NETWORK CODING

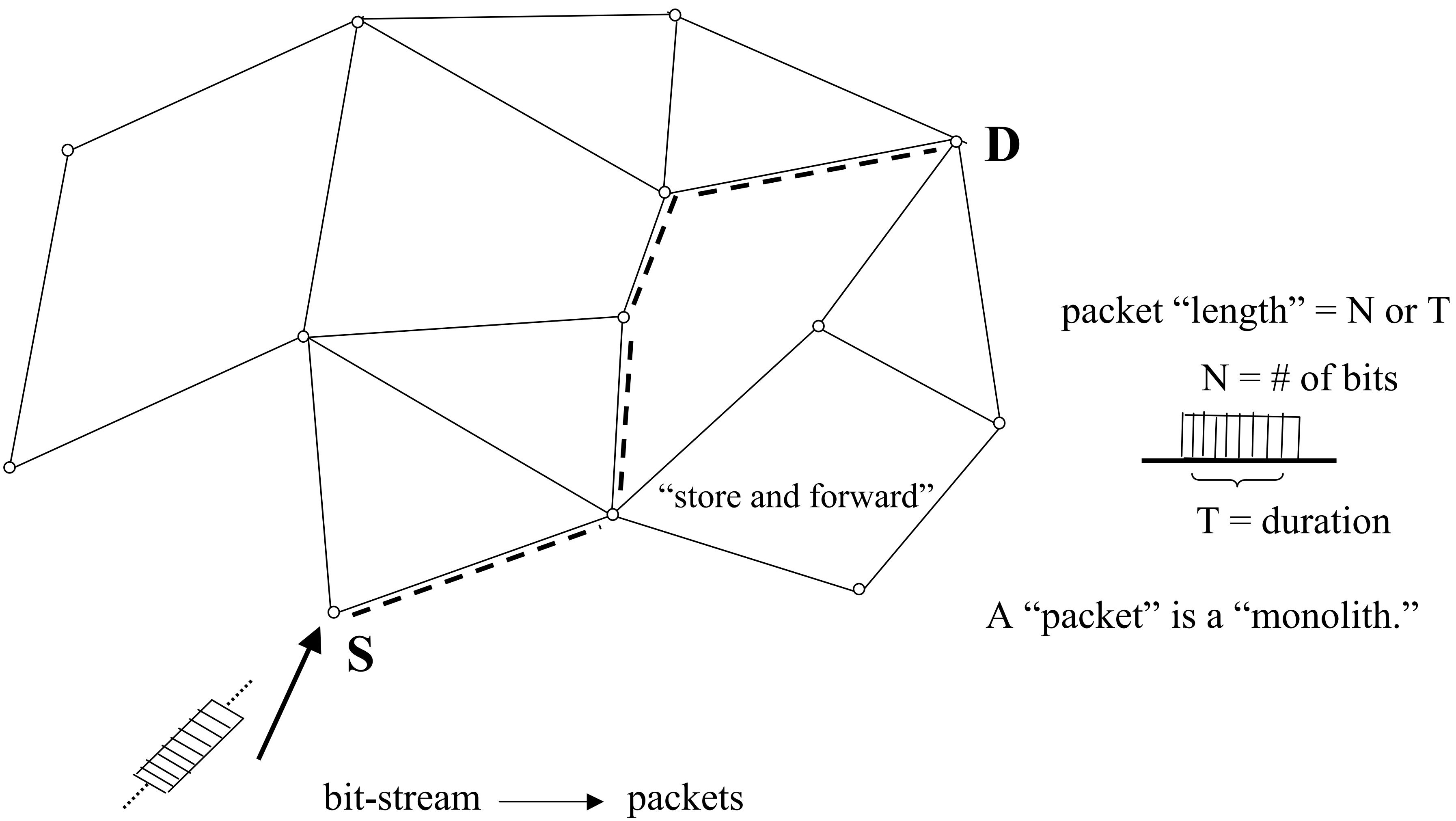
- A NEW PARADIGM FOR NETWORKING -

February 29, 2008
University of Minnesota

Anthony Ephremides

University of Maryland

THE CURRENT PARADIGM

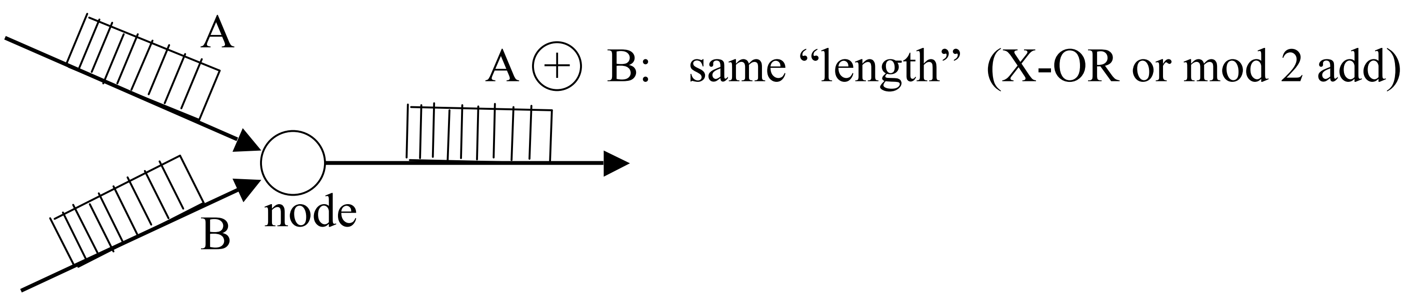


A "packet" is a "monolith."

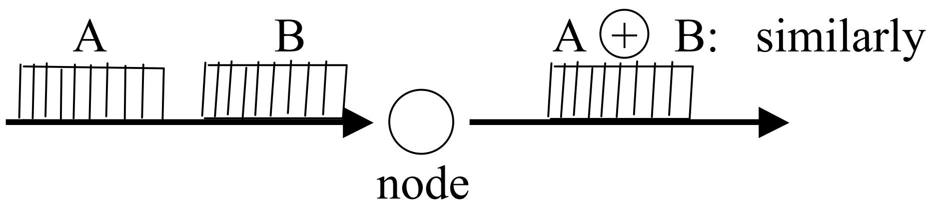
WHAT IS WRONG WITH IT?

- "ORIGINAL SIN" of networking : Layering
- Physical layer was forgotten (or "Separated")
- Recall: packets consist of bits (a simple and obvious truth)
- Bits can be operated upon

(i)

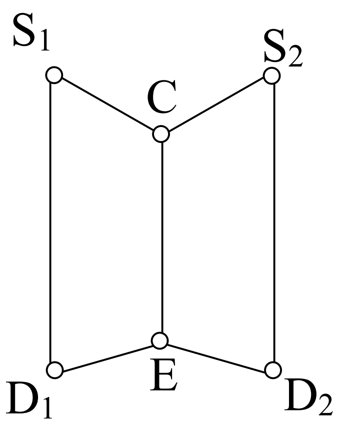


(ii)



- Why may this be good?
- Because it permits the output to be a function of the input!

THE "BUTTERFLY" EXAMPLE (~1999)



- S_1 wants to deliver packet A to both D_1 and D_2
- S_2 wants to deliver packet B to both D_1 and D_2
- Each link can carry one packet in each slot
  (capacity of each link = 1)

Q: How many slots are needed to complete this delivery under "store-and-forward"?

1: - S_1 sends A to C and D_1 (over S_1C and S_1D_1 respectively)
   - S_2 sends B to C and D_2 (over S_2C and S_2D_2 respectively)

2: - C sends A to E

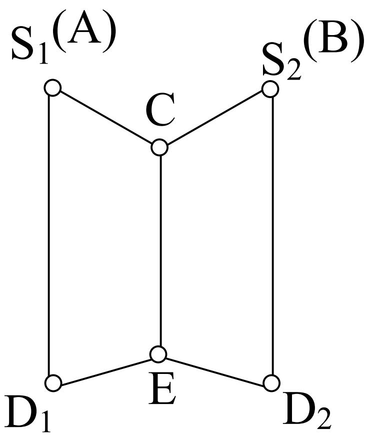
3: - C sends B to E
   - E sends A to D_2

4: - E sends B to D_1

Answer: FOUR (4) SLOTS

ALTERNATIVE

Q: How many slots are needed to complete the same delivery under, so called, "**NETWORK CODING**"?



1. - S_1 sends A to C and D_1 (over S_1C and S_1D_1, respectively)
   - S_2 sends B to C and D_2 (over S_2C and S_2D_2, respectively)
2. - C sends $A \oplus B$ to E
3. - E sends $A \oplus B$ to D_1 and D_2 (over ED_1, and ED_2, respectively)

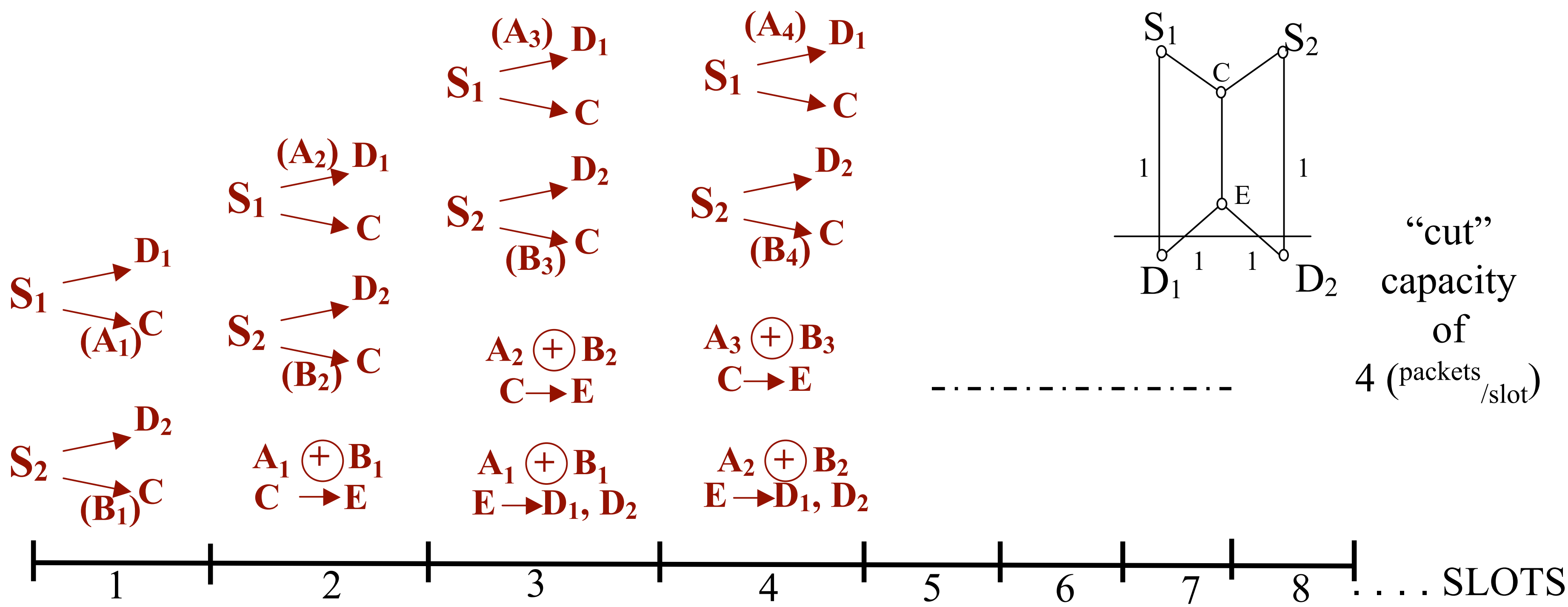
(D1 and D2 recover the missing packet by "X-OR" ing $A \oplus B$ with the one they already have)

Answer: THREE (3) SLOTS

Savings: 25%

Note: THIS WAS "MULTICASTING" (i.e., each packet had multiple destinations)

SUSTAINED OPERATION
("Capacity"-Achieving)



In steady-state, in each slot there are two packets received by each destination, e.g.

Slot 3: D_1 receives A_3 and B_1
D_2 receives B_3 and A_1
Slot 4: D_1 receives A_4 and B_2
D_2 receives B_4 and A_2

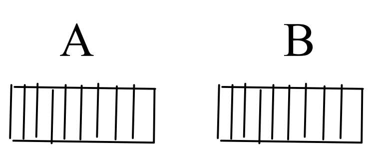
} i.e. 4 packets/slot

<u>THIS IS NOT ALL THERE IS</u>

- Max-flow Limit can be achieved by network coding in “arbitrary” <u>multicast</u> networks

- “ALPHABET” size: each symbol in the packet need not be a single bit, but, rather, one of Q different values (strings of bits)

<u>REMARKABLE RESULT</u>: It is sufficient to combine packets linearly with randomly chosen co-efficients to achieve this capacity result



A = 00110100

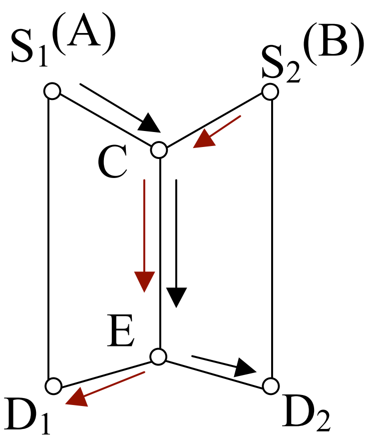
B = 10100010

→ **a A $\oplus$ b B where a, b = 0 or 1**

(X-OR uses a=b=1)

10010110

WHAT ABOUT UNICAST?



Now - S_1 wants to deliver A to D_2 only

- S_2 wants to deliver B to D_1 only

Network Coding

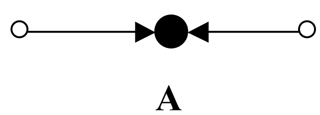
1. $S_1 \xrightarrow{A} C, D_1$, $S_2 \xrightarrow{B} C$, D_2
2. $C \xrightarrow{A \oplus B} E$
3. $E \xrightarrow{A \oplus B} D_1$, D_2

Store-and-Forward

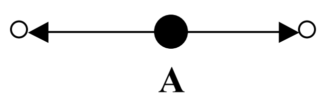
1. $S_1 \xrightarrow{A} C$, $S_2 \xrightarrow{B} C$
2. $C \xrightarrow{A} E$
3. $C \xrightarrow{B} E$, $E \xrightarrow{A} D_2$
4. $E \xrightarrow{B} D_1$

But not capacity achieving
(i.e. only 2 packets/slot)
(Back-Pressure algorithm)

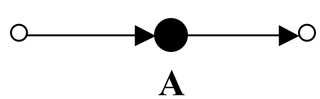
AND NOW WIRELESS

- What is different?
- 

  X

  Node cannot receive simultaneously two different messages
  (actually……it can, but…..)
- 

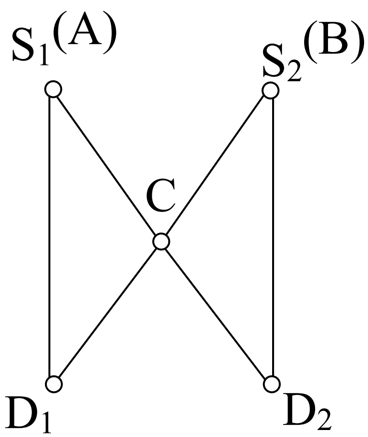
  X

  Node cannot send simultaneously two different messages (it can certainly send the same message to multiple receivers)
  (actually…….. it can, but……)
- 

  X

  Node cannot send and receive simultaneously
  (no ifs and buts!)

THERETORE



* first: about the shown links *

* Here only one node can transmit at a time *

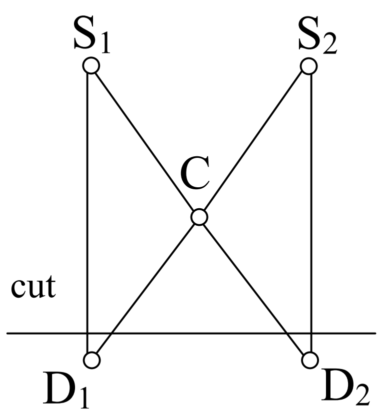
Store and Forward:

1. $S_1 \xrightarrow{A} C$, $S_1 \xrightarrow{A} D_1$
2. $S_2 \xrightarrow{B} C$, $S_2 \xrightarrow{B} D_2$
3. $C \xrightarrow{(A)} D_1$, $C \xrightarrow{A} D_2$
4. $C \xrightarrow{B} D_1$, $C \xrightarrow{(B)} D_2$

Network Coding:

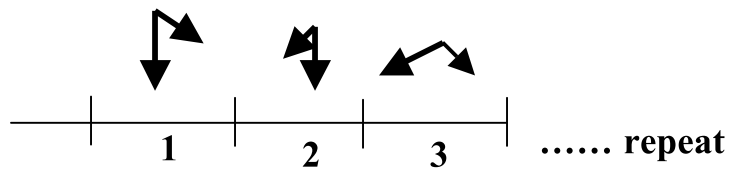
1. $S_1 \xrightarrow{A} C$, $S_1 \xrightarrow{A} D_1$
2. $S_2 \xrightarrow{B} C$, $S_2 \xrightarrow{B} D_2$
3. $C \xrightarrow{A \oplus B} D_1$, $C \xrightarrow{A \oplus B} D_2$

AND HENCE



Limited pipelining (in this case none)

2 packets delivered to 2 destinations (total of 4) over 3 slots

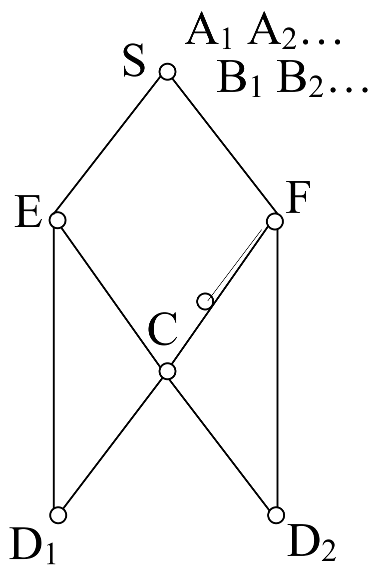


Cut capacity ? <u>NOT</u> 4 / slot

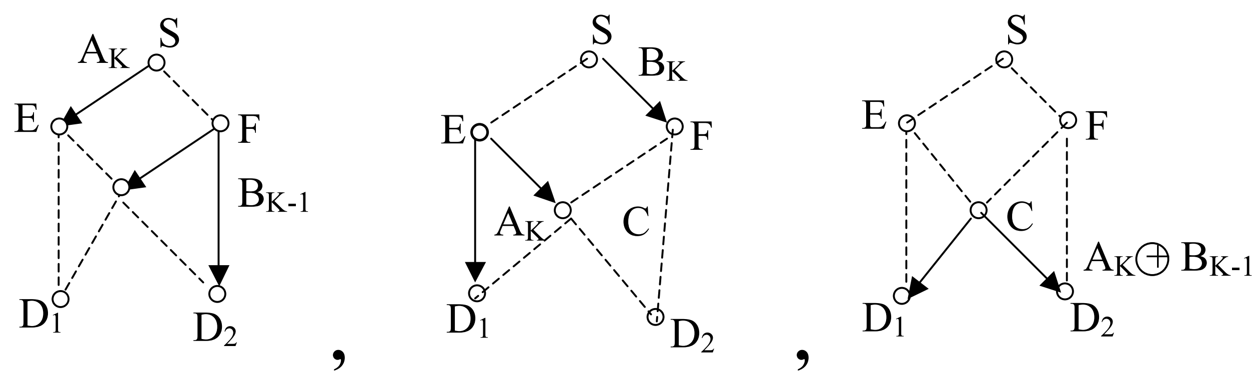
- Needs to be redefined ("degree" of node **<u>and</u>** time division effect)

- Then random linear coding again achieves max-flow/min-cut limit

HOWEVER: IS IT ALL GAINS?



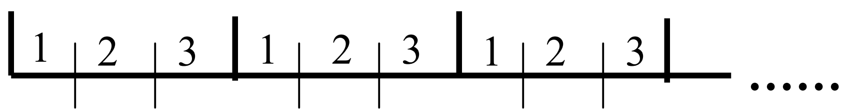
Feasible simultaneous activations:



(1) (2) (3)

Time division over them

Throughput: 4 packets per 3 slots = 1.33



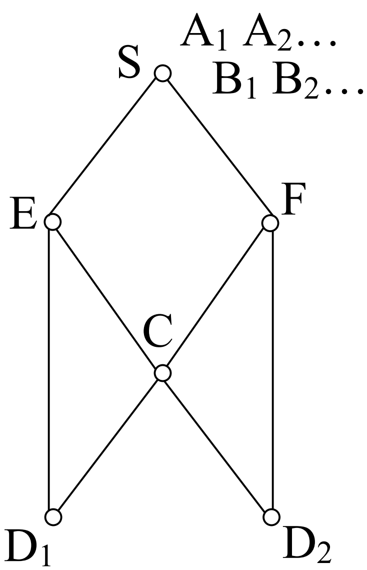
A_K needs 3 slots for D_2 and 2 slots for D_1

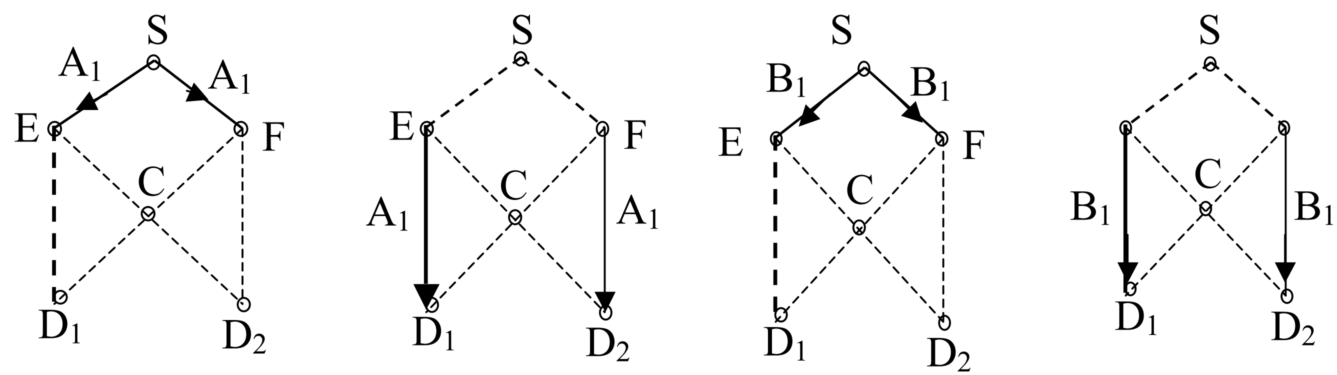
B_{K-1} needs 5 slots for D_1 and 3 slots for D_2

….average = 3.25 slots

WHILE

Store and Forward





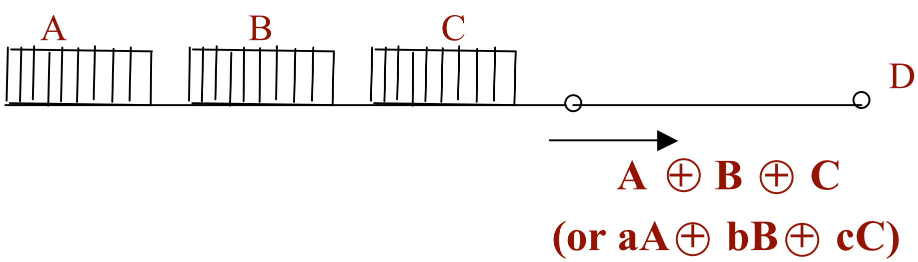
(1) (2) (3)

Throughput: 4 packets over 4 slots = 1

Delay: 2 slots (or even 3 slots)

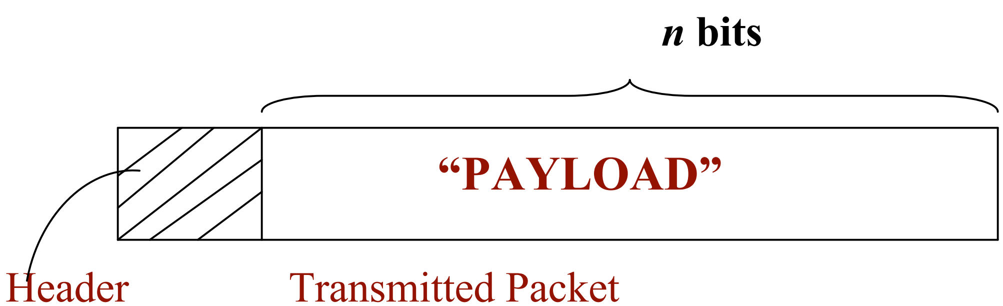
<u>CONCLUSION</u>: FOR WIRELESS: INTIMATE CONNECTION TO MEDIUM ACCES (MAC)

FOCUS ON ONE LINK



- Receiver needs to know the values of the coefficients
- Alphabet size: Q

Hence



: $a_1 A_1, \oplus a_2 A_2 \oplus \ldots \oplus a_k A_K$

Block of K packets (each consisting of *n* bits) gets squeezed into *n* bits

OVERHEAD: K • log Q bits

*independent of length *n* *

HENCE: OH ⟶ 0 AS *n* ⟶ ∞

FOCUS ON ONE LINK (con't.)

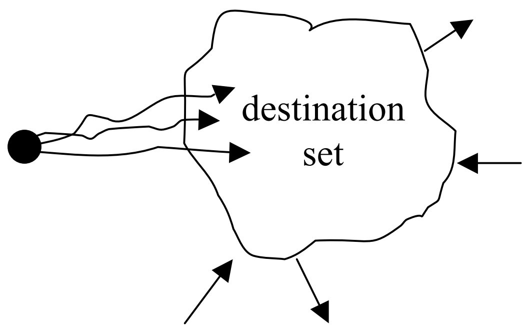
- Need K linearly independent combinations of A_1…….,A_K to decode
- Coefficients are chosen randomly
- Typically N>K packets need to be sent
- So why do it this way?

(if Q: LARGE, N → K; but even then delay is K rather than K/2)

A: - Fountain coding for content distribution

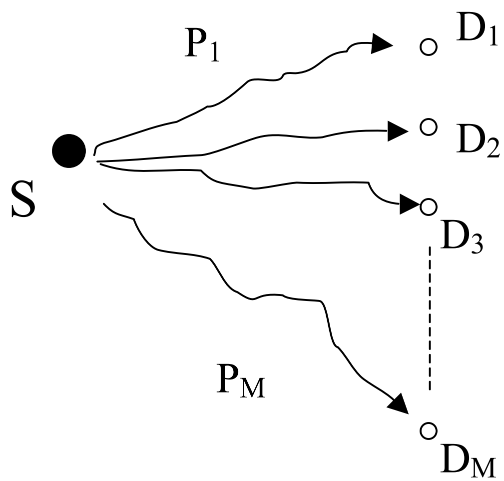
- Multicasting over independent channels

FOUNTAIN CODING
(content distribution)



- Large file (K = large)
- Keep sending random linear combinations of the K packets
- "Rateless" property
- Different users get sufficient numbers of packets to decode the file at different times
- Microsoft "AVALANCHE" (somewhat like "Bit-Torrent")

WIRELESS MULTICASTING



- Channel to each D_i is subject to failures (fades, buffer overflow, etc.)
- Model:

  Packet erasure probability

- P_i : probability a packet fails (is erased) on i^{th} channel
- Independence from channel-to-channel and from packet-to-packet

WIRELESS MULTICASTING (con't.)

- Alternatives: ARQ or NETWORKING CODING
- ARQ: Repeat each packet transmission until all users receive it correctly

  (limited by worse channel)
- Network coding: Keep sending random linear combinations of ALL (i.e. K) packets
  - every user gets new information in each transmission
  - good-channel users receive all K packets early
  - poor-channel users receive all K packets with delay
  - on average: network coding does better

BUT: OH (K log Q) ; need $n \longrightarrow \infty$

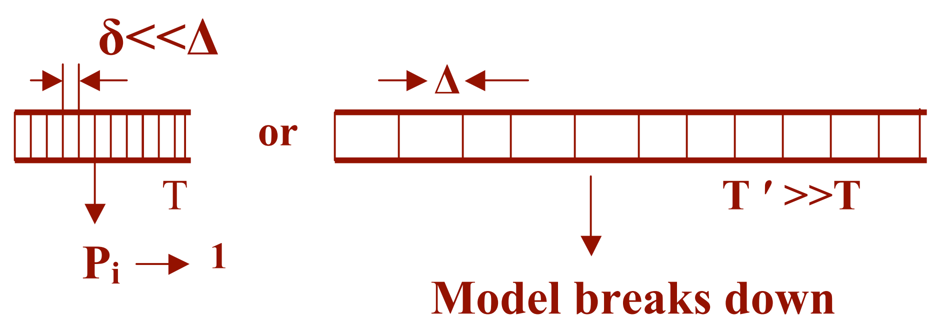
NEW PROBLEM

Q: can P_i stay fixed as $n \longrightarrow \infty$?

A: in real channels: **NO**

Recent "Results": Network coding achieves "Capacity" in general packet erasure networks as $n \longrightarrow \infty$ and the P_i's stay fixed

BUT: As $n \longrightarrow \infty$:



NEW PROBLEM

- Actually we can determine how P_i depends on n

  P_i (n log Q) increasing in n and Q

  and approaches 1 as n or Q go to infinity

- Can determine (in principle) optimal value of alphabet size and length

- **Leads to the ultimate question of combining information and communication theory with networking**