Distributed Resilient Consensus

Nitin Vaidya
University of Illinois at Urbana-Champaign

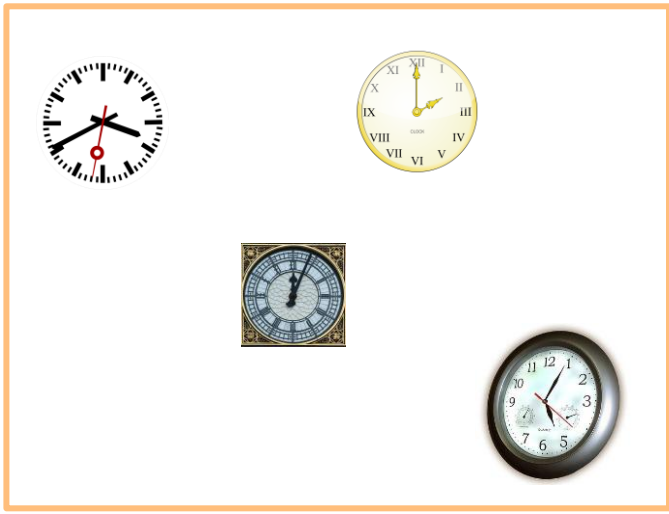
Work supported in part by

Consensus … Dictionary Definition

- General agreement
- Majority opinion

Many Faces of Consensus

- What time is it?



- Network of clocks …

agree on a common notion of time

Many Faces of Consensus

- Where to go for dinner?

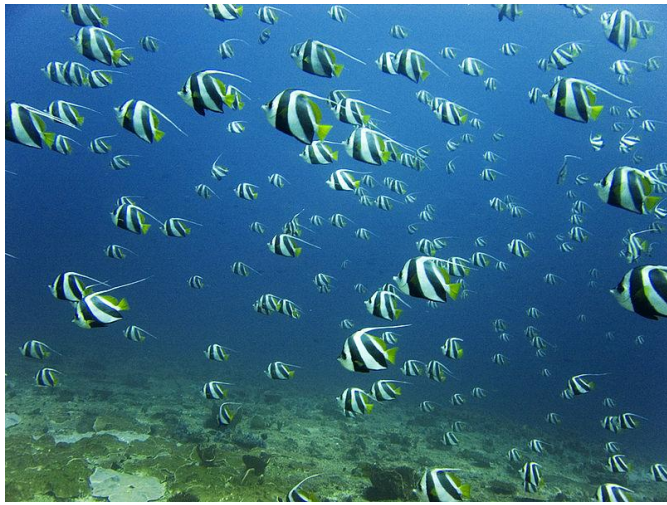


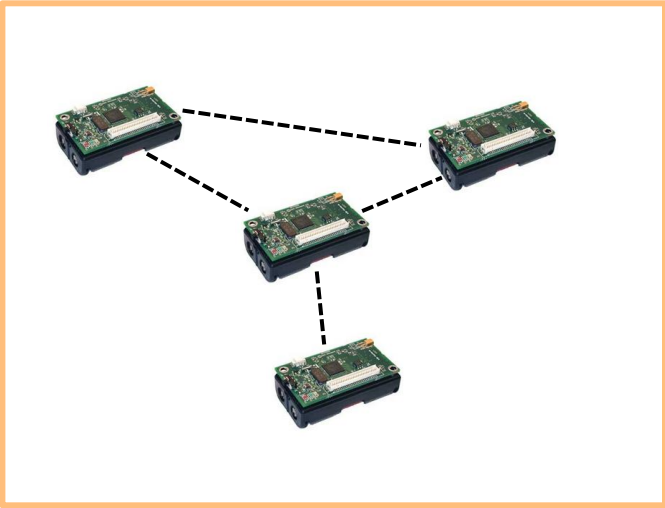
Photo credit: Jon Hanson, Wiki Commons

- Network of friends…

agree on a location

Many Faces of Consensus

- What is the temperature?

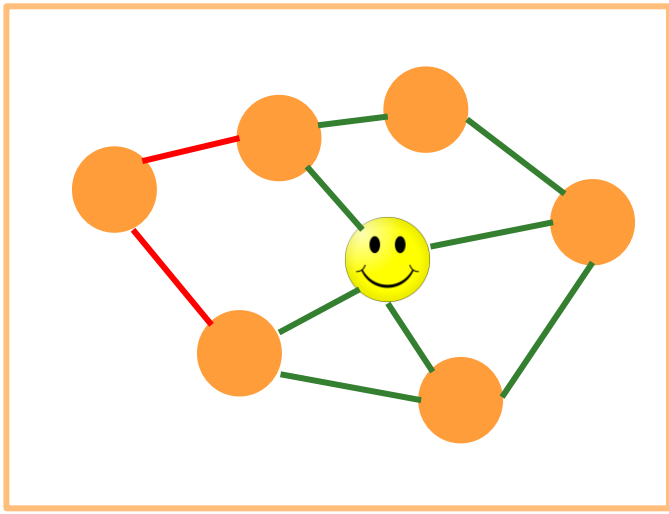


- Network of sensors …

agree on current temperature

Many Faces of Consensus

- Should we trust  ?



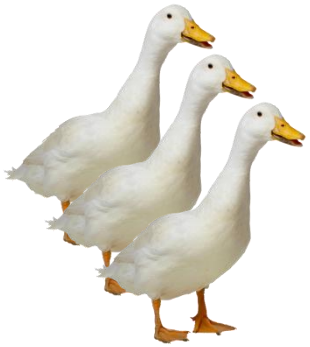
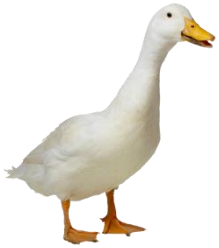
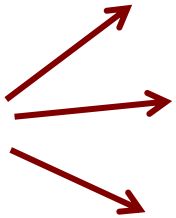
- Web of trust …

agree whether  is good or evil

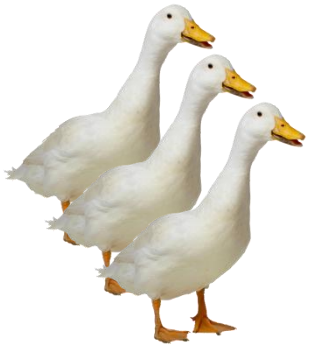
Many Faces of Consensus

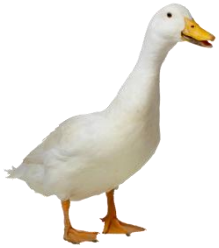
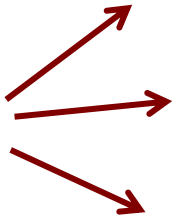
- All nodes have non-null input / only a subset do
- No failures / failures allowed (node/link)
- Synchronous/asynchronous
- Deterministically correct / probabilistically correct
- Exact agreement / approximate agreement

Consensus … one definition

-  want to decide which way to fly
- Every  proposes a direction  
- Consensus ≈ decision in convex hull of inputs

Average Consensus

-  want to decide which way to fly

- Every  proposes a direction  

- Consensus ≈ average of inputs

Selective History

1980: Byzantine consensus

1983: Impossibility of consensus with asynchrony & failure

Tsitsiklis 1984: Decentralized control [Jadbabaei 2003]

1986: Approximate consensus with asynchrony & failure

1989: Swarm intelligence

1996: Failure detectors

1990-98: Paxos

[Kleinberg] Small World 2000: [Milgram 1960s]

Consensus

- 30+ years of research
- Anything still new under the sun ?

More accurate network models

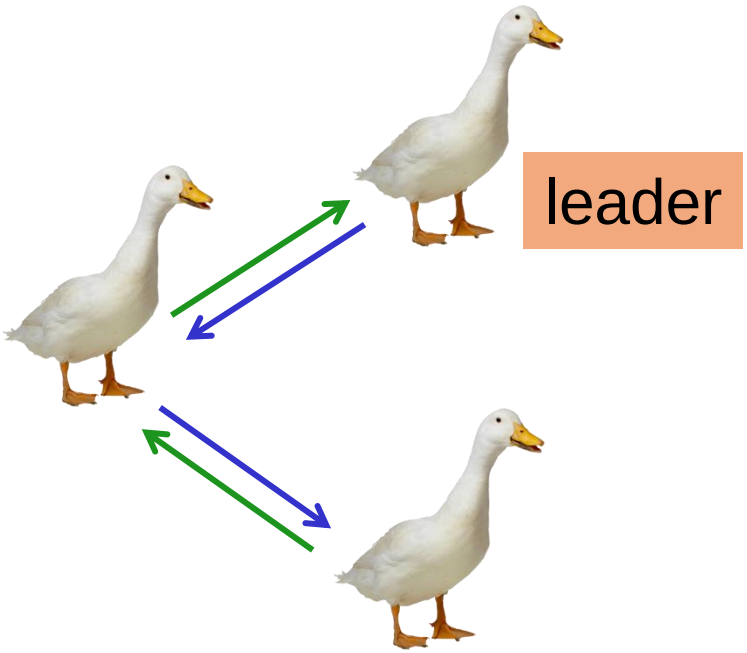
Motivated by Wireless Networks

- Lossy links
- Directed links
- Capacity-constrained links

Review

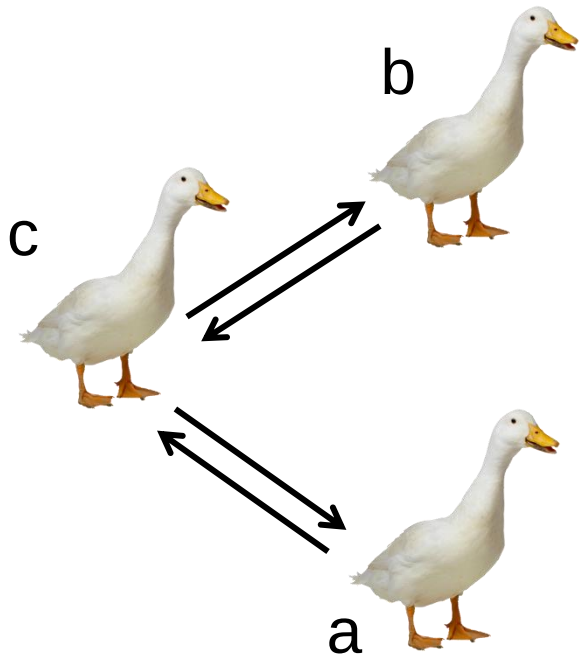
Centralized Solution

- A leader collects inputs & disseminates decision



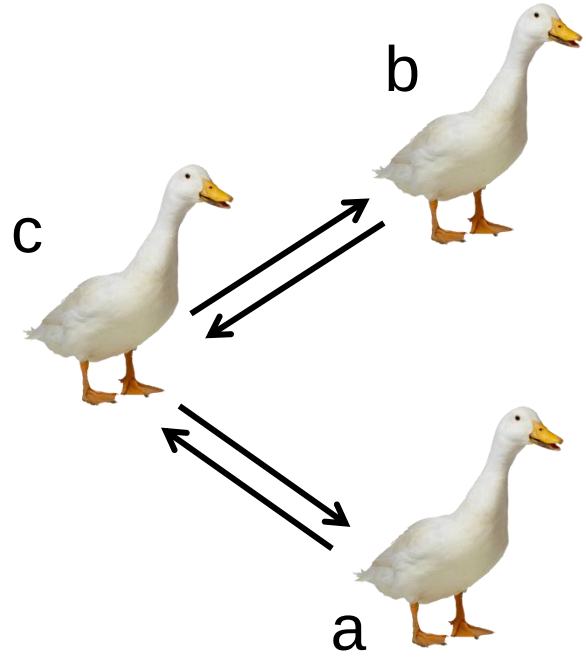
Distributed Iterative Solution

- Initial state a, b, c = input



Distributed Iterative Solution

- State update (iteration)

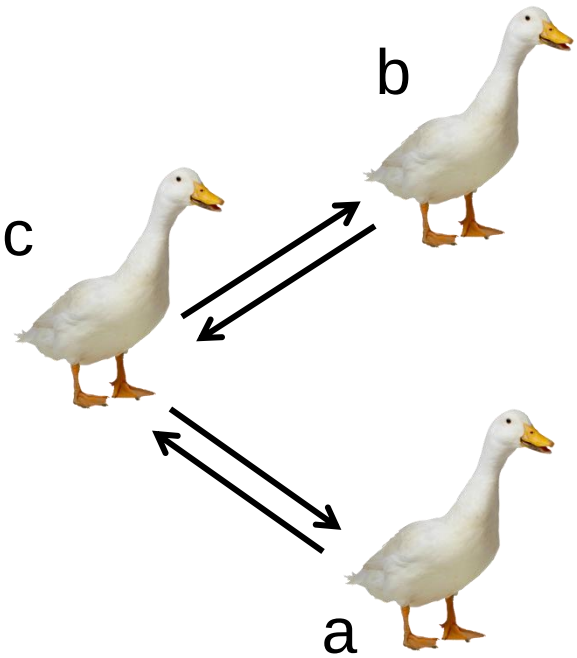


$b = 3b/4 + c/4$

$c = a/4 + b/4 + c/2$

$a = 3a/4 + c/4$

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} := \underbrace{\begin{pmatrix} 3/4 & 0 & 1/4 \\ 0 & 3/4 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix}}_{M} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = M \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$



$b = 3b/4 + c/4$

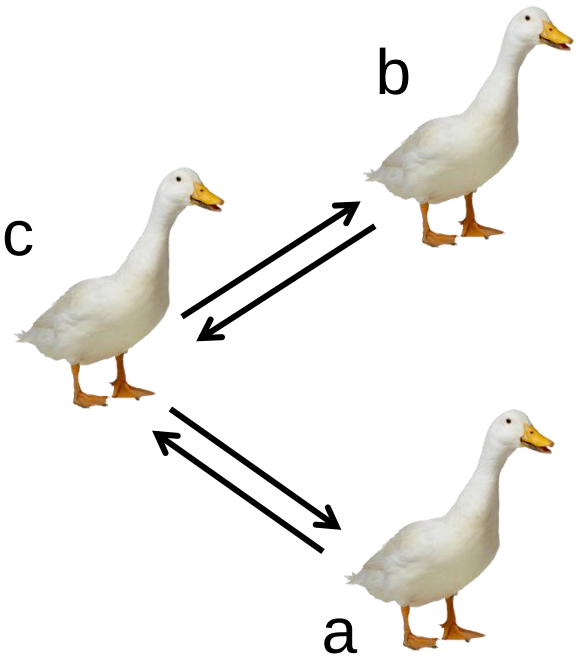
$c = a/4 + b/4 + c/2$

$a = 3a/4 + c/4$

after 2 iterations

after 1 iteration

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} := M \left[M \begin{pmatrix} a \\ b \\ c \end{pmatrix} \right] = M^2 \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$



$b = 3b/4 + c/4$

$c = a/4 + b/4 + c/2$

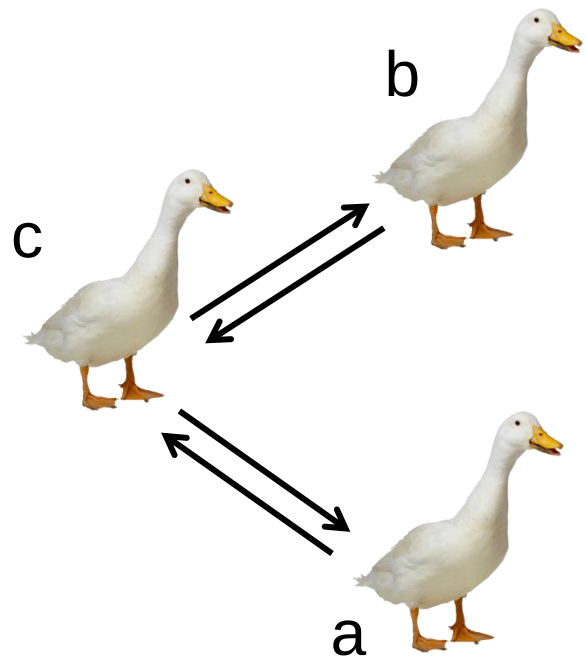
$a = 3a/4 + c/4$

after k iterations

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} := M^k \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$b = 3b/4 + c/4$

$c = a/4 + b/4 + c/2$



$a = 3a/4 + c/4$

Well-Known Results

Well-Known Results

Reliable links & nodes:

Row stochastic M

- Consensus achievable iff
  at least one node can reach all other nodes

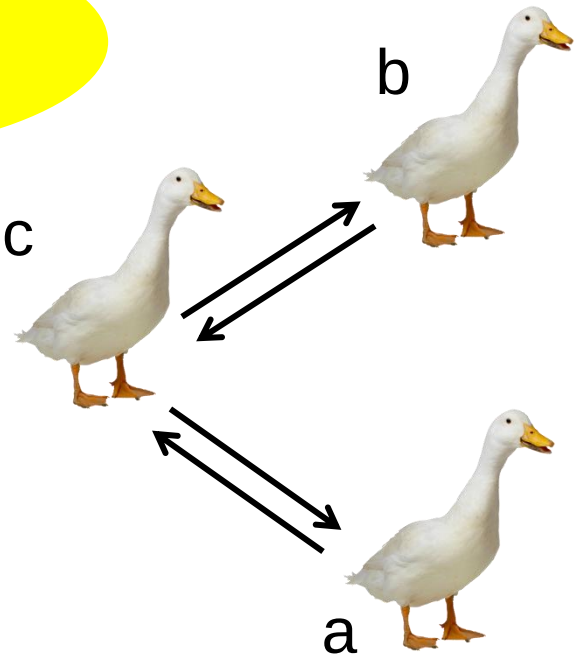
- Average consensus achievable iff
  strongly connected graph

Doubly stochastic M

with suitably chosen transition matrix M

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} := M^k \begin{pmatrix} a \\ b \\ c \end{pmatrix} \rightarrow \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Doubly stochastic M



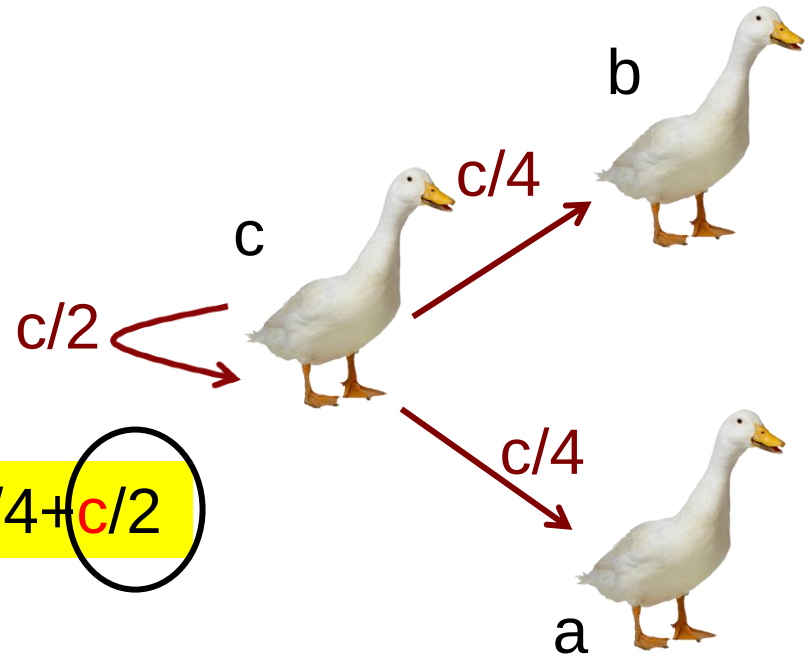
$b = 3b/4 + c/4$

$c = a/4 + b/4 + c/2$

$a = 3a/4 + c/4$

An Implementation: Mass Transfer + Accumulation

- Each node "transfers mass" to neighbors via messages
- Next state = Total received mass



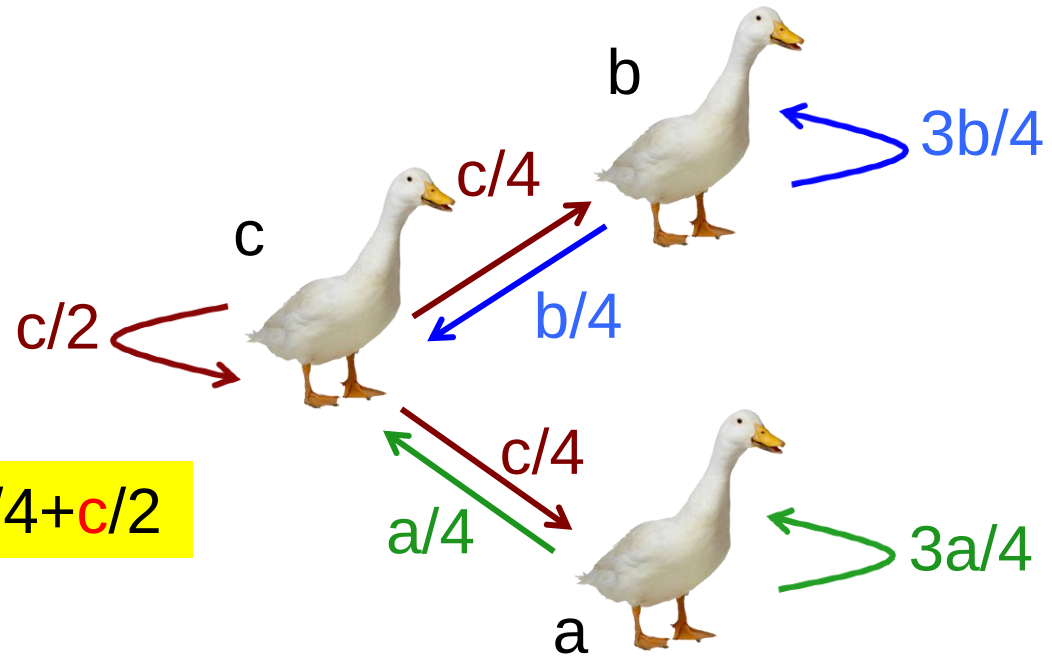
$c = a/4 + b/4 + c/2$

$b = 3b/4 + c/4$

$a = 3a/4 + c/4$

An Implementation:
Mass Transfer + Accumulation

- Each node "transfers mass" to neighbors via messages
- Next state = Total received mass



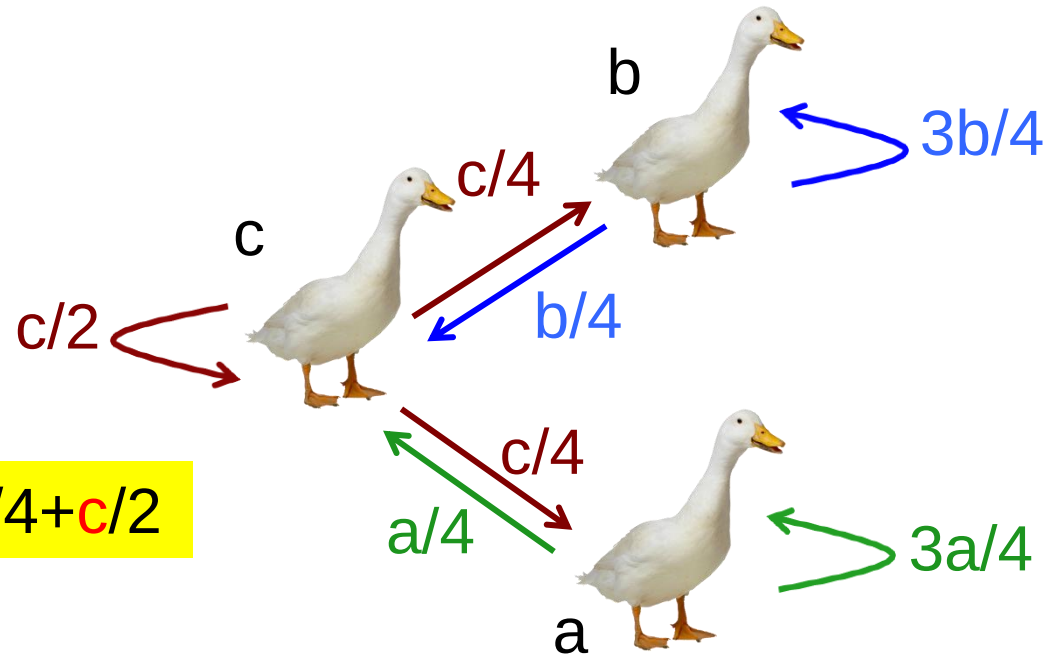
b = 3b/4+ c/4

c = a/4+b/4+c/2

a = 3a/4+ c/4

Conservation of Mass

- a+b+c constant after each iteration



$b = 3b/4 + c/4$

$c = a/4 + b/4 + c/2$

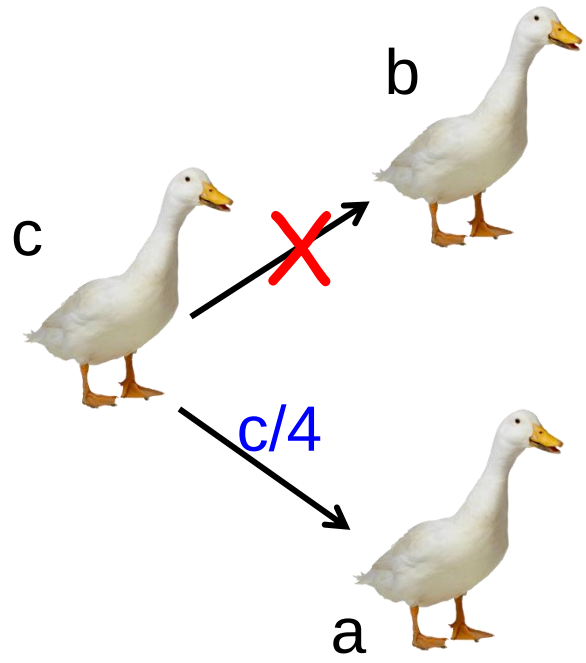
$a = 3a/4 + c/4$

Outline

- Lossy links
- Directed links
- Capacity-constrained links

Wireless Transmissions Unreliable

$c = a/4 + b/4 + c/2$

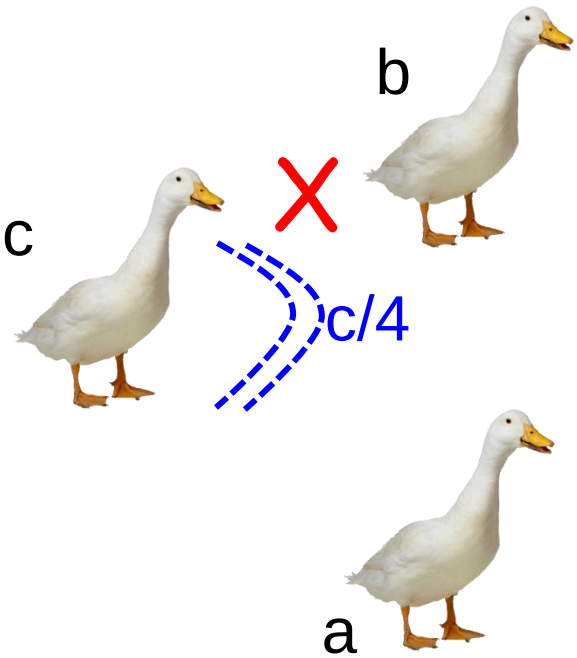


$b = 3b/4 + \cancel{c/4}$

$a = 3a/4 + c/4$

Impact of Unreliability

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 3/4 & 0 & 1/4 \\ 0 & 3/4 & 0 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$



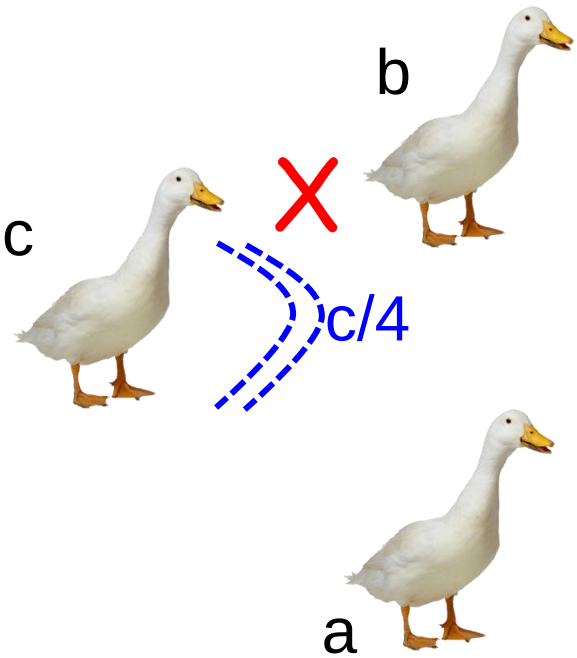
b = 3b/4+ ~~c/4~~

c = a/4+b/4+c/2

a = 3a/4+ c/4

Conservation of Mass

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 3/4 & 0 & 1/4 \\ 0 & 3/4 & 0 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$



b = 3b/4+ c/4

c = a/4+b/4+c/2

a = 3a/4+ c/4

Average consensus over lossy links ?

Existing Solution

Assume that

transmitter KNOWS

when a message is not delivered

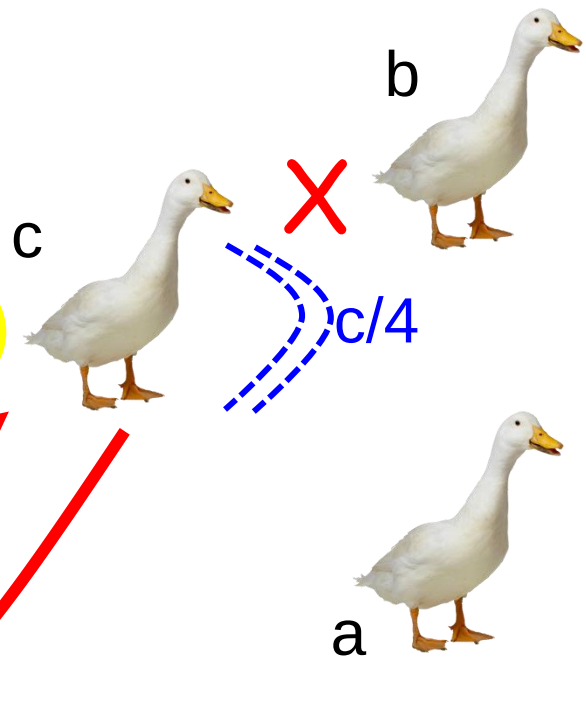
Existing Solution

When mass not transferred to neighbor,

keep it to yourself

Convergence … if nodes intermittently connected

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 3/4 & 0 & 1/4 \\ 0 & 3/4 & 0 \\ 1/4 & 1/4 & 3/4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$



b = 3b/4+ ~~c/4~~

c = a/4+b/4+c/2+c/4

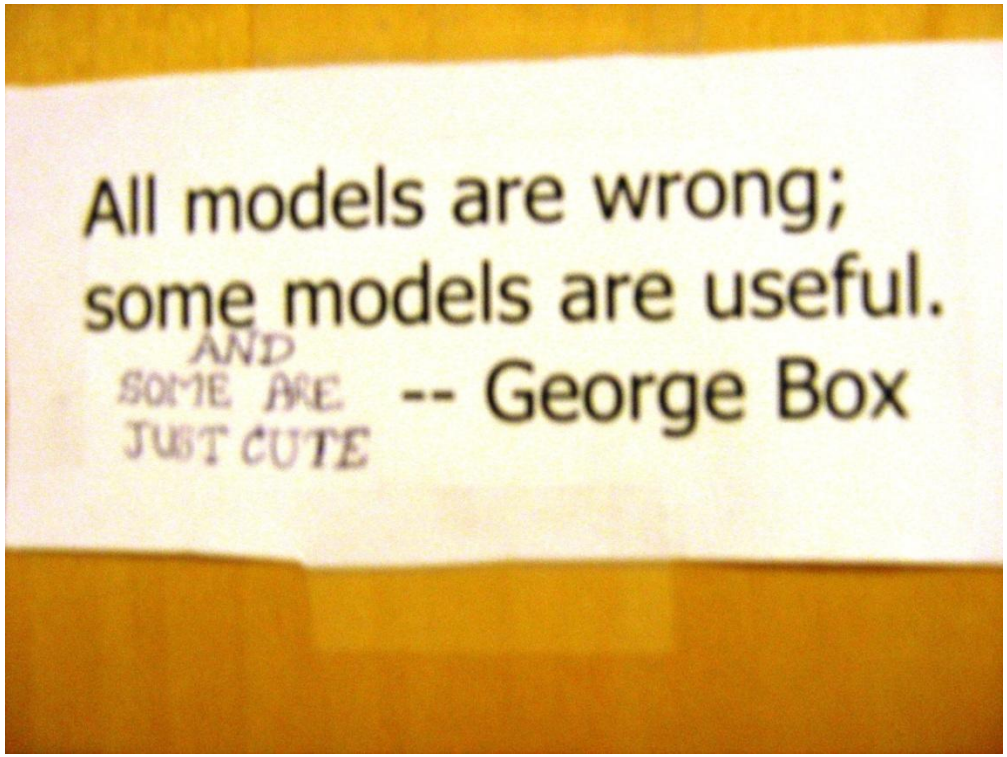
a = 3a/4+ c/4

Link Model

Assume that

transmitter KN~~O~~WS

when a message is not delivered


All models are wrong;
some models are useful.
AND
SOME ARE
JUST CUTE
-- George Box

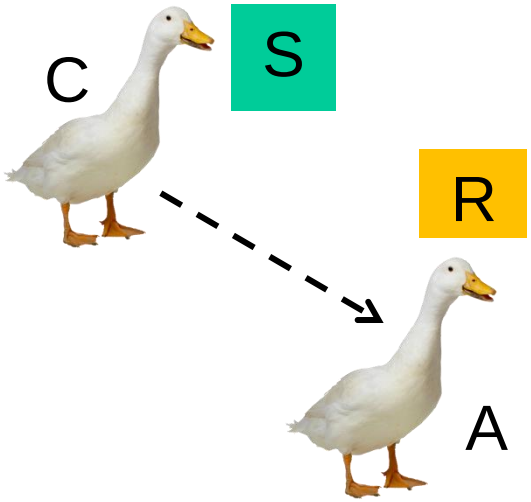
Better Model ?

No common knowledge regarding message delivery

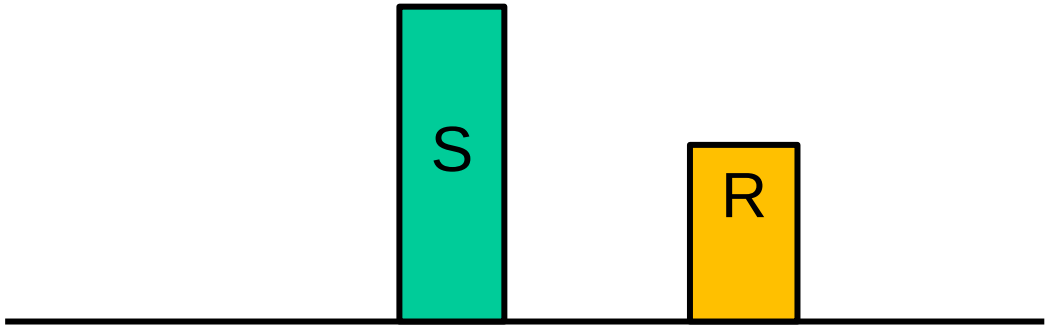
Solution

- Introduce memory

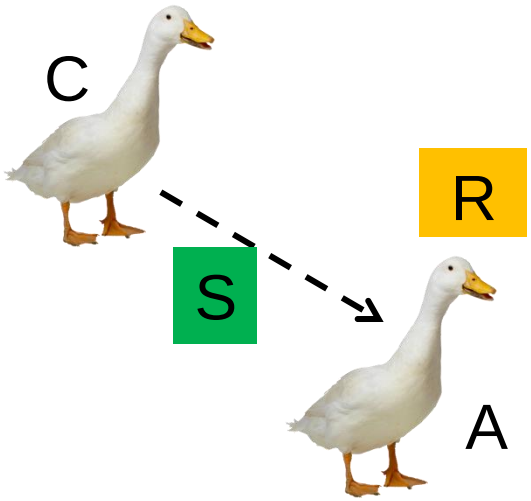
Solution Sketch



- S = mass C wanted to transfer to node A in total so far
- R = mass A has received from node C in total so far

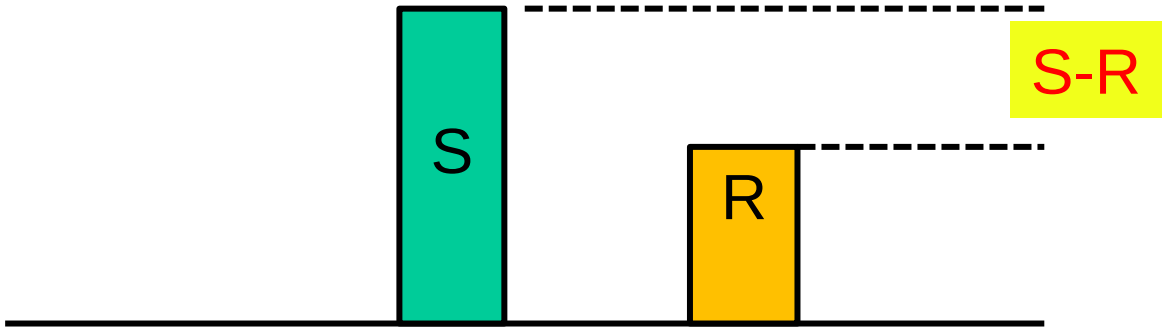


Solution Sketch



- Node C transmits quantity S
  …. message may be lost

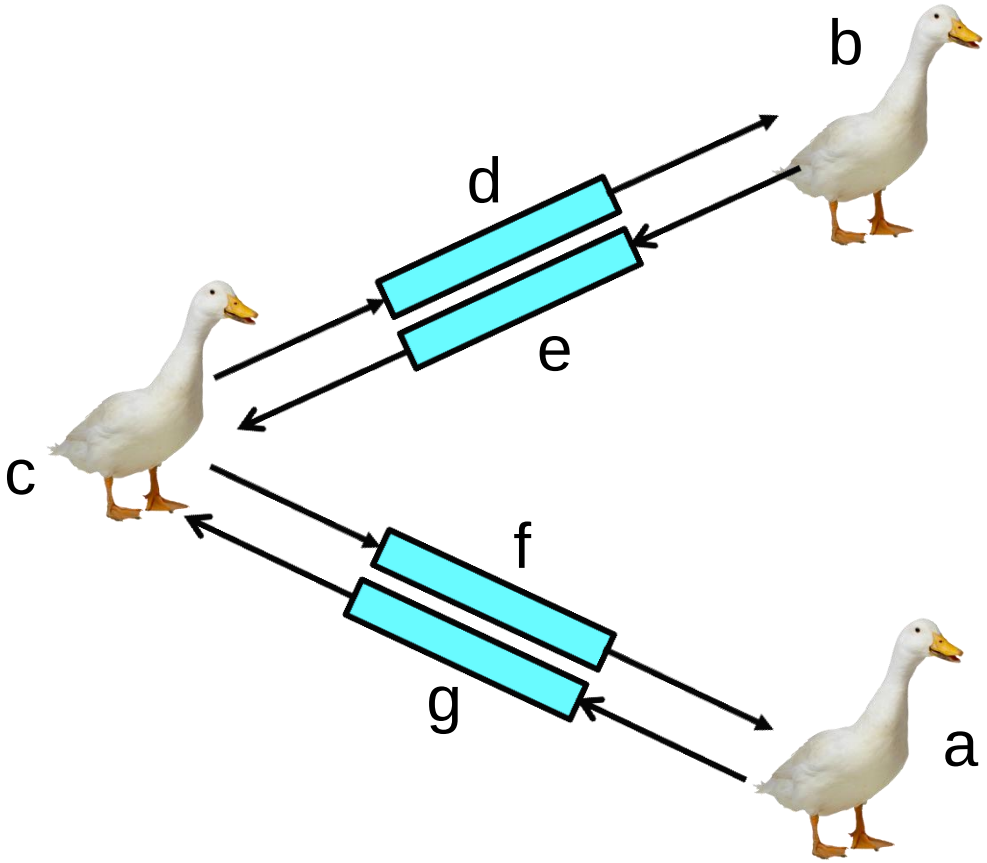
- When it is received,
  node A accumulates (S-R)



What Does That Do ?

What Does That Do ?

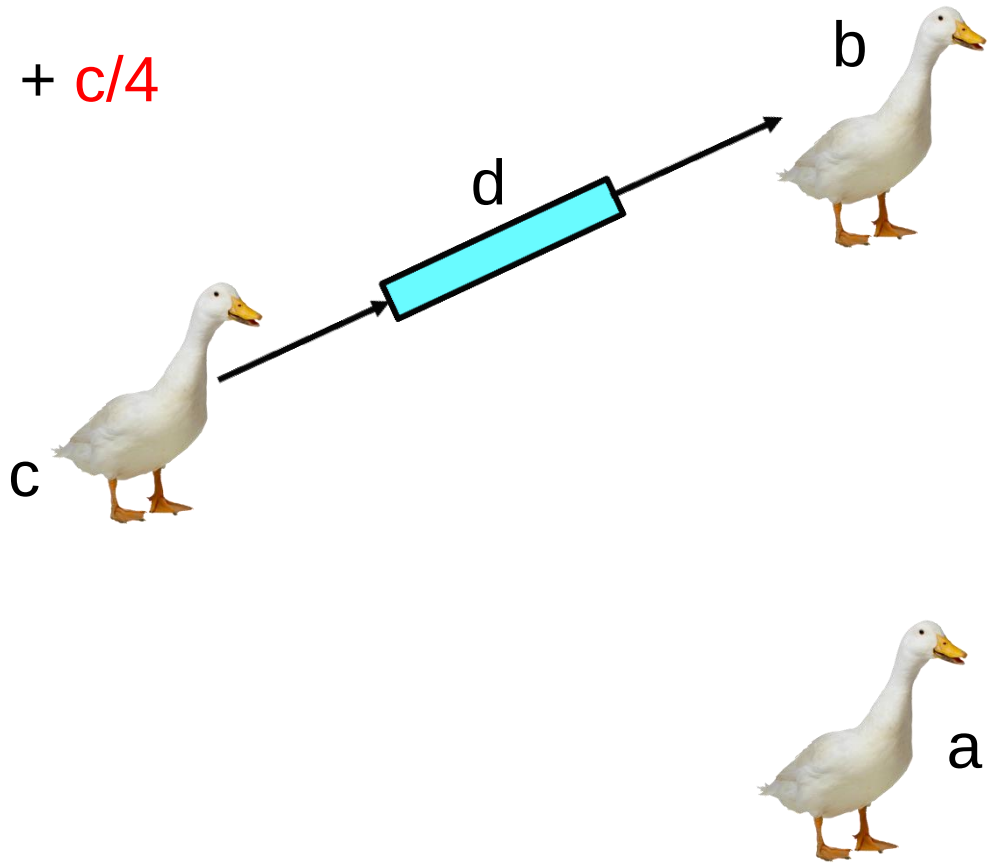
- Implements virtual buffers



Dynamic Topology

- When C→B transmission unreliable, mass transferred to buffer (d)

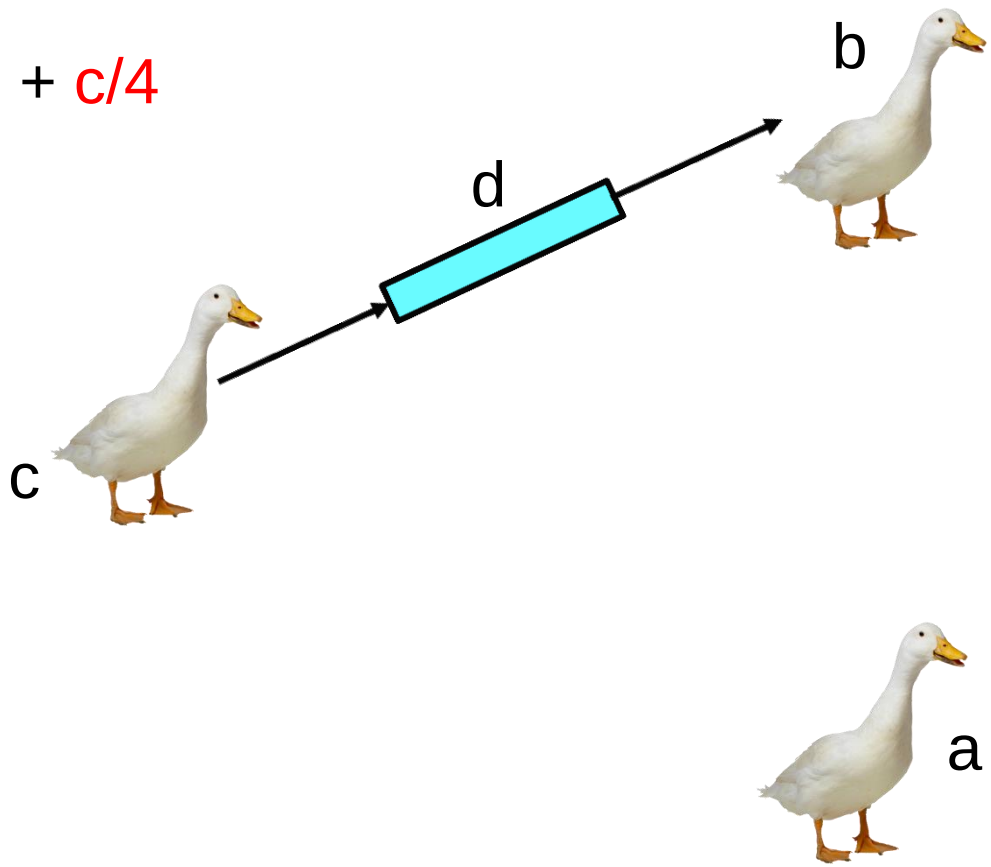
- d = d + c/4



Dynamic Topology

- When C→B transmission unreliable, mass transferred to buffer (d)

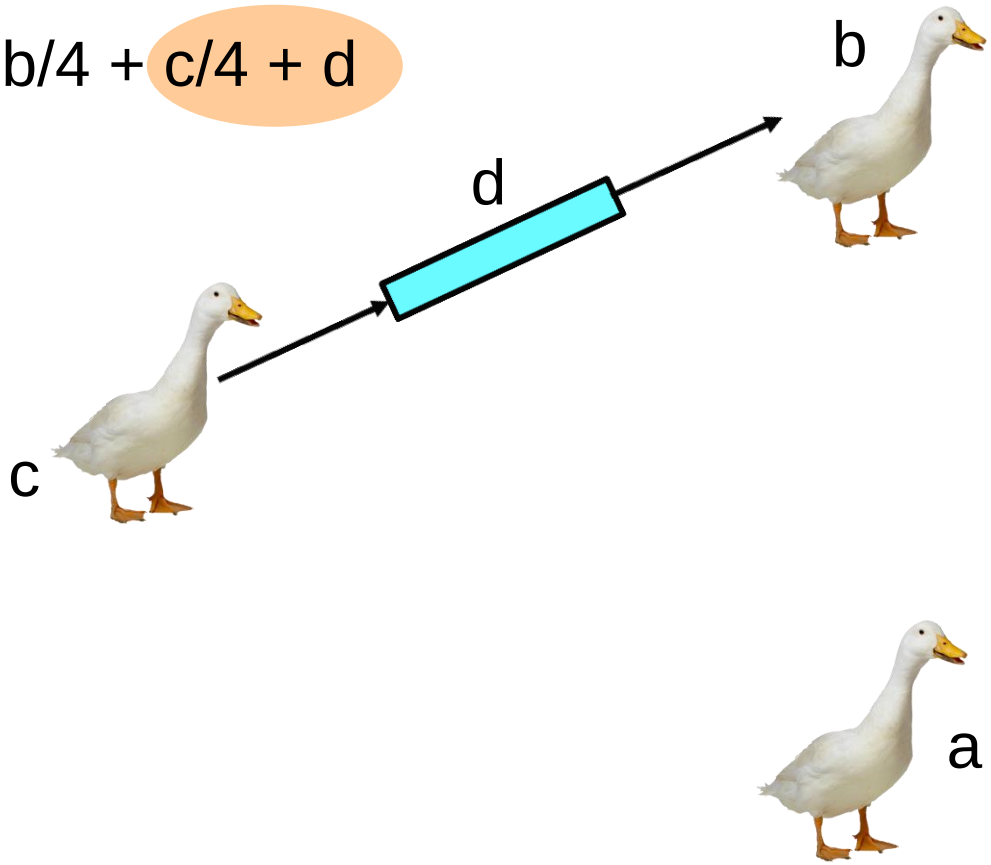
- d = d + c/4



No loss of mass even with message loss

Dynamic Topology

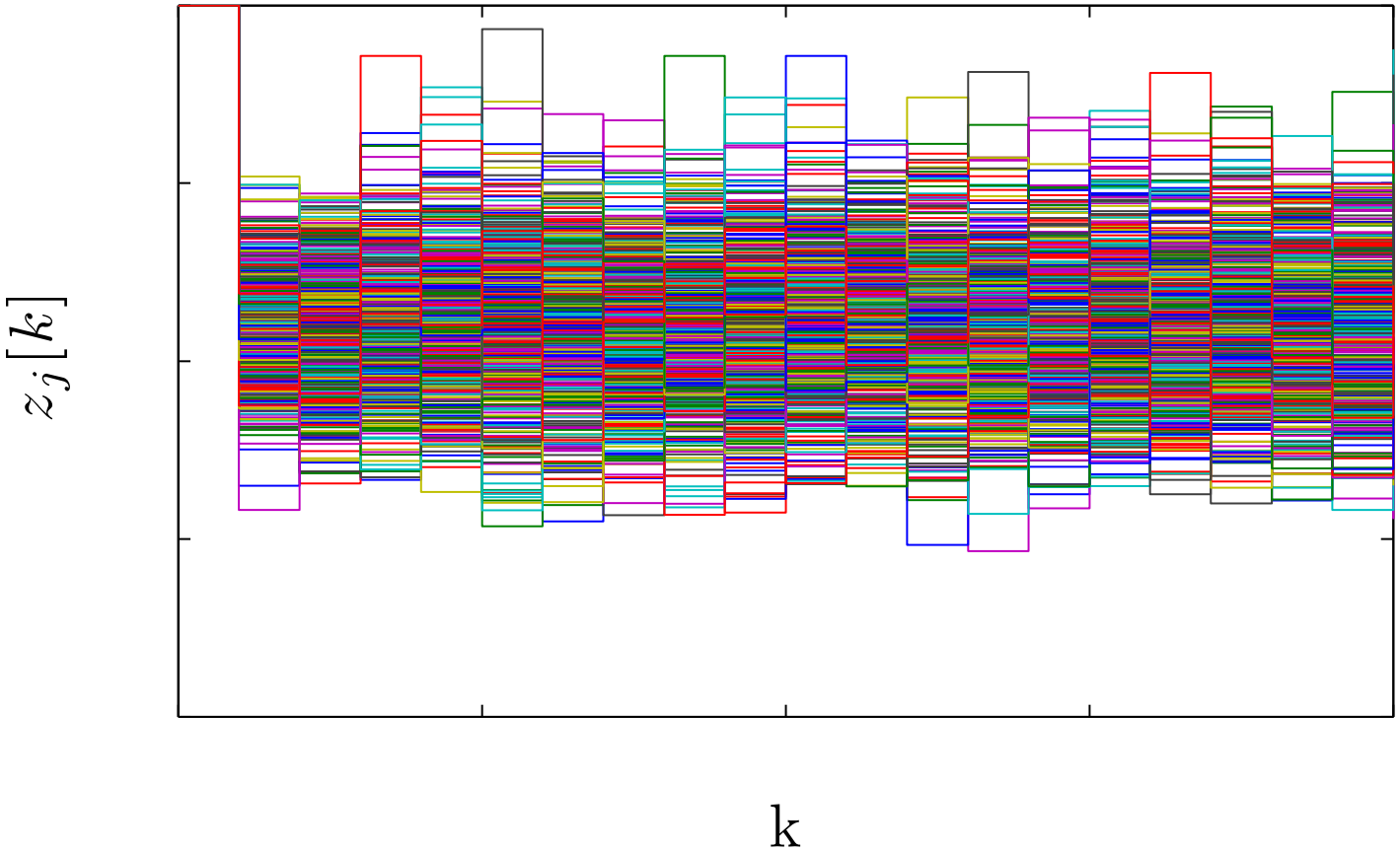
- When C→B transmission reliable, mass transferred to b
- b = 3b/4 + c/4 + d



No loss of mass even with message loss

Does This Work ?

Does This Work ?



Why Doesn't it Work ?

- After k iterations, state of each node has the form

  z(k) * sum of inputs

  where z(k) changes each iteration (k)

- Does <u>not</u> converge to average

Solution

- Run two iterations in parallel
  - First : original inputs
  - Second : input = 1

Solution

- Run two iterations in parallel
  - First : original inputs
  - Second : input = 1
- After k iterations …

first algorithm:	z(k) * sum of inputs
second algorithm:	z(k) * number of nodes

Solution

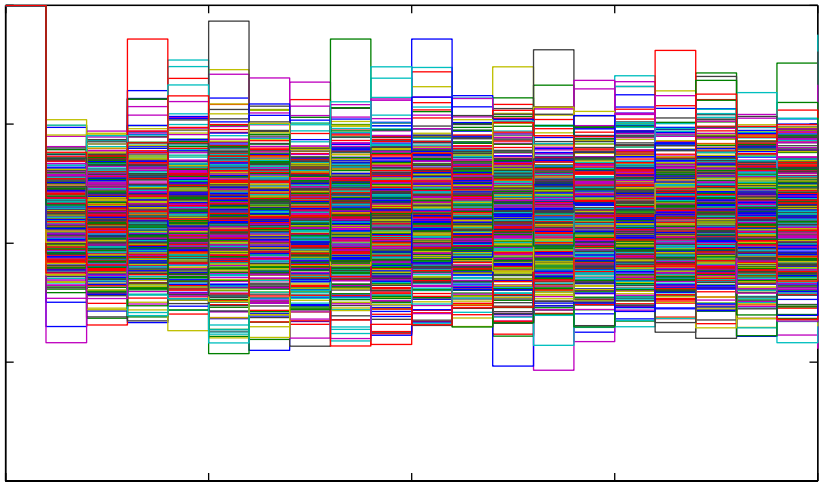
- Run two iterations in parallel
  - First : original inputs
  - Second : input = 1

- After k iterations …

$$\frac{\text{first algorithm: } z(k) * \text{sum of inputs}}{\text{second algorithm: } z(k) * \text{number of nodes}}$$

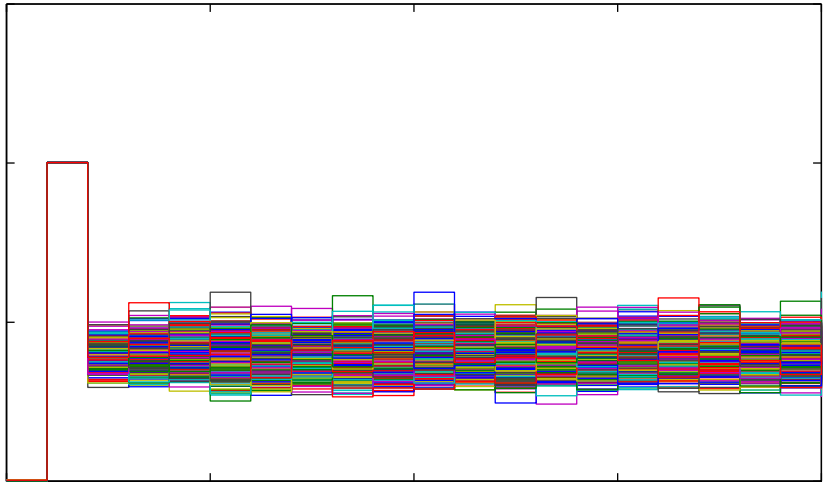
ratio = average

numerator



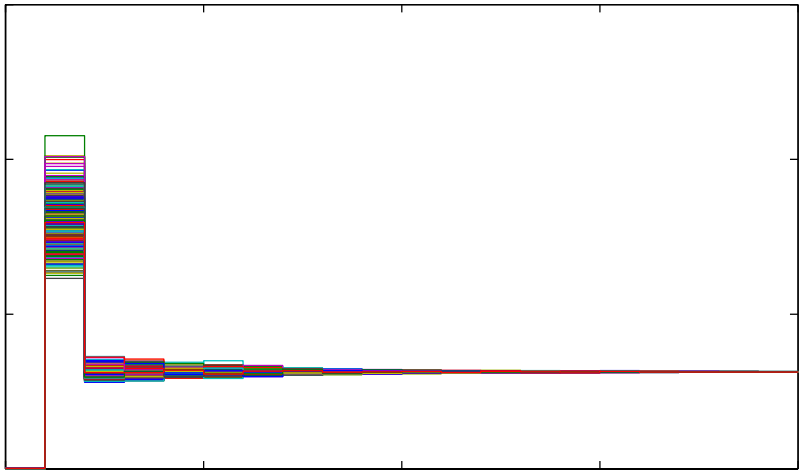
k

denominator



k

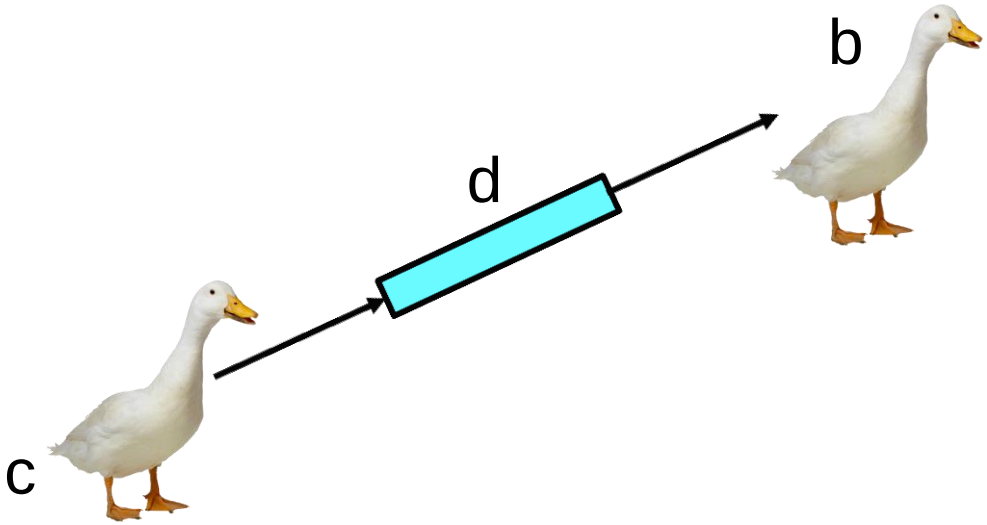
ratio



k

Open Problem

- How to handle lossy links in other iterative computations?



Outline

- Lossy links
- Directed links
- Capacity-constrained links

Outline

- Lossy links
- Directed links ← Iterative Byzantine Consensus
- Capacity-constrained links

Byzantine Consensus

Up to f nodes faulty

- Agreement: Good nodes agree on (*approximately*) identical value
- Validity: If all good nodes have identical input, agreed value equals that input

Iterative Consensus in Complete Graphs

Each iteration:

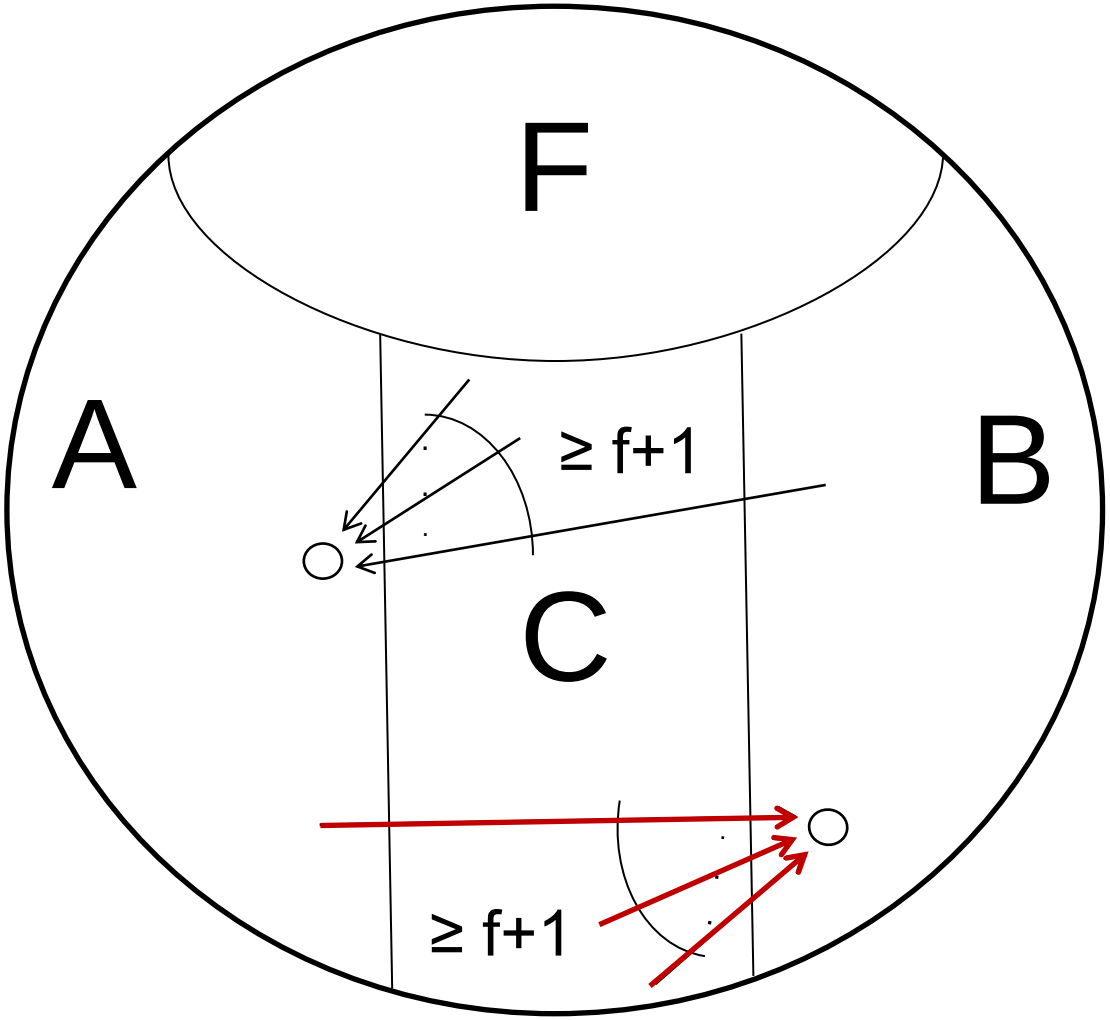
- Send state to others … receive from others
- Drop low and high f values
- New state = average of the rest

Incomplete Graphs

- Which directed graphs can solve

  Byzantine consensus iteratively ?

Necessary & Sufficient Condition



Matrix Representation

- s = state vector of all nodes

$$s = M * s \quad ?$$

- Faulty nodes can behave arbitrarily

Matrix Representation

- s = state vector of FAULT-FREE nodes

$$s = M[t] * s$$

where M[t] depends behavior of faulty nodes in t-th iteration

Matrix Representation

- s = state vector of FAULT-FREE nodes

$$s = M[t] * s$$

where M[t] depends behavior of faulty nodes in t-th iteration

Many such M[t] exist

Matrix Representation

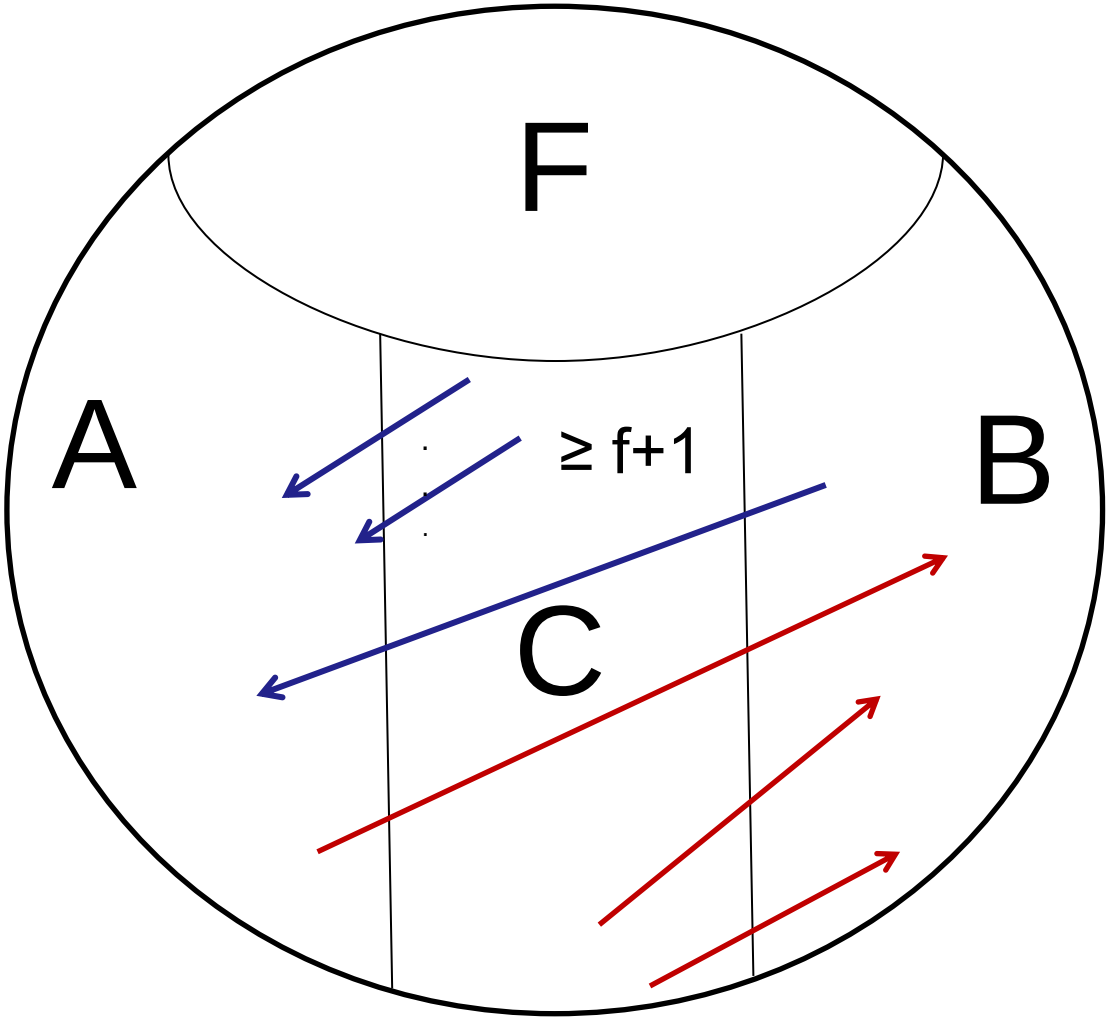
- Many such M[t] exist
- When the graph satisfies the necessary condition, possible to <u>choose</u> M[t] with adequate “connectivity”
- Results in correct decision

Byzantine Consensus

- Similar question for exact consensus
- Without constraining to iterative structure

Which directed graphs can solve the problem ?

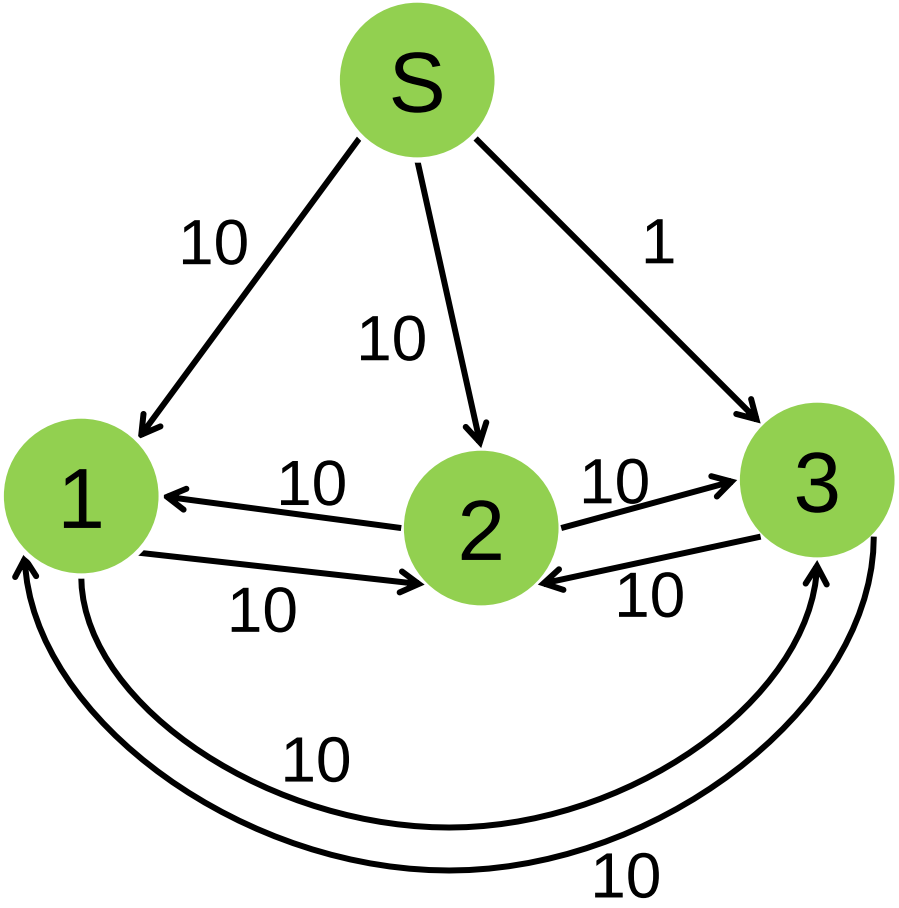
Necessary & Sufficient Condition



Outline

- Lossy links
- Directed links
- Capacity-constrained links

Link Capacity Constraints



Byzantine Consensus … Known Bounds

- $\Omega(n^2)$ messages in worst case
- f+1 rounds of communication

Link Capacity Constraints

- How to quantify the impact ?

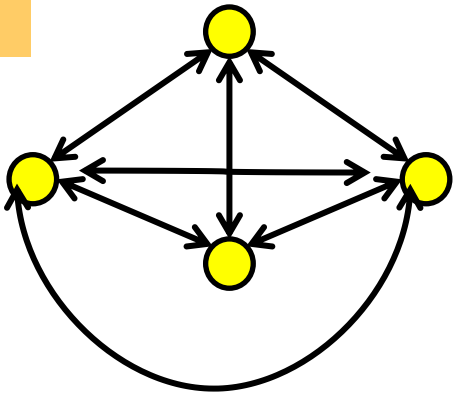
Metric 1:
Communication Cost per Bit

$$\frac{\text{Total communication cost (in bits)}}{\text{Number of bits of Byzantine broadcast}}$$

Metric 1:
Communication Cost per Bit

$$\frac{\text{Total communication cost (in bits)}}{\text{Number of bits of Byzantine consensus}}$$

Ignores network characteristics



Metric 2: Throughput

- Borrow notion of throughput from networking

- *b(t)* = number of bits agreed upon in *[0,t]*

$$Throughput = \lim_{t \to \infty} \frac{b(t)}{t}$$

Problem Definition

- What is the achievable throughput of consensus over a capacitated network ?

- Some progress … but not completely solved

Summary

- Lossy links
- Directed links
- Capacity-constrained links

Summary

- Open problems