Design and Modeling of a New File Transfer Architecture to Reduce Undetected Errors Evaluated in the FABRIC Testbed

Prateek Jain, Boston University
Arash Sarabi, Arizona State University
Abraham Matta, Boston University
Violet R. Syrotiuk, Arizona State University

Motivation

- Scientific instruments generate petabytes of data
- Errors can result in inaccurate interpretations especially when events are critical or rare
  - e.g., finding the Higgs Boson or gravitational waves by the LHC at CERN
  - Astronomical data with NASA
  - Medical images
- Traditional error detection — CRC at DLL and Internet checksum at Transport Layer
- Sources of errors — network transmission or memory access or hardware problems
  - Statistically CRC is supposed to miss 1 in 4 billion errors
  - Practically between one packet in 16 million and one packet in 10 billion will have an error that goes undetected through TCP Checksum[1]
- Our Focus is large scale file transmissions:
  - Errors that arise from transmission in the network
  - Errors that occur in intermediate systems or at the source or at the destination
- We consider even a single bit error in the final transmitted file will render the file useless

[1] J. Stone and C. Partridge, "When the CRC and TCP checksum disagree," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '00)*, Stockholm, Sweden, 2000, pp. 309-319, doi: 10.1145/347059.347561.

Related Work

	XDD	FDT	GridFTP	Globus	BBCP	XRootD	BitTorrent	IBM Aspera	Effingo
L4 Protocol	TCP	TCP	TCP	TCP	TCP	TCP	TCP/uTP	UDP	TCP
File-Level	✗	✓	✓	✓	✓	✓	✗	✓	✓
Chunk-Level	✗	✗	✓	✗	✓	✓	✓	✓	✓
Method Used	Internet checksum	md5	Adler32, CRC-32, md5	md5	Adler32, CRC-32, md5	CRC32C, md5	sha-1	sha-{1,256, 384,512},md5	Hash Based

Related Work

	XDD	FDT	GridFTP	Globus	BBCP	XRootD	BitTorrent	IBM Aspera	Effingo
L4 Protocol	TCP	TCP	TCP	TCP	TCP	TCP	TCP/uTP	UDP	TCP
File-Level	✗	✓	✓	✓	✓	✓	✗	✓	✓
Chunk-Level	✗	✗	✓	✗	✓	✓	✓	✓	✓
Method Used	Internet checksum	md5	Adler32, CRC-32, md5	md5	Adler32, CRC-32, md5	CRC32C, md5	sha-1	sha-{1,256,384,512},md5	Hash Based

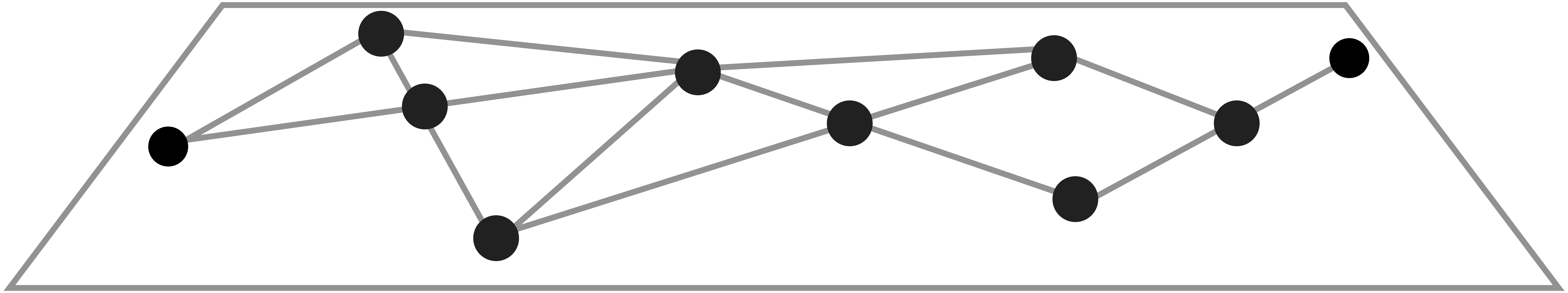
- End-to-End Chunk-Level Verification — failed — retransfer chunk
  - Verify integrity of data chunks during transfer
  - Chunk sizes range from 256 KB to 1 MB
  - Effingo uses larger chunks (8 MB to 64 MB)
- End-to-End File Level Verification — failed — retransfer file
  - Checks the entire file after transfer to ensure complete integrity
- Limitations of Existing Tools
  - In-network resources are not utilized
  - No decoupling of network functions such as security and error detection
  - Not flexible as per the network characteristics and user needs
  - None provide estimates for the **Undetected Error Probability (UEP)**

Multi Level Error Detection (MLED) Architecture

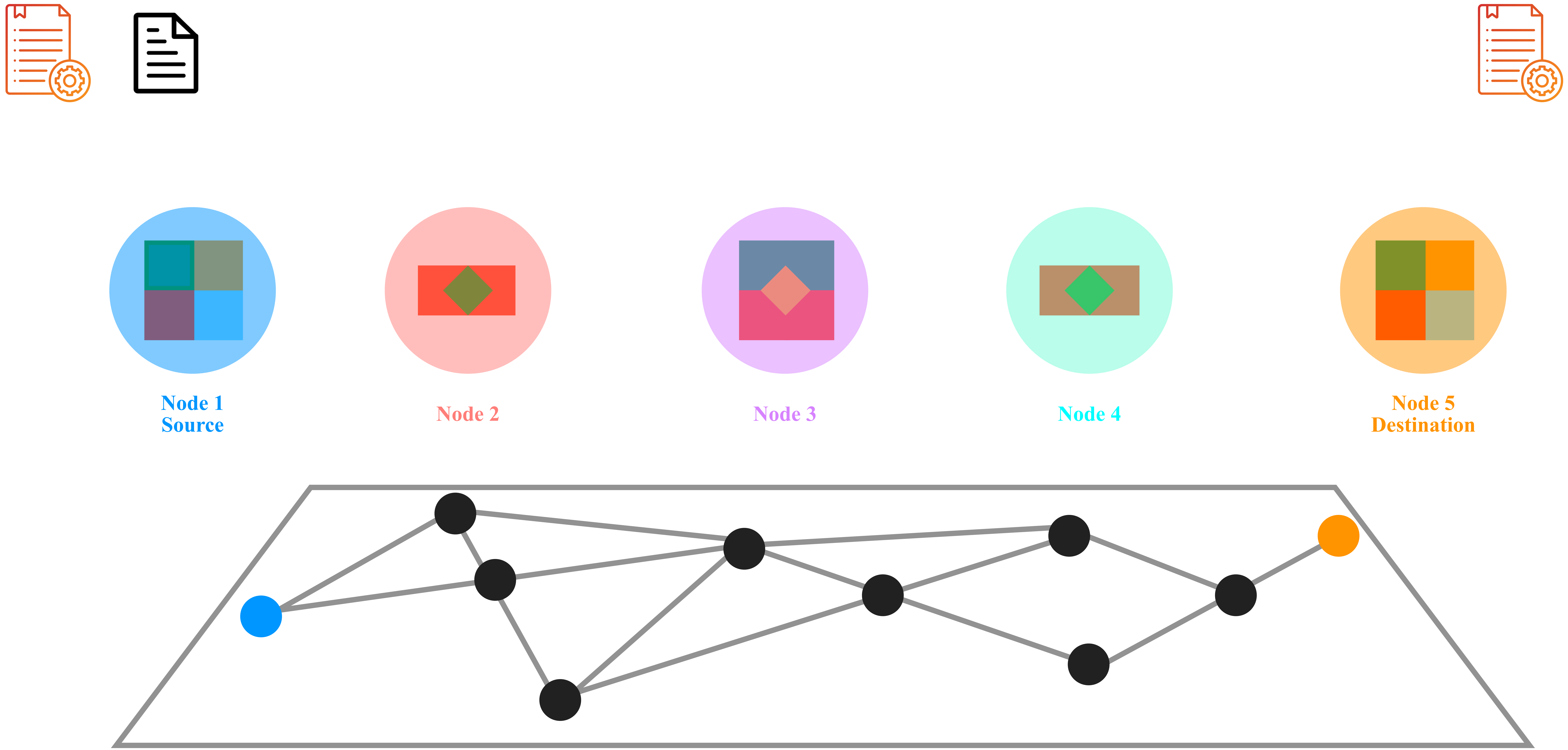
- MLED is a recursive framework based on Recursive Inter Network Architecture (RINA)
- It utilizes in-network resources for error detection and reduces UEP in large-scale file transfer
- Defined as **MLED(*n*, *P*)**, where:
  - $n \geq 3$ levels ensure recursive structure and differentiate architecture from traditional network stack with two level checks
  - Each level i has j layers defined as L_{ij} which implements a layer specific configurable policy P_{ij} over its scope
  - P is the set of all the policies
- Decouples various network functions for each layer — brings modularity and flexibility

S.No	Policy	Current Implementation
1	Error Detection	CRC8/16/32, TCP Checksum, Hash (MD5/SHA1)
2	Congestion Control	Cubic / BBR
3	Routing	Static
4	Flow Control	Sliding window similar to TCP
5	Recovery	Re-transmission (ARQ) upon NACK
6	Addressing	Static
7	Payload length	Any positive integer with constraint

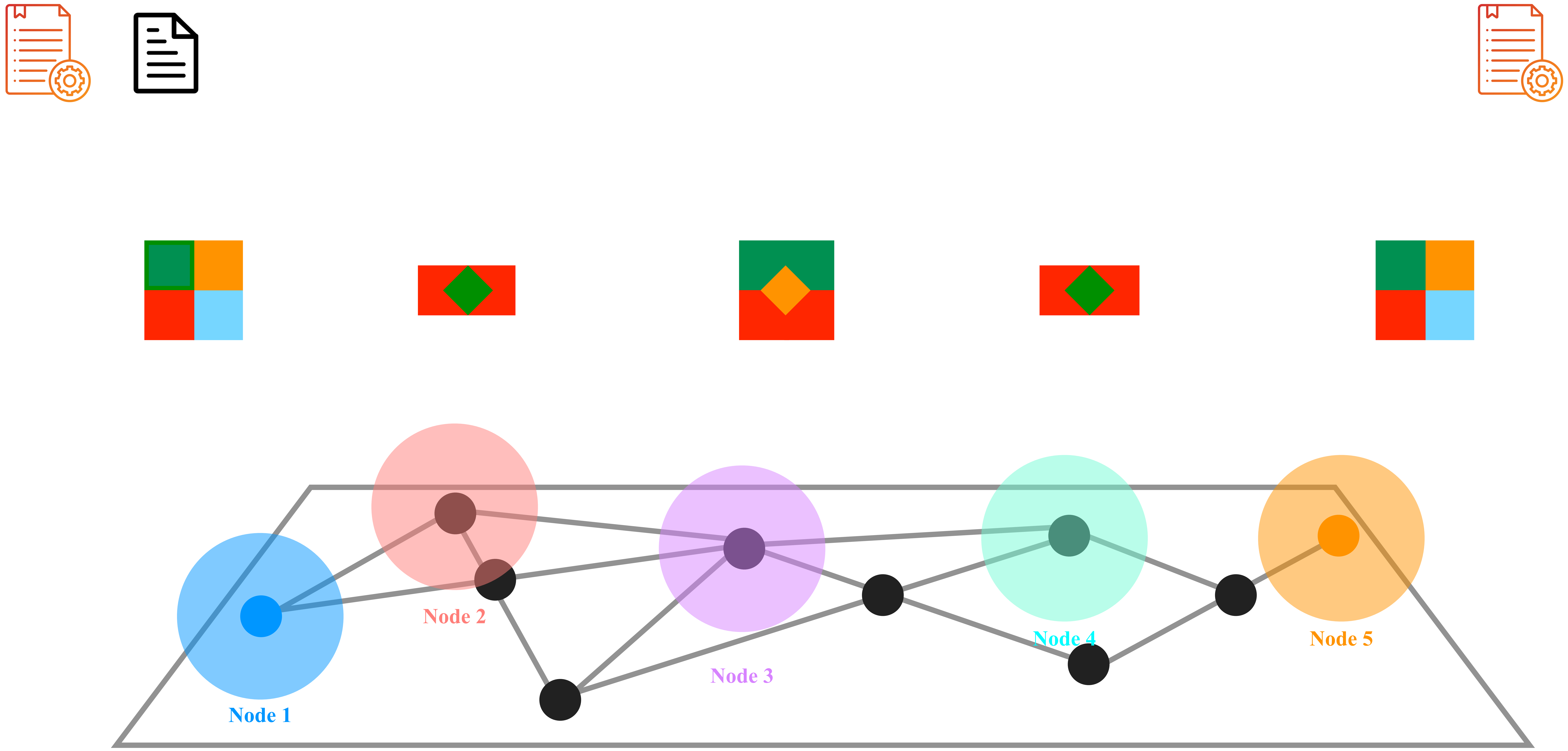
MLED Architecture



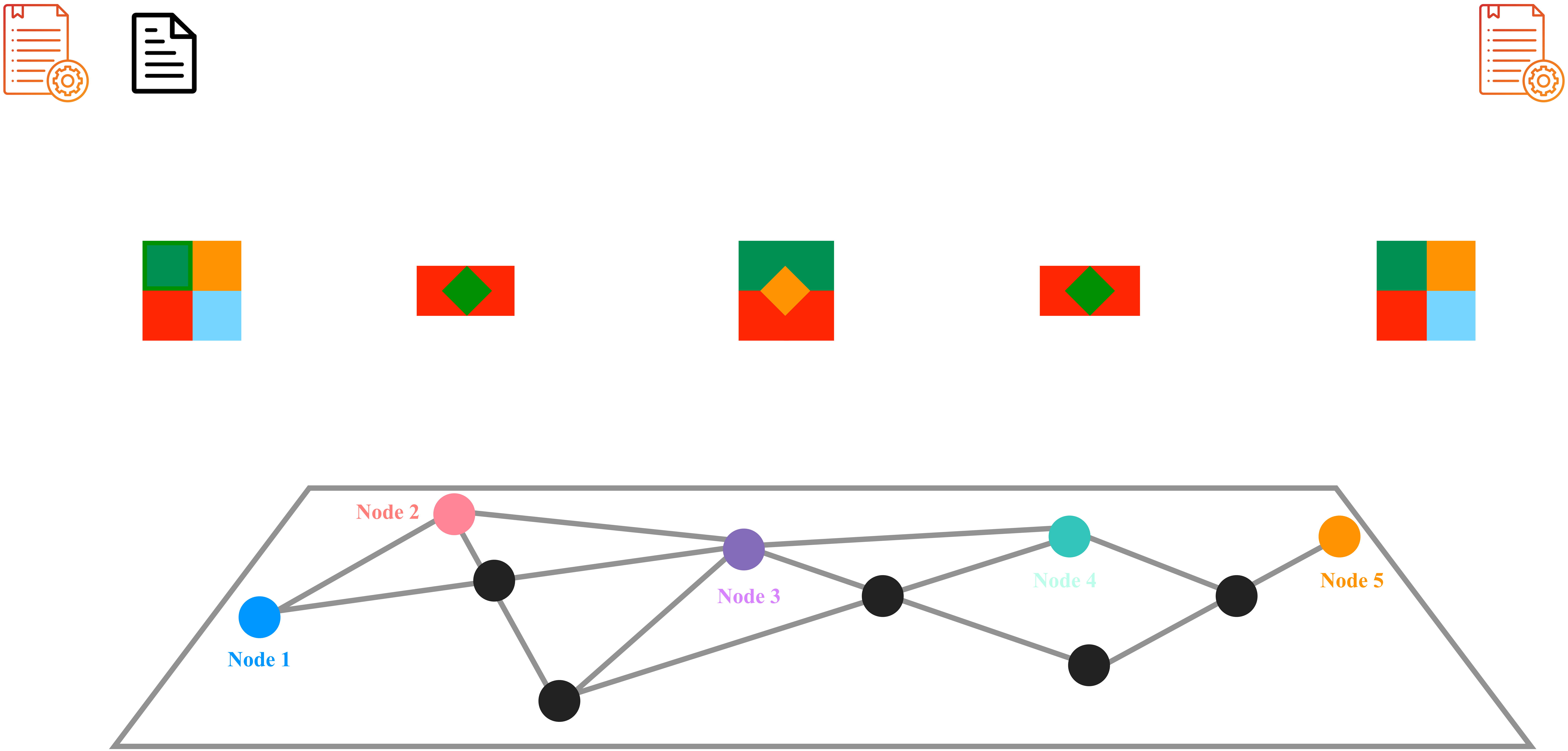
MLED Architecture



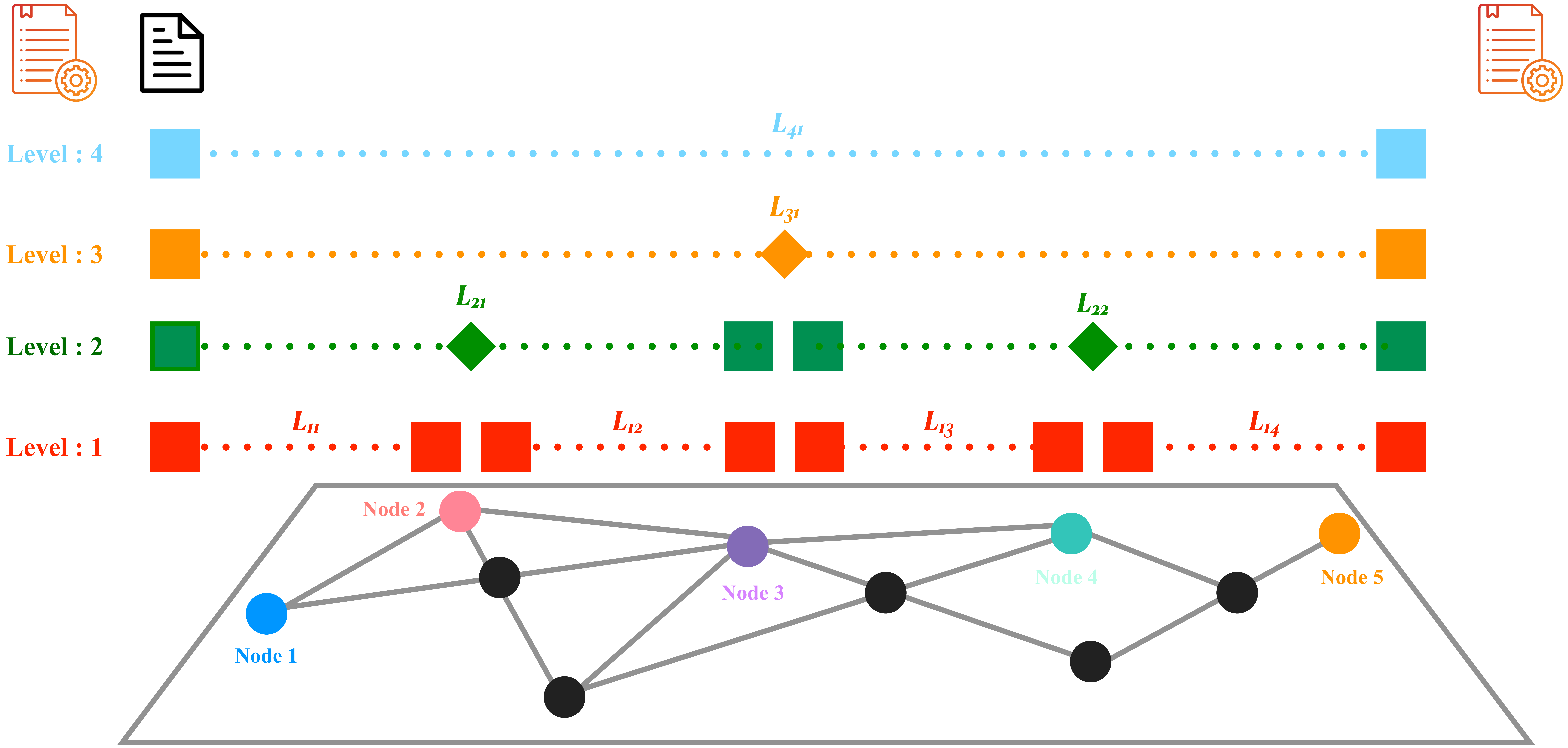
MLED Architecture



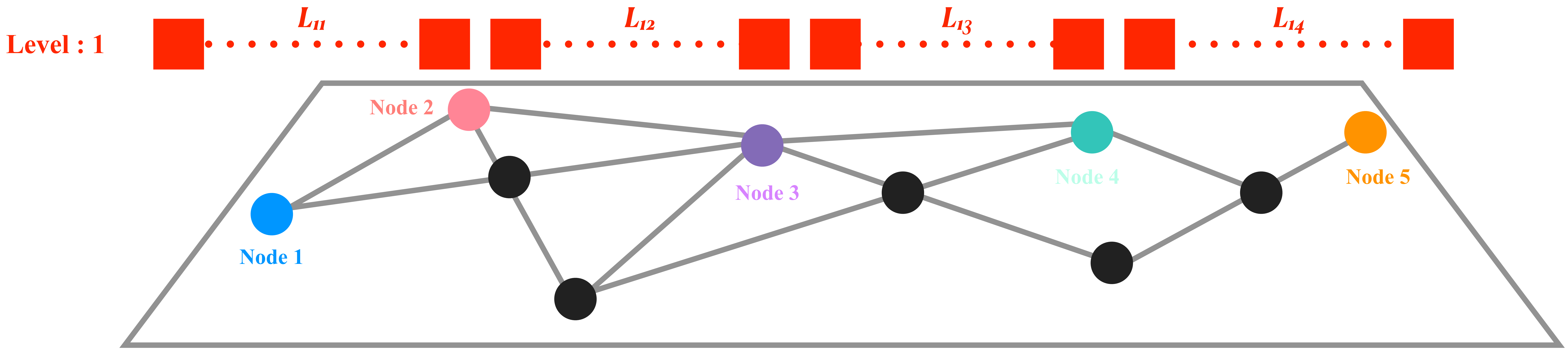
MLED Architecture



MLED Architecture

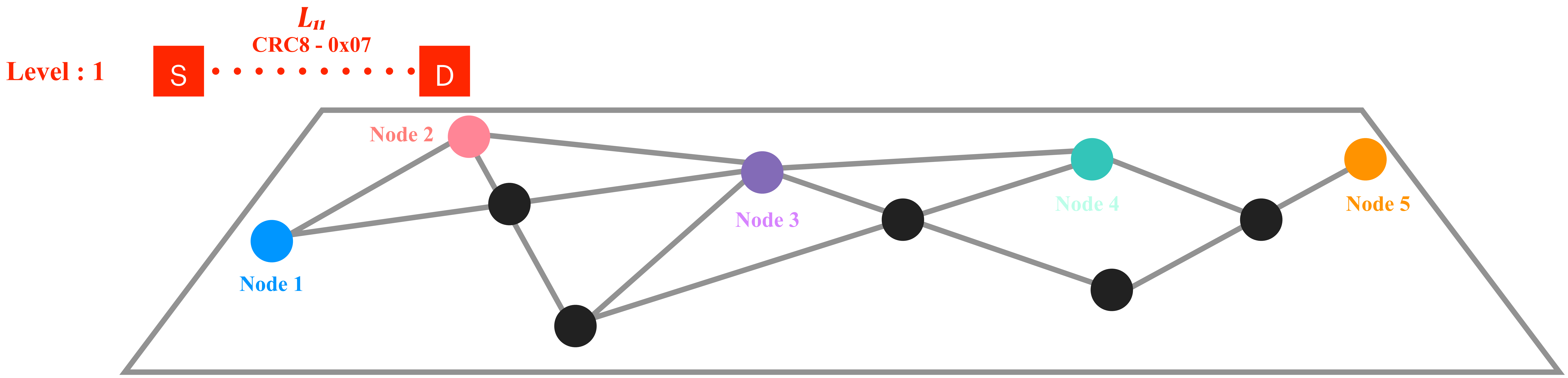


MLED Layers and Virtual Links



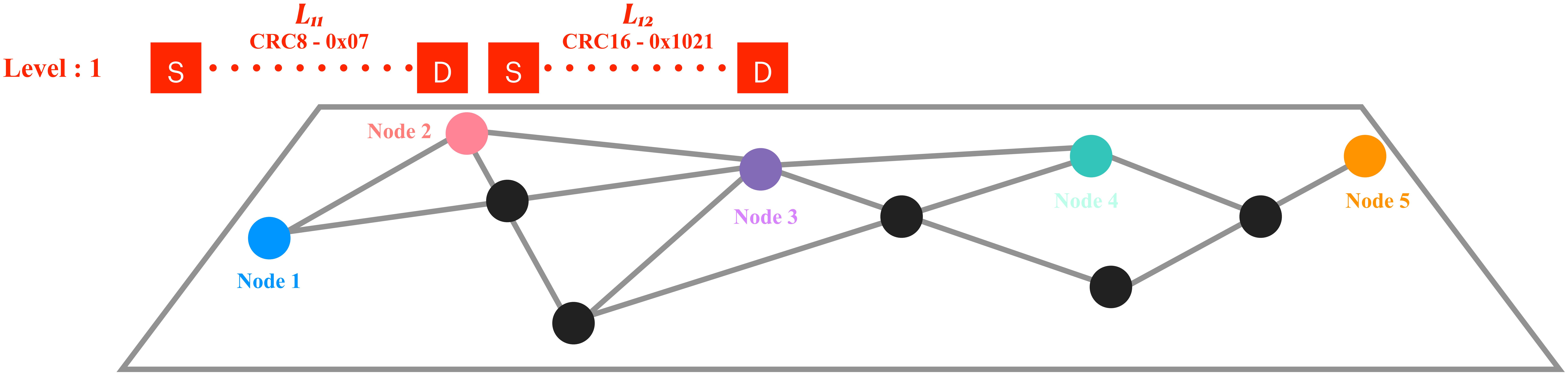
MLED Layers and Virtual Links

Source, Destination and Relay Processes



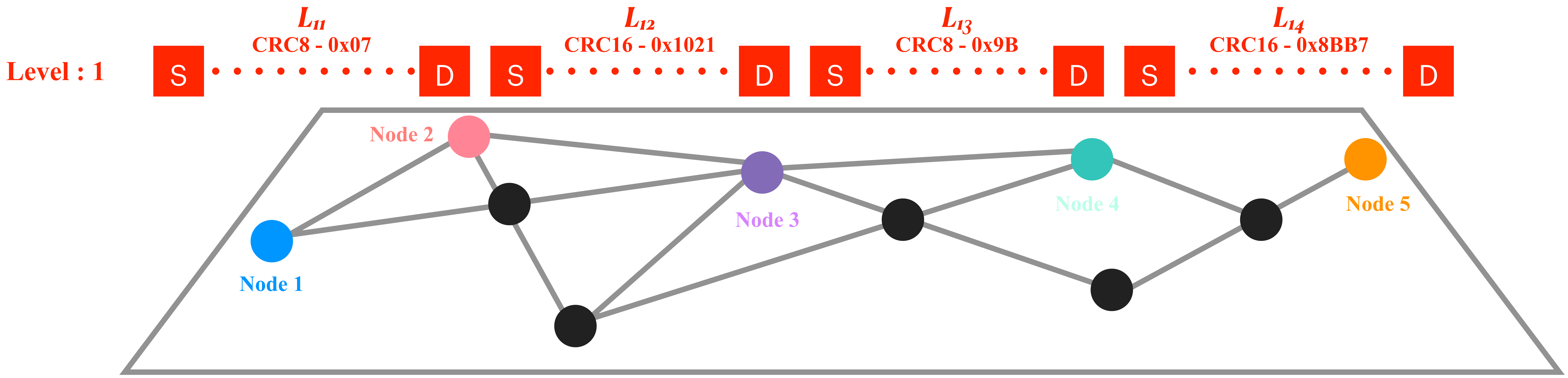
MLED Layers and Virtual Links

Source, Destination and Relay Processes



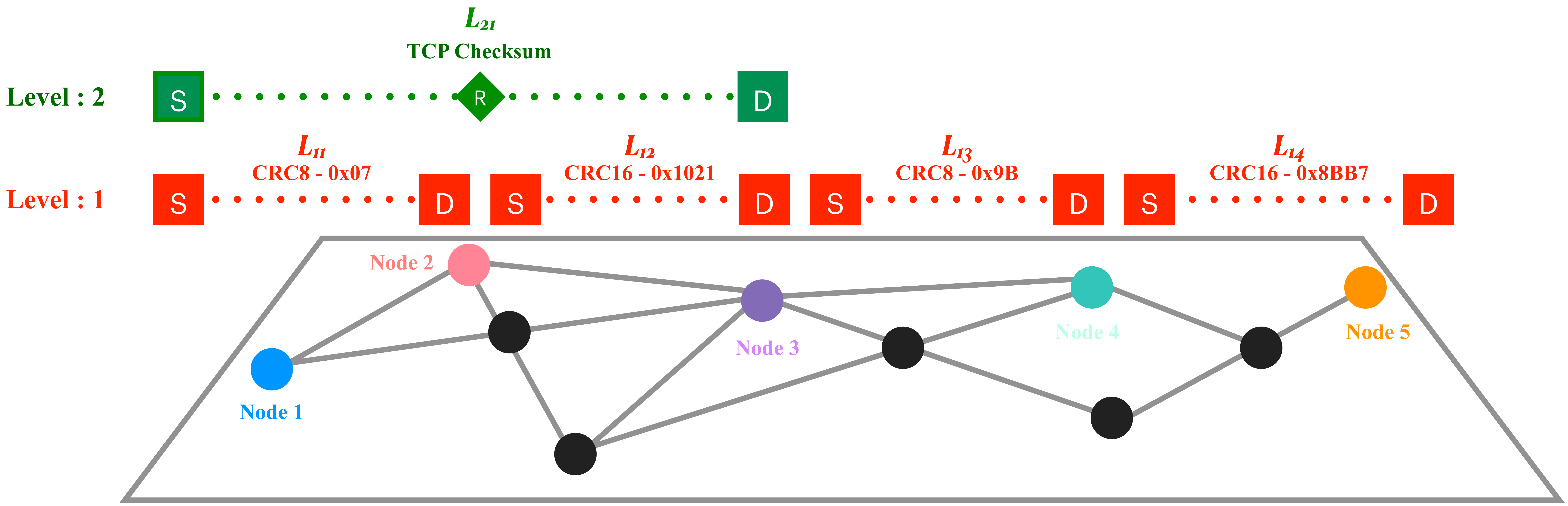
MLED Layers and Virtual Links

Source, Destination and Relay Processes



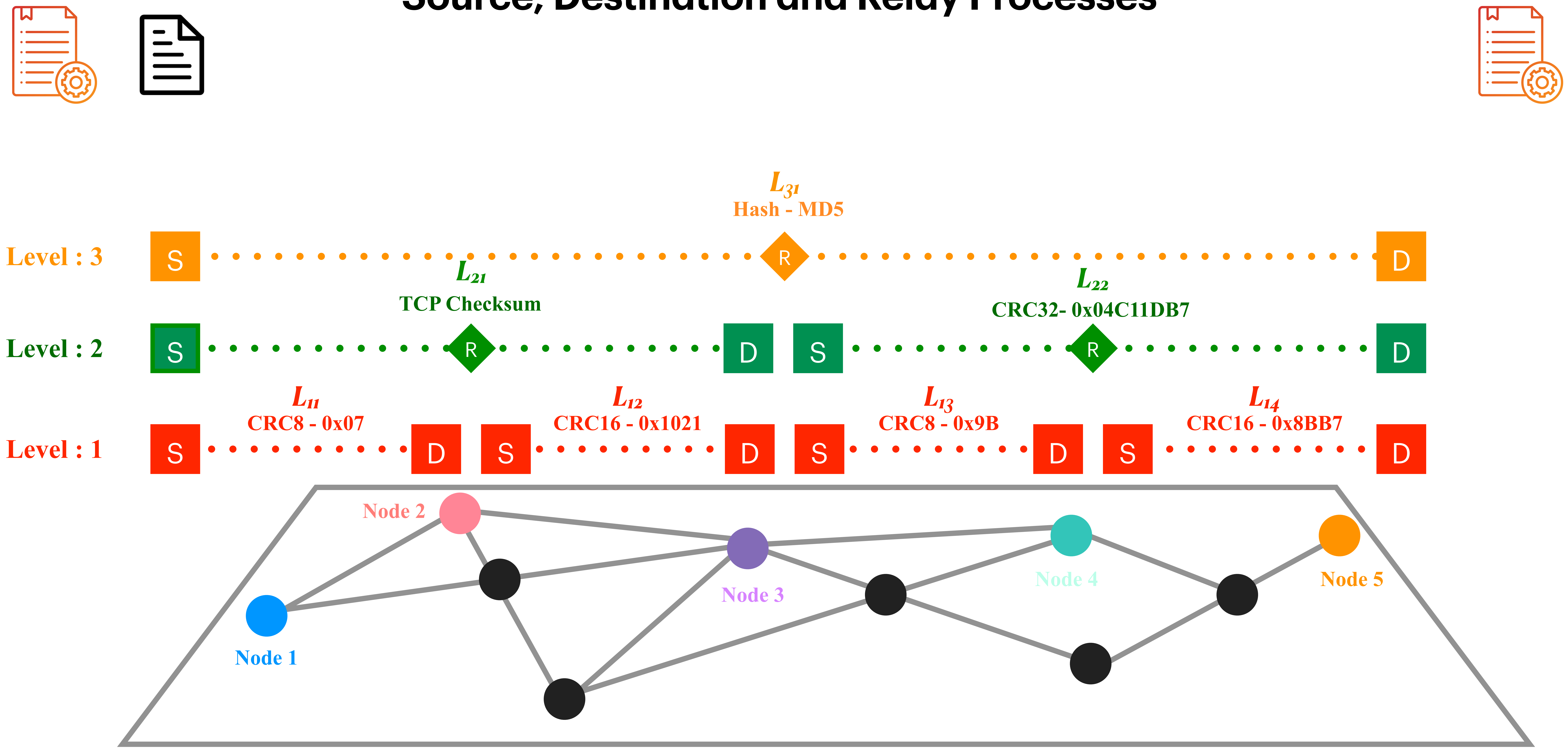
MLED Layers and Virtual Links

Source, Destination and Relay Processes



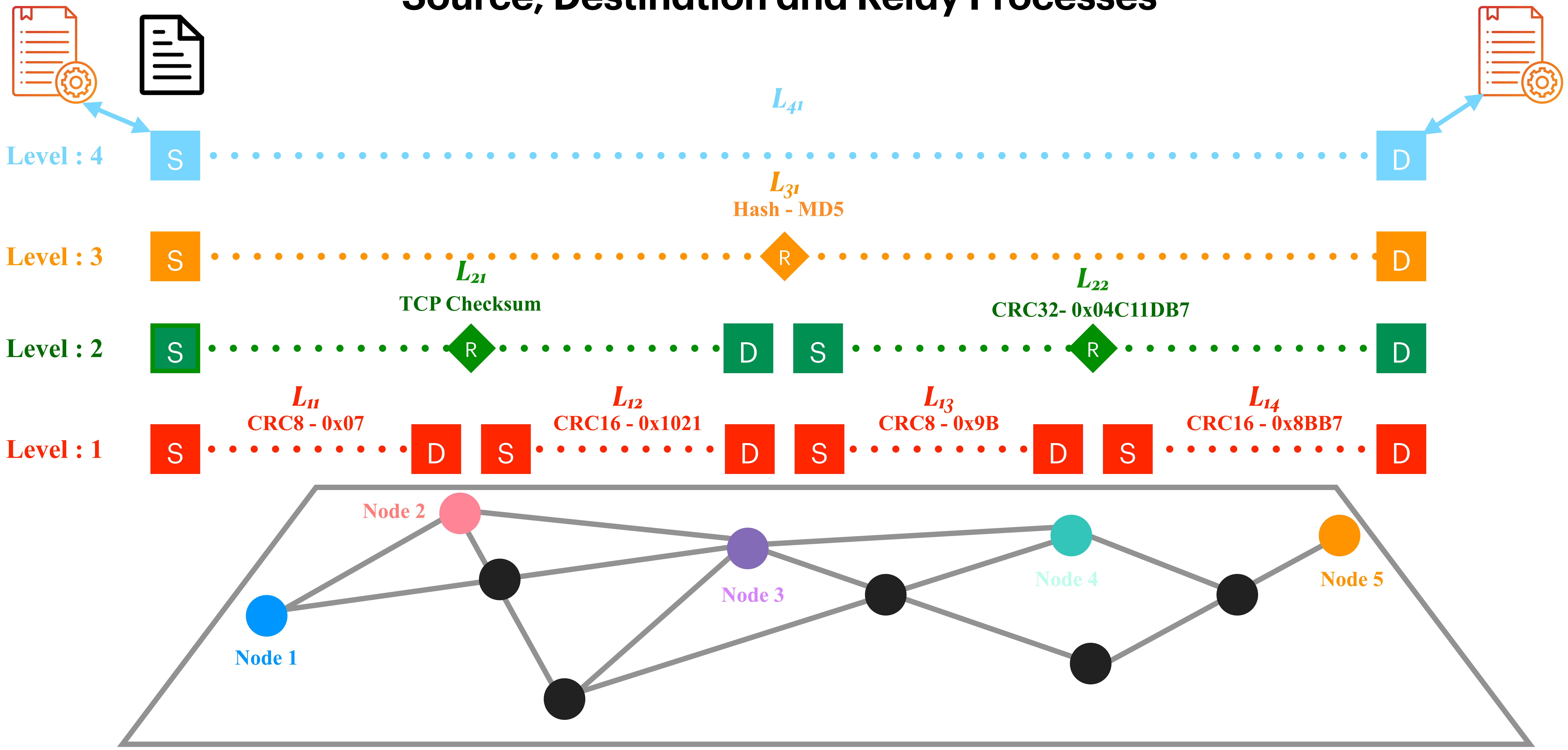
MLED Layers and Virtual Links

Source, Destination and Relay Processes

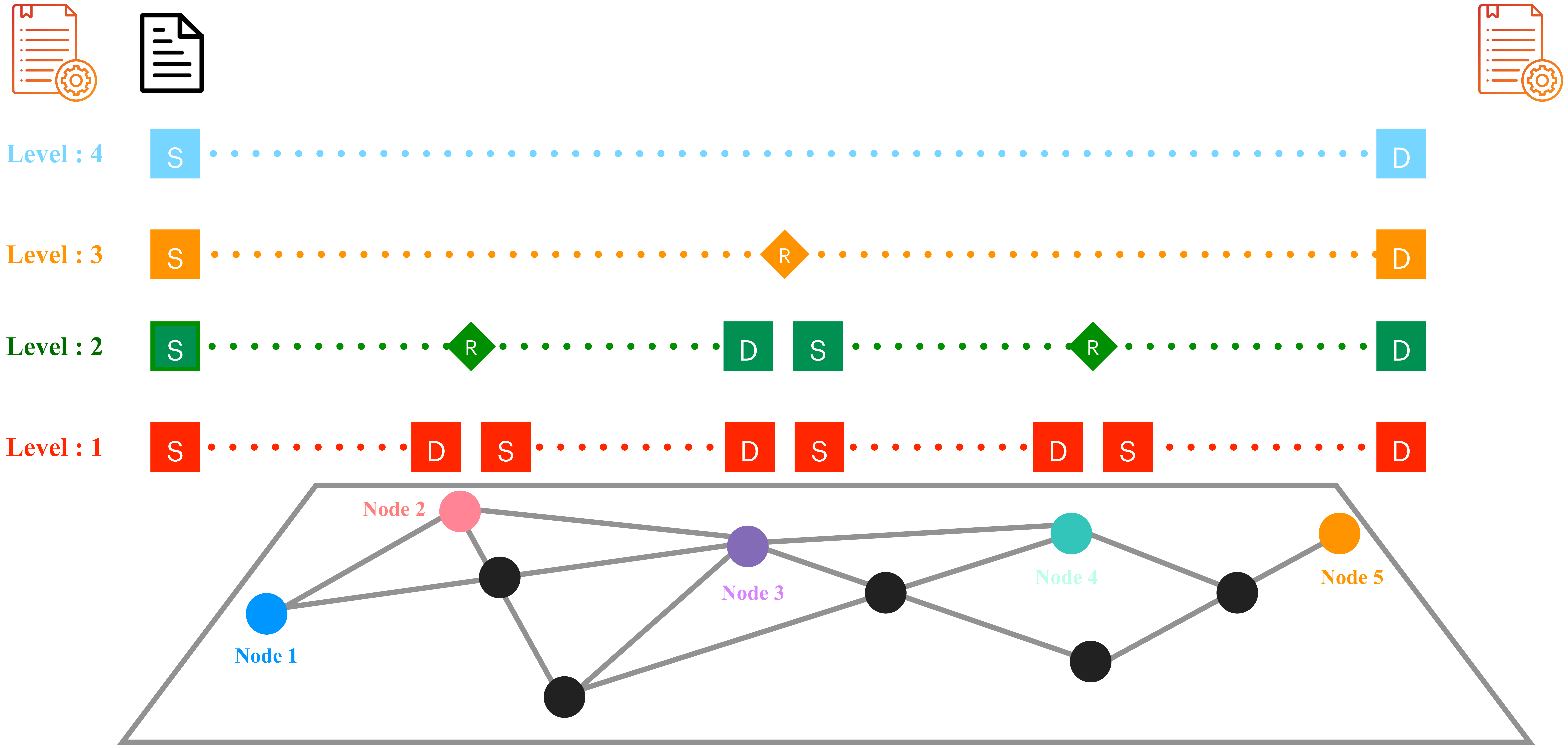


MLED Layers and Virtual Links

Source, Destination and Relay Processes

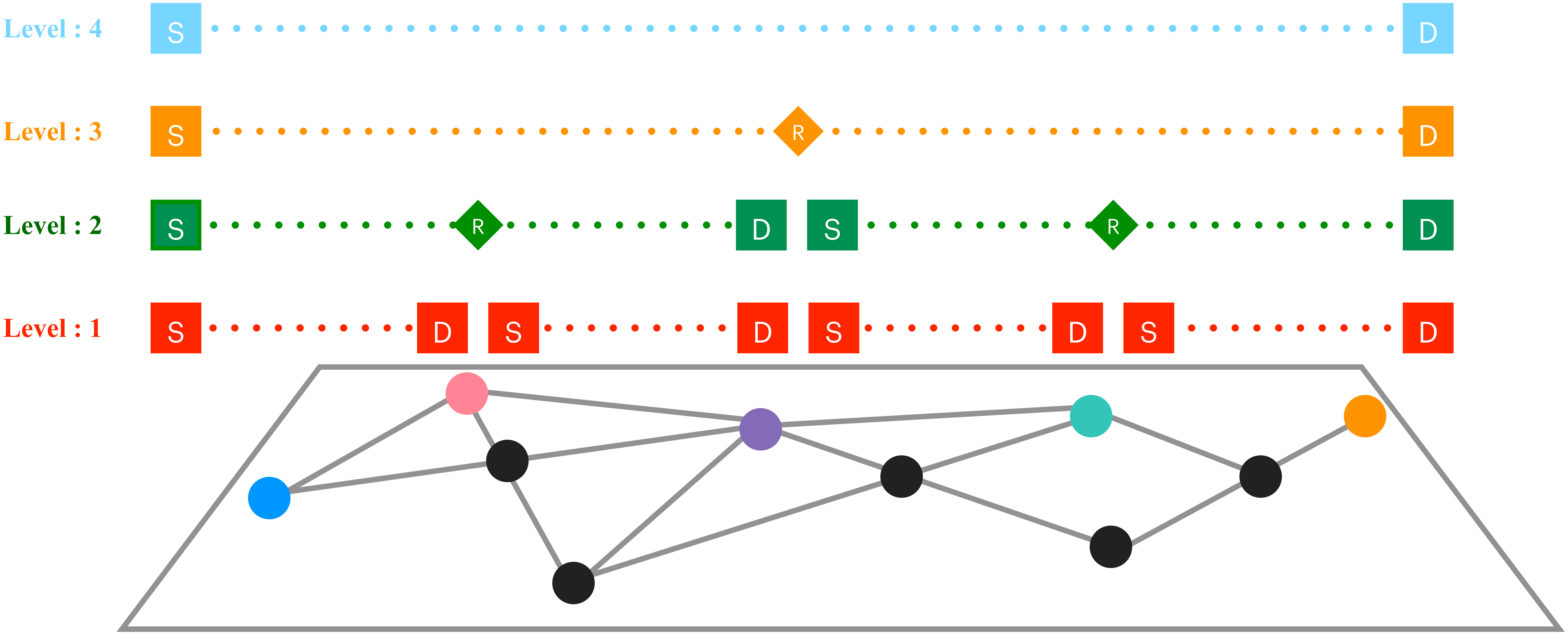


MLED File Transfer



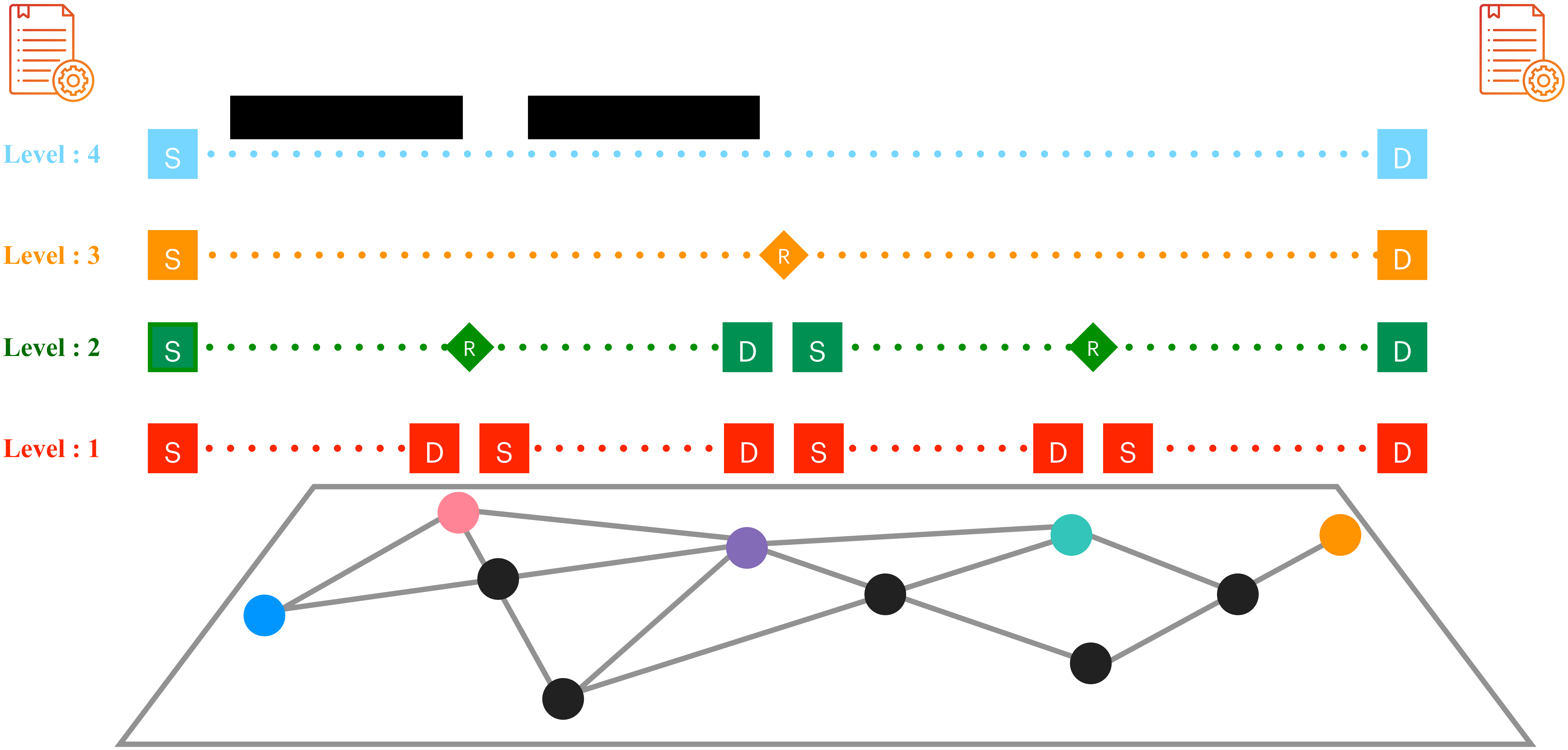
MLED File Transfer

[redacted]

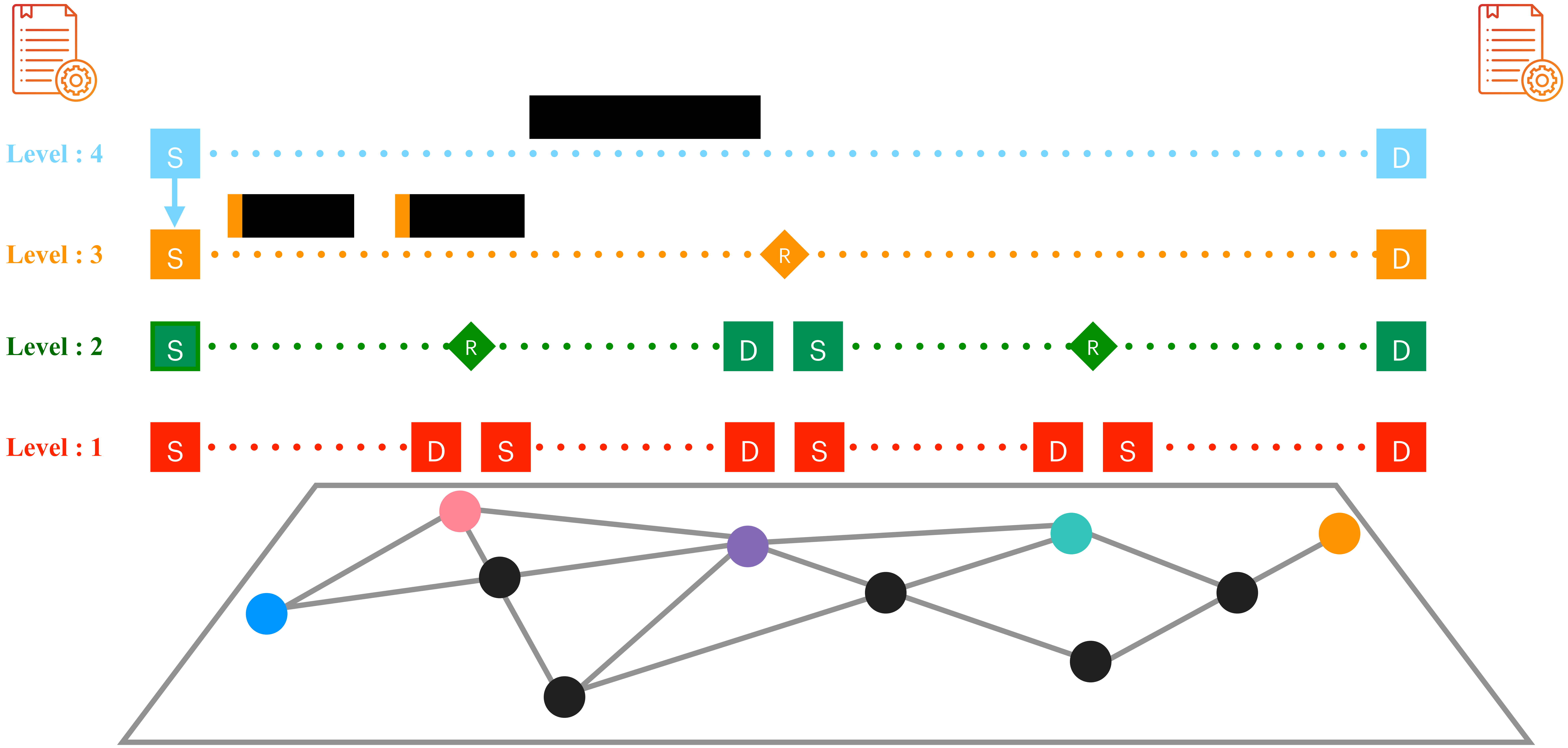


MLED File Transfer

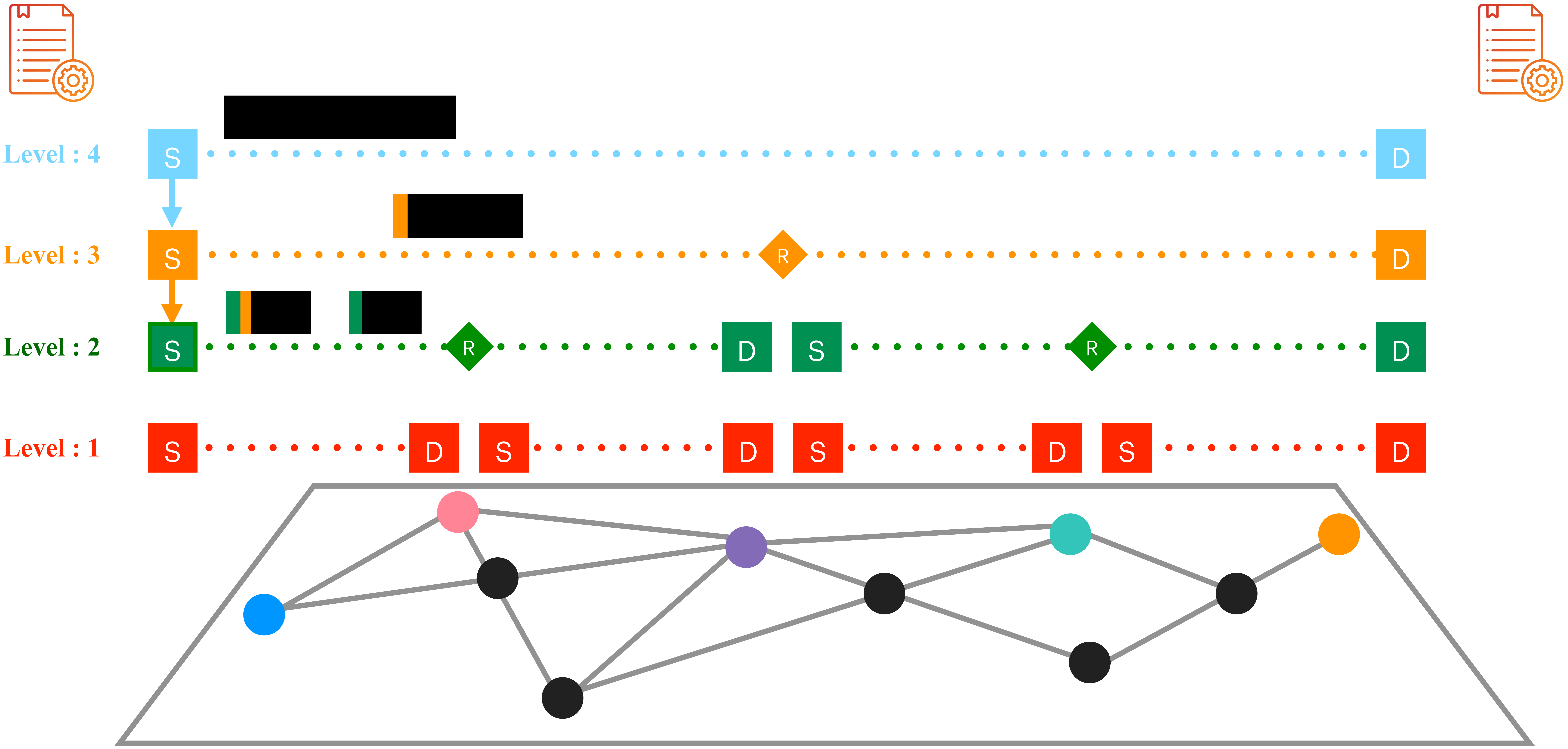
[redacted] [redacted]



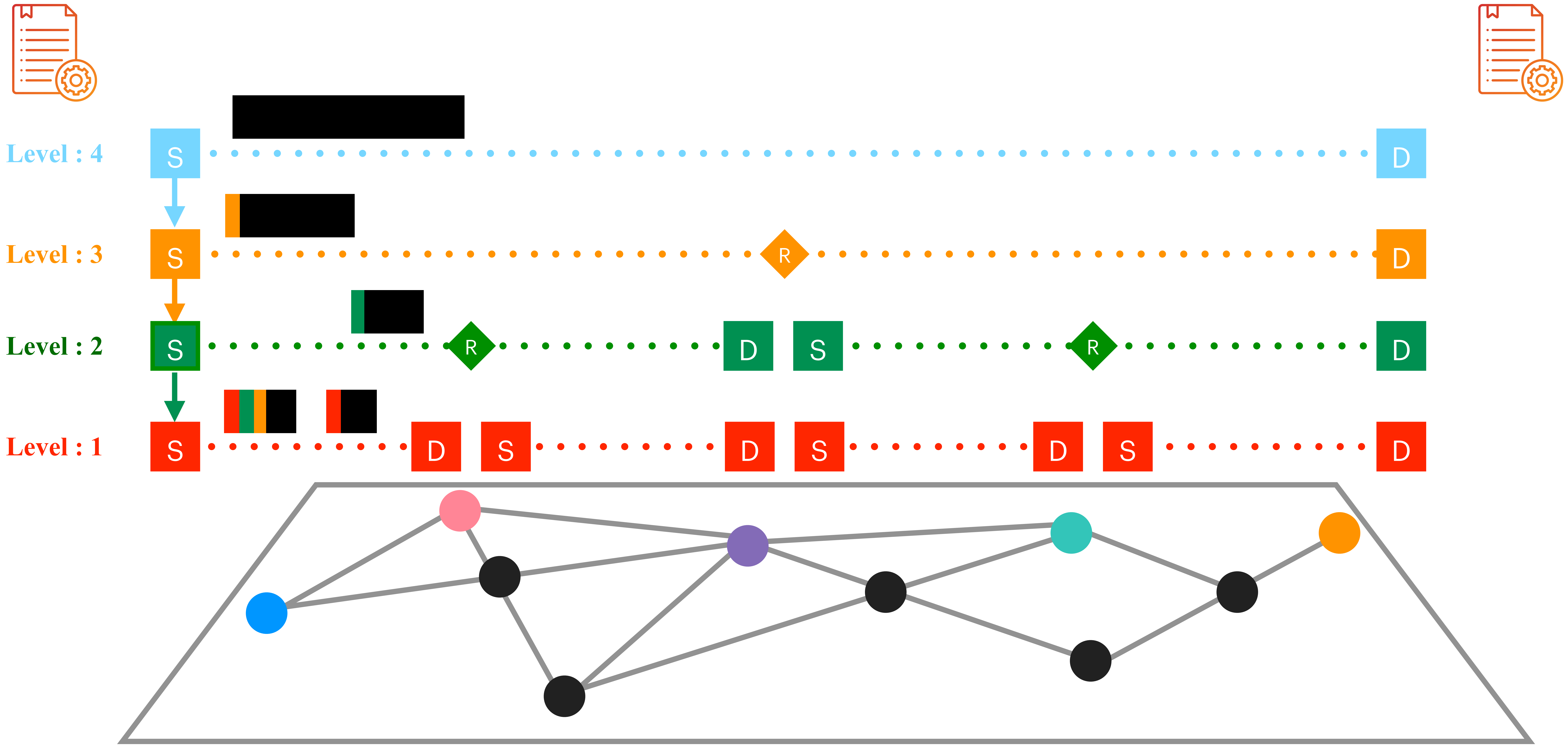
MLED File Transfer



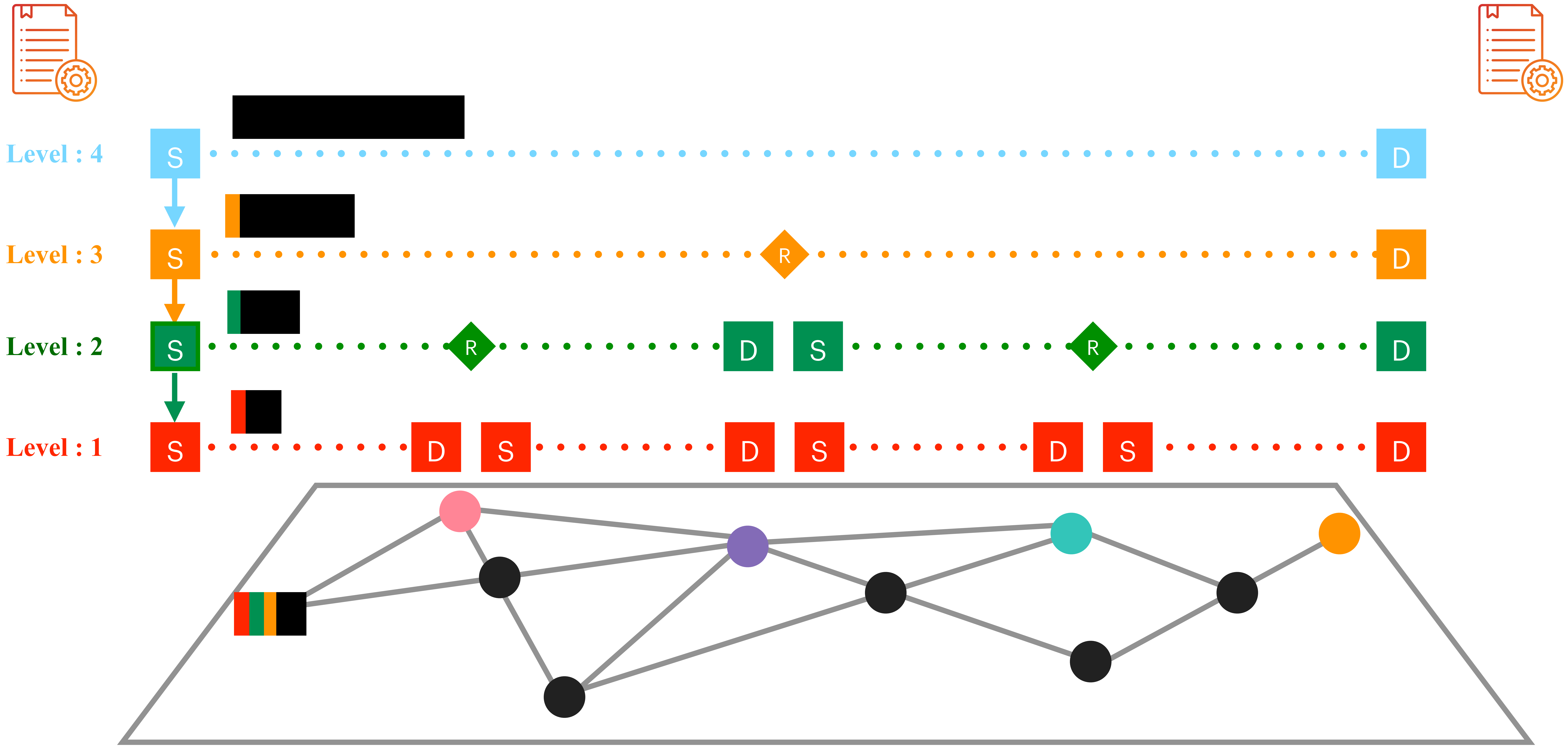
MLED File Transfer



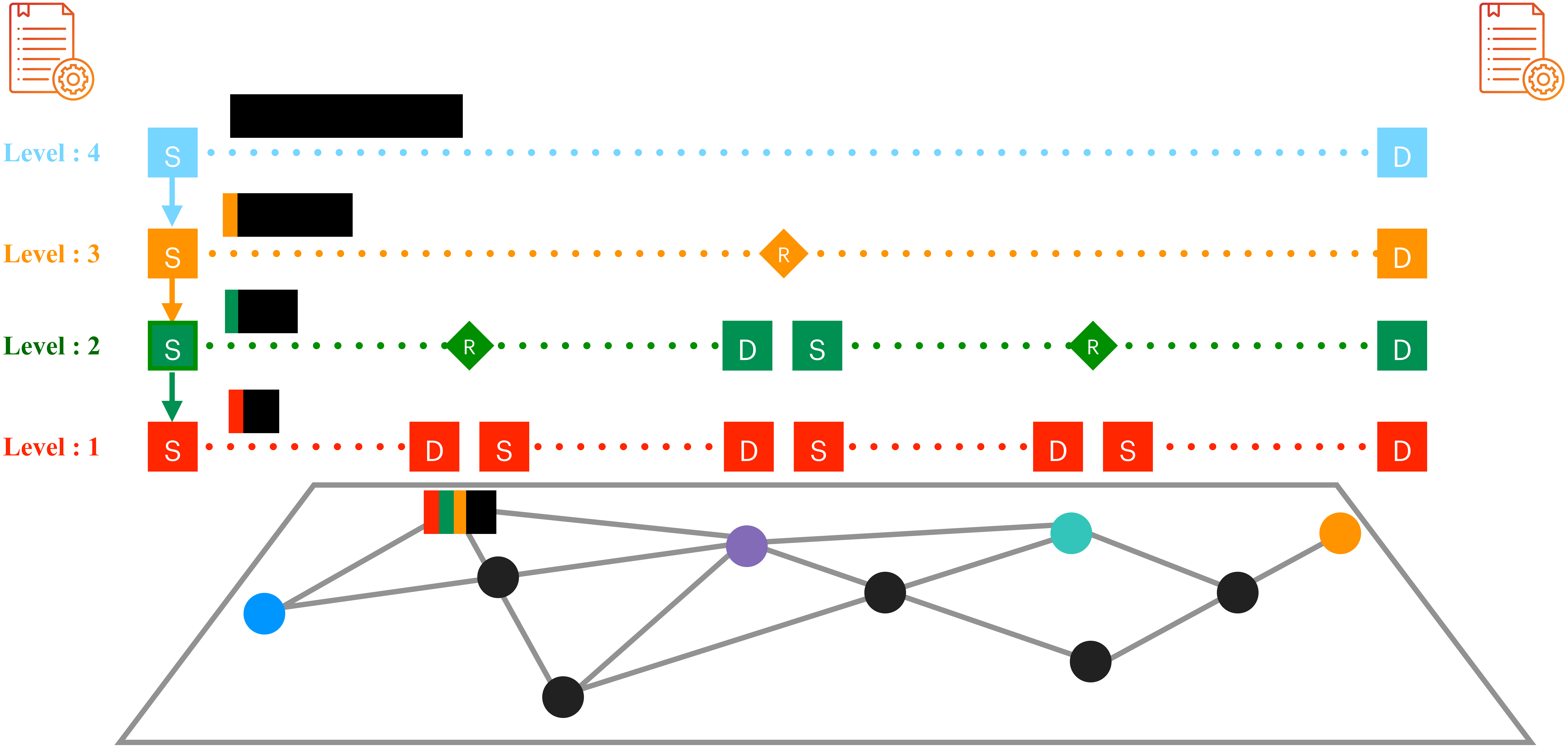
MLED File Transfer



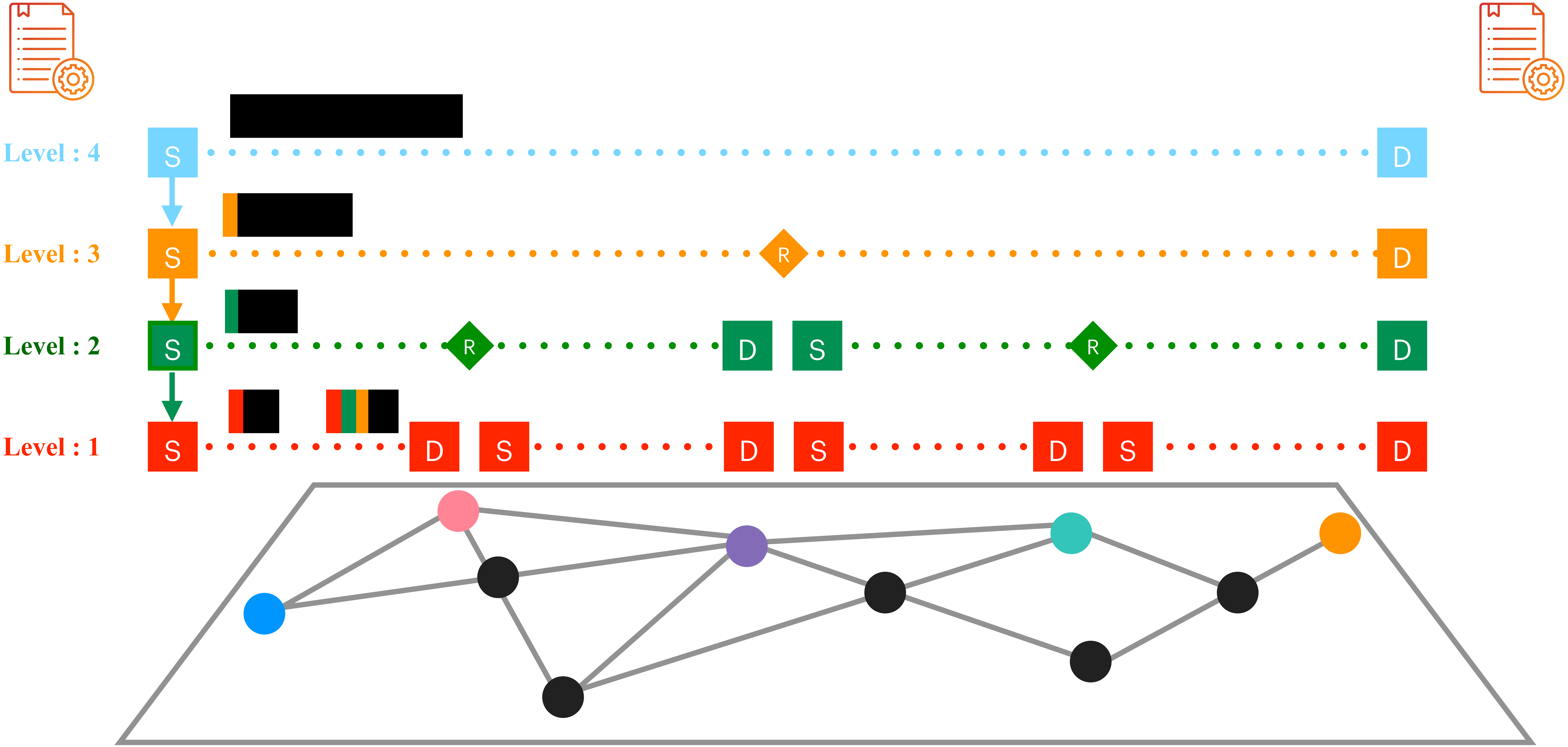
MLED File Transfer



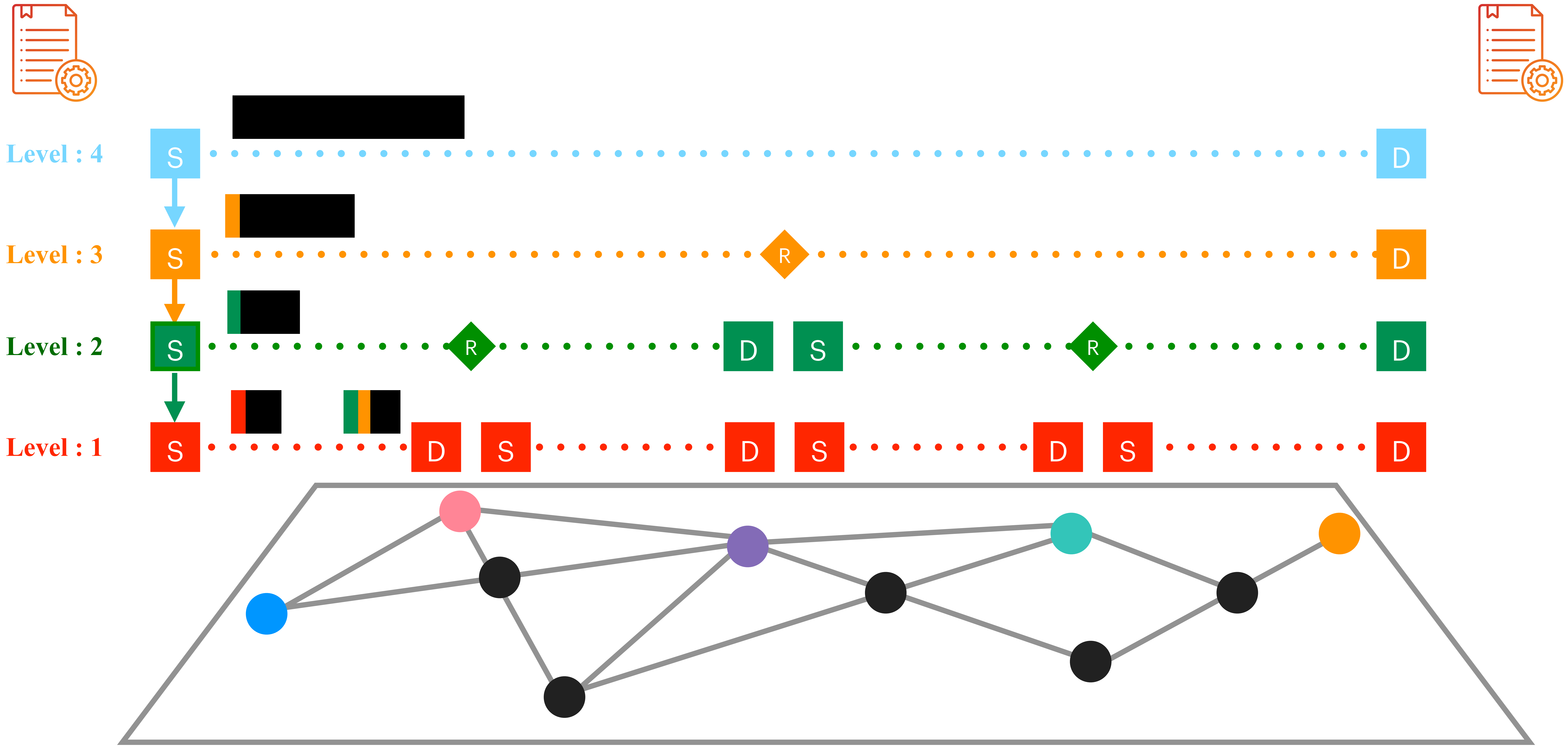
MLED File Transfer



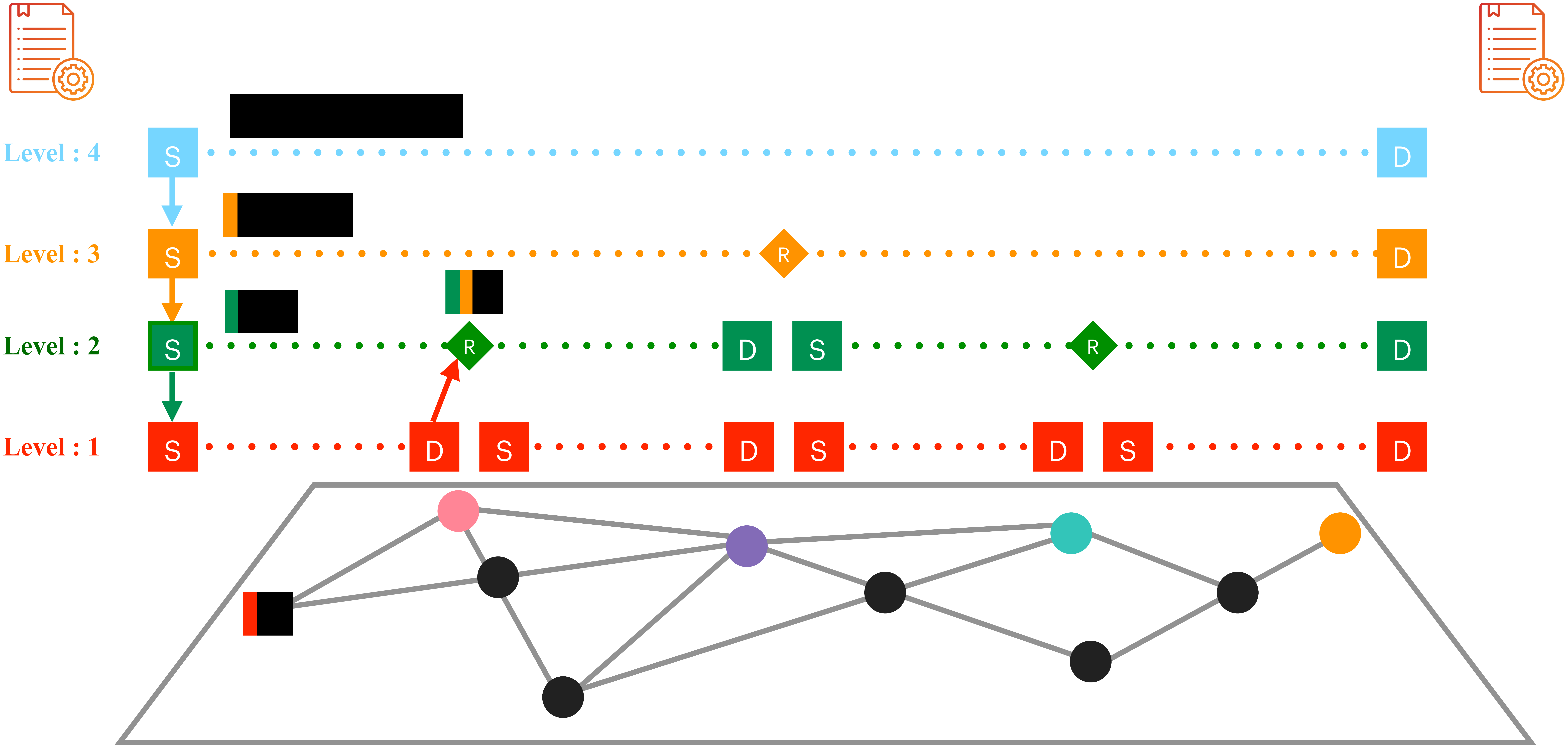
MLED File Transfer



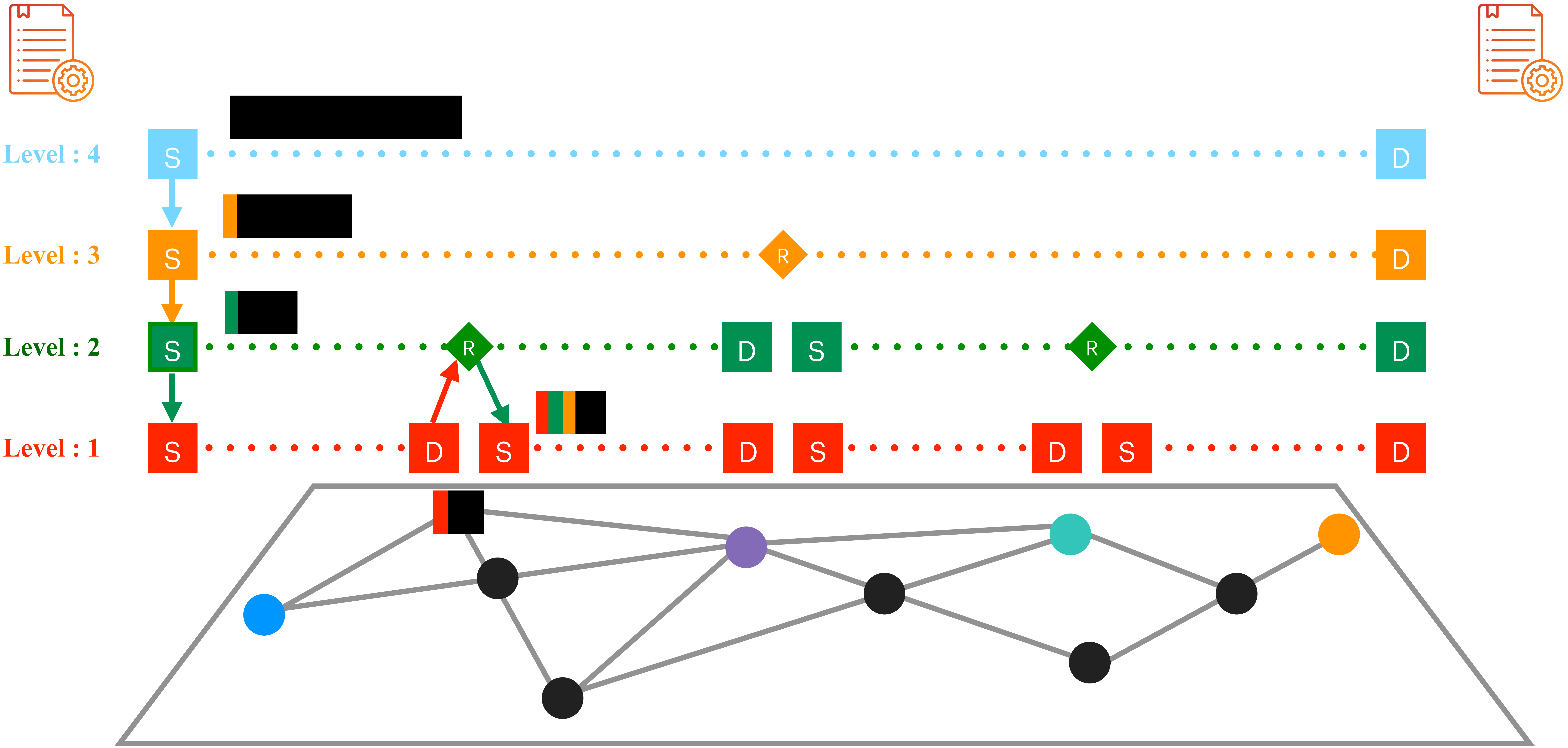
MLED File Transfer



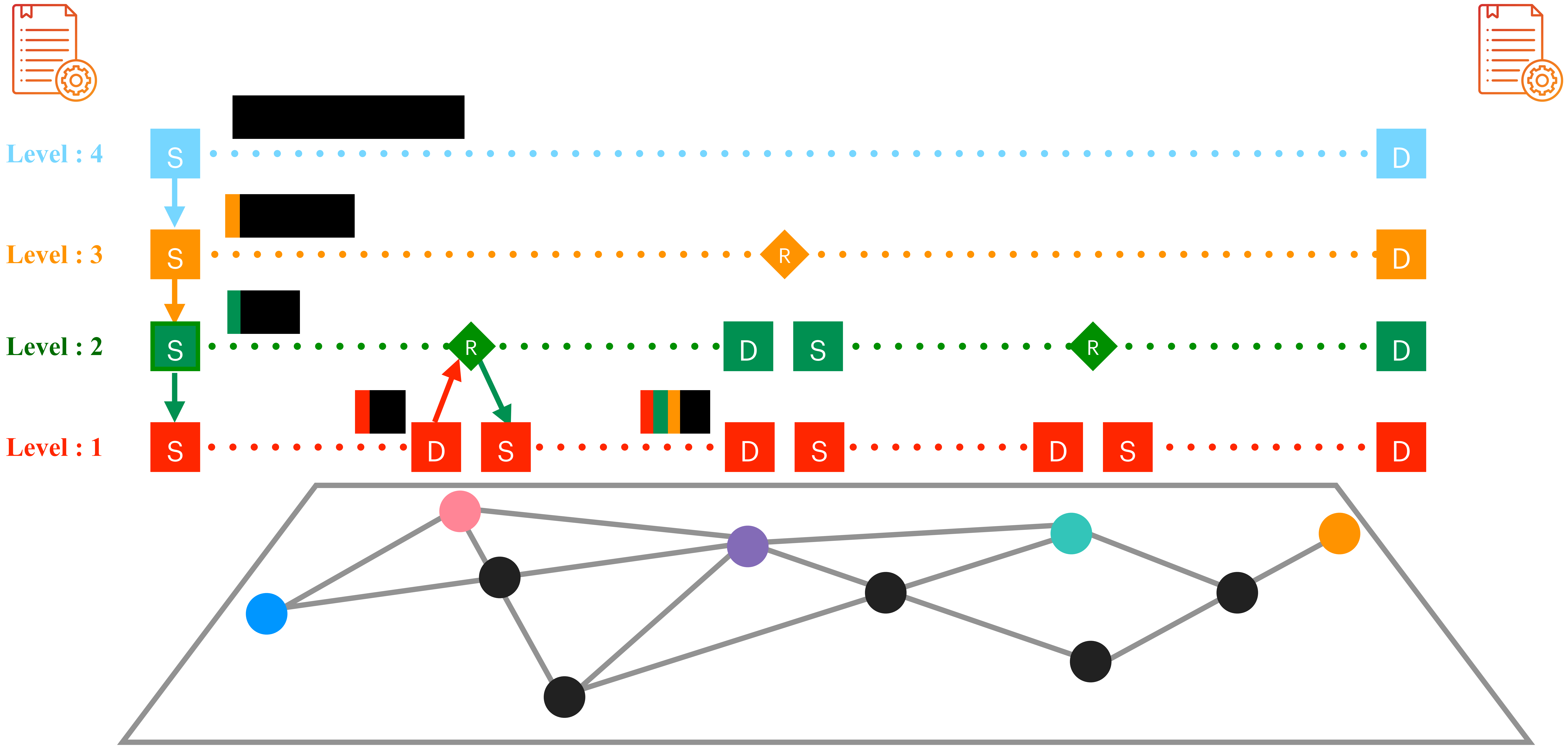
MLED File Transfer



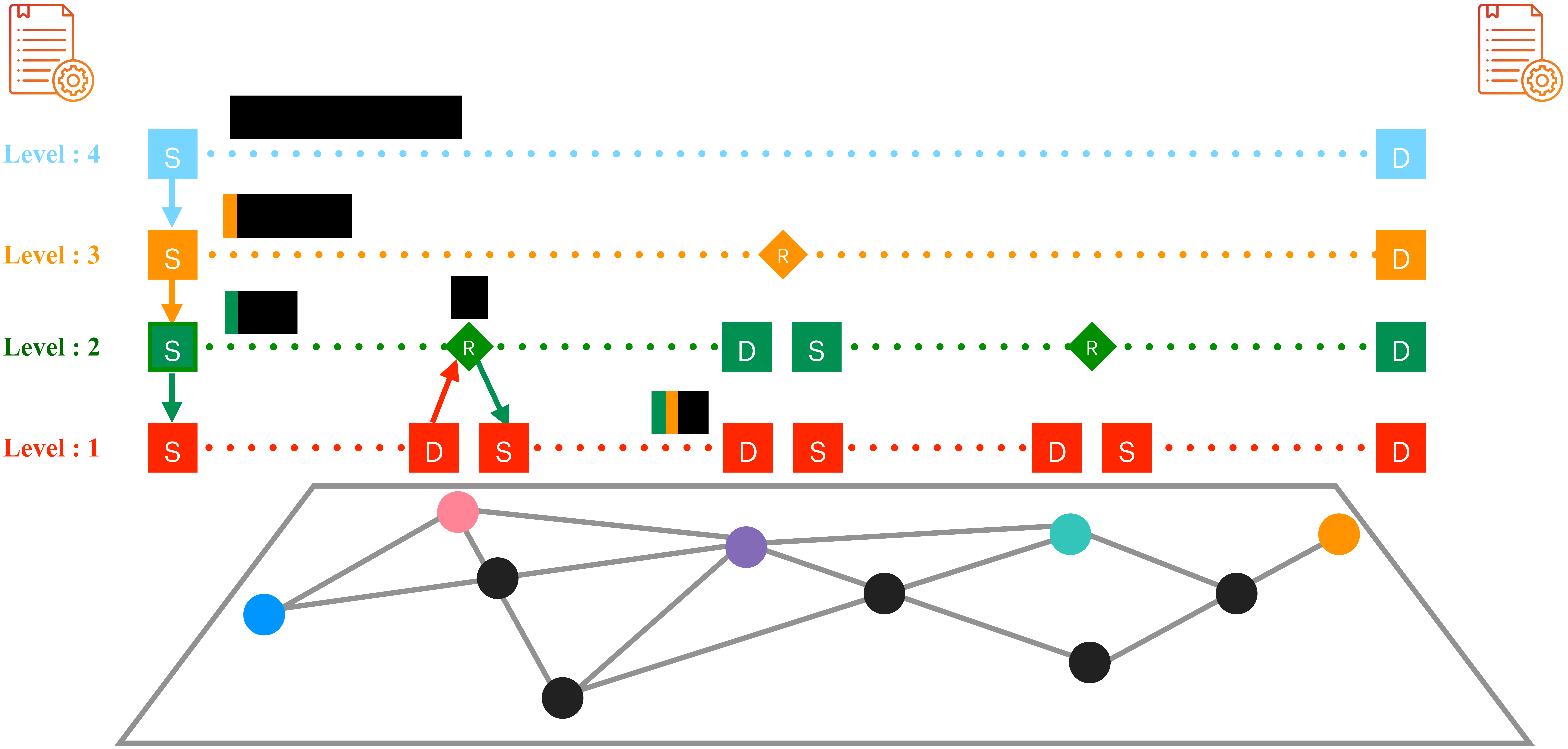
MLED File Transfer



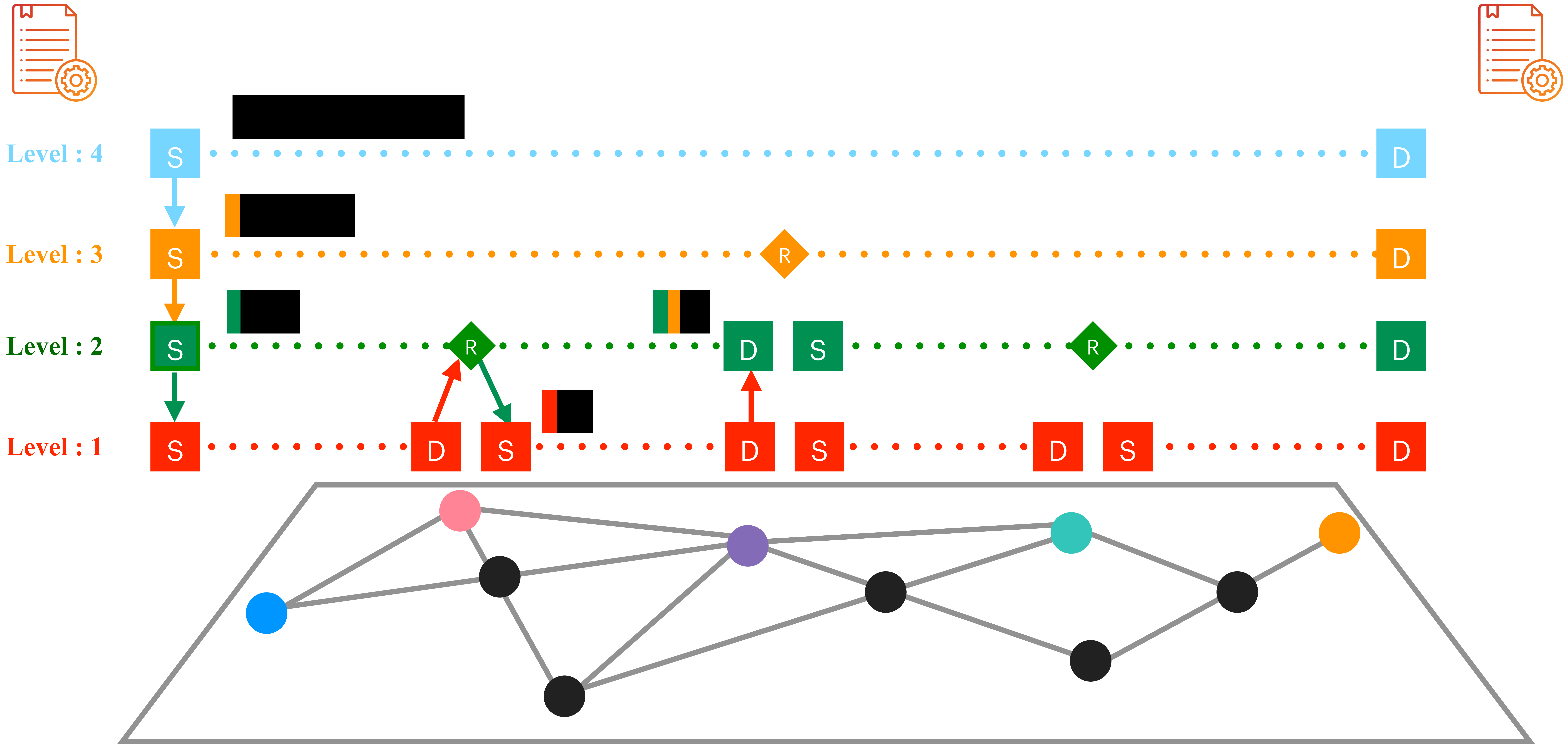
MLED File Transfer



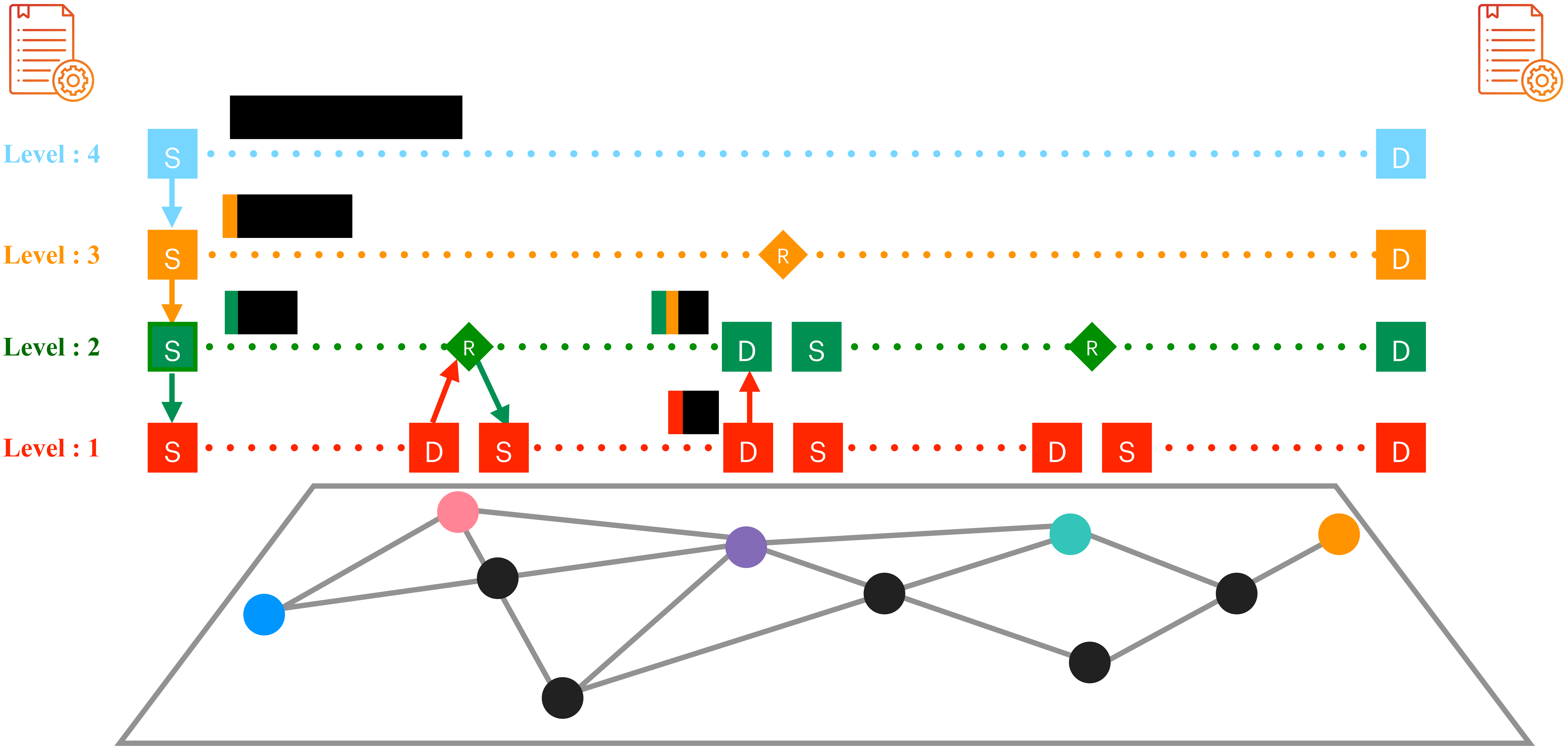
MLED File Transfer



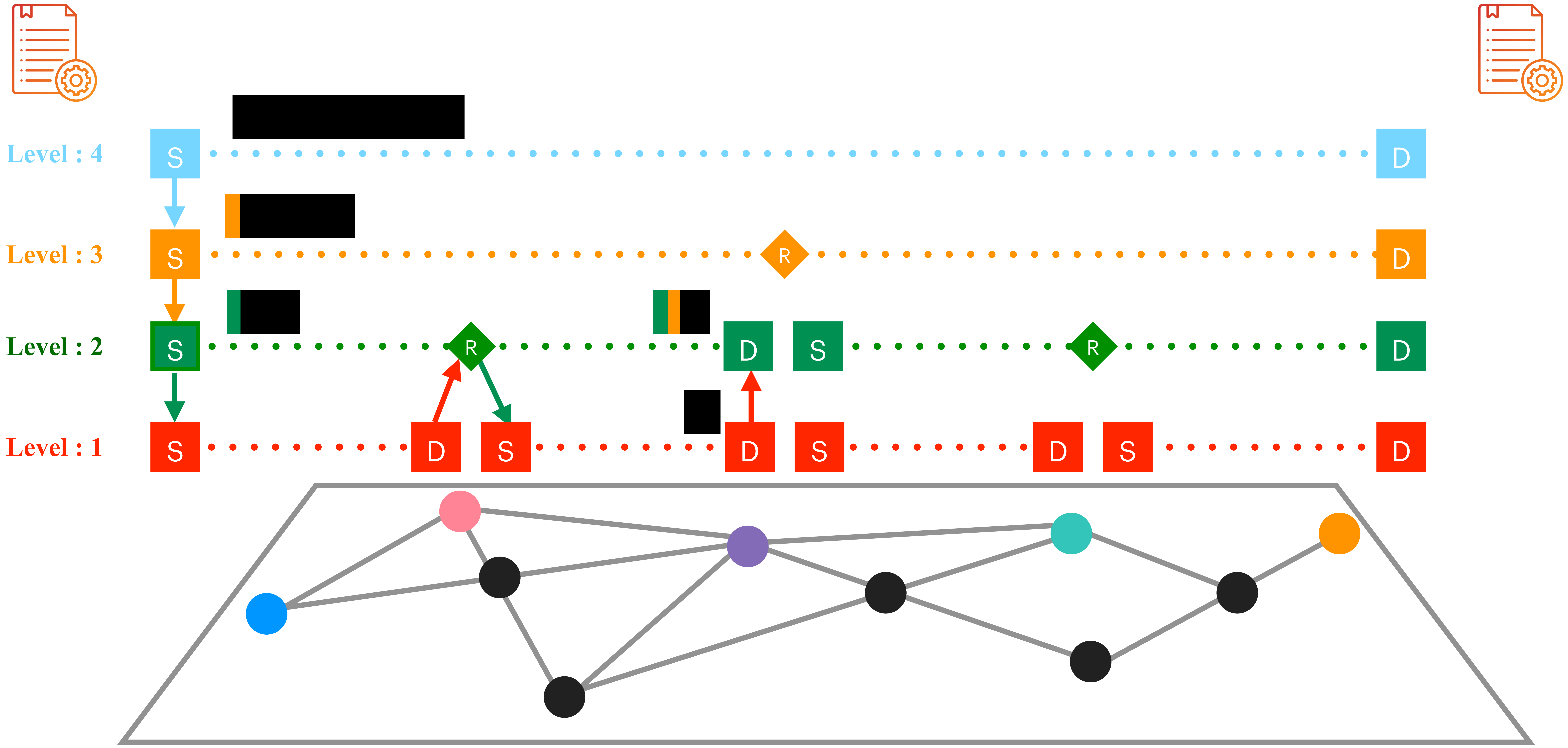
MLED File Transfer



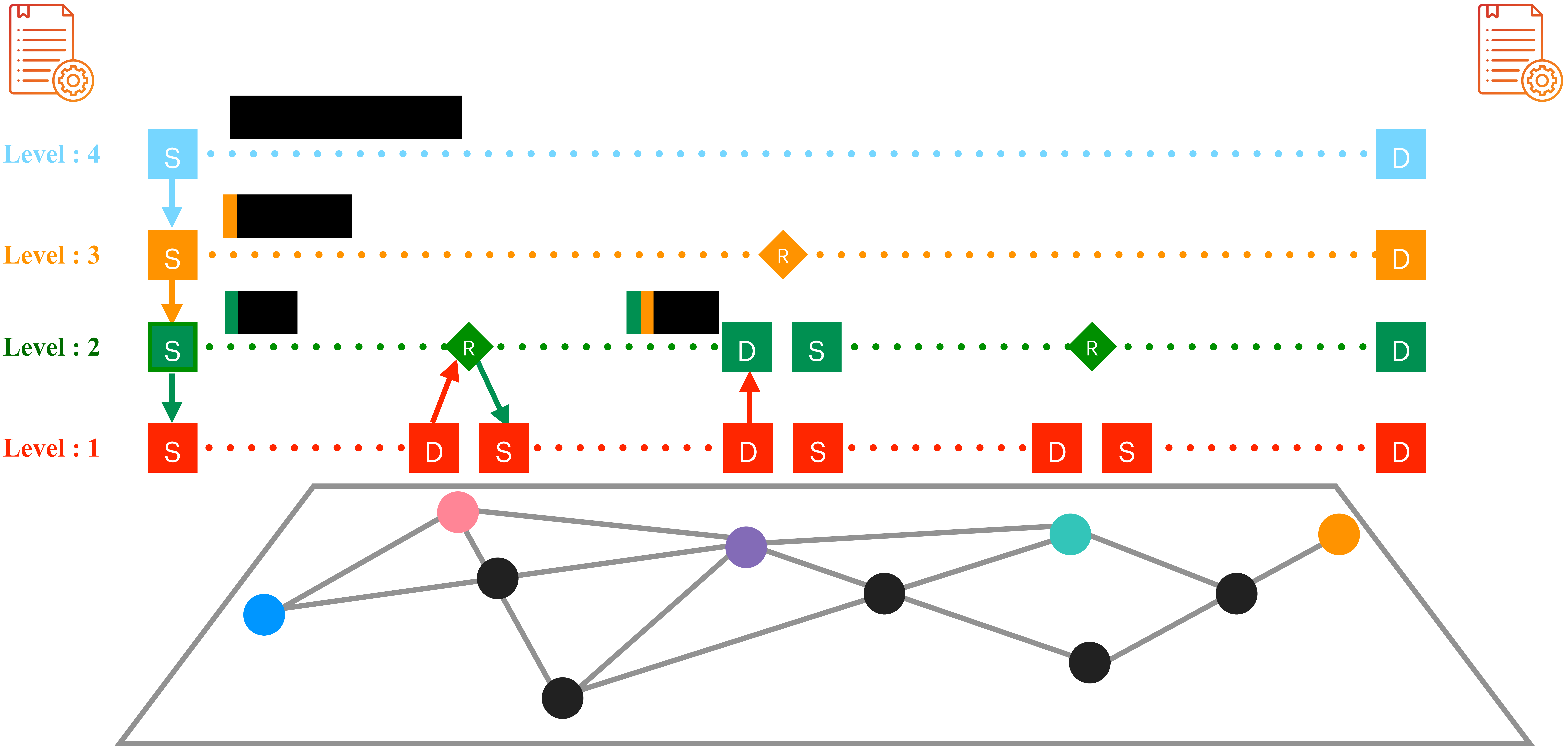
MLED File Transfer



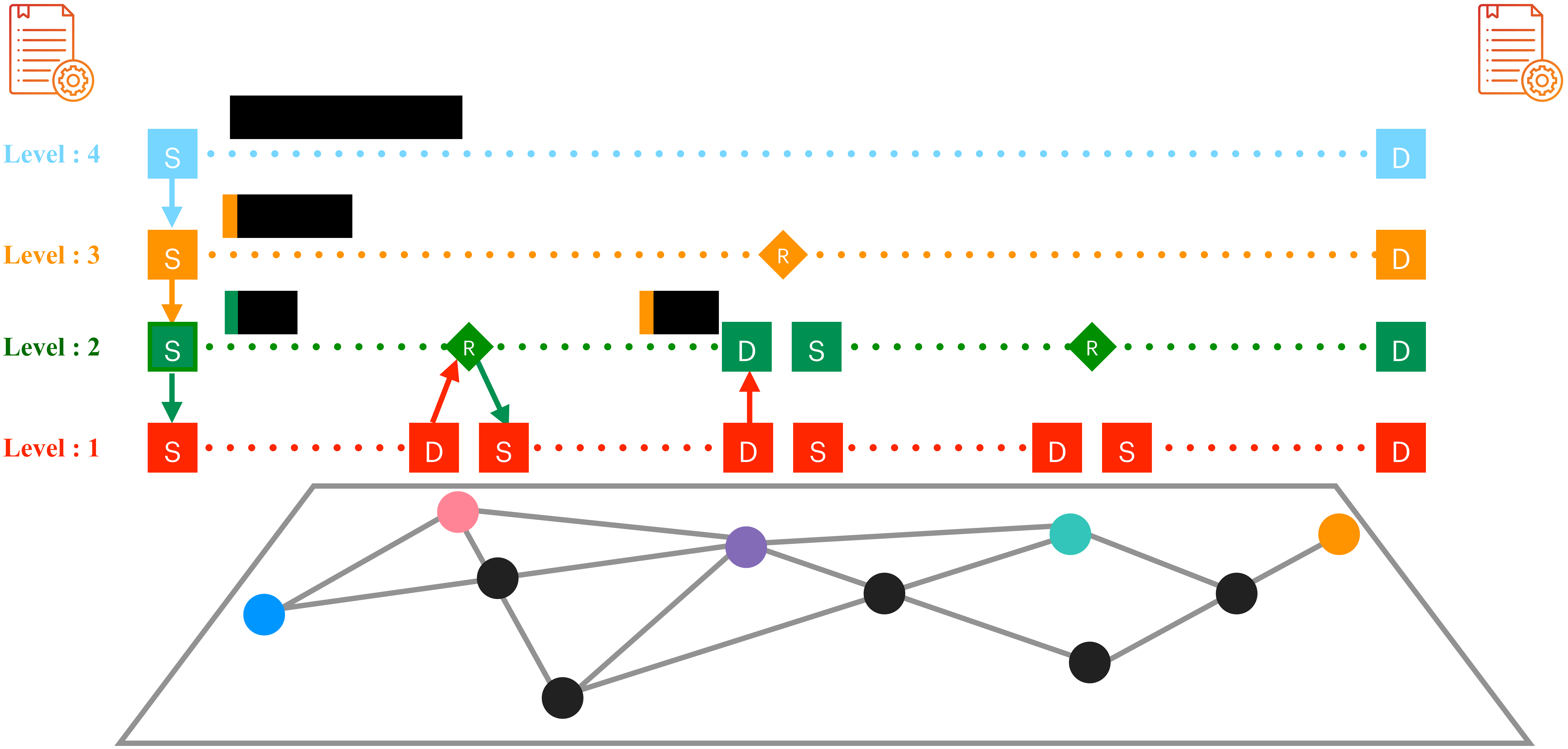
MLED File Transfer



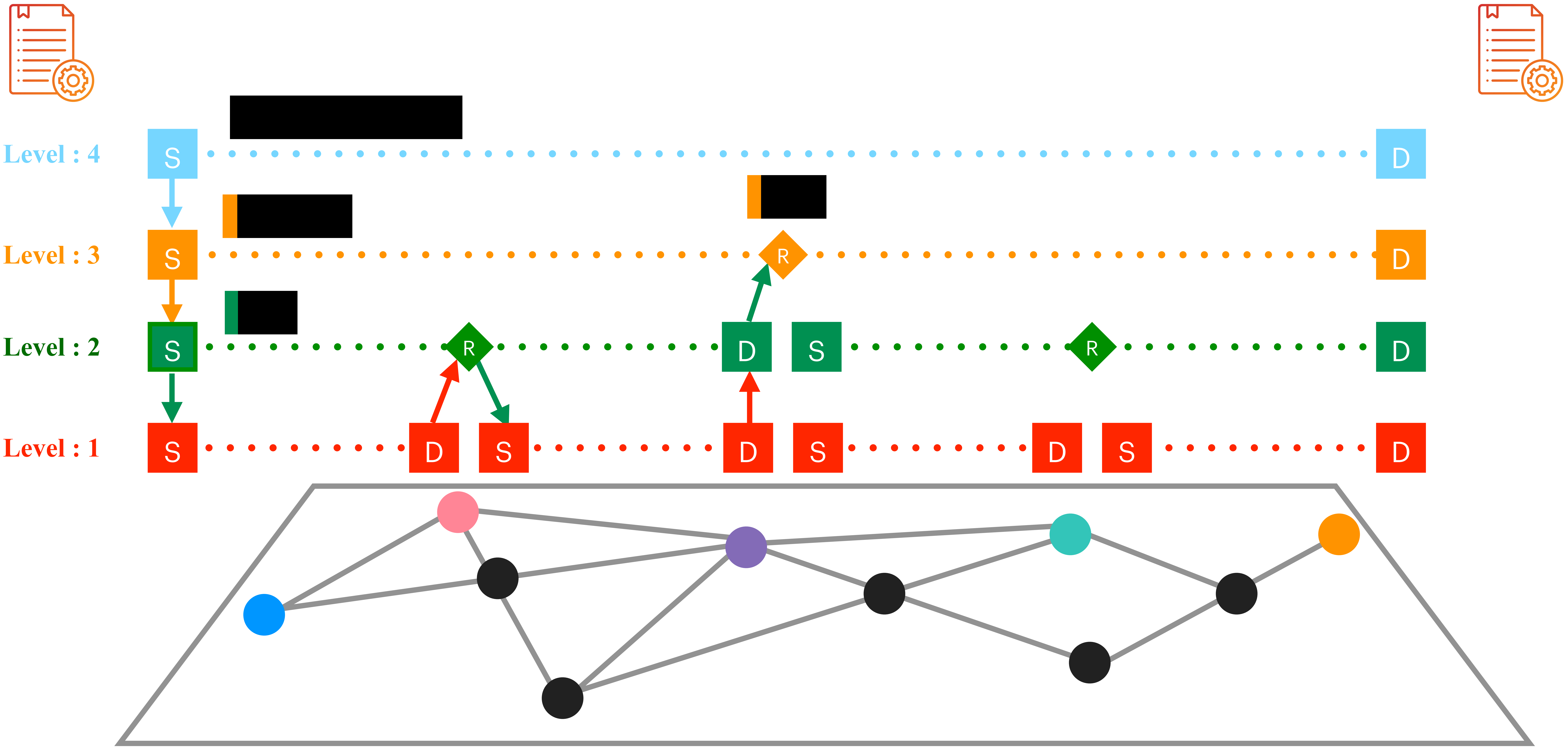
MLED File Transfer



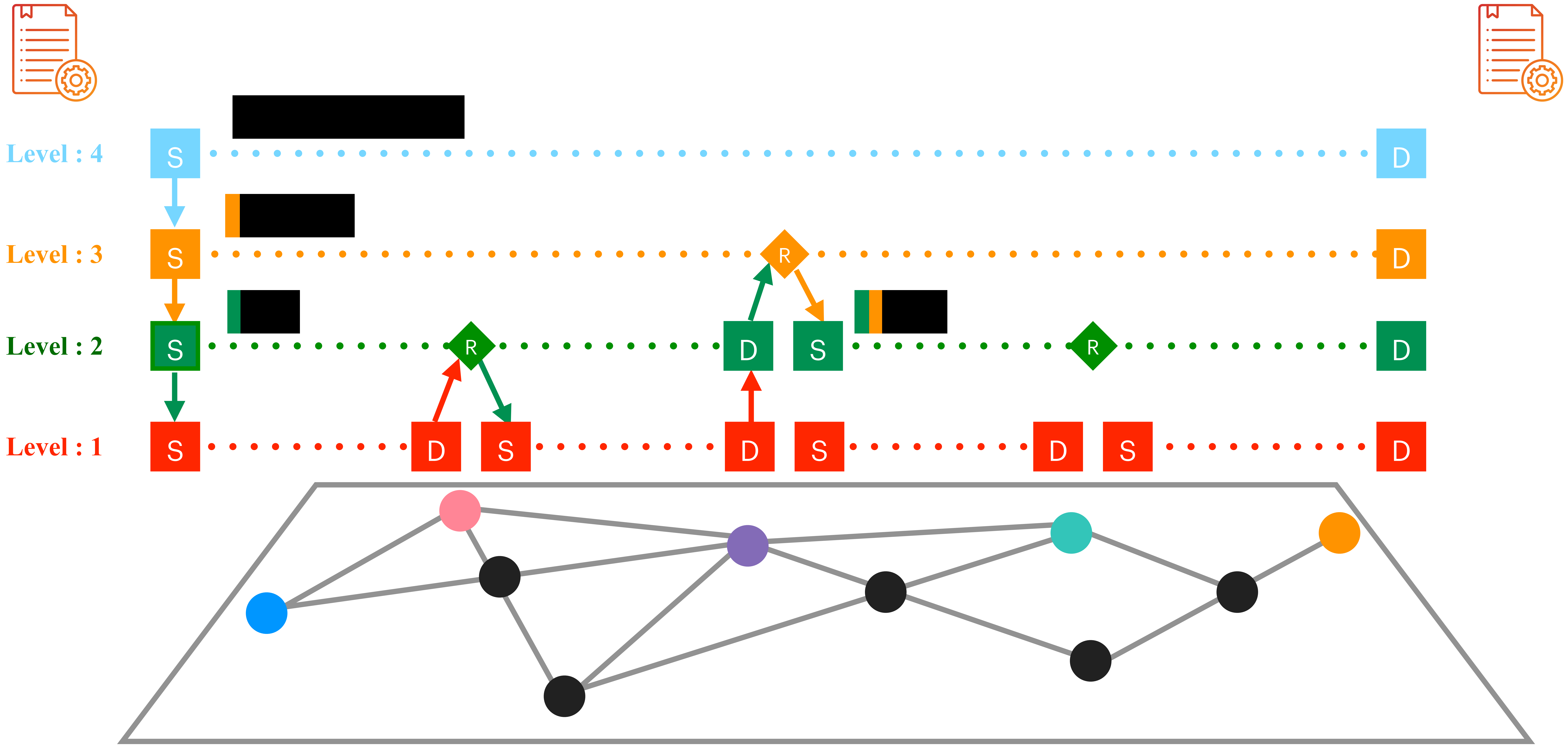
MLED File Transfer



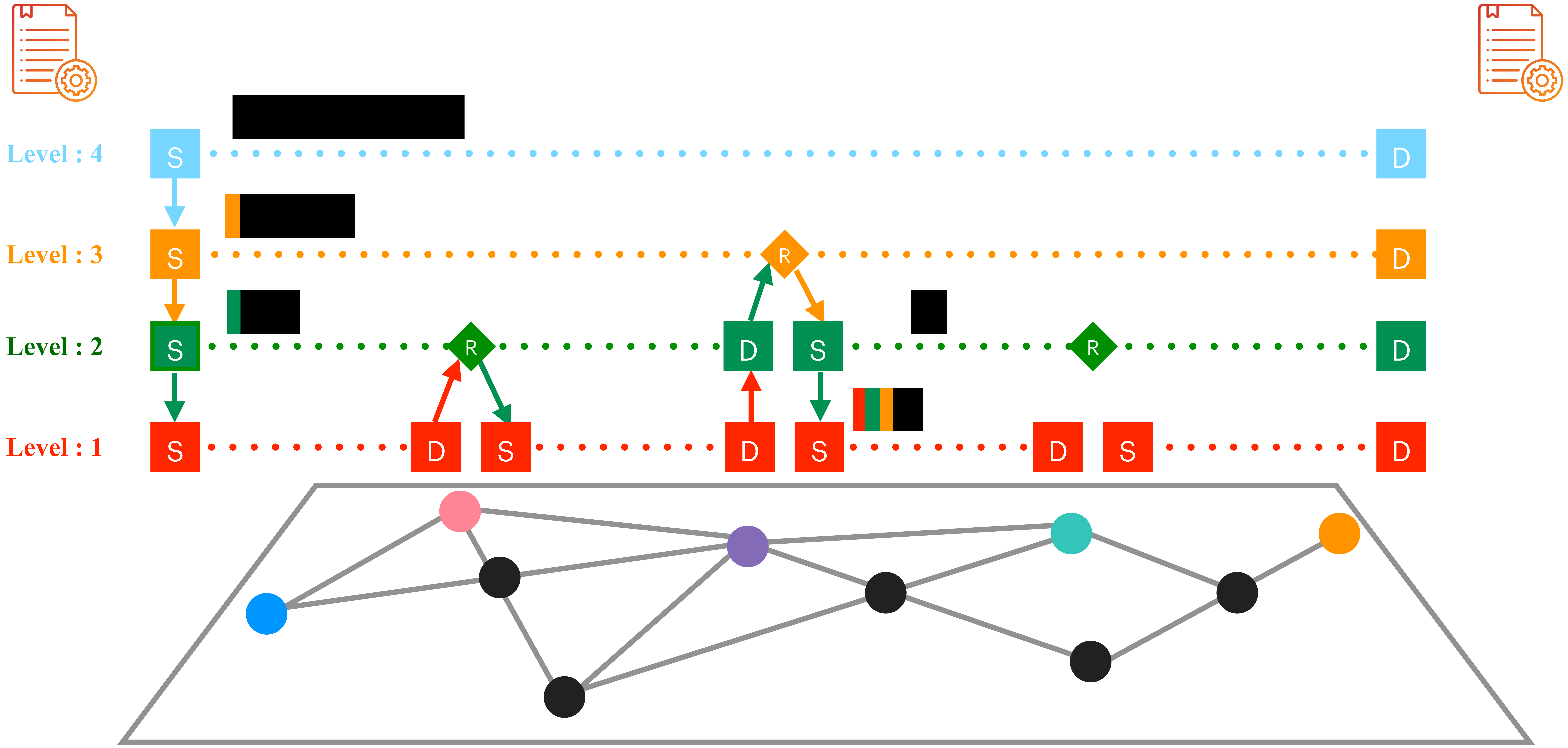
MLED File Transfer



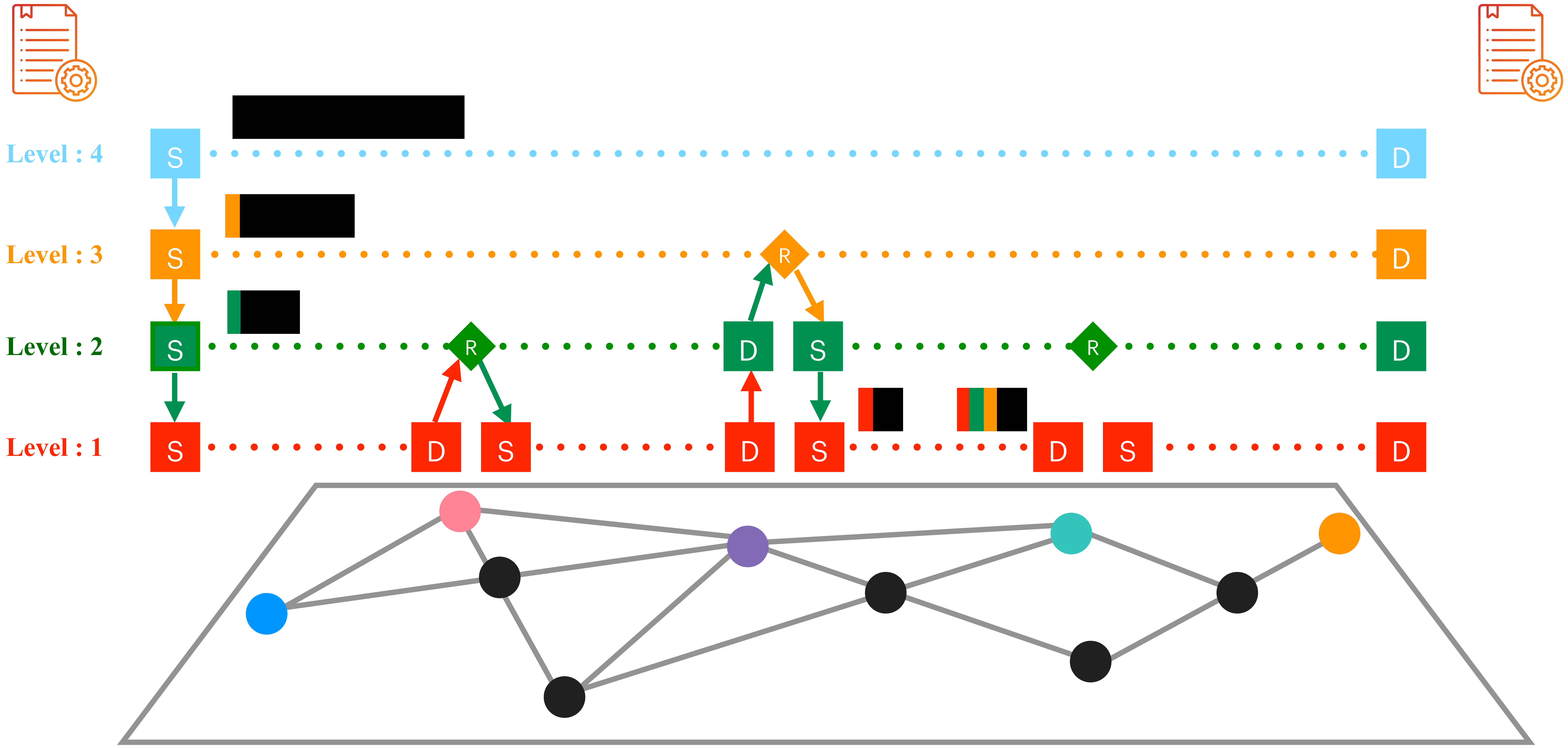
MLED File Transfer



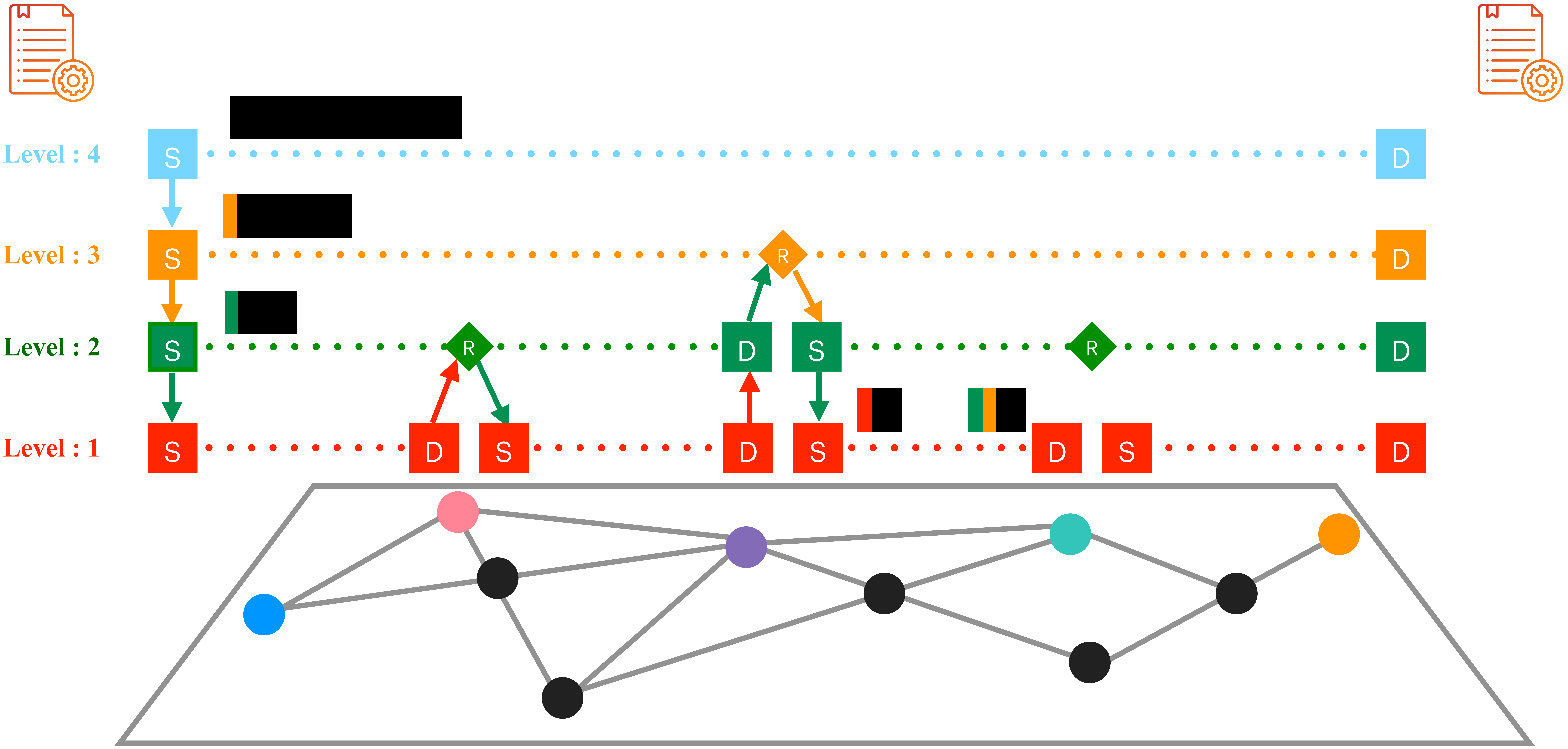
MLED File Transfer



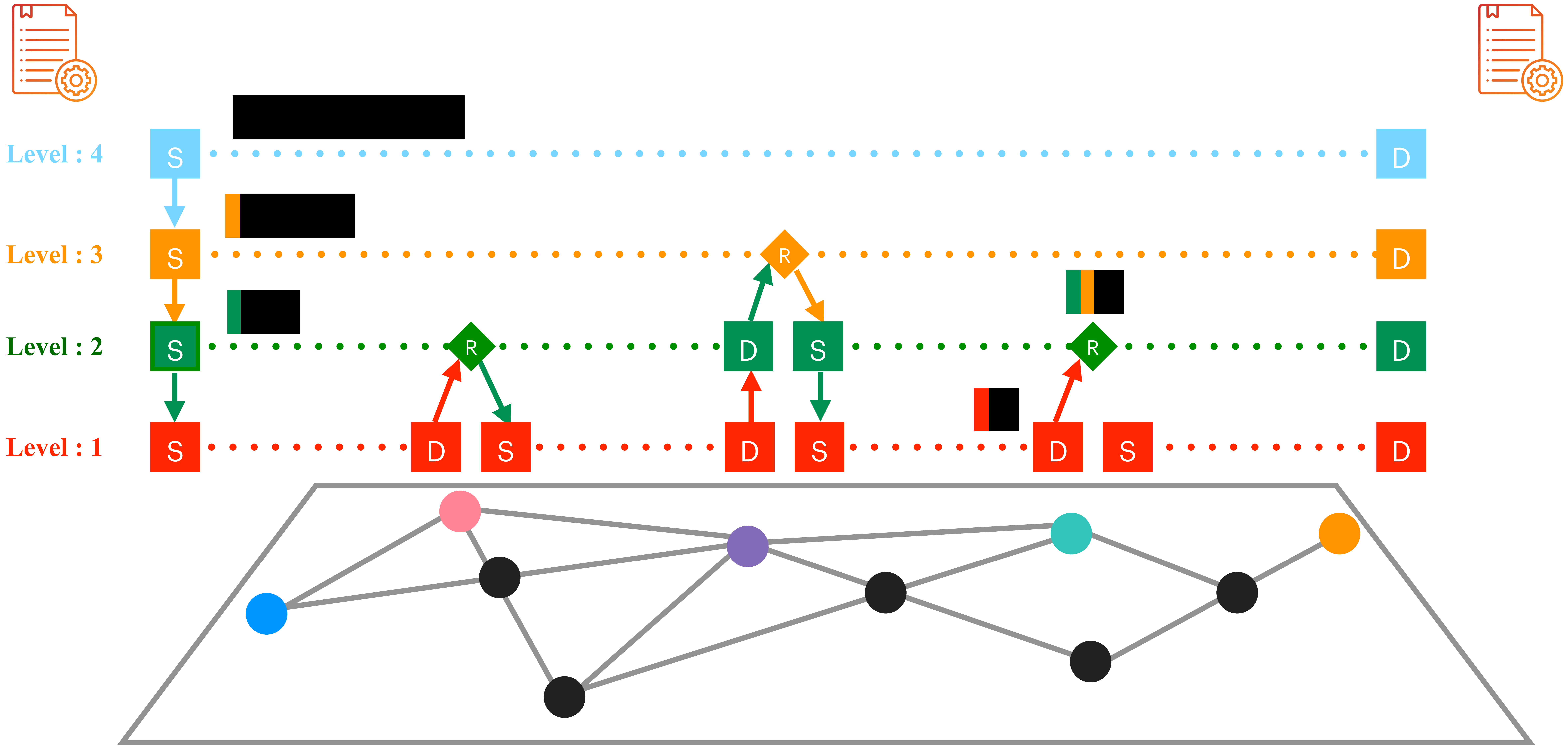
MLED File Transfer



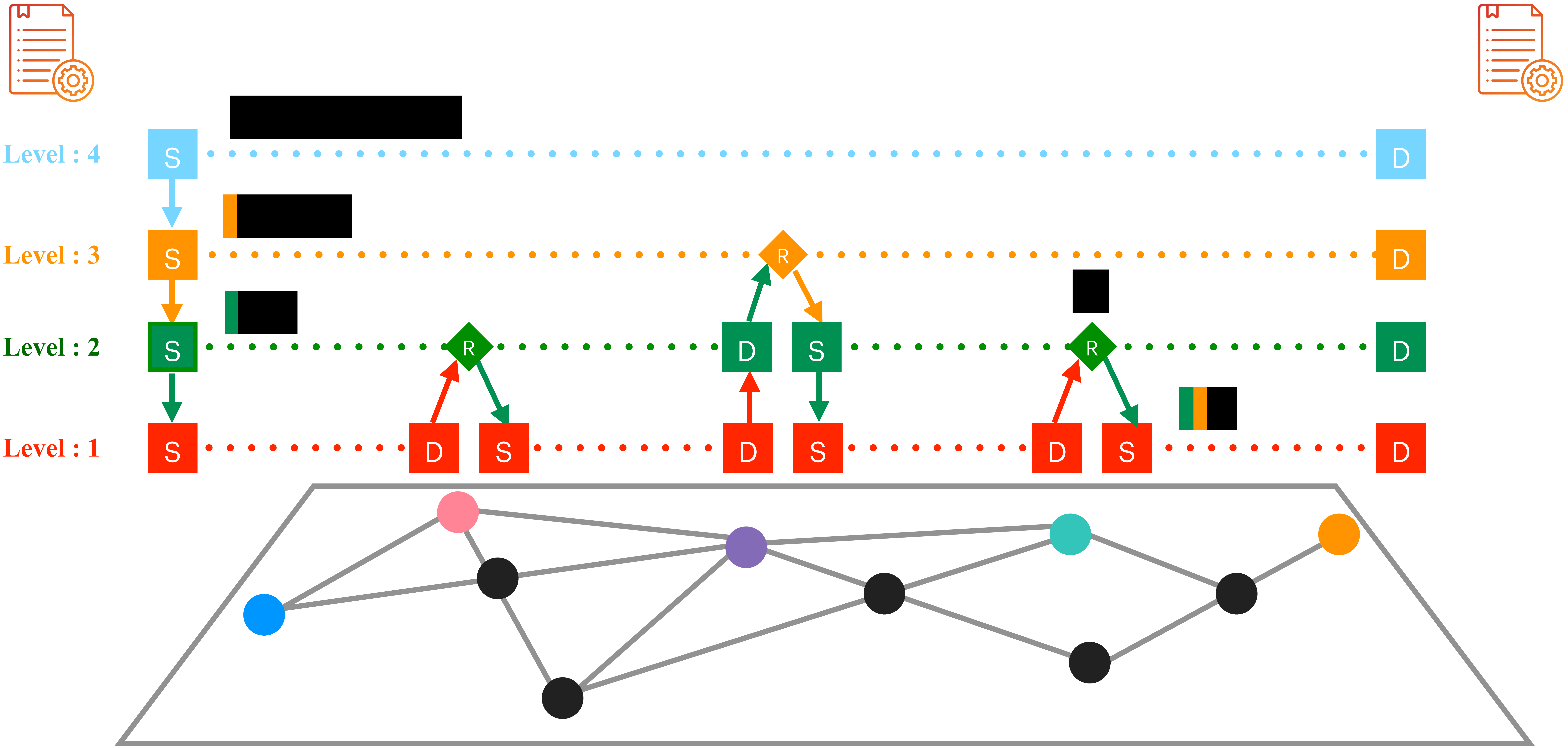
MLED File Transfer



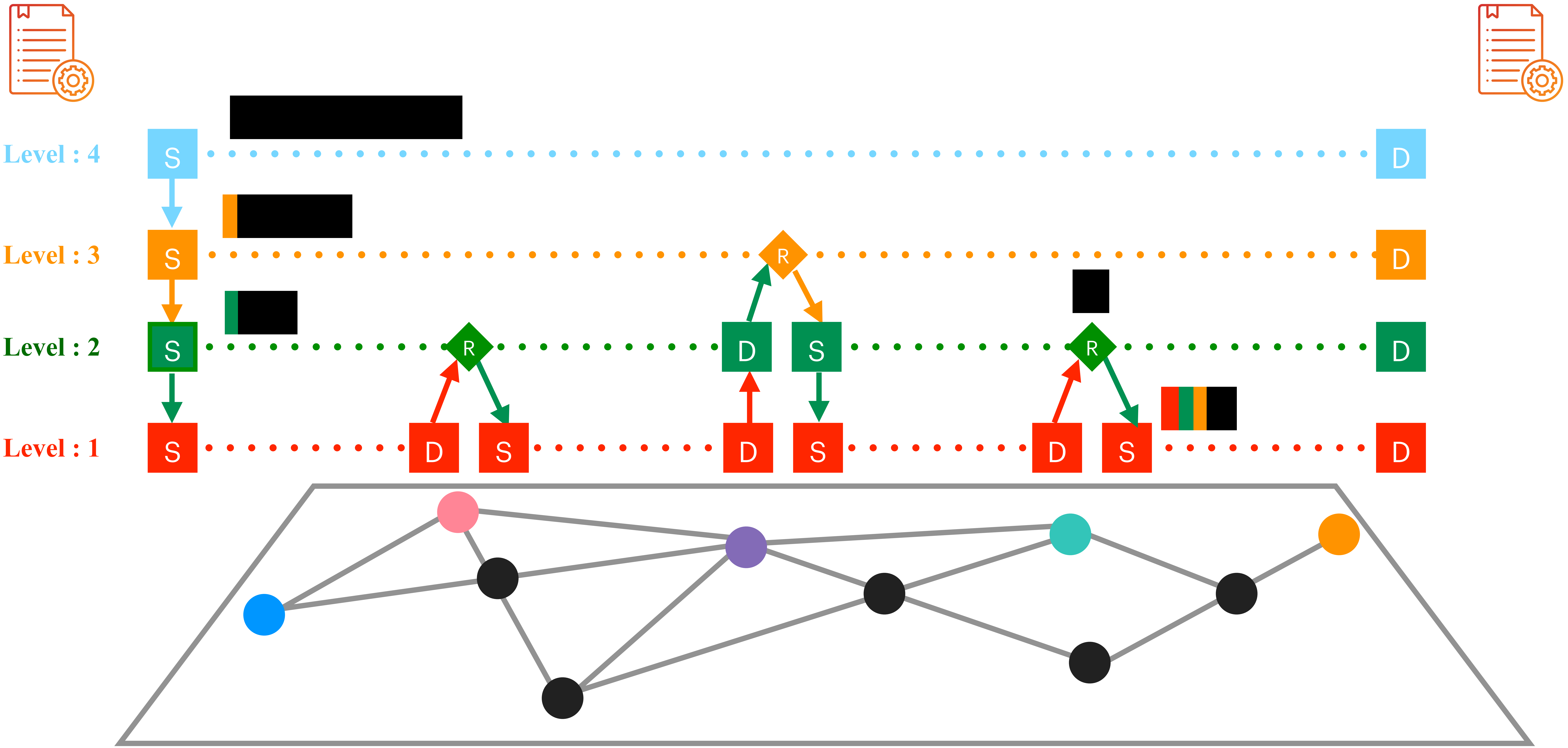
MLED File Transfer



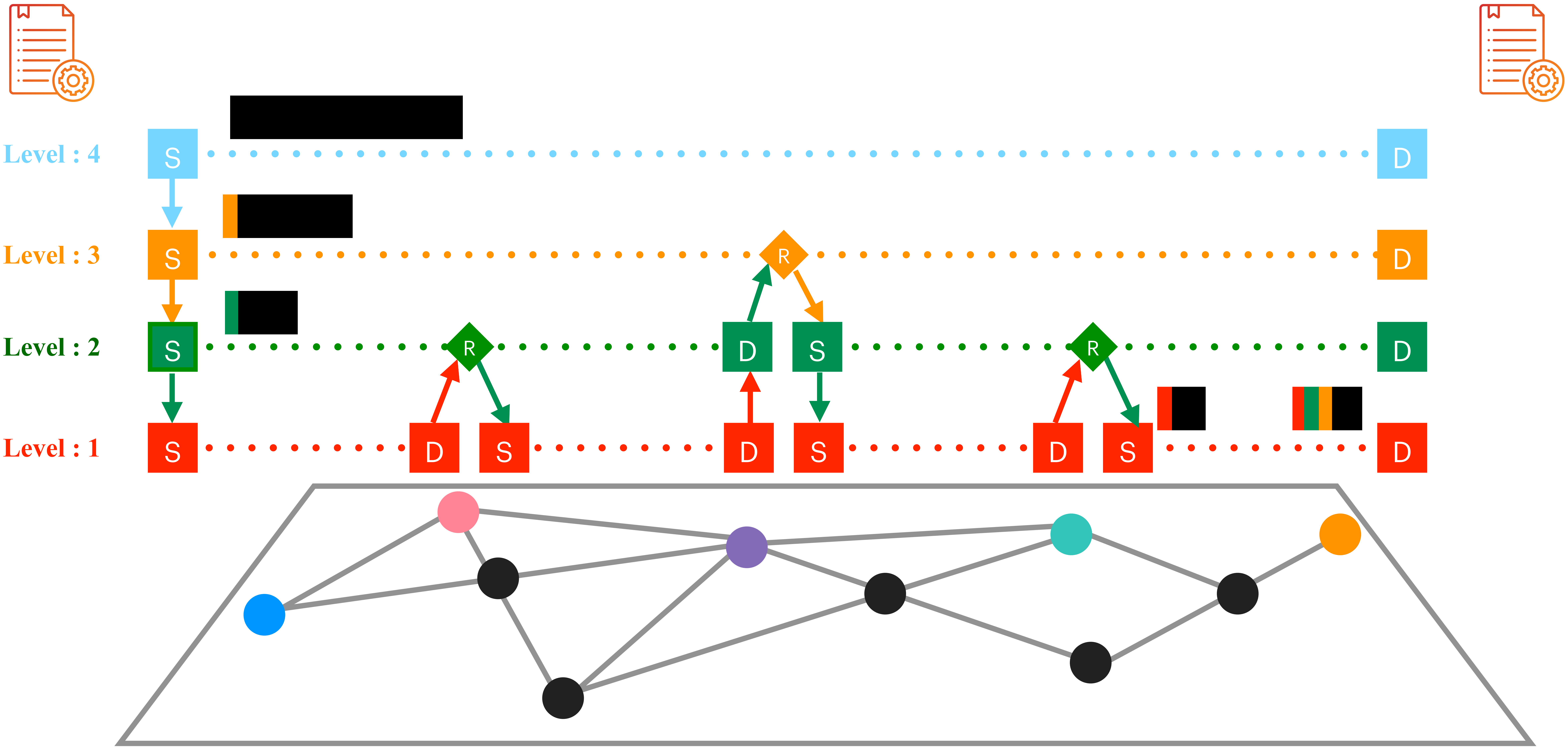
MLED File Transfer



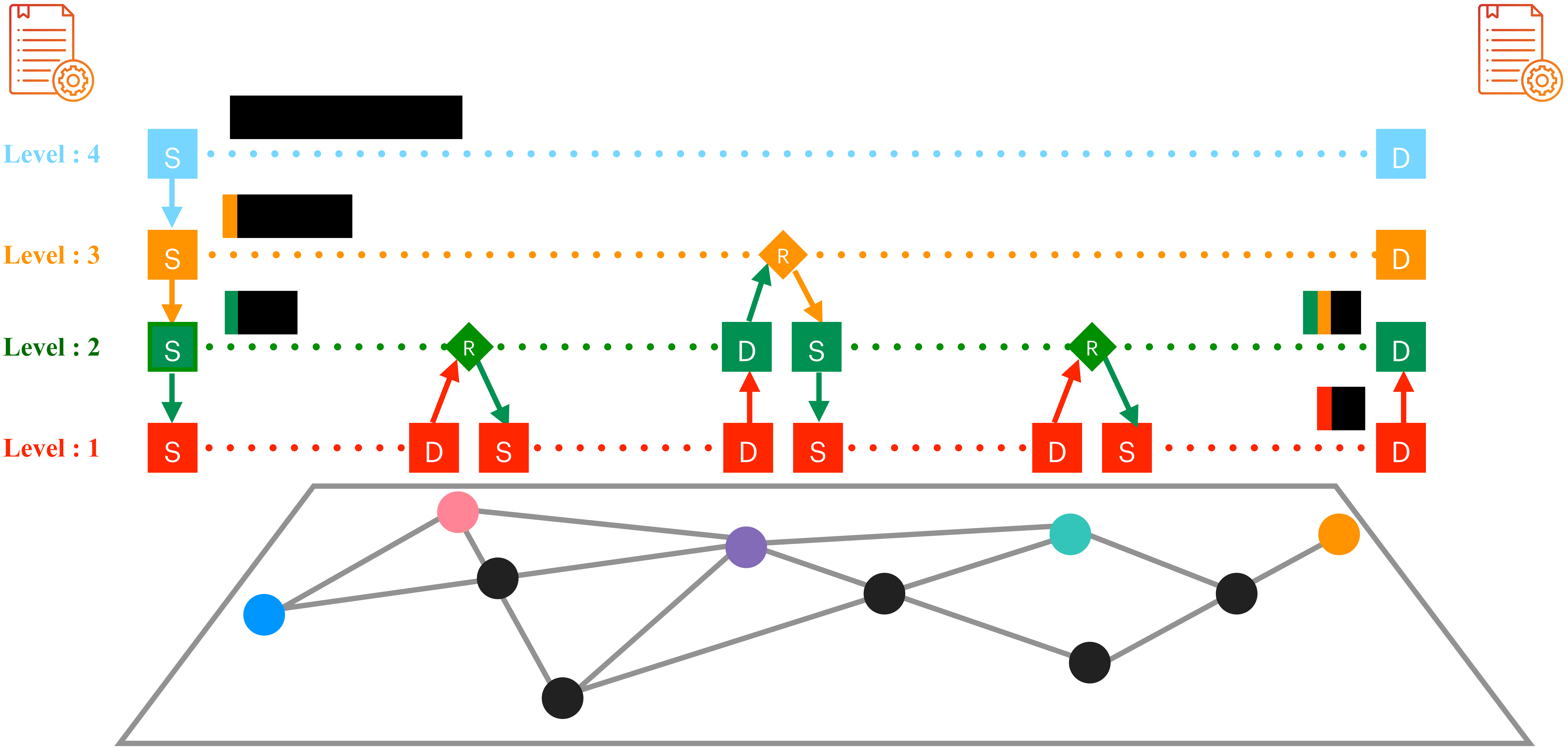
MLED File Transfer



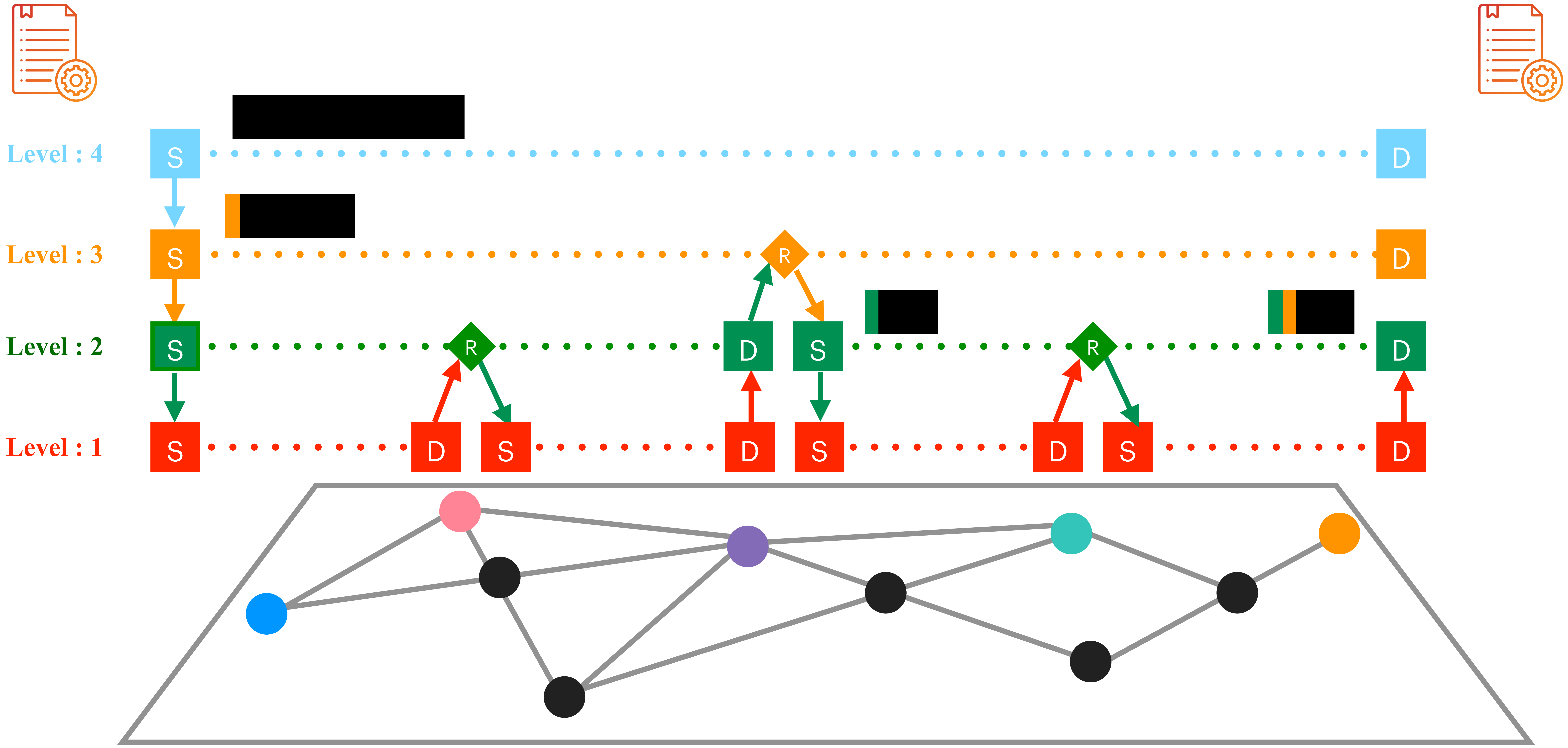
MLED File Transfer



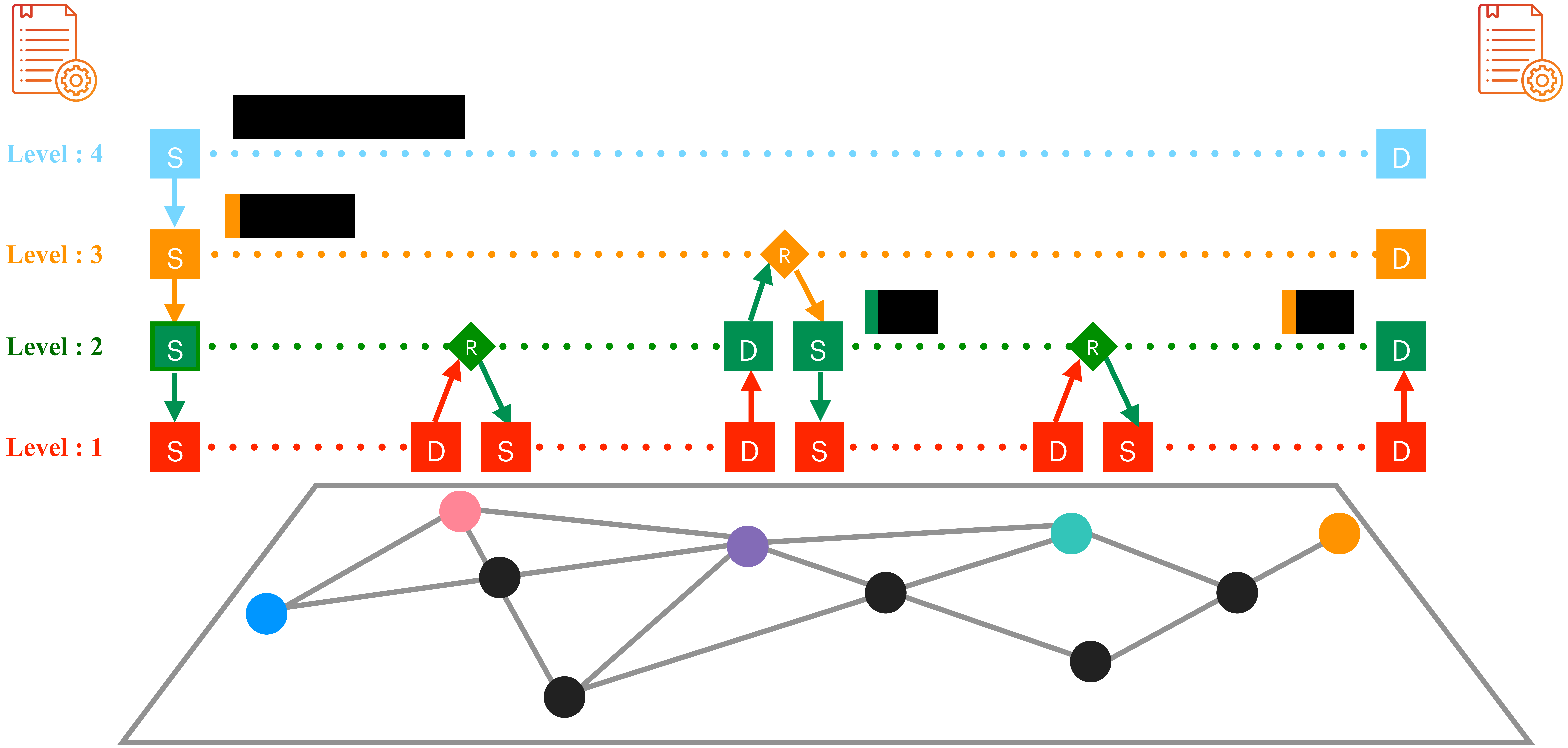
MLED File Transfer



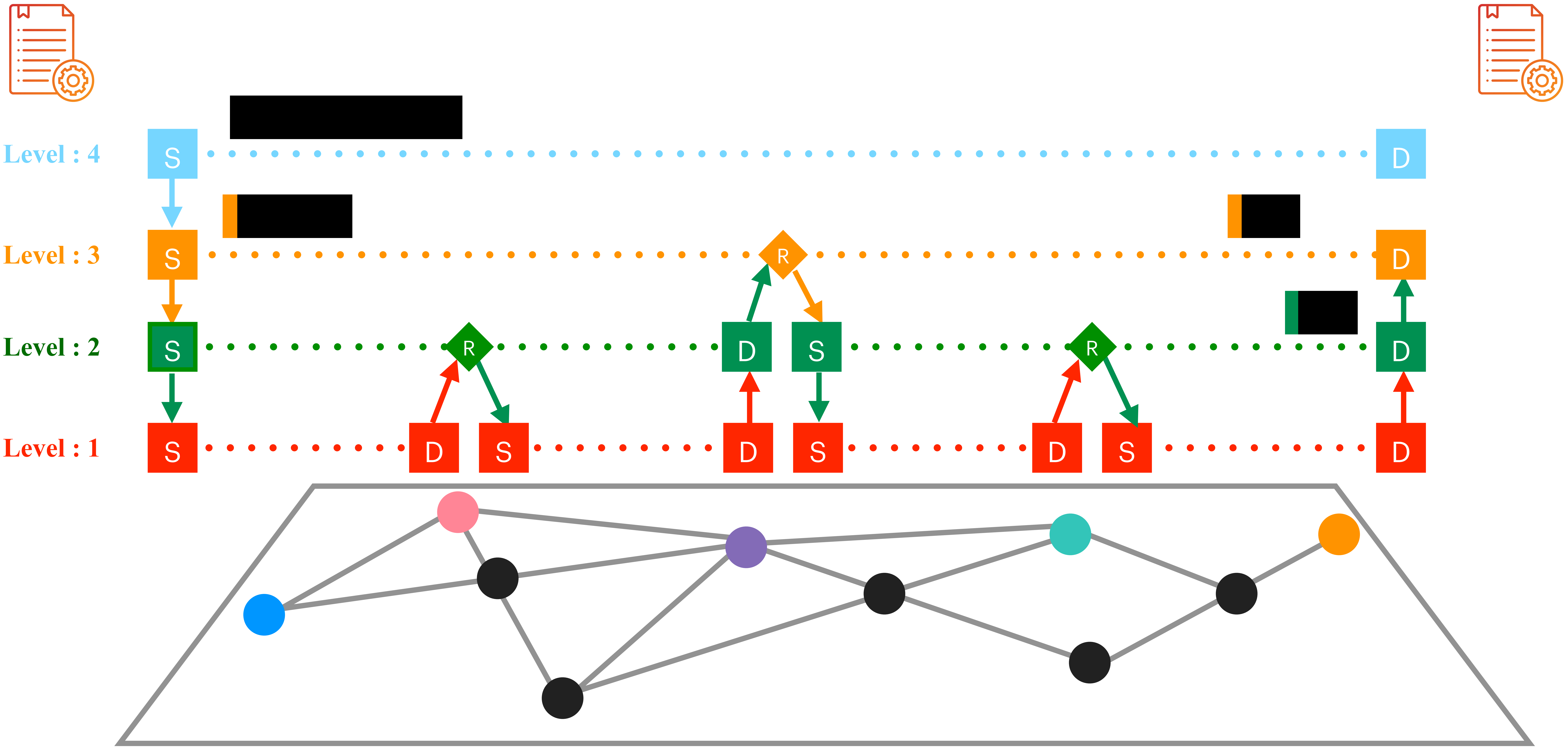
MLED File Transfer



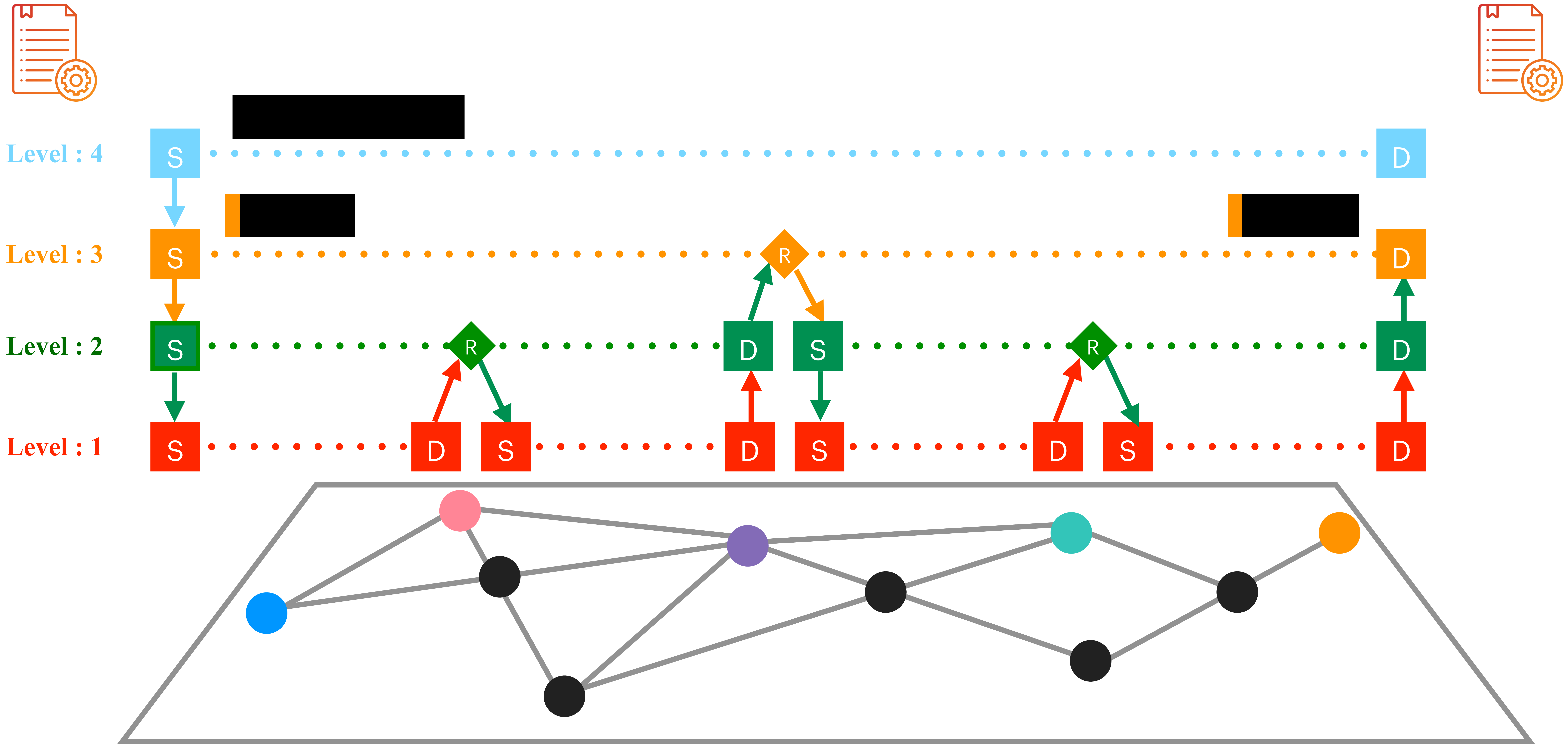
MLED File Transfer



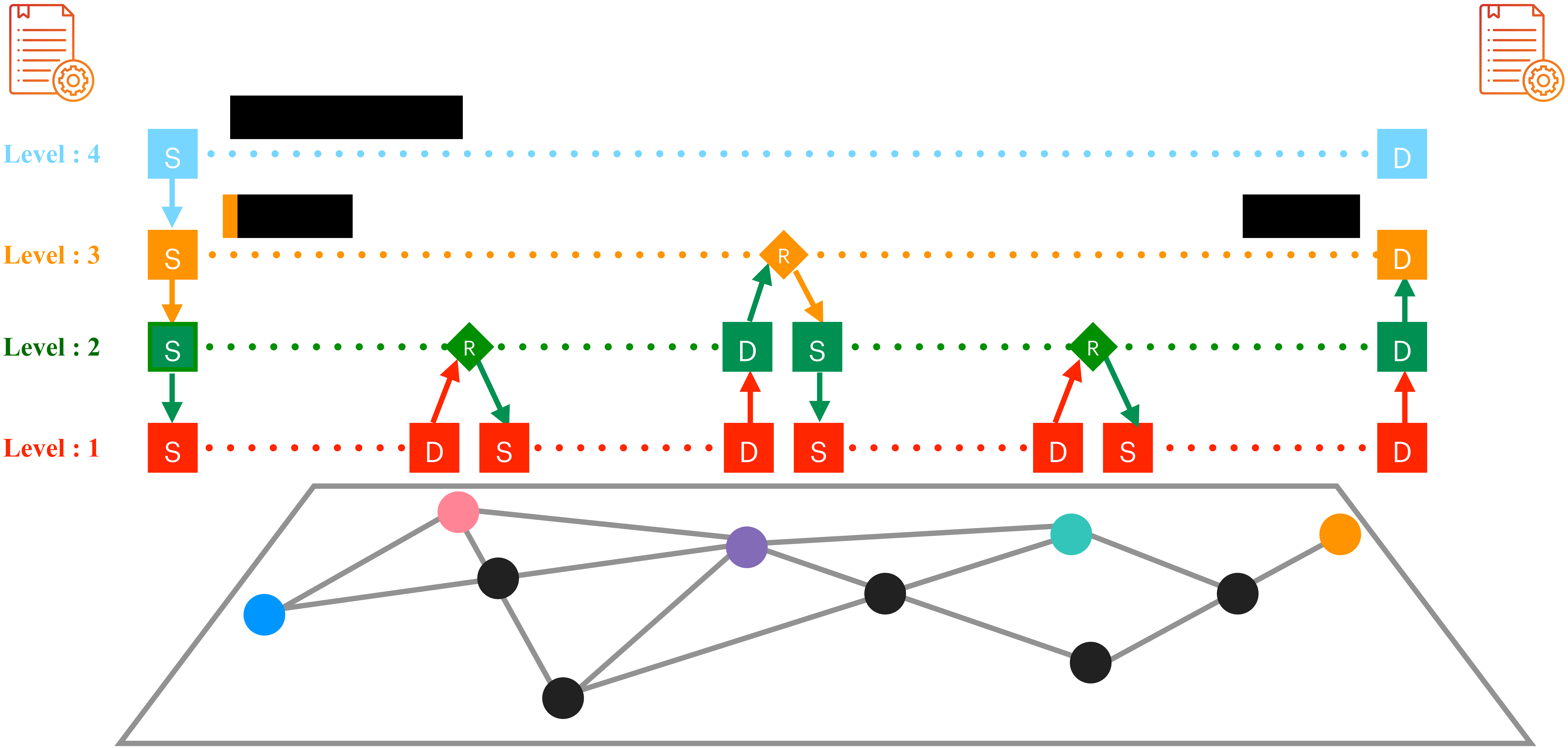
MLED File Transfer



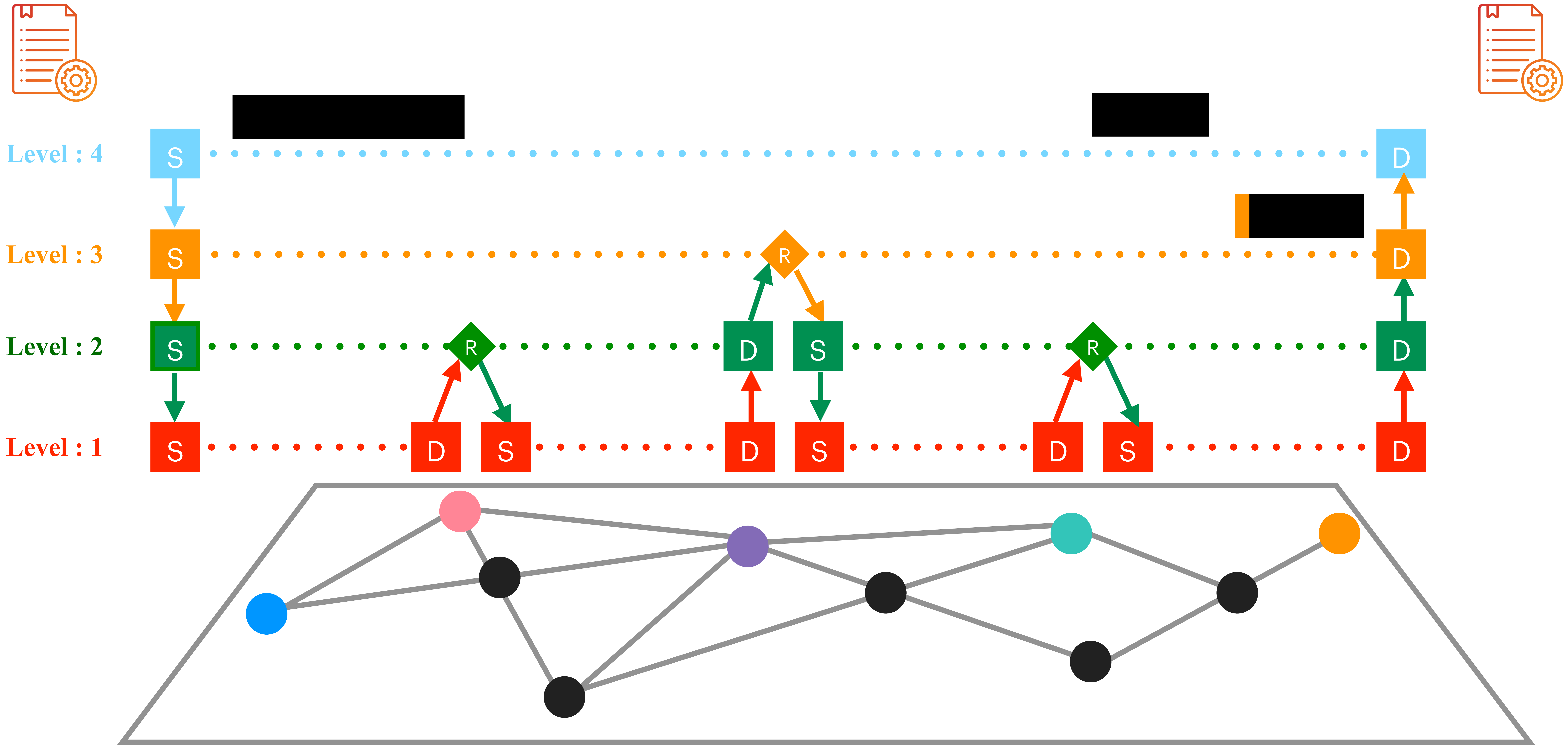
MLED File Transfer



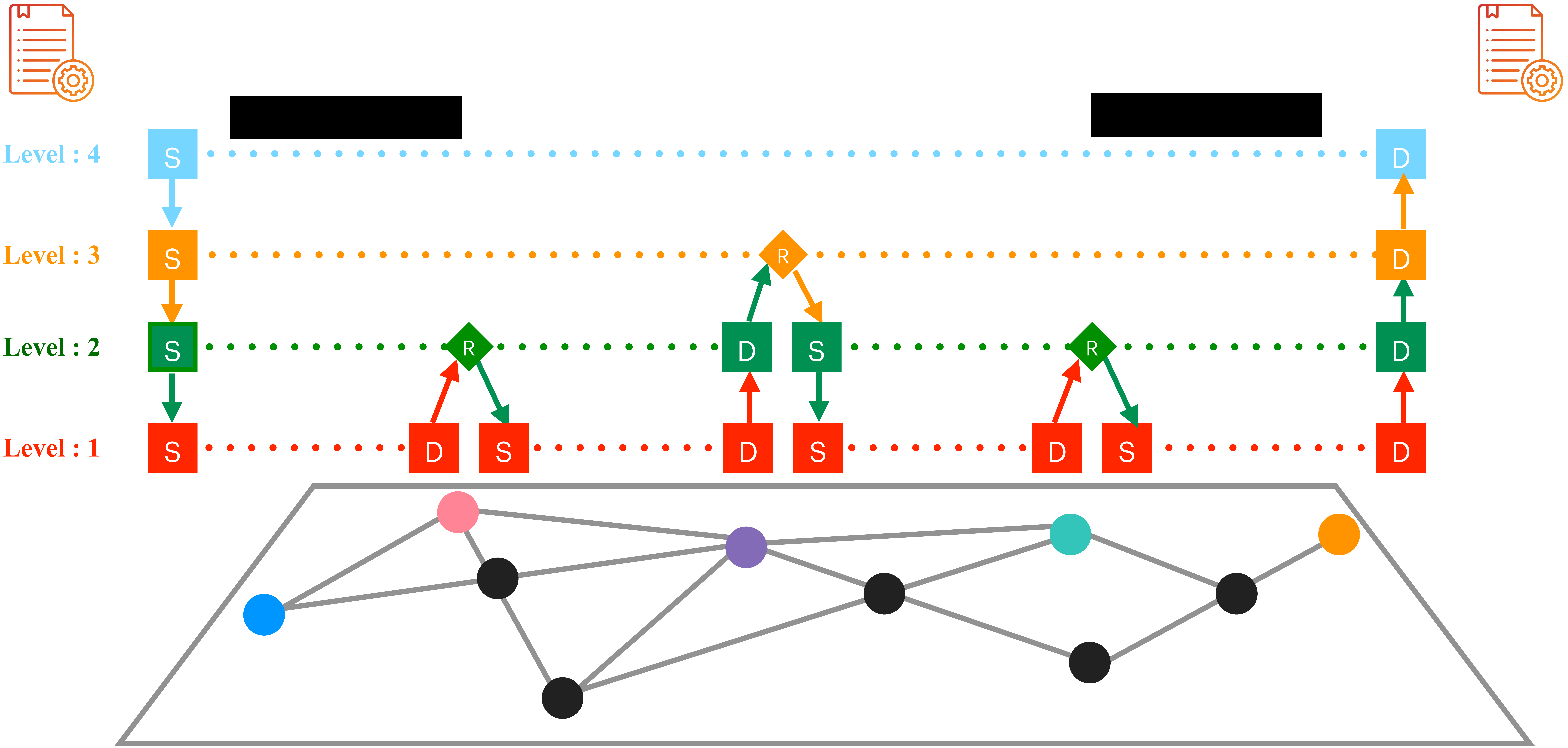
MLED File Transfer



MLED File Transfer



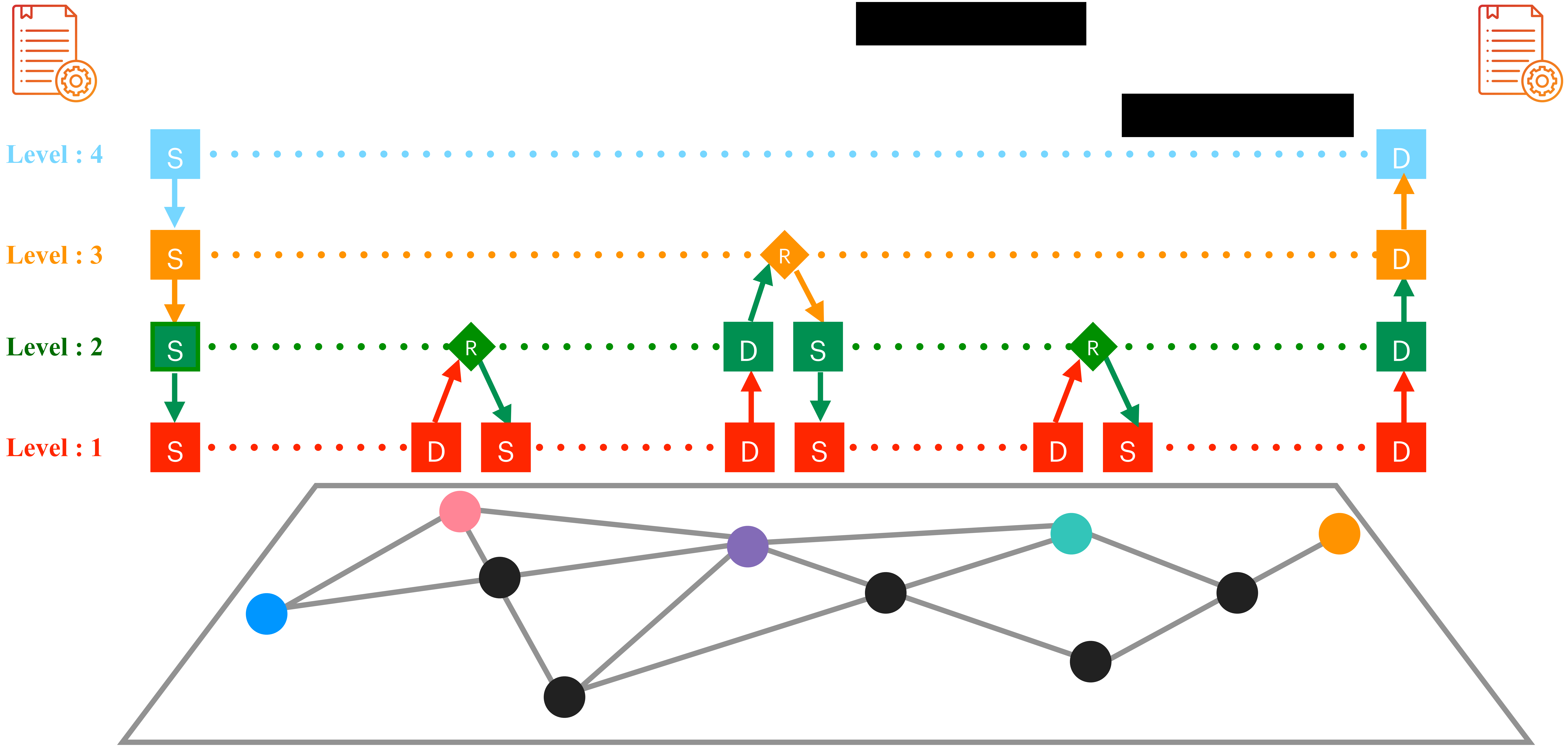
MLED File Transfer



MLED File Transfer

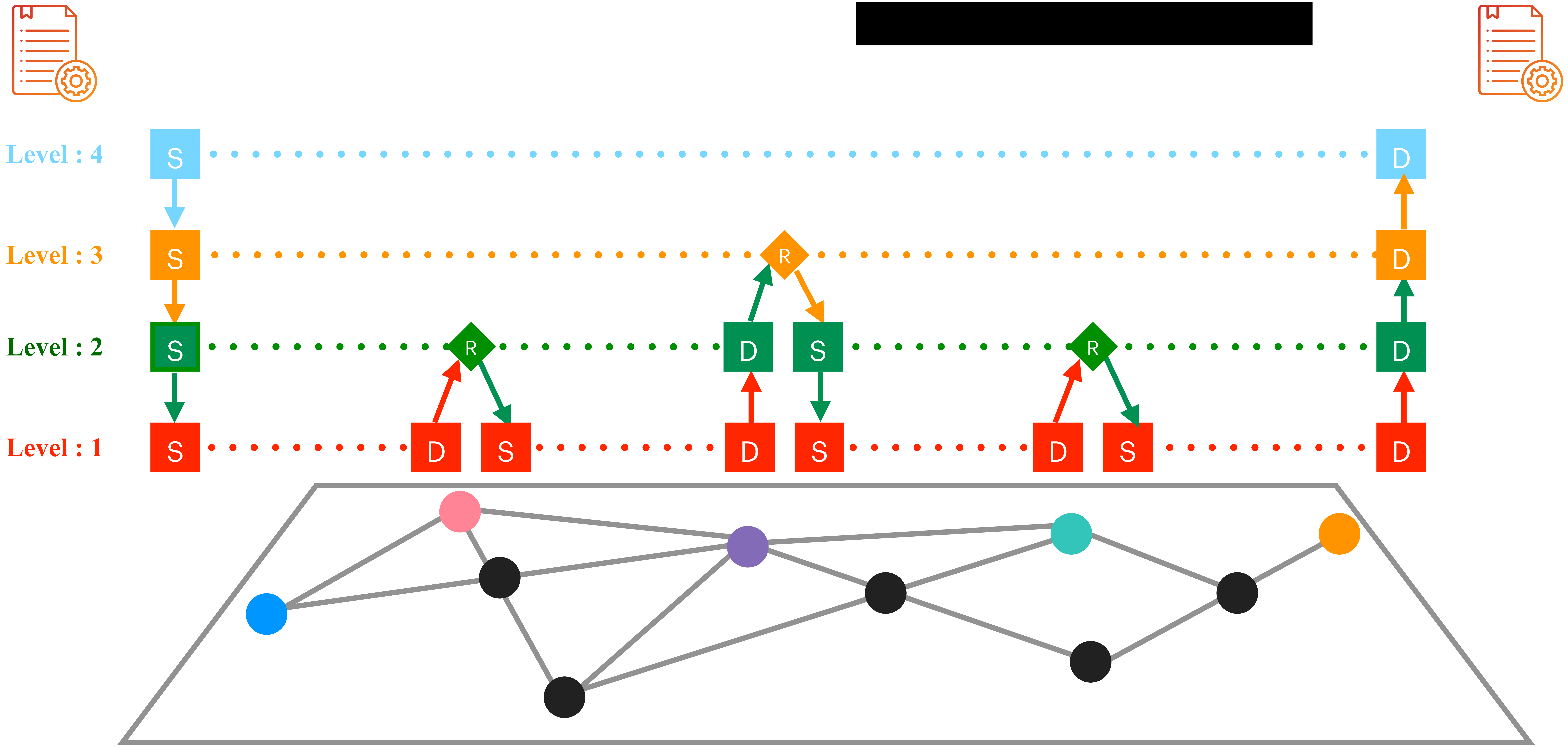
[redacted]

[redacted]

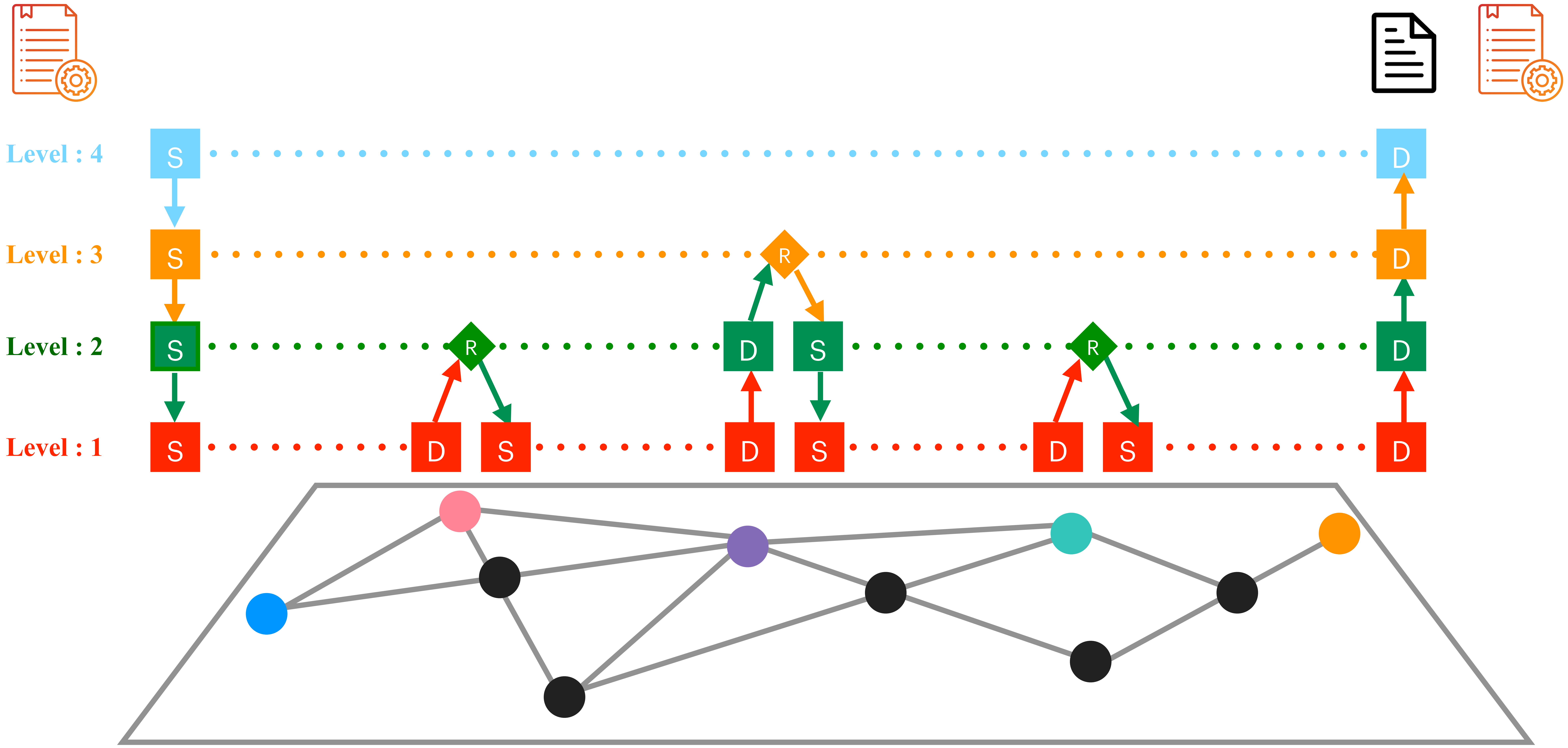


MLED File Transfer

[redacted]



MLED File Transfer



Mathematical Model of MLED

- Let $\mathbf{ID_{ij}}$ be the set of layer identifiers at level $\boldsymbol{i-1}$ that realizes the operation of layer $\boldsymbol{L_{ij}}$. We define effective UEP for a layer $\boldsymbol{L_{ij}}$ as:

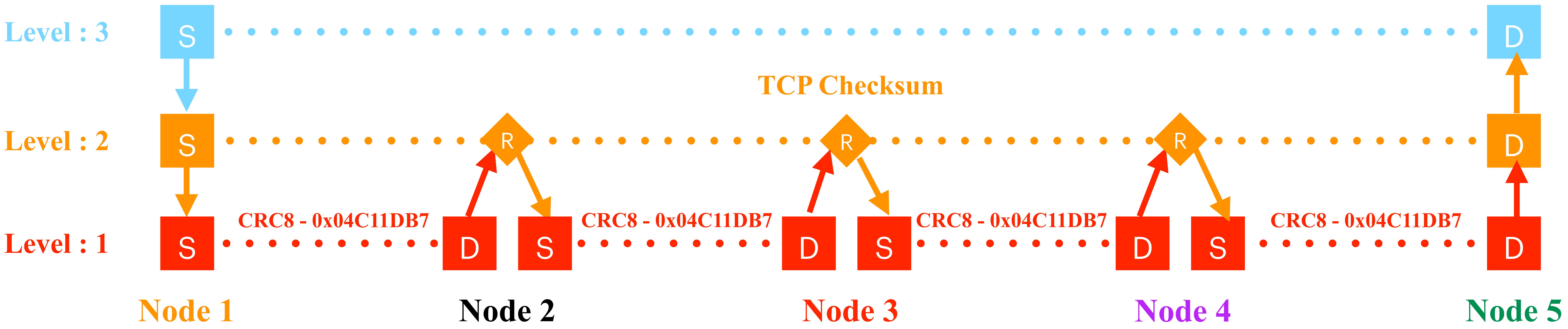
$$E_{\text{UEP}}(L_{ij}, \text{ID}_{ij}) = \begin{cases} \text{UEP}(L_{ij}) \cdot \max\limits_{j \in \text{ID}_{ij}} \left\{ E_{\text{UEP}}(L_{(i-1)j}, \text{ID}_{(i-1)j}) \right\}, & \text{if } \text{ID}_{ij} \neq \emptyset \\ \text{UEP}(L_{ij}), & \text{otherwise} \end{cases}$$

- We define the reduction factor in UEP β as the reduction in UEP from the inclusion of extra levels in MLED.
- β quantifies the improvement in error detection by extending MLED(n, P) with additional levels to obtain MLED(n, P) and is expressed as:

$$\beta = \frac{UEP_{MLED(n',P')}}{UEP_{MLED(n,P)}} = \frac{\frac{1}{2^{\sum_{i=1}^{n'} \ell_i}}}{\frac{1}{2^{\sum_{j=1}^{n} \ell_j}}} = \frac{\frac{1}{2^{\sum_{i=1}^{n} \ell_i}} \cdot \frac{1}{2^{\sum_{k=1}^{m} \ell_k}}}{\frac{1}{2^{\sum_{j=1}^{n} \ell_j}}} = \frac{1}{2^{\sum_{k=1}^{m} \ell_k}} < 1$$

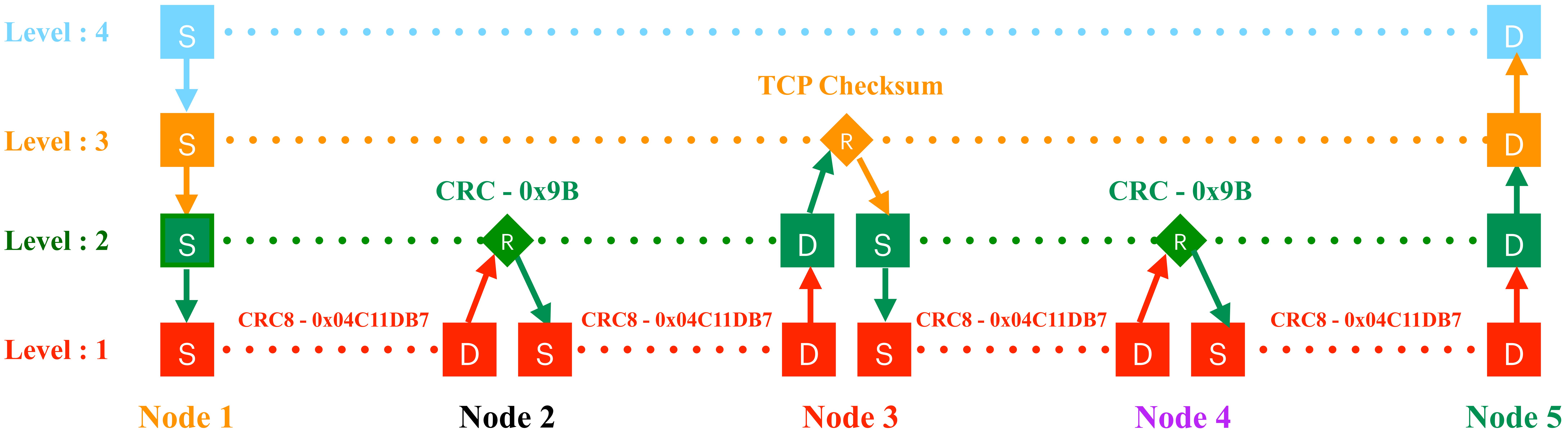
Experimental Setup

Traditional Approach



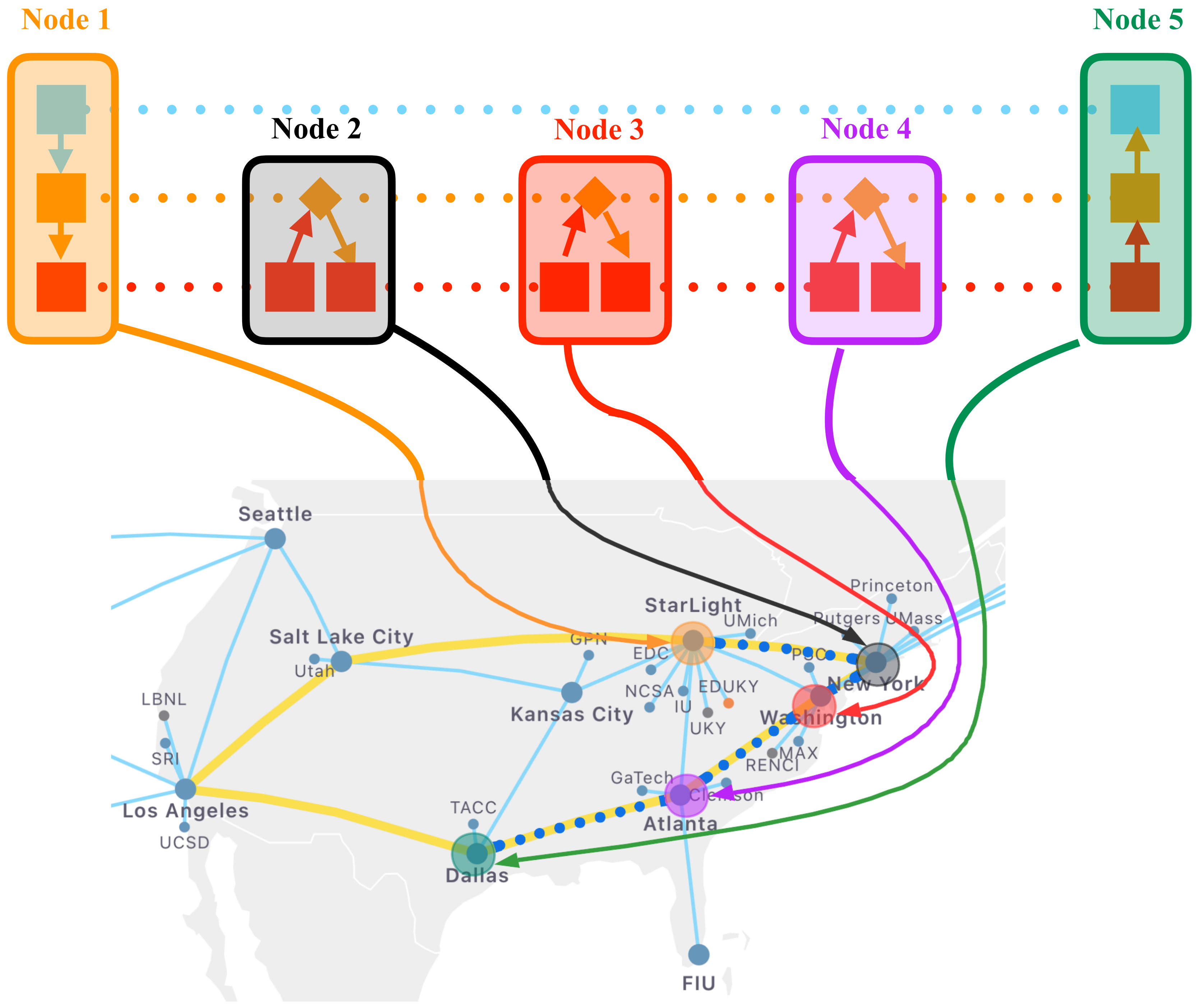
Experimental Setup

MLED Architecture : One Augmented Level

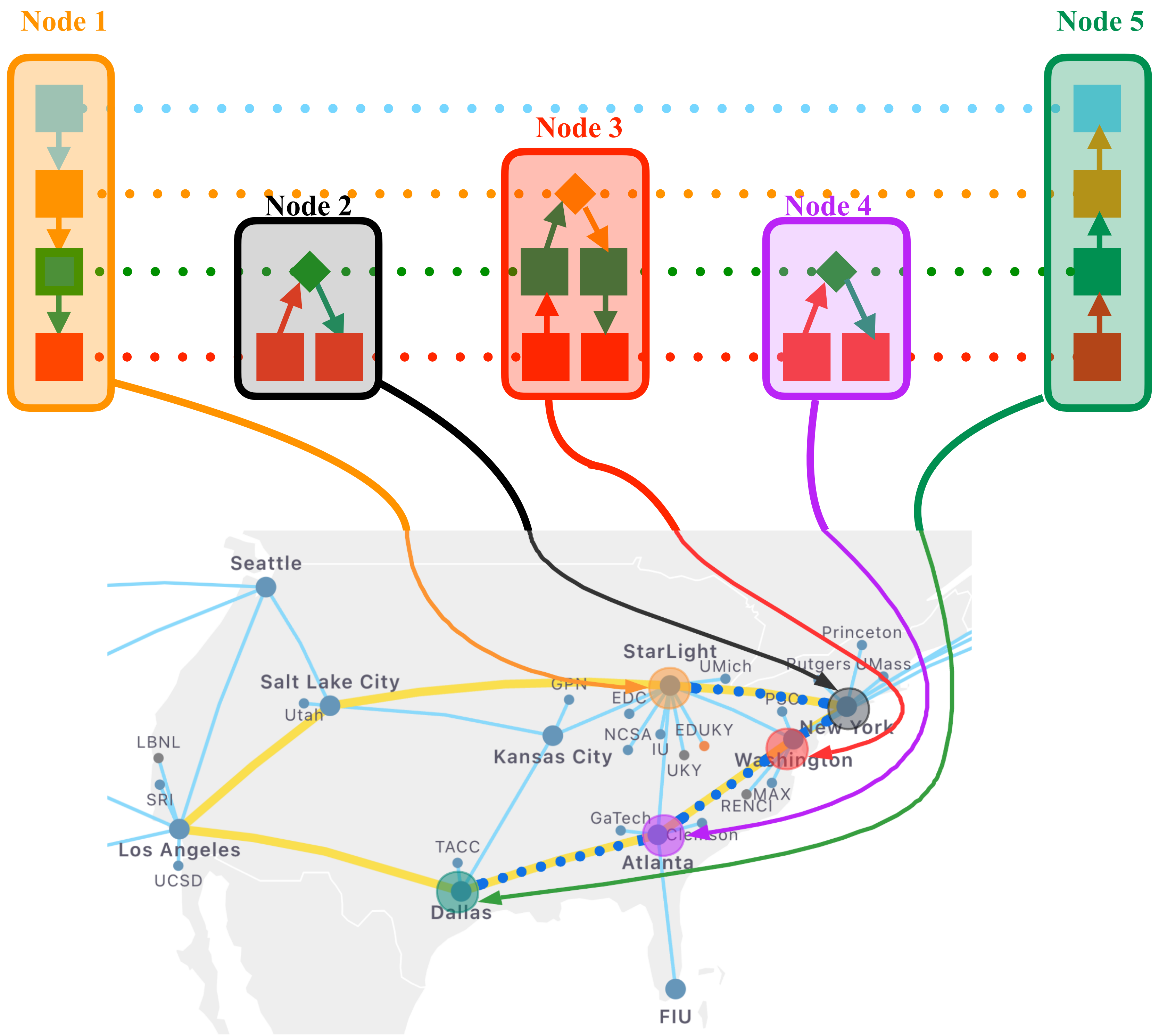


FABRIC

- FABRIC is an International testbed infrastructure
- Enables cutting-edge experimentation and research at-scale in the areas of networking, cybersecurity, distributed computing, storage, virtual reality, 5G, machine learning, and science applications



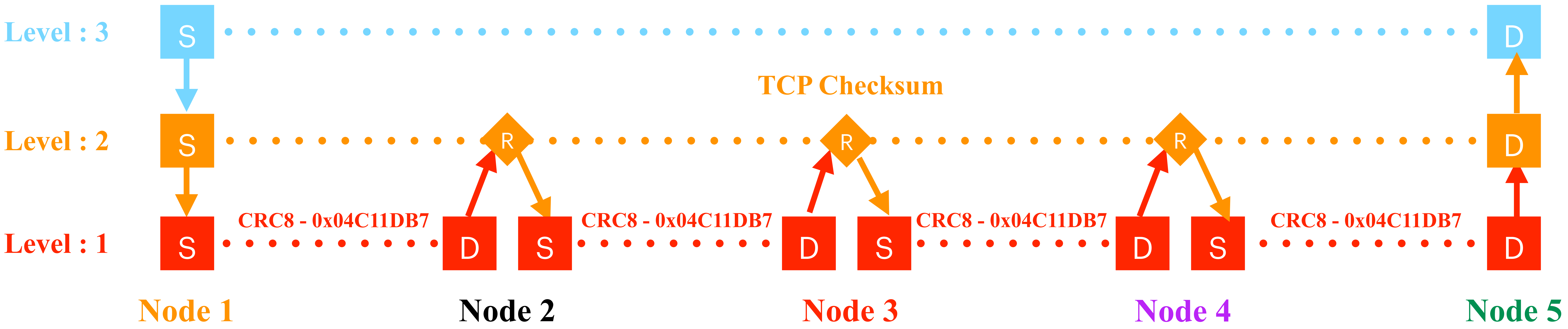
Traditional Architecture on FABRIC



MLED Architecture on FABRIC

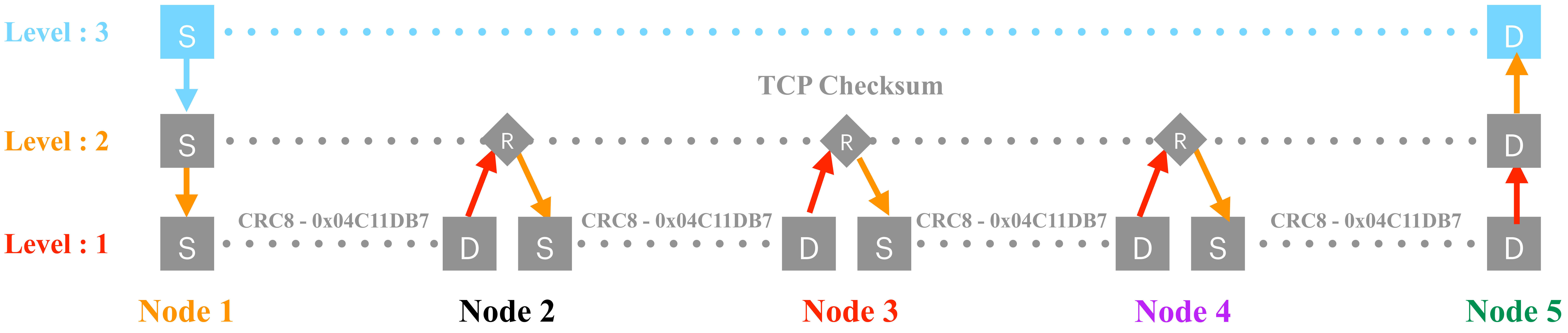
Adversarial Model for Error Introduction

Traditional Architecture : Errors Undetectable by both CRC and TCP Checksum



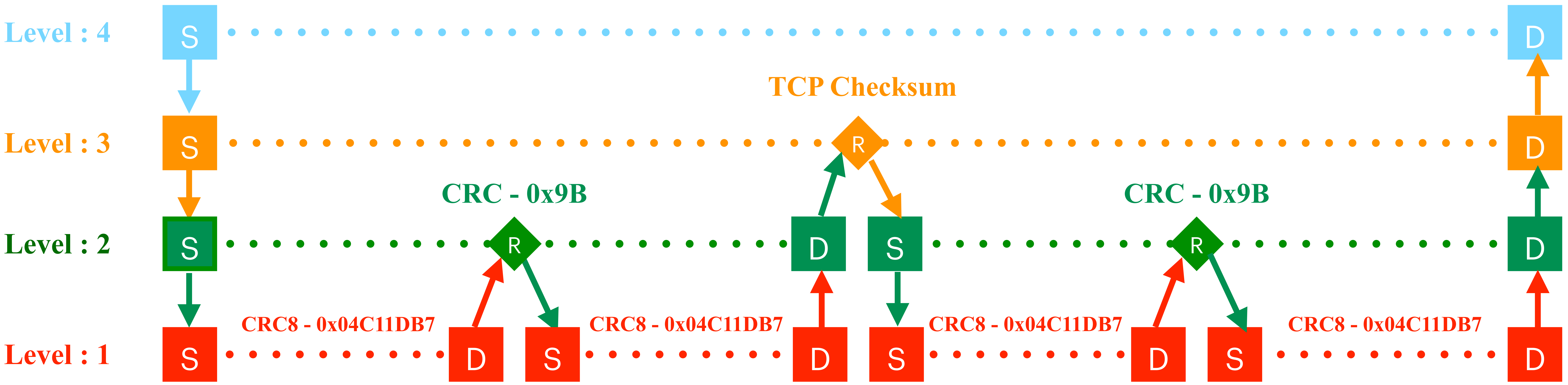
Adversarial Model for Error Introduction

Traditional Architecture : Errors Undetectable by both CRC and TCP Checksum



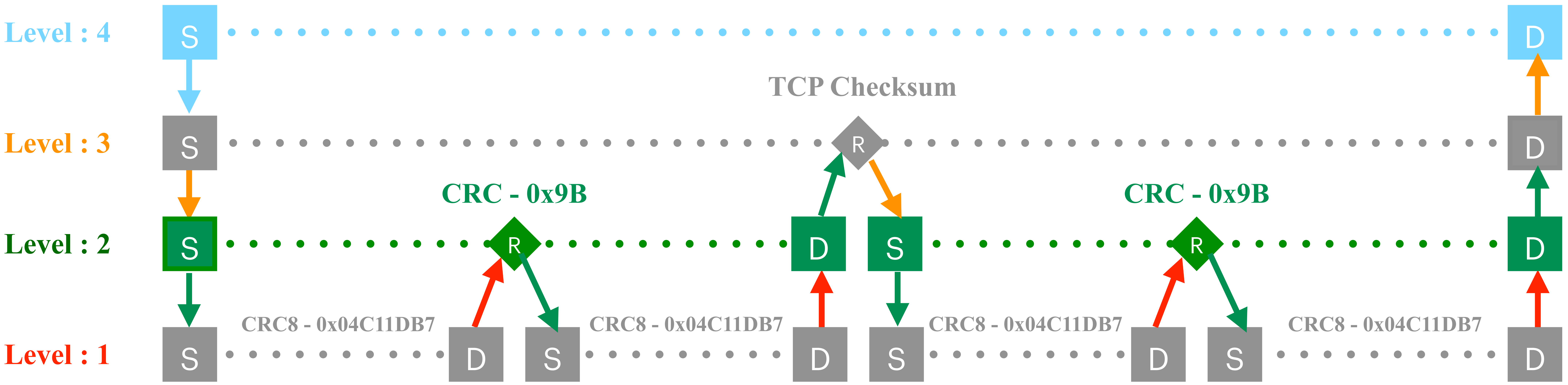
Adversarial Model for Error Introduction

MLED Architecture : Errors Undetectable by both CRC and TCP Checksum



Adversarial Model for Error Introduction

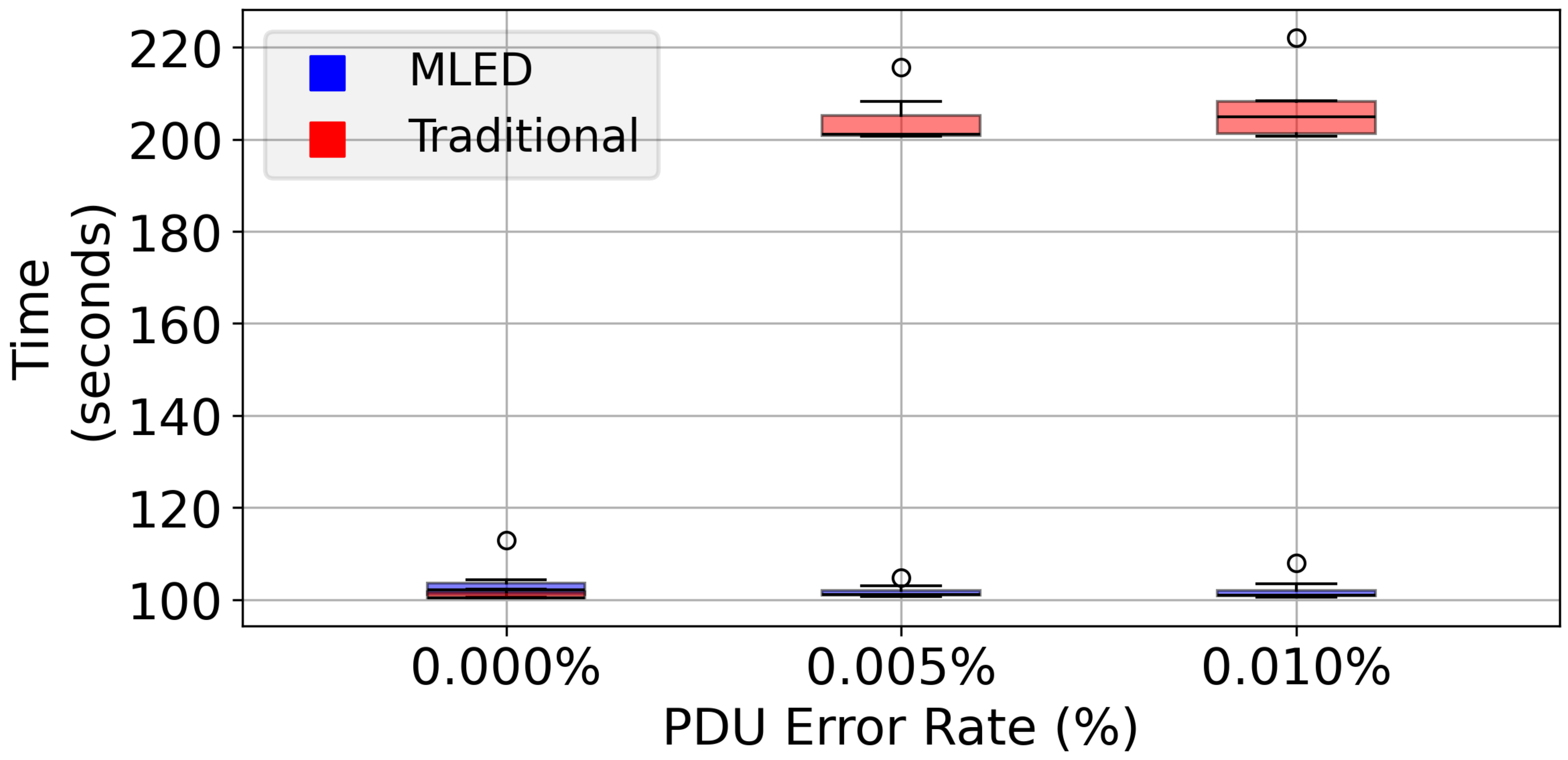
MLED Approach : Errors Undetectable by both CRC and TCP Checksum



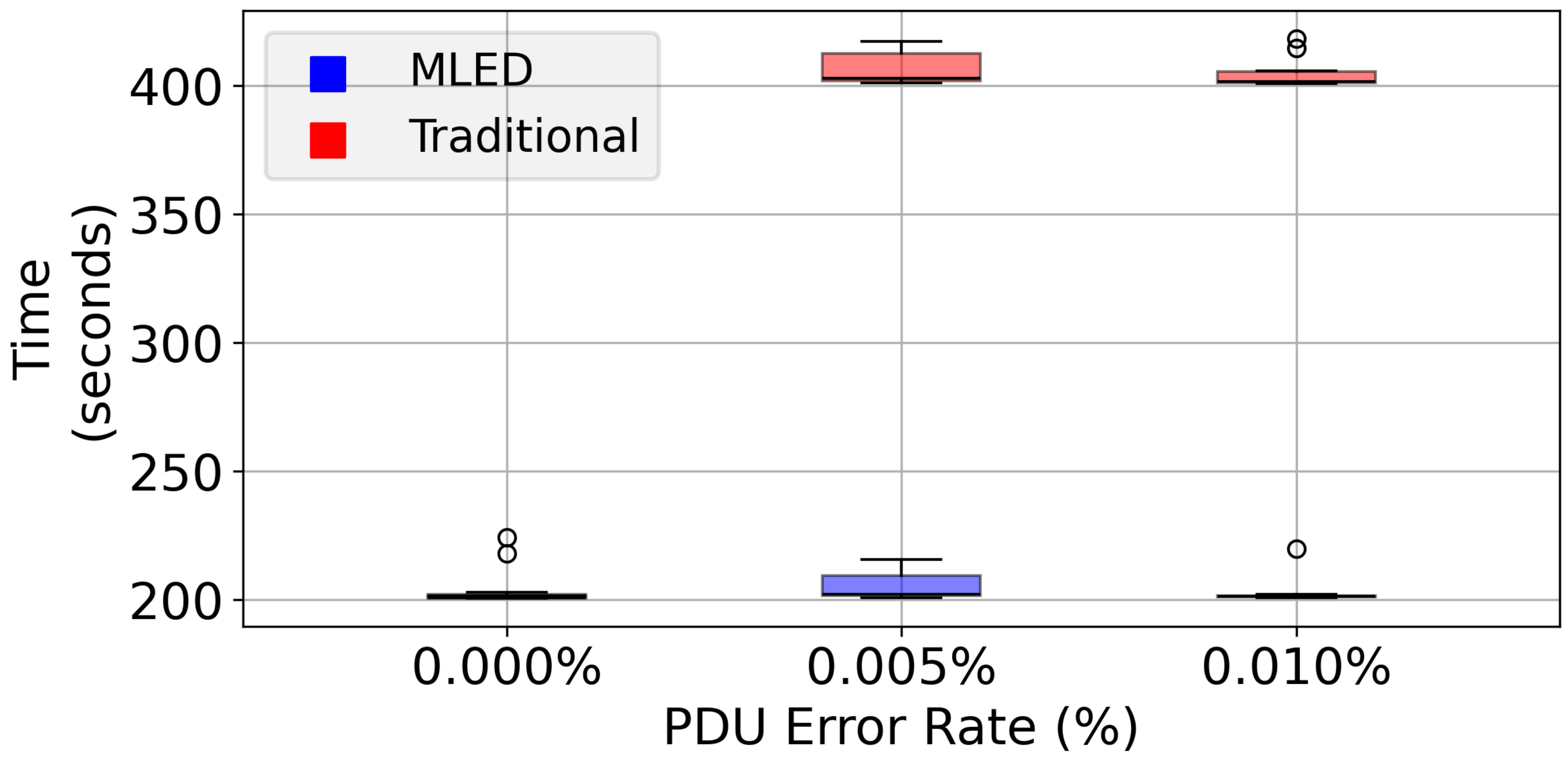
Results

File Delivery Time for given PERs and file sizes under MLED and the traditional approach

- If the final file is corrupt, we double the transmission time considering that the file has to be retransmitted



10 GB File

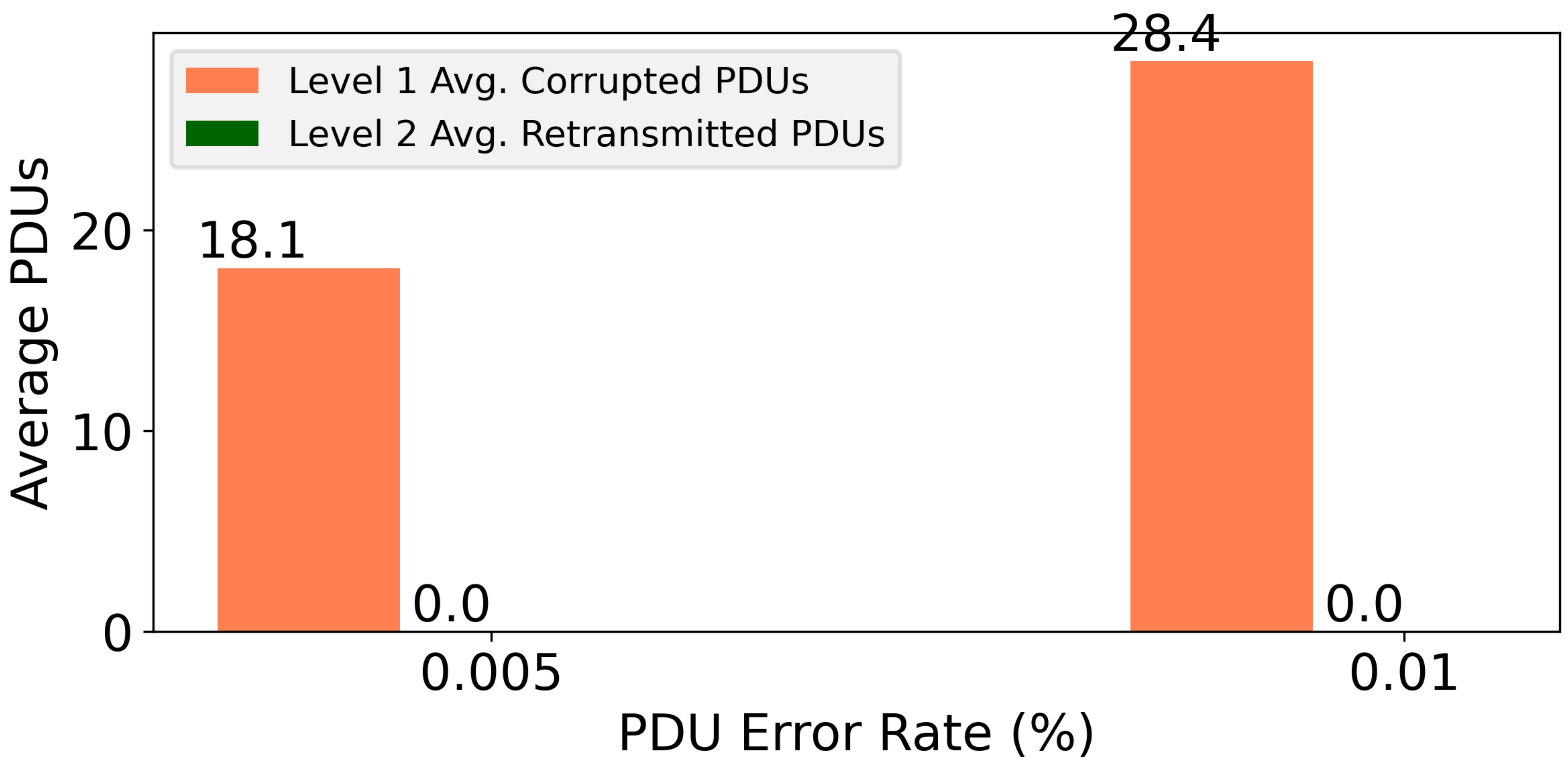


20 GB File

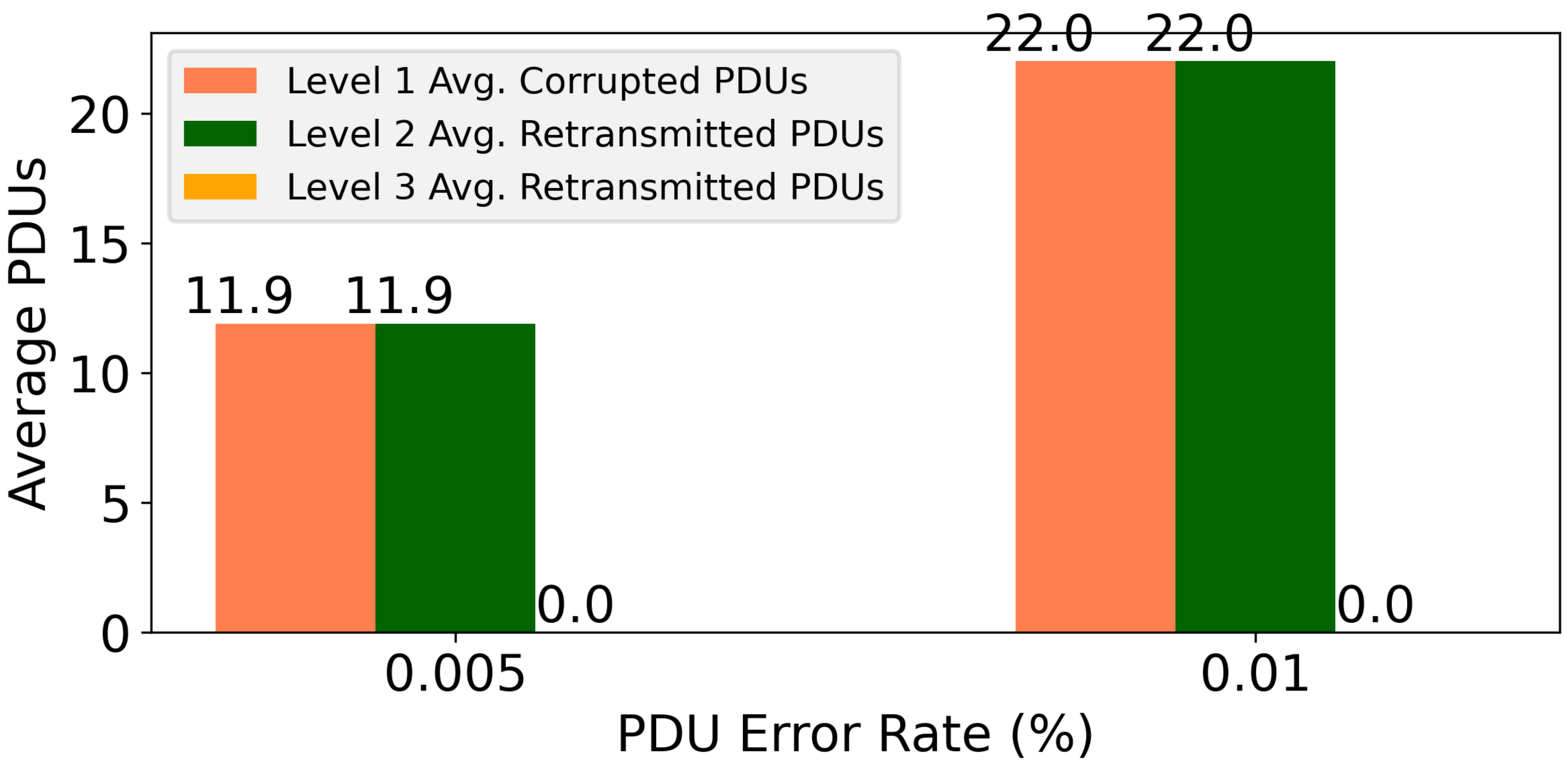
- **Time taken from** sending the request to read the first block of data **till** the last uncorrupted block is received at the destination
- For large size files, MLED takes half the time to deliver the file with non-zero PER

Results

Average number of corrupted and retransmitted PDUs at different levels for a 20 GB file



Traditional Approach



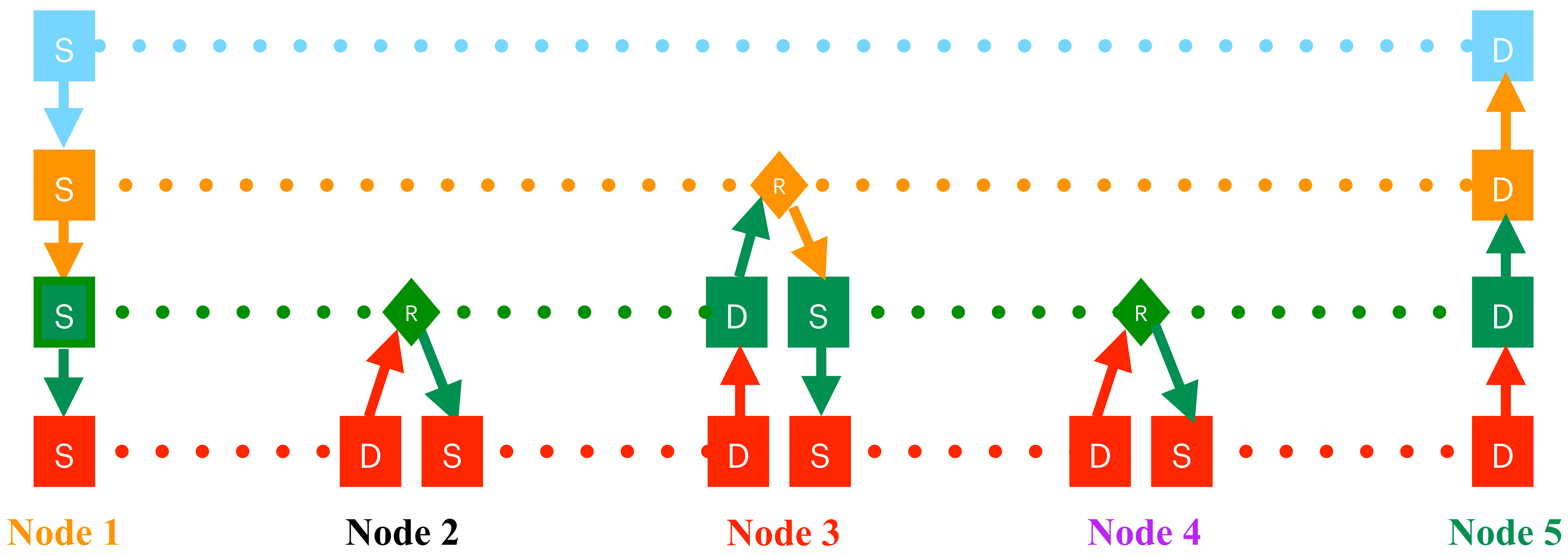
MLED Approach

- Under the traditional approach, errors introduced at level 1 are not detected at level 2
- Under MLED, all errors introduced at level 1 are detected at level 2 with a simple 8-bit CRC check with `0x9B` as the generator polynomial

Results

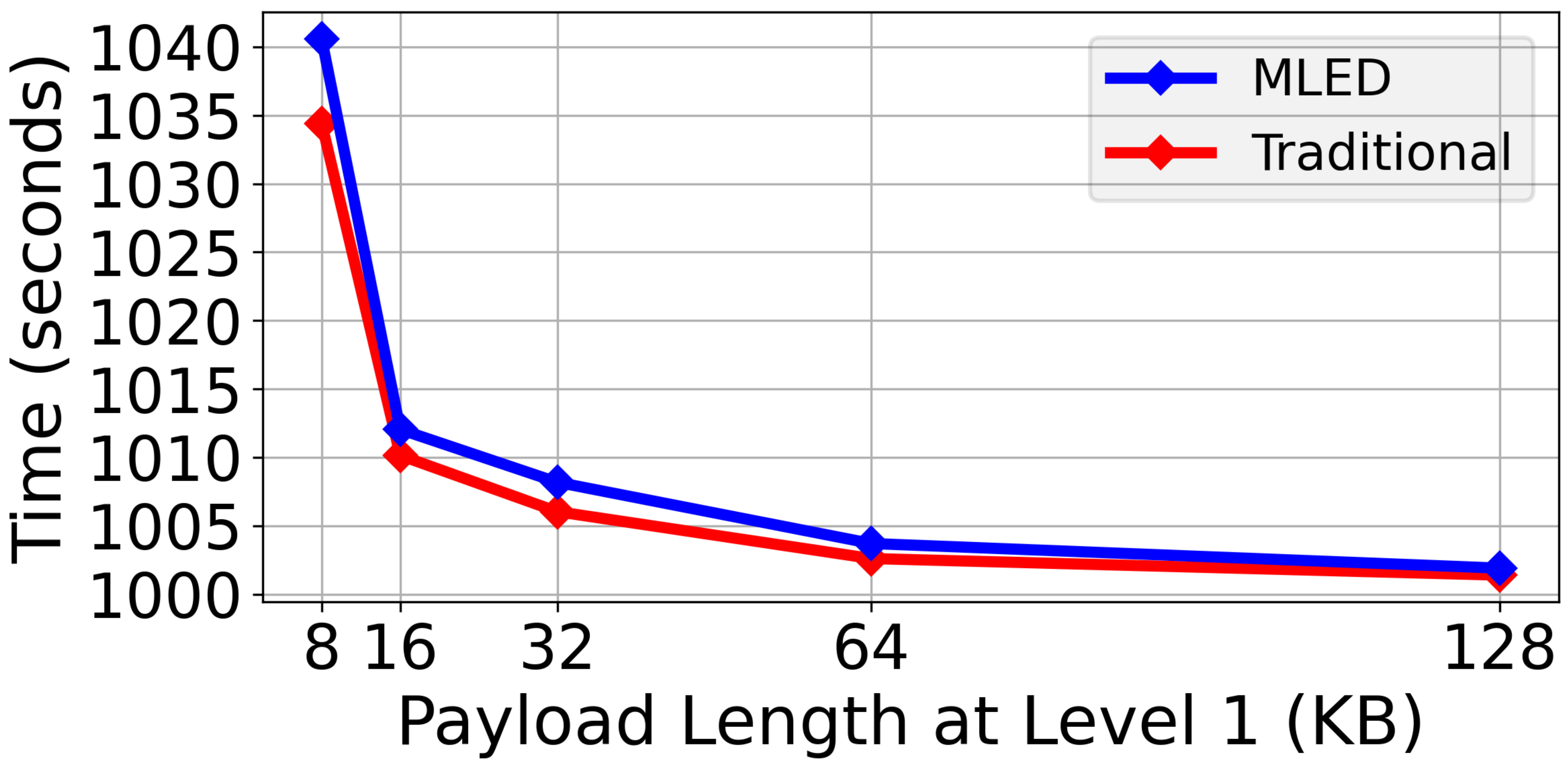
Time taken to transfer a 20 GB file as a function of payload length at level 1 for different error rates

- Errors are undetectable only at level 1 and can be detected at level 2 in both approaches
  - Traditional — at node 5
  - MLED — at node 3
- Plots show how faster recovery makes up for the additional processing required under MLED

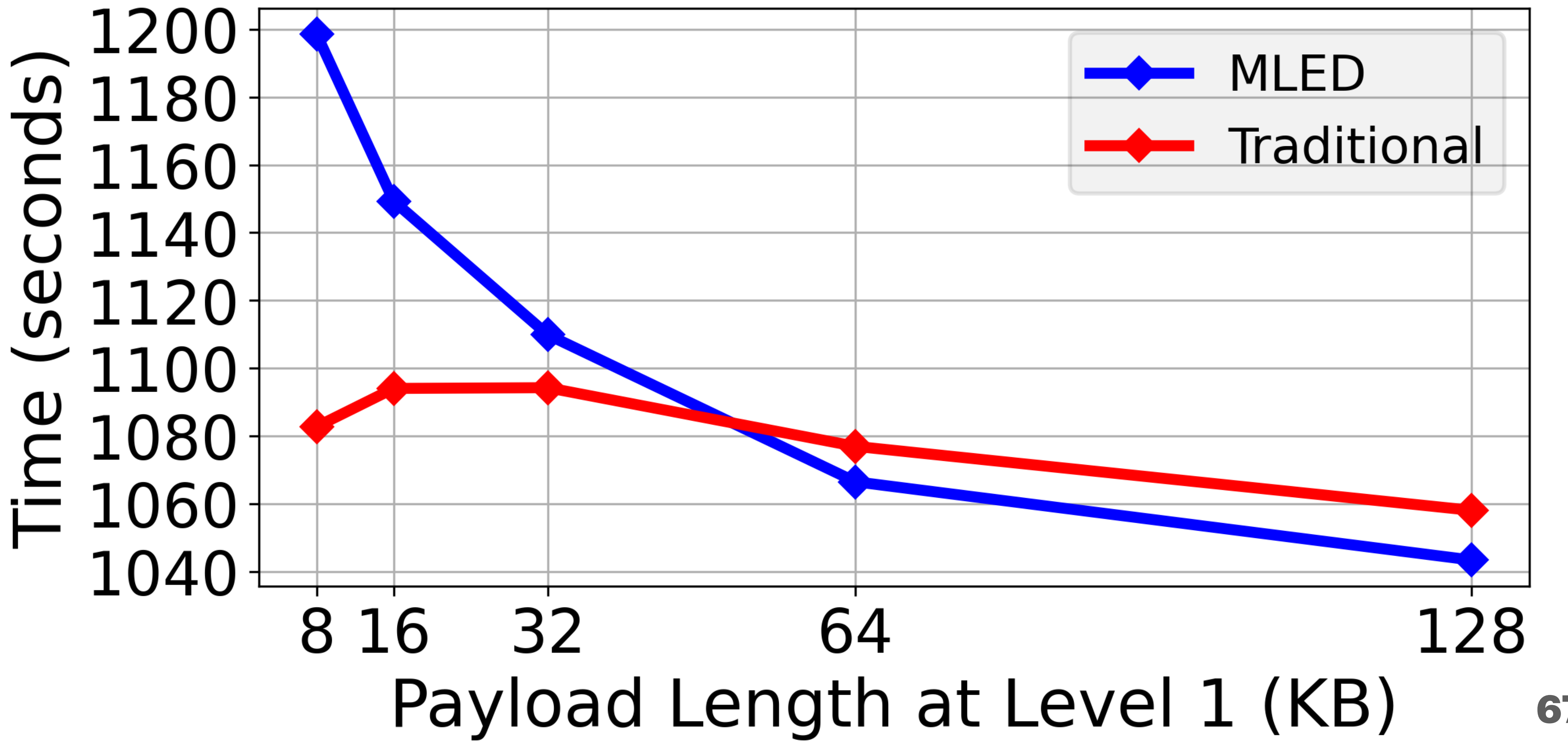


MLED Approach

PER 0.0 %



PER 0.1 %



Future Work

- Use FPGAs to accelerate CRC calculations
- Develop more layer-specific policies to fine-tune error detection
- Clean-slate P4 deployments
- Integrate non-TCP protocols for broader applicability
- Enable multi-flow data transfers for improved throughput
- Design algorithms for optimal MLED configuration

Resources

- Scan this QR Code to access resources related to MLED including this presentation.
- This work was supported in part by NSF grants CNS-2215671 and CNS-2215672.