Course: Wireless Networks I
Topic: Analysis of MANETs and Network Coding

Ubinet Master 2011-2012

L. Sassatelli
sassatelli@i3s.unice.fr

Outline

I. Delay-Throughput trade-offs:
   1. Static ad hoc networks under the physical and protocols models
   2. Mobile ad hoc networks
   3. Distributed MIMO: nodes as antenna arrays

II. Network coding:
   1. Introduction
   2. Application: COPE
   3. Application: Routing and energy efficiency
   4. Application: NC meets TCP

Limitation of cellular networks

Customers Angered as iPhones Overload AT&T

By JENNA WORTHAM
Published: September 2, 2009

Slim and sleek as it is, the iPhone is really the Hummer of cellphones.

Enlarge This Image

Michael Appleton for The New York Times
AT&T monitors its network from its operations center in Bedminster, N.J.
More Photos »

It's a data guzzler. Owners use them like minicomputers, which they are, and use them a lot. Not only do iPhone owners download applications, stream music and videos and browse the Web at higher rates than the average smartphone user, but the average iPhone owner can also use 10 times the network capacity used by the average smartphone user.

TWITTER
LINKEDIN
COMMENTS (322)
SIGN IN TO E-MAIL
PRINT
REPRINTS
SHARE

MARTHA MARCY MAY MARLENE
NOW PLAYING

"They don't even realize how much data they're using," said Gene Munster, a senior securities analyst with Piper Jaffray.

The result is dropped calls, spotty service, delayed text and voice messages and glacial download speeds as AT&T's cellular network strains to meet the demand. Another result is outraged customers.

Cellphone owners using other carriers may gloat now, but the problems of AT&T and the iPhone portend their future. Other networks could be stressed as well as more sophisticated phones encouraging such intense use become popular, analysts say.

Multimedia

Slide Show
AT&T Races to Expand the Network

Related

Bits: Big City, Big iPhone Troubles (September 3, 2009)

"It's almost worthless to try and get on 3G during peak times in those cities," Mr. Munster said, referring to the 3G network. "When too many users get in the area, the call drops." The problems seem particularly pronounced in New York and San Francisco, where Mr. Munster estimates AT&T's network shoulders as much as 20 percent of all the iPhone users in the United States.

AT&T says that the majority of the nearly $18 billion it will spend this year on its networks will be diverted into upgrades and expansions to meet the surging demands on the 3G network. The company intends to erect an additional 2,100 cell towers to fill out patchy coverage, upgrade existing cell sites by adding fiber optic connectivity to deliver data faster and add other technology to provide stronger cell signals.

VentureBeat Profiles | Events | Jobs | Newsletters | Entrepreneur Corner | Videos | Mec

MAIN | MOBILEBEAT | GREENBEAT | GAMESBEAT | DEALSBEAT | DEMOBEAT

iPhone users eating up AT&T's network

May 11, 2009 | Paul Boutin — 2 Comments

Like — Sign Up to see what your friends like. — +1 — Share — Tweet

iPhone users consume two to four times as much network data volume as other smartphone users, according to traffic measurement company Comscore.

That's increasingly a problem for AT&T, which serves all those iPhone users in the US and must pay for the bandwidth to handle it all. AT&T starts at a loss by subsidizing customers' iPhones. Then, it charges them only $30 per month for unlimited data download & upload — or the same it charges users of other smartphone, even though those users are cheaper to serve.

iPhone users now make up for 7.5% of AT&T's subscribers. Their data-hungry lifestyles have strained AT&T's infrastructure so much that "AT&T will need to add cell towers and spend more on the back-haul lines that connect the towers to the rest of the network," the WSJ reports, after citing stats from Alcatel-Lucent, a network equipment maker showing how bandwidth-hogging Web browsing is. Web browsing consumers 32% of data-related airtime but 69% of bandwidth, while email used 30% of data airtime but only 4% of bandwidth, the study found.

It's easy to pontificate — as the Journal's analyst sources do — that AT&T should do away with unlimited data plans, or raise the price on iPhone users. But the company's discount pricing on the expensive iPhone and its accompanying high-bandwidth lifetyle aren't an accident. With more touchscreen iPhone competitors like the Palm Pre coming to market, and AT&T's exclusive status as iPhone carrier set to expire next year, the company is clearly willing to pour money into locking down as many iPhone customers as it can.

(Photo credit: Anirudh Koul)

http://www.nytimes.com/2009/09/03/technology/companies/03att.html
http://venturebeat.com/2009/05/11/iphone-users-eating-up-atts-network/

(Mobile) Ad Hoc Networks: Overview

- Growing demand for contents --> infrastructure-centric networking paradigm appears inadequate

  --> promising alternative to offload the telcos' networks: exploit the user interactions to convey information: Ad hoc Networks
- **Goal of ad hoc nets**: allowing communication between (mobile) users in the absence of infrastructure.
- Such networks can be
  - Interference-limited
  - Or connectivity-limited

  --> use of nodes as relays to achieve end-to-end communication: Store-Carry-And-Forward paradigm
- entails a certain communication delay

  --> MANETs are also referred to as Delay Tolerant Networks (DTNs).

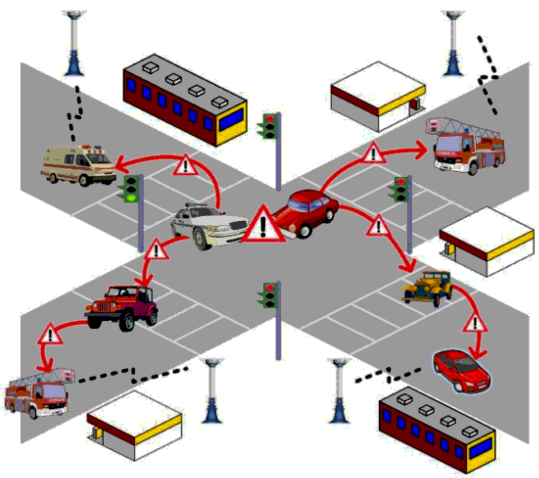
Mobile Ad Hoc Networks: Applications

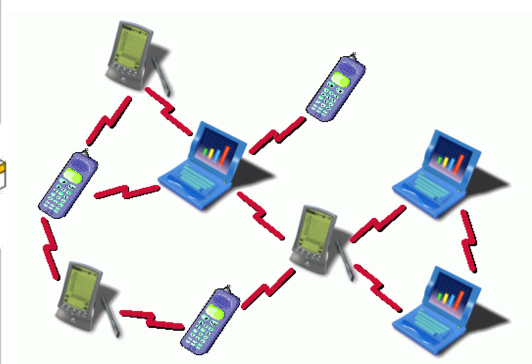
Civilian applications

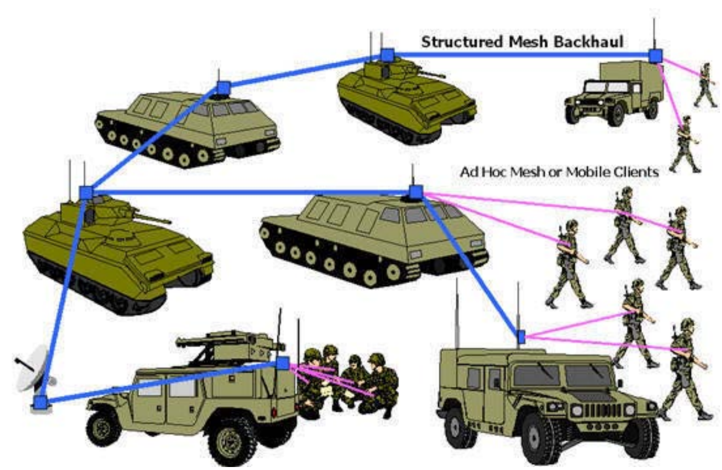
- Pocket-switched networks
- Vehicular networks
- Sensor networks

Military applications

- Deployment and communication on the battlefield







Mobile Ad Hoc Networks: Problems

- How to perform routing?
- How to perform scheduling?
- How to minimize the delivery delay under some energy constraint?
- How to deal with interference?
- What is the highest per-session throughput one can expect?
- How to deal with privacy?
- What mobility model best describes the targeted network of application?
- …

Outline

I. Delay-Throughput trade-offs:
   1. Static ad hoc networks under the physical and protocols models
   2. Mobile ad hoc networks under the physical and protocols models
   3. Distributed MIMO: nodes as antenna arrays

II. Network coding:
   1. Introduction
   2. Application: COPE
   3. Application: Routing and energy efficiency
   4. Application: NC meets TCP

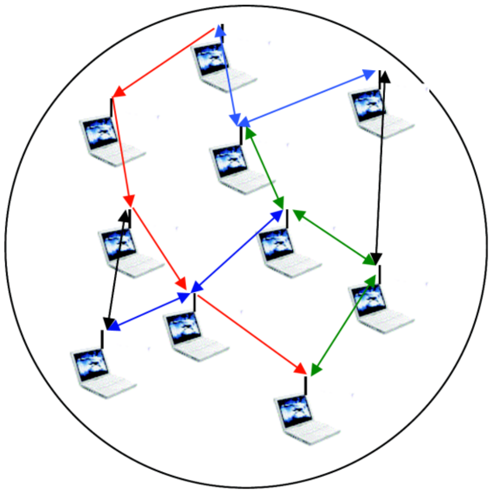
The problems

- How much traffic can wireless networks carry?
  (Or what is the capacity of wireless networks?)

- And how should information be transferred in wireless networks?

Multi-hop wireless networks

- Communication networks formed by nodes with radios
  - Spontaneously deployable anywhere
  - Automatically adaptive to number of nodes, traffic requirements, locations



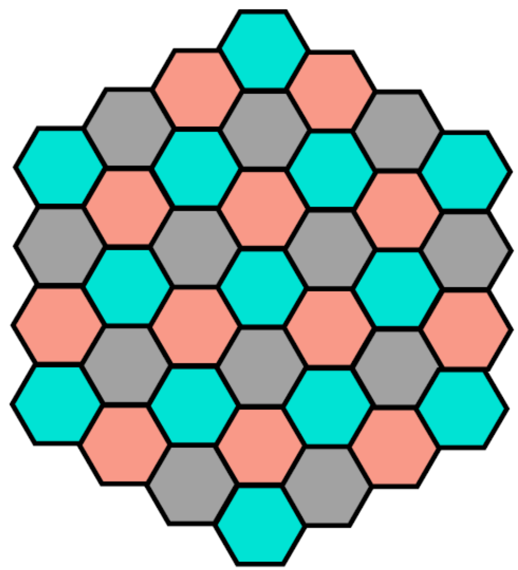
- "Multi-hop transport"
  - Nodes relay packets until they reach their destinations

Two fundamental properties of the wireless medium

- It is subject to fading and attenuation
  - Signals get distorted
  - Time varying channel
  - Unreliable
- It is a shared medium
  - Users share the same spectrum
  - Users are located next to each other
  - Transmissions can interfere with each other
  - So users need to cooperate to use the medium

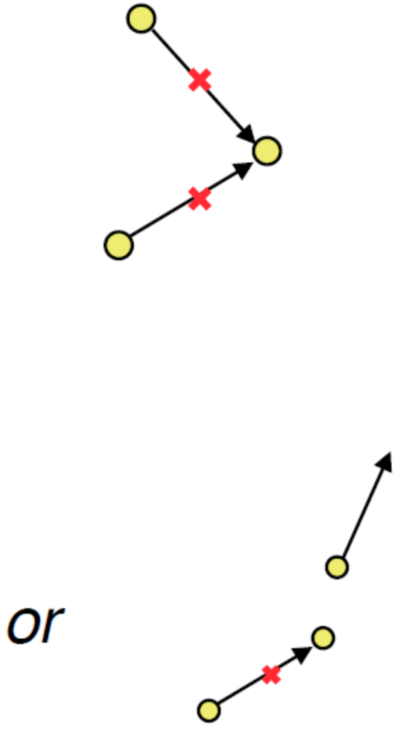
Spatial reuse of spectrum

- Spatial reuse of frequency in cellular systems



Shared nature of wireless medium

- Packets can "collide" destructively
  - Destructive interference
  - Nothing can be decoded from two concurrent transmissions in same region



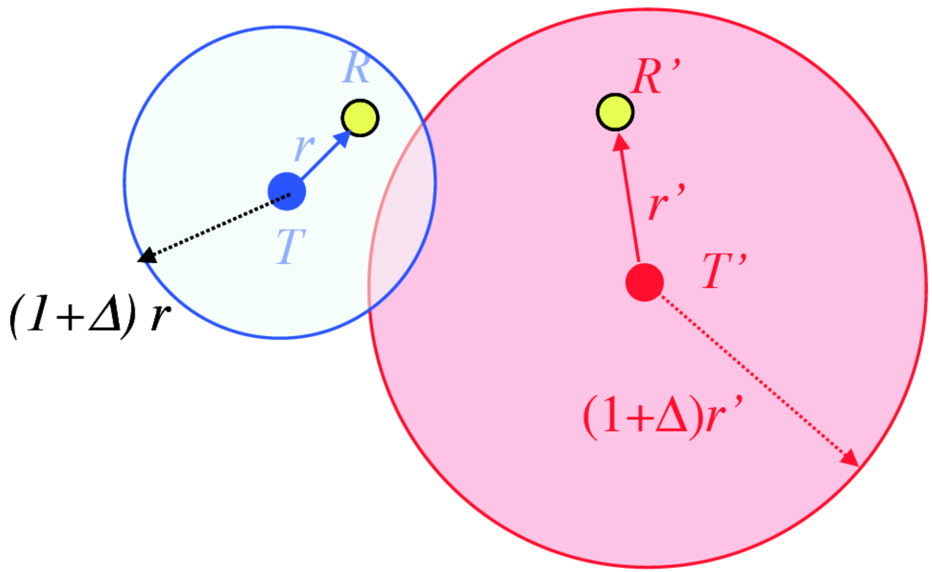
One model for successful sharing: the Protocol Model

- <u>Protocol Model</u>

  Receiver R should be

  (i) within range r of its own transmitter T

  (ii) outside footprint $(1+\Delta)r'$ of any other transmitter T' using range r'



Other models for successful sharing

- Signal to Noise Ratio (SNR)

$$\text{Signal to Noise Ratio} = \text{SNR} := \frac{\text{Received Signal Strength}}{\text{Noise}} = \frac{P_i / r_i^{\alpha}}{N}$$

- Signal to *Interference plus Noise* Ratio (SINR)

$$\text{SINR} := \frac{\text{Received Signal Strength}}{\text{Interference Strength + Noise}} = \frac{P_i / r_i^{\alpha}}{\sum_{j \neq i} P_j / r_j^{\alpha} + N}$$

- Model 2: Reception successful if SINR exceeds a threshold:

$$\text{SINR} = \frac{P_i / r_i^{\alpha}}{\sum_{j \neq i} P_j / r_j^{\alpha} + N} \geq \beta$$

The Physical Model

- Model 3: Transmitter-to-Receiver Communication Rate depends on SINR:

$$\text{Rate} = B \log \left(1 + \frac{P_i r_i^{-\alpha}}{N + \sum_{j \neq i} P_j r_j^{-\alpha}} \right) \text{ bps}$$

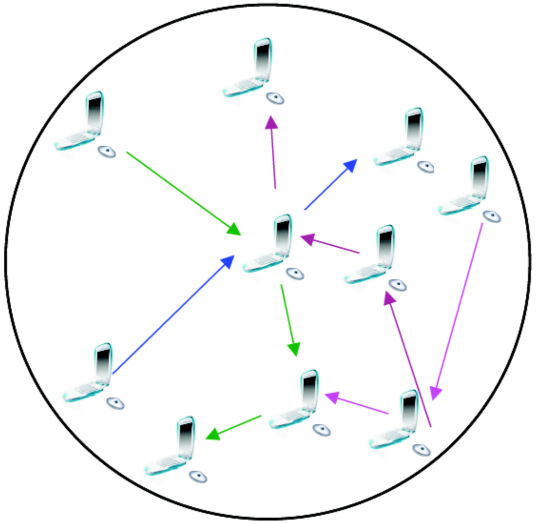
The Physical SINR Model

- Physical Model: Signal-to-Interference-Plus Noise Ratio (SIR) Model

$$\text{SINR Ratio} = \frac{P_i r_i^{-\alpha}}{N + \sum_{j \neq i} P_j r_j^{-\alpha}} \geq \beta$$

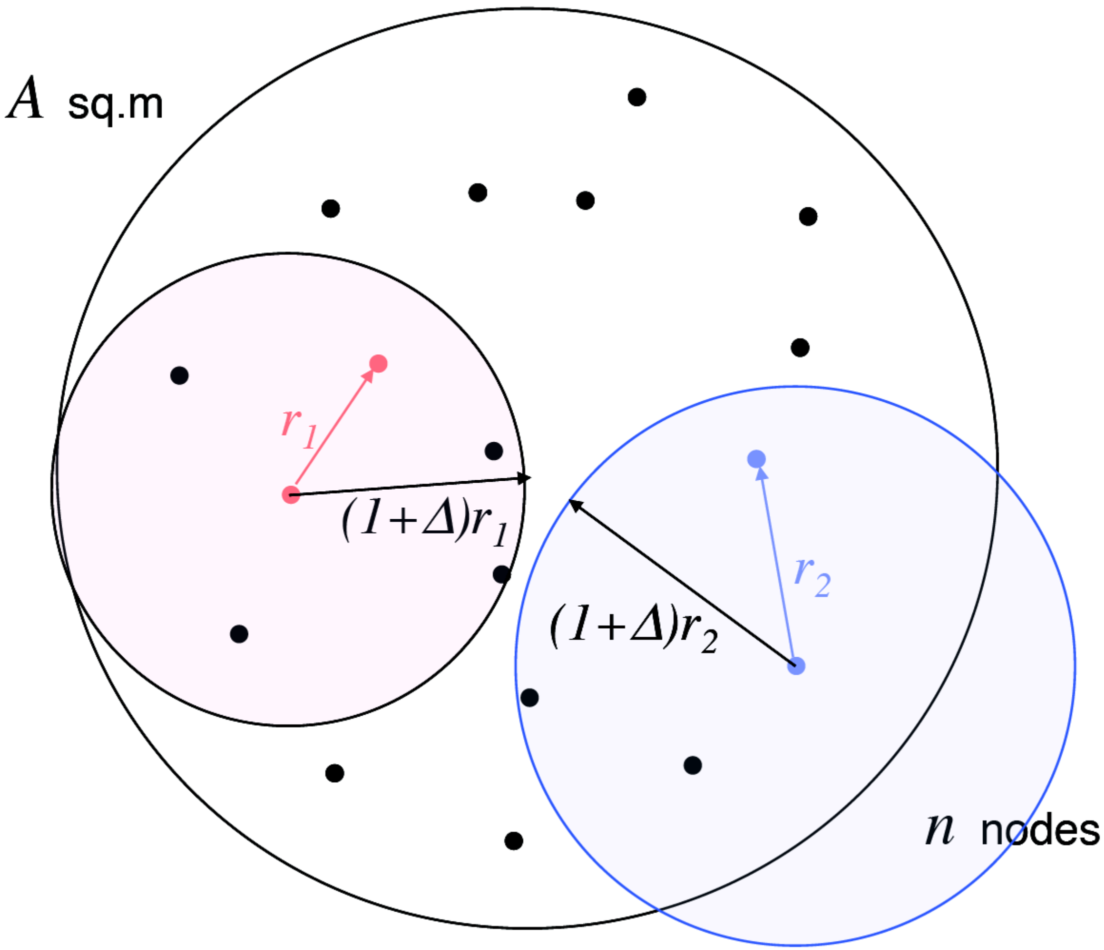
  - P_i = power of i-th node
  - N = Noise power
  - r_j = Distance of j -th transmitter from given receiver
  - $r^{-\alpha}$: Signal Power Path Loss, $\alpha > 2$
  - β = SIR for successful reception

A framework for studying wireless networks



- Model
  - Disk of area A m^2
  - n nodes
  - Each can transmit at W bits/sec
- Wireless channel is a shared medium
  - Packets are successfully received when there is no local interference
- How much information can such wireless networks carry?
  - Throughput for each node: Measured in Bits/Sec
  - Transport capacity of entire network: Measured in Bit-Meters/Sec
  - Scaling with the number of nodes n

Transmissions consume area

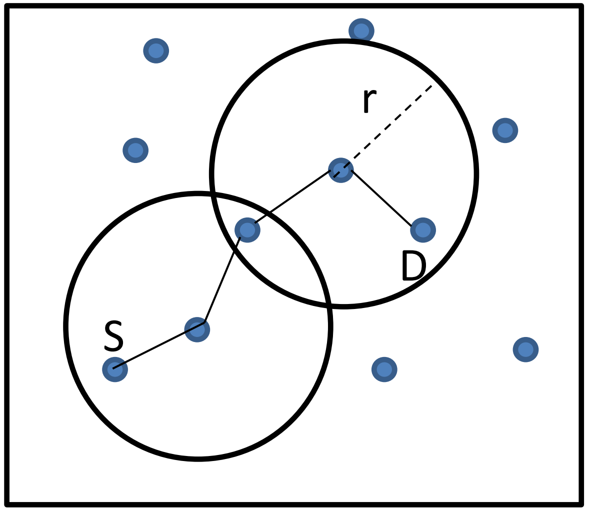


Notation

We recall the following notation: (i) $f(n) = O(g(n))$ means that there exists a constant c and integer N such that $f(n) \leq cg(n)$ for $n > N$. (ii) $f(n) = o(g(n))$ means that $\lim_{n\to\infty} f(n)/g(n) = 0$. (iii) $f(n) = \Omega(g(n))$ means that $g(n) = O(f(n))$, (iv) $f(n) = \omega(g(n))$ means that $g(n) = o(f(n))$. (v) $f(n) = \Theta(f(n))$ means that $f(n) = O(g(n))$; $g(n) = O(f(n))$.

Static Ad Hoc networks: Delay-throughput tradeoff

- Settings:
  - network area = 1m²
  - N nodes
  - N unicast sessions



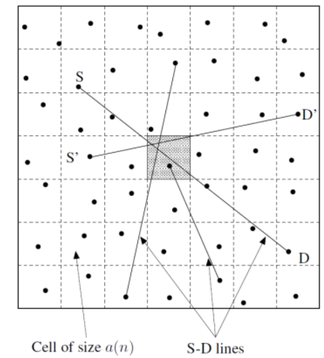
- When several nodes transmit simultaneously, a receiver can successfully receive the data sent by the desired transmitter only if the interference from the other nodes is sufficiently small: SINR>β

- Direct transmission --> minimum delay, lowest per-session throughput
- Lowering *r* --> use of relay nodes

- Density: N increases while the network area remains constant

--> What are the best per-session throughput and delay and how do they scale with N?

Static Ad Hoc networks : Delay-throughput tradeoff



- Partition into regular cells of area $a_N=O(r_N^2)$
  --> Na_N nodes per cell on average

- Average delay $D_N=O(h_N(r))$ average number of hops: $$D_N = O\left(\frac{1}{\sqrt{a_N}}\right)$$
  --> ND_N packets to be relayed at each TS
  --> one relay handles $ND_N/N=D_N$ packets of different sessions each TS

- But a specific relay is activated only once every Na_N TS
  --> each session gets a throughput of

$$T_N = O\left(\frac{1}{Na_N D_N}\right)$$

$$T_N = O\left(\frac{1}{N\sqrt{a_N}}\right)$$

$$D_N = O\left(\frac{1}{\sqrt{a_N}}\right)$$

P. Gupta and P.R. Kumar, *The capacity of wireless networks,* IEEE Transactions on Information Theory, Vol. 46, No. 2, March 2000
A. El Gamal, J. Mammen, B. Prabhakar and D. Shah, *Throughput-delay trade-off in wireless networks,* Infocom 2004

Static Ad Hoc networks : Delay-throughput tradeoff

- Specific cases:
  - Nodes scattered on a squared grid:

$$r = \frac{1}{\sqrt{N}}, \quad T_N = O\left(\frac{1}{\sqrt{N}}\right), \quad D_N = O\left(\sqrt{N}\right)$$

  - Nodes randomly scattered:

$$r = \sqrt{\frac{\log(N)}{N}}, \quad T_N = O\left(\frac{1}{\sqrt{N\log(N)}}\right), \quad D_N = O\left(\sqrt{\frac{N}{\log(N)}}\right)$$

Theorem: Throughput that can be supported (Gupta & K '00)

$$\lim_{n\to\infty} \Pr(\lambda(n) = \frac{c}{\sqrt{n\log n}} \text{ is feasible}) = 1, \text{ and}$$

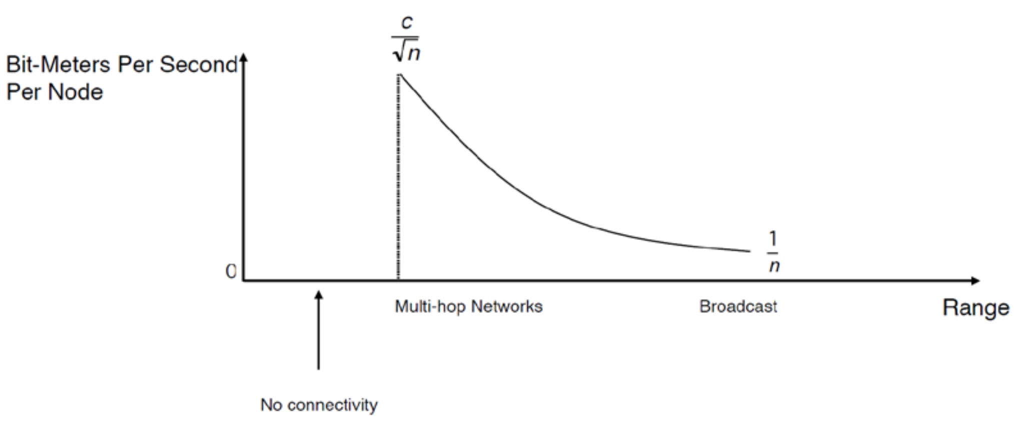
$$\lim_{n\to\infty} \mathrm{P}(\lambda(n) = \frac{c'}{\sqrt{n\log n}} \text{ is feasible}) = 0$$

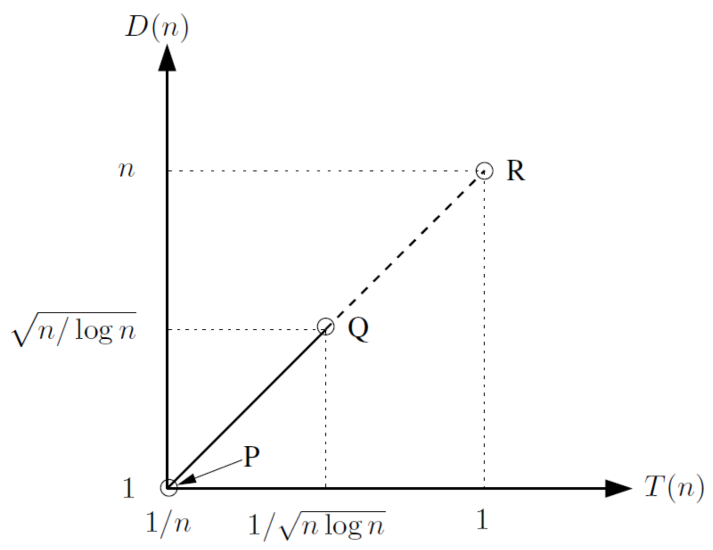
Sharp cutoff phenomenon

P. Gupta and P. R. Kumar, *The capacity of wireless networks,* IEEE Transactions on Information Theory, Vol. 46, No. 2, March 2000

Static Ad Hoc networks : Delay-throughput tradeoff

Why multi-hop?
- Multi-hop increases traffic carrying capacity
- It may also increase delay





Outline

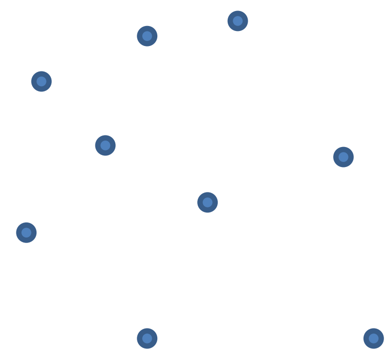
I. Delay-Throughput trade-offs:
   1. Static ad hoc networks under the physical and protocols models
   2. Mobile ad hoc networks under the physical and protocols models
   3. Distributed MIMO: nodes as antenna arrays

II. Network coding:
   1. Introduction
   2. Application: COPE
   3. Application: Routing and energy efficiency
   4. Application: NC meets TCP

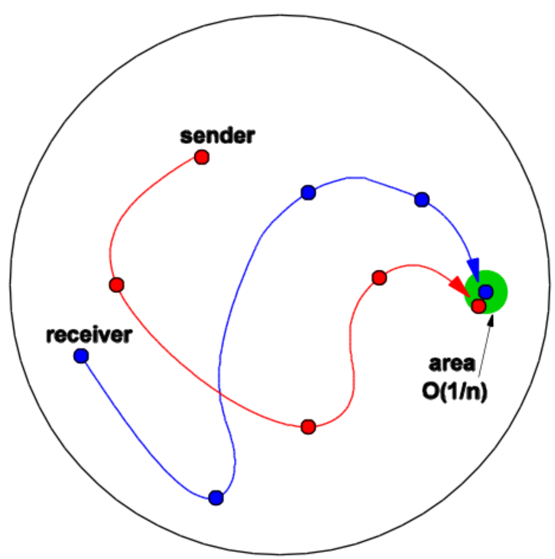
Mobile Ad Hoc networks: capacity

- **Mobility increases the capacity of Ad Hoc networks**:

  T_N=*O(1)* using two-hop routing

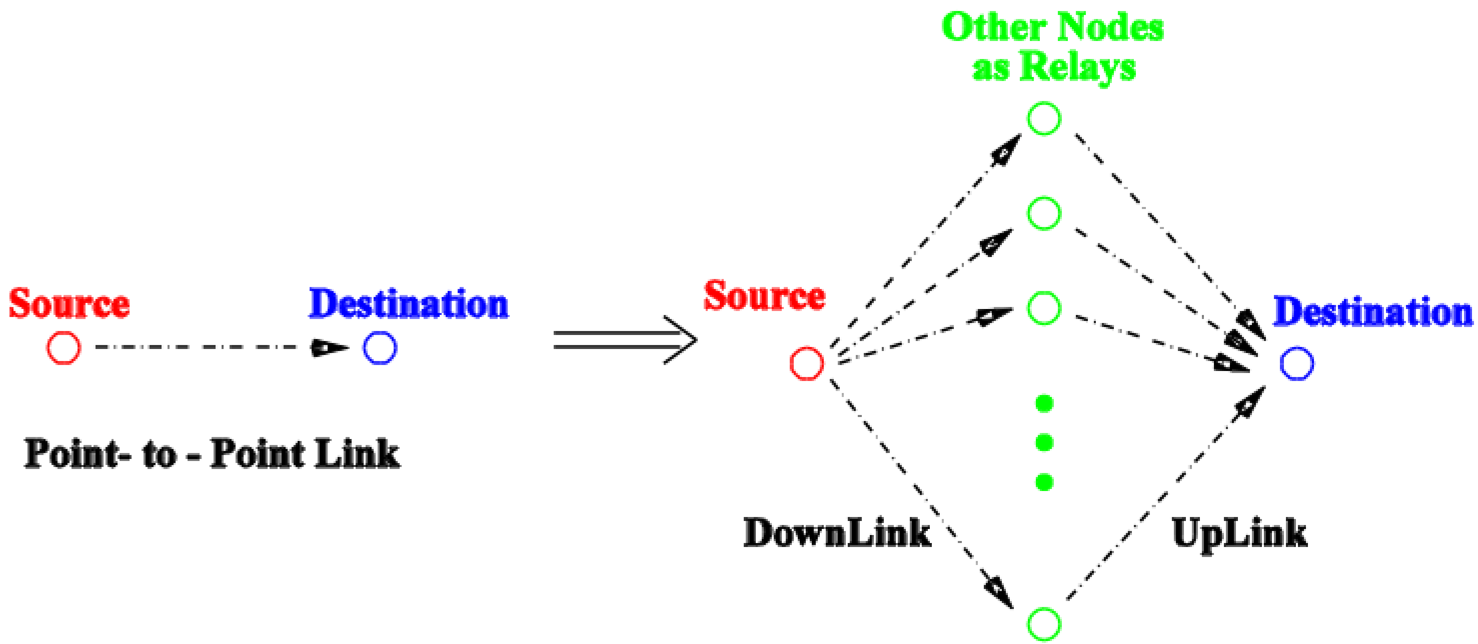


- Mobility model: stationary, ergodic, uniform, iid
- Direct communication does not work:
  - The source and destination are nearest neighbors only O(1/n) of the time.



M. Grossglauser and D. Tse, *Mobility increases the capacity of Ad Hoc networks,* IEEE/ACM Transactions on Networking, Vol. 10, No. 4, August 2002

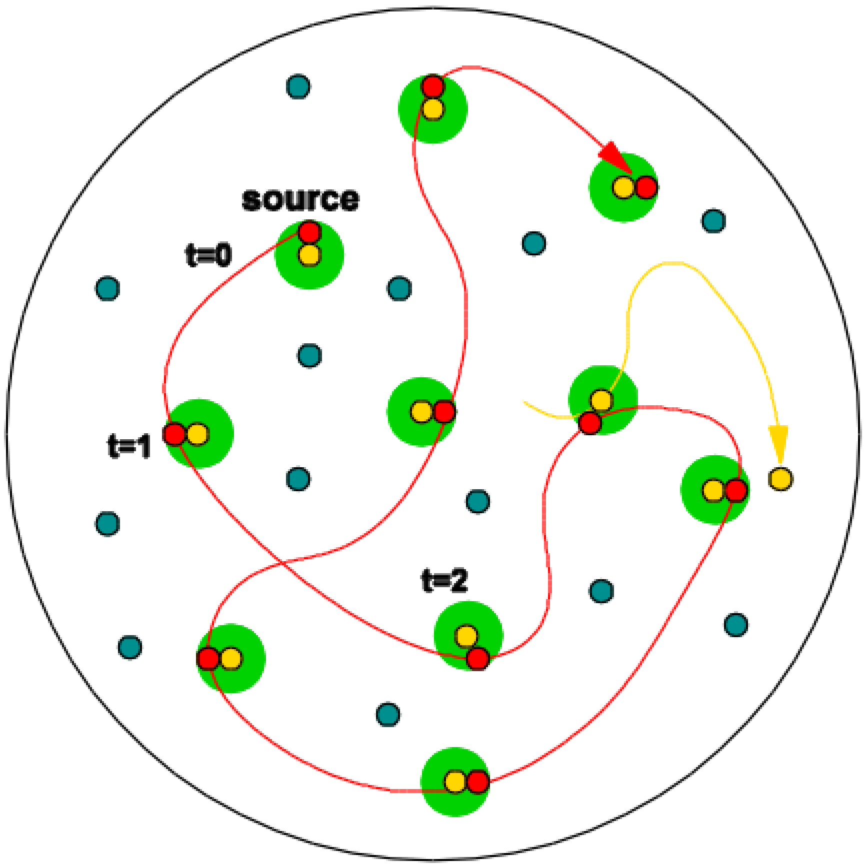
Multiuser Diversity via Relaying



Multiuser diversity created artificially using all other nodes as relays.

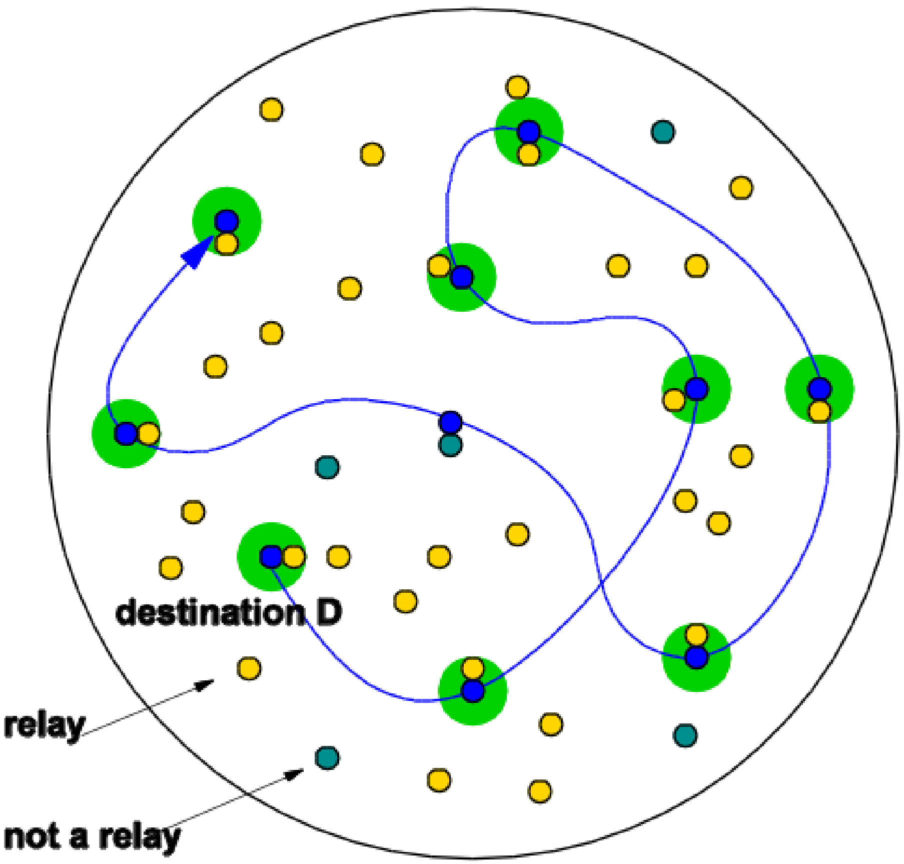
Two-hop routing
Phase 1: Source to Relays

- At each time slot, source relays a packet to nearest neighbor.
- Different packets are distributed to different relay nodes.



Two-hop routing
Phase 2: Relays to Destination

- Steady state: all nodes have packets destined for D.
- Each relay node forwards packets to D only when it gets close.

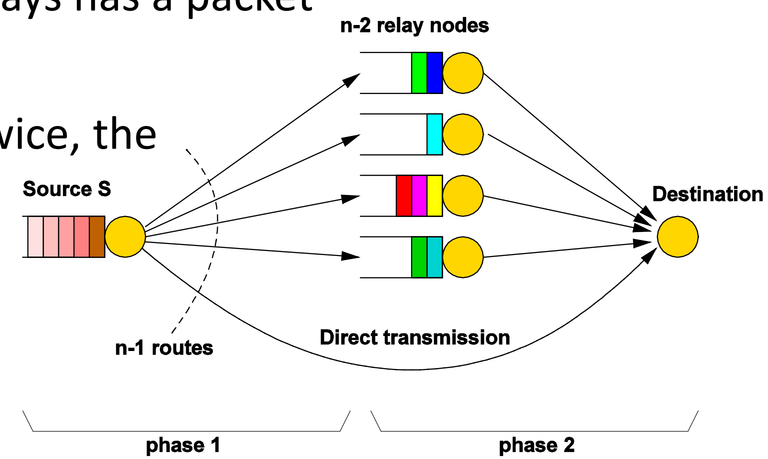


Phase I and II Staggered

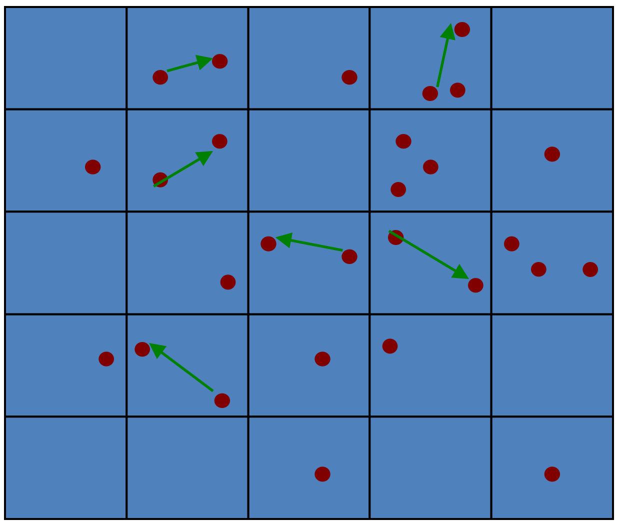
- Key ingredients:
  - It is possible to schedule O(N) concurrent successful transmissions per TS with local communication
  - Each packet goes through only one relay node that temporarily buffers the packet until final delivery to the destination is possible.
  - In steady-state, the packets of every source node will be distributed across all the nodes in the network

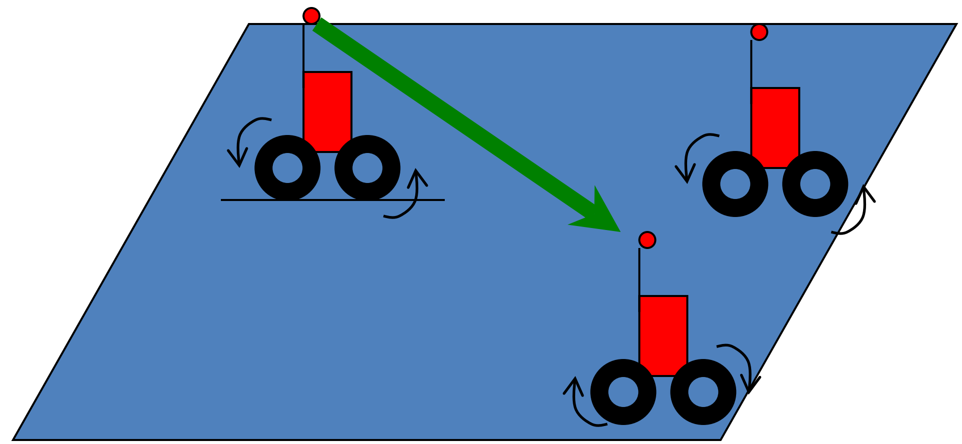
    --> every node in the network will have packets buffered destined to every other node

    --> a scheduled sender–receiver pair always has a packet to send
  - As no packet is transmitted more than twice, the achievable total throughput is O(N).



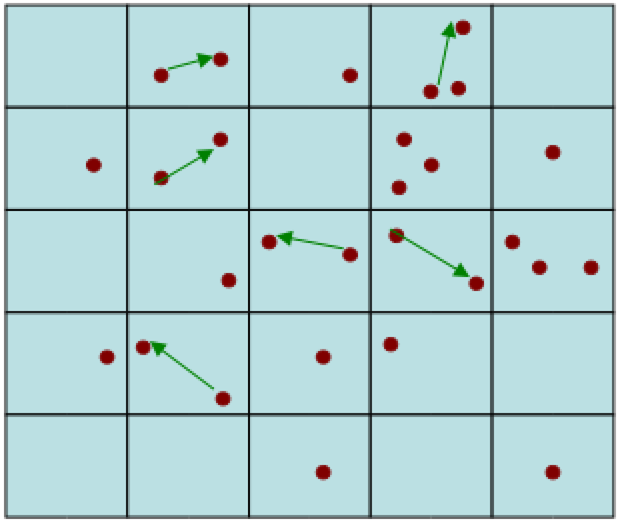
Improving Delay in Ad-Hoc Mobile Networks via Redundant Packet Transfers





Grossglauser-Tse 2-hop relay algorithm yields:
O(1) thruput, O(*N*) delay

<u>Question</u>: Can we improve delay by sending multiple copies of the same packet?



Cell partitioned network model:

N nodes, C cells

$d = N/C$ = user/cell density

- Timeslotted system
- 1 transmission per cell
- no intercell interference

(physical layer constraints)

Mobility model:

(i) Markov Random Walk

(ii) i.i.d. jump mobility (extreme model)

Traffic model:

Each user $i \in \{1, \ldots, N\}$ sends to a unique destination $d(i) \in \{1, \ldots, N\}$.

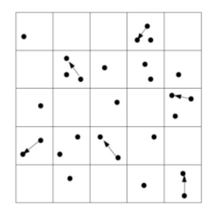
Example: $1 \longleftrightarrow 2$, $3 \longleftrightarrow 4$, $5 \longleftrightarrow 6$, …

Mobile Ad Hoc networks: DTT

- Algorithms which do not use redundancy cannot achieve an average delay of less than *O(N).*
- No algorithm (with or without redundancy) which restricts packets to 2-hop paths can provide an average delay better than $O\left(\sqrt{N}\right)$.

scheme	throughput	delay
no redundancy	$O(1)$	$O(N)$
redundancy 2-hop	$O(1/\sqrt{N})$	$O(\sqrt{N})$
redundancy multi-hop	$O(\frac{1}{N\log(N)})$	$O(\log(N))$

- C cells in the area 1, d=N/C nodes per cell



- Delay-Throughput Tradeoff: $$\frac{D_N}{T_N} \geq O(C)$$

M. J. Neely and E. Modiano, *Capacity and Delay Tradeoffs for Ad-Hoc Mobile Networks,* IEEE Transactions on Information Theory, Vol. 51, No. 6, June 2005

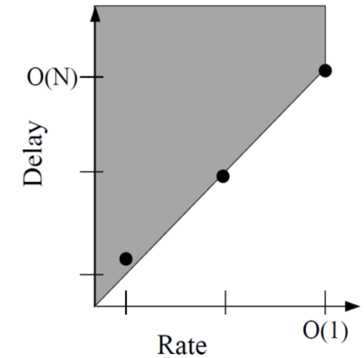
Mobile Ad Hoc networks: DTT

- Let R be the average redundancy per packet

- Intuition when *$d=N/C=o(N)$ and $R=o(N)$*:
  - per TS: $\lambda RN \leq N$
  - once R copies have been spread out: $T_2(R) = \frac{C}{R}$
  - Since $D_N(R) \geq T_2(R)$, we get

$$\frac{D_N(R)}{T_N(R)} \geq O(C) = O\left(\frac{N}{d}\right)$$

Mobile Ad Hoc networks: DTT



- Specific cases:
  - *C=O(N), d=O(1)*:

scheme	throughput	delay
no redundancy	$O(1)$	$O(N)$
redundancy 2-hop	$O(1/\sqrt{N})$	$O(\sqrt{N})$
redundancy multi-hop	$O(\frac{1}{N\log(N)})$	$O(\log(N))$

- Another DTT, achieved by two-hop routing and coding, in case we allow the transmission range to vary with the desired delay: $r^2 = \frac{1}{\sqrt{ND}} \Rightarrow R = \sqrt{\frac{N}{D}}, T_1(R) = d = Nr^2$

$$T_N = \sqrt{\frac{D_N}{N}}$$

Outline

I. Delay-Throughput trade-offs:
   1. Static ad hoc networks under the physical and protocols models
   2. Mobile ad hoc networks under the physical and protocols models
   3. Distributed MIMO: nodes as antenna arrays

II. Network coding:
   1. Introduction
   2. Application: COPE
   3. Application: Routing and energy efficiency
   4. Application: NC meets TCP

The Interference Barrier

- Lots of recent advances in physical layer wireless communication (multiple antennas MIMO, space-time codes, opportunistic scheduling, turbo codes, hybrid ARQ....)
- From theory to practice in a decade.
- Gains pertain mainly to point-to-point or multiple access performance.
- But performance of many wireless systems ultimately limited by interference.
- Breaking this interference barrier will be the next step.

Examples of Interference Barrier

- Cellular networks: inter-cell interference
- Ad hoc networks: interference from simultaneous transmissions
- Wireless LANs: interference between adjacent networks
- Cognitive networks: interference between primary and secondary users and between multiple secondary systems

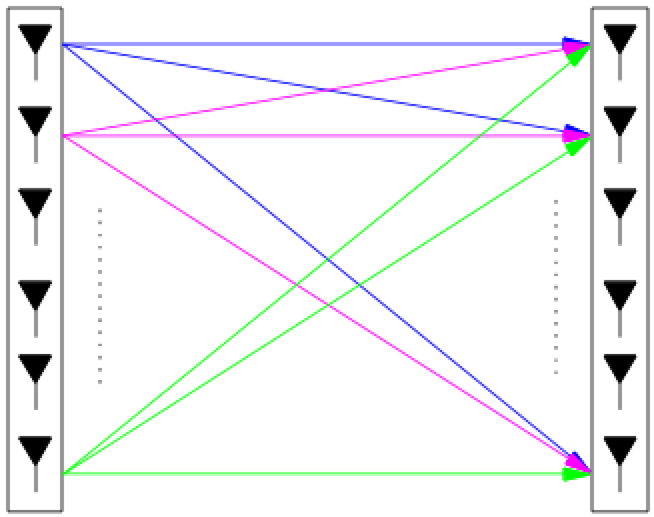
Breaking the interference barrier

- Several approaches to break the interference barrier:
  - cooperative distributed MIMO
  - exploiting mobility to localize interference
  - interference alignment

- Key message:

  Solving the interference problem requires a combination of physical layer and architectural ideas.

MIMO in One Slide



H: channel matrix
Φ: covariance matrix of the transmit signal
K: covariance matrix of the noise

$$C = \max_{\Phi} E_H \left\{ \log_2 \left[\det \left(H \Phi H^{*T} (K^n)^{-1} + I_{n_R} \right) \right] \right\}$$

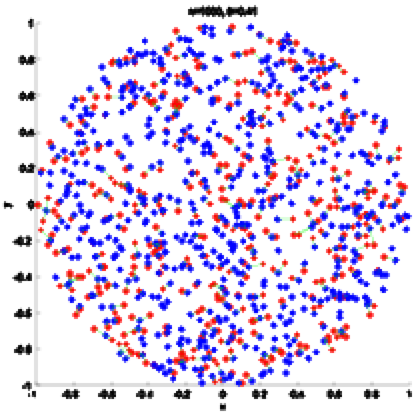
- When the transmitter has no knowledge about the channel, it is optimal to use a uniform power distribution --> $\Phi = \frac{P_T}{n_T} I_{n_T}$
- The number of parallel subchannels is determined by the rank of the channel matrix: $rank(H) = k \le \min(n_T, n_R)$

$$C = \max_{\Phi} E_H \left\{ \sum_{i=1}^{k} \log_2 \left(1 + \frac{P_T}{n_T \sigma^2} \lambda_i \right) \right\}$$

M-by-M MIMO system with a sufficiently random channel supports M simultaneous data streams.

Gupta-Kumar capacity is interference-limited

Can we get linear scaling thanks to MIMO?



- Long-range transmission causes too much interference.
- Multi-hop means each packet is transmitted many times.
- To get linear scaling, must be able to do many simultaneous long-range transmissions.
- How to deal with interference?
- A natural idea: distributed MIMO!
- But cooperation overhead is bottleneck.

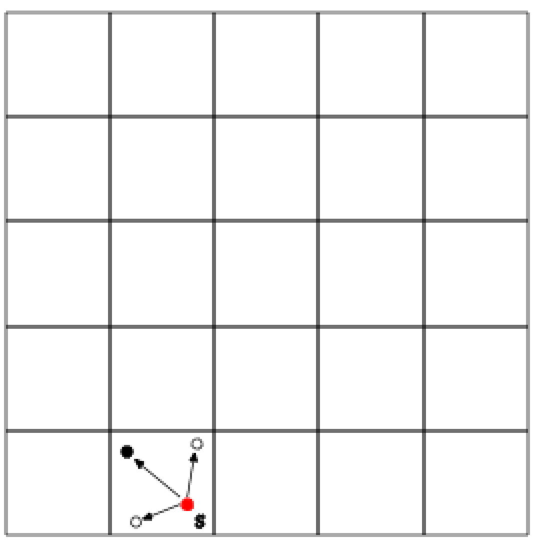
- What kind of cooperation architecture minimizes overhead?

--> A. Ozgur, O. Lévêque, and D. Tse, Hierarchical Cooperation Achieves Optimal Capacity Scaling in Ad Hoc Networks. IEEE Transactions on Information Theory, 53(10):3549-3572, 2007.

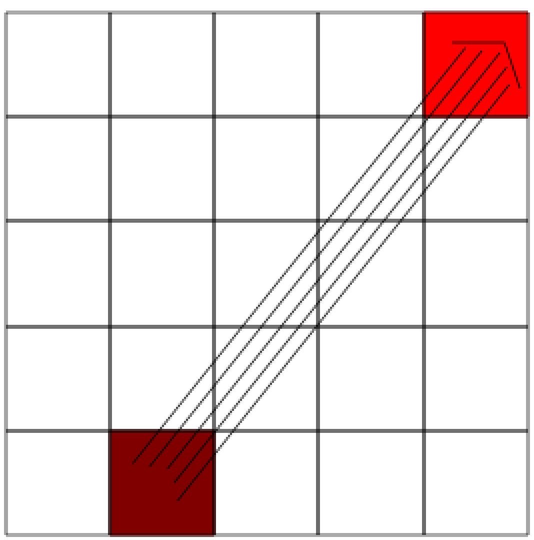
A 3-phase scheme

- Divide the network into clusters of size M nodes.
- Focus first on a specific S-D pair.
- source s wants to send M bits to destination d.

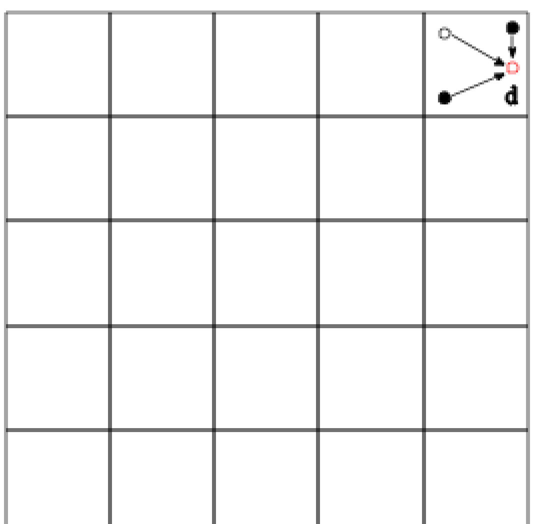
Phase 1 :
Setting up Tx cooperation:
1 bit to each node in Tx cluster



Phase 2:
Long-range MIMO between s and d clusters.



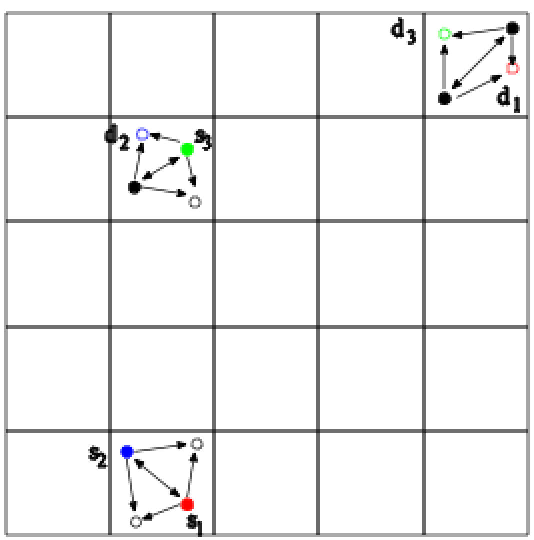
Phase 3:
Each node in Rx cluster quantizes signal into k bits and sends to destination d.



Parallelization across S-D Pairs

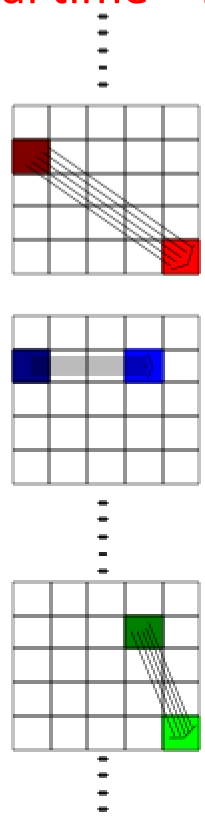
Phase 1:
Clusters work in parallel.
Sources in each cluster take turn distributing their bits.

Total time = M^2



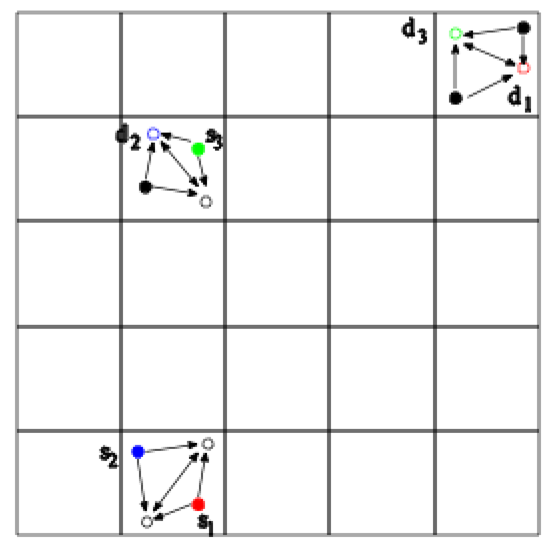
Phase 2:
1 MIMO trans.
at a time.

Total time = n



Phase 3:
Clusters work in parallel.
Destinations in each cluster take turn collecting their bits.

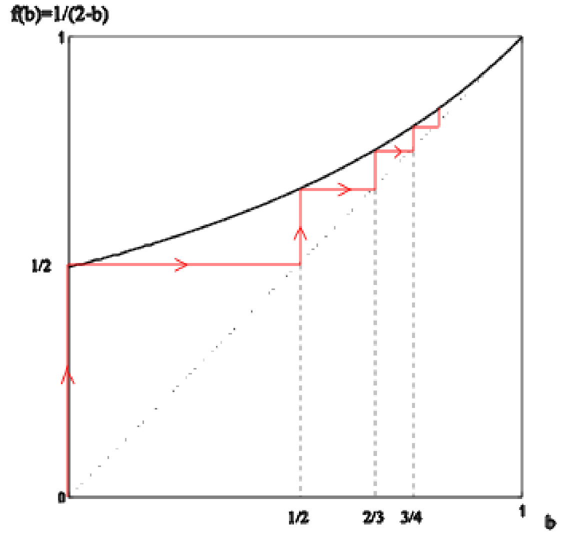
Total time = kM^2



Recursion for throughput calculation

- Level b, with $b \in [0,1]$:
  - The net of size *n* is partioned into cells of size *M*
  - Assume an aggregate thru of $T(n) = n^b$ (hence M^b) is feasible
  - Total number of bits transferred: nM
  - Total time in all 3 phases: $\frac{M^2}{M^b} + n + \frac{QM^2}{M^b}$
  - Aggr. thru: $\frac{nM}{M^{2-b} + n + QM^{2-b}}$

    is max for $M = n^{\frac{1}{2-b}}$, giving $T(n) = \frac{1}{Q+2} n^{\frac{1}{2-b}} > n^b$

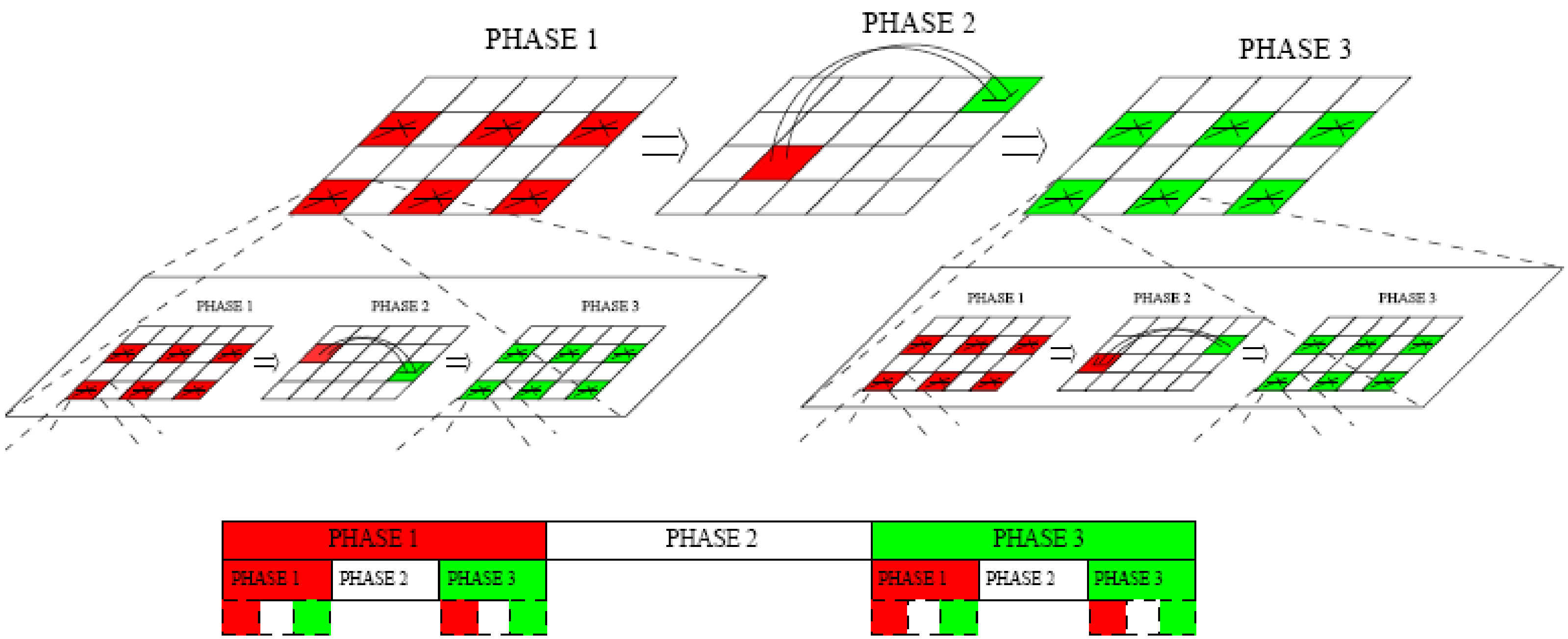


MIMO + Hierarchical Cooperation -> Linear Scaling

Setting up Tx cooperation

Long-range MIMO

Cooperate to decode



By having many levels of hierarchy, we can get as close to linear scaling as we wish.

Outline

I. Delay-Throughput trade-offs:
   1. Static ad hoc networks under the physical and protocols models
   2. Mobile ad hoc networks under the physical and protocols models
   3. Distributed MIMO: nodes as antenna arrays

II. Network coding:
   1. Introduction
   2. Application: COPE
   3. Application: Routing and energy efficiency
   4. Application: NC meets TCP

Introduction to Network Coding

- Theory
  - Max-Flow Min-Cut Theorem
  - Multicast Problem
  - Network Coding
- Practice

Max-Flow Min-Cut Theorem

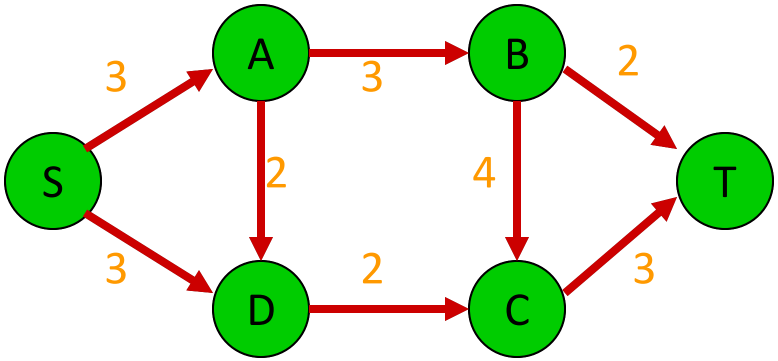
- Definition
- Graph
- Min-Cut and Max-Flow

Definition

- (From Wiki) The **max-flow min-cut theorem** is a statement in optimization theory about maximal flows in flow networks
- The maximal amount of flow is equal to the capacity of a minimal cut.
- In layman terms, the maximum flow in a network is dictated by its bottleneck.

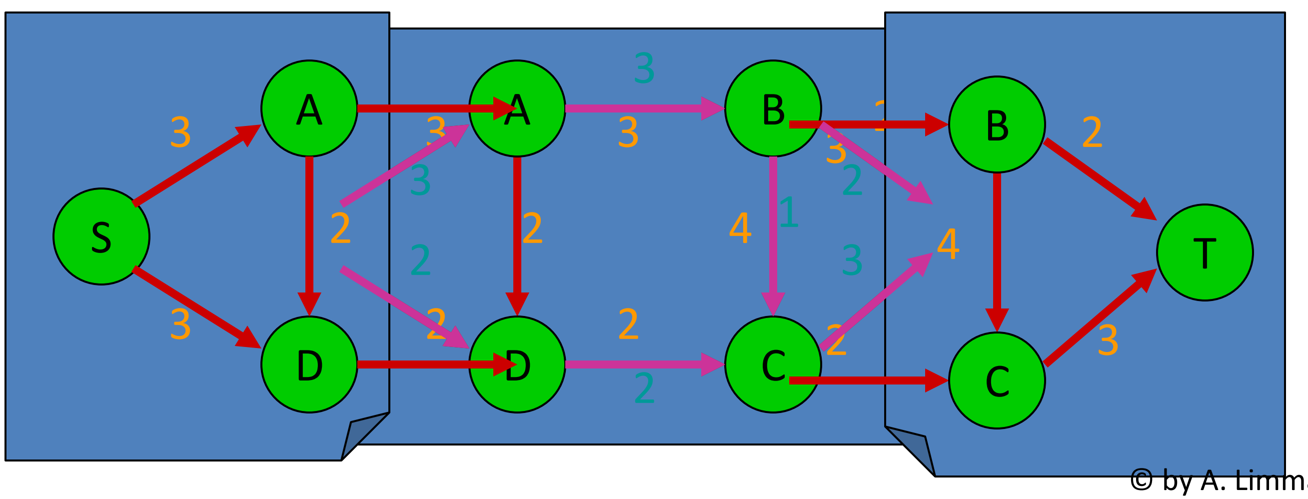
Graph

- Graph G(V,E): consists of a set V of vertices and a set E of edges:
  - V consists of sources, sinks, and other nodes
  - A member e(u,v) of E has a capacity c(u,v) to send information from u to v



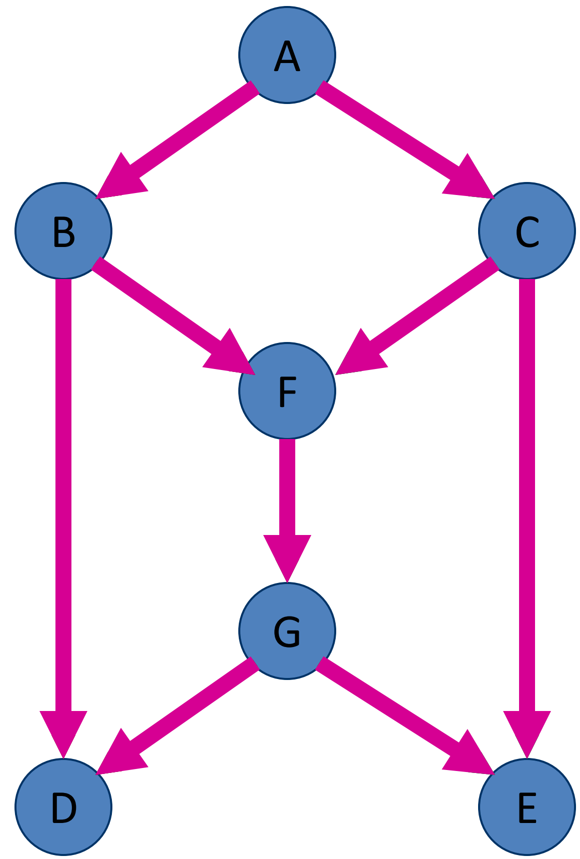
Min-Cuts and Max-Flows

- Cuts: Partition of vertices into two sets
- Size of a Cut = Total Capacity Crossing the Cut
- Min-Cut: Minimum size of Cuts = 5
- Max-Flows from S to T
- Min-Cut = Max-Flow



Multicast Problem

- Butterfly Networks: Each edge's capacity is 1.
- Max-Flow from A to D = 2
- Max-Flow from A to E = 2
- Multicast Max-Flow from A to D and E = 1.5
- Max-Flow for each individual connection is not achieved.

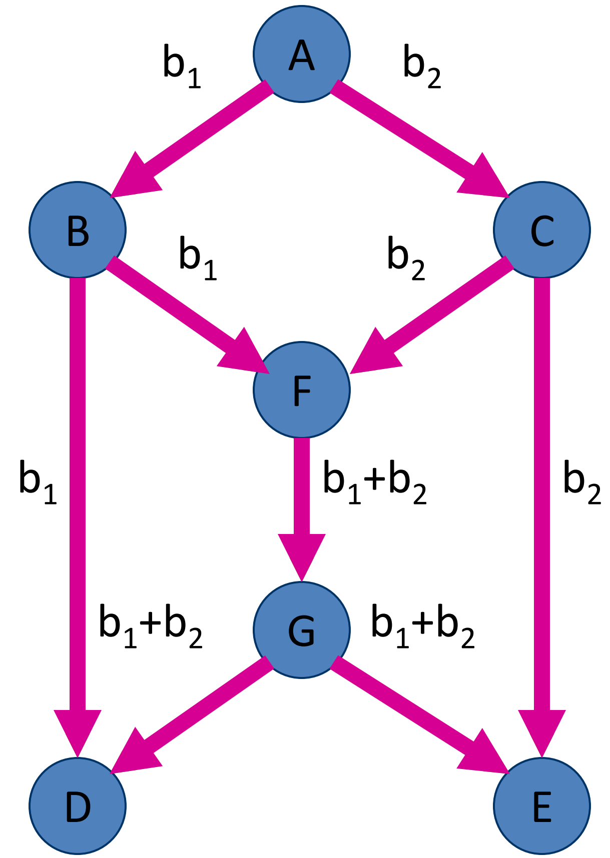


Network Coding

- Introduction
- Linear Network Coding
- Transfer Matrix
- Network Coding Solution
- Connection between an Algebraic Quantity and a Graph Theoretic Tool
- Finding Network Coding Solution

Introduction

- Ahlswede et al. (2000)
  - With network coding, every sink obtains the maximum flow.
- Li et al. (2003)
  - Linear network coding is enough to achieve the maximum flow



Linear Network Coding

- Random Processes in a Linear Network
  - Source Input: $X(v,l) = \{x_0(v,l), x_1(v,l), \ldots\}$
  - Info. Along Edges: $Y(e) = \{y_0(e), y_1(e), \ldots\}$
  - Si… $\ldots l) = \{z_0(v, \ldots$
- Rela… …them

Weighted Combination of processes generated at v

Weighted Combination of processes from adjacent edges of e

Weighted Combination from all incoming edges

e comes out of v

$$Y(e) = \sum_{l=1}^{\mu(v)} \alpha_{l,e} X(v,l) + \sum_{e':head(e')=tai\ldots} \beta_{e\ldots}$$

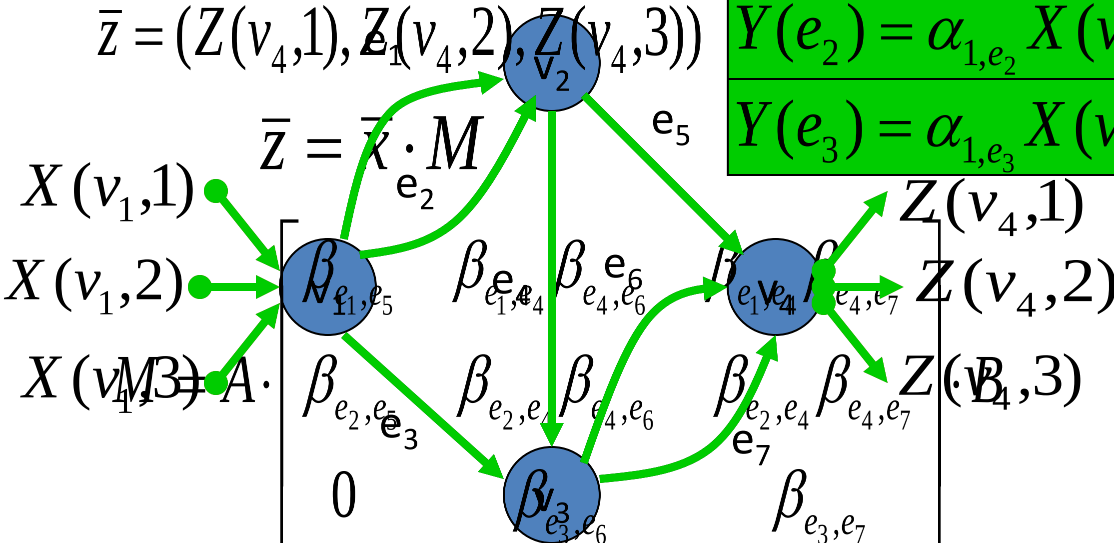
$$Z(v,j) = \sum_{e':head(e')=v} \varepsilon_{e',j} Y(e')$$

Transfer Matrix

Let $\bar{x} = (X(v_1,1), X(v_1,2), X(v_1,3))$

$\bar{z} = (Z(v_4,1), Z(v_4,2), Z(v_4,3))$

$$\bar{z} = \bar{x} \cdot M$$



$$M = A \cdot \begin{bmatrix} \beta_{e_1,e_5} & \beta_{e_1,e_4}\beta_{e_4,e_6} & \beta_{e_1,e_4}\beta_{e_4,e_7} \\ \beta_{e_2,e_5} & \beta_{e_2,e_4}\beta_{e_4,e_6} & \beta_{e_2,e_4}\beta_{e_4,e_7} \\ 0 & \beta_{e_3,e_6} & \beta_{e_3,e_7} \end{bmatrix} \cdot B$$

$$A = \begin{bmatrix} \alpha_{1,e_1} & \alpha_{1,e_2} & \alpha_{1,e_3} \\ \alpha_{2,e_1} & \alpha_{2,e_2} & \alpha_{2,e_3} \\ \alpha_{3,e_1} & \alpha_{3,e_2} & \alpha_{3,e_3} \end{bmatrix}$$

$$B = \begin{bmatrix} \varepsilon_{e_5,1} & \varepsilon_{e_5,2} & \varepsilon_{e_5,3} \\ \varepsilon_{e_6,1} & \varepsilon_{e_6,2} & \varepsilon_{e_6,3} \\ \varepsilon_{e_7,1} & \varepsilon_{e_7,2} & \varepsilon_{e_7,3} \end{bmatrix}$$

$$Y(e_1) = \alpha_{1,e_1} X(v,1) + \alpha_{2,e_1} X(v,2) + \alpha_{3,e_1} X(v,3)$$

$$Y(e_2) = \alpha_{1,e_2} X(v,1) + \alpha_{2,e_2} X(v,2) + \alpha_{3,e_2} X(v,3)$$

$$Y(e_3) = \alpha_{1,e_3} X(v,1) + \alpha_{2,e_3} X(v,2) + \alpha_{3,e_3} X(v,3)$$

$$Y(e_4) = \beta_{e_1,e_4} Y(e_1) + \beta_{e_2,e_4} Y(e_2)$$

$$Y(e_5) = \beta_{e_1,e_5} Y(e_1) + \beta_{e_2,e_5} Y(e_2)$$

$$Y(e_6) = \beta_{e_3,e_6} Y(e_3) + \beta_{e_4,e_6} Y(e_4)$$

$$Y(e_7) = \beta_{e_3,e_7} Y(e_3) + \beta_{e_4,e_7} Y(e_4)$$

$$Z(v_4,1) = \varepsilon_{e_5,1} Y(e_5) + \varepsilon_{e_6,1} Y(e_6) + \varepsilon_{e_7,1} Y(e_7)$$

$$Z(v_4,2) = \varepsilon_{e_5,2} Y(e_5) + \varepsilon_{e_6,2} Y(e_6) + \varepsilon_{e_7,2} Y(e_7)$$

$$Z(v_4,3) = \varepsilon_{e_5,3} Y(e_5) + \varepsilon_{e_6,3} Y(e_6) + \varepsilon_{e_7,3} Y(e_7)$$

Network Coding Solution

$$\bar{z} = \bar{x} \cdot M$$

$$M = A \cdot \begin{bmatrix} \beta_{e_1,e_5} & \beta_{e_1,e_4}\beta_{e_4,e_6} & \beta_{e_1,e_4}\beta_{e_4,e_7} \\ \beta_{e_2,e_5} & \beta_{e_2,e_4}\beta_{e_4,e_6} & \beta_{e_2,e_4}\beta_{e_4,e_7} \\ 0 & \beta_{e_3,e_6} & \beta_{e_3,e_7} \end{bmatrix} \cdot B$$

$$A = \begin{bmatrix} \alpha_{1,e_1} & \alpha_{1,e_2} & \alpha_{1,e_3} \\ \alpha_{2,e_1} & \alpha_{2,e_2} & \alpha_{2,e_3} \\ \alpha_{3,e_1} & \alpha_{3,e_2} & \alpha_{3,e_3} \end{bmatrix},$$

$$B = \begin{bmatrix} \varepsilon_{e_5,1} & \varepsilon_{e_5,2} & \varepsilon_{e_5,3} \\ \varepsilon_{e_6,1} & \varepsilon_{e_6,2} & \varepsilon_{e_6,3} \\ \varepsilon_{e_7,1} & \varepsilon_{e_7,2} & \varepsilon_{e_7,3} \end{bmatrix}$$

NETWORK CODING SOLUTION EXISTS IF DETERMINANT OF M IS NON-ZERO

- We want $\bar{z} = \bar{x}$
- Choose A to be an identity matrix.
- Choose B to be the inverse of

$$\begin{bmatrix} \beta_{e_1,e_5} & \beta_{e_1,e_4}\beta_{e_4,e_6} & \beta_{e_1,e_4}\beta_{e_4,e_7} \\ \beta_{e_2,e_5} & \beta_{e_2,e_4}\beta_{e_4,e_6} & \beta_{e_2,e_4}\beta_{e_4,e_7} \\ 0 & \beta_{e_3,e_6} & \beta_{e_3,e_7} \end{bmatrix}$$

Connection between an Algebraic Quantity and a Graph Theoretic Tool

- Koetter and Médard (2003): Let a linear network be given with source node v, sink node v', and a desired connection $c = (v, v', \chi(v, v'))$ of rate $R(c)$. The following three statements are equivalent.
  - 1. The connection $c = (v, v', \chi(v, v'))$ is possible.
  - 2. The Min-Cut Max-Flow bound is satisfied
  - 3. The determinant of the $R(c) \times R(c)$ transfer matrix M is non-zero over the ring $\mathrm{F}_2[\ldots, \alpha_{l,e}, \ldots, \beta_{e',e}, \ldots, \varepsilon_{e',j}, \ldots]$

Finding Network Coding Solution

- Koetter and Médard (2003): Greedy Algorithm
- Let a delay-free communication network G and a solvable multicast problem be given with one source and N receivers. Let R be the rate at which the source generates information. There exists a solution to the network coding problem in a finite field F_{2^m} with

$$m \leq \lceil \log_2(NR+1) \rceil$$

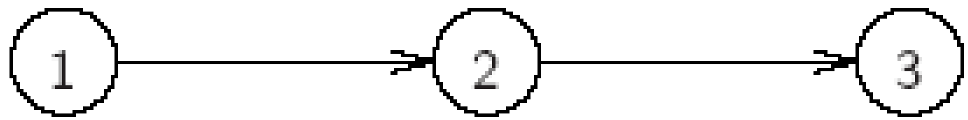
Random Network Coding

Lemma 2.5 *Let P be a nonzero polynomial in $\mathbb{F}[\xi_1, \xi_2, \ldots]$ of degree less than or equal to $d\eta$, in which the largest exponent of any variable ξ_i is at most d. Values for $\xi_1, \xi_2, \ldots$ are chosen independently and uniformly at random from $\mathbb{F}_q \subseteq \mathbb{F}$. The probability that P equals zero is at most $1 - (1 - d/q)^\eta$ for $d < q$.*

--> Choosing the coding coefficient uniformly at random in $\mathbb{F}_q$, with *q* large enough, is sufficient to ensure high probability of decoding at the sink(s)

T. Ho and D.S. Lun, *Network Coding: An Introduction*, Cambridge University Press, 2008

Erasure reliability



ε_{12}: Erasure probability on link (1, 2).
ε_{23}: Erasure probability on link (2, 3).

End-to-end erasure coding:
– Capacity is $(1 - \varepsilon_{12})(1 - \varepsilon_{23})$ packets per unit time.

As two separate channels:
– Capacity is $\min(1 - \varepsilon_{12}, 1 - \varepsilon_{23})$ packets per unit time.
– Can use block erasure coding on each channel. But delay is a problem.

Practical Issues

- Network Delay
- Centralized Knowledge of Graph Topology
- Packet Loss
- Link Failures
- Change in Topology or Capacity

Outline

I. Delay-Throughput trade-offs:
   1. Static ad hoc networks under the physical and protocols models
   2. Mobile ad hoc networks under the physical and protocols models
   3. Distributed MIMO: nodes as antenna arrays

II. Network coding:
   1. Introduction
   2. Application: XOR in the air
   3. Application: Routing and energy efficiency
   4. Application: NC meets TCP

XORs in The Air: Practical Wireless Network Coding

S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard and J. Crowcroft. *XORs in the air: Practical wireless network coding*. In Proceedings of SIGCOMM 2006.

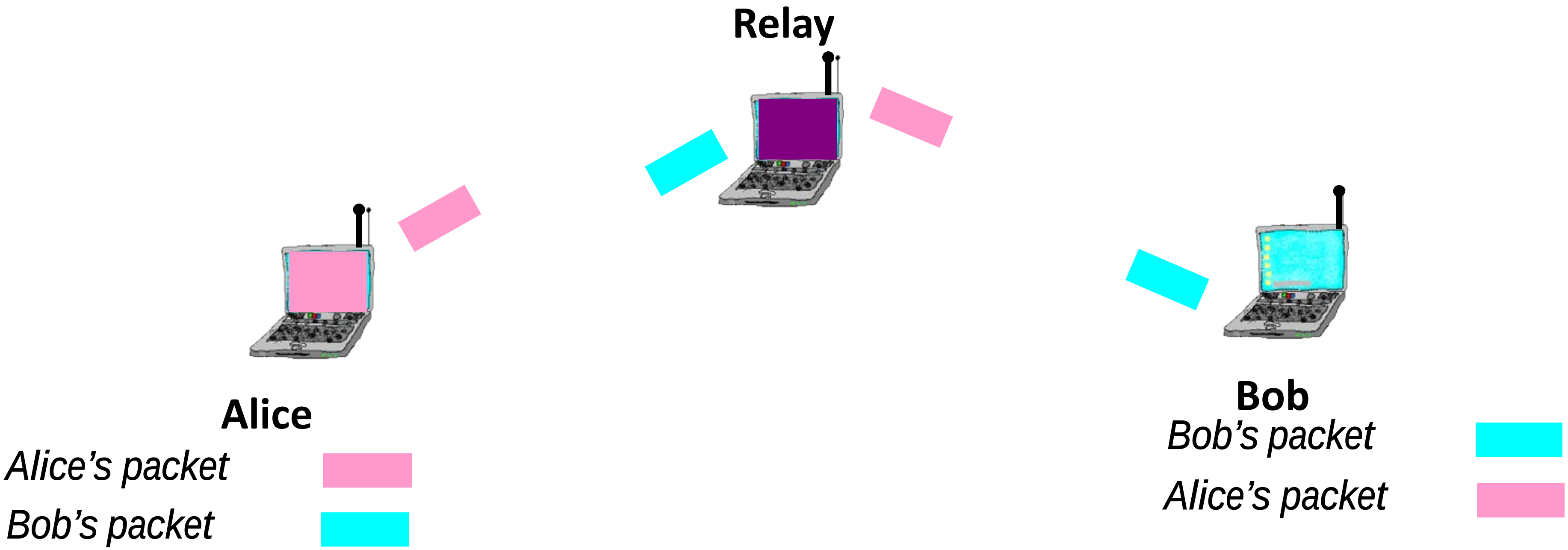
The problem

- Wireless networks are highly resource constrained
  - Bandwidth is the most expensive
  - Power is sometimes an issue too

    --> Serious problems for mesh networks
- How to optimize throughput?
  - Can we send more information?
  - Can we reduce bandwidth requirement?

    --> Do both at the same time?

An information exchange scenario



- Multi-hop unicast requires 4 transmissions
- Can we do better?

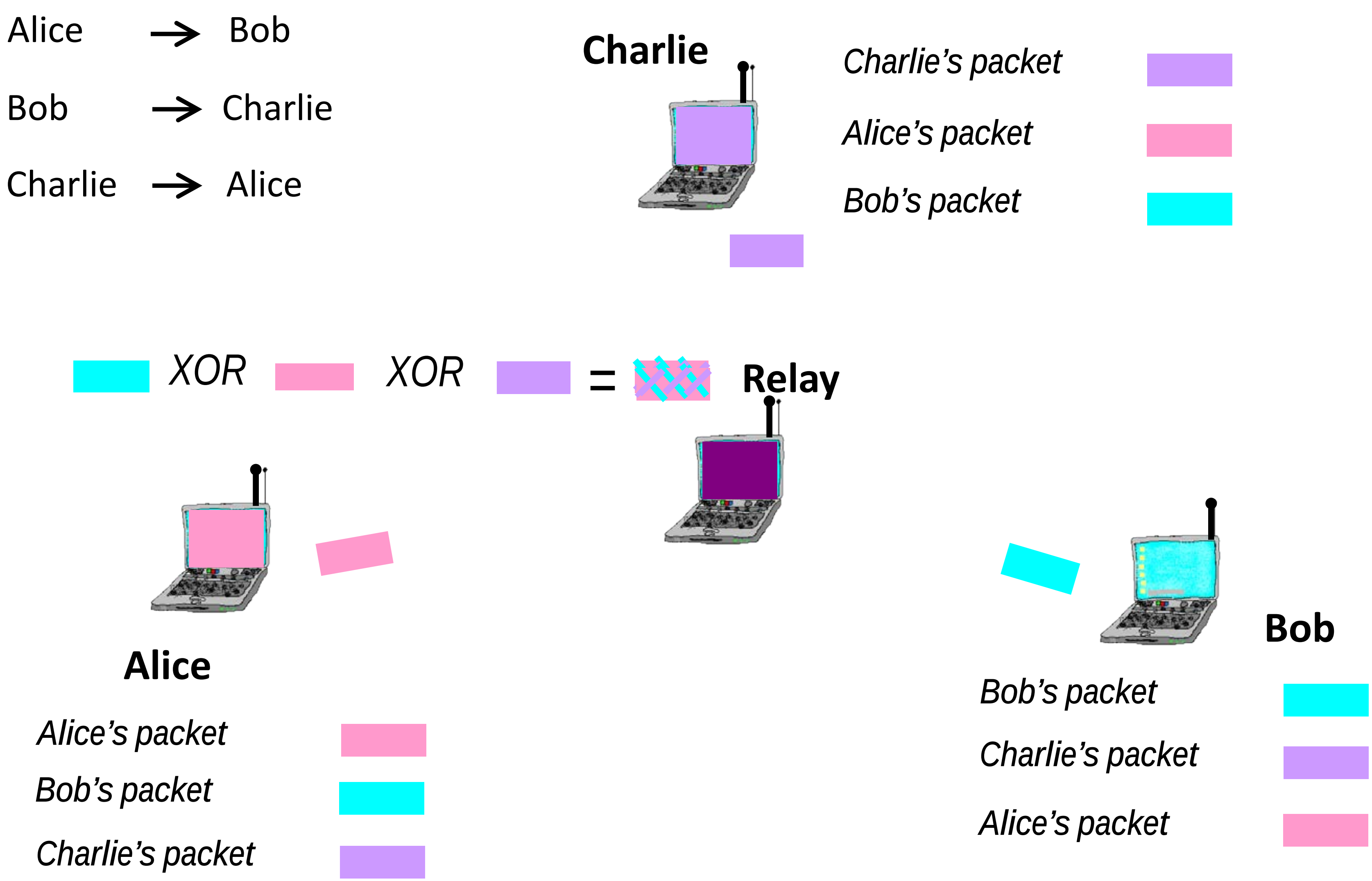
Can Network Coding help? - An idea

3 transmissions instead of 4
→ Saves bandwidth & power
→ 33% throughput increase

The COPE approach

- Considers multiple unicast flows
  - Generalizes the duplex flow scenario
- Opportunistic coding using local info
  - Overhear packets to increase coding gain
  - Online, distributed and deployable
- Emulation and testbed results
  - First real-world implementation

COPE: Opportunistic Coding Protocol



How it works…(Cont.)

- Relay – Encoding
  - Checks packets in queue
  - Combines packets traversing the same three hops in opposite directions
  - Metadata in a header between MAC and IP
  - Broadcast encoded packets
- Alice/Bob – Decoding
  - Keep copies of sent packets
  - Detect the extra header (decoding info)
  - Retrieve the right packet to decode
- Distributed and local action only!

Generalize to COPE

- Nodes snoop on the medium
  - Reception reports to neighbours
- When encoding
  - Identify what packets neighbours have
    - Reception reports and guesses
  - Encode as many packets as possible
    - Provided intended recipients can decode them
- Still distributed and local action only!

The importance of being opportunistic

- Opportunistic coding
  - Only encode if packets in queue
  - No delay penalty
  - Insensitive to flow characteristics

- Opportunistic listening
  - Helps create more coding opportunities

‘Pseudo-broadcast’

- COPE gain is from broadcast medium
- But 802.11 broadcast doesn’t work!
  - No reliability scheme to mask collision loss
  - Send packets at lowest bit rate
  - May actually reduce throughput!
- Pseudo-broadcast
  - Send encoded packets as if unicast
  - Other neighbours overhear
  - Benefit as a unicast packet

Implementation

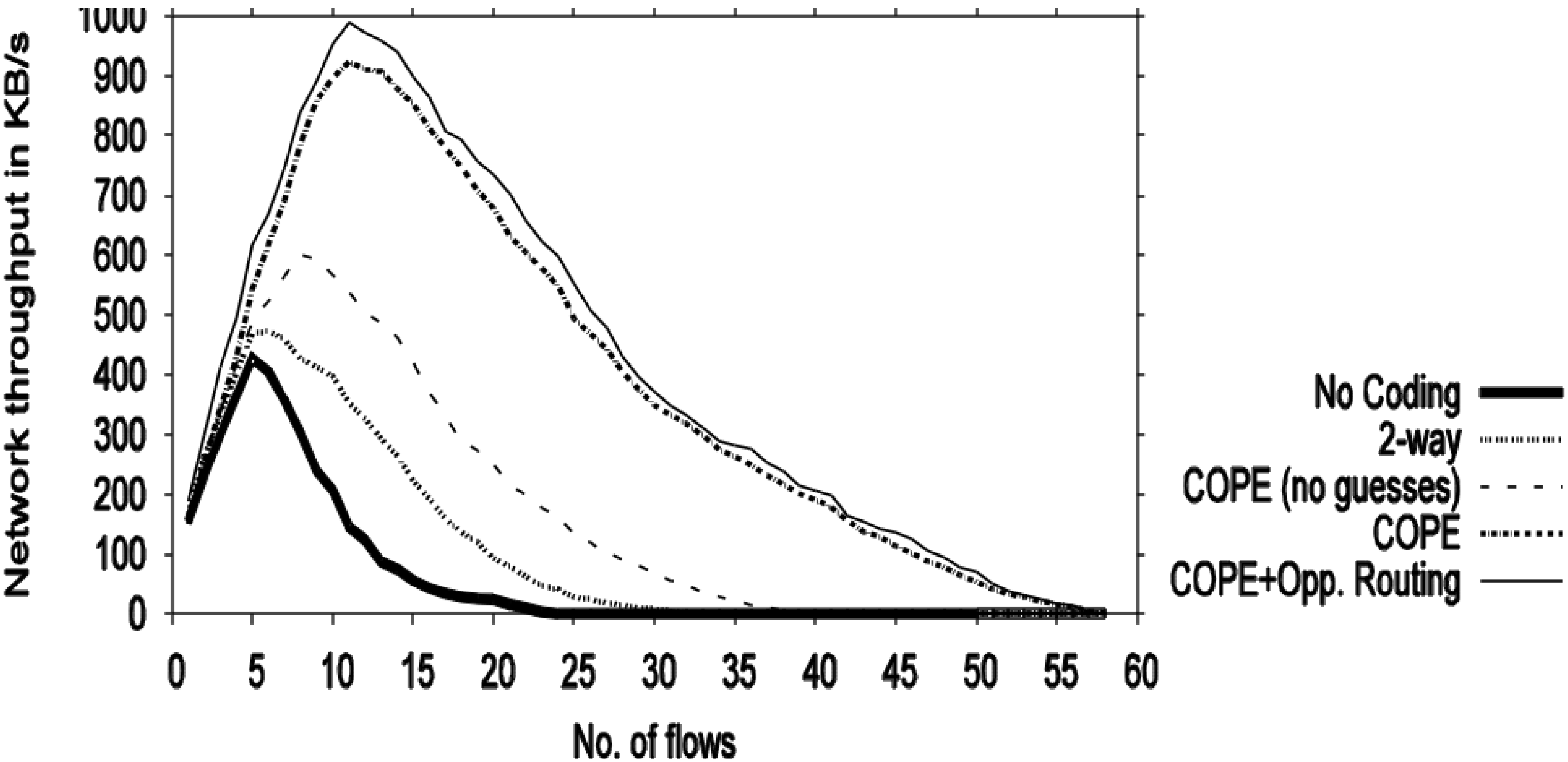
- A shim between MAC and IP
  - Agnostic to protocols above/below
- Emulations
  - General COPE
  - Emsim (part of Emstar) environment
- Testbed
  - Based on the Alice/Bob scenario
  - Extension to Roofnet code (in Click)

Emulation Scenario

- 100 nodes in 800m x 800m
  - Consider range ~50m
- Random senders/receivers
  - Senders always backlogged
  - Bit rate at 11 Mb/s
- Geographic routing
- Metric: end-to-end data traffic throughput over all flows

Emulation performance

Throughput (KB/s)



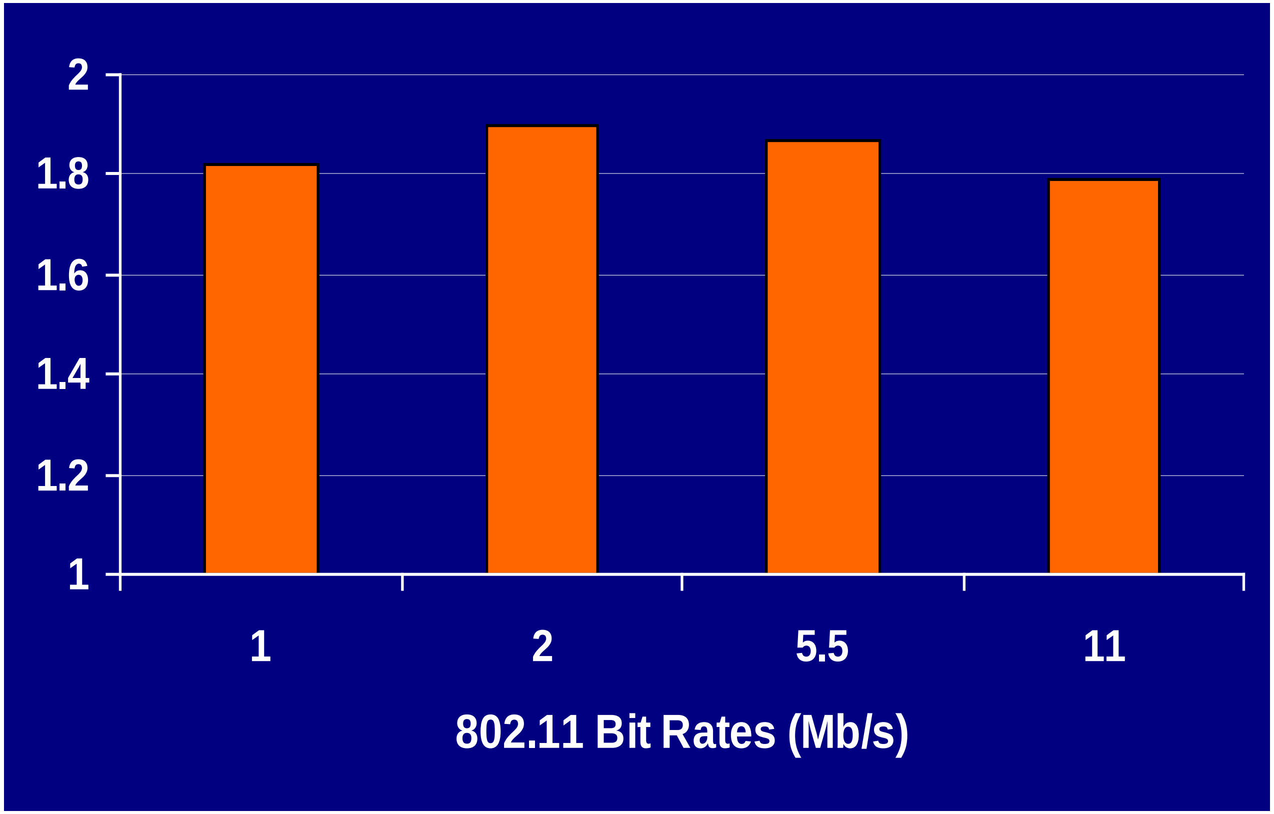
Coding always outperforms no-coding

Testbed setup

- Indoor PCs with 802.11b cards
  - Intersil Prism 2.5 802.11b chipset
  - Connected to omni-directional antenna
  - RTS/CTS disabled
  - 802.11 ad hoc mode
- Randomly chosen 3 nodes from testbed
  - Static routes
  - End nodes send UDP traffic to each other

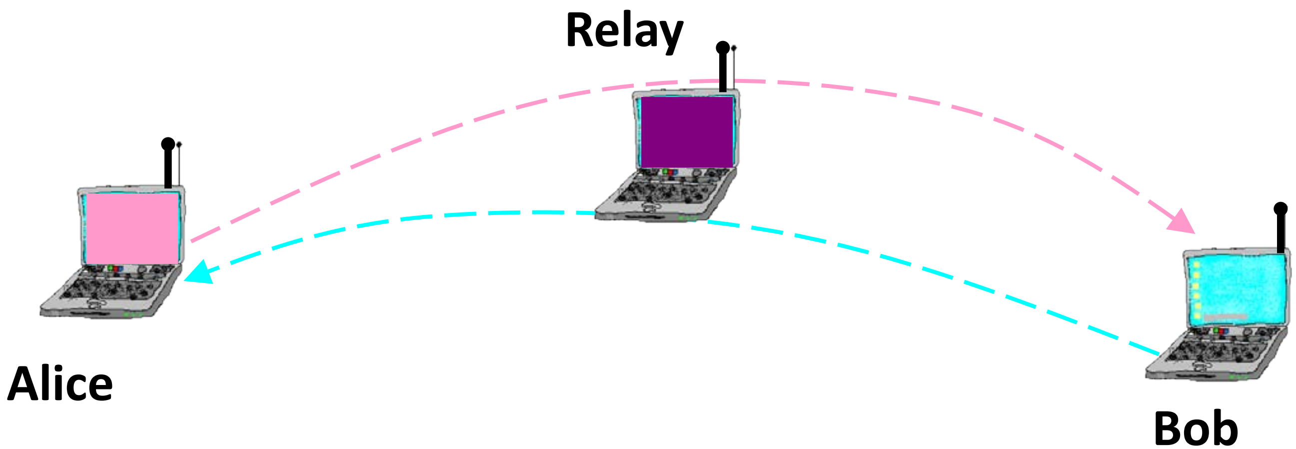
Testbed results

Ratio of Throughput with Coding to No-Coding



Encoding almost doubles the throughput

Why more than 33%?



MAC is fair -> 1/3 BW for each node

- Without coding, relay needs twice as much bandwidth as Alice or Bob
- With coding, all nodes need equal bandwidth

Summary

- Opportunistic approach allows practical integration of network coding into current stack
- Throughput can double in practice
  - Cross-layer effects
  - Congestion plays in our favour
- First implementation of network coding in a wireless environment

Outline

I. Delay-Throughput trade-offs:
   1. Static ad hoc networks under the physical and protocols models
   2. Mobile ad hoc networks under the physical and protocols models
   3. Distributed MIMO: nodes as antenna arrays

II. Network coding:
   1. Introduction
   2. Application: COPE
   3. Application: Routing and energy efficiency
   4. Application: NC meets TCP

Efficient Network Coded Data Transmission in DTN

Yunfeng Lin, Baochun Li, Ben Liang, "Efficient Network Coded Data Transmissions in Disruption Tolerant Networks," in the Proceedings of IEEE INFOCOM 2008, Phoenix, Arizona, April 2008.

Yunfeng Lin, Baochun Li, Ben Liang, "Stochastic Analysis of Network Coding in Epidemic Routing," in IEEE Journal on Selected Areas in Communications, Special Issue on Delay and Disruption Tolerant Wireless Communication, Vol. 26, No. 5, pp. 794-808, June 2008.

Motivation – Constraints in DTN

- Opportunistic connections between nodes
- Nodes have limited transmission capabilities
- Buffer space limitations
- Battery power limited
- Nodes are mobile
- Delay in packet delivery will be large
- Node density is low

Network model

- Settings:
  - only a single unicast session
  - λ: average number of meetings a node has per time unit
  - sparse DTN: λ=Nβ remains constant as N increases
    --> the network is connectivity-limited
  - mobility model: fast and uniform (RW, RWP,…)

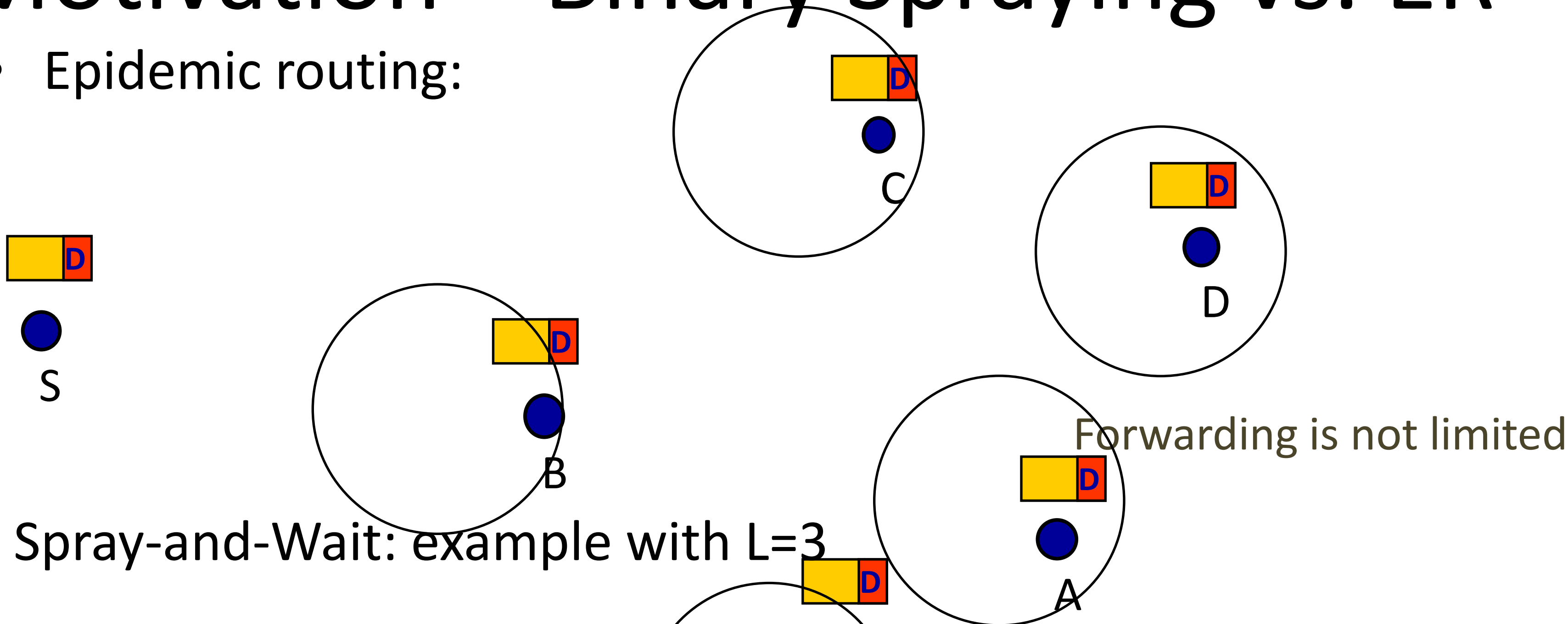
--> routing strategies must permit timely delivery of information to a certain destination with high probability: use of replication

- replication leads to energy and memory consumption
- finite duration of radio contacts --> file split into packets

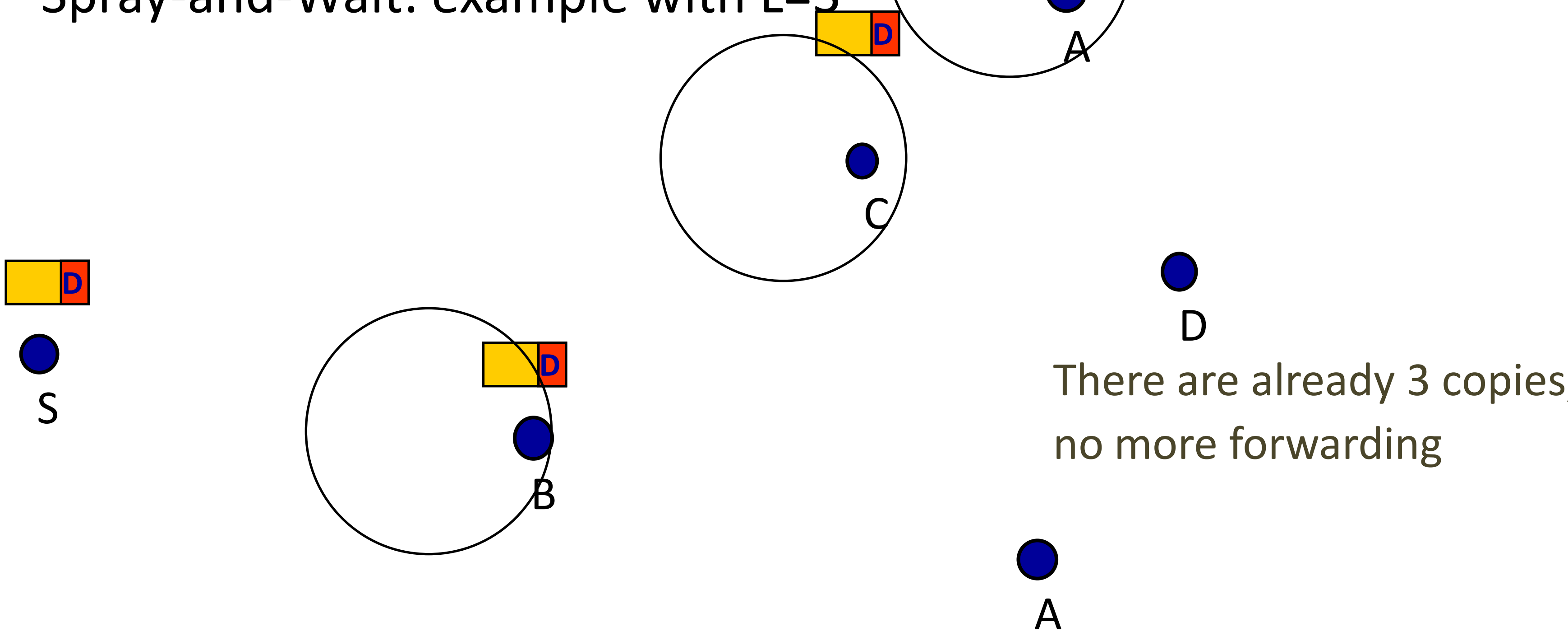
- **Objective**: optimize the file transfer from S to D by minimizing both its delay, the memory and energy required by the store and forward process

Motivation – Binary Spraying vs. ER

- Epidemic routing:



- Spray-and-Wait: example with L=3

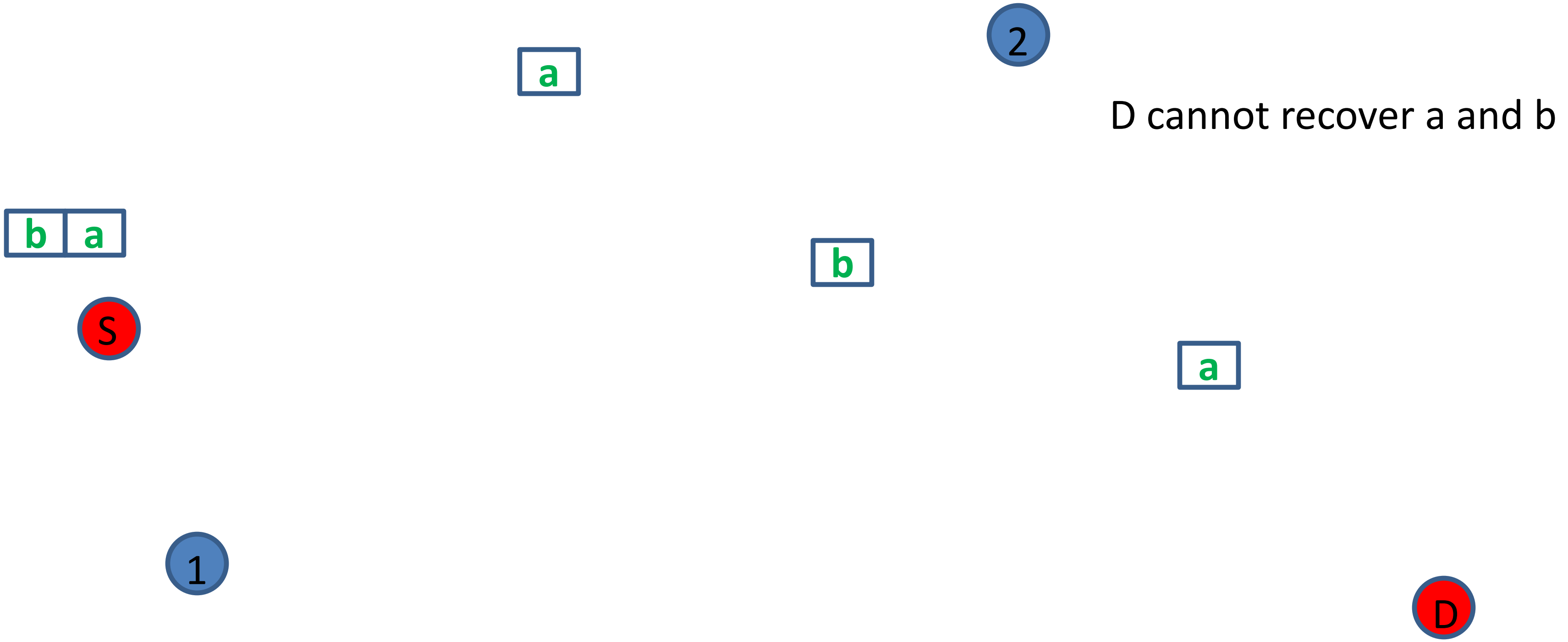


Binary Spraying Vs ER (cont'd)

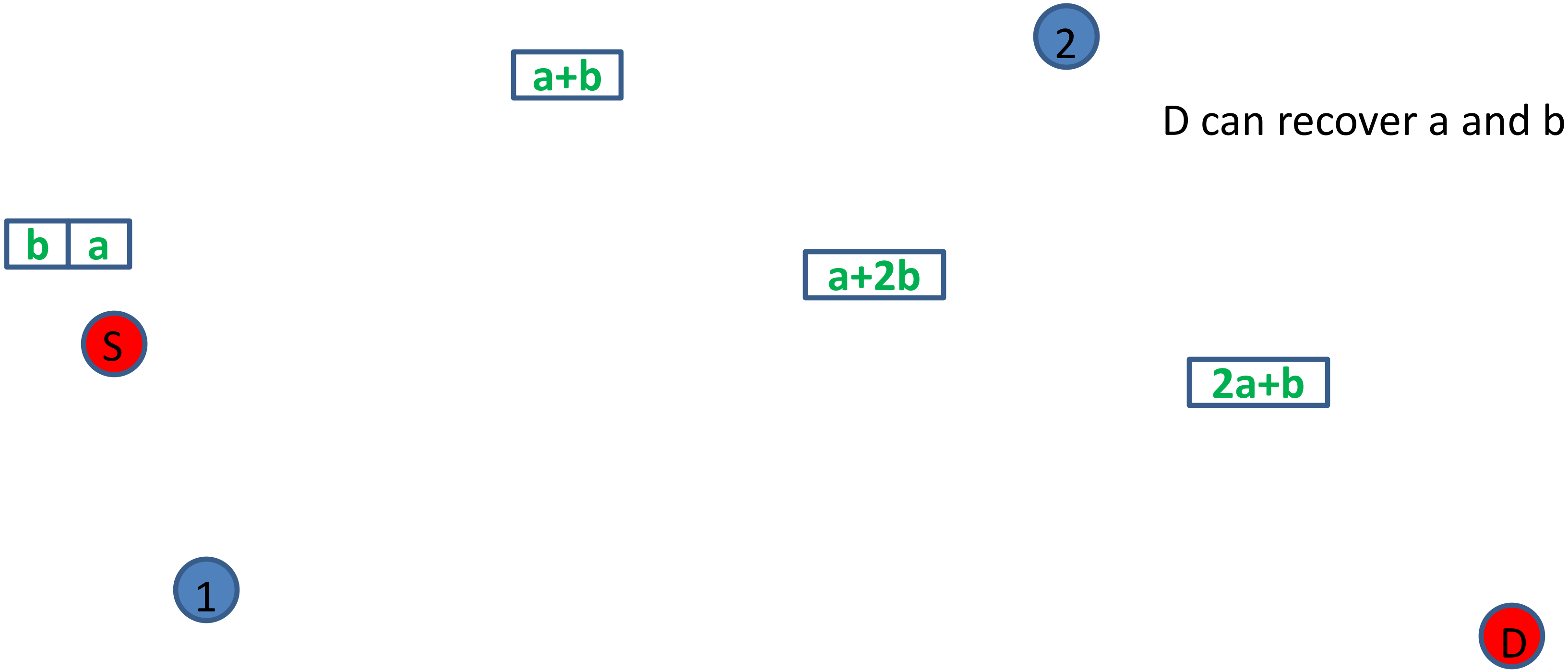
- Epidemic routing:
  - no limit on the number of transmissions (≤ nb of pkts . N)
  - mean time for delivery of one packet: $\leq \log_2(N)$

- Spray-and-Wait:
  - number of transmissions ≤ nb of pkts . L
  - mean time for delivery of one packet: $\leq \log_2(L)+N/L$

Motivation – NC Vs Replication

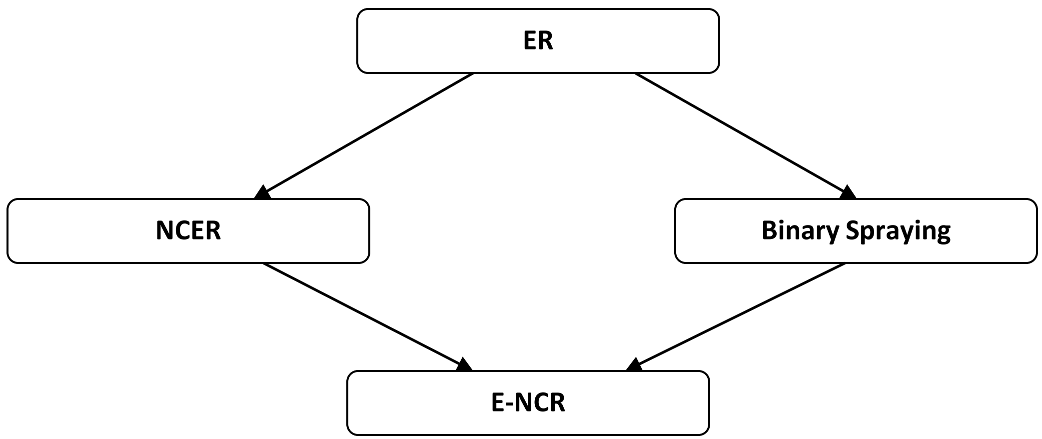


Motivation – NC Vs Replication



Protocol - Principle

- This protocol called the E-NCR, is a combination of Network coding and Binary spraying.

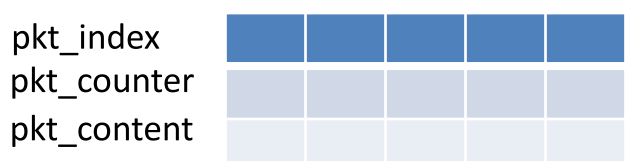


NCER – Network Coding based Epidemic routing
ER – Epidemic Routing
E-NCR - Efficient Network coding based routing

Protocol - Assumptions

- There is one source S with *K* info packets to be transmitted, *n* relay nodes and a destination D
- For every opportunistic contact, only one packet can be transmitted.
- Relay nodes have buffer space B, defined as $1 \leq B \leq K$
- No other back-ground traffic
- A packet in the buffer of a node is purged as soon as an ACK is received from D or the Time-to-live field reaches zero.

Buffer structure:



E-NCR: an example

K=2
K'=3
L=7

Buffer structure:
pkt_index
pkt_counter
pkt_content

Time	Node 1	Node 2	Buffer content Node 1	Buffer content Node 2
0	S		1, 2, 3 7, 7, 7 d=a+b, e=2a+3b, f=a+2b	
1	S	R1	1, 2, 3 4, 7, 7 d, e, f	1 3 d
2	S	R2	1, 2, 3 4, 4, 7 d, e, f	2 3 e
3	R1	R2	1, 2 2, 1 d, e	2, 1 2, 1 e, d
4	R2	R3	2, 1 1, 1 e, d	2 1 3d+5e

Protocol - Description

SOURCE-RELAY:

```
K' = K + some more encoded packets
L = c * log k, where c is some constant
i = 0;
S = K';
do
{
   if(detect any node and <i,l> not already there with that node)
   {
      send an encoded packet <i, L, co-efficients, packet>
       i++;
   }

}while(S != i);
```

Protocol - Description

RELAY-RELAY, SENDER SIDE:

```
do
{
 if(detect any node X)
 {
     get spray list of X; //list element is a tuple <i, l>,where i is index of packet, ‘l’ is the
                //remaining spray count
     do
     {
            compare this->spraylist with x->spraylist;
     if(any this->spraylist-><i, l> such that l >=0 and i does not exist in x->spraylist)
            {
            send encoded packet <i, floor(l/2)> to node x;
            update tuple <i, l> to <i, ceil(l/2)>;
            }
     }while(end of x->spraylist);
 }
}while(true);
```

Protocol - Description

RELAY-RELAY, RECEIVER SIDE:

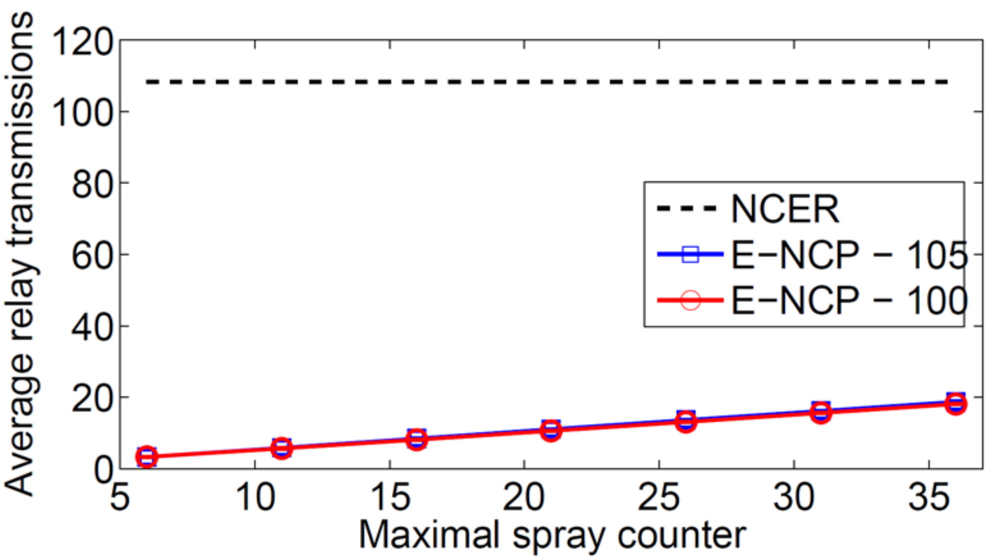
```
if(packet received)
{
    if(buffer size == max_buffer_size)
    {
    encode incoming packet with all
    packets in list;
    }
    else
    {
    place packet in free slot;
    }
    add <i,l> of incoming packet to spray
    list;
}
```

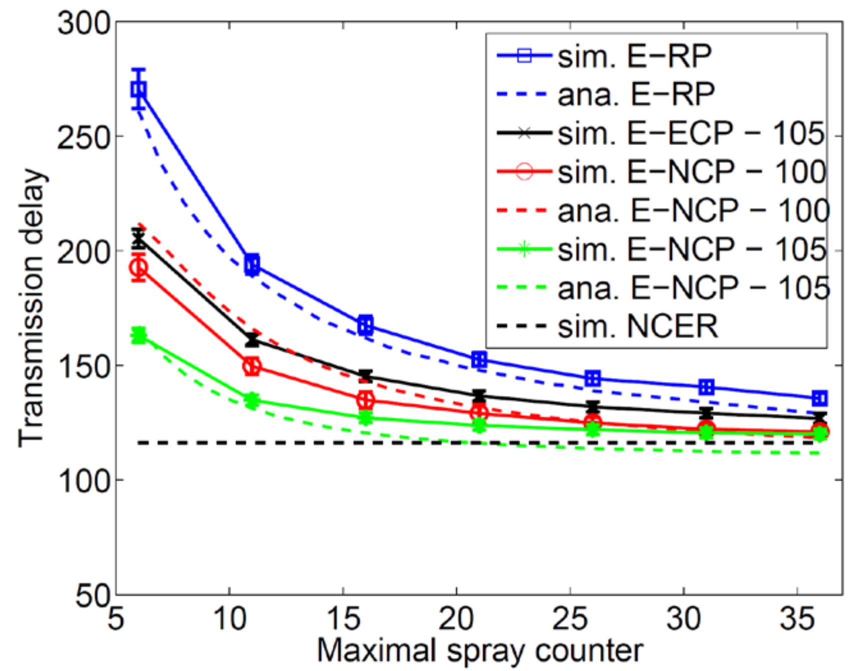
DESTINATION:

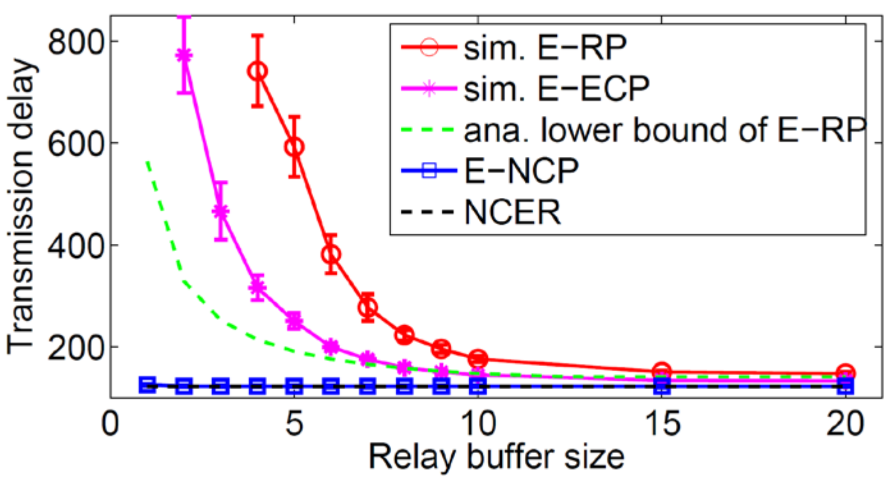
```
do
{
    if(got a packet)
    {
        add to packet list
        try to decode list of packets;
        if(decode possible)
        {
            exit loop;
        }
    }

}while(true);
```

Performance







Some Limitations

- Destination has to wait till minimum of K encoded packets are received
- Some packets which have linear dependence could arise during encoding at relays.

Outline

I. Delay-Throughput trade-offs:
   1. Static ad hoc networks under the physical and protocols models
   2. Mobile ad hoc networks under the physical and protocols models
   3. Distributed MIMO: nodes as antenna arrays

II. Network coding:
   1. Introduction
   2. Application: COPE
   3. Application: Routing and energy efficiency
   4. Application: NC meets TCP

Network Coding Meets TCP

J. K. Sundararajan, D. Shah, M. Médard, S. Jakubczak, M. Mitzenmacher and J. Barros, "Network Coding Meets TCP: Theory and Implementation", Proceedings of the IEEE, pp. 490 – 512, March 2011.

J. Sundararajan, D. Shah, M. Medard, M. Mitzenmacher and J. Barros, "Network coding meets TCP", INFOCOM 2009, April 2009.

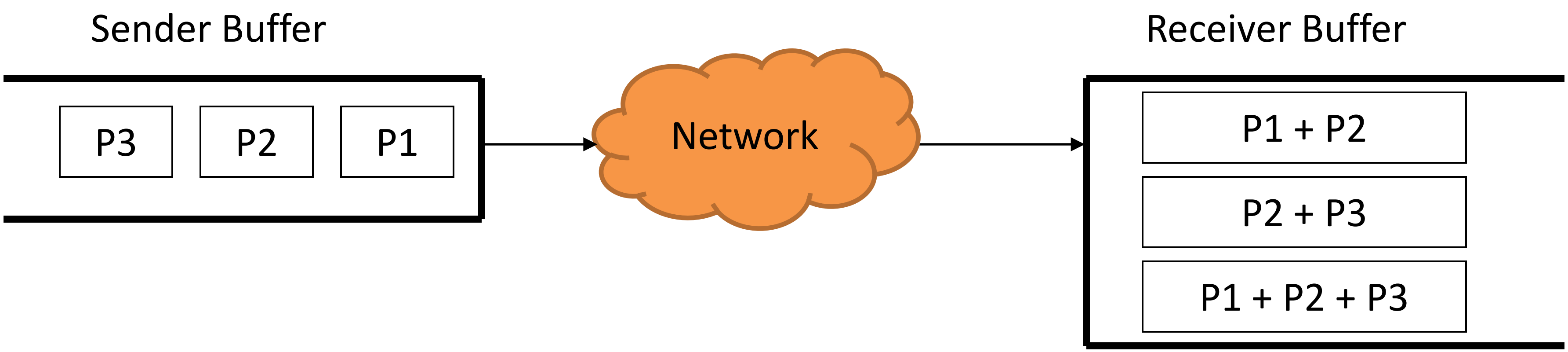
Practice?

- Will network coding achieve wide use in practice, or just a mathematical toy?
  - Jury is still out… but lots of believers.
    - Lots of theory, projects.
    - Avalanche, COPE, MORE,…
- Potential problem: incremental deployment / backward compatibility.
  - Standard problem for anything new.

TCP and Coding

- For incremental deployment, best to be compatible or friendly with TCP.
- Not easy; TCP not designed for coding.
- TCP *combines* reliability and congestion control; with coding, you don't want reliability.
  – But still the need for congestion control.

The Problem



- Can't acknowledge a packet until you can decode.
- Usually, decoding requires a number of packets.
- Code / acknowledge over small blocks to avoid delay, manage complexity.

Compare to ARQ

Context: Reliable communication over a (wireless) network of packet erasure channels

ARQ

- Retransmit lost packets
- Low delay, queue size
- Streaming, not blocks
- Not efficient on broadcast links
- Link-by-link ARQ does not achieve network multicast capacity.

Network Coding

- Transmit linear combinations of packets
- Achieves min-cut multicast capacity
- Extends to broadcast links
- Congestion control requires feedback
- Decoding delay: block-based

Goals

- Devise a system that behaves as close to TCP as possible, while masking non-congestion wireless losses from congestion control where possible.
  - Standard TCP/wireless problem.
- Stream-based, not block-based.
- Low delay.
- Focus on wireless setting.
  - Where network coding can offer biggest benefits.
  - Not necessarily a universal solution.

Main Idea : Coding ACKs

- What does it mean to "see" a packet?
- Standard notion: we have a copy of the packet.
  - Doesn't work well in coding setting.
  - Implies must decode to see a packet.
- New definition: we have a packet that will allow us to decode *once enough* useful packets arrive.
  - Packet is useful if linearly independent.
  - When enough useful packets arrive can decode.

Coding ACKs

- For a message of size *n*, need *n* useful packets.
- Each coded packet corresponds to a degree of freedom.
- *Instead of acknowledging individual packets, acknowledge newly arrived degrees of freedom.*

Coding ACKs

Original message : $p_1, p_2, p_3 \ldots$

Coded Packets

$4p_1 + 2p_2 + 5p_3$

c_1	4 2 5 0 0 0 0
c_2	3 1 2 5 0 0 0
c_3	1 2 3 4 1 0 0
c_4	3 3 1 2 1 0 0
c_5	1 2 5 4 5 0 0

$$\begin{pmatrix} 4 & 2 & 5 & 0 & 0 & 0 & 0 \\ 3 & 1 & 2 & 5 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 1 & 0 & 0 \\ 3 & 3 & 1 & 2 & 1 & 0 & 0 \\ 1 & 2 & 5 & 4 & 5 & 0 & 0 \end{pmatrix}$$

Coding ACKs

Original message : p_1, p_2, p_3…

Coded Packets

$4p_1 + 2p_2 + 5p_3$

c_1	4 2 5 0 0 0 0
c_2	3 1 2 5 0 0 0
c_3	1 2 3 4 1 0 0
c_4	3 3 1 2 1 0 0
c_5	1 2 5 4 5 0 0

$$\begin{pmatrix} 4 & 2 & 5 & 0 & 0 & 0 & 0 \\ 3 & 1 & 2 & 5 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 1 & 0 & 0 \\ 3 & 3 & 1 & 2 & 1 & 0 & 0 \\ 1 & 2 & 5 & 4 & 5 & 0 & 0 \end{pmatrix}$$

When c_1 comes in, you've "seen" packet 1; eventually you'll be able to decode it. And so on…

Coding ACKs

Original message : $p_1, p_2, p_3\ldots$

Coded Packets

$4p_1 + 2p_2 + 5p_3$

c_1	4 2 5 0 0 0 0
c_2	3 1 2 5 0 0 0
c_3	1 2 3 4 1 0 0
c_4	3 3 1 2 1 0 0
c_5	1 2 5 4 5 0 0

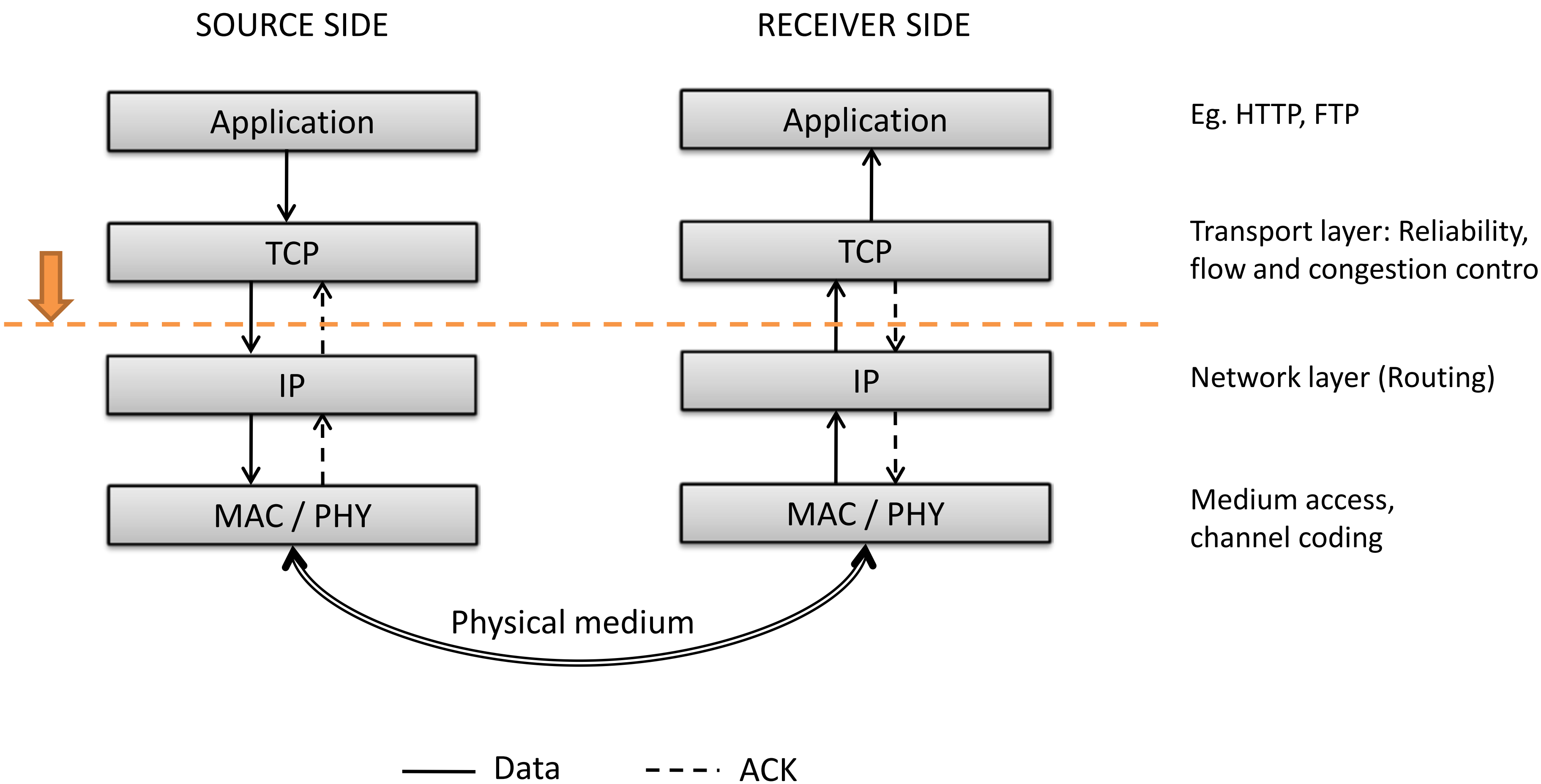
$$\begin{pmatrix} 1 & 4 & 5 & 3 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 & 6 & 0 & 0 \\ 0 & 0 & 1 & 6 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Use Gaussian elimination as packets arrive to check for a new seen packet.

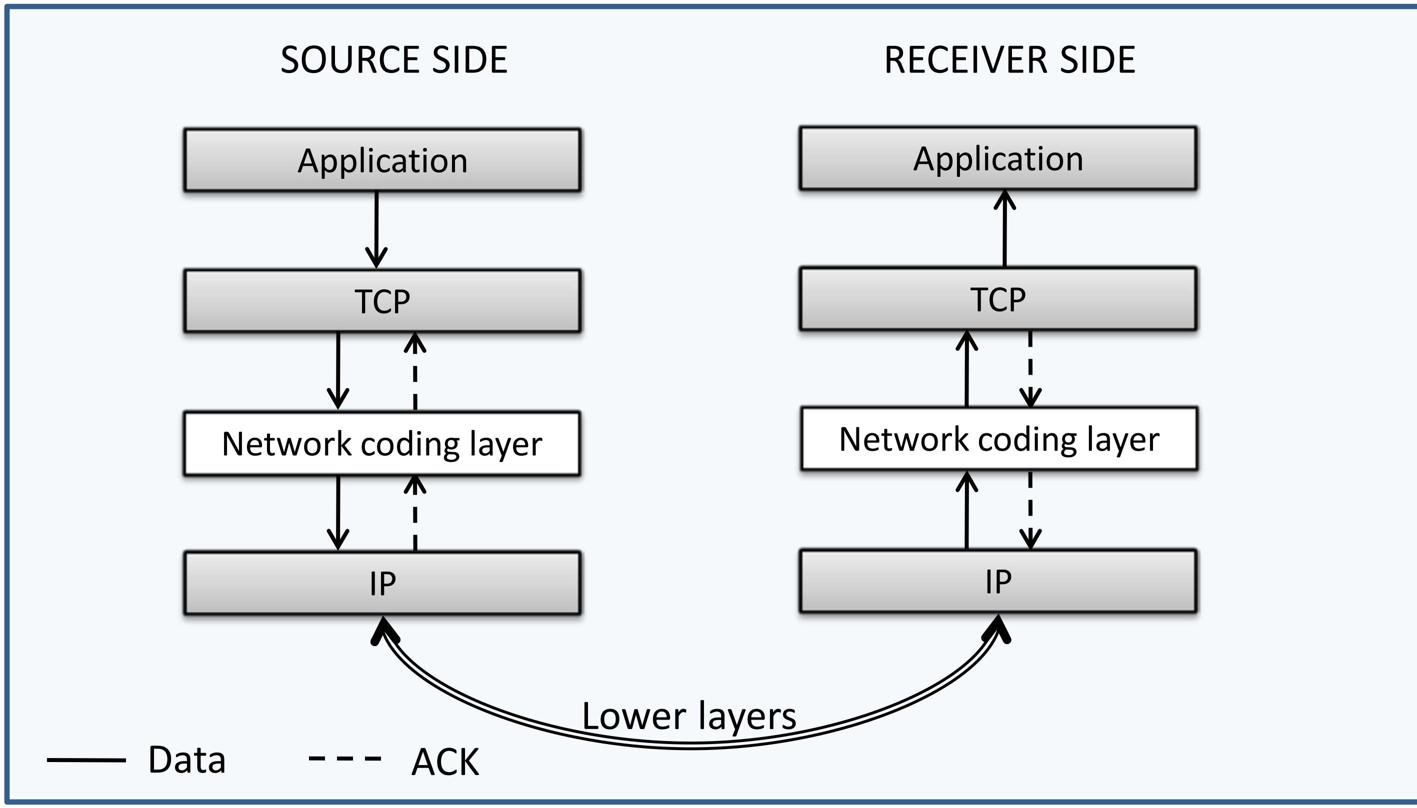
Formal Definition

- A node has *seen* a packet p_k if it can compute a linear combination p_k+q where q is a linear combination of packets with index larger than k.
- When all packets have been seen, decoding is possible.

Layered Architecture



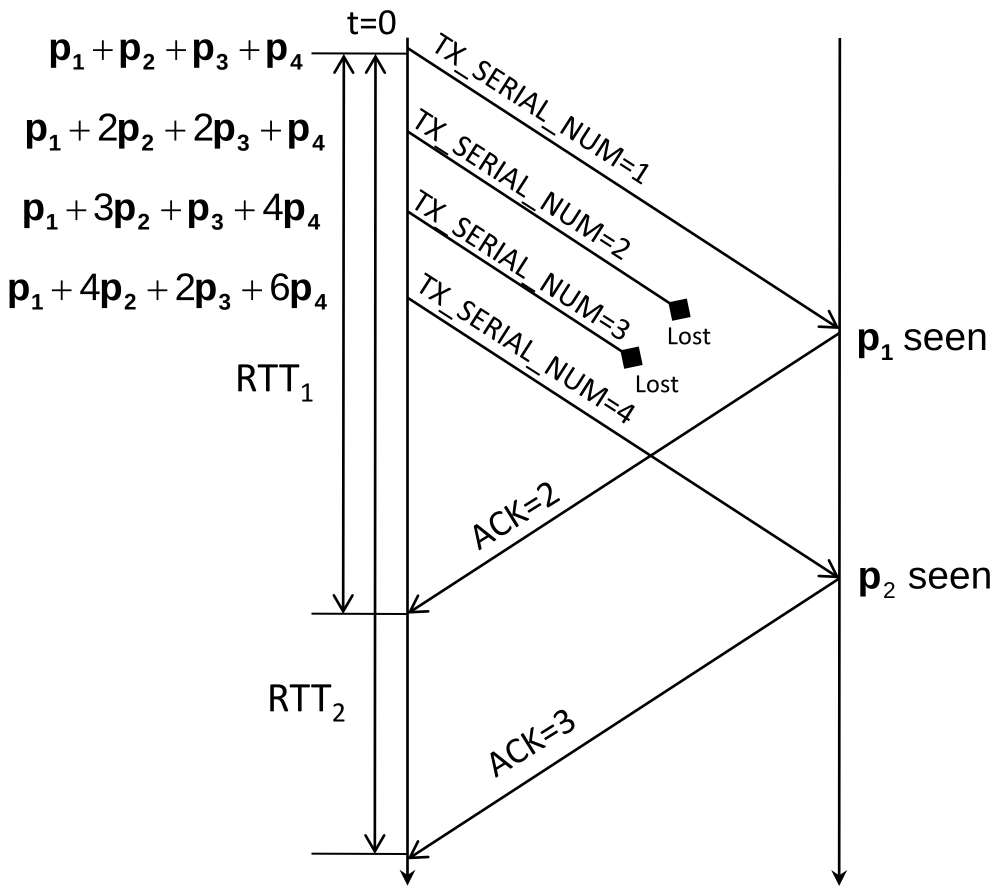
TCP using Network Coding



The Sender Module

- Buffers packets in the current window from the TCP source, sends linear combinations.
- Need for redundancy factor *R.*
  - Sending rate should account for loss rate.
  - Send a constant factor more packets.
  - Open issue : determine *R* dynamically?

Measurement of RTTs



© by Michael Mitzenmacher

The Receiver Module

- Acknowledgment: ACK a packet upon seeing it (even before it is decoded).
- With high probability (if field size is large), every random linear combination will cause *next unseen* packet to be seen.
- Buffer incoming linear combinations until they can be decoded.
  - Possibly can decode early.
  - Interesting design tradeoff for future work.
- Upon decoding, deliver the packets to the TCP sink.

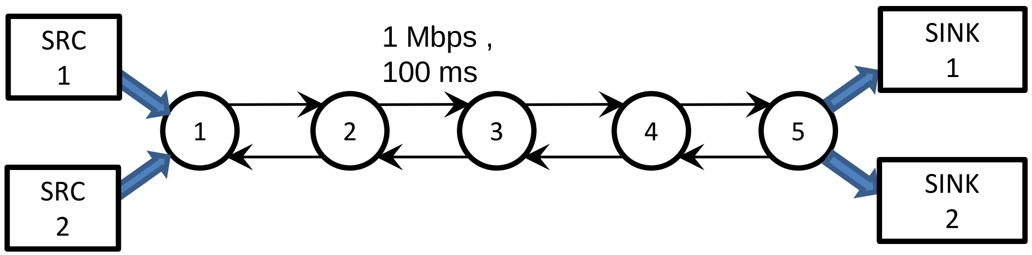
Redundancy

- Too low R
  - TCP times out and backs off drastically.
- Too high R
  - Losses recovered – TCP window advances smoothly.
  - Throughput reduced due to low code rate.
  - Congestion increases.
- Right R is $1/(1-p)$, where p is the loss rate.

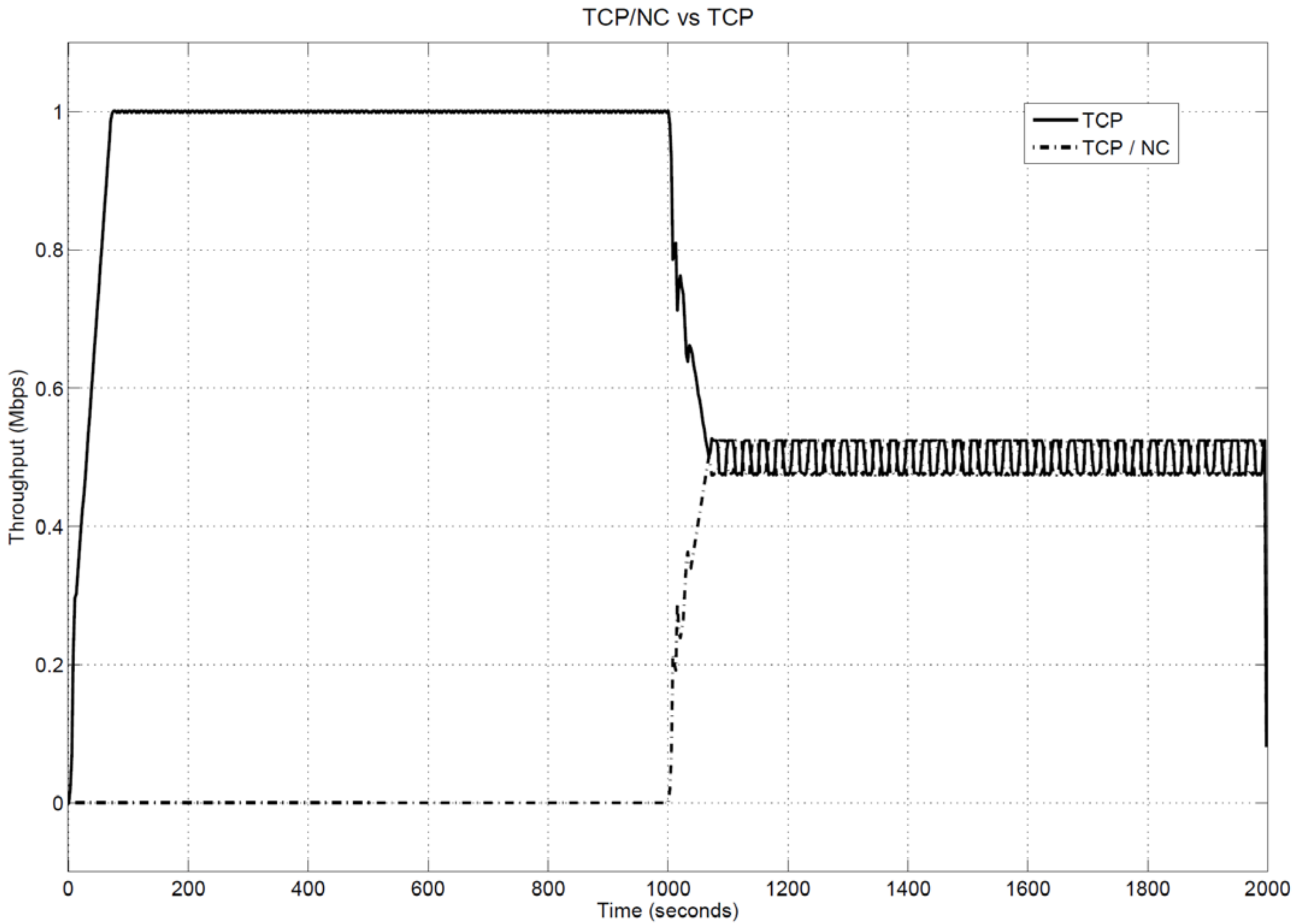
Which TCP to Use?

- Use redundancy to match sending rate to desired data rate.
  - Masking wireless losses not due to congestion.
  - TCP Reno reacts to losses; does not seem suitable here.
    - Continuing work – make this approach TCP Reno compatible.
- Instead use TCP Vegas.
  - Sets window based on Round Trip Times.
  - We use RTTs not of packets, but of degrees of freedom.

Some Simulations

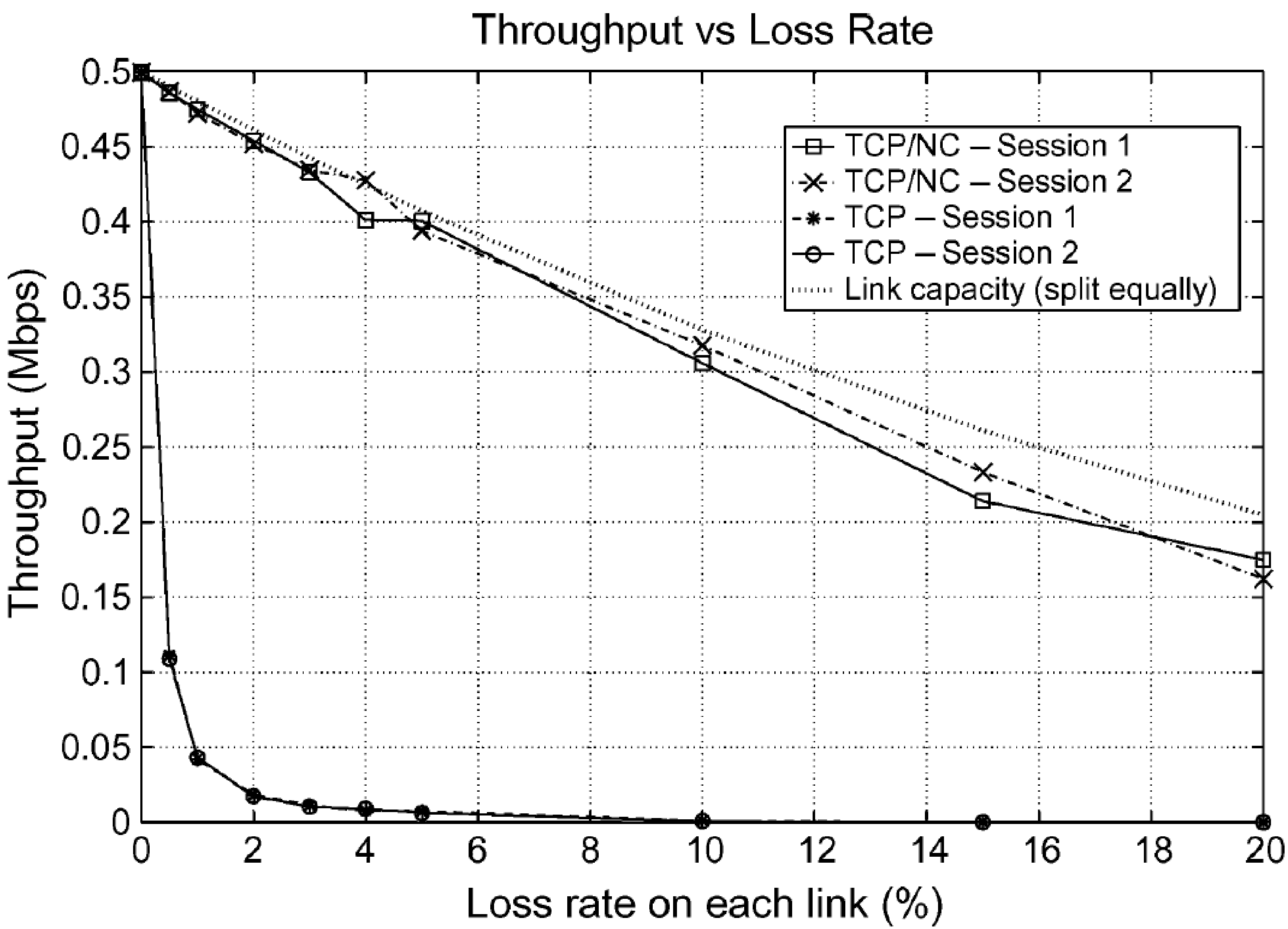


Fairness



0% Loss Rate, Redundancy 1

Resilience to Losses



Conclusions

- New coding layer proposed between TCP and IP.
- Novel ACK mechanism provides clean interface between network coding and existing congestion control protocols.
- Ideas also work with intermediate node coding.
- Possible extensions to **multipath TCP** and to multicast sessions.
- Not a final solution, but a step towards realizing the potential of network coding in practice.
  - Proof of concept ; theory.
  - Next stage: deployments underway.