CodeDrip: Data Dissemination Protocol with Network Coding for Wireless Sensor Networks

Nildo dos Santos Ribeiro Júnior[1]
Marcos A. M. Vieira[1]
Luiz F. M. Vieira[1]
Omprakash Gnawali[2]

[1]Universidade Federal de Minas Gerais, Brazil

[2]University of Houston, USA

Topic: Improve Dissemination

Technique: Network Coding with XOR

Result: Up to 20% fewer packets and 25% faster

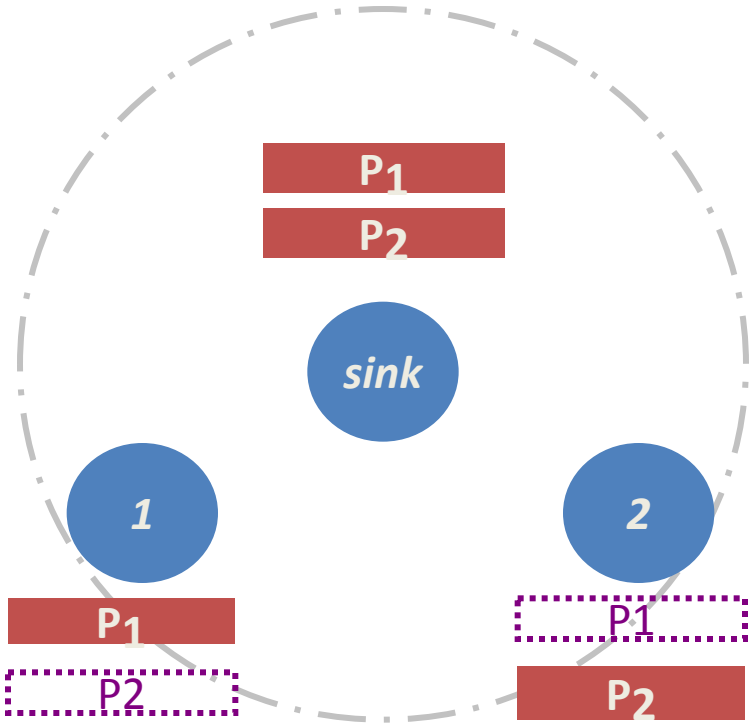
Plan

- Network Coding
- Data Dissemination
- CodeDrip
- Evaluation

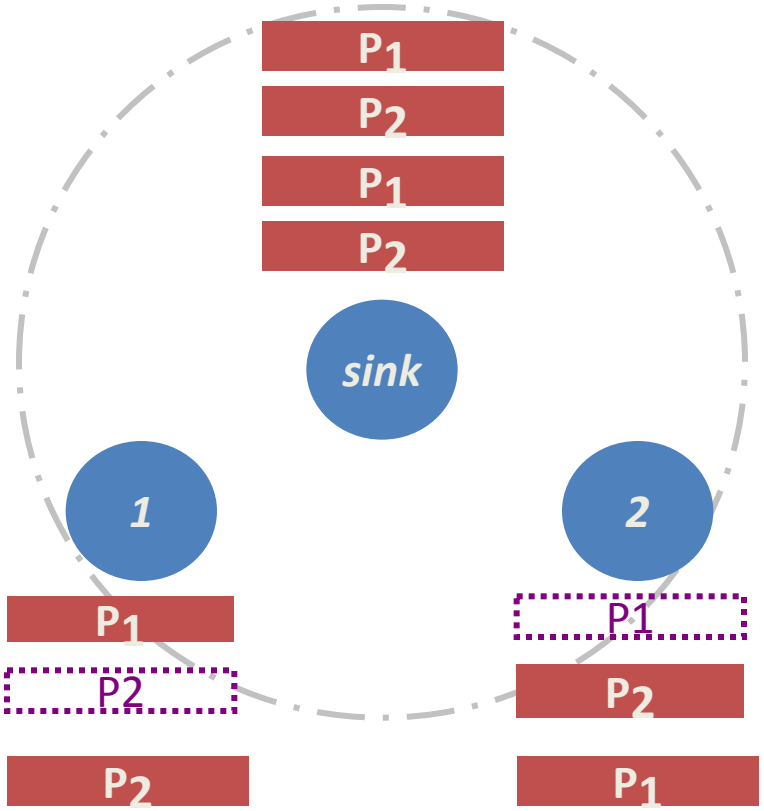
Network Coding

- Combine some of the packets sent in the network
- Simple and efficient: XOR
- Lessens the impact of packet loss on efficiency

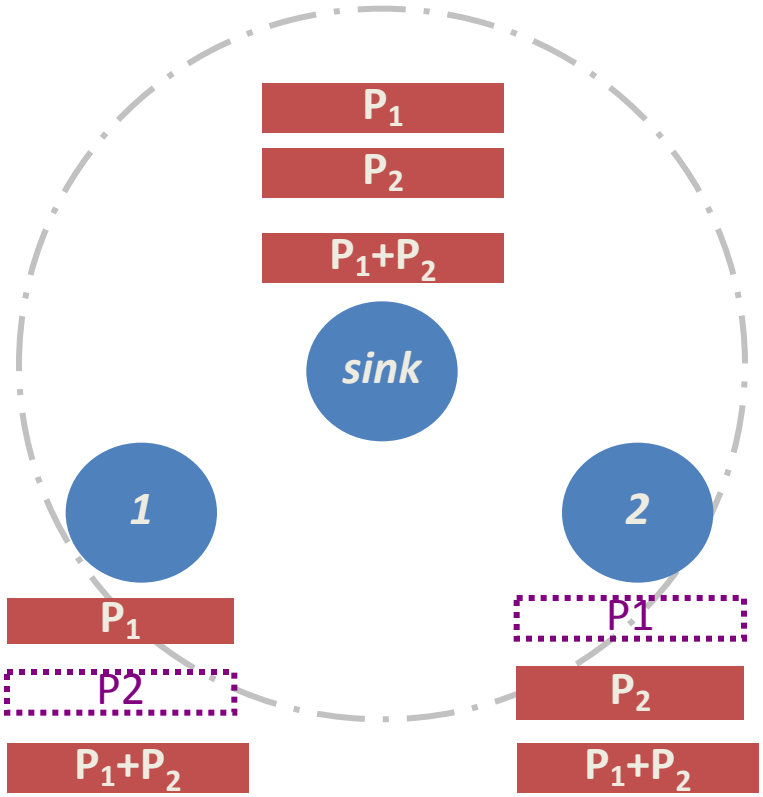
Packets: Sink to Network



Standard Retransmissions



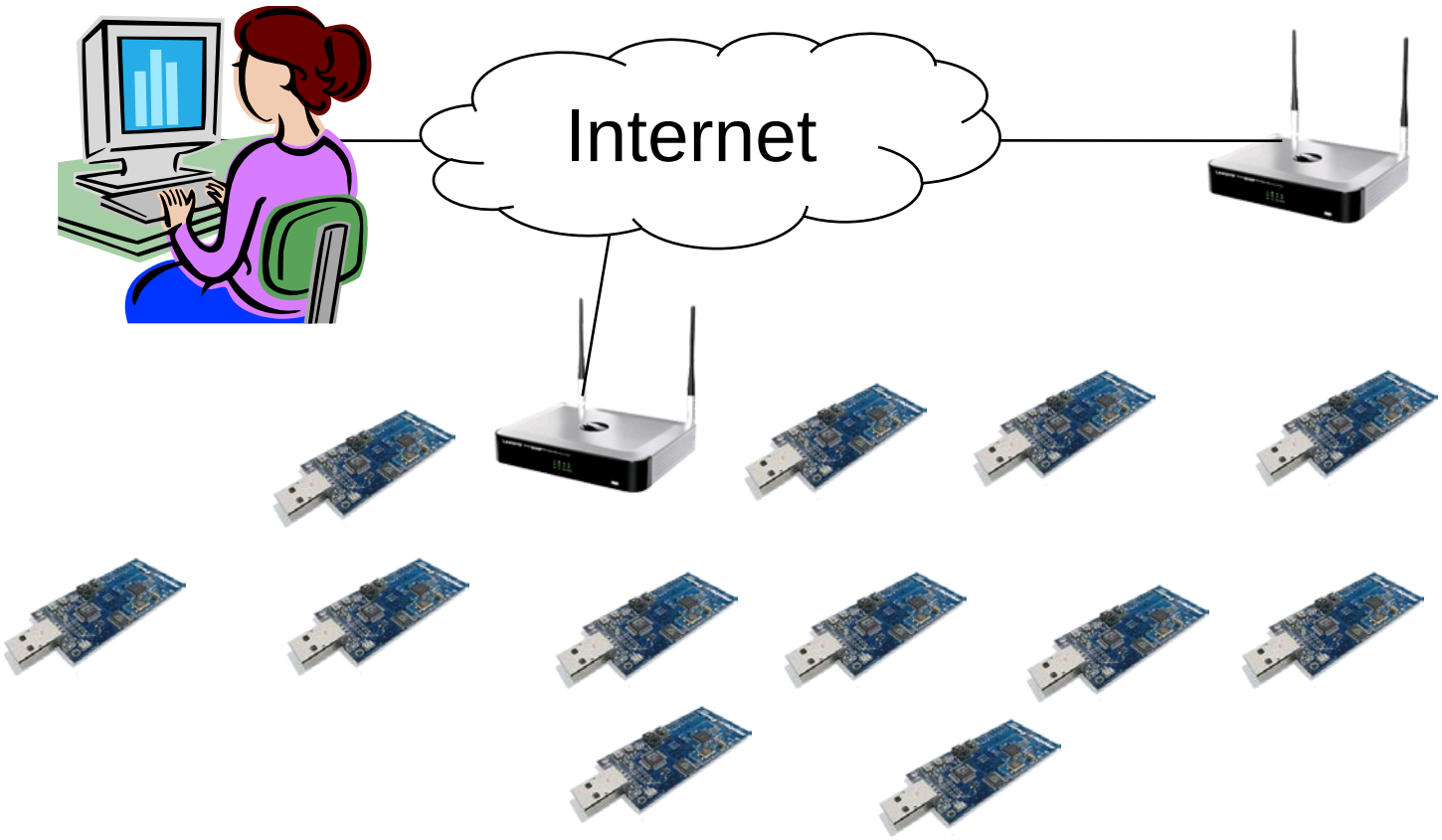
Network Coding



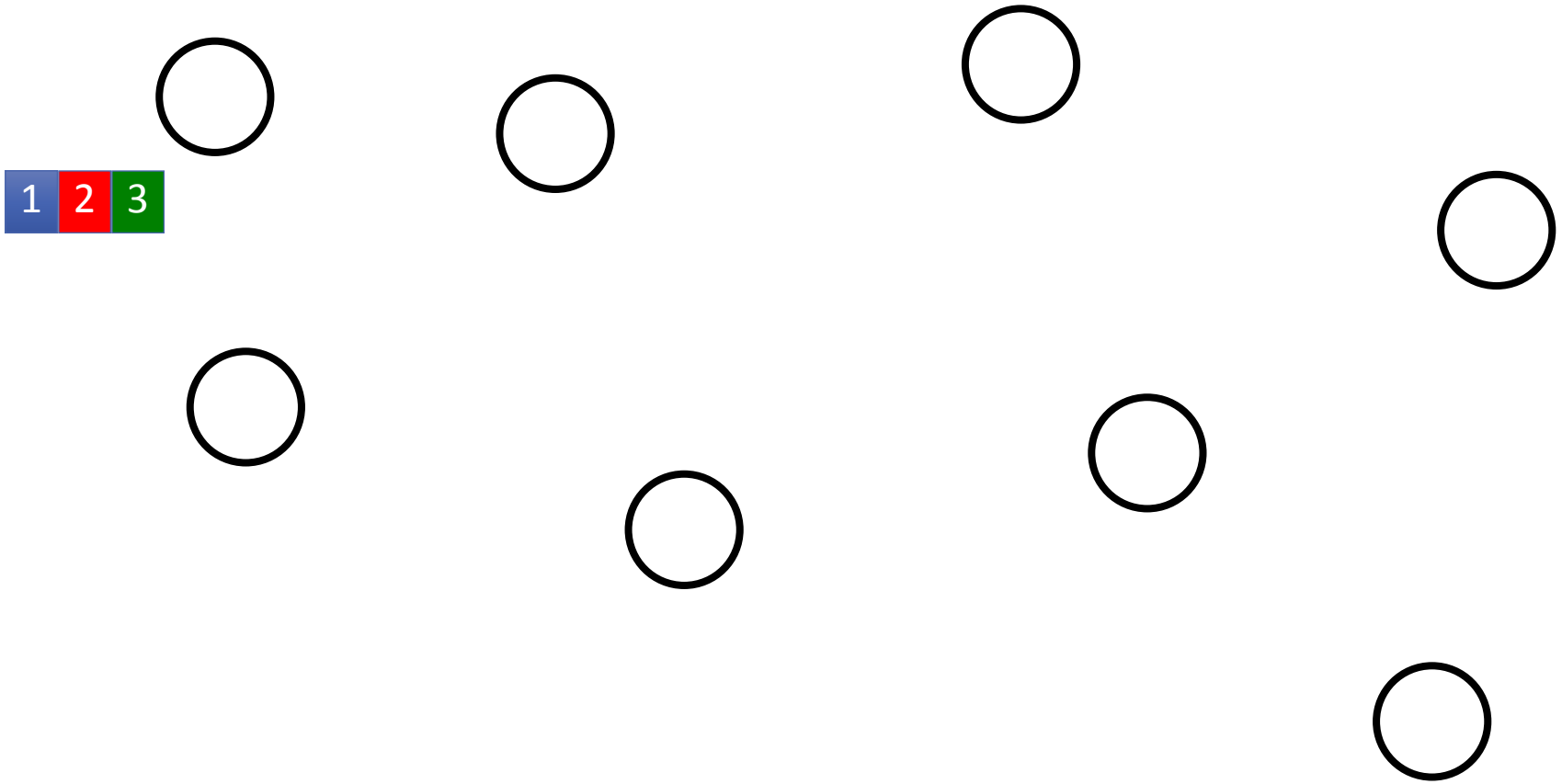
Plan

- Network Coding
- Data Dissemination
- CodeDrip
- Evaluation

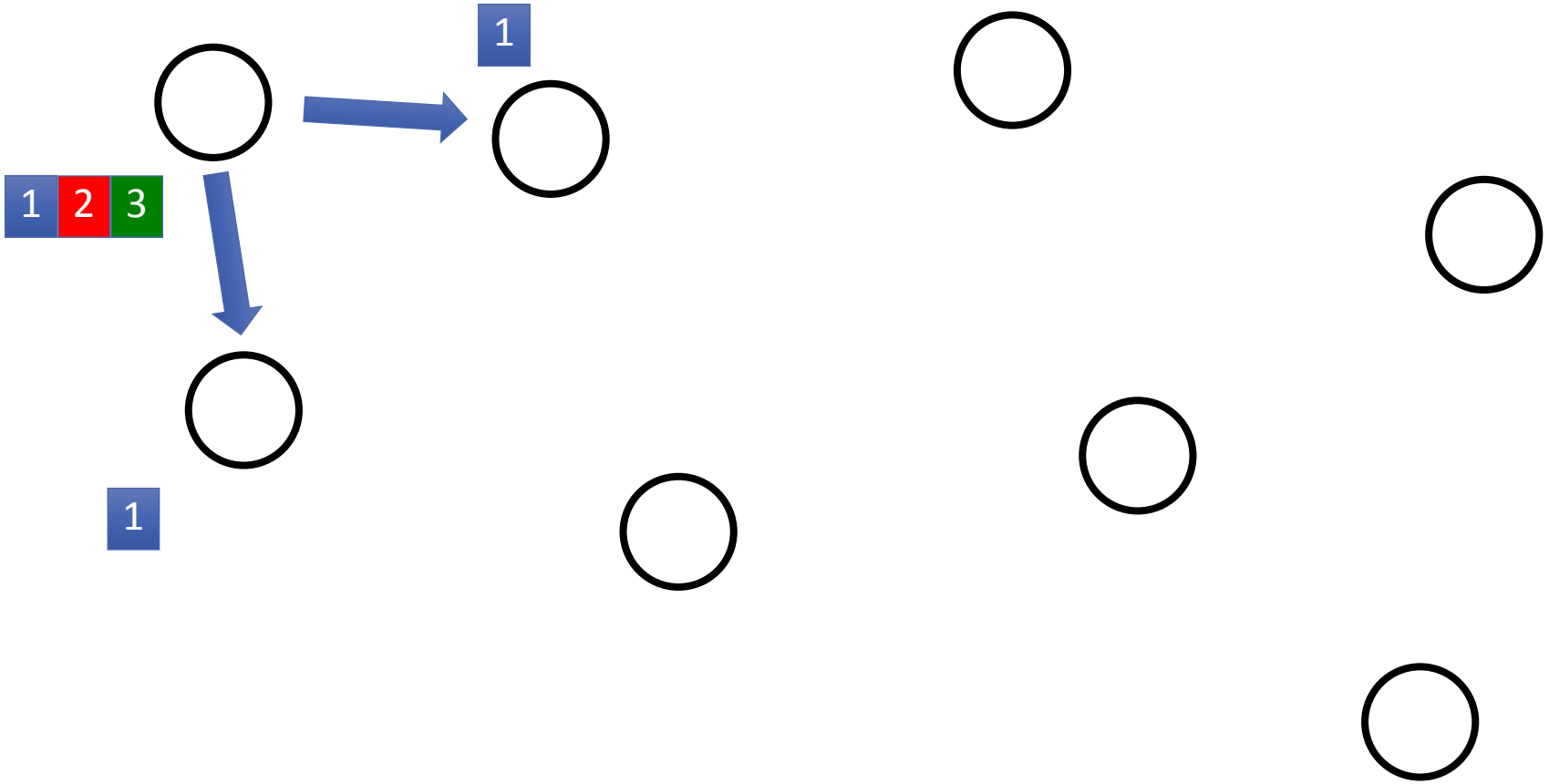
Wireless Sensor Networks



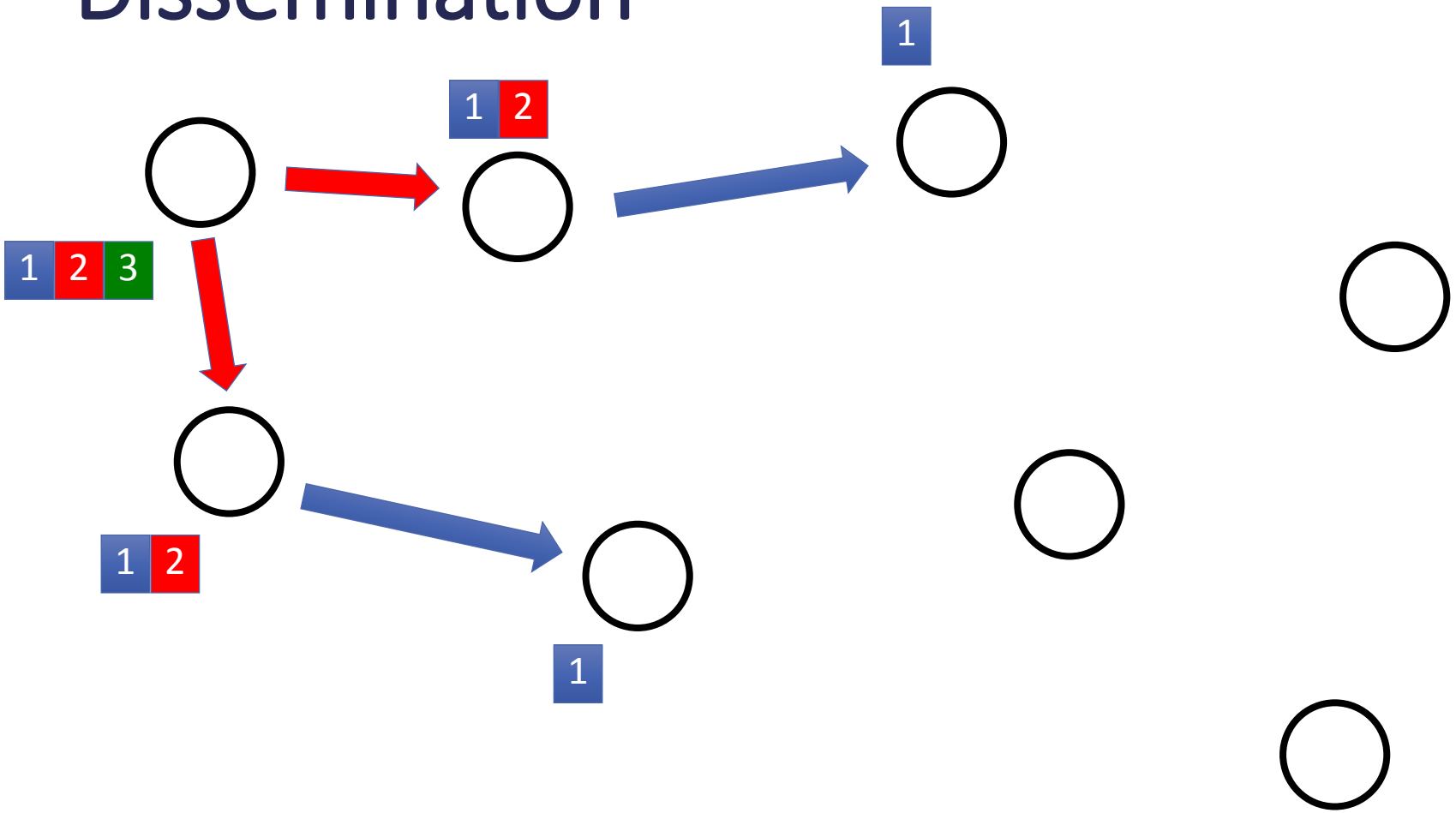
Dissemination



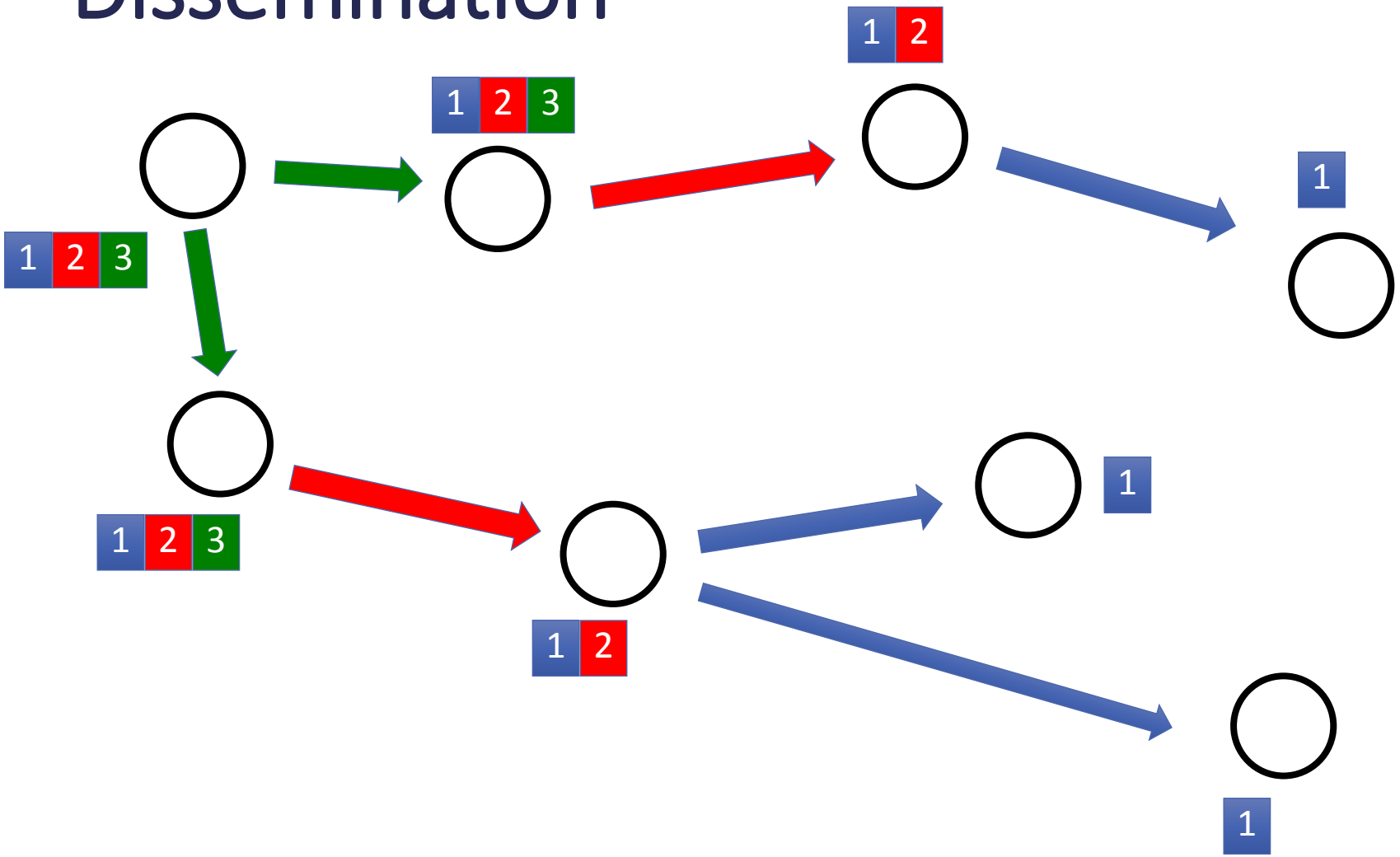
Dissemination



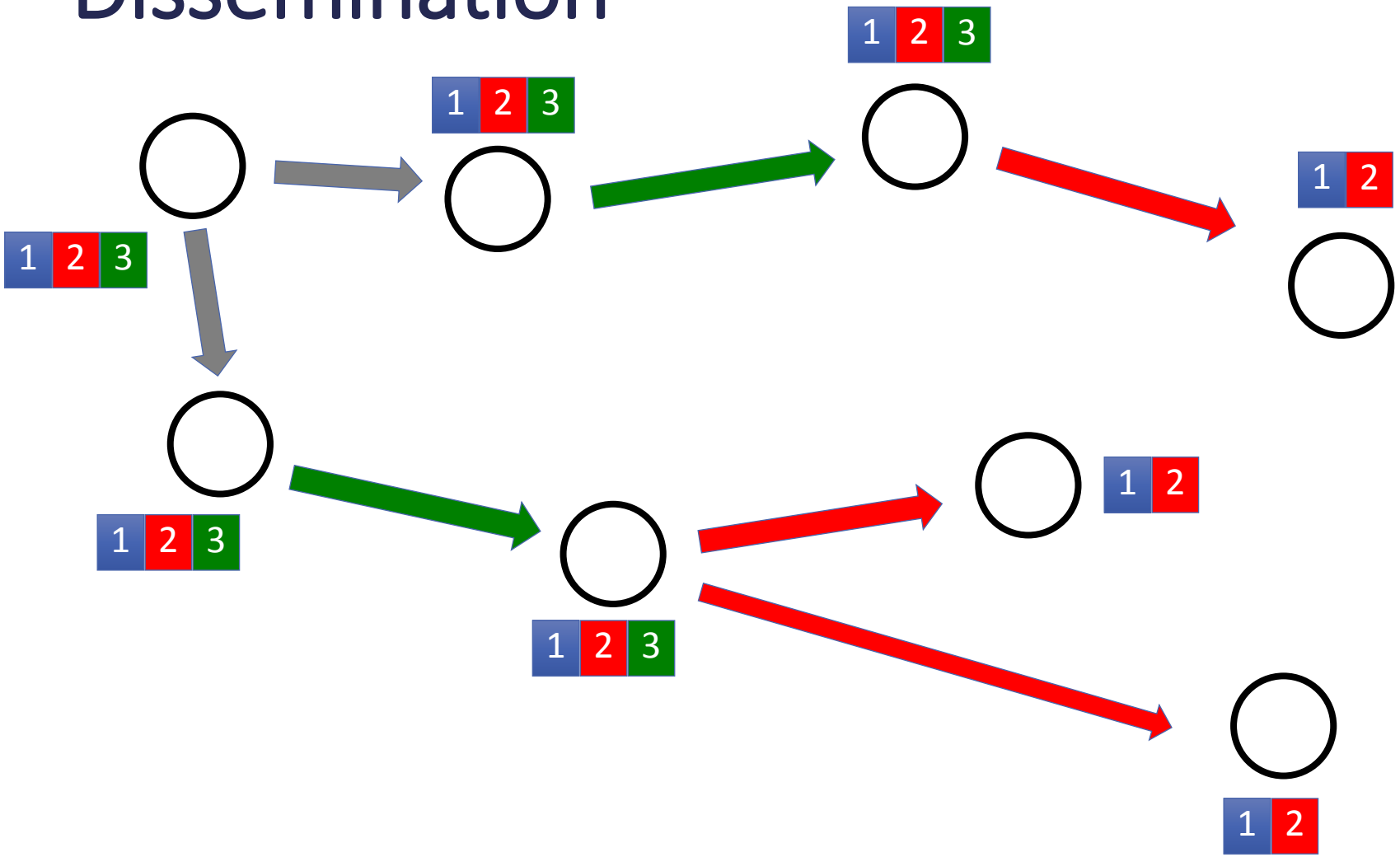
Dissemination



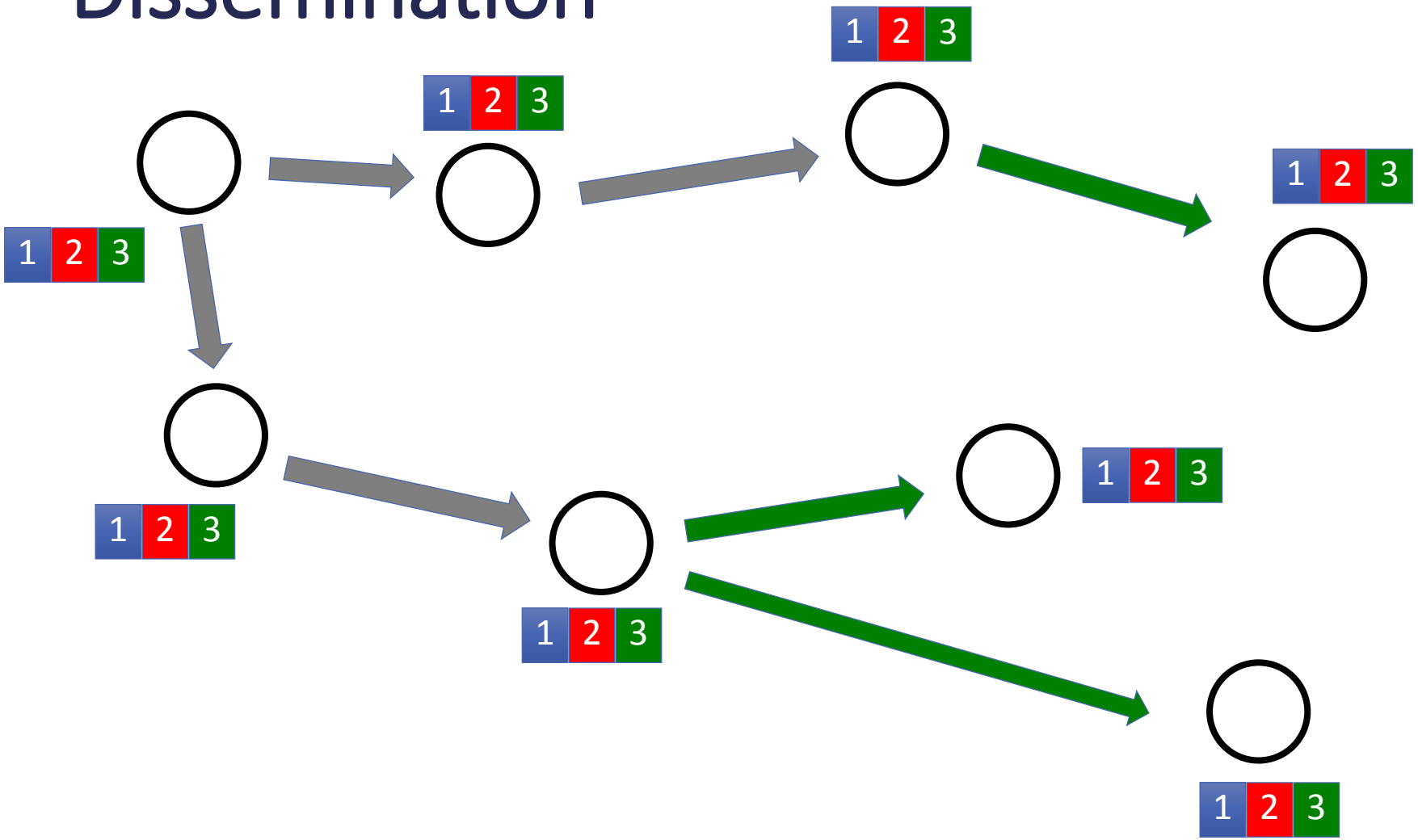
Dissemination



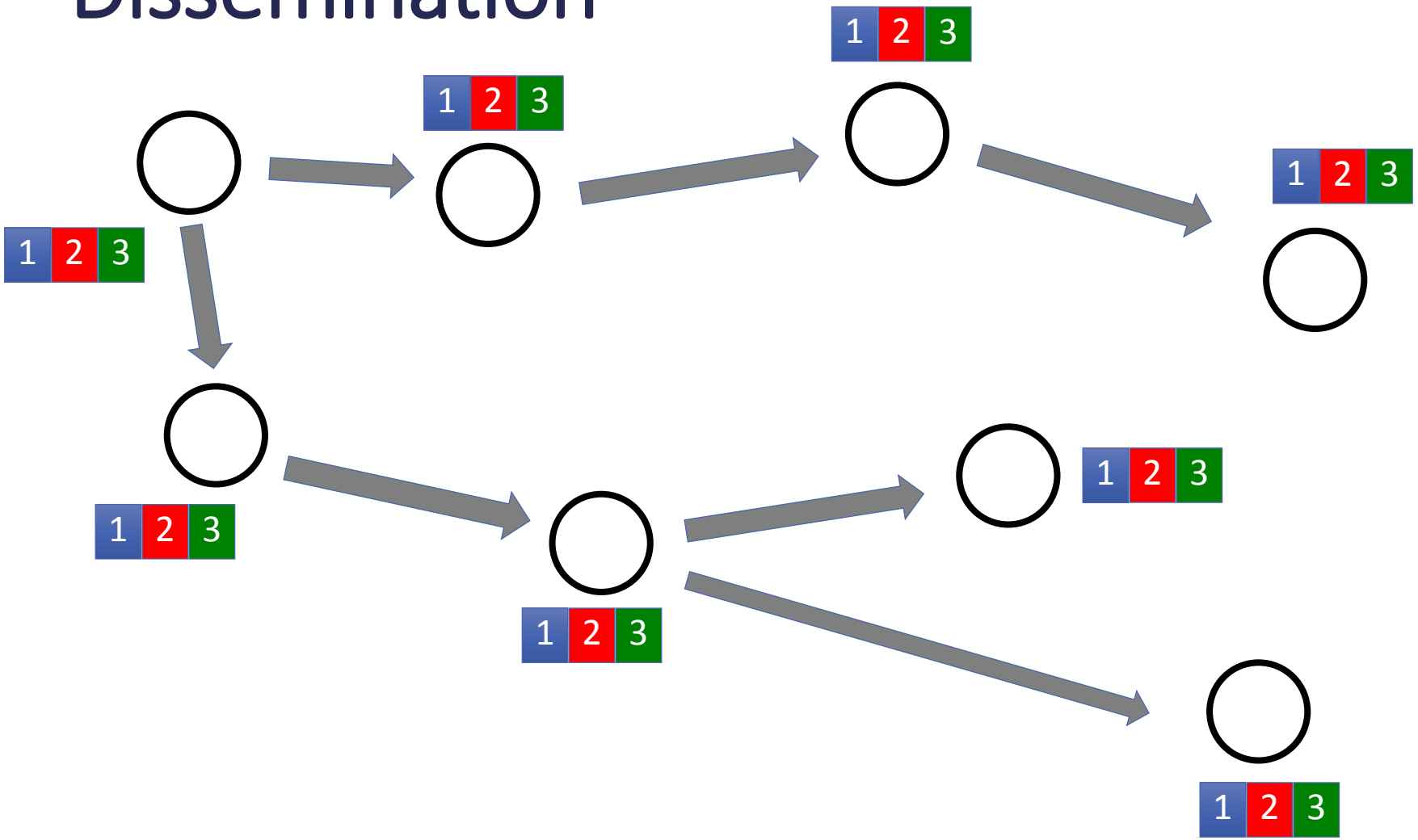
Dissemination



Dissemination



Dissemination



Data Dissemination

- Transmission errors common due to contention
- Desirable
  - Low power
  - Fast

Drip

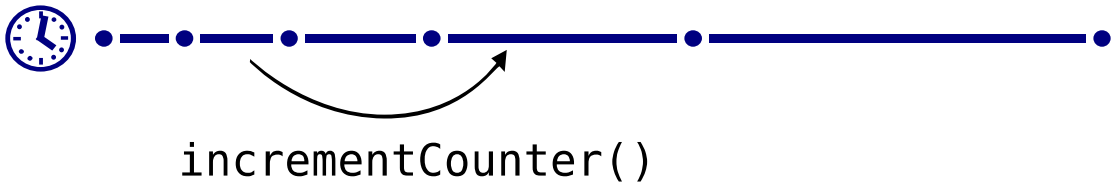
- Data dissemination protocol for WSNs
- Used to disseminate small data
- Uses Trickle Timers to time packet transmission

Trickle Timer

- A timer that increments its period over time

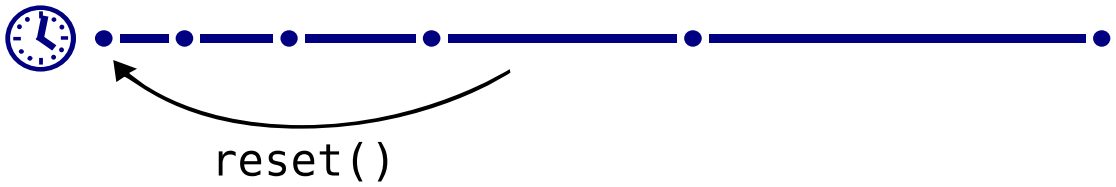
Trickle Timer

- A timer that increments its period over time



Trickle Timer

- Reset when there is a new version of data available



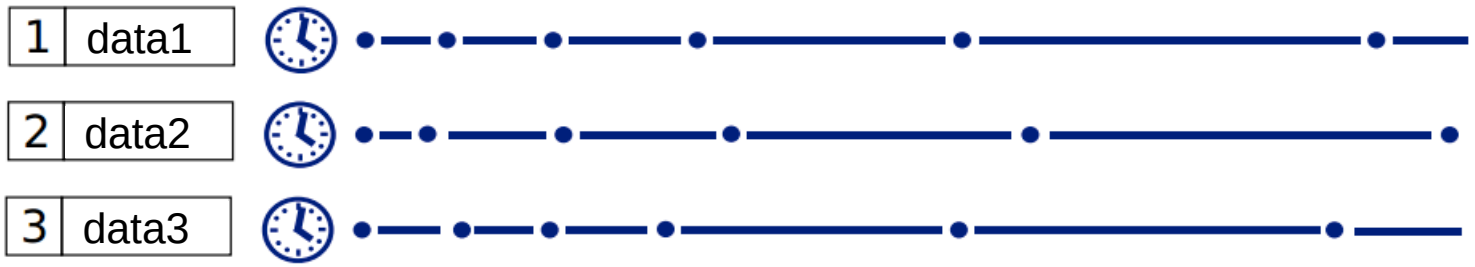
Trickle

Event	Action
τ Expires	Double τ, up to τ_h. Reset c, pick a new t
t Expires	If c < k, transmit
Receive same metadata	Increment c
Receive newer metadata	Set τ to τ_l. Reset c, pick a new t
Receive older metadata	Send updates

[Levis 2004]

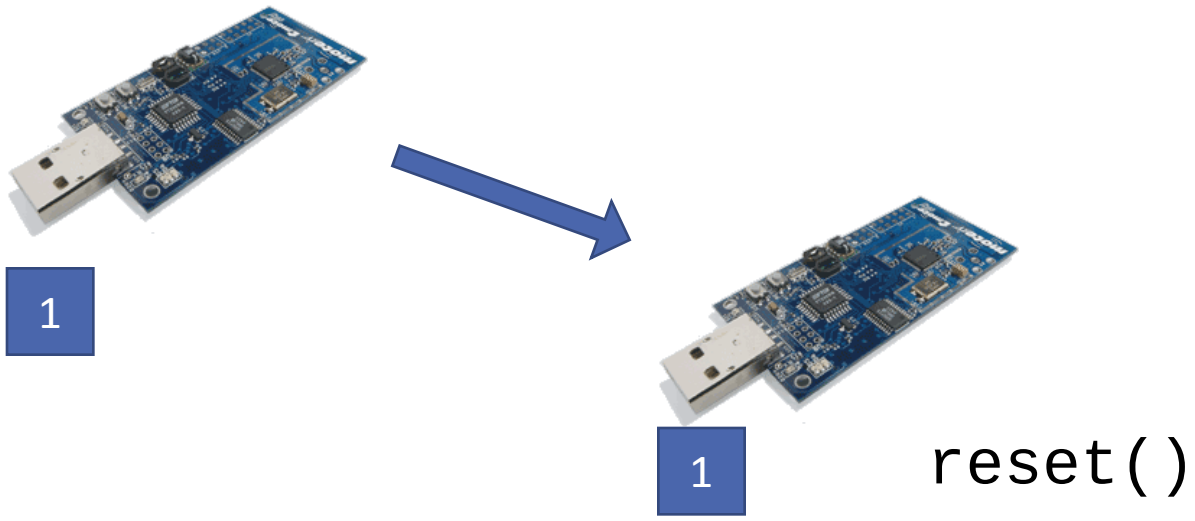
Drip

- Each data being disseminated has:
  - a unique key
  - a Trickle Timer associated with it



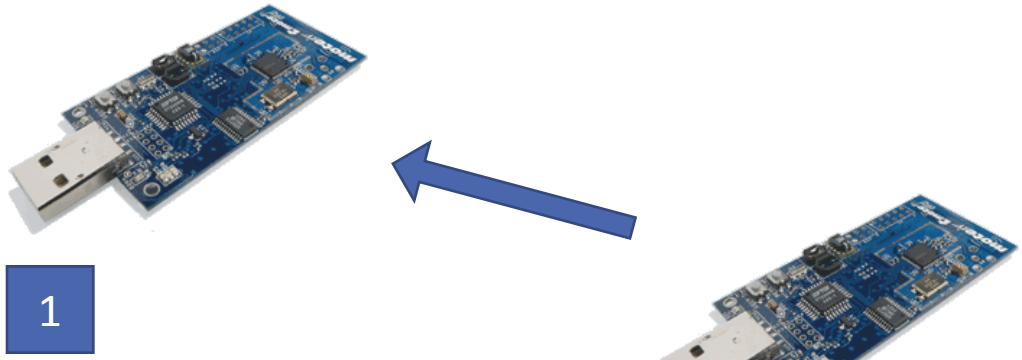
Drip

- When a node receives a new packet, it calls `reset()` for the associated Trickle Timer.



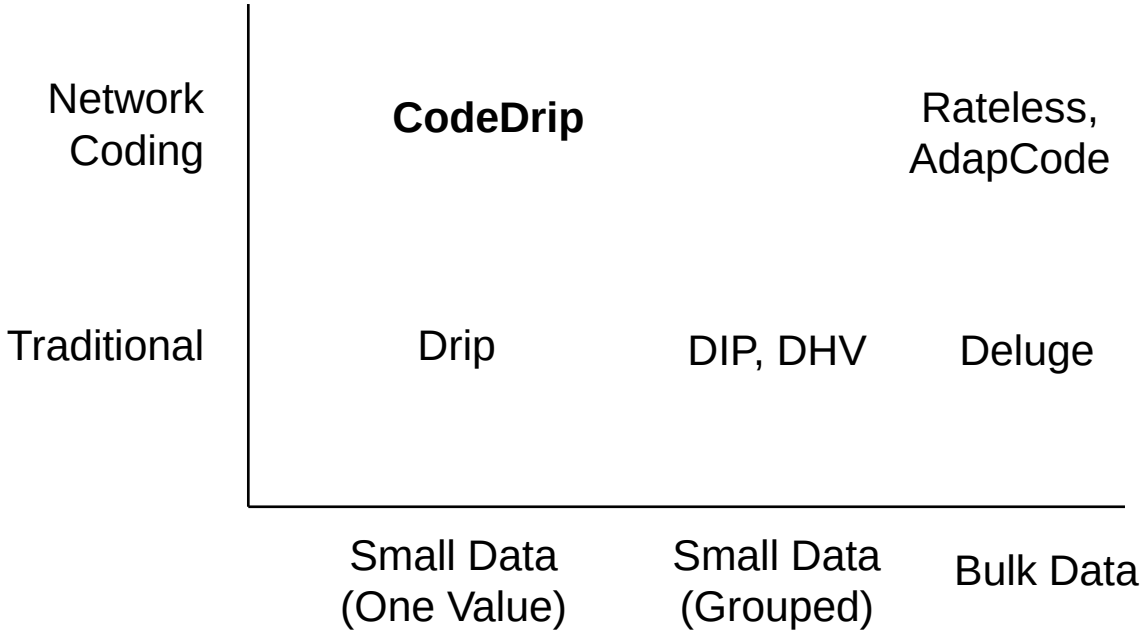
Drip

- When a node receives a packet it already has, it calls `incrementCounter()` for the associated Trickle Timer.



`incrementCounter()`

Dissemination Protocols

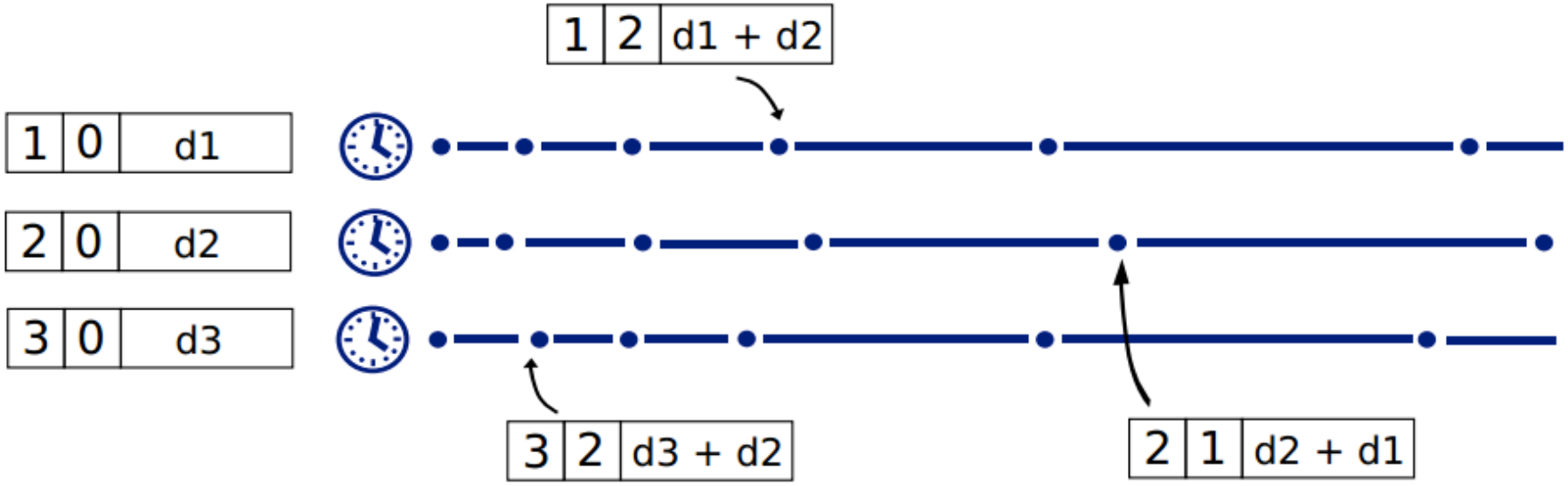


Plan

- Network Coding
- Data Dissemination
- CodeDrip
- Evaluation

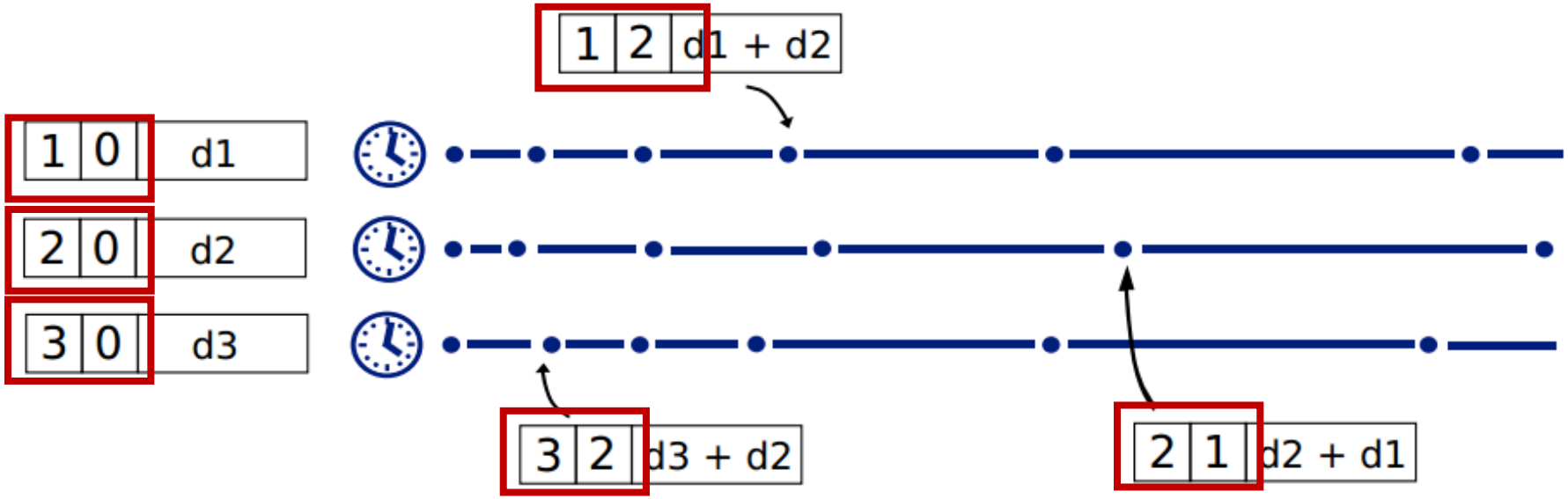
CodeDrip

- Similar to Drip but uses Network Coding.



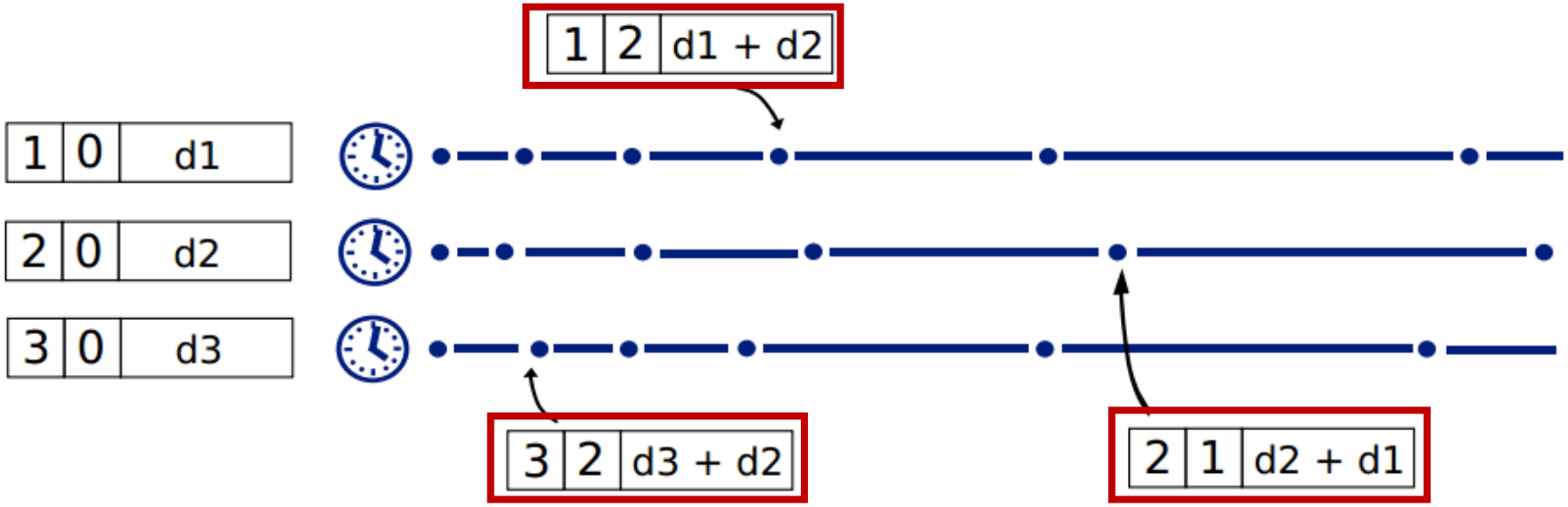
CodeDrip

- Packets have two or more keys, indicating which data was combined.



CodeDrip : Transmission

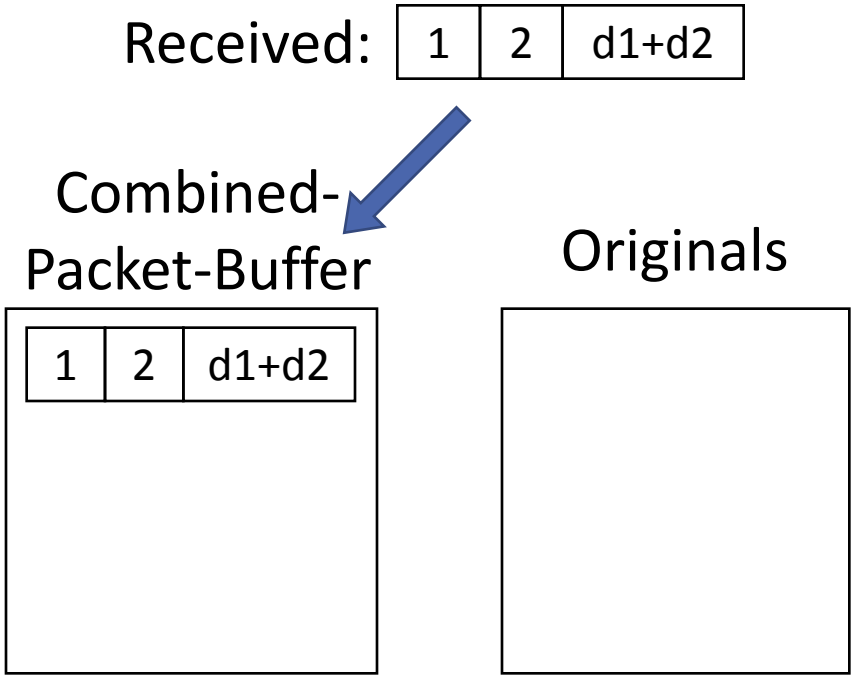
- In some transmissions, the data is combined with another one chosen randomly.



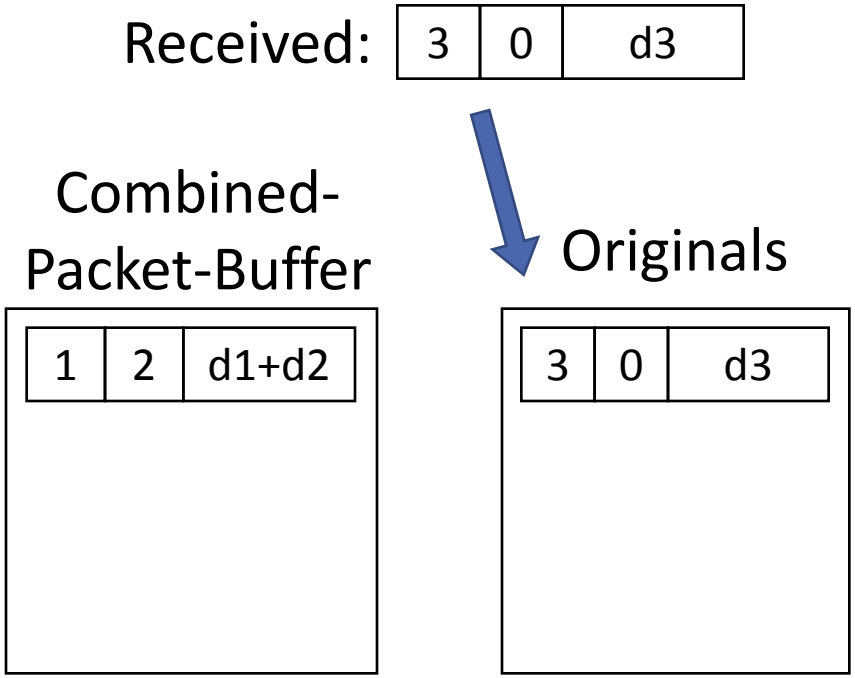
CodeDrip : Reception

- Keep all combined packets received but unable to decode in the combined-packet-buffer.
- Upon each packet reception, try to decode the message in the combined-packet-buffer.

CodeDrip

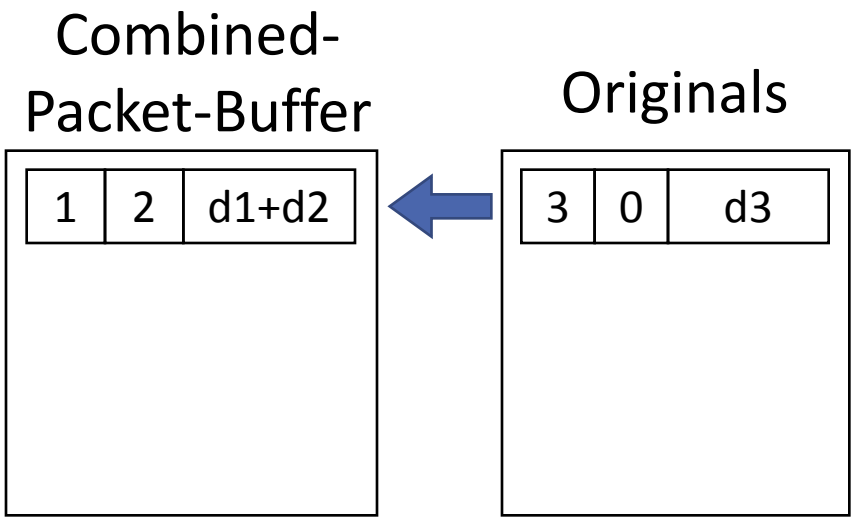


CodeDrip

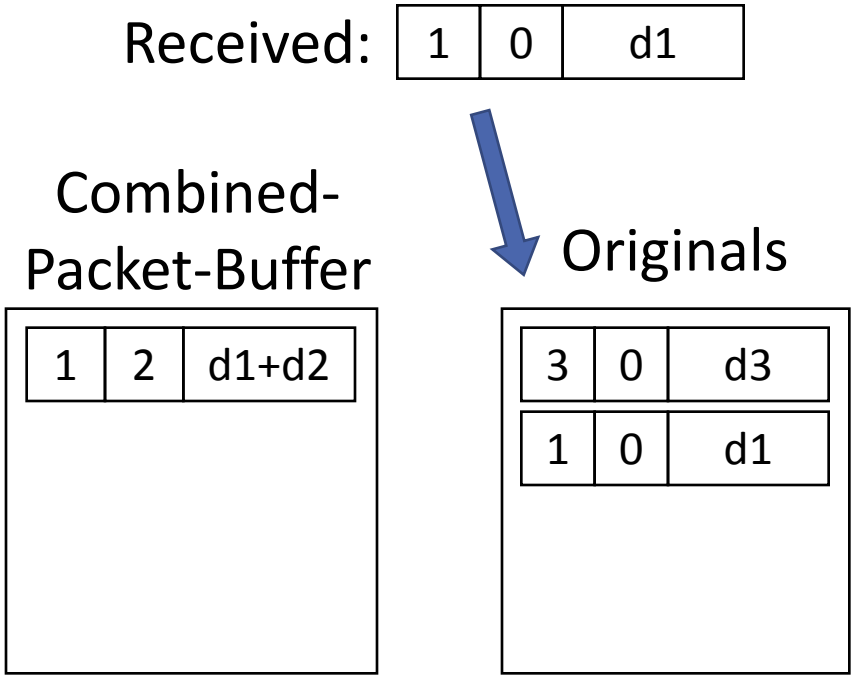

Received:
3
0
d3
Combined-
Packet-Buffer
1
2
d1+d2
Originals
3
0
d3

CodeDrip

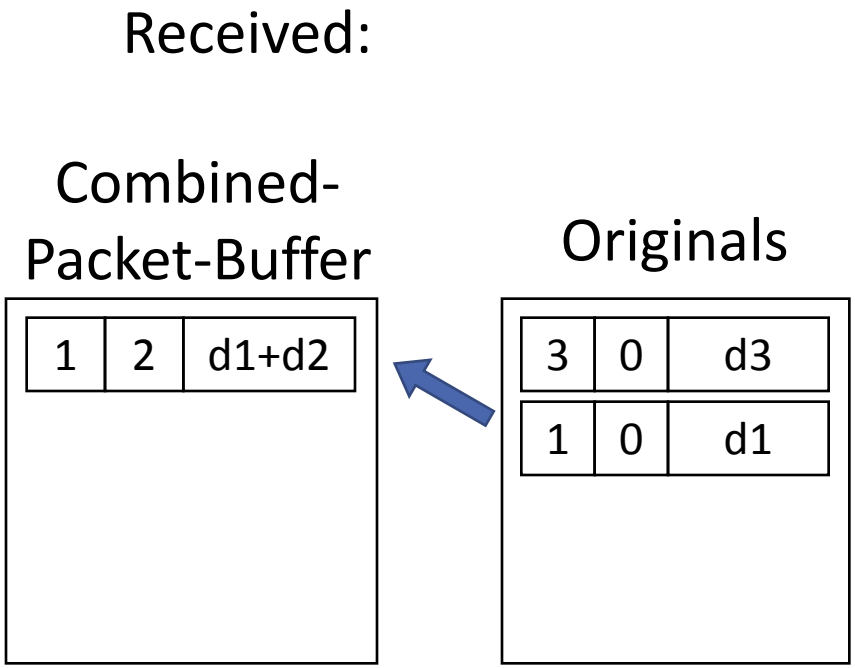
Received:



CodeDrip


Received:
1
0
d1
Combined-
Packet-Buffer
1
2
d1+d2
Originals
3
0
d3
1
0
d1

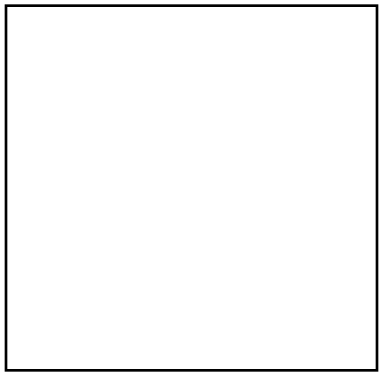
CodeDrip


Received:
Combined-
Packet-Buffer
1
2
d1+d2
Originals
3
0
d3
1
0
d1

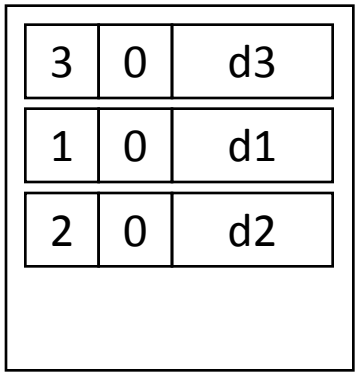
CodeDrip

Received:

Combined-Packet-Buffer



Originals



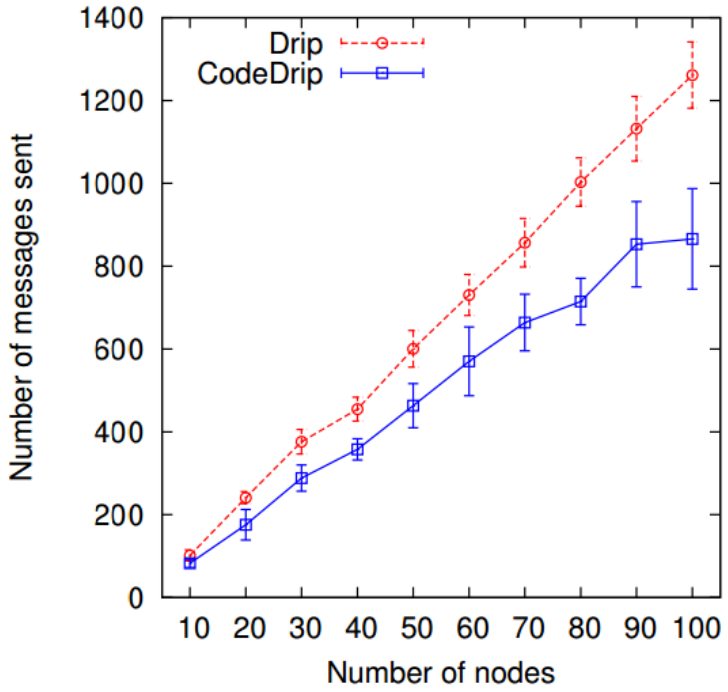
Plan

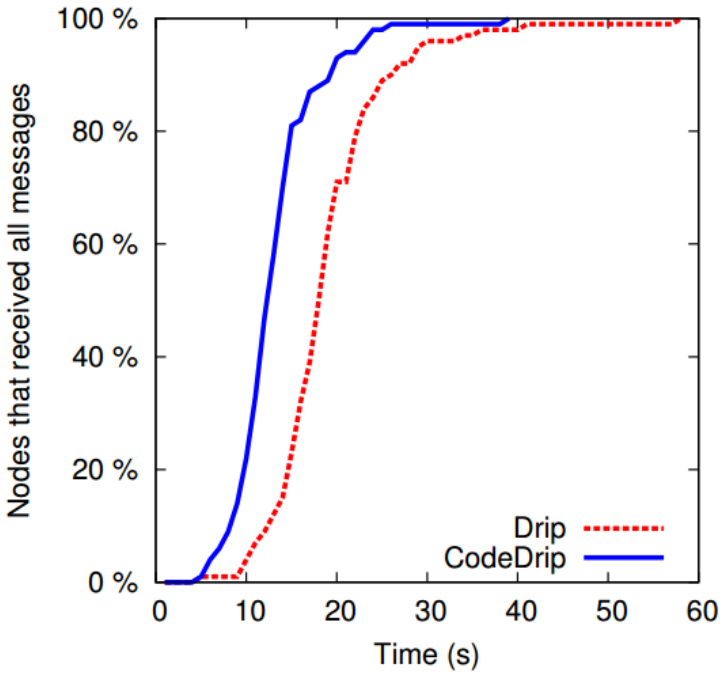
- Network Coding
- Data Dissemination
- CodeDrip
- Evaluation

Experiment Setup

- Compare CodeDrip against Drip/DIP/DHV
- Simulation
  - TOSSIM
  - 100 nodes
  - Disseminate 3 values from the sink
- Testbed
  - Kansei (TelosB)
  - 10-70 nodes
  - Disseminate 10 values from the sink

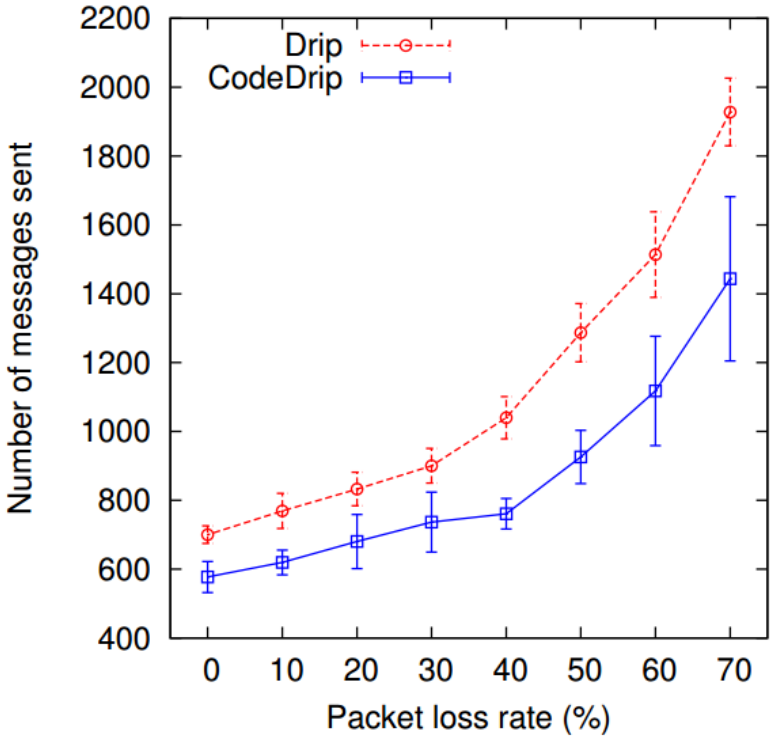
Results Simulation

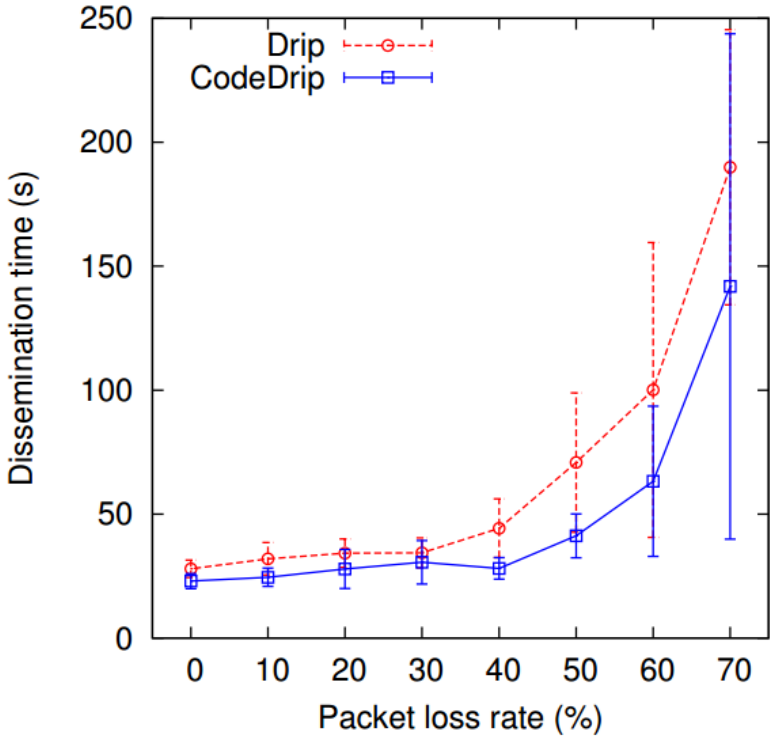




CodeDrip faster and more efficient than Drip

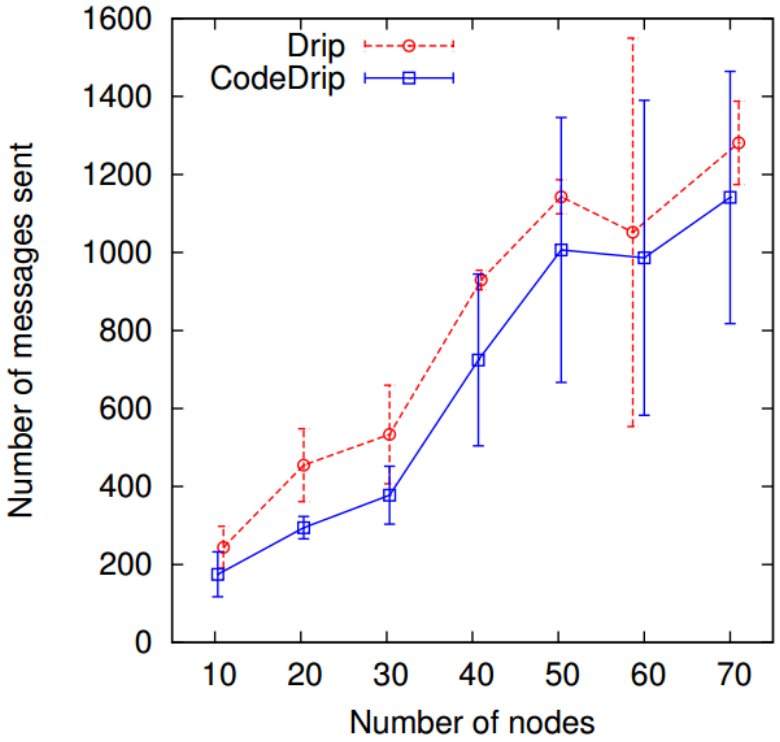
Results Simulation

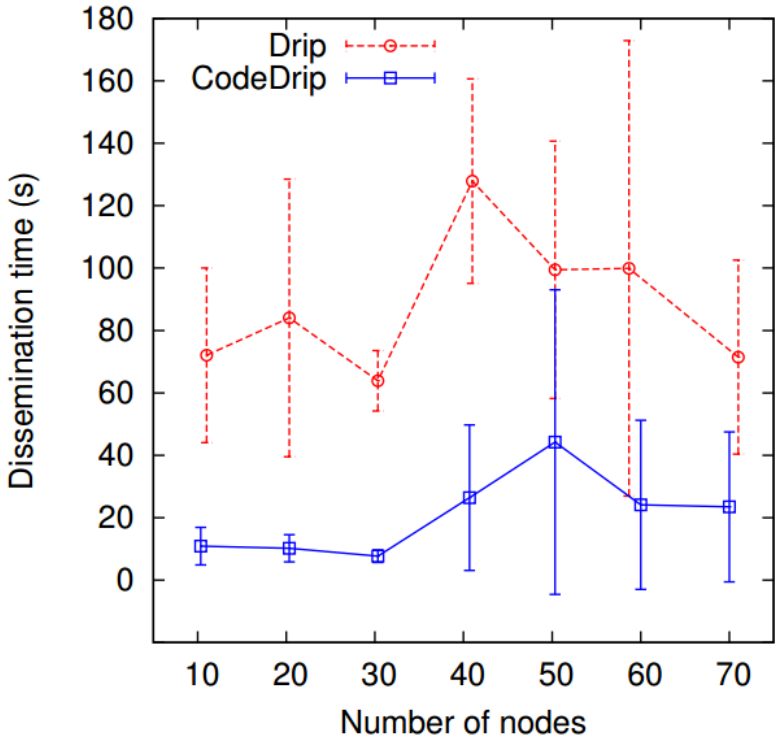




CodeDrip gain increases in lossy topologies

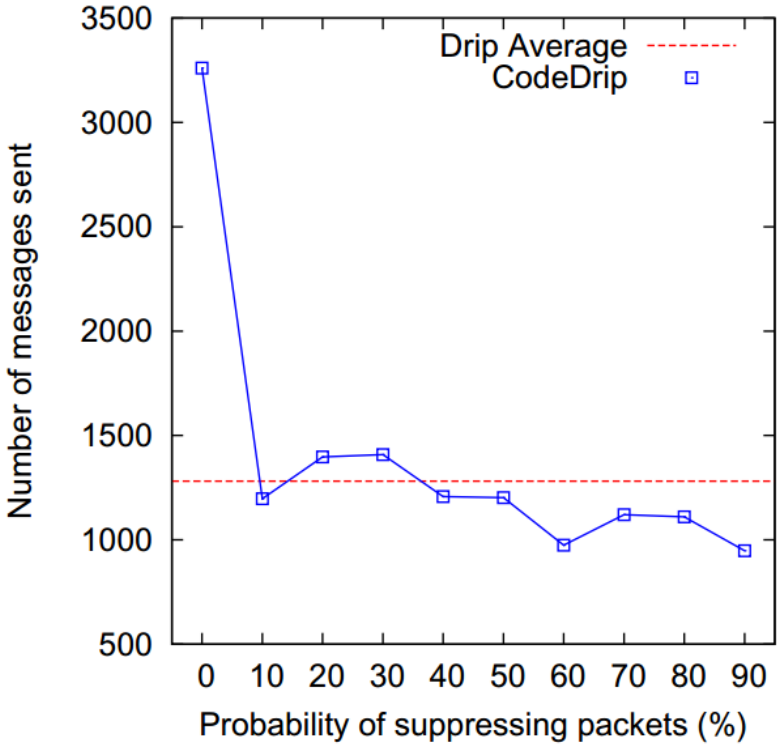
Results Testbed

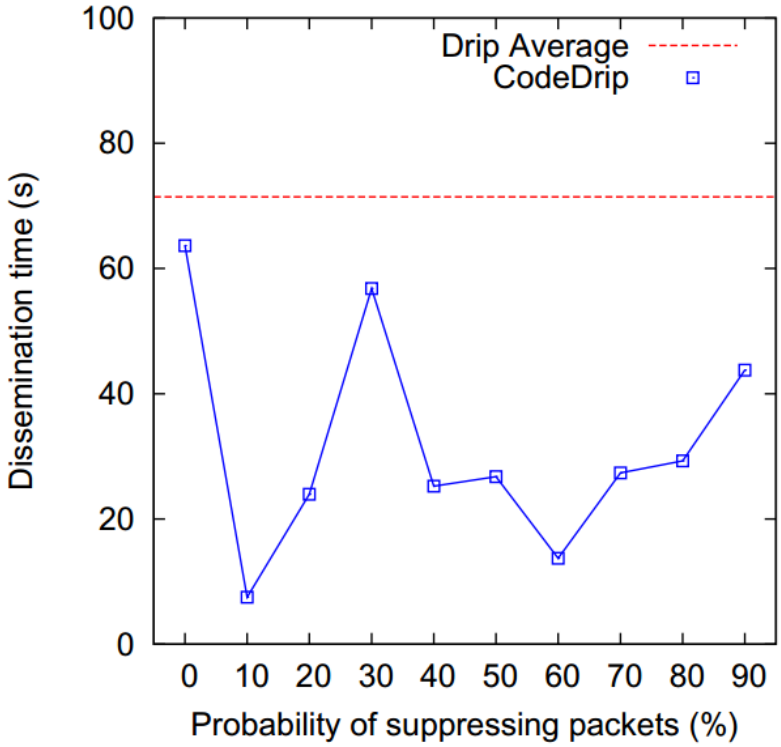




CodeDrip more efficient and faster than Drip

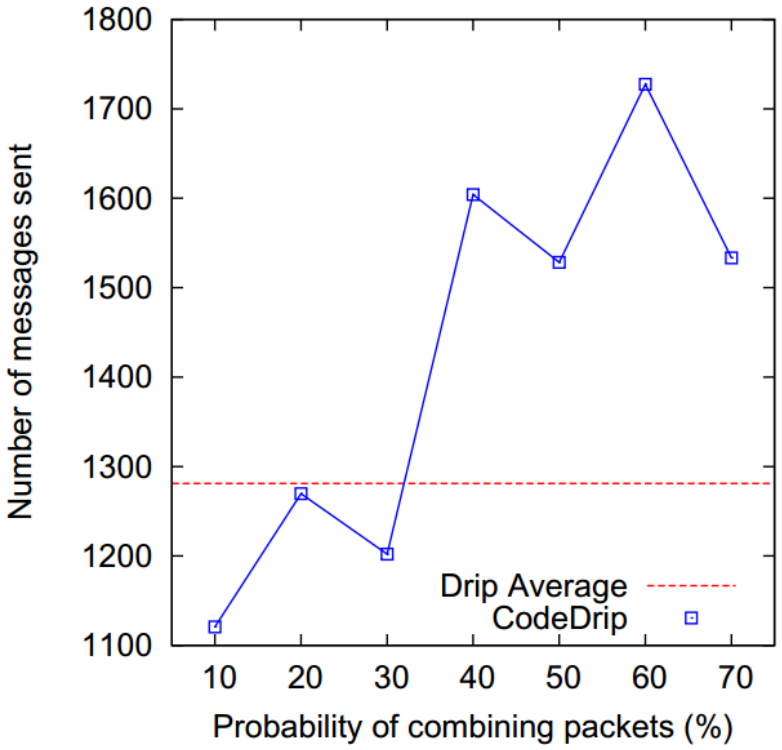
Results Testbed





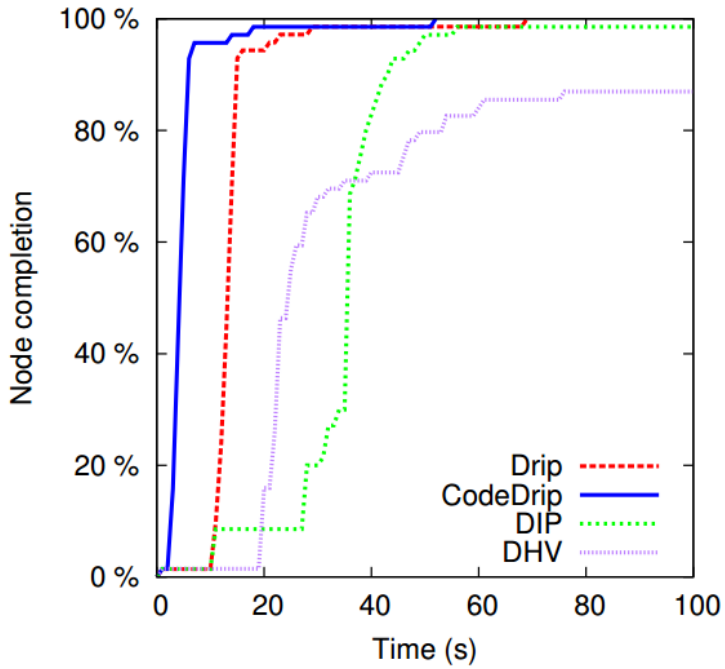
Suppression Probability of 0.4-0.8 maybe appropriate

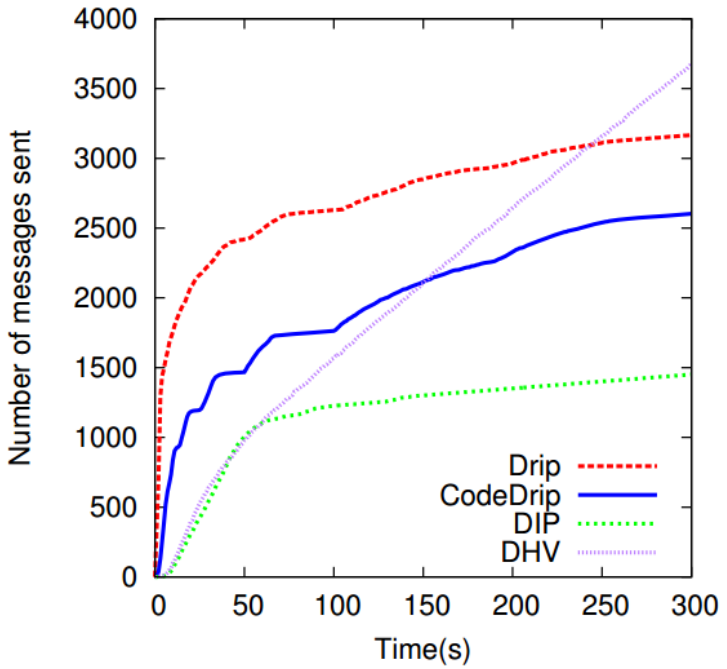
Results Testbed



Larger combining probability reduces decoding probability

Results Testbed





CodeDrip completes faster and is generally more efficient

Conclusion

- CodeDrip is more efficient and faster than Drip/DIP/DHV for dissemination of small number of values
- Metadata overhead modest
- Code publicly available for experimentation http://www2.cs.uh.edu/~gnawali/codedrip/

Memory usage

Metric	CodeDrip	Drip	DIP	DHV
RAM (bytes)	900	845	959	928
ROM (bytes)	17980	21274	22130	21478
Code size (bytes)	42655	50123	49493	49003

- CodeDrip
  - Smaller ROM size than Drip because all data was encoded in one interface.