimize the recovery time.
- Evaluate SMFRepair and MSRepa in a simulation and real cluster:
  - Reduced 36.65% of single-node recovery time and 55.10% of multi-node recovery time.

Design of SMFRepair

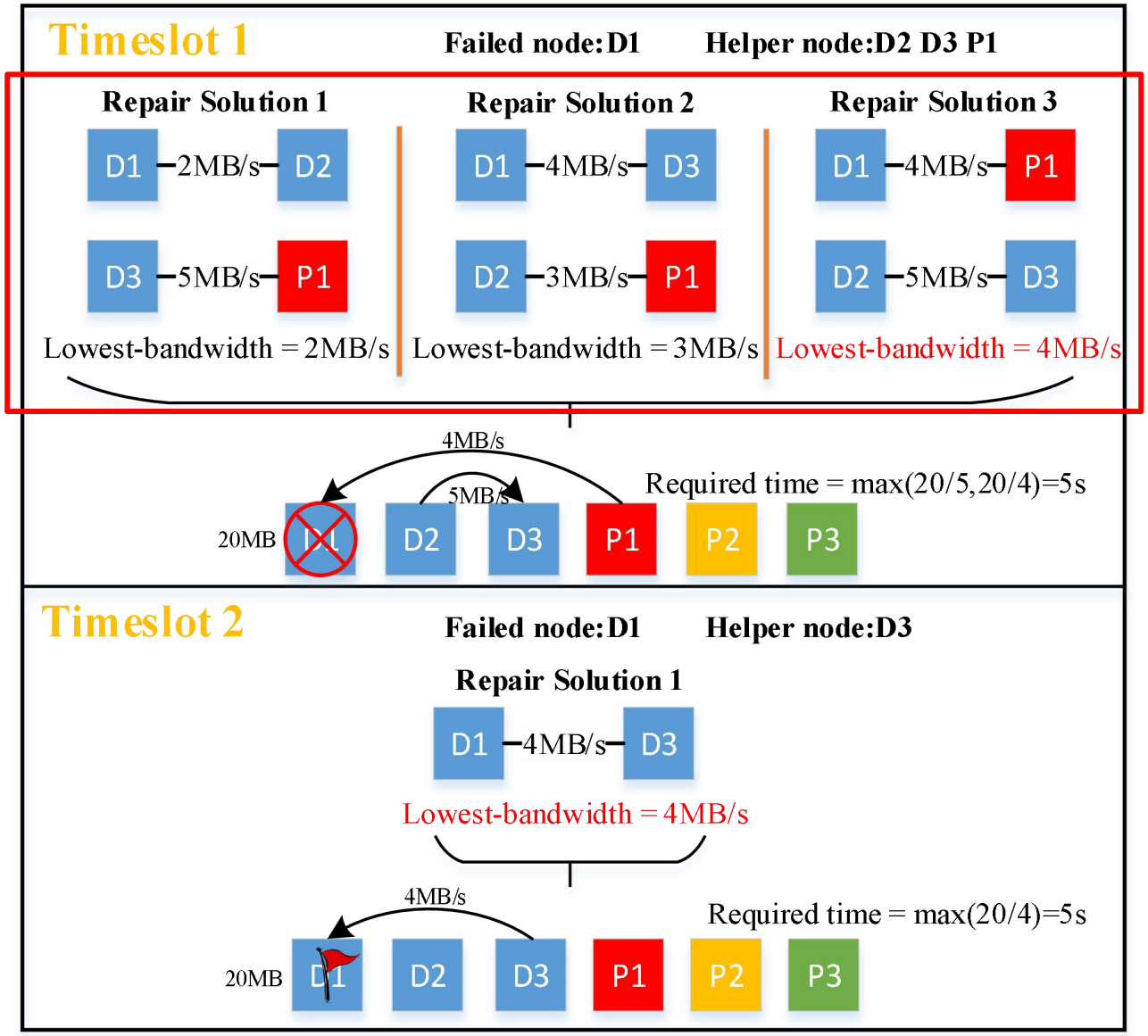
- Step 1: Generate the *Source-stripe single-level forwarding* links based on the bandwidth between nodes.
  - Algorithm finds a repair solution within the stripe through enumeration, which consumes the least time for each timeslot.
- Step 2: Find the lowest-bandwidth link at each timeslot, and generate a *Nonsource-stripe multi-level forwarding* link to replace.
  - Algorithm uses idle nodes to bypass the lowest-bandwidth link by constructing a repair link tree.

•**Single-level forwarding:** The source node data directly transfers to the target node.
•**Multi-level forwarding:** The source node data is transferred to the target node after passing through several nodes.
•**Source-stripe forwarding:** When repairing the failed node, the link forwarding only passes through helper nodes in source stripe.
•**Nonsource-stripe forwarding:** When repairing the failed node, the link forwarding also passes through idle nodes in non-source stripe.

Design of SMFRepair

- Step 1: enumerate all possible repair solutions for each timeslot, compare the lowest bandwidth of all repair solutions, and select the repair solution with the largest lowest-bandwidth.

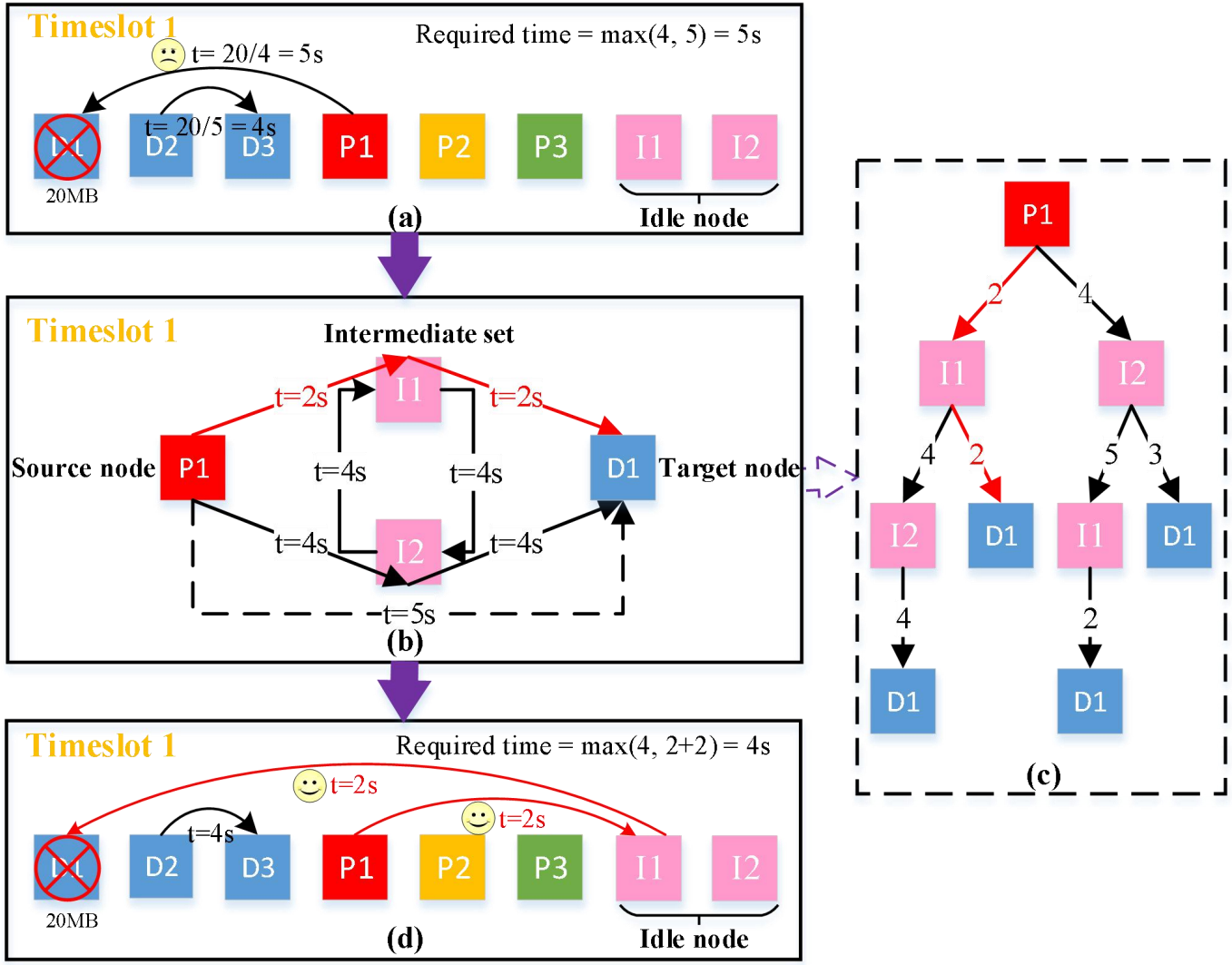
Example: we select D2, D3, and P1 to repair D1. The lowest-bandwidth link of repair solution 3 is 4MB/s, larger than that of 2MB/s of repair solution 1 and that of 3MB/s of repair solution 2.



Design of SMFRepair

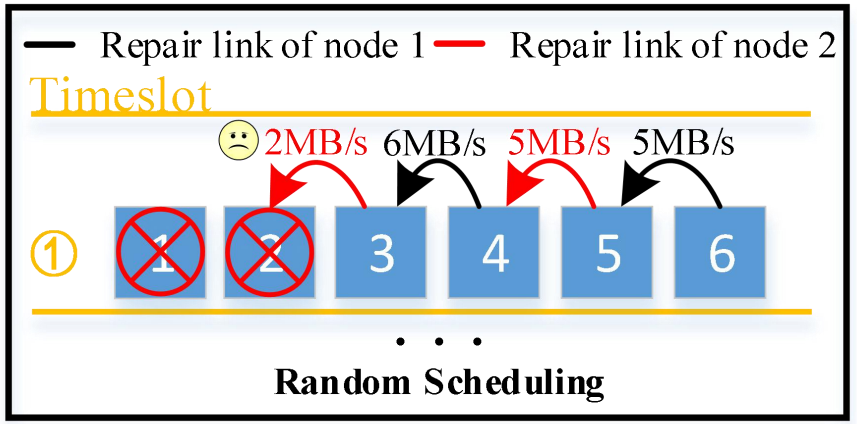
- Step 2: uses idle nodes to bypass the lowest-bandwidth link by constructing a repair link tree.

Example: timeslot 1 takes max(20/5, 20/4) = 5s because of lowest-bandwidth link P1→D1. Using idle node I1 and I2 to bypass bottleneck link. In figure(c), link P1→I1→D1 can be found to replace.

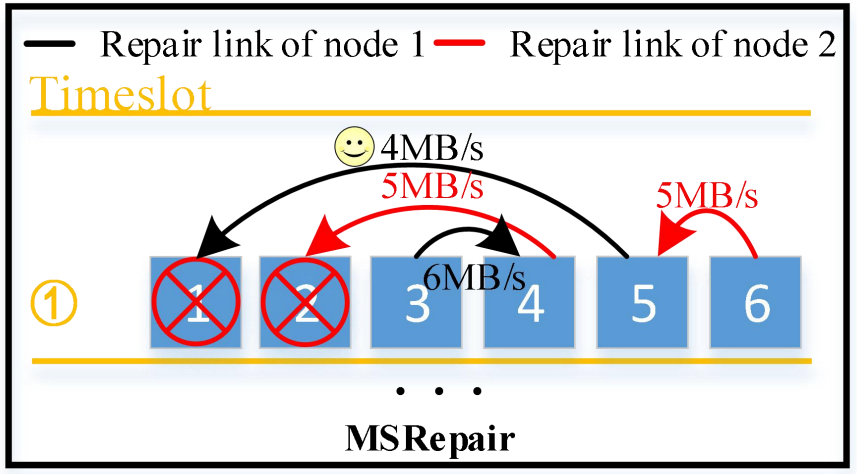


Design of MSRepair

- Use full-duplex technology to allow each node to receive and send data simultaneously
  - **Random scheduling** can cause low-bandwidth links

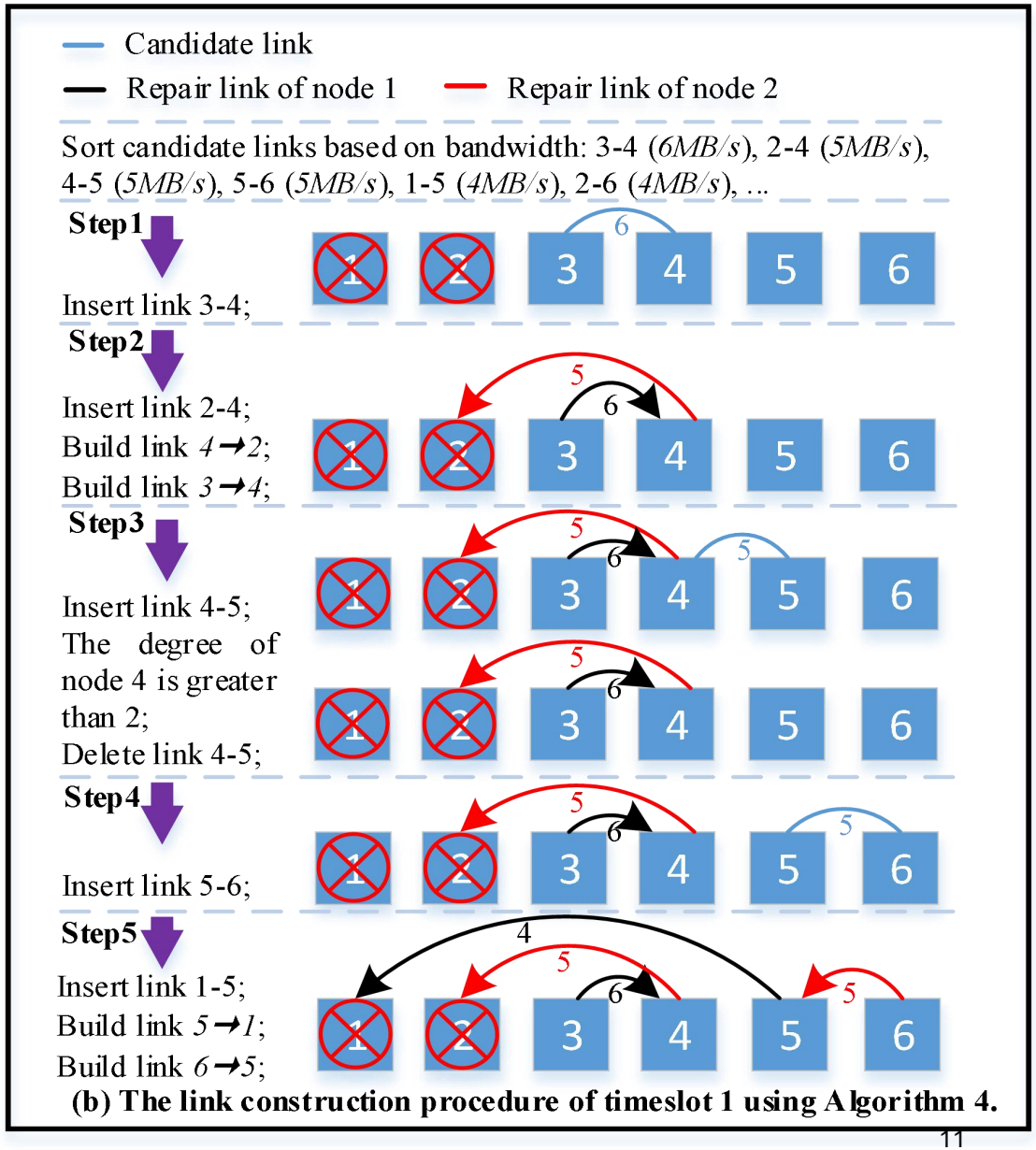


- The low-bandwidth link can be avoided by rationally scheduling parallel repairs of multi-node
  - **MSRepair** designs a fast greedy algorithm to maximize the lowest-bandwidth link



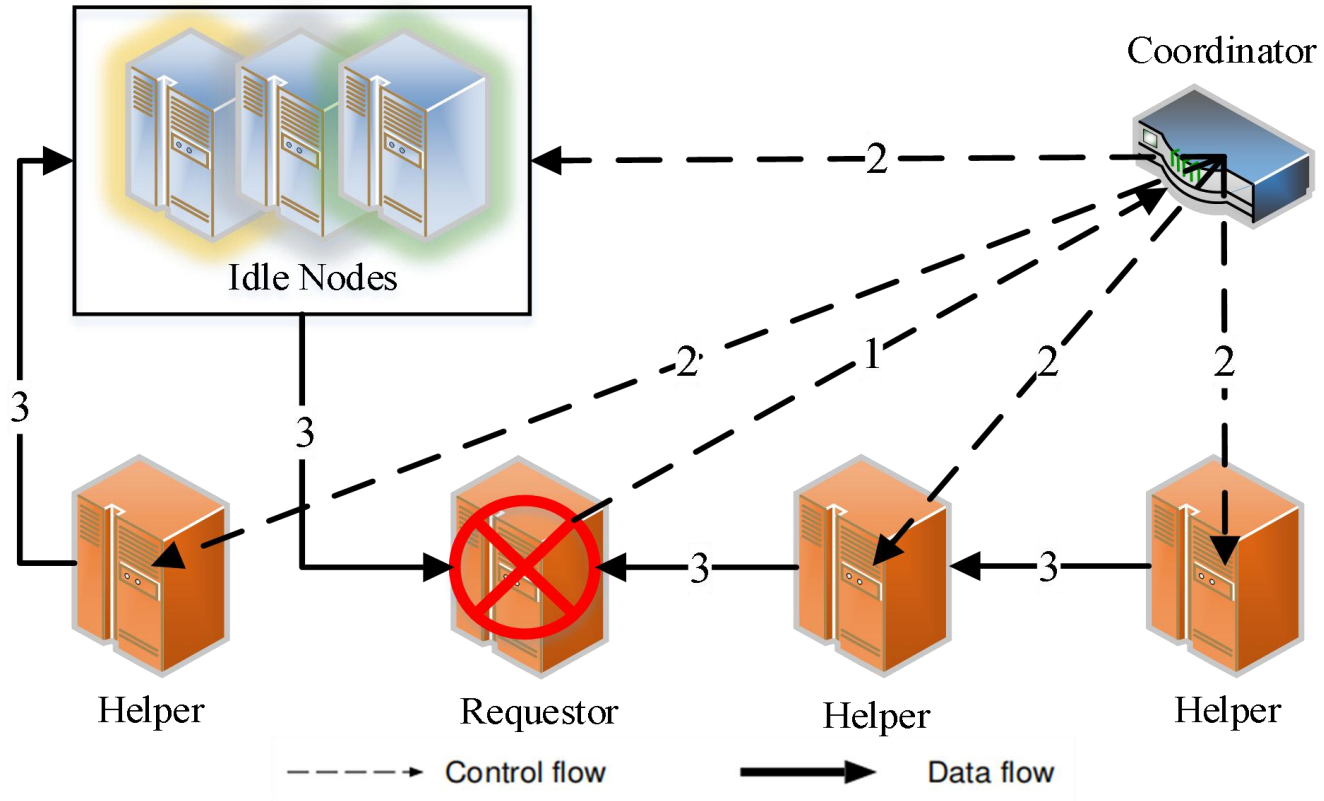
Design of MSRepair

- Sorts the candidate link bandwidth from largest to smallest
- Links with large bandwidth will be inserted first to maximize the lowest-bandwidth link
- Determine the link direction and repair tasks according to the node status of the candidate link



(b) The link construction procedure of timeslot 1 using Algorithm 4.

System Implementation

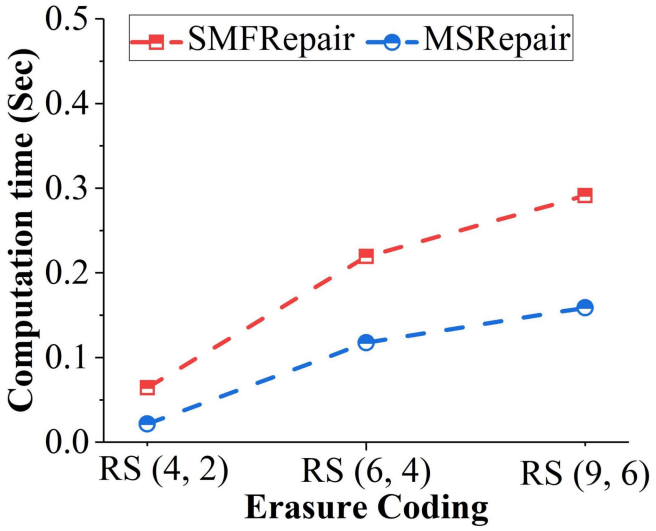


- Written in Python on simulation platform and in C++ on real platform
- Erasure coding is implemented via Jerasure v1.2 and use TCP sockets to realize network transmission
- Each node also creates different threads to process control flow and data flow

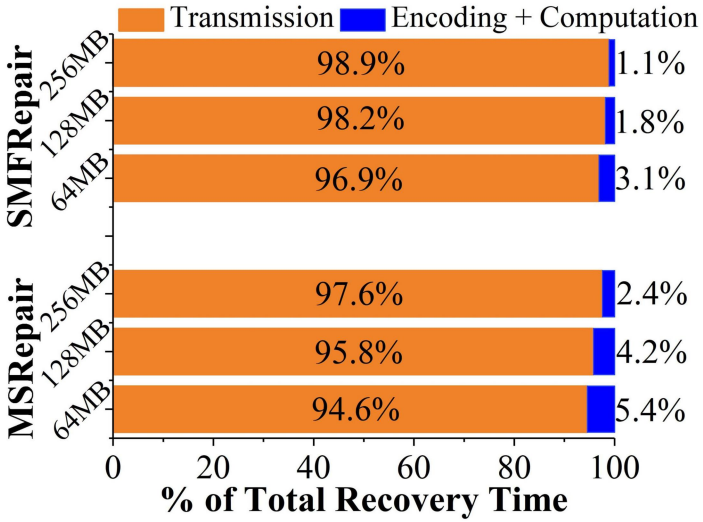
Evaluation

- Pareto distribution to simulate the heterogeneous network
- The number of idle nodes: 5 (for repair forwarding only)
- Node size: 128MB
- Compared techniques
  - **Single-node**: PPR [EurSys'16], PPT [ICPP'19]
  - **Multi-node**: Random, PPCT [ICPP'19]

Computation and Transmission Time



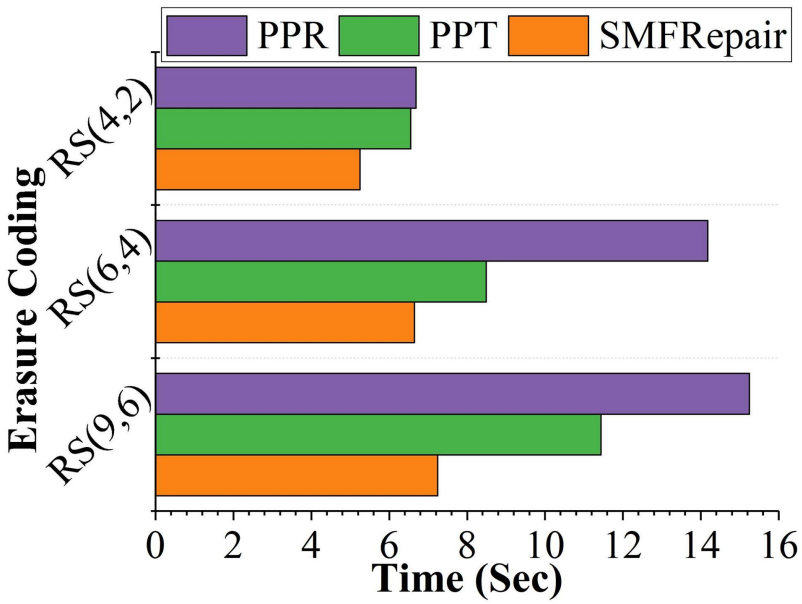
(a) Computation time



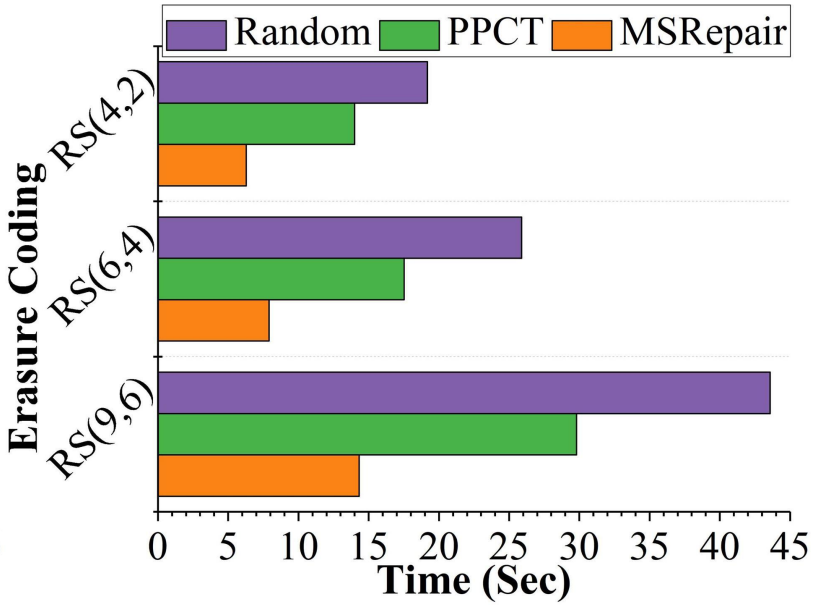
(b) Ratio of transmission time

- Both techniques do not exceed 0.3 seconds, which are extremely efficient to derive the repair solutions.
- The ratio of transmission time from 96.9% to 98.9% for SMFRepair, and from 94.6% to 97.6% for MSRepair.

Coding Parameters



(a) Single-node recovery



(b) Multi-node recovery

- SMFRepair reduces single-node recovery time by up to 36.65%.
- MSRepair reduces multi-node recovery time by up to 55.10%.

Conclusions

- First propose that idle nodes are used to participate in the recovery process
- Propose SMFRepair and MSRepair techniques for single-node and multi-node recovery
- Implement SMFRepair and MSRepair in simulation and real platforms
- Show SMFRepair and MSRepair efficiency in repair

THANK YOU

Contacts:
Hai Zhou haizhou@hust.edu.cn