Liquid Data Networking

John Byers, Boston University

Michael Luby, ICSI and BitRipple, Inc.

Integrating erasure codes into ICN

Erasure code benefits

- Resilience to packet loss
- Reduced latency via downloads over multiple interfaces
- Improved support for mobile clients
- Improved caching performance

Potential integration overheads

- Signaling, latency, security, response, storage

Previous work does not extract the full benefits

- Each has issues with at least some of these overheads

Our approach

- Network architecture design that provides benefits and minimizes overheads
- Seamless integration into NDN

LDN focus within ICN

Object to network data name-mapping

- Object is unit of data that is useful: video segment, image, email, file
- Packet is unit of data that is transported or cached in network
- Mapping is from object name to packet name

Request-response paradigm

- How clients and nodes request packets for an object
- How nodes respond to requests with packets

Security considerations

- End-to-end object verification
- Packet verification to prevent DoS attacks

NDN approach

object source packets

Name-mapping

D $D.0$ $D.1$ $D.2$ $D.3$ $D.4$

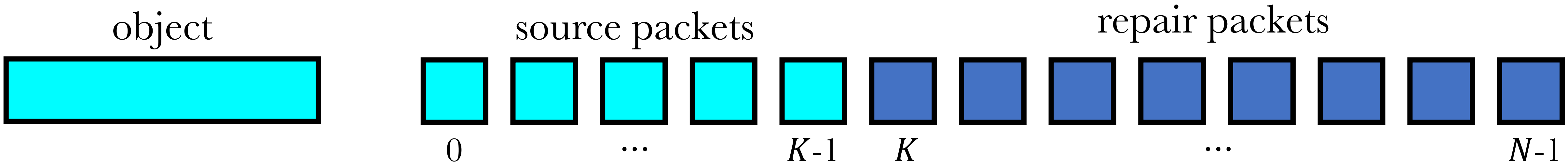
Request-response

Client explicitly requests $D.0$, $D.1$, $D.2$, $D.3$, $D.4$ to recover D

Security

Packet verification: Publishers sign packets, nodes and clients validate signed packets
Object verification: clients accept an object if all its source packets are valid

Erasure codes in ICN



M. Bilal and S.G. Kang. "Network-Coding Approach for Information-Centric Networking". In: IEEE Systems Journal. Vol. 2. 2018.

K. Lei, S. Zhong, F. Zhu, K. Xu, and H. Zhang. "A NDN IoT Content Distribution Model with Network Coding Enhanced Forwarding Strategy for 5G". In: IEEE Transactions on Industrial Informatics. 2017.

W.X. Liu, S.Z. Yu, G. Tan, and J. Cai. "Information-centric networking with built-in network coding to achieve multisource transmission at network layer". In: Computer Networks 115. 2017.

Y. Liu and S.Z. Yu. "Network coding-based multisource content delivery in Content Centric Networking". In: Journal of Network and Computer Applications 64. 2016.

K. Matsuzono, H. Asaeda, and T. Turletti. "Low latency low loss streaming using in-network coding and caching". In: Proc. of IEEE INFOCOM. 2017.

M.J. Montpetit, C.Westphal, and D. Trossen. "Network coding meets information centric networking: An architectural case for information dispersion through native network coding". In: Proc. 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design Architecture, Algorithms, and Applications. 2012, pp. 3136–3136.

G. Parisis, V. Sourlas, K. Katsaros, W. Chai, G. Pavlou, and I. Wakeman. "Efficient Content Delivery through Fountain Coding in Opportunistic Information-Centric Networks". In: Computer Communications 100 (Dec. 2016). DOI: 10.1016/j.comcom.2016.12.005.

J. Saltarin, E. Bourtsoulatze, N. Thomos, and T. Braun. "Adaptive Video Streaming With Network Coding-Enabled Named Data Networking". In: IEEE Transactions on Multimedia. Vol. 10. 2017.

J. Saltarin, E. Bourtsoulatze, N. Thomos, and T. Braun. "CA network coding approach for content-centric networks". In: Proc. of IEEE INFOCOM. 2016.

Q. Wu, Z. Li, and G. Xie. "CodingCache: multipath-aware CCN cache with network coding". In: 3rd ACM SIGCOMM Workshop on Information centric Networking. 2013, pp. 4142–4142.

G. Zhang and Z. Xu. "Combing [sic] CCN with network coding: An architectural perspective". In: Computer Networks 94. 2016.

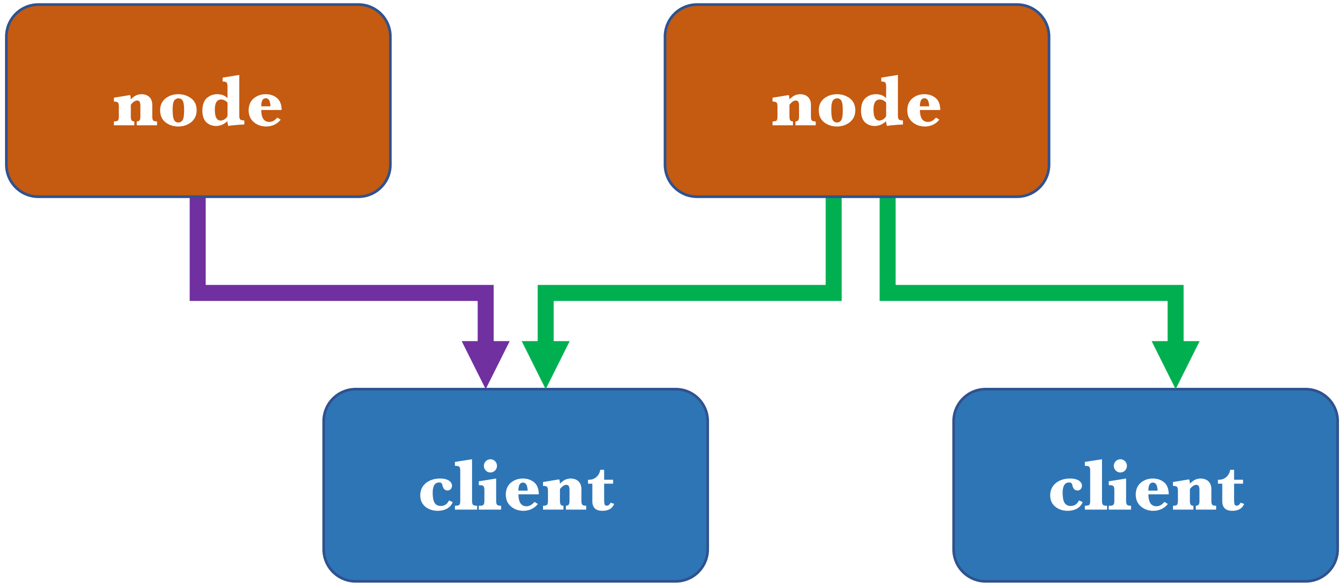
Coordination strategies

Additive response

Same client receives **different** packets from **different** nodes

Common response

Different clients receive **same** packets from **same** node



Prior request strategies

Specific data

Client asks for specific packets of data by name

How to enable common response?

Random data

Client requests an amount of data packets

Response is randomly generated encoded data

How to distinguish between common and additive response?

Useful data

Client specifies data it has already received

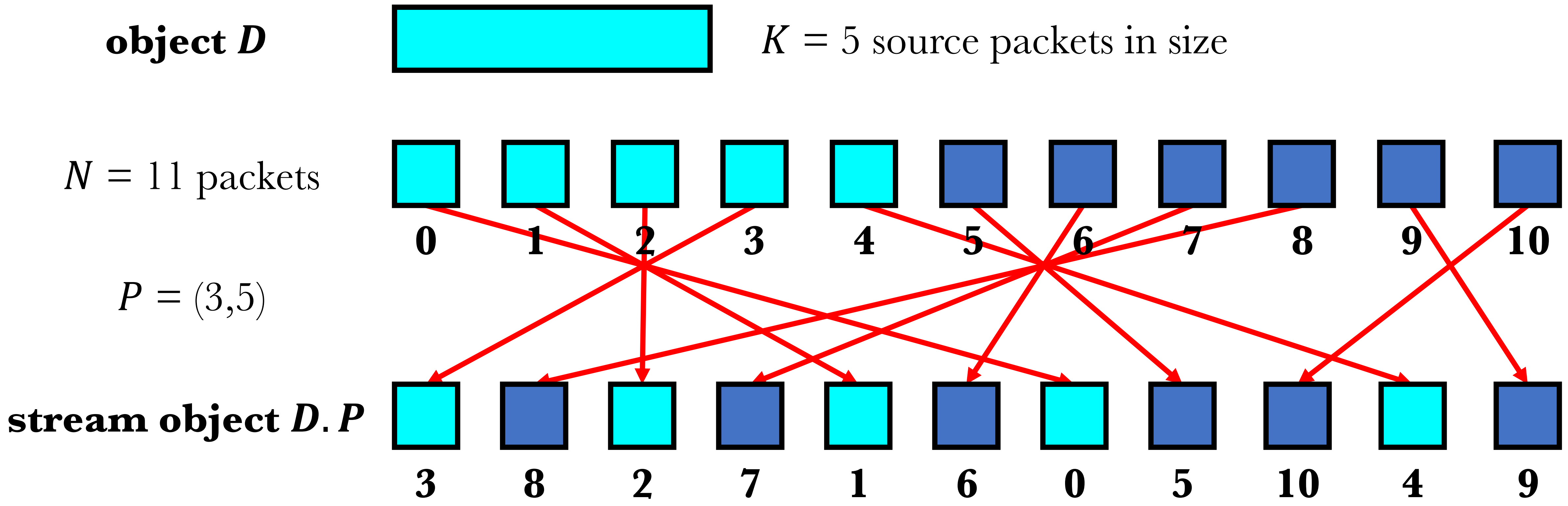
Response is additional data that will be useful

Issue with additive response

LDN enables coordination

SOPI $P = (A, B)$, where $A \epsilon \{0, \ldots, N-1\}, B \epsilon \{1, \ldots, N-1\}$

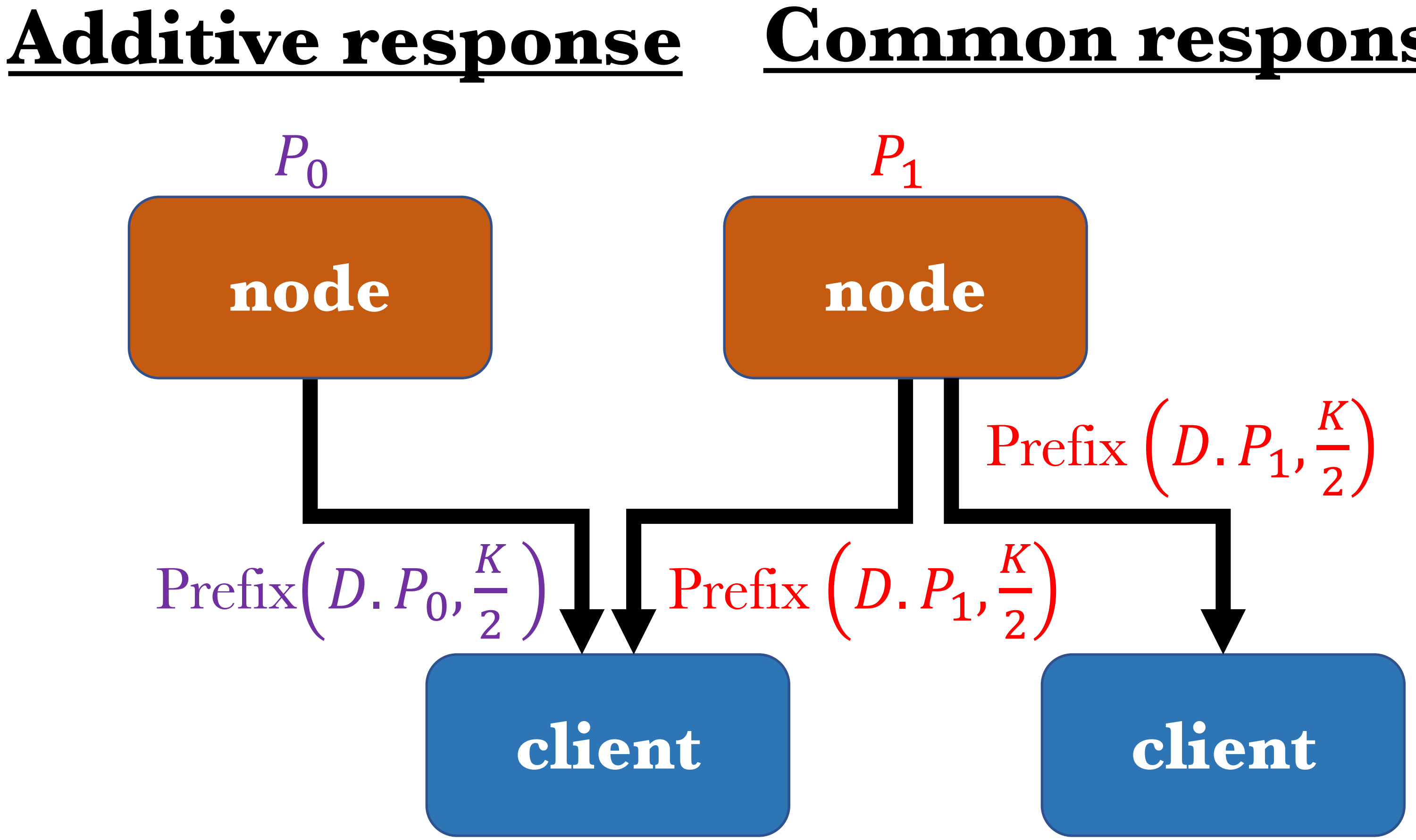
Defines permutation $\{A, A+B, A+2 \cdot B, \ldots, A+(N-1) \cdot B\}$,
where each term is modulo prime N



LDN request-response paradigm

SOPI assigned to each node

Client requests prefix of stream object associated with SOPI



LDN desirable properties

Support large K

- Small K forces splintering objects into many source blocks, causes large response overheads

Support $N > K^2$

- Minimal prefix overlap of stream objects with different SOPIs
- Can choose SOPIs randomly *
- Can deterministically design SOPIs *

Erasure code properties

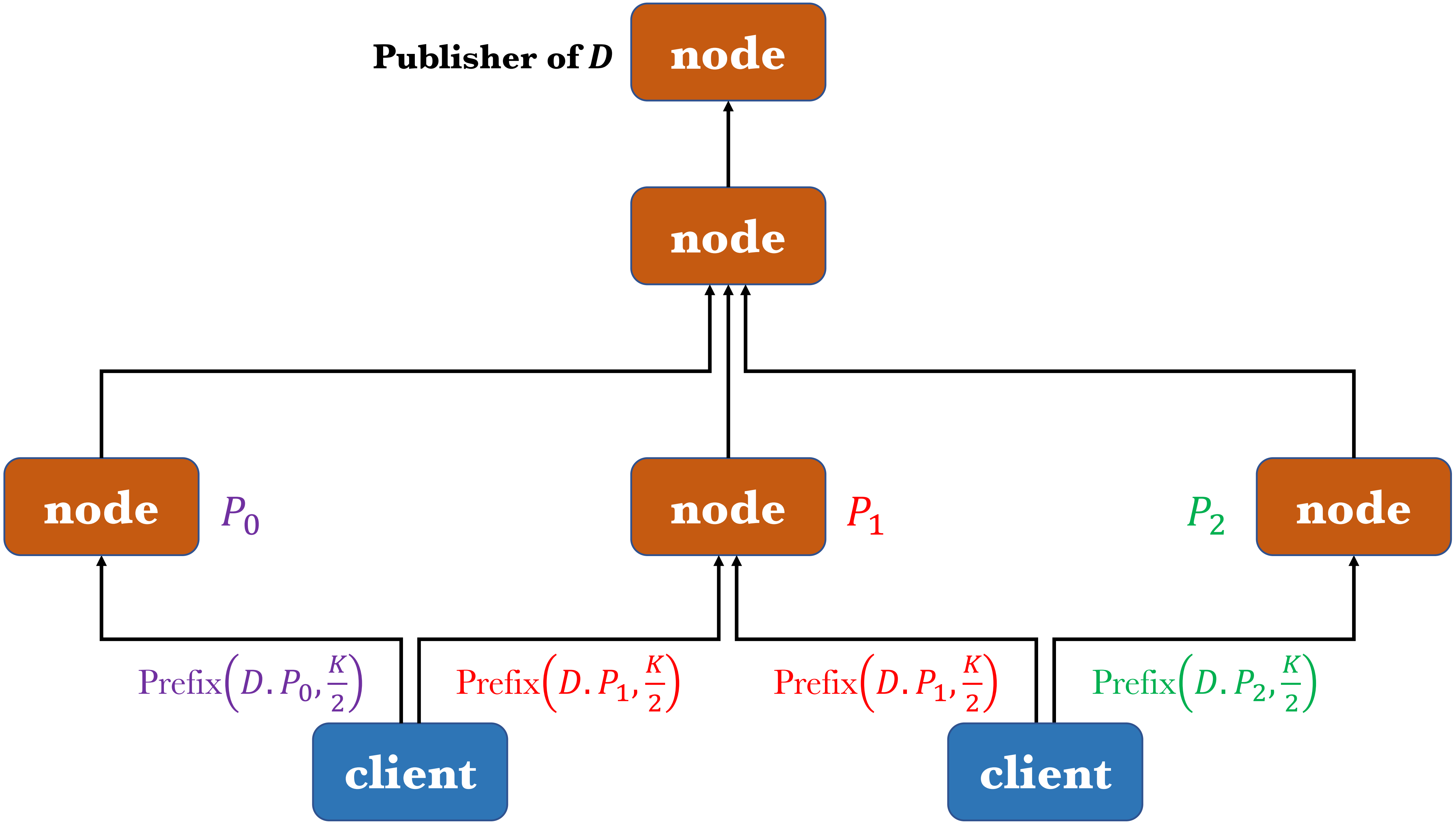
- Linear coding complexity
- Even when coding repair packets only

Erasure code choices

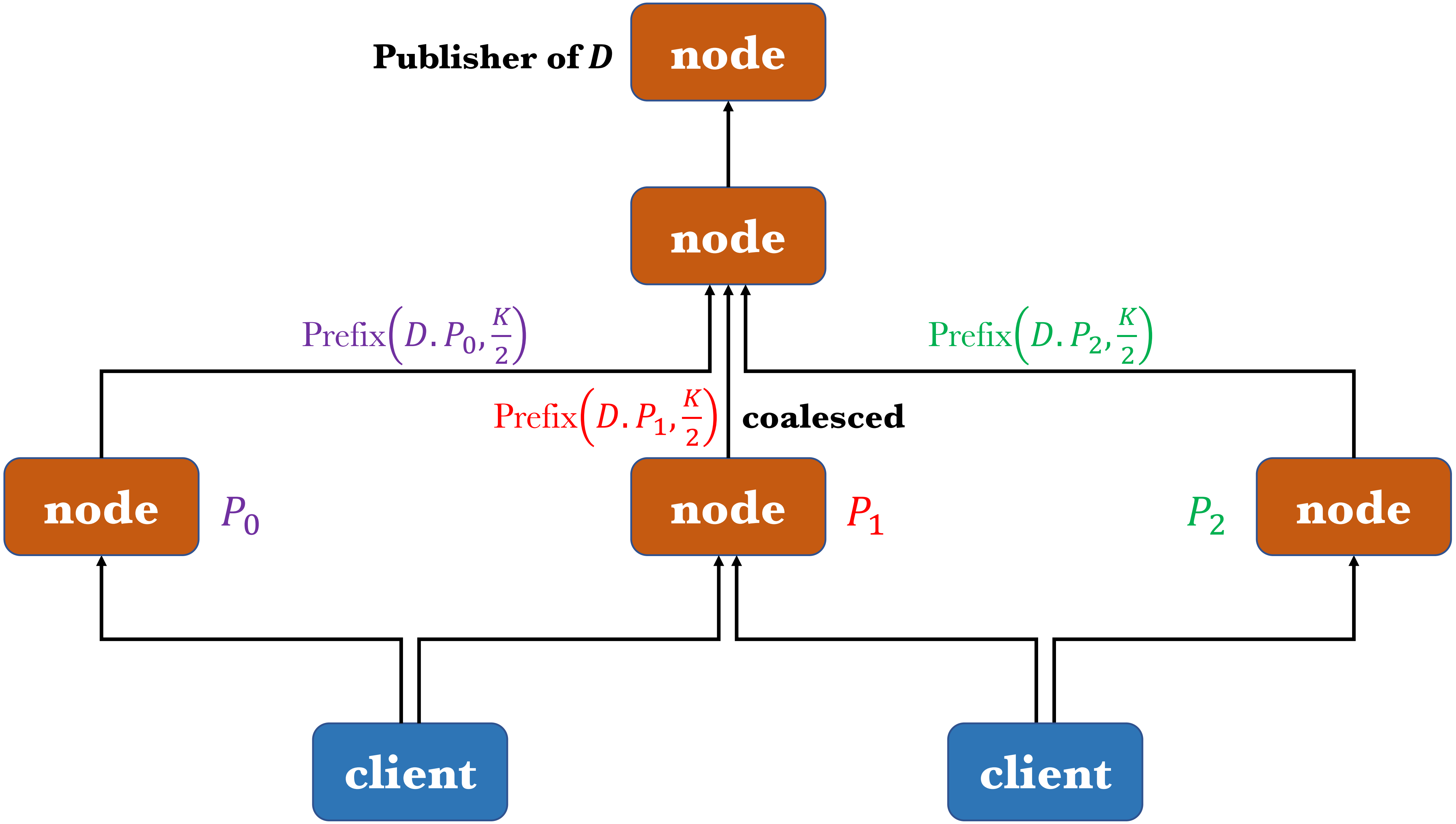
- RLNC, Reed-Solomon – significant complexity/response overhead tradeoffs
- BAT fountain code has ok tradeoffs
- RaptorQ fountain code has near optimal tradeoffs

* M. Luby. "SOPI design and analysis for LDN". In: 2020. arXiv: 2008.13300 [cs.NI].

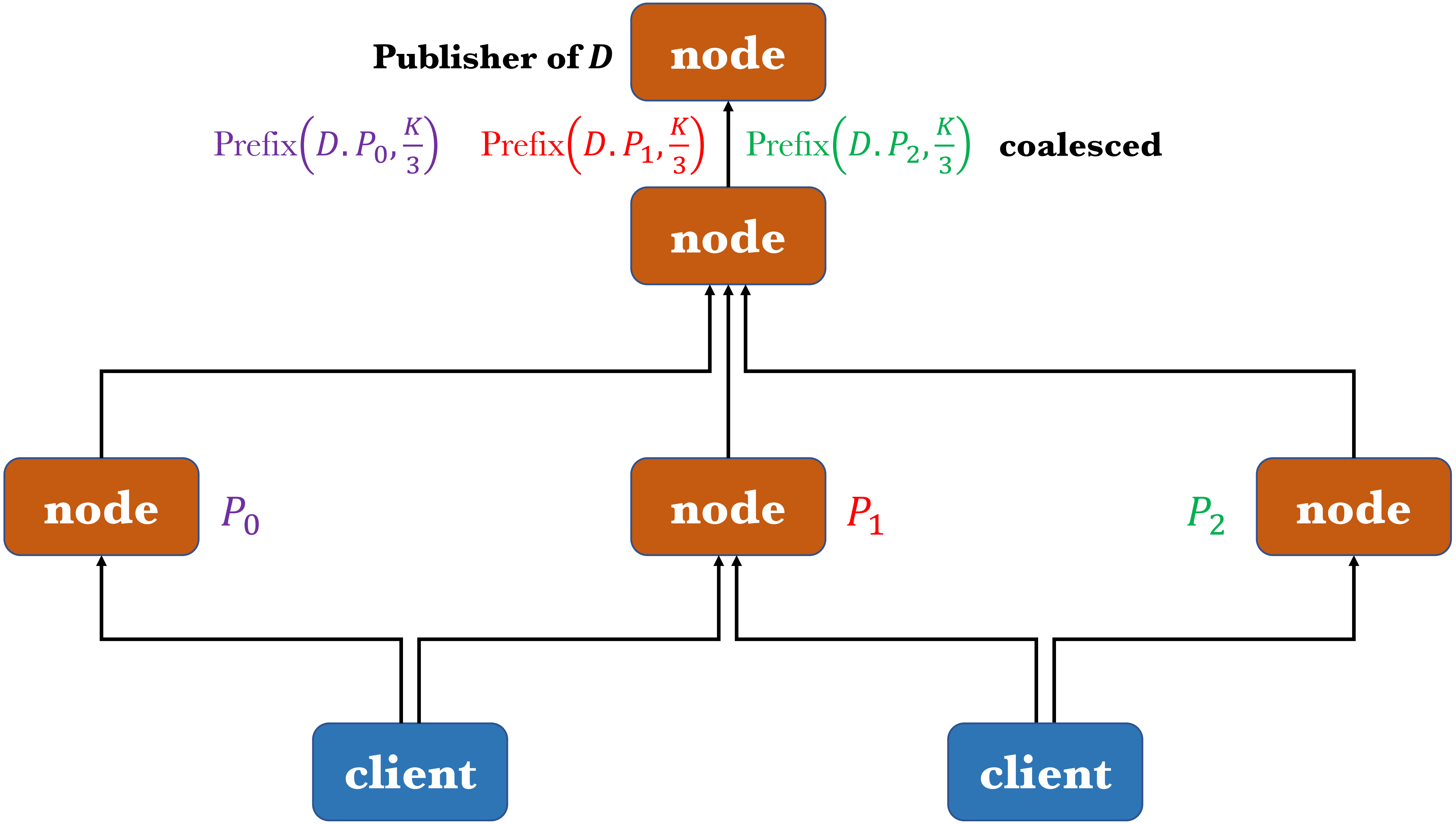
LDN turkey network example



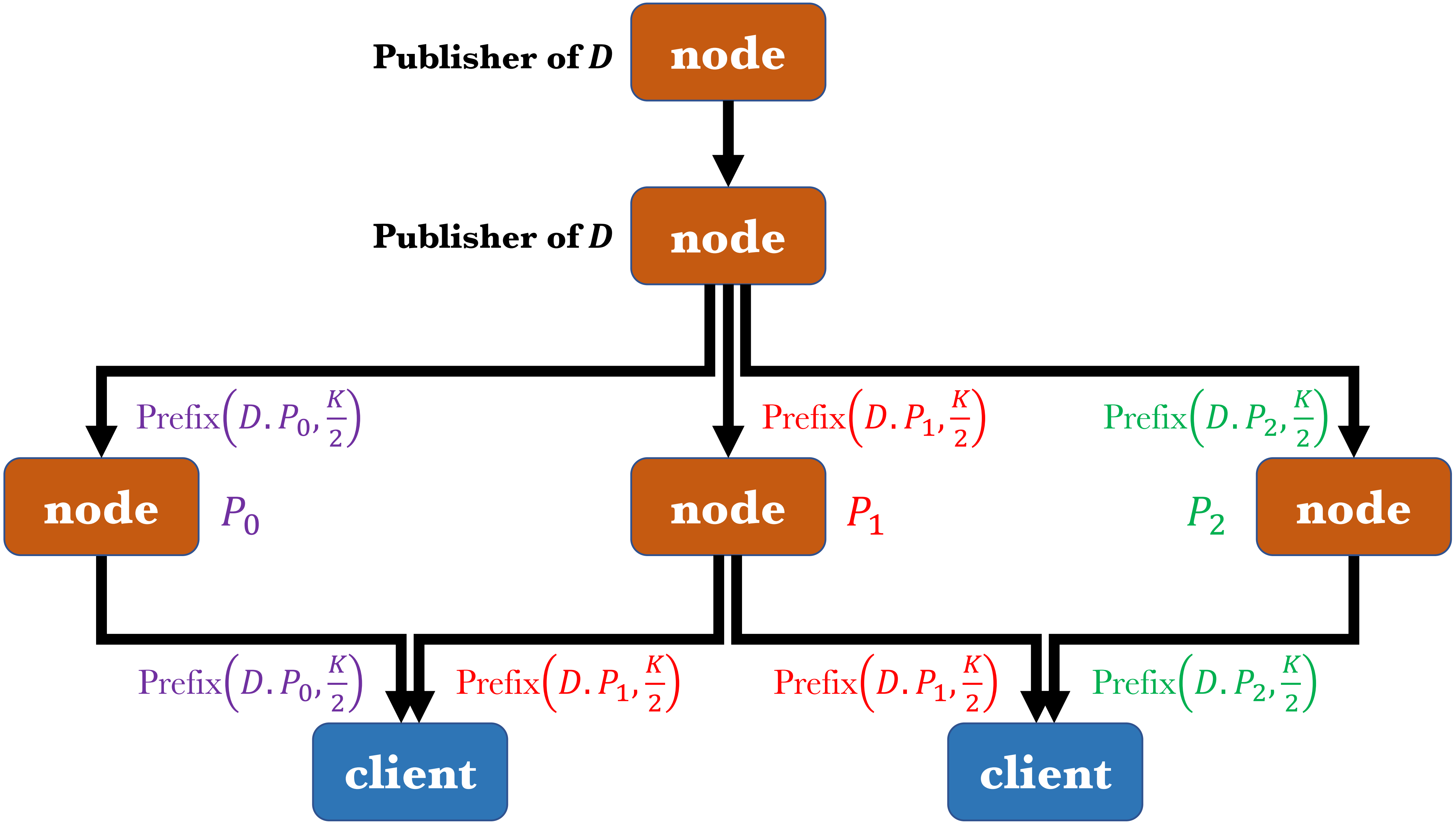
LDN turkey network example



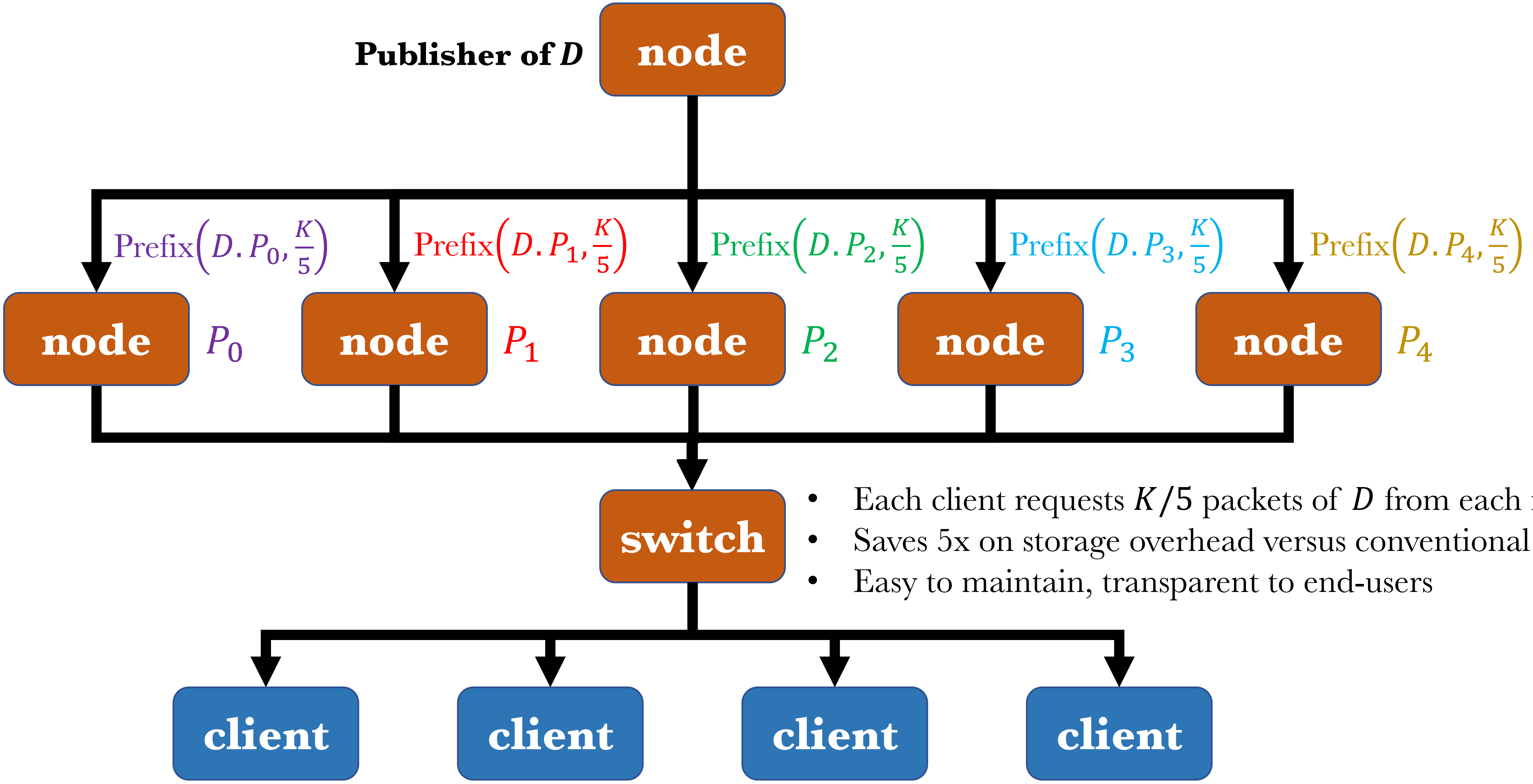
LDN turkey network example



LDN turkey network example



LDN load balancing example



- Each client requests $K/5$ packets of D from each node
- Saves 5x on storage overhead versus conventional LB
- Easy to maintain, transparent to end-users

LDN security

originator

Original generator of object

Preferable if network independent

publisher

Where object is available in network

- Node where object is injected into network
- Node that generates object from packets received

Object verification

Originators sign objects

Clients verify objects

Ensures end-to-end integrity

Packet verification

Publishers verify objects

Publishers sign packets+object creds

Nodes & clients verify packets

- Misbehaving publishers can be identified and blackballed

Summary

Clients are in control

- Clients make explicit requests – avoid response overheads

Choosing and assigning SOPIs

- Different SOPIs assigned to nodes from which same client downloads
- Small number of SOPIs overall may be possible

Security

- Distinction between originator and publisher seems important

LDN is an architectural extension of NDN

- Interest message/response protocols can be the same as in NDN
- NDN naming and other features can be directly leveraged
- Benefits of erasure codes achieved, and overheads avoided

Thanks!