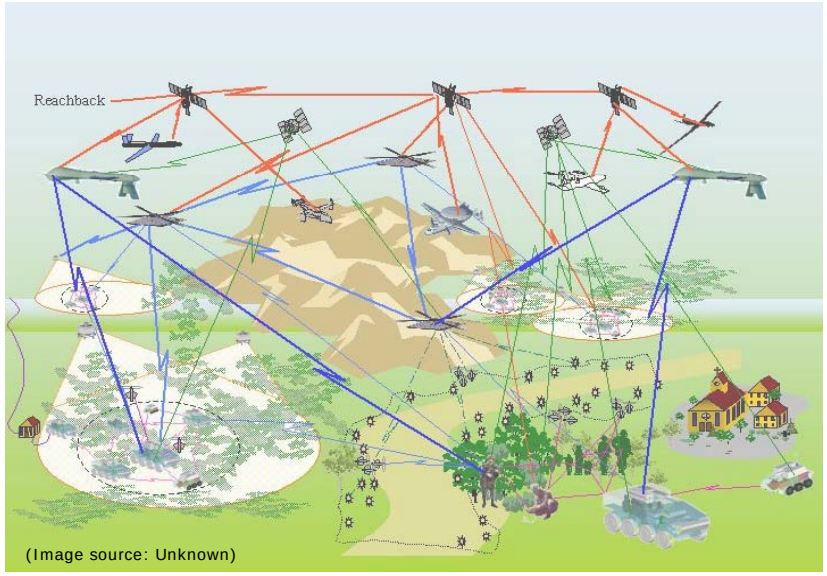
Efficient Data Dissemination in MANET using Network Coding

Joon-Sang Park
Hongik Univ.
Dept. of Computer Eng.
jsp@hongik.ac.kr

MANET (Mobile Ad-hoc NETworks)

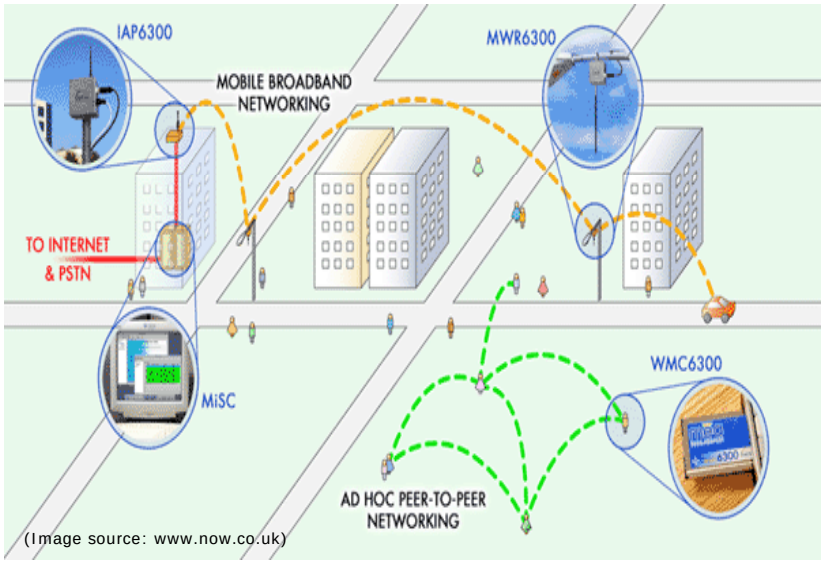
- A wireless network of mobile hosts with no "fixed infrastructure"

Army/DARPA Future Combat System



(Image source: Unknown)

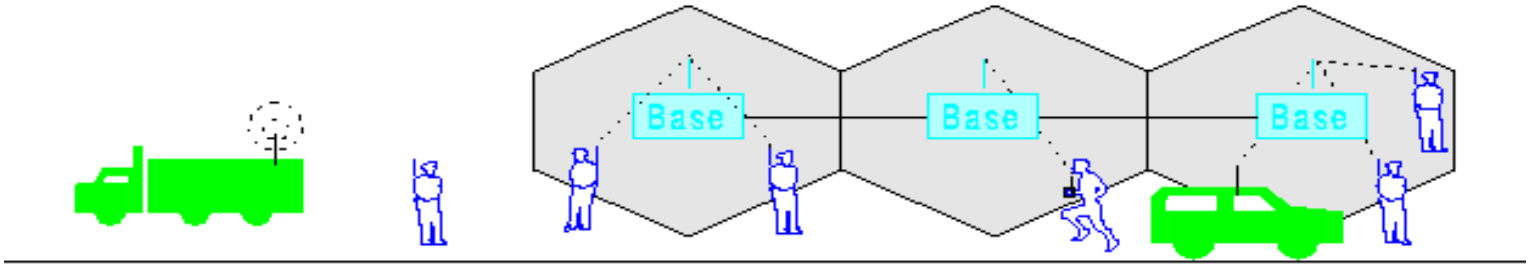
Mesh "Neighborhood" networks



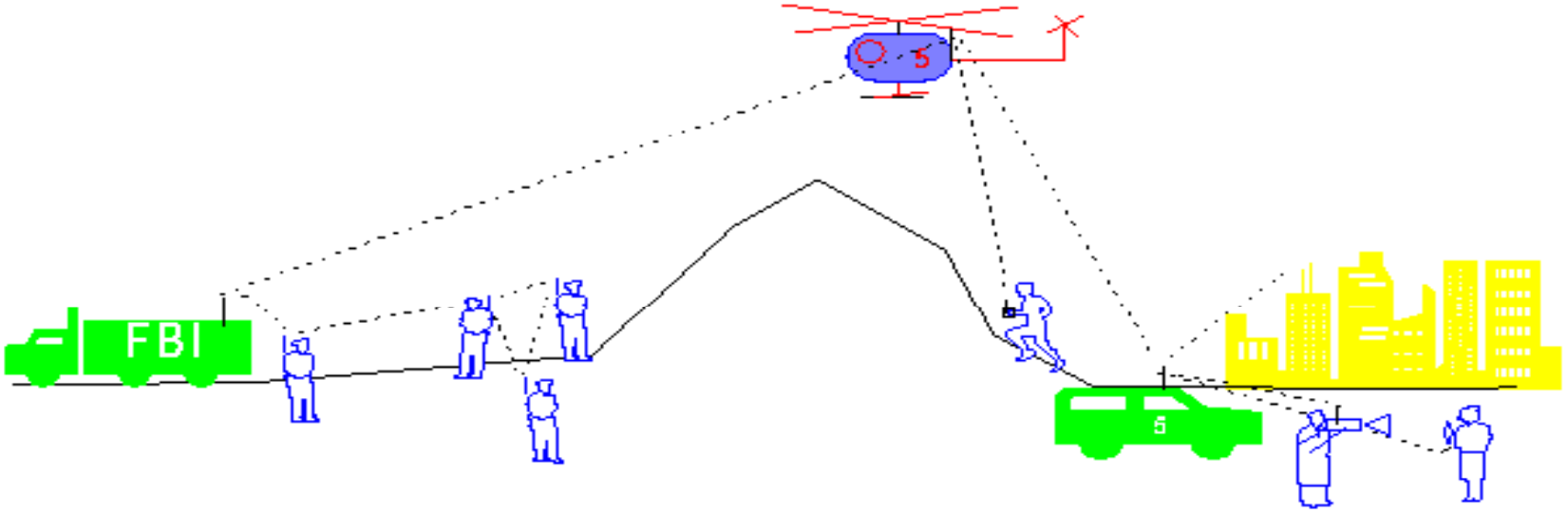
(Image source: www.now.co.uk)

Infrastructure vs Ad-hoc

Infrastructure Network (WiFi or Cellular)



Ad-hoc, Multihop wireless Network



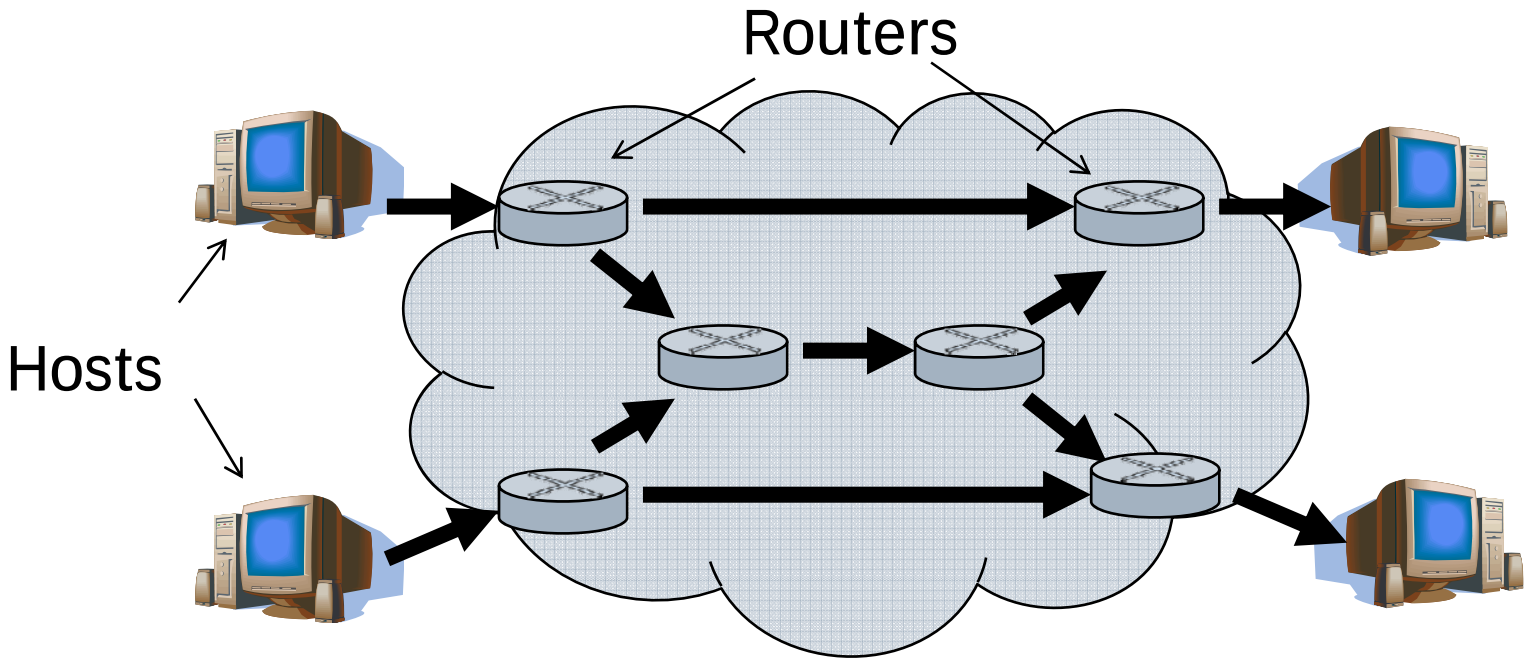
Data dissemination in MANET

- Applications
  - Multimedia data streaming/downloads (e.g., videos, news)
  - Mobile IPTV
  - Delay Tolerant Broadcasting
  - Software updates and patches (e.g., navigation map) in vehicular ad hoc nets
- Challenges
  - Mobility, error-prone channel, limited bandwidth, etc

Outline

- Network coding overview
- NC-based data dissemination in MANET
  - Mobile P2P approach: CodeTorrent
  - Multicasting: CodeCast

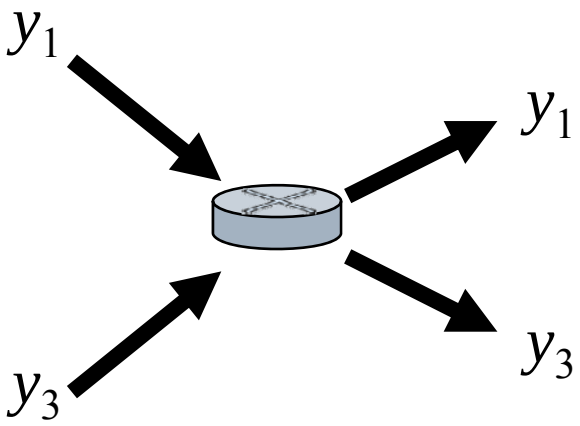
A Computer Network



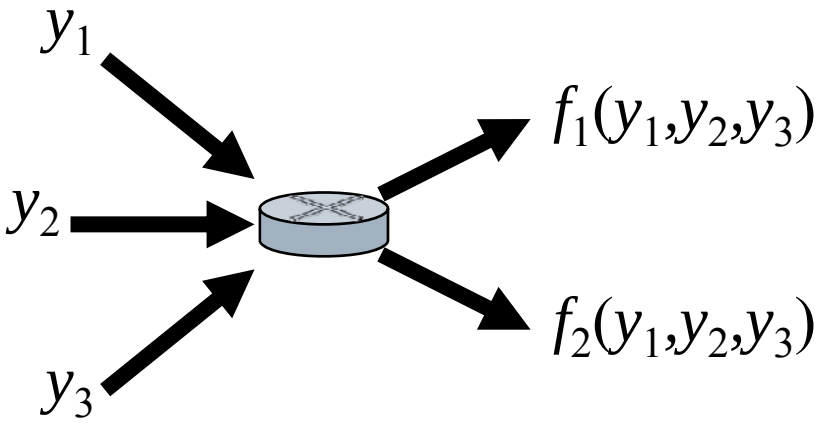
- Hosts are sources/destinations of data
- Routers relay data

Network Coding

- Performing coding on the content of packet on routers



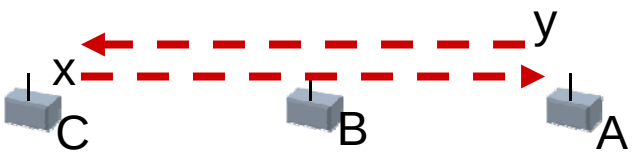
Routing

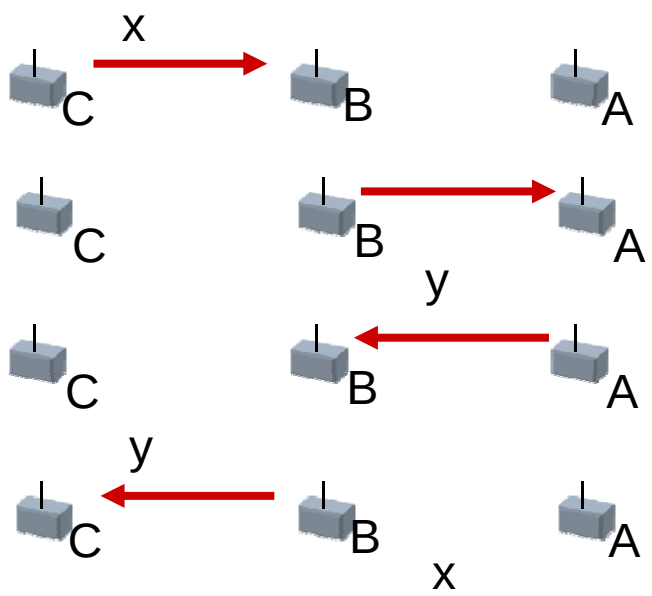


Network Coding

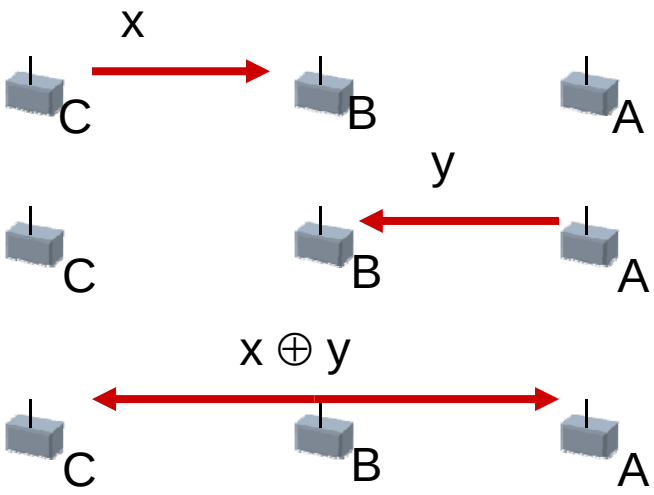
- Increases multicast capacity in wired networks - Alswede et. al, 2000.

Benefit of Network Coding





4 transmissions without NC

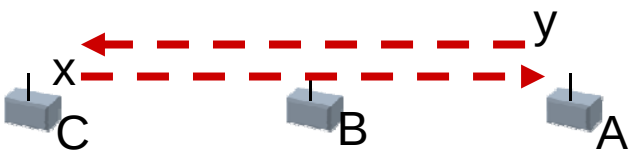


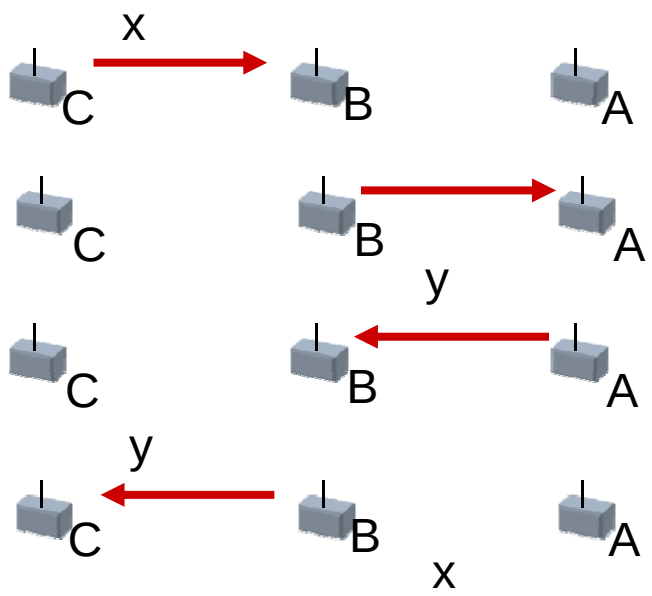
$y = x \oplus (x \oplus y)$

3 transmissions with NC

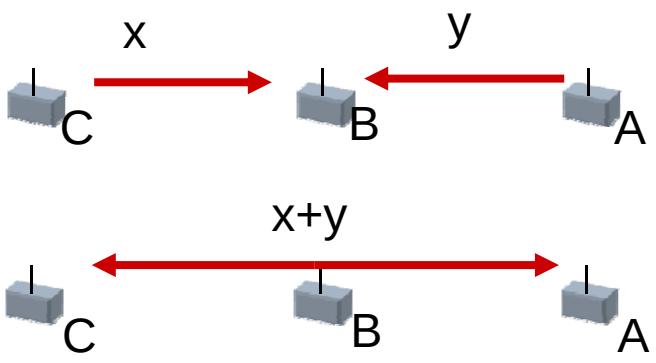
(Katabi'05, Chou'04)

Benefit of Network Coding





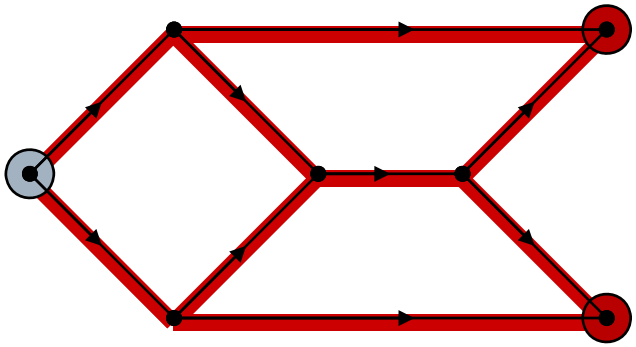
4 transmissions without NC



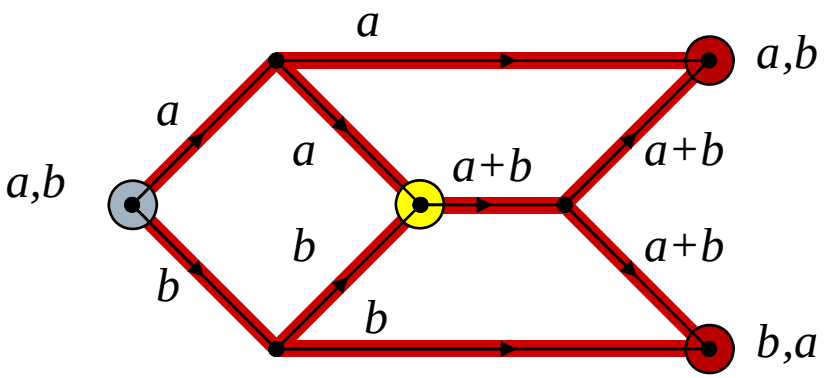
2 transmissions with PNC/ANC

(Zhang'06, Katabi'07)

NC achieves multicast capacity



optimal routing
throughput = 1



network coding
throughput = 2

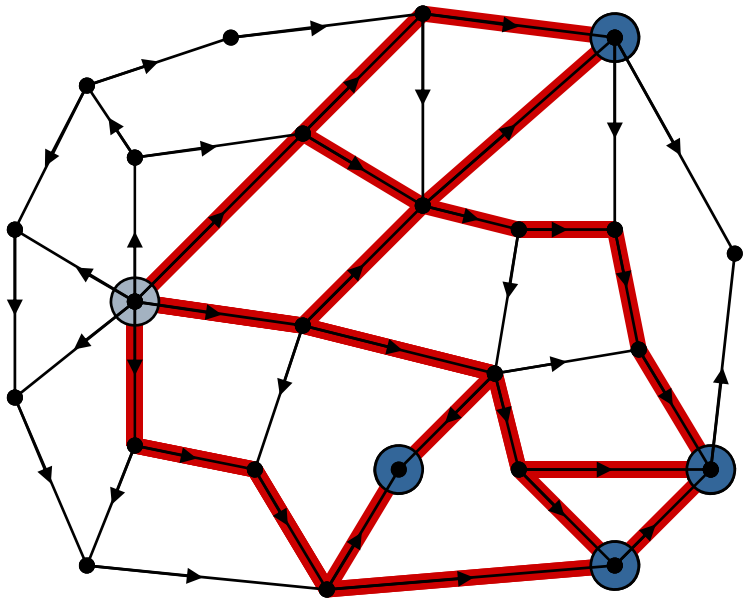
- Alswede, Cai, Li, Yeung (2000):
  - $\min_{t \in T}$ MinCut(s,t) is always achievable by network coding
  - $h = \min_{t \in T}$ MinCut(s,t) is "multicast capacity"

sender

receiver

coding node

How to code?



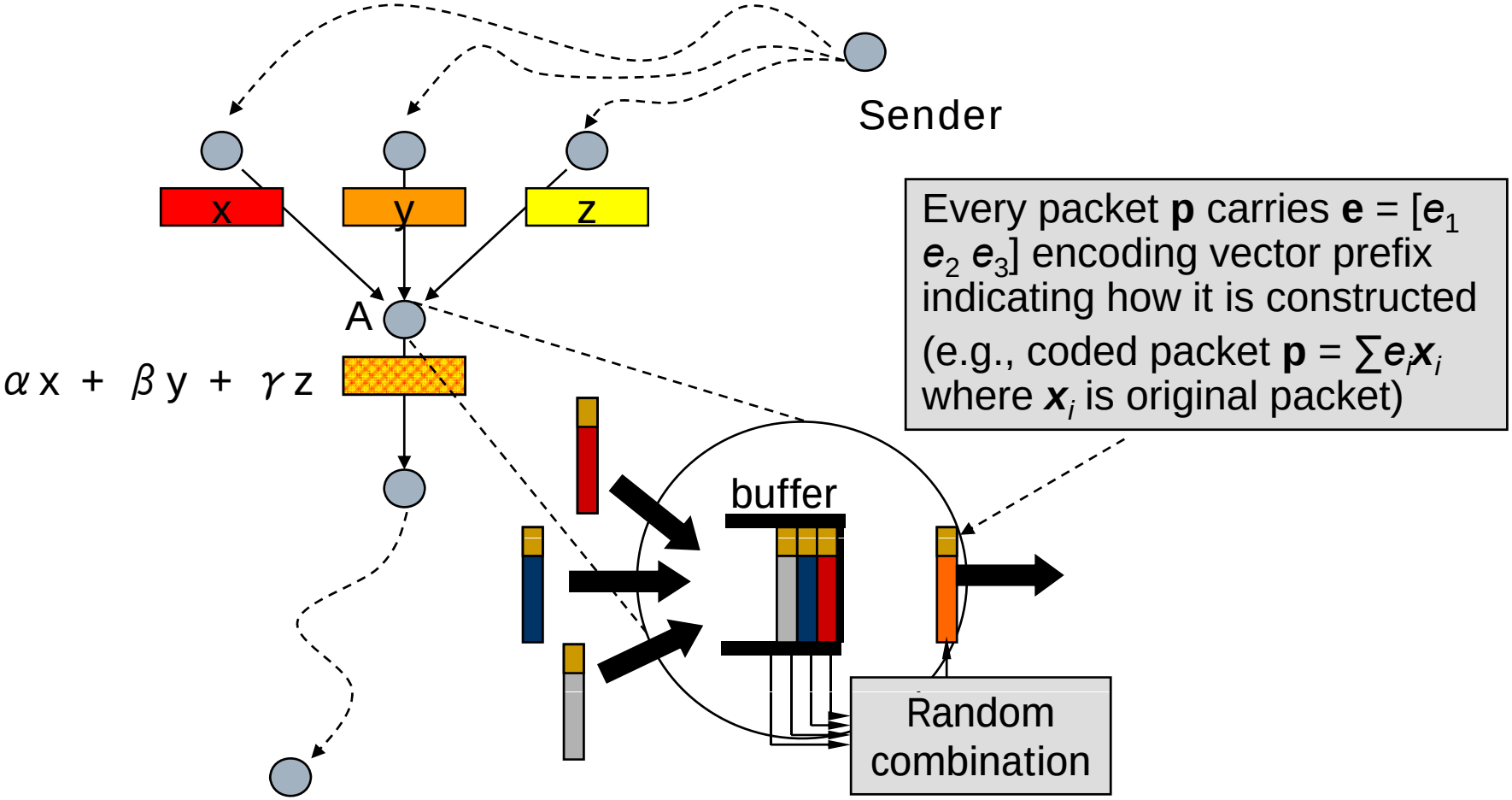
Given:

Directed graph (V,E)

Sender s

Receiver set T
(subset of V)

Random Linear Coding [Chou'03, Ho'03]



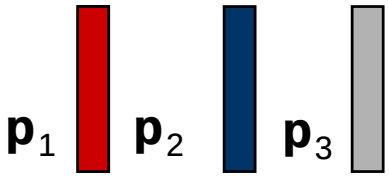
Every packet **p** carries **e** = [e_1 e_2 e_3] encoding vector prefix indicating how it is constructed (e.g., coded packet **p** = $\sum e_i \boldsymbol{x}_i$ where $\boldsymbol{x}_i$ is original packet)

Intermediate nodes randomly mix incoming packets to generate outgoing packets

Preliminaries- Linear Algebra

- Vectors (lowercase boldface)
  - $\mathbf{r}_1 = (1, 2, 3)$, $\mathbf{r}_2 = (1, 2, 1)$
  - Linear combination
    - $a*\mathbf{r}_1 + b*\mathbf{r}_2$ where a and b are scalars
  - Linear independence
    - if there is no pair a and b such that $a*\mathbf{r}_1 + b*\mathbf{r}_2 = 0$ and a are b non-zero
  - Packets represented as vectors
    - 256byte packet = > a set of 256 elements
    - Linear combination of two packets => linear combination of two vectors

Random Linear Coding



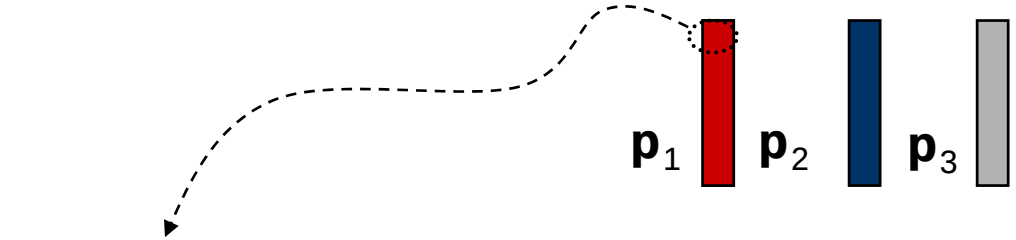
Original Packets

$a*\mathbf{p}_1 + b*\mathbf{p}_2 + c*\mathbf{p}_3 = \mathbf{n}_1$

$g*\mathbf{p}_1 + h*\mathbf{p}_2 + e*\mathbf{p}_3 = \mathbf{n}_3$

$d*\mathbf{p}_1 + e*\mathbf{p}_2 + f*\mathbf{p}_3 = \mathbf{n}_2$

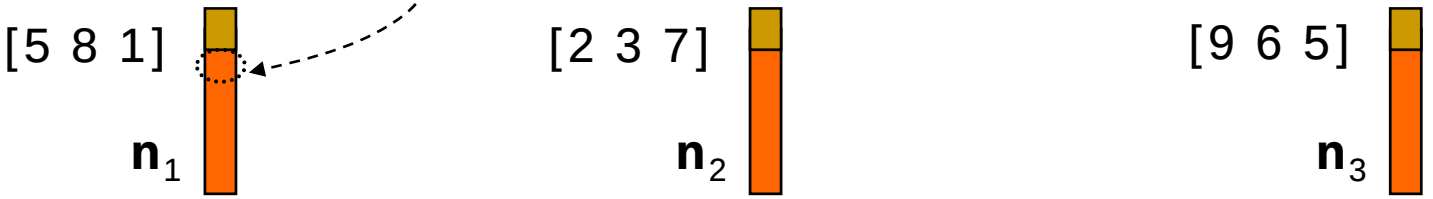
Random Linear Coding (cont.)



$5*\mathbf{p}_1[1] + 8*\mathbf{p}_2[1] + 1*\mathbf{p}_3[1] = \mathbf{n}_1[1]$

$2*\mathbf{p}_1[1] + 3*\mathbf{p}_2[1] + 7*\mathbf{p}_3[1] = \mathbf{n}_2[1]$

$9*\mathbf{p}_1[1] + 6*\mathbf{p}_2[1] + 5*\mathbf{p}_3[1] = \mathbf{n}_3[1]$



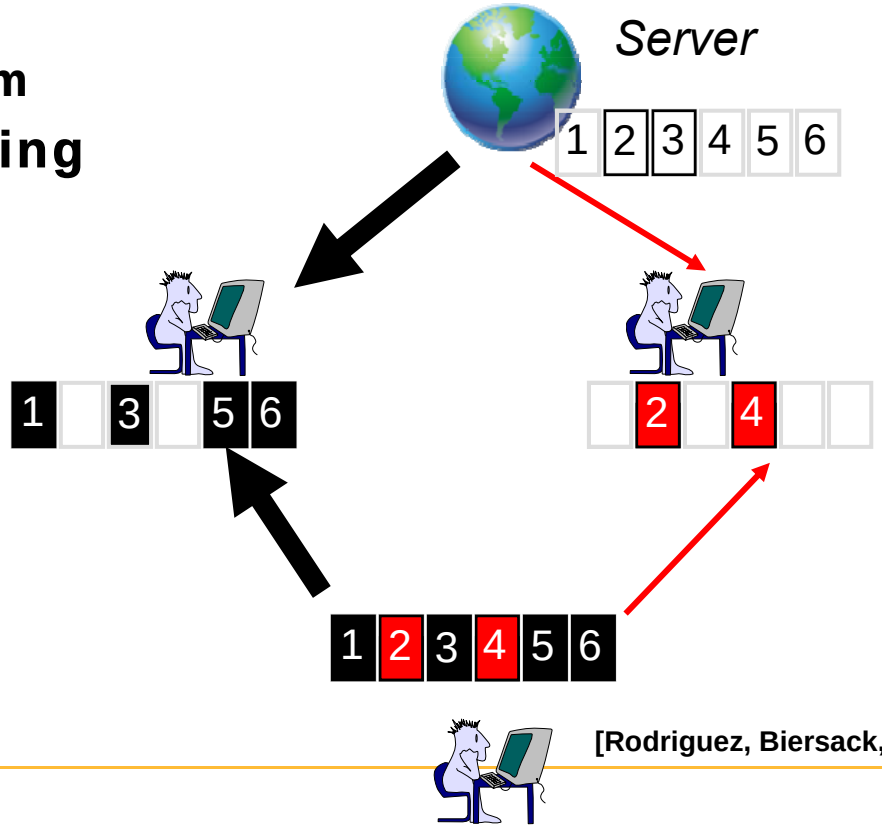
$$\begin{bmatrix} \mathbf{n}1 \\ \mathbf{n}2 \\ \mathbf{n}3 \end{bmatrix} = \begin{bmatrix} 5 & 8 & 1 \\ 2 & 3 & 7 \\ 9 & 6 & 6 \end{bmatrix} \begin{bmatrix} \mathbf{p}1 \\ \mathbf{p}2 \\ \mathbf{p}3 \end{bmatrix}$$

Recover original by matrix inversion

$$\mathbf{P} = \mathbf{H}^{-1}\mathbf{N}$$

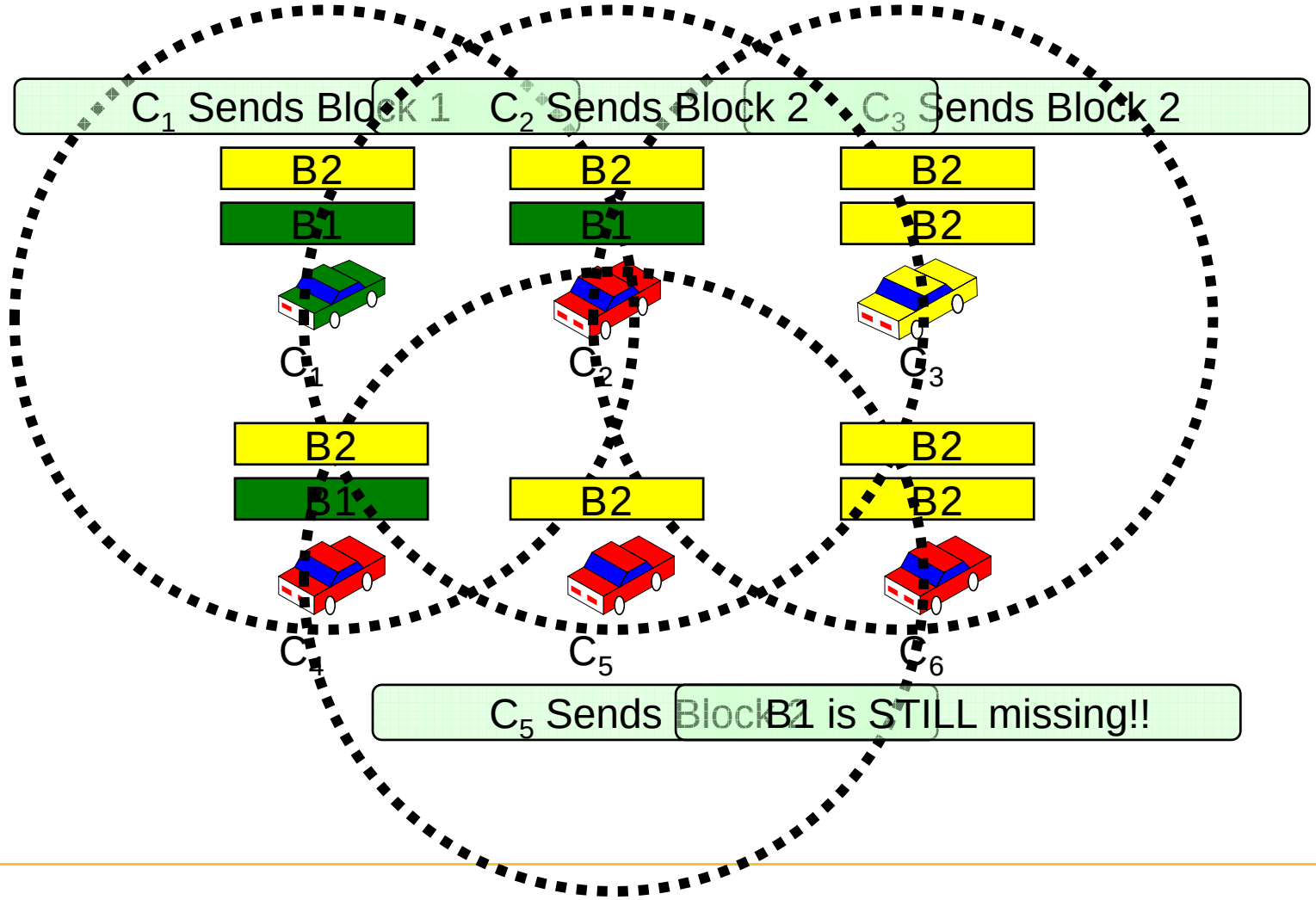
Network Coding in P2P Swarming

- **P2P File Swarming**
  - **File is divided into many small pieces for distribution**
  - **Clients request different pieces from the server / other peers**
  - **When all pieces are downloaded, clients can re-construct the whole file**
  - **Missing coupon problem**
- **P2P using Network Coding**
  - **Avalanche, Infocom'05**

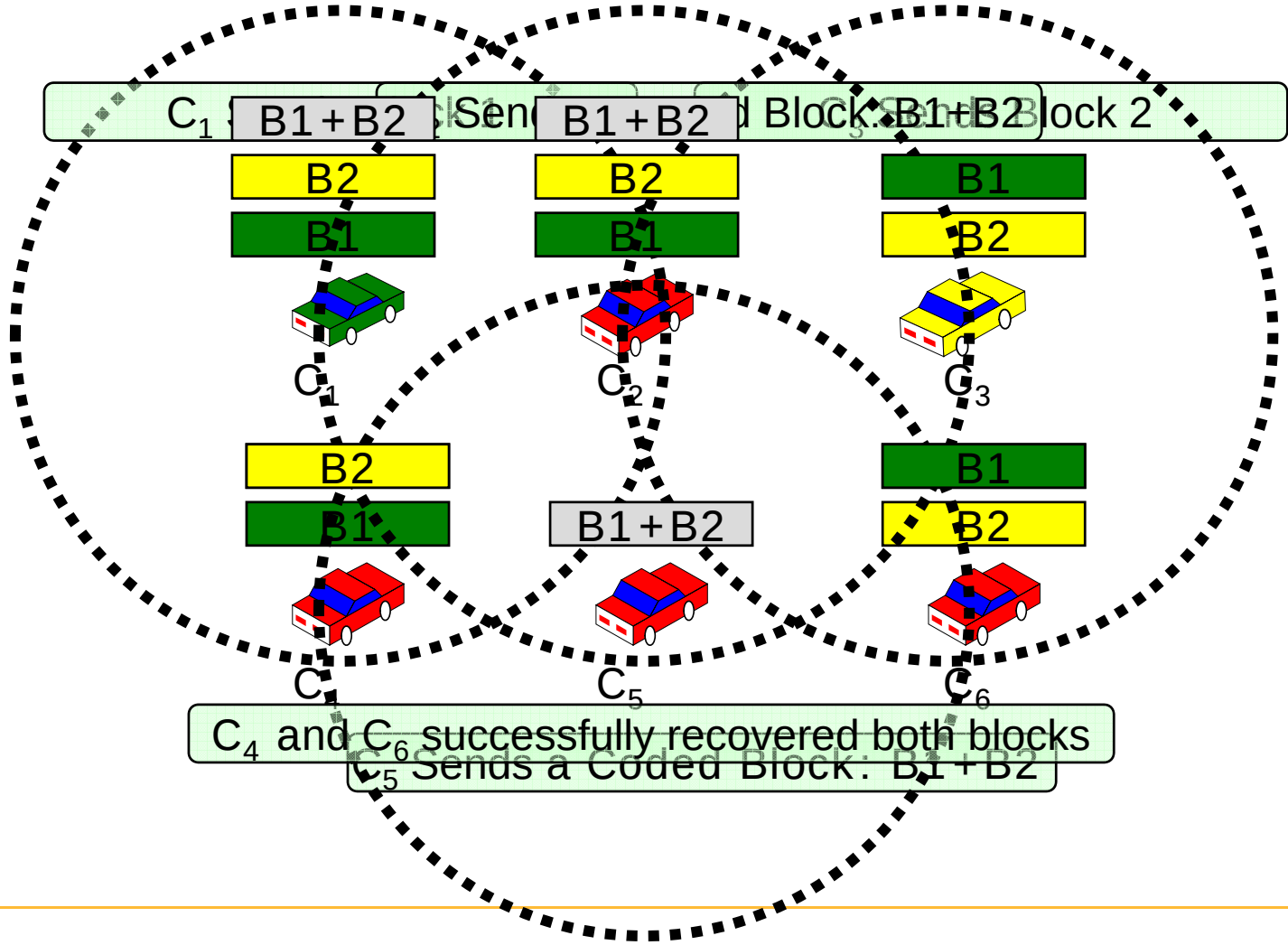


[Rodriguez, Biersack, Infocom'00]

Swarming Limitation: Missing Coupon!

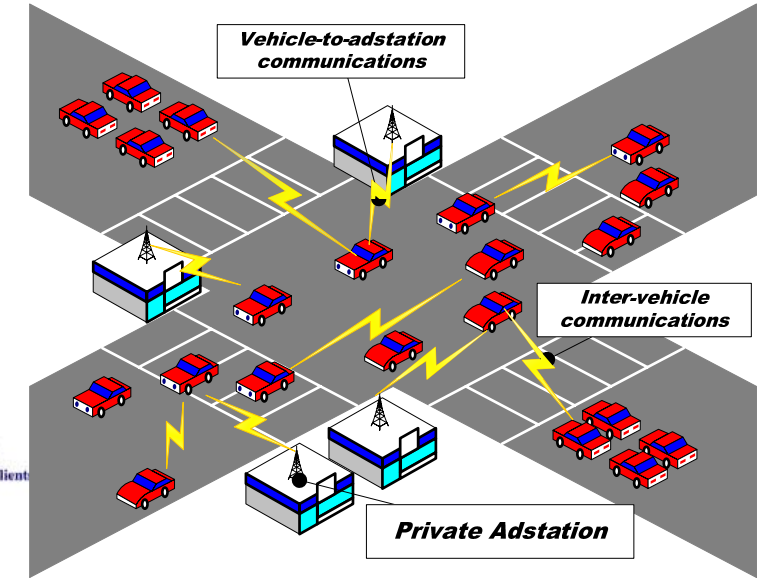


Network Coding Helps Coupon Collection

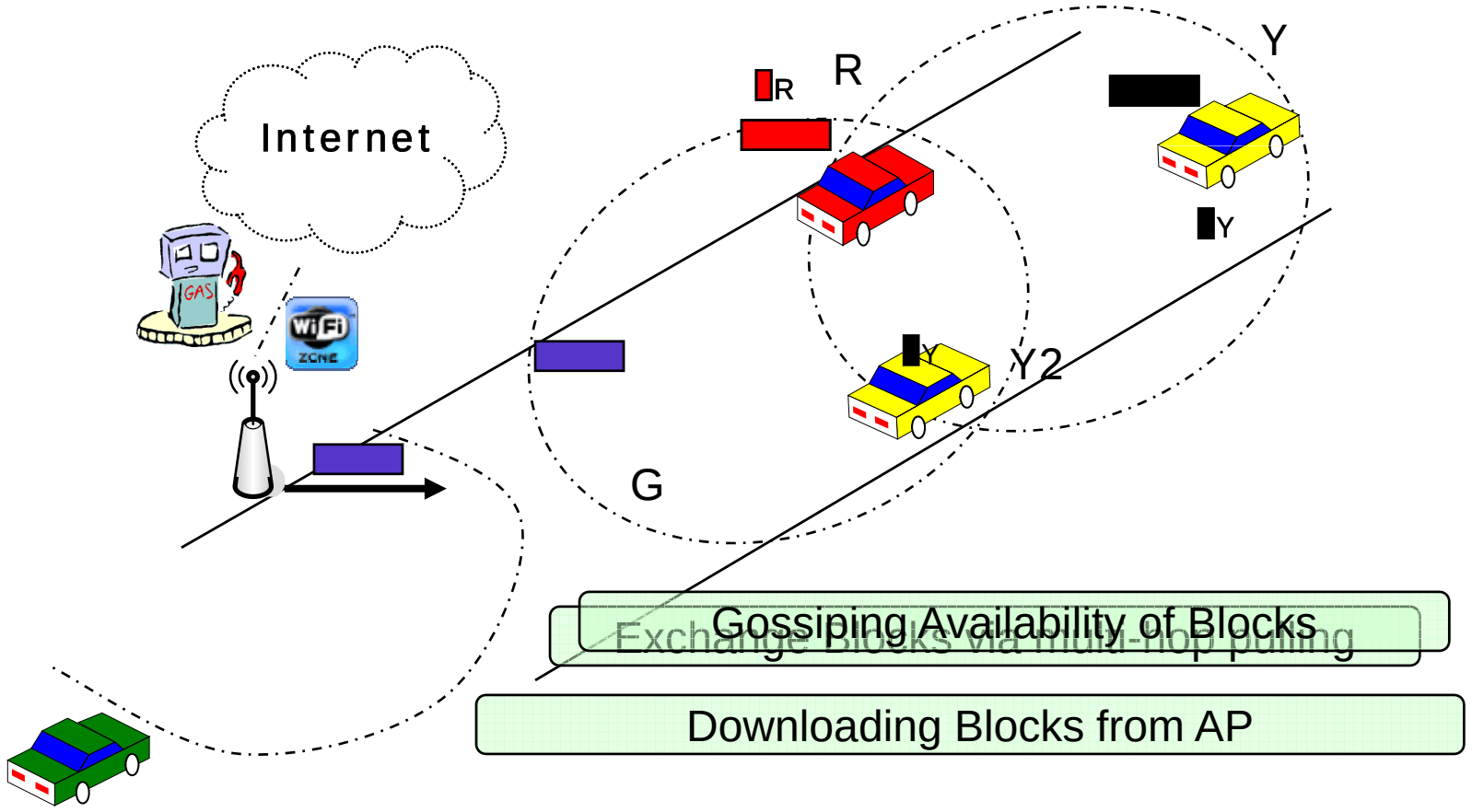


Mobile P2P

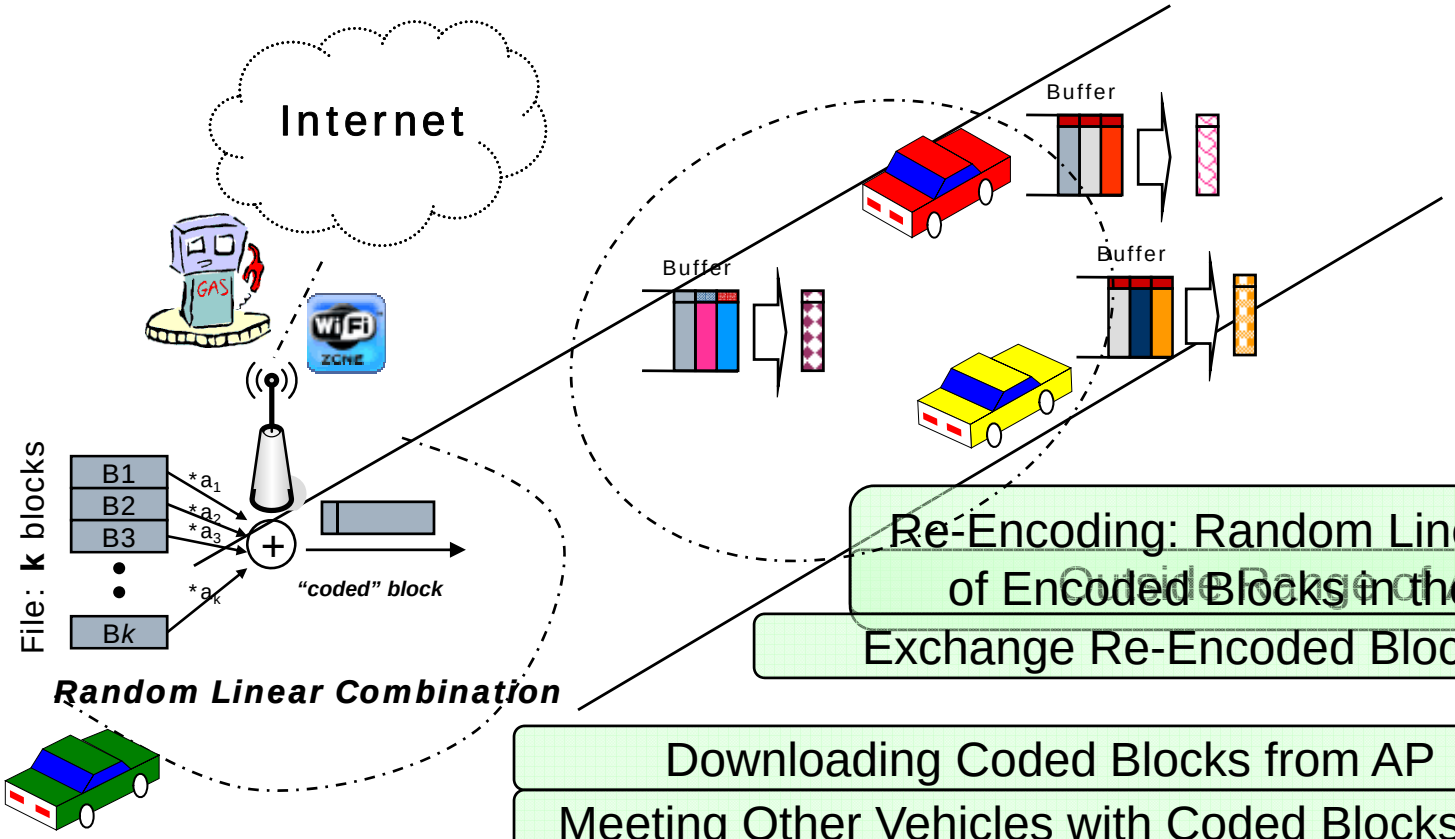
- P2P in mobile wireless networks
  - Fully connected
  - Intermittently connected (opportunistic ad-hoc)
  - Should leverage mobility!



P2P in opportunistic ad-hoc networks: mobility assisted dissemination



CodeTorrent: Basic Idea



Re-Encoding: Random Linear Comb. of Encoded Blocks in the Buffer

Exchange Re-Encoded Blocks

Downloading Coded Blocks from AP

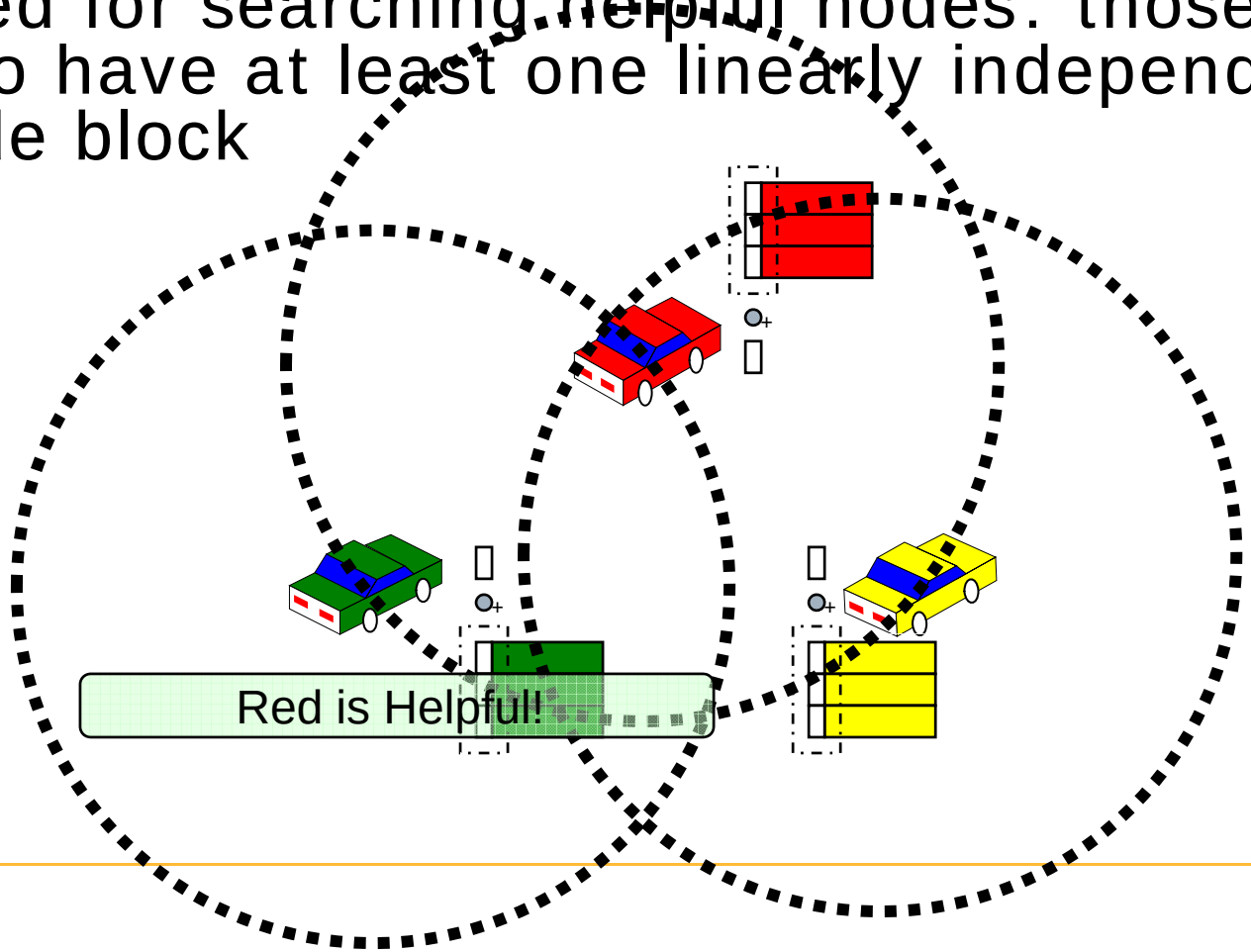
Meeting Other Vehicles with Coded Blocks

Design Rationale

- Single-hop better than multihop
  - Multi-hop data pulling does not perform well in MANET (routing O/H is high)
  - Users in multi-hop may not forward packets not useful to them (lack of incentive)!
- Network coding
  - *Mitigates* rare piece problem
  - *Maximizes* the benefits of overhearing
- Exploits mobility
  - Carry-and-forward coded blocks

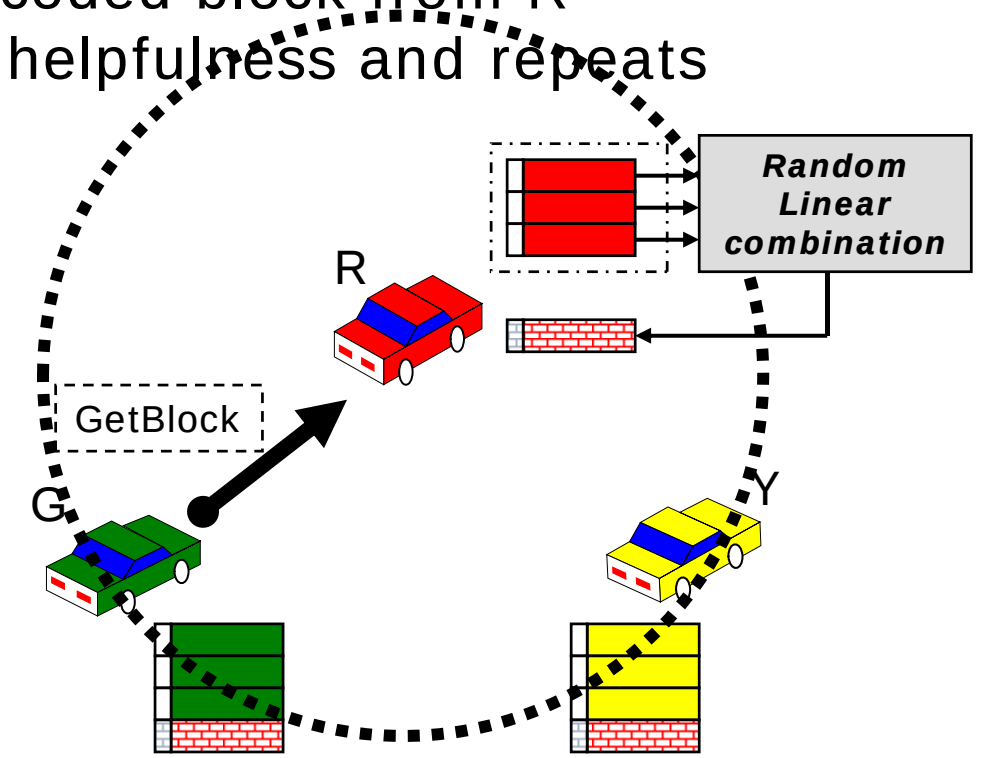
CodeTorrent - Beaconing

- Periodic broadcasting of peer ID and its code vector
- Used for searching helpful nodes: those who have at least one linearly independent code block



CodeTorrent - Single-hop pulling

- A peer pulls coded blocks from the helpful peers
  1. G pulls a coded block from R
  2. G checks helpfulness and repeats

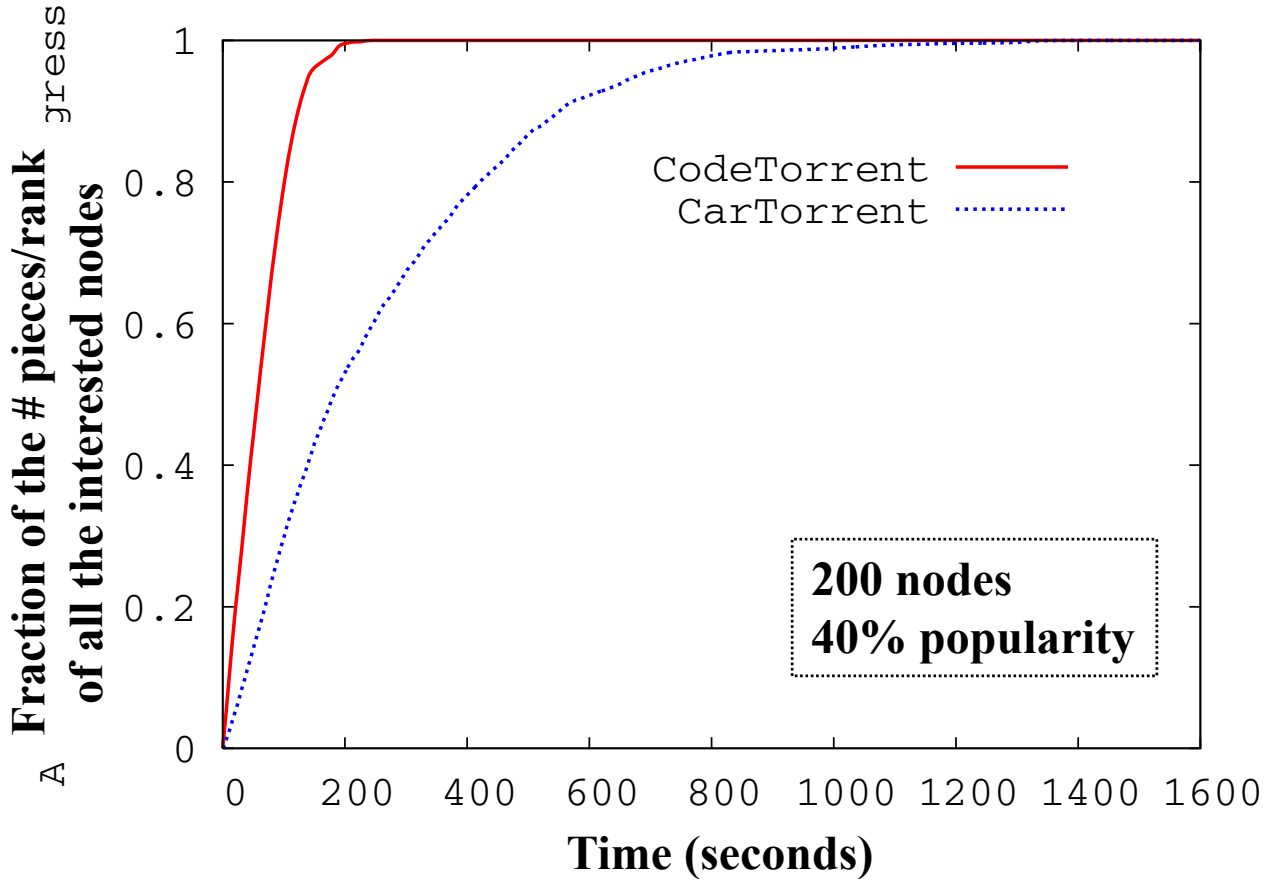


Simulations - Setup

- Qualnet 3.9
- IEEE 802.11b / 2Mbps
- Terrain: 2.4x2.4 km^2
- Distributing 1MB file
  - 4KB/block * 250 blocks
  - 1KB per packet
- Randomly located 3 Aps
- Comparing *CarTorrent* (w/ AODV) with *CodeTorrent*
  - AODV w/ net-diameter 3 hops
  - *CodeTorrent* with GF(256)

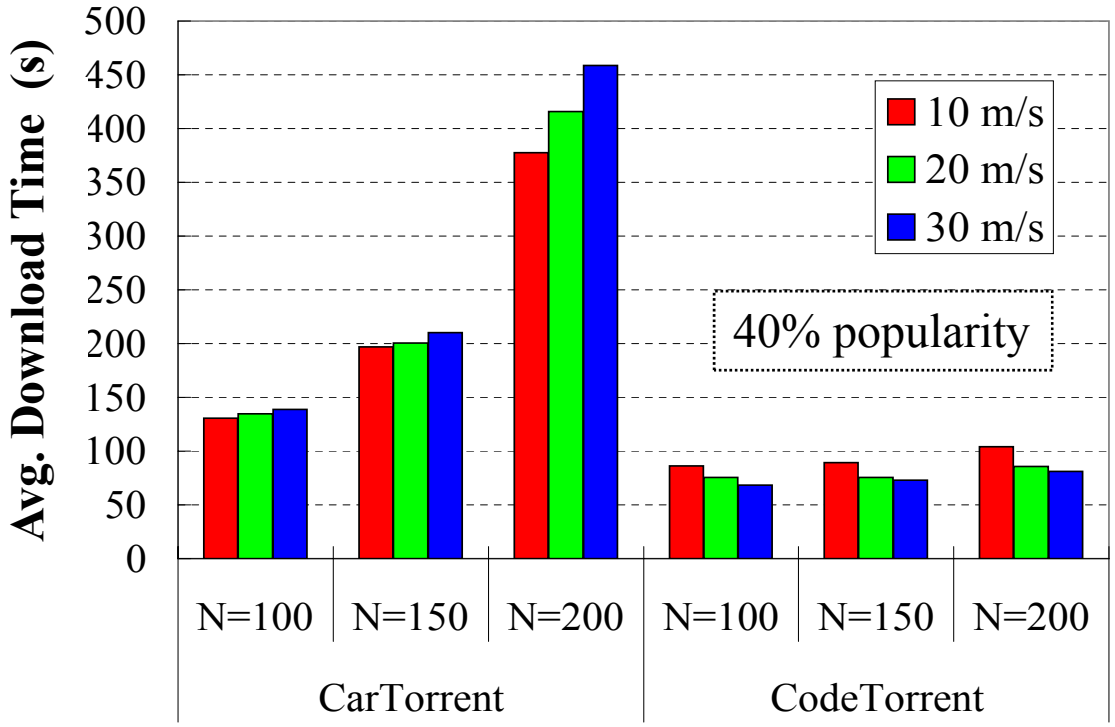
Simulation Results

- Overall downloading progress



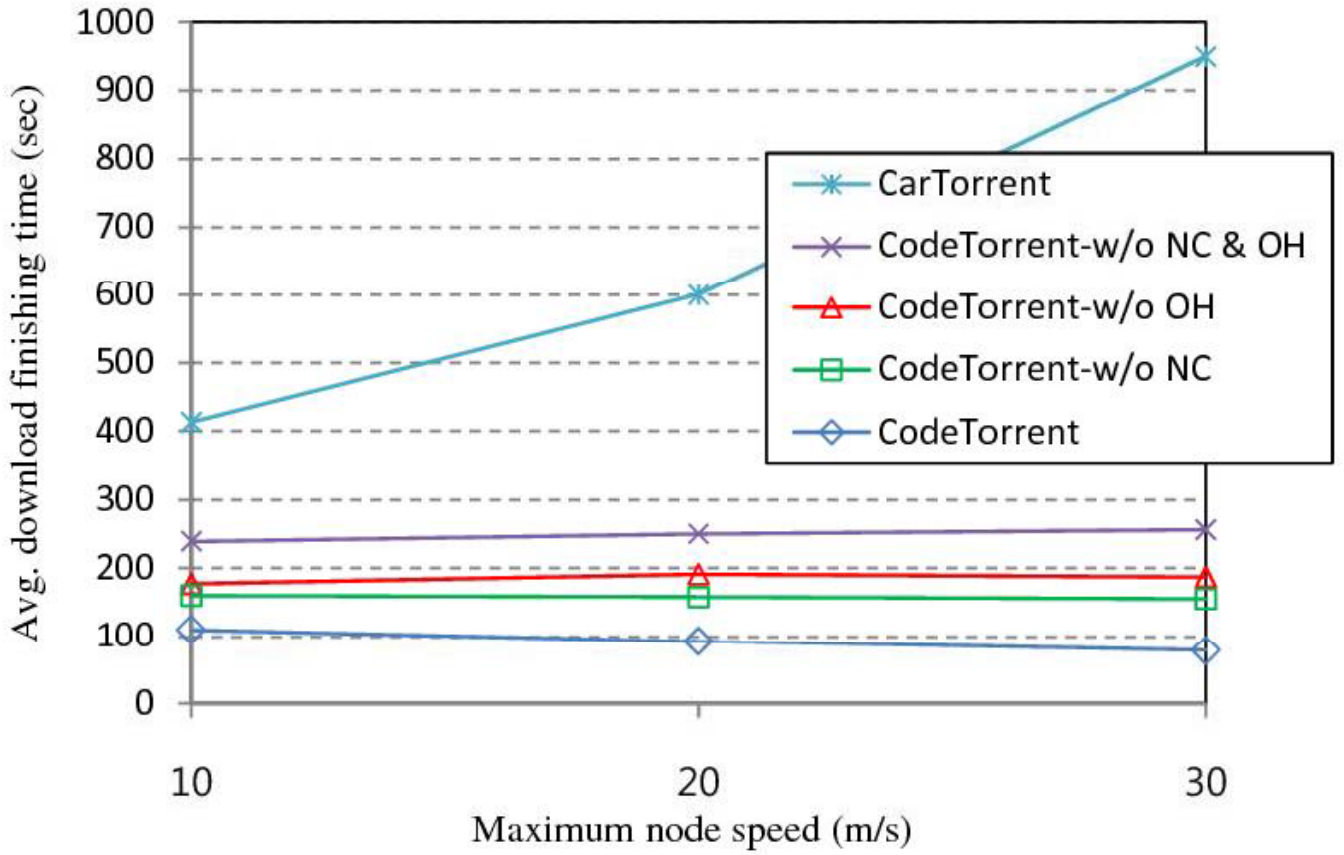
Simulation Results (2)

- Impact of mobility
  - Speed helps disseminate from AP's and C2C
  - Speed hurts multihop routing (*CarT*)
  - Car density + multihop promotes congestion (*CarT*)



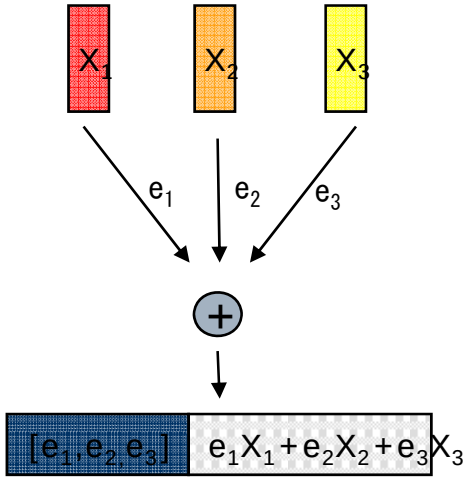
Simulation Results (3)

- Where does performance gain come from?



Overhead Issues

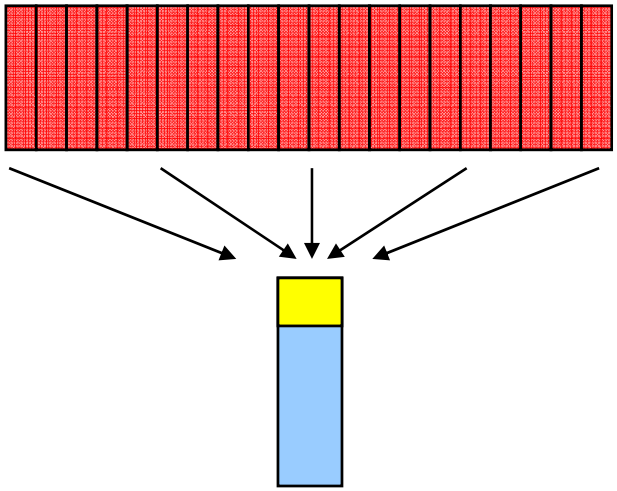
- Packet header
  - Each packet carries encoding vector and its size scales with # of pieces
  - if file size = 1GB & packet size = 1KB, 1M pieces exist and thus encoding vector size = 1MB, too large!
- Computation Overheard
  - Decoding takes $O(n^3)$



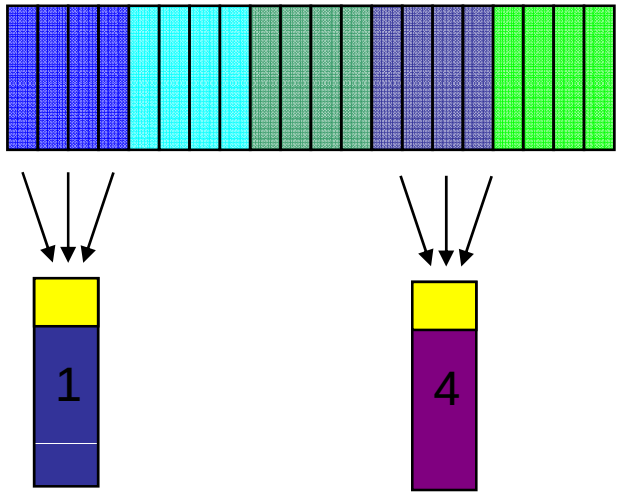
Reducing overhead

- Divide file into “generations”
  - Pieces only in the same generation can be combined
  - Coupon collection problem again!

50MB one generation

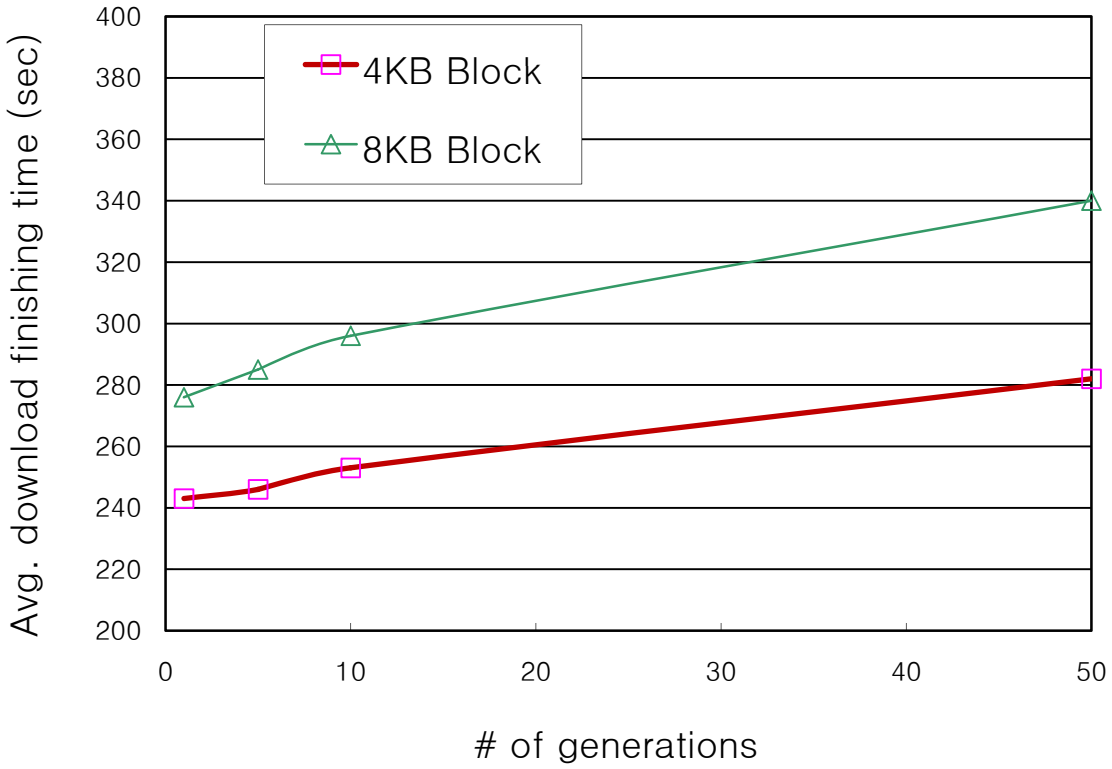


5 generations of 10MB



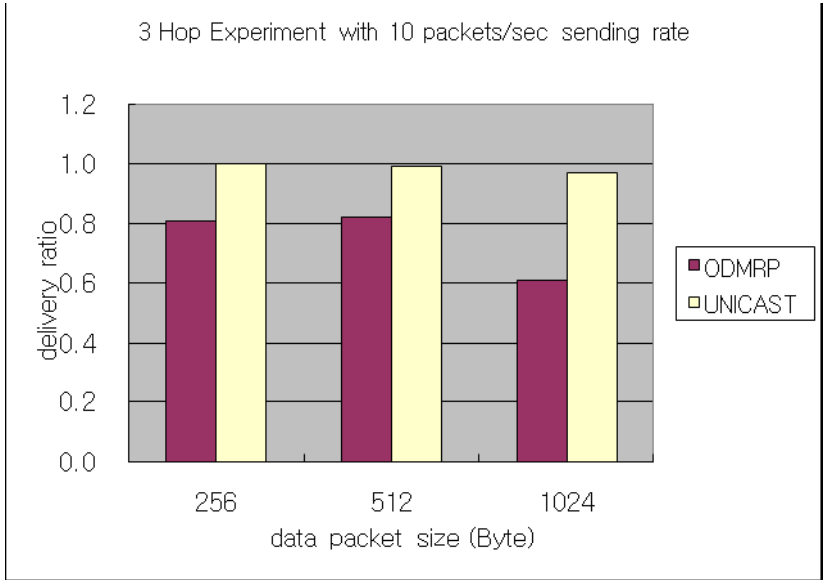
Simulation Results (4)

- Impact of generation size



Multicast in Real Testbed

- Very unreliable!
  - ODMRP + 802.11
  - ≈ 80% delivery (2-hop)
  - ≈ 60% delivery (3-hop)



Robust Multicast using NC

- In MANET one must consider:
  - Random errors; External interference /jamming
  - Motion; path breakage
- Target application:
  - Multicast (buffered) video streaming
    - Some loss tolerance
    - Some delay tolerance (store & playback at destination) - non interactive

Problem Statement

- Multicast streaming in mobile wireless networks is non-trivial
  - Streaming requires: high reliability (but not 100%), low delay (but not 0)
  - But network is: unreliable, bandwidth-limited

- Major concern: packet drops
  - Lossy wireless channel (uncorrelated, random like errors)
  - Route breakage due to mobility, congestion, etc (correlated errors)

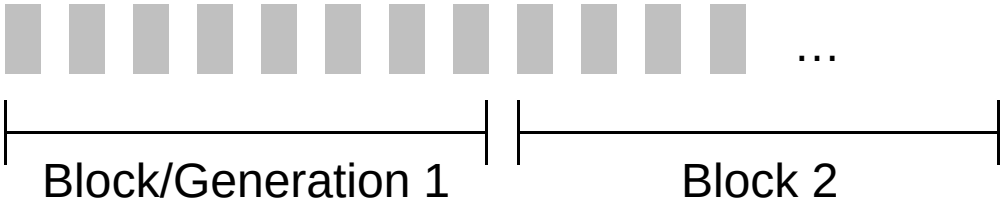
Conventional vs NC Multicast

- Conventional Approaches
  - Time diversity => O/H, delay?
    - Recovery scheme a la ARQ (Reliable Multicast)
    - (End-to-end) Coding (FEC, MDC, ...)
  - Multipath diversity (ODMRP, ...) => O/H?

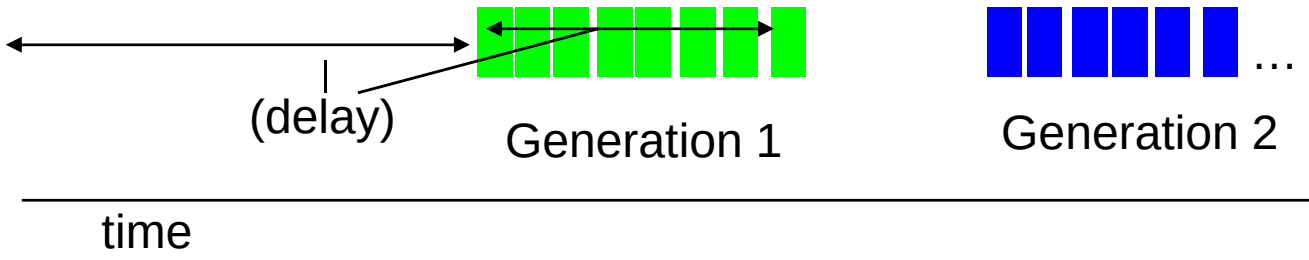
- NC Approach
  - Main ingredient: Random Linear Coding (RLC)
  - Exploit time and multipath diversity
  - Controlled-loss (near 100%), bounded-delay (hundreds of ms)
  - Suitable for buffered streaming
  - Real time version (tens of ms delay bound) exists

Coding at sender

An application generates a stream of frames



Network layer generates stream of coded packets



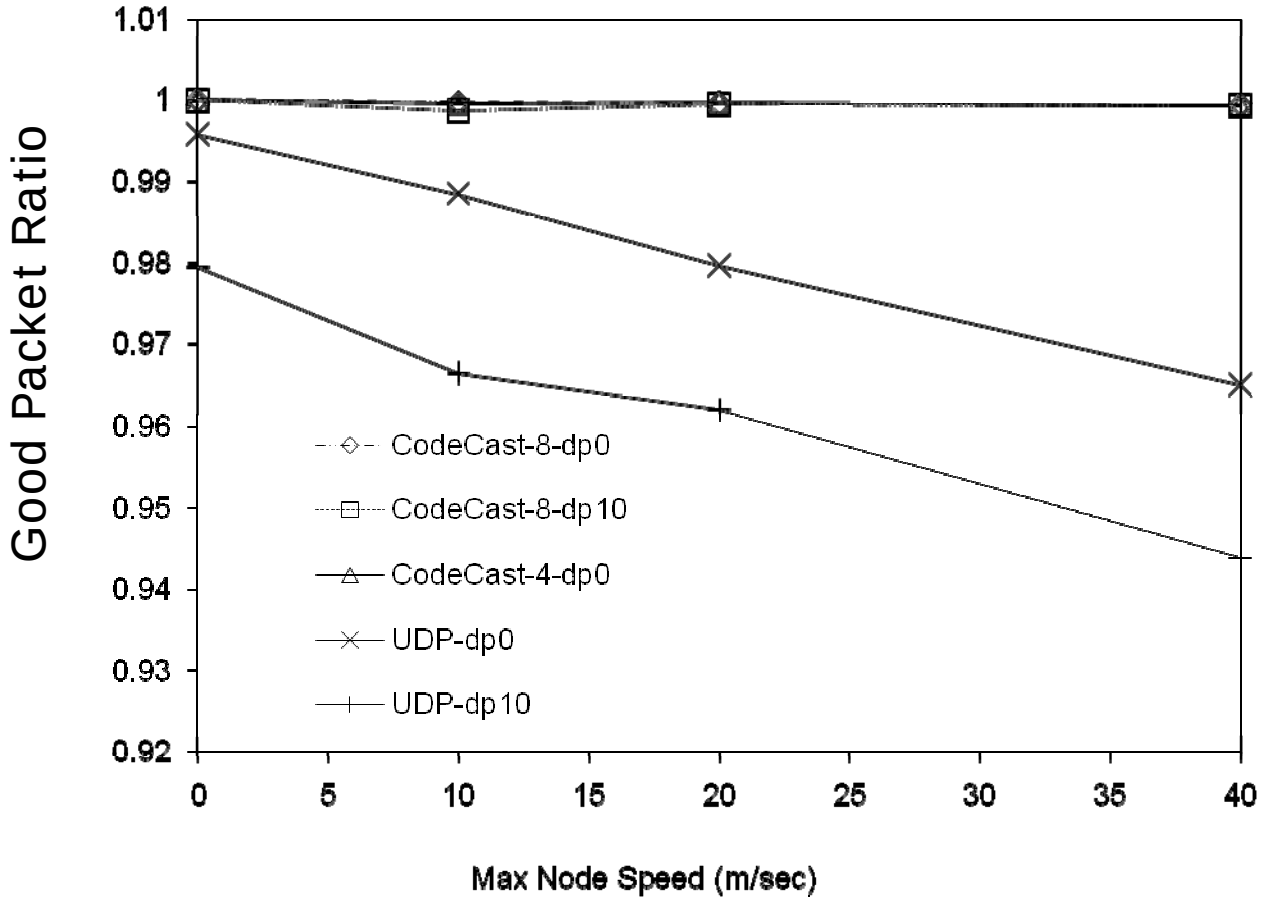
A random linear combination of Block 1 frames

A random linear combination of Block 2 frames

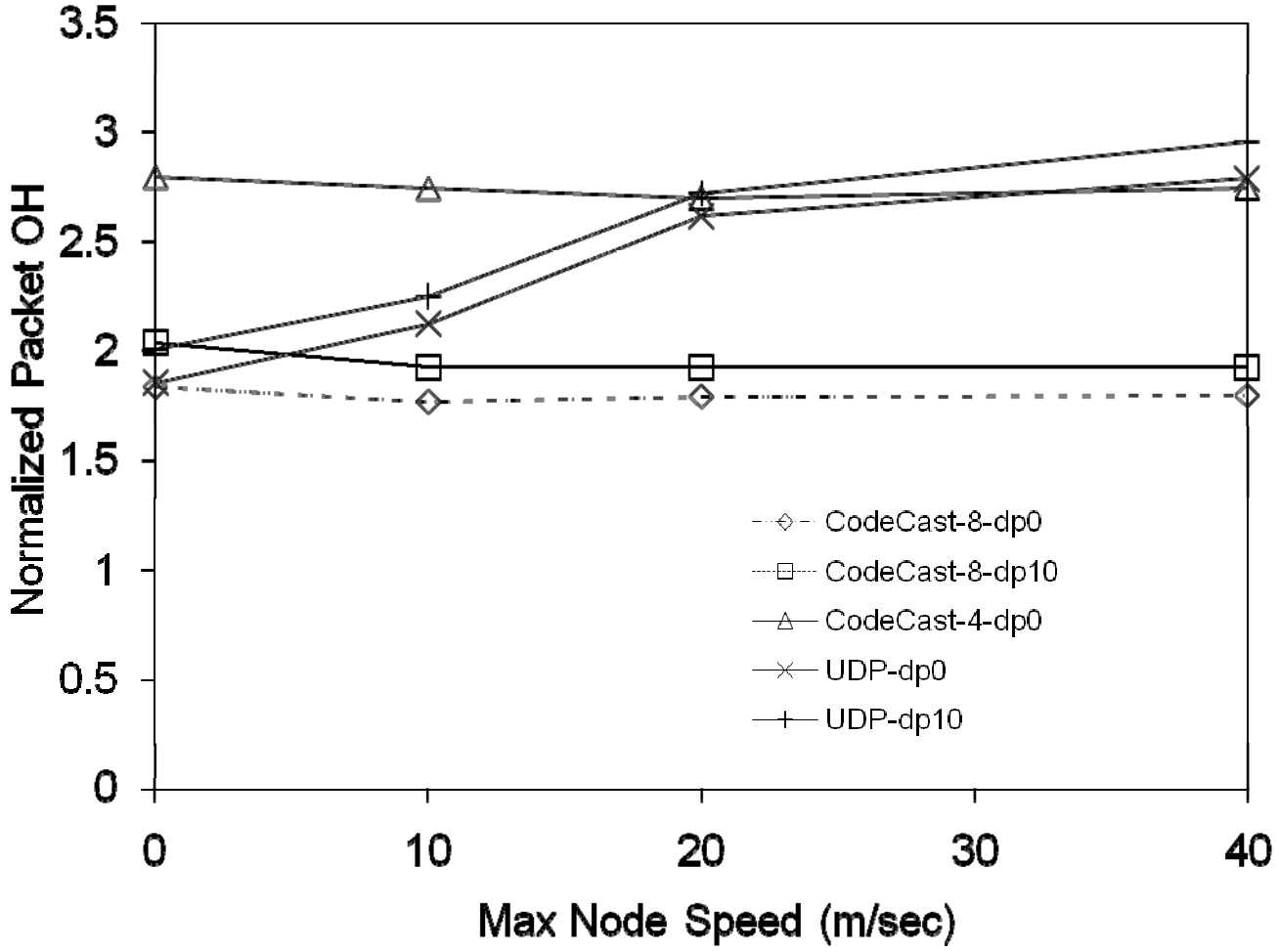
Simulation

- Settings
  - QualNet
  - 100 nodes on 1500 x 1500 m^2
  - 5 Kbytes/sec traffic (512B packet)
  - Random Waypoint Mobility varying maximum speed unless
- Metrics
  - Good packet ratios: num. of data packets received within deadline (1sec) vs. total num. of data packets generated
  - Normalized packet O/H: total no. of packets generated vs no. of data packet received
  - Delay: packet delivery time from the source application to receiver applications

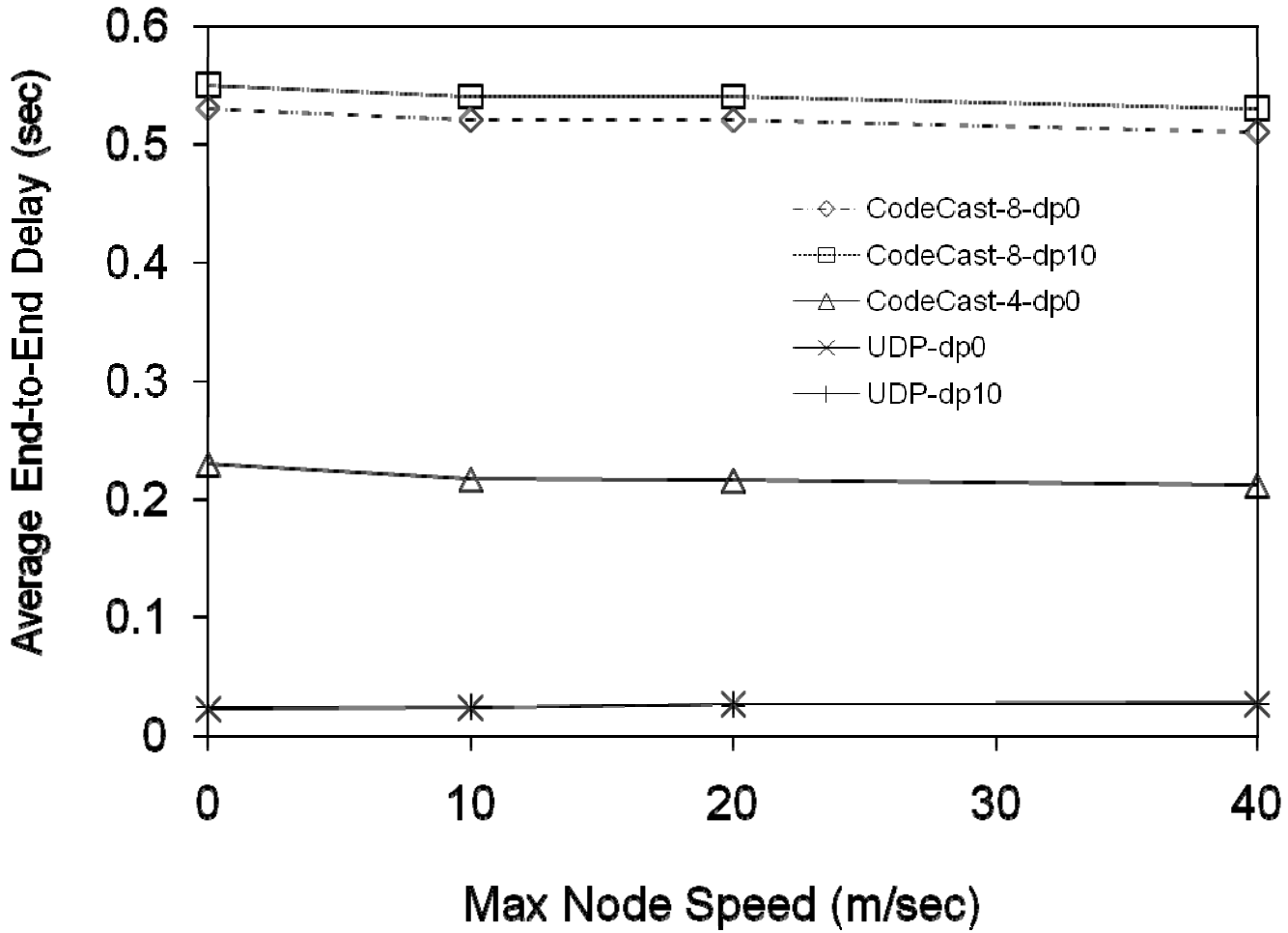
ODMRP vs NC: Reliability



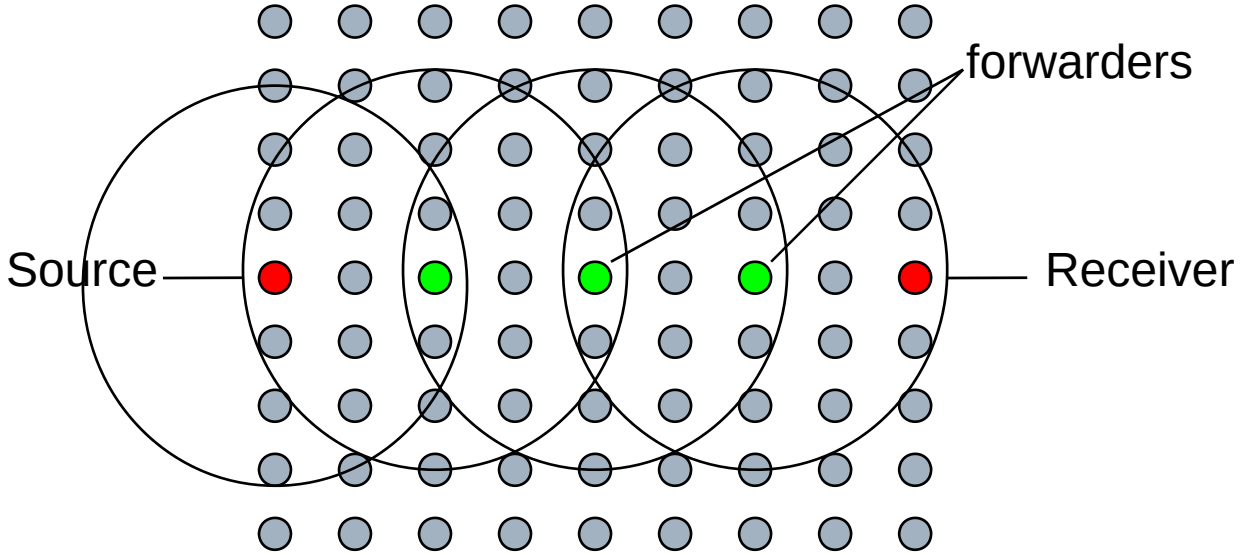
ODMRP vs NC: Efficiency



ODMRP vs NC: Delay

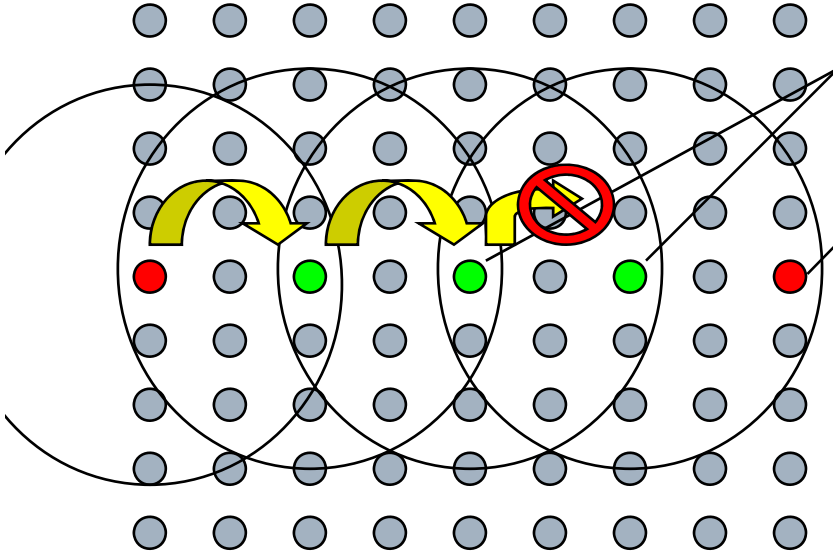


Forwarding – Conventional Routing



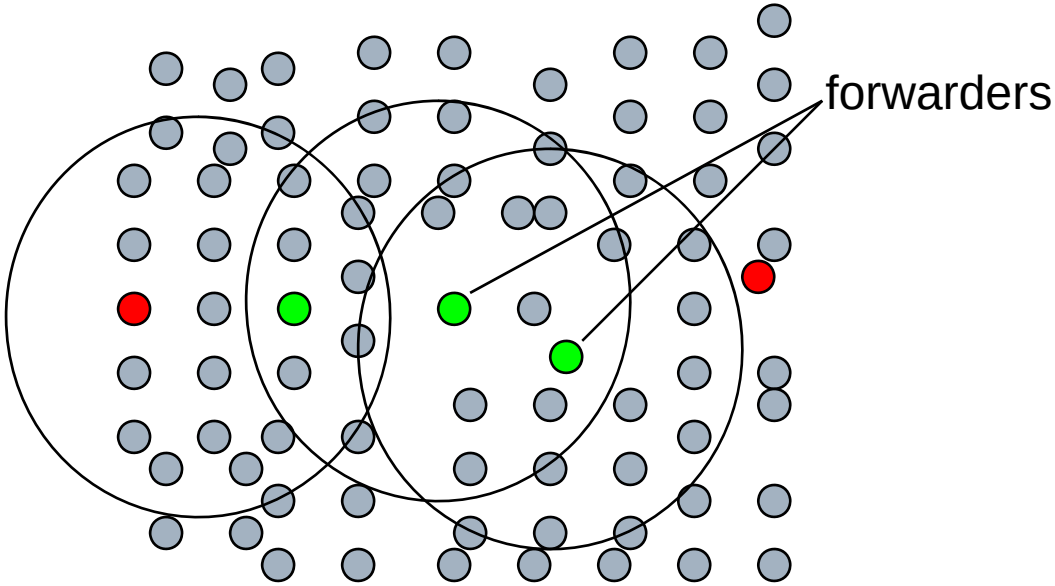
Select least number of nodes as forwarders to form a path b/w a S-R pair and each forwarder transmits each packet once

Problem – Conventional Routing

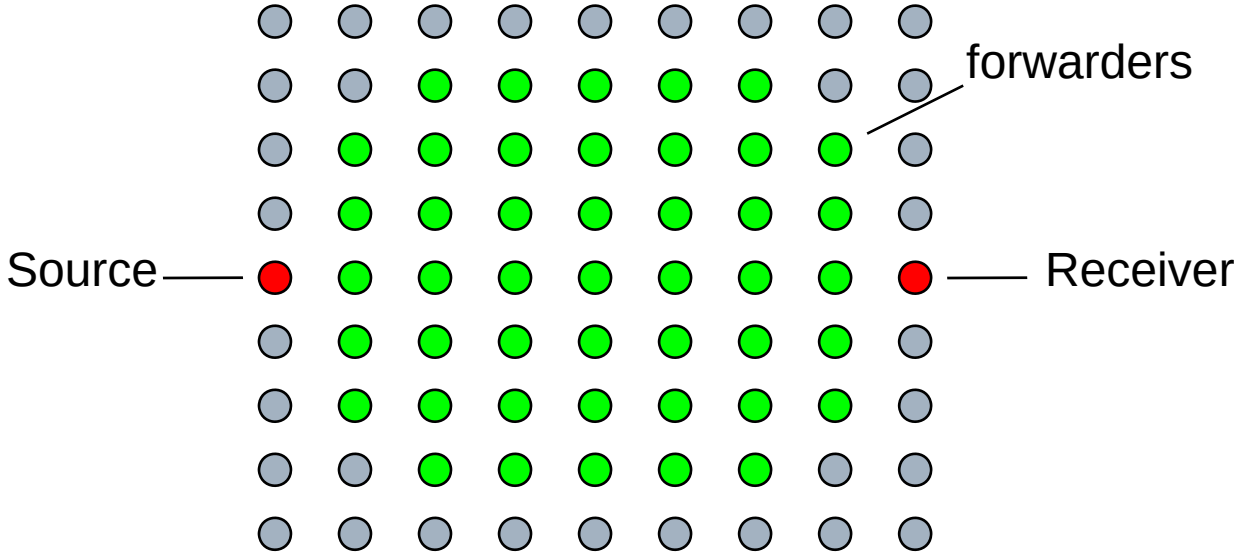


What if random error occurs?

What if route breaks?



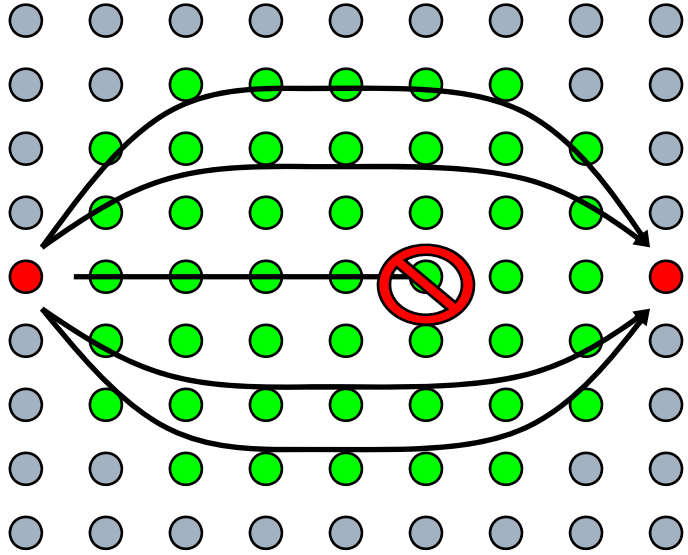
Forwarding – NC approach



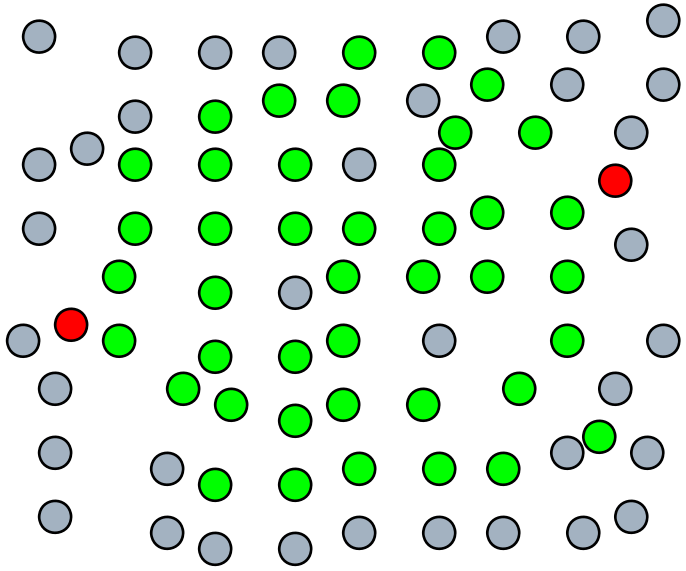
Select most nodes in between a S-R pair as forwarders and each forwarder transmits one packet per generation once (what about packet storm problem?)

A node becomes a forwarder if (hop count to Source + hop count to Receiver) is less than hop distance of S-R pair

Robustness of NC approach

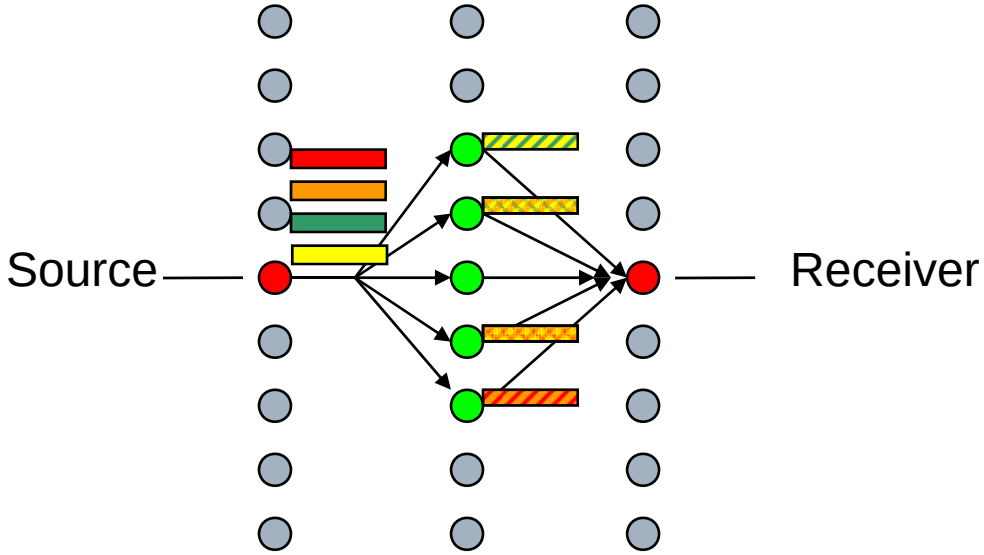


Robust to random errors



Robust to mobility

NC suppresses packet storms



Setting up a multicast mesh

- Every coded packet carries in the header three more fields, *vldd*, *dist*, and *nust*
- The one-bit field *vldd* is set if either the sender is a multicast receiver or has received a previous block packet with *vldd* bit set from one of the sender's downstream nodes
- A node considers a neighboring node to be downstream if the neighboring node transmits a packet with a larger *dist* value than the dist value the node maintains
- Each node maintains as a local variable *dist*, indicating the hop distance from the multicast data source and copies its value to every code packet the node transmits.
- Every time a node transmits a coded packet, *dist* is recalculated as one plus the biggest *dist* value found in the headers of the packets which are combined to yield the coded packet
- Conversely, a node considers a neighboring node to be a upstream node if the neighboring node transmits a coded packet in a new block or a smaller *dist* value
- Each node also maintains *nust*, indicating the number of upstream nodes as a local variable and records its value in the header of every packet the node transmits
- A node broadcasts to the neighborhood *r* coded packets

Thank you!