DISTRIBUTED WIRELESS BROADCAST PROTOCOLS WITH NETWORK CODING FOR SINGLE/MULTIPLE SOURCES

IETF/IRTF 92
DALLAS

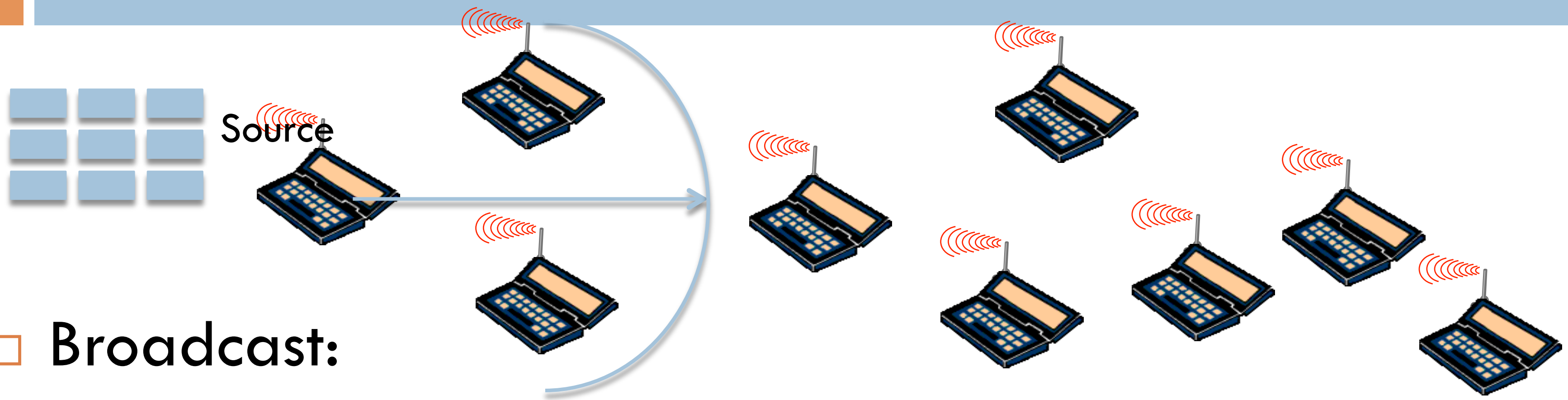
Cedric Adjih, Ichrak Amdouni, Oliver Hahm (Inria)
Claudio Greco, Michel Kieffer (CentraleSupelec)

IPR Statement

- No IPR from our side
- From IPR disclosures at IETF, look for:

IETF IPR Disclosure	Patent	This presentation
ID #2183	"Randomized distributed network coding", US 7706365	Random linear [re]coding (slide 4, and following)

Wireless Multihop Broadcast



- Broadcast:
  - Single source: many packets to entire network
  - Multiple sources: one packet to entire network
- Use cases in Wireless Sensor Networks:
  - "OTA" (over-the-air programming)
  - Data collection with "unmanaged" network
- Without NC: SMF (RFC 6221), Trickle (RFC 6206)

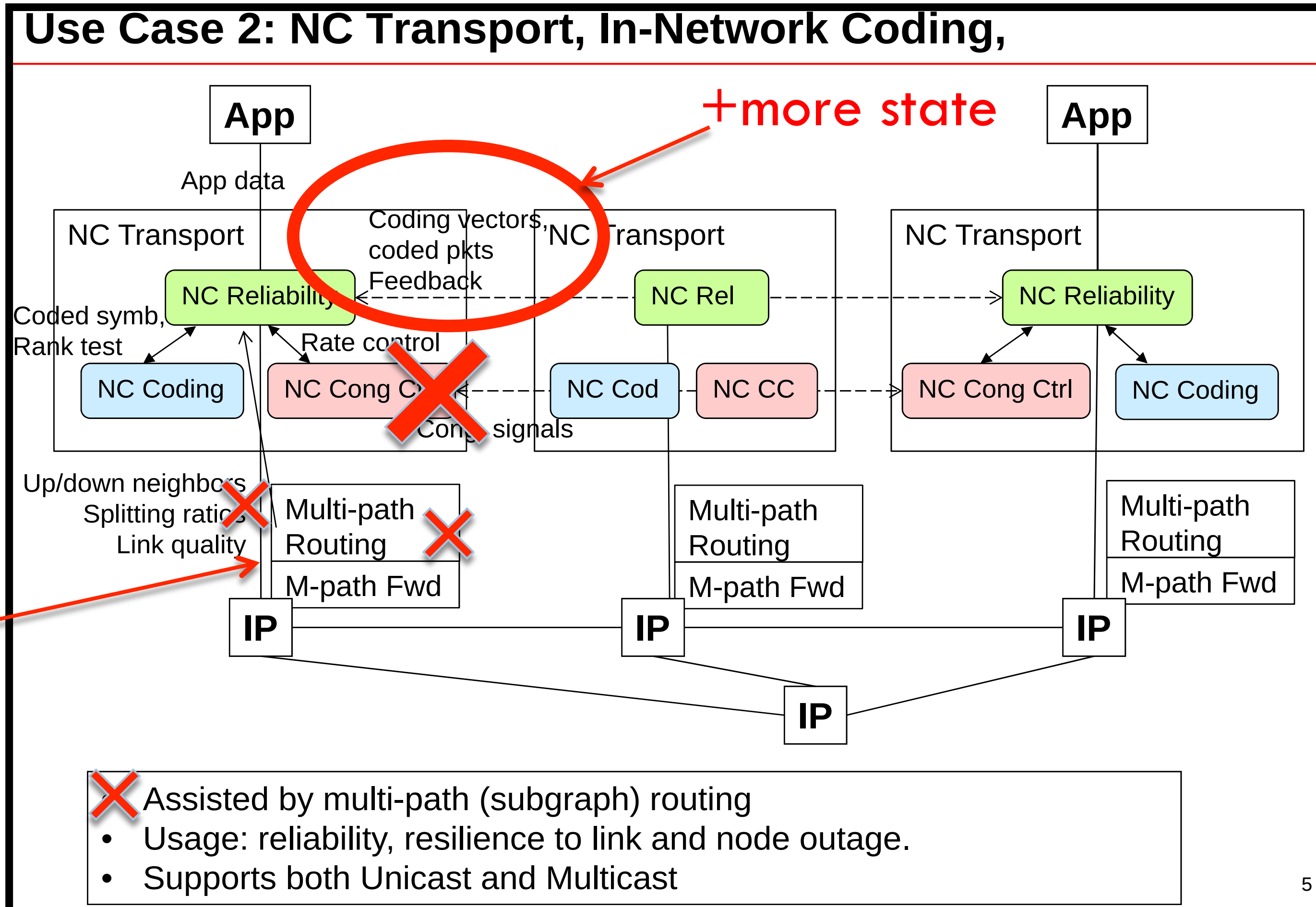
Fully Distributed Protocols

- Fully Distributed Broadcast Protocol:
  - No knowledge of the entire network (sources/dest.)
- Ex: protocol DRAGONCAST/DragonNet
  - I. Amdouni, C. Adjih, and T. Plesse “Network Coding in Military Wireless Ad Hoc and Sensor Networks: Experimentation with DragonNet”, accepted at ICMCIS 2015
  - S-Y. Cho and C. Adjih, "Wireless Broadcast with Network Coding: DRAGONCAST", Inria RR-6569, July 2008
  - Every node retransmits coded payloads at a given packet rate per second (e.g. with random linear coding)
  - Coded payloads are maintained in a (decoding) set
  - Control plane: state piggybacked on coded payloads
    - Decoded payloads, number of neighbors, ex: …
- This presentation: sliding encoding window (single source), encoding vectors (multiple source)

Corresponding use cases

Simple "neighborcast"

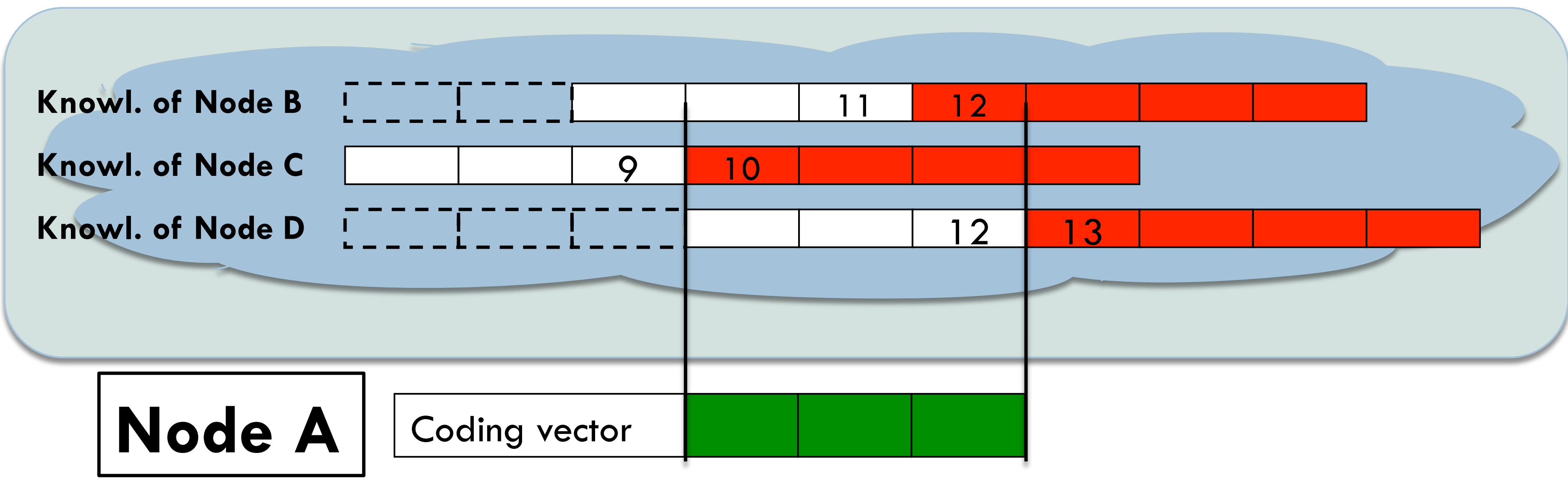
+more state



From http://www.ietf.org/proceedings/91/slides/slides-91-nwcrg-0.pdf

Single source: Sliding Window

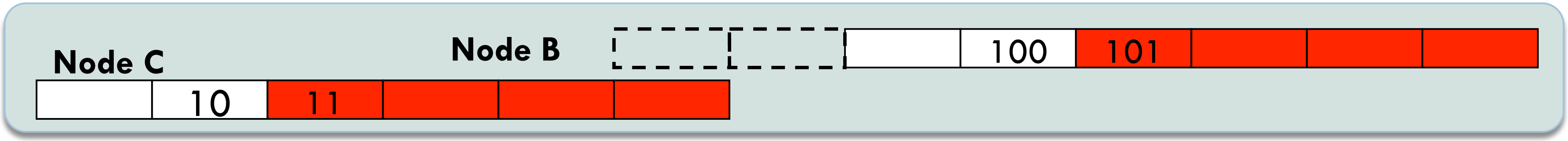
- Sliding Encoding Window (SEW in DRAGONCAST):
  - Each node transmits decoding state: “first undecoded”
  - Node generates packets considering neighbor state



  - Simple functionning

Single source: CISEW

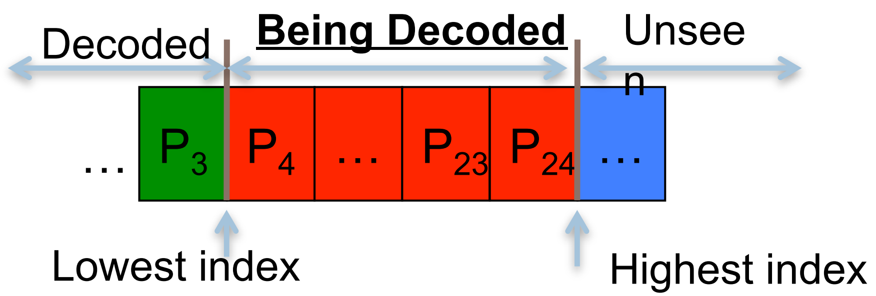
- Coding Interval-based Sliding Encoding Window
  - I.Amdouni, C. Adjih, « Coding Interval-based Sliding Encoding Window », draft-amdouni-nwcrg-cisew-00 (work in progress), July 2014, http: //tools.ietf.org/html/draft-amdouni-nwcrg-cisew-00
- Redesign of SEW, aware of:
  - Heterogeneous decoding rate at nodes



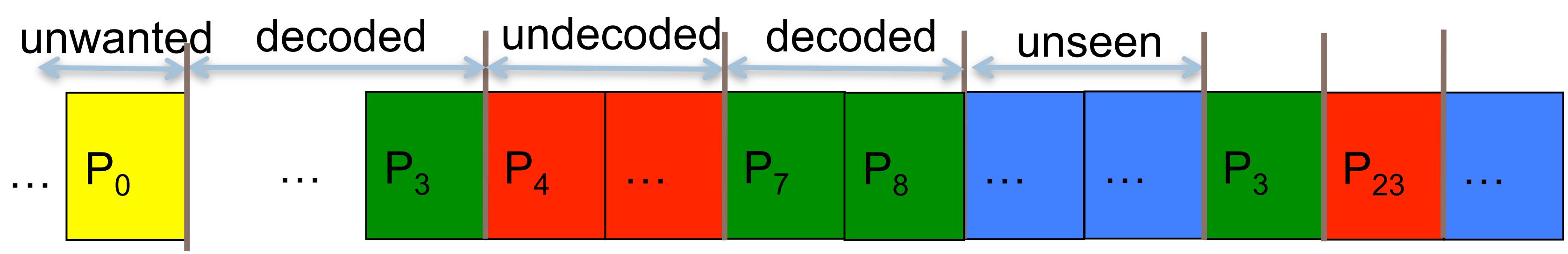
    - Introduce « losses »
  - Limited buffer size (overflow)
    - Choice between throwing decoded or undecoded packets
    - Combinations may become useless: P_{11}+… if P_{11} dropped
- Encoding strategy:
  - Fit **at best** neighbors needs in terms of payloads
  - Needs more information about the state of neighbors
    - state advertizement from signaling

CISEW: Finer Signaling

- SEW signaling = state:

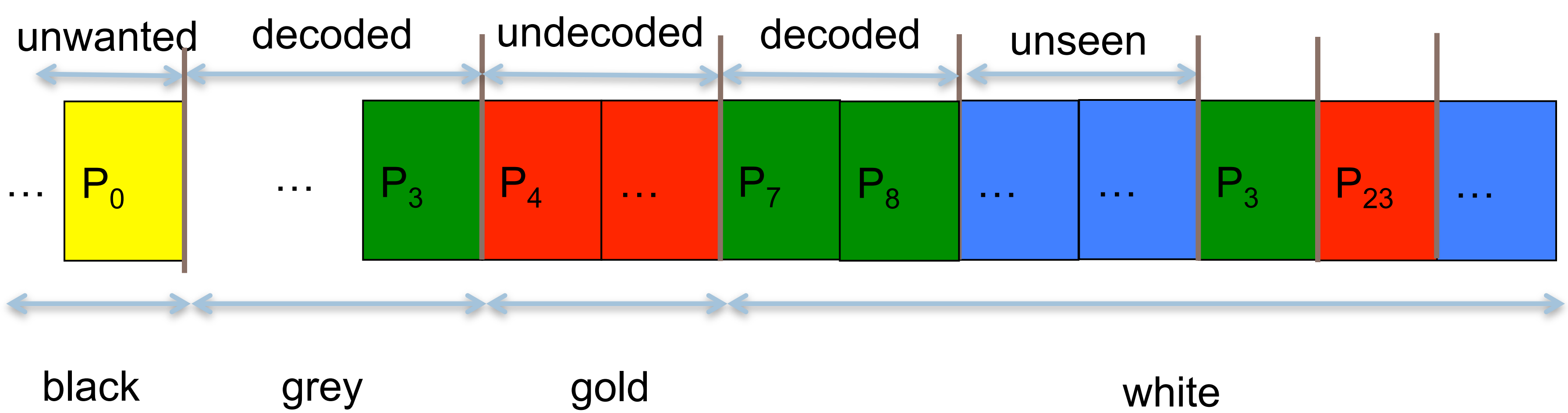


- CISEW state: each original payload is one of:
  - Decoded (or « lost ») but no longer available
  - Decoded and available
  - Not yet decoded but received in one/some linear combinations
  - Not yet decoded but never received

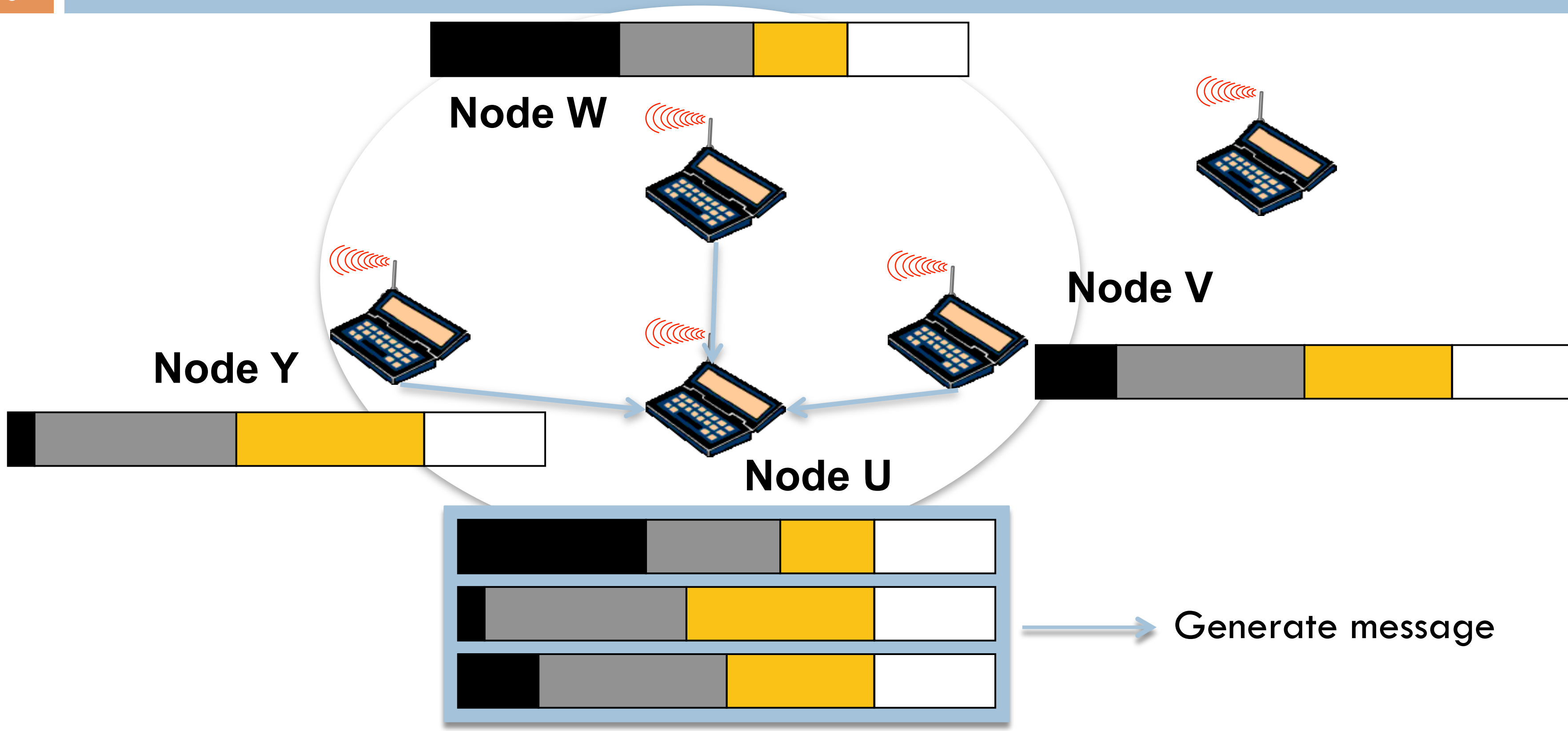


4 types of index intervals

- **Black:** unwanted indices (e.g. payloads that are no longer buffered)
- **Grey:** indices that the node is not interested in, but would not harm decoding
- **Gold:** indices that the node is interested in, in the near future
- **White:** indices that the node is interested in
  - How to set these intervals: it is a **POLICY**



Functioning

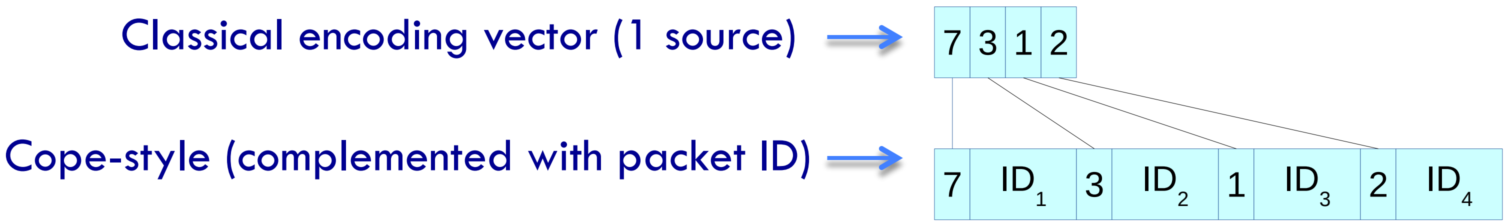


- **Question1:** How to set intervals ?
- **Question2:** How to set encoding windows ?
  - No universal answer: flexibility

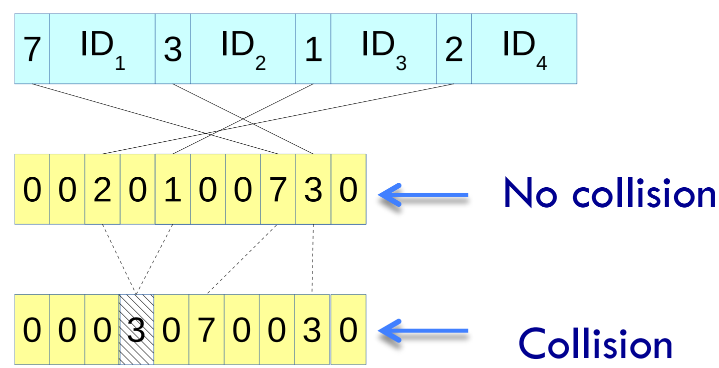
Separation:
state advertizement from signaling
principles and policies

Multiple sources (inter-flow NC)

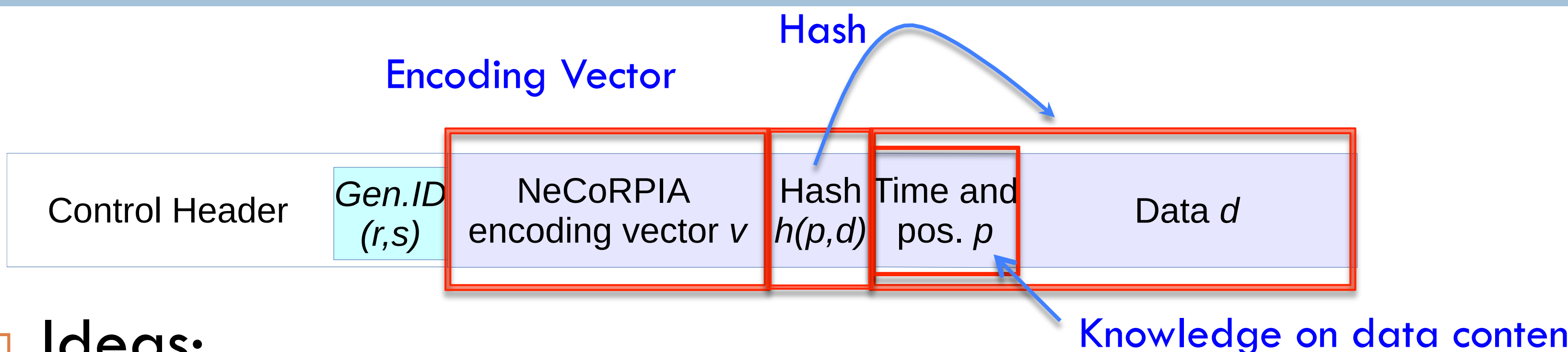
- Multiple sources in full distributed network:
  - How to index payloads? $7P_1+3\ P_2+P_3+2P_4$: whose P_1?



- NeCoRPIA
  - C. Greco, M. Kieffer, and C. Adjih, “NeCoRPIA: Network Coding with Random Packet-Index Assignment for Mobile Crowdsensing”, accepted at ICC 2015
  - Just choose a Random Payload Index
  - Problem: “payload index collisions”



Multiple sources (inter-flow NC)

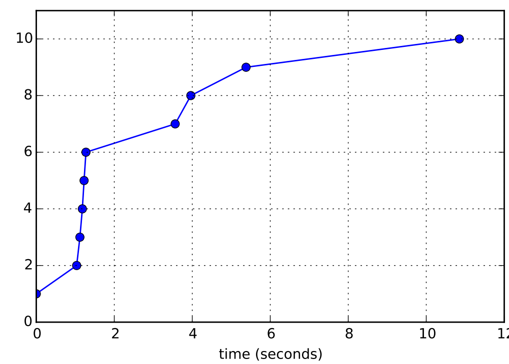


- Ideas:
  - Use knowledge of payload content to guide resolution
    - Crowdsensing application: time and position
  - Hash on content: mechanism to check decoded
  - Gaussian Elimination on the full packets
- NeCoRPIA (ICC'15):
  - constraint satisfaction problem defined over a finite field
  - Relaxation, in a sequence linear of linear programs

NeCoRPIA-lite

- Implementation (GF(2))
  - OS: RIOT http://riot-os.org/
  - Testbed, w/ WSN nodes

FIT IoT-lab https://iot-lab.info/



- Example with 10 nodes

```
0110000010100000 a832659e9c6d7d01830c40
0110001111100000 6cd02e2fa0fadab3892a20
0000000011000000 5962
0110000100000000 823d
0100001000000000 9e1e
0100000011100000 4e38
0000001101000000 c9c7
0010001000000000 48c3
0100001001000000 72cf
0110000111000000 7e83
```

```
0100000000000000 e2cd5a24f4f8e4bac2a579
0010000000000000 3410eb0434ee9573d4e870
0000001000000000 7cd31
0000000100000000 54e00
0000000010000000 b5b37
0000000001000000 ecd15
0000000000100000 cb5ca
0000000000000000 3ecba
0000000000000000 0d25e
0000000000000000 a5dcc
```

```
0100000000000000 efe8b649...(Q0+Q8)
0010000000000000 af0788cd...(Q1+Q9+Q7)
0000001000000000 4218b499...(Q2+Q7)
0000001000000000 4f3d58f4...(Q2+Q8+Q7)
0000001000000000 e7c470d3...(Q2+Q9+Q7)
0000000100000000 cff76483...(Q3+Q9+Q7)
0000000010000000 106fb599...(Q4+Q9)
0000000001000000 490d9bab...(Q5+Q9)
0000000001000000 d21af862...(Q5+Q7)
0000000000100000 63a58d4a...(Q6+Q8+Q9)
```

Conclusion

- Elements for NC in wireless multi-hop networks
  - Interest of the RG ?

THANK YOU

DragonNet Packet Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Node ID              |        Sequence Number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              DRAGON Parameters                                |
|                                       +-+-+-+-+-+-+-+-+-+-+-+-+
|                                       |                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                       |
|                                                               |
|              Flow Parameters                                  |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                 Coded Data                                    |
|                                       +-+-+-+-+-+-+-+-+-+-+-+-+
|                                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4.1: The format of DragonNet message as specified for WSNs.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Message Interval       | Sliding Encoding Window Size |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Coded Block size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Rank of the Transmitter   |          High Index           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Low  Index           |     Number of Neighbors       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           LastTime            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

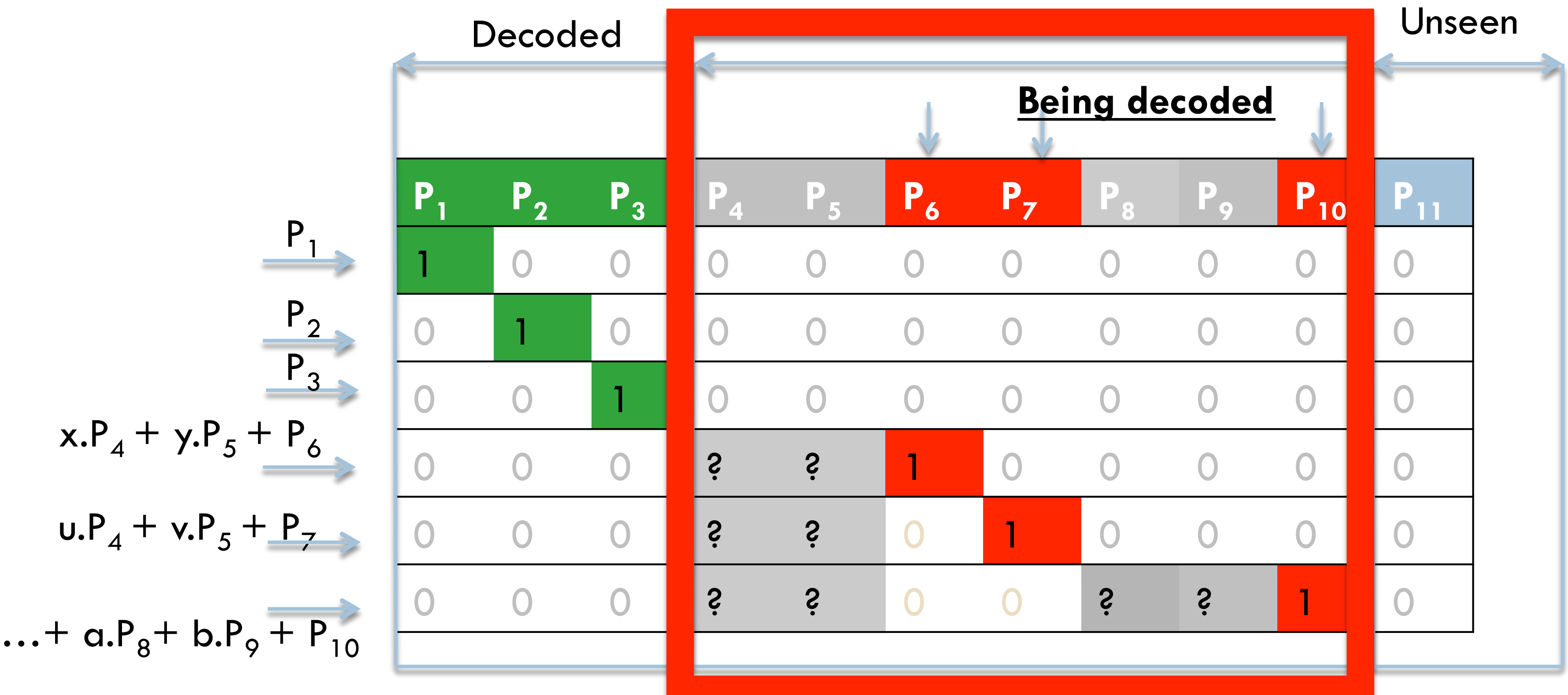
Figure 4.2: Dragon Related Parameters.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Coef Pos Min             |        Coef Pos Max           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Data Size                |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                     Coded Payload Data                        |
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4.4: Coded Payload Related Parameters.

SEW: Sliding Encoding Window

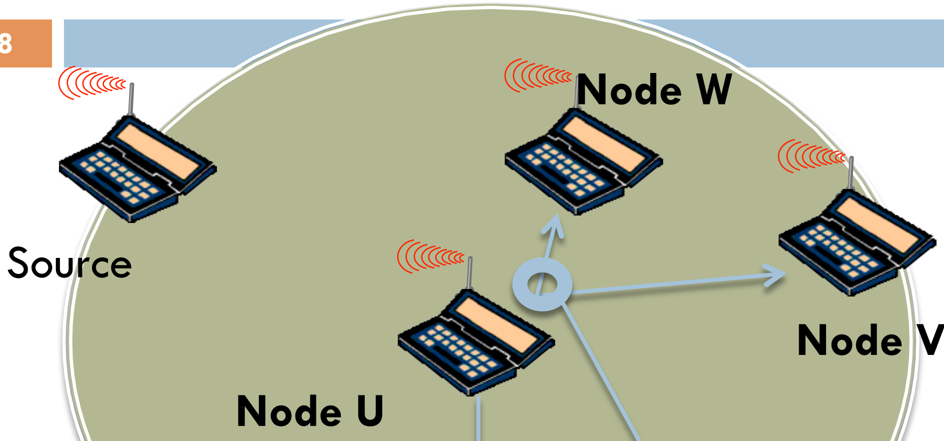
- Principle: “real-time” robust decoding
- Variant of Gaussian elimination (“inverted” RREF)



CISEW Policy

- Depends on:
  - Computing and storage capacity of nodes:
    - E.g; a huge advertized Gold interval -> too many undecoded packets, no suitable for low capacity sensors
  - Application requirements:
    - E.g1: real time application: nodes decoding should evolve in parallel, nodes should have close intervals. A too late node should increase its interval even if some payloads are not decoded in order not prevent neighbors from progressing
    - E.g2: For code distribution (over the air reflashing in WSNs), all payloads are equally important, a node must still requesting the same undecoded indices if its neighbors are much more progressed.

Protocol Overview



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.=3
Encoding vector=(1,3,5,0,…)
Content = [$P_1 + 3\,P_2 + 5\,P_4$]

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 =3