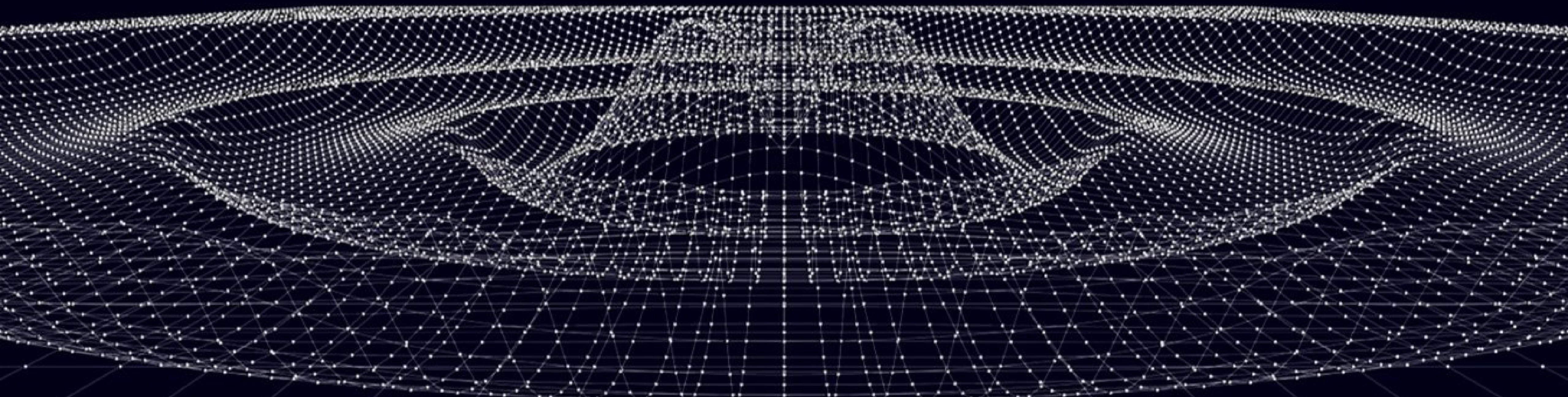
Delivering the Metaverse using Erasure Codes

Michael Luby, CEO and Cofounder
BitRipple, Inc.



The Challenge

Immersive experiences

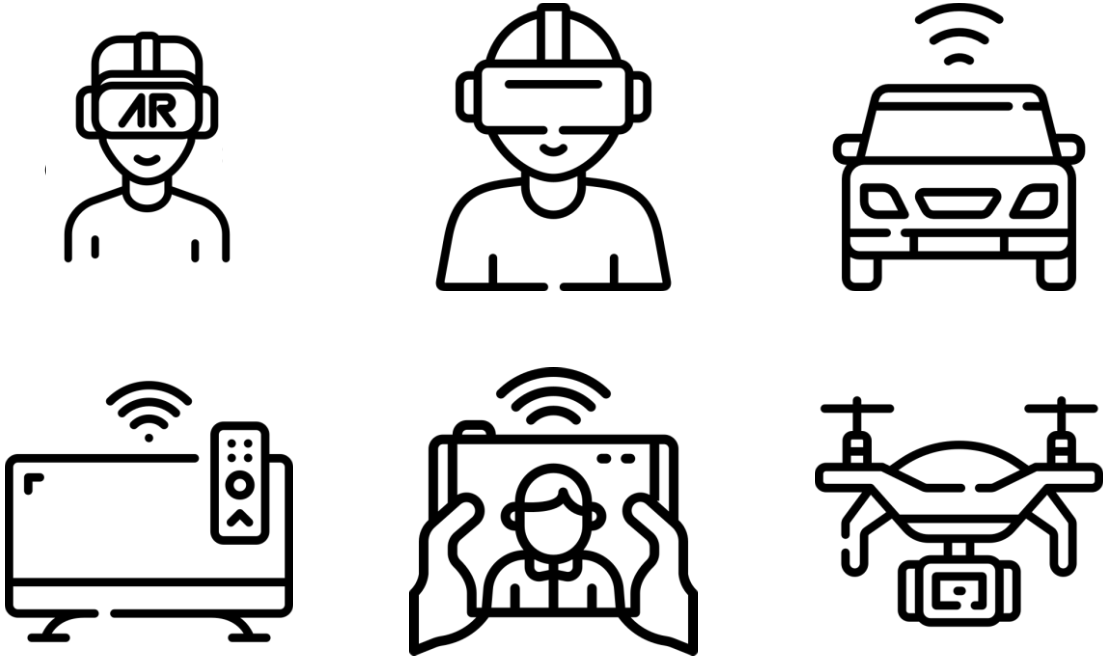
Interactive & high data volumes

Deliver lots of content with tight latencies

Data rate: 10 Mbps to beyond 1 Gbps

Latency: 1 second to less than 10 ms

Wireless



The Reality: existing content delivery solutions can't support immersive experiences + wireless

Responsiveness for Immersive Experiences*

50 ms latency – gamers frustrated

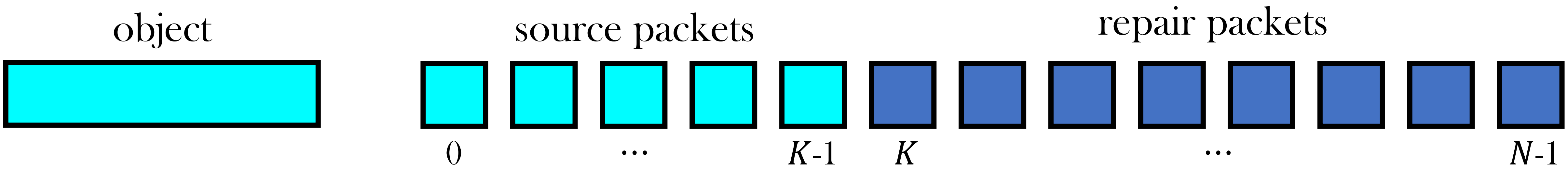
110 ms latency – non-gamers impeded

150 ms latency – unplayable

... a 10ms increase in latency reduces weekly play time by 6%

*The Metaverse Primer
Networking and the Metaverse, Jun 29, 2021, Matthew Ball & Jacob Navok

Erasure Codes for Internet delivery



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.

Integrating Erasure Codes into the Internet

Potential benefits

- Resilience to packet loss
- Consistently minimum delivery latency
- Improve support for mobile clients
- Improve delivery bandwidth using multiple access points
- Improve caching performance

Potential overheads

- Bandwidth, CPU, coding latency, signaling

Previous work

- Each has at least some issues with benefits and overheads

Erasure Code Desiderata

Support large K

- Natively support any size object
- Small K forces splintering objects into many blocks and causes inefficient bandwidth usage

Support $N > K^2$

- Support coordination
- Support for N limited to linear in K causes coordination issues

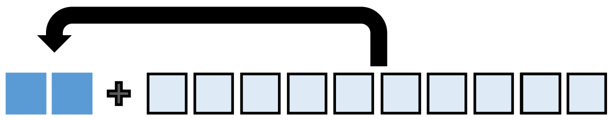
Linear complexity

- Minimize CPU impact
- Even when encoding or decoding from only repair packets
- Quadratic or larger complexity causes CPU complexity and additional latency

Erasure code choices

- RLNC and Reed-Solomon force significant compromises
- RaptorQ code is near optimal

Liquid Data



RaptorQ is used to generate liquid data from block
repair packets ( ) source packets ( )

Sender

RaptorQ is used to recover block from liquid data
missing packets (x)

Receiver

Why RaptorQ?

Any size blocks (K between 1 and 56,000)

Any amount of liquid data (N up to 2,000,000,000 for any K)

Almost instantaneous encoding/decoding

1. J. Byers, M. Luby. **Liquid Data Networking.** 7th ACM Conference on Information-Centric Networking (ICN ’20), September, 2020
2. Shokrollahi, M. Luby. **Raptor Codes.** *Foundations and Trends in Communications and Information Theory*, 2011.
3. RaptorQ Forward Error Correction Scheme for Object Delivery, **IETF RFC 6330**, August 2011
4. ATSC 3.0 standard (**NextGen TV standard**), May 2020

Liquid Data Approach

Only liquid data is sent or cached

- Can generate as much (or little) liquid data as needed from a data block
- Liquid data sent in packets thru the network from senders to receivers
- Liquid data can be cached in the network

Liquid data provides flexibility and is optimal

- Data block is recoverable from any set of liquid data equal to block size
- All liquid data packets have equal value in recovering data block

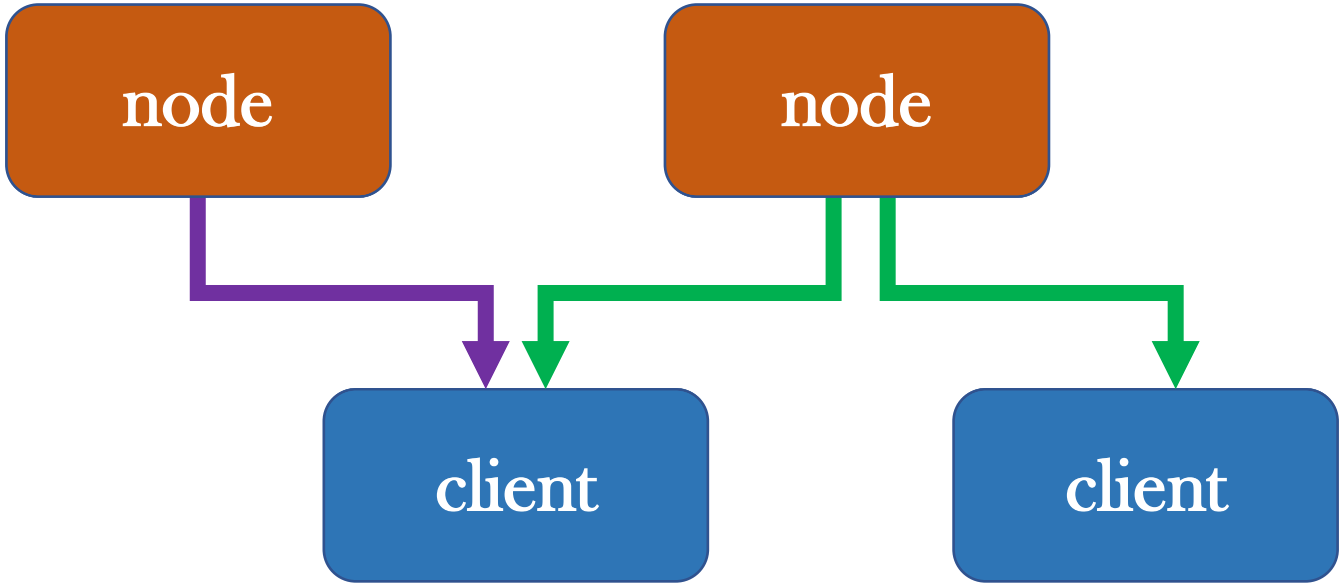
Coordination

Additive response

Same client receives
different liquid data from
different nodes

Common response

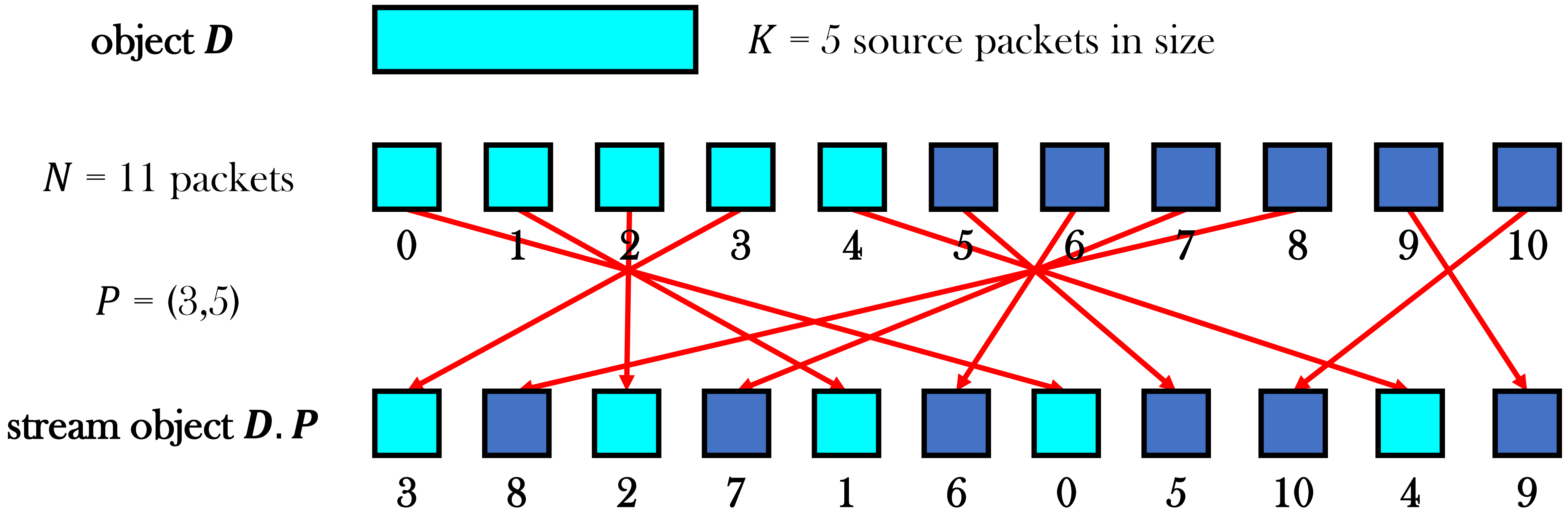
Different clients receive
same liquid data from
same node



SOPIs

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

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



Visualizing Stream Objects

All available liquid data for object D in original order ($N > K^2$)

Generate prefix of stream object $D.P_0$ for object D with respect to SOPI P_0

Generate prefix of stream object $D.P_1$ for object D with respect to SOPI P_1

Minimal prefix overlap of different stream objects

Randomly chosen SOPIs: minimal with high probability*

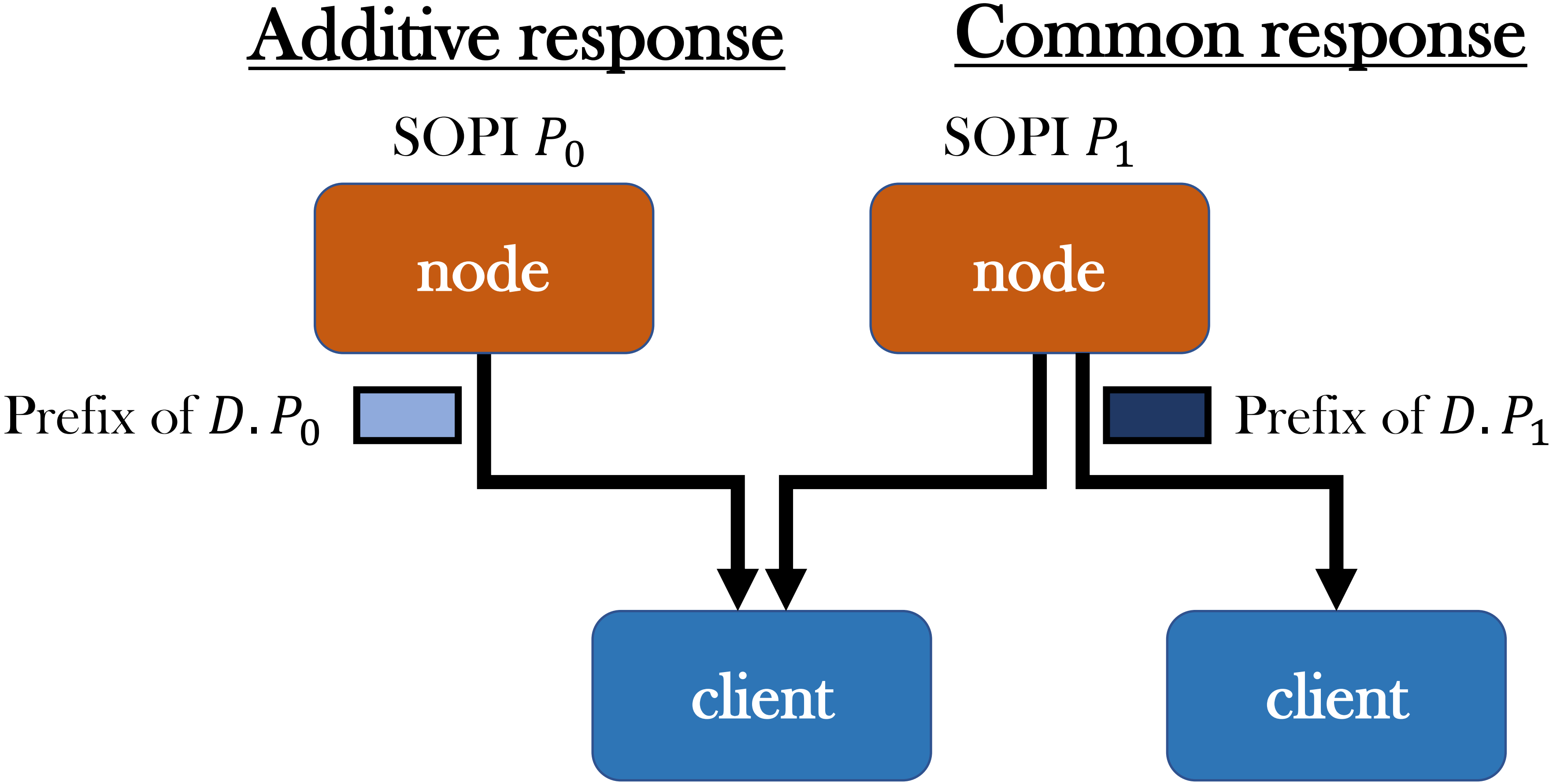
Deterministic design SOPIs: guaranteed minimal*

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

Coordination with SOPIs

SOPI assigned to each node

Client receives prefix of stream object associated with SOPI



Summary of Liquid Data Approach

Benefits

- Resilience to any amount of packet loss
- Consistently minimum delivery latency
- Support for mobile clients is seamless
- Can increase delivery bandwidth using multiple access points
- Caching performance near optimal

Minimal overheads

- Bandwidth usage is near optimal
- CPU usage is minimal
- Coding latency is minimal
- Signaling is straightforward

BitRipple

Thank You

www.bitripple.com