BROADCAST WITH NETWORK CODING: DRAGONCAST (DRAFT-ADJIH-DRAGONCAST-00)

Cédric Adjih, SongYean Cho, Emmanuel Baccelli

Robust Wireless Broadcast

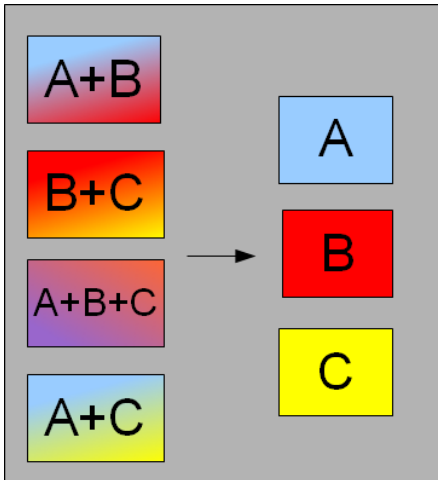
- Problem Statement:
  - **Robust Network Broadcast ?**
- Context:
  - Wireless multi-hop networks, packet networks (IP)
  - Single source broadcast
- Applicability/Requirements
  - Time-variation in topology, mobility
  - Unreliability of wireless communications

Source: (wikipedia: Felin)

Standard Broadcast Solutions

- RFC 6206 : Trickle (ROLL, sensor networks)
- RFC 6621 : Simplified Multicast Forwarding (SMF), (MANET, ad hoc networks)

- We can achieve more robustness with network coding
  - "Natural" FEC
  - Reduced signaling



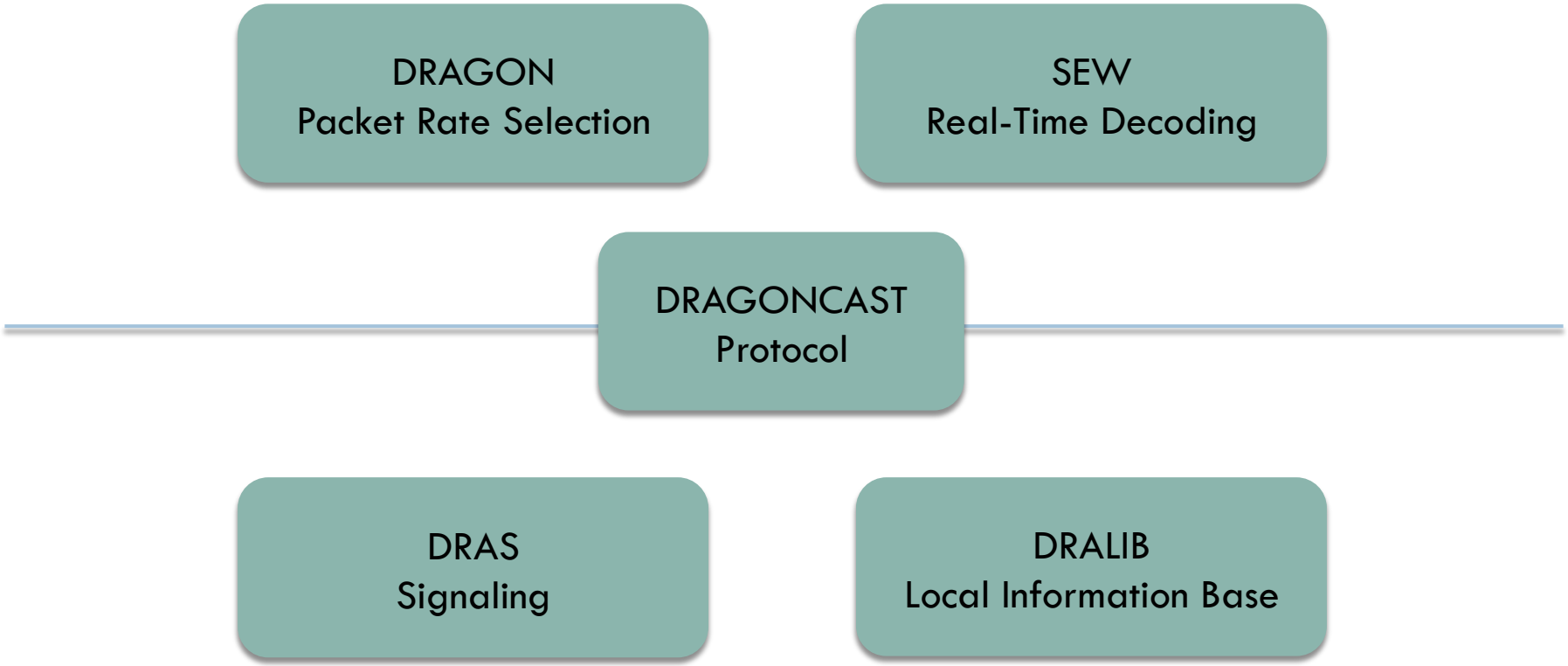
DRAGONCAST

- Single source robust broadcast leveraging network coding
- Simple, dynamic, generic and modular
  - Simple:
    - Only information of neighbor state
    - Simplified control plane (headers)
  - Dynamic:
    - Allows: topology change, transient losses, …
  - Generic:
    - No assumptions (interference, mobility, loss probability…)
    - Not specific to MAC/Phy layers

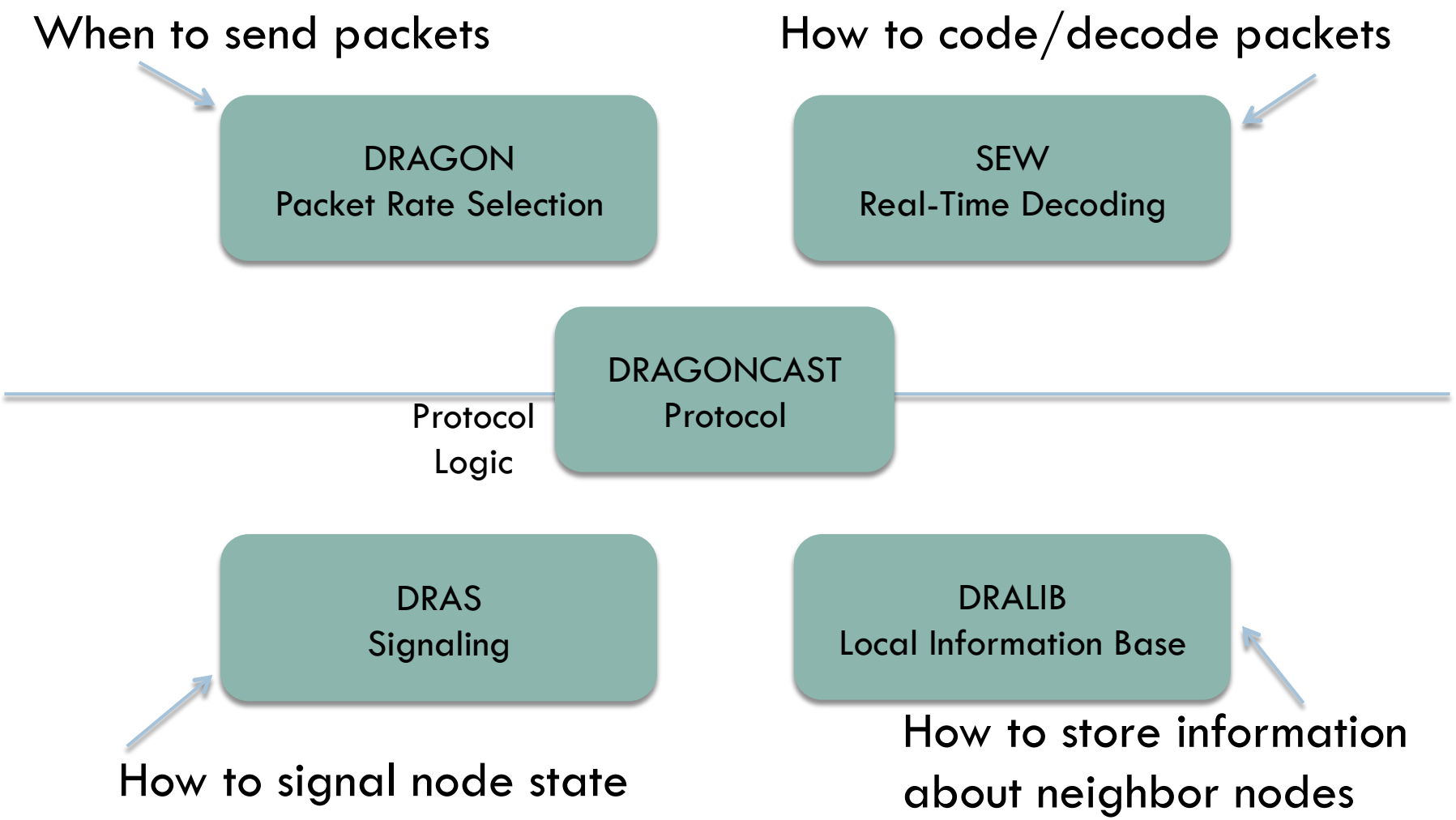
Overview of Proposed Protocol

- DRAGONCAST key aspects:
  - Wireless Network Coding, with intra-flow coding
  - Gen. Coded packets with random linear coding
  - Distributed functioning
- Primitive: wireless transmission (received by neigh.)
  - Neighborcast: ALL_MANET_ROUTERS
- Applicability:
  - Operation below capacity = congestion control unneeded

DRAGONCAST Modular Arch.

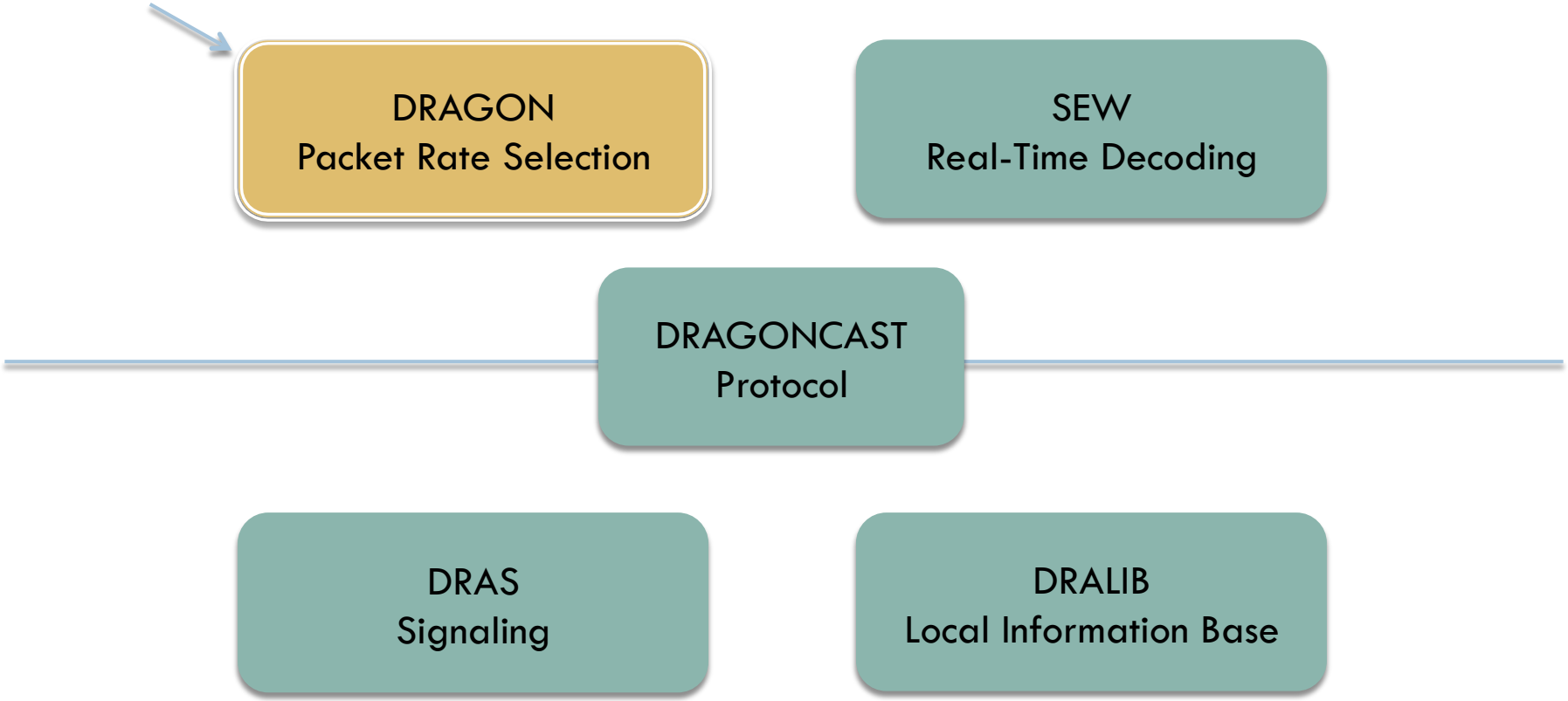


DRAGONCAST Modular Arch.



DRAGONCAST Modular Arch.

When to send packets



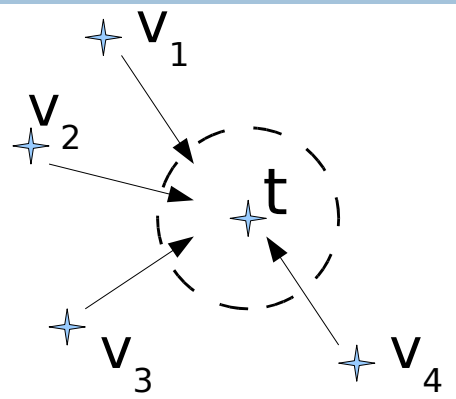
Packet Rate Policy: DRAGON

- Policy module: when to send packets?
- Principle:
  - Every node retransmits with a given “packet rate”
  - Packet rate is adjusted with a feedback control
- **Proposed** heuristic D.R.A.G.O.N. 龍:
  - Dynamic Rate Adaptation from Gap with Other Nodes
- Robust Heuristics:
  - Higher packet rate when a neighbor is “falling behind”
  - Nodes cooperate to “help” node
- Based on “Neighbor Information Set” (in DRALIB)
  - For every neighbor: **rank, number of neighbors**

[CA08a] Cho, S-Y. and C. Adjih, "Wireless Broadcast with Network Coding: Dynamic Rate Selection", Conference MedHocNet 2008, June 2008

Packet Rate Policy: DRAGON

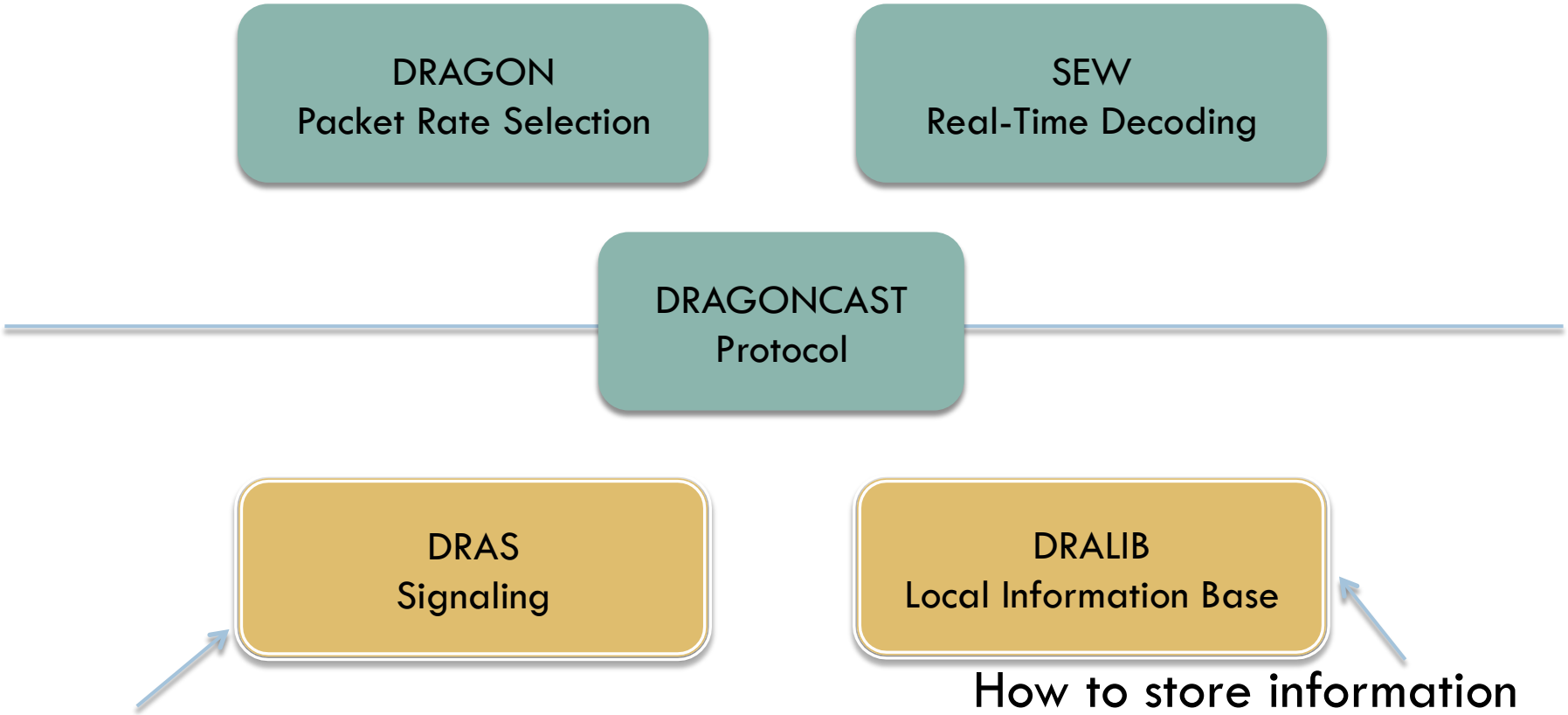
- Documented in draft-adjih-dragoncast
  - Local Received Rate
    total packet rate of from neighbors
  - Greater than: K x avg. gap
  - Gap closed in time ≈ 1 / K



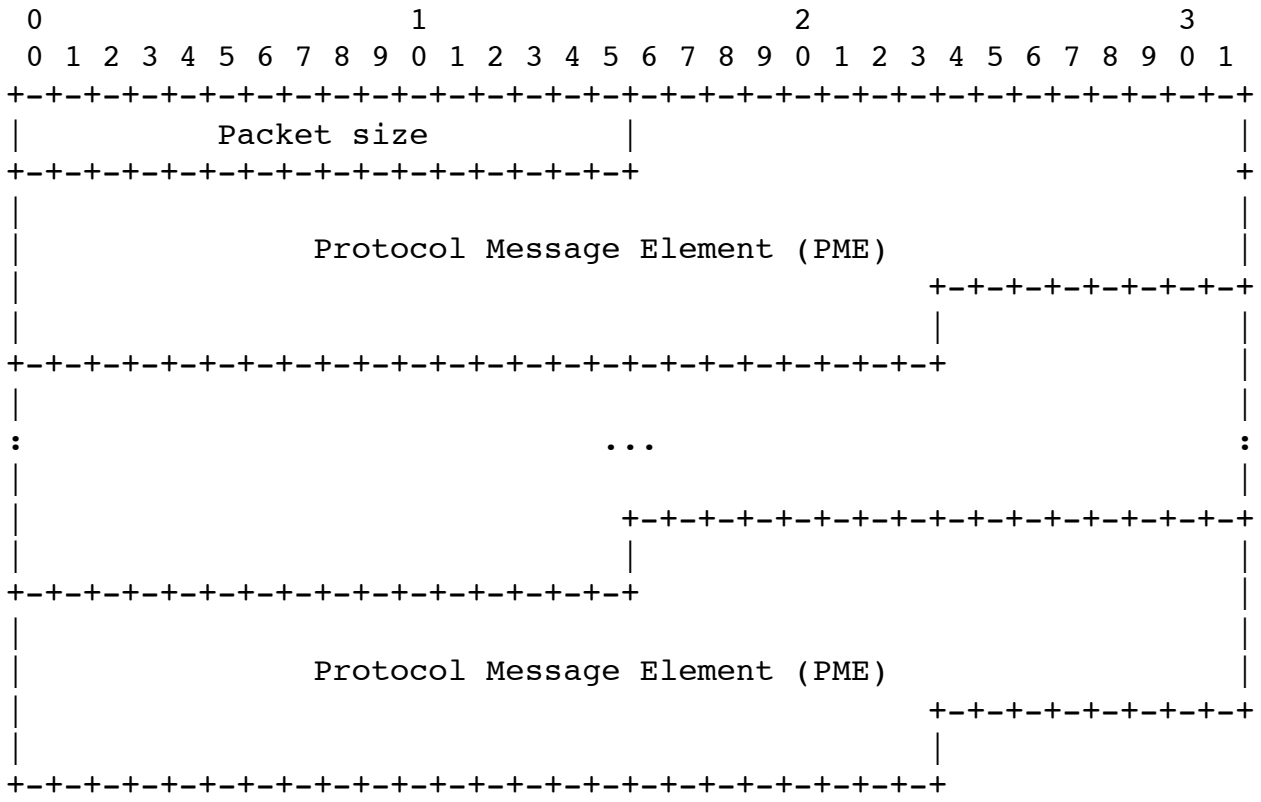
$$g_v(\tau) \triangleq \max_{u \in H_v} \frac{D_v(\tau) - D_u(\tau)}{|H_u|}$$

- **Running code + proof of convergence**
- Modular architecture allows other approaches
  - Static networks: linear program
  - Better control algorithms, back-pressure, …

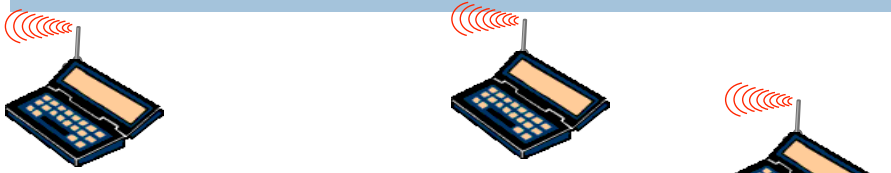
DRAGONCAST Modular Arch.



DRAS: Packet format



Protocol Overview (1/3)



Node U

State of node U (DRALIB)

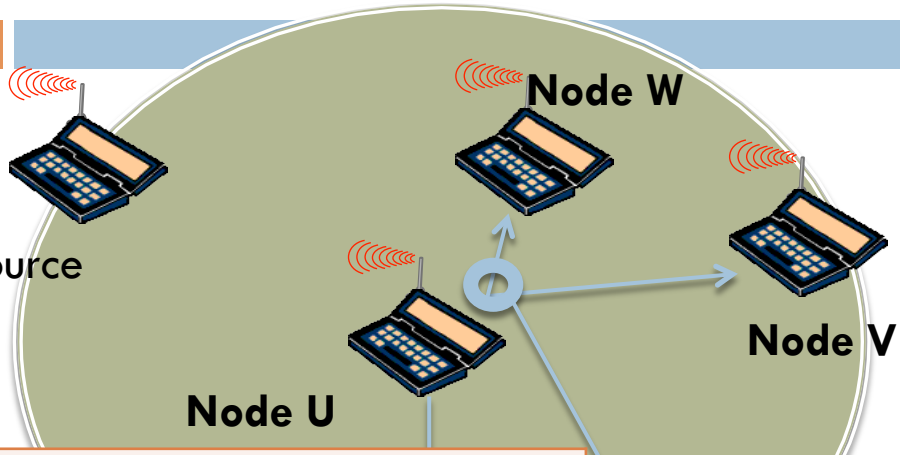
"Decoding Information Base"

- $Q_1 = P_1 + 2P_2 + 4P_3$
- $Q_2 = P_1 + 2P_2 + P_3$
- $Q_3 = P_2 + P_3$

(rank=3)

[…]

Protocol Overview (2/3)

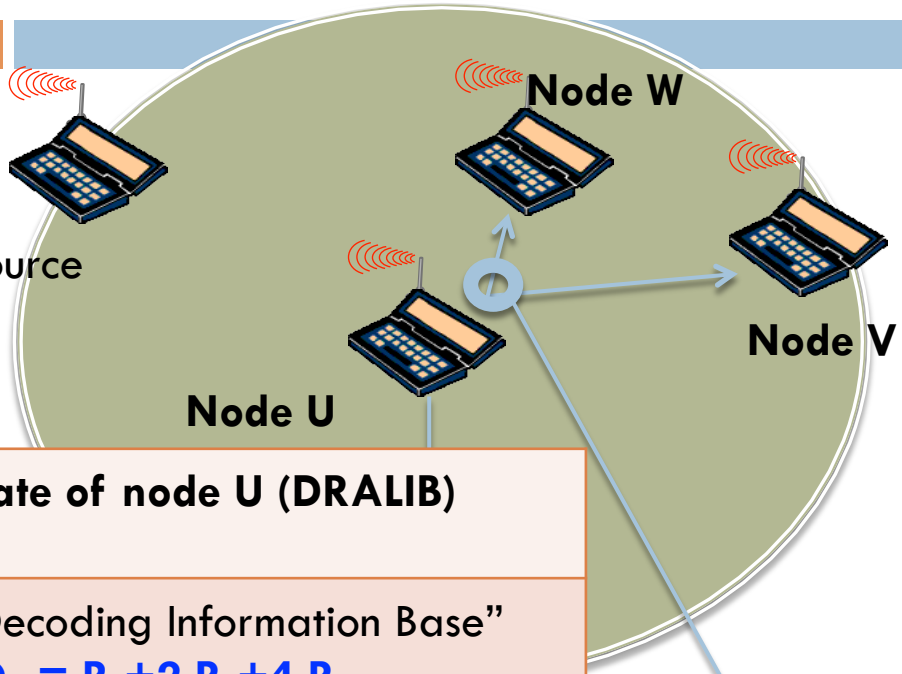


State of node U (DRALIB)
"Decoding Information Base" •$Q_1 = P_1+2\ P_2+4\ P_3$ •$Q_2 = P_1+2\ P_2+P_3$ •$Q_3 = P_2+P_3$ (rank=3)
[…]

(RLC: Q_1+Q_3)

Coded Packet (w/ DRAS)	
Rank=3 | Known neigh.=2	← Header for DRAGON
Encoding vector=(1,3,5,0,…)	
Content = [$P_1+3\ P_2+5\ P_4$]	

Protocol Overview (3/3)



State of node V (DRALIB)

"Decoding Information Base"
[…]
•$Q_k = P_1+3\ P_2+5\ P_4$

Neighbor Information Set
[…]

Node U:
rank = 3, #neigh =2

State of node U (DRALIB)

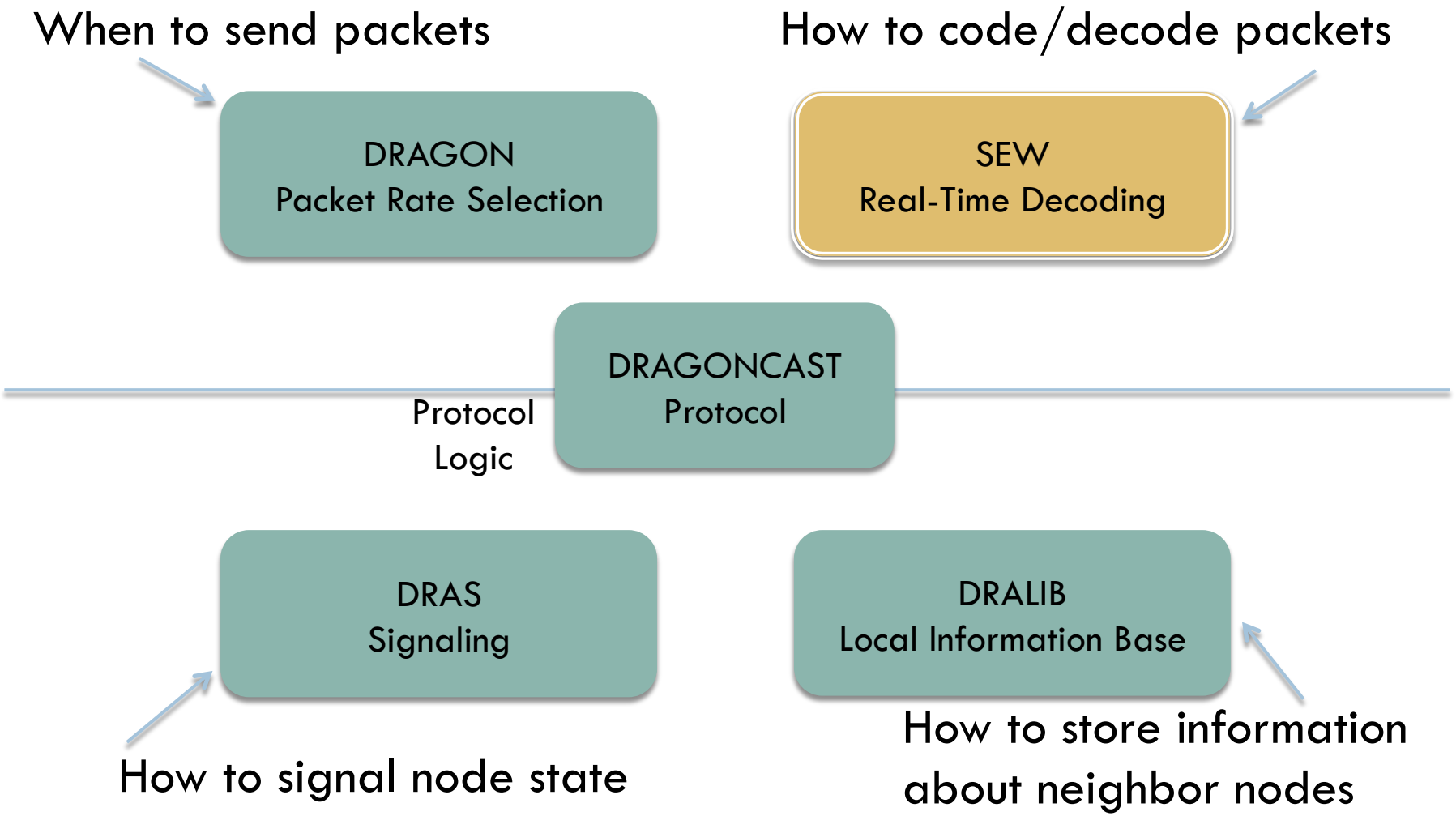
"Decoding Information Base"
•$Q_1 = P_1+2\ P_2+4\ P_3$
•$Q_2 = P_1+2\ P_2+P_3$
•$Q_3 = P_2+P_3$
(rank=3)

[…]

(RLC: Q_1+Q_3)

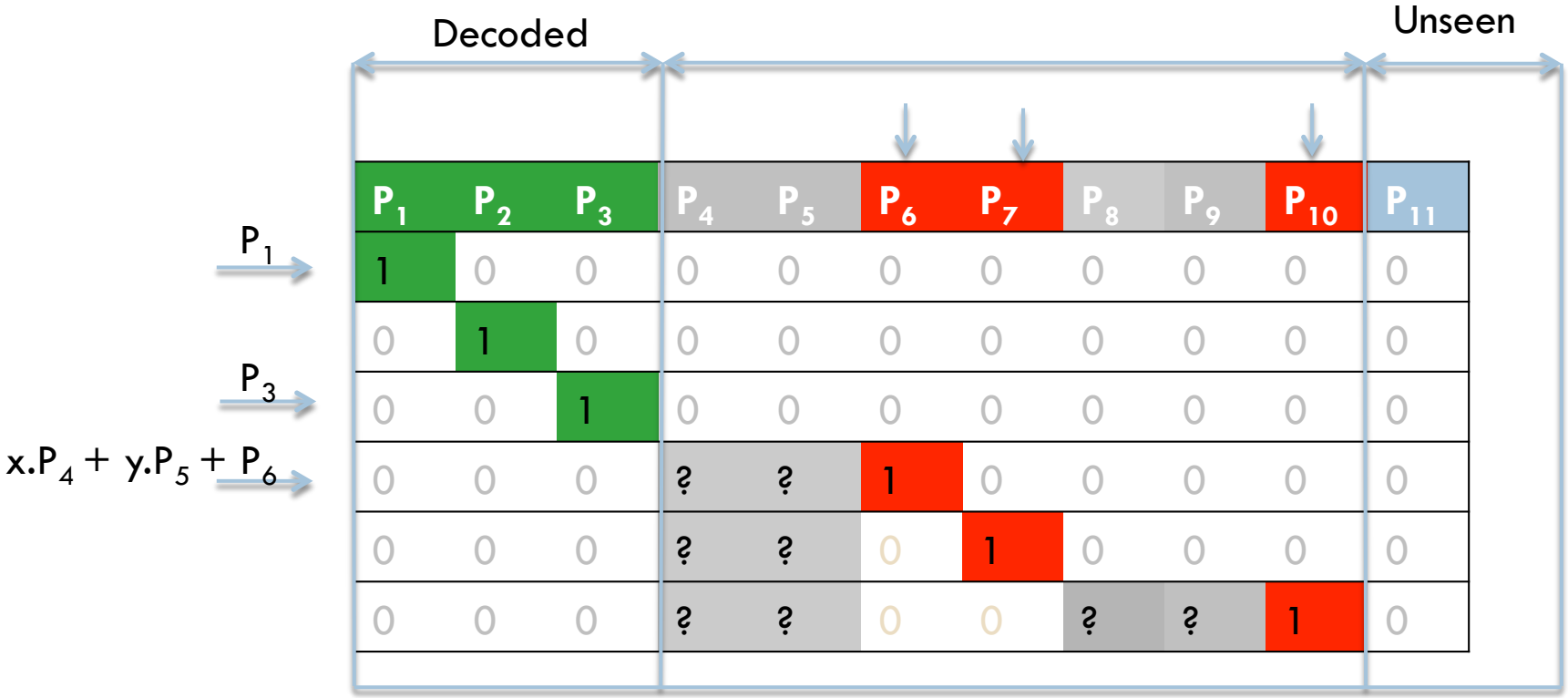
Coded Packet (w/ DRAS)
Rank=3 | Known neigh.=2
Encoding vector=(1,3,5,0,…)
Content = [$P_1+3\ P_2+5\ P_4$]

DRAGONCAST Modular Arch.



SEW: Sliding Encoding Window

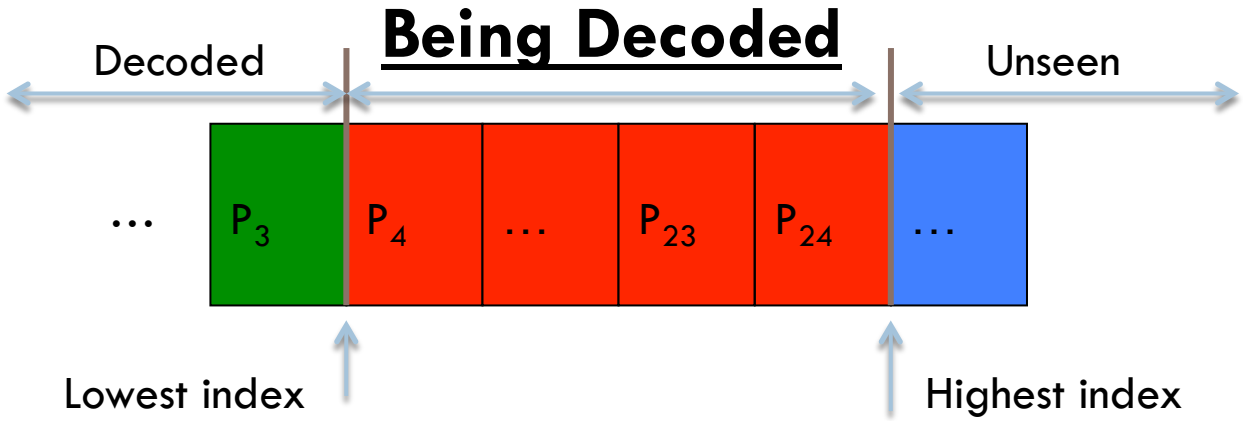
- Principle: “real-time” robust decoding
- Variant of Gauss-Jordan elimination in packet set



[CA08b] Cho, S-Y. and C. Adjih, "Wireless Broadcast with Network Coding: DRAGONCAST", Inria RR-6569, July 2008.

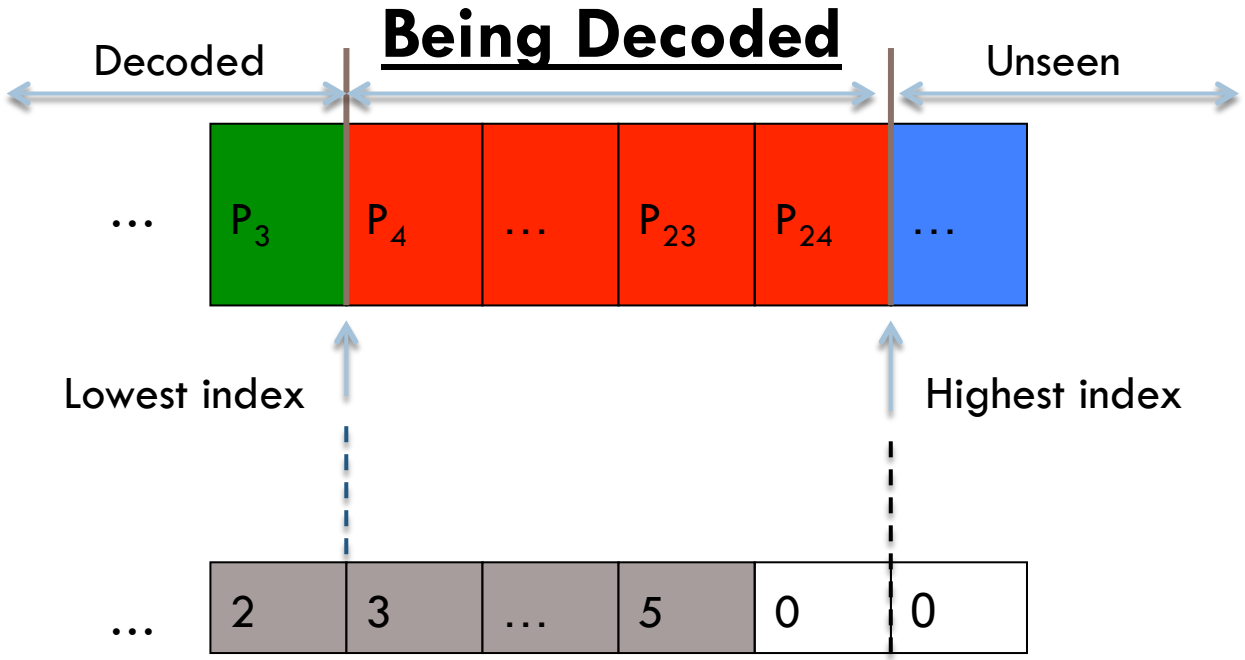
SEW: Sliding Encoding Window

- Macroscopic view:



SEW: Sliding Encoding Window

- Macroscopic view:



Any linear combination of packets with indices < Highest index will help decoding

SEW: Real-Time Decoding

- Idea of SEW:
  - Add a header with “lowest index”, “highest index”
  - -> Neighbor Information Set
- Distributed cooperation:
  - Only send coded packets that will help the most lagging neighbor
  - If L = “lowest index of neighbor”
    - Send packets with indices [L, L+1, L+2, …. L+W]
- Window size W, tradeoff:
  - Efficiency –vs- decoding delay

Conclusion

- Draft in -00 state
  - Generic modular architecture defined
  - Some TBDs within modules
- Interest of the RG for:
  - This type of problem statement/application ?
  - This type of architecture ?
- Ready for input of wg
  - On different modules
  - On the architecture

THANK YOU

WNC Broadcast

- Basic broadcast protocol with random linear coding:
  - Sources generates P_1, P_2, P_3, …. with fixed rate
  - Every retransmission in the network is a linear combination of source packets:
    - $Q = a_1 P_1 + a_2 P_2 + \ldots + a_k P_k$
  - Every node keeps a set of coded packets
  - Every node regularily retransmits packets with a given rate:
    - Create random linear combination
    - Send coded packets, with associated combination
  - Decoding: linear algebra