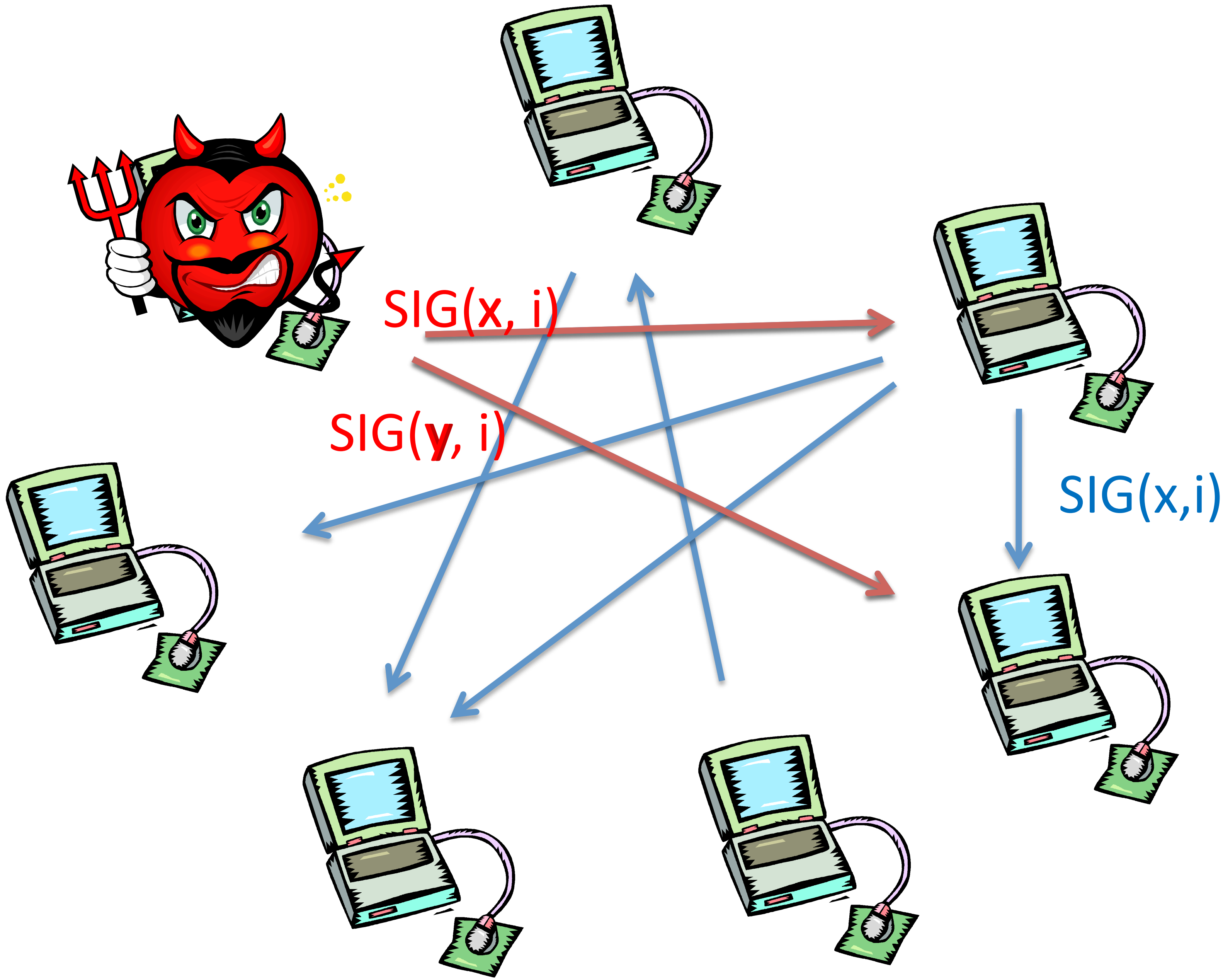
Byzantine Fault Tolerance

Eleanor Birrell

November 23, 2010


SIG(x, i)
SIG(y, i)
SIG(x,i)

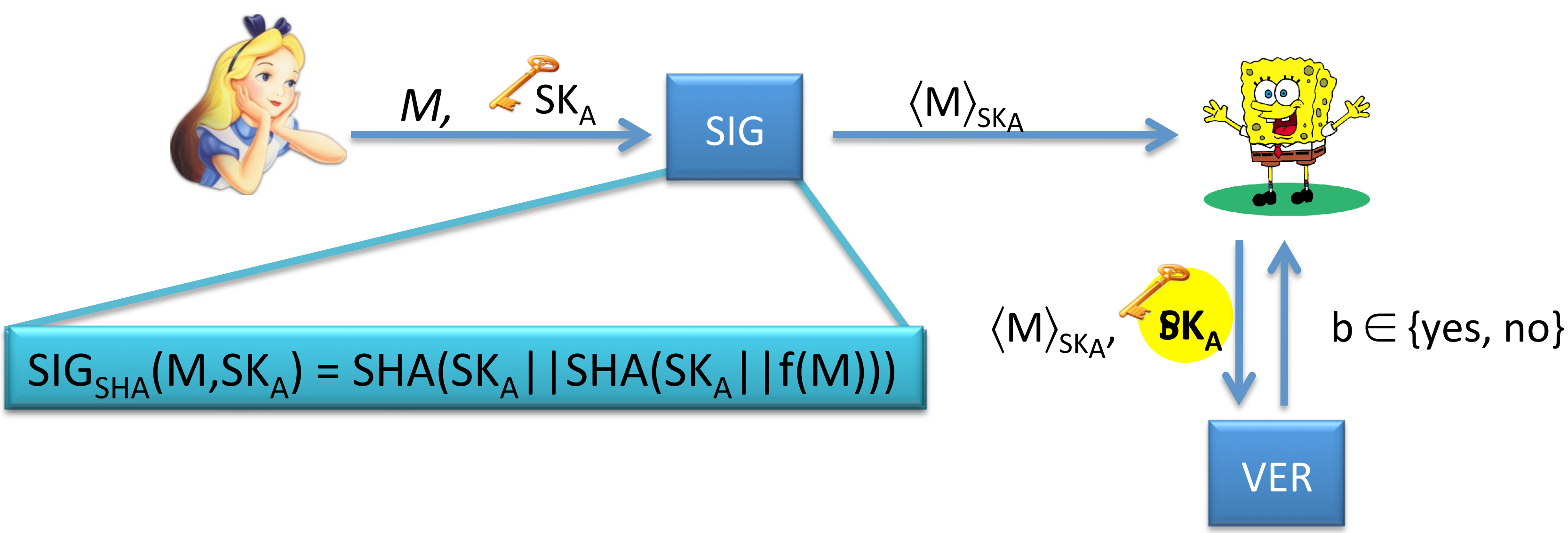
Authenticated Messages

Digital Signatures

- Public-Key
- Inefficient

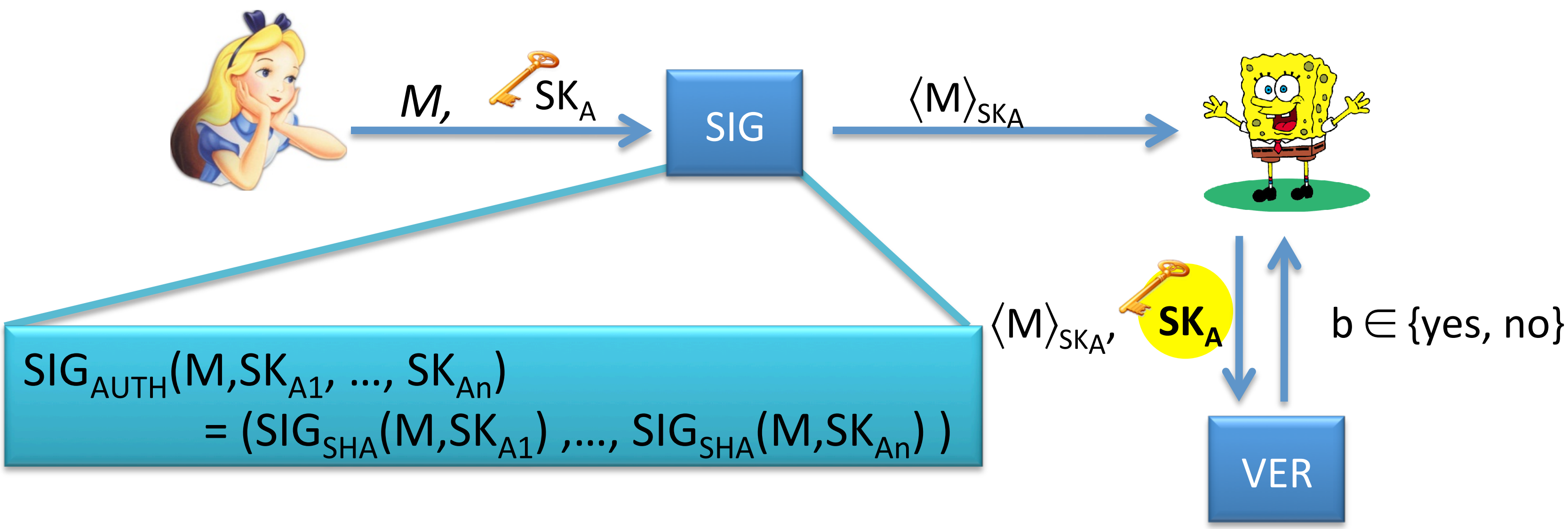
Message Authentication Codes (MAC)

- Secret-Key
- 3 orders of magnitude faster



Authenticators

- MACs cannot be authenticated by a third party
  - Solution: create vector of MACs (called *authenticator*) with one code for each node
  - Verification O(1) but generation O(n)



Byzantine Fault Tolerance (Results)

(m = traitors, n = total)	Synchronous	Semi-Sync	Asynchronous
Oral Messages: Negative			
Positive			
Authenticated: Negative			
Positive			

Byzantine Fault Tolerance (Results)

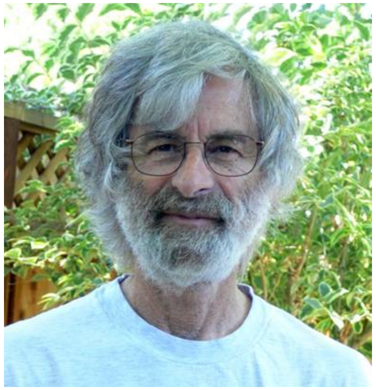
(m = traitors, n = total)	Synchronous	Semi-Sync	Asynchronous
Oral Messages: Negative	n ≤ 3m [LSP80]		m ≥ 1 [FLP82]
Positive	n ≥ 3m+1 [LSP80]		
Authenticated: Negative			m ≥ 1 [FLP82]
Positive	n ≥ 1 [LSP80]	n ≥ 3m+1 [CL99]	

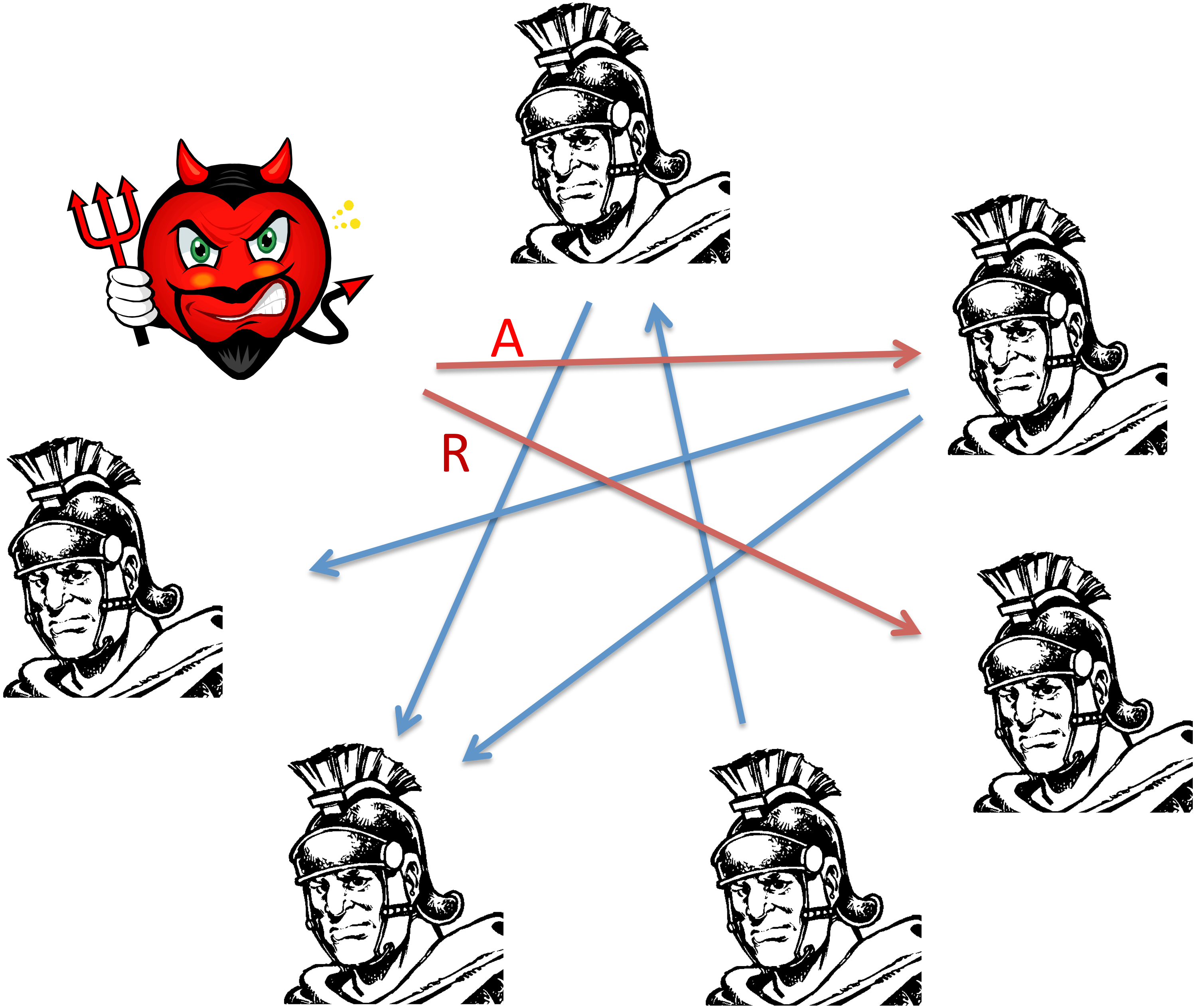
Byzantine Fault Tolerance (Results)

(m = traitors, n = total)	Synchronous	Semi-Sync	Asynchronous
Oral Messages: Negative	n ≤ 3m [LSP80]		m ≥ 1 [FLP82]
Positive	n ≥ 3m+1 [LSP80]		
Authenticated: Negative			m ≥ 1 [FLP82]
Positive	n ≥ 1 [LSP80]	n ≥ 3m+1 [CL99]	**???**

L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem (1982)

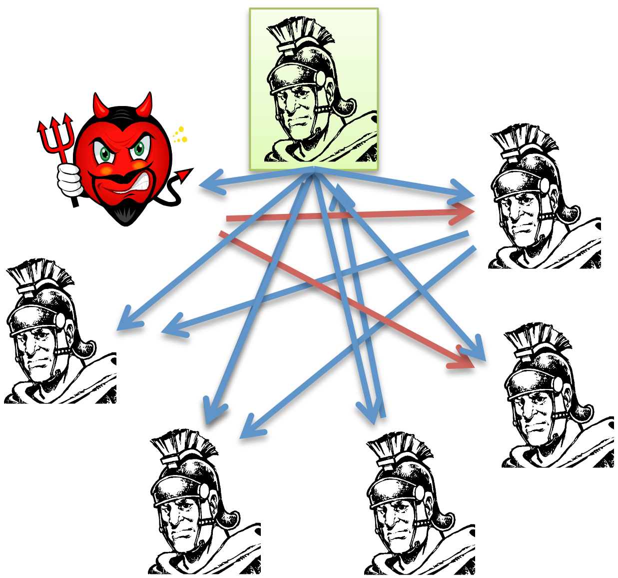
- Leslie Lamport
  - PhD Brandeis 1972 (Math)
  - SRI, DEC, Compaq, MSR
  - Clocks, Paxos, LaTex
- Robert Shostak
  - PhD Harvard 1974
  - SRI, Ansa (Paradox), Portera, Vocera
- Marshall Pease
  - SRI International





A
R

Byzantine Generals Problem



- Simple Model:
  - Army of n Generals
  - Each Gen. i has opinion $v(i) \in$ {Attack, Retreat}
- Goals:
  - Agree on same opinion
  - Agree on good opinion

- Interactive Consistency conditions (ICCs):

Commanding general i sends $v_j(i)$ to Lt. j:

  - All loyal Lt. obey same order
  - If i is loyal, every loyal Lt. j obeys order $v_j(i)$
- Solution if ICCs hold for all i

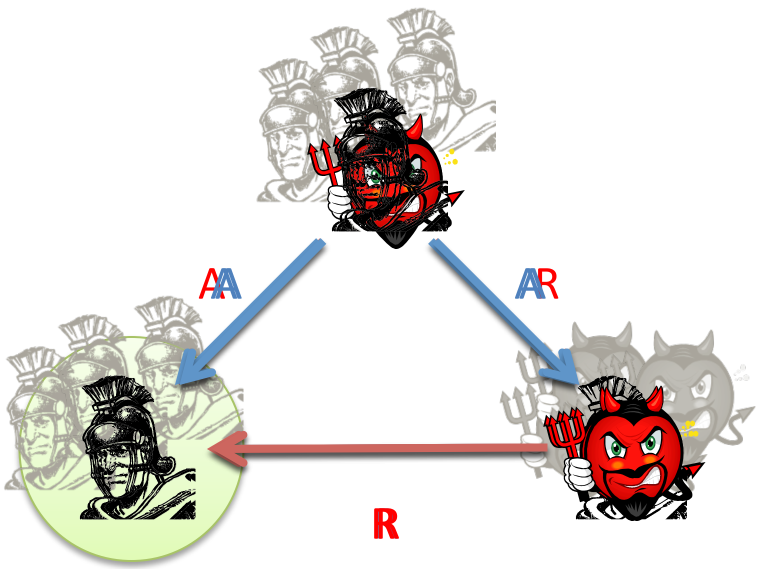
BFT with Un-Auth. Messages

- (A1) Every message is delivered correctly
- (A2) The receiver knows who sent the message
- (A3) The absence of a message can be detected

Impossibility Results

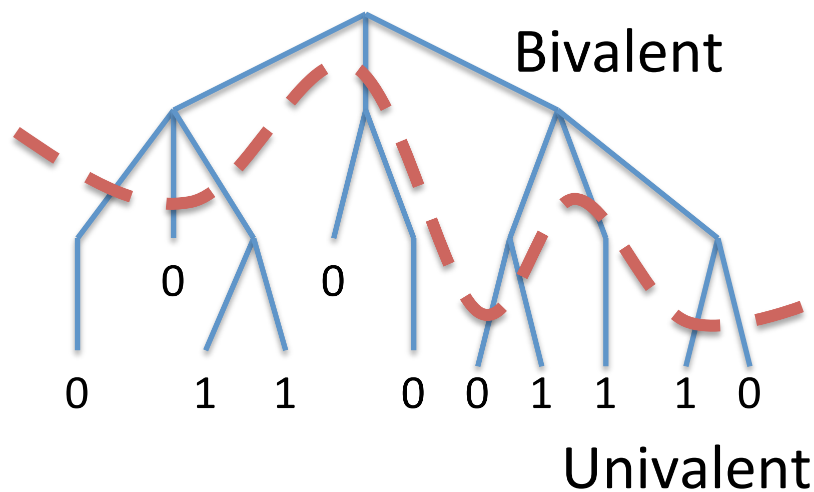
Sync. Communication [LSP80]

- Impossible: n ≤ 3m + 1



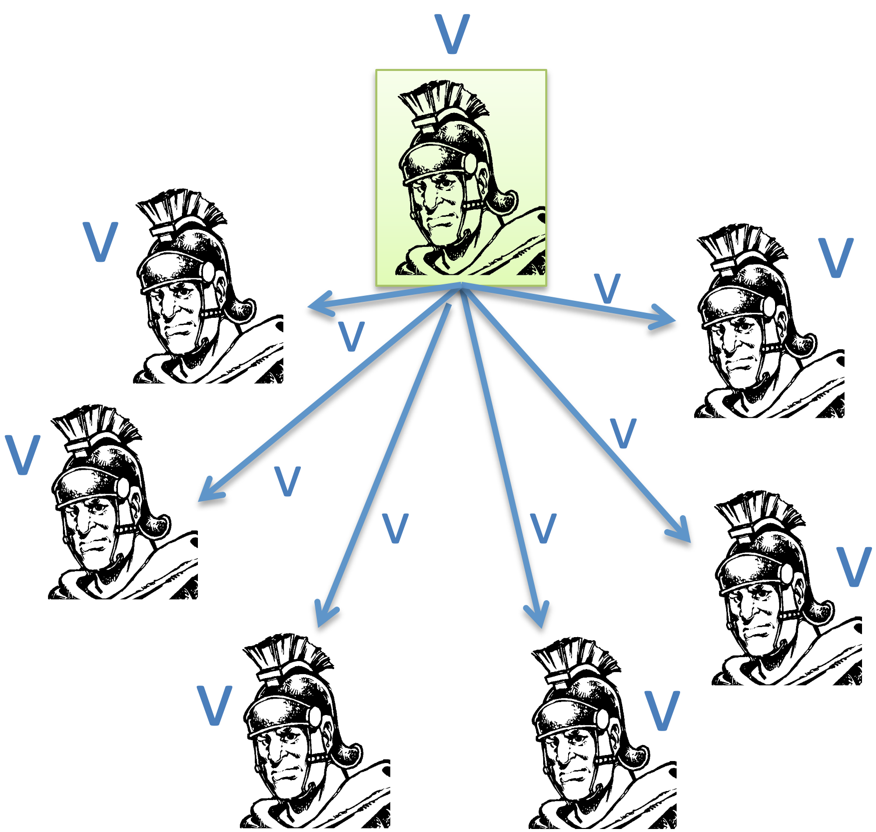
Async. Communication [FLP82]

- Impossible: m ≥ 1

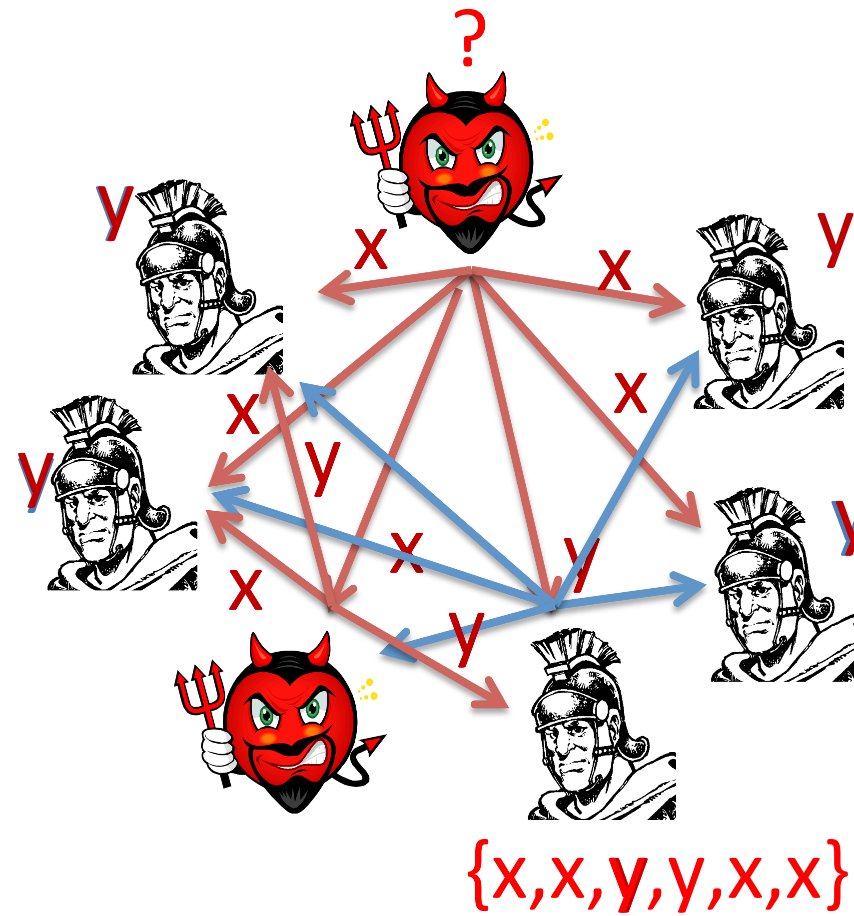


A Solution with Oral Messages
$$(n \geq 3m + 1)$$

OM(i, v, n, 0):



OM(i, v, n, m):



A Solution with Oral Messages ($n \geq 3m + 1$)

- *OM(i, v, n, 0):*
  - Com. Gen. *i* sends $v_{i,j} = v$ to every Lt. *j*
  - All Lt. *j* uses the value $v_{i,j}$ (default = RETREAT)
- *OM(i, v, n, m):*
  - Com. Gen *i* sends $v_{i,j} = v$ to every Lt. *j*
  - Lt. *j* initiates *OM(j,* $v_{i,j}$*, m-1, n-1)* to send the value $v_{i,j}$ to each of the *n-2* other Lts. (default = RETREAT)
  - Let $v_{j,k}$ be the value Lt. *k* received from Lt. *j* in step 2, default RETREAT. Lt. *k* uses the value *MAJ(*v_1*, …,* v_{n-1} *).*

Byzantine Fault Tolerance (Results)

(m = traitors, n = total)	Synchronous	Semi-Sync	Asynchronous
Oral Messages: Negative	$n \leq 3m$ [LSP80]		$m \geq 1$ [FLP82]
Positive	$n \geq 3m+1$ [LSP80]		
Authenticated: Negative			
Positive			

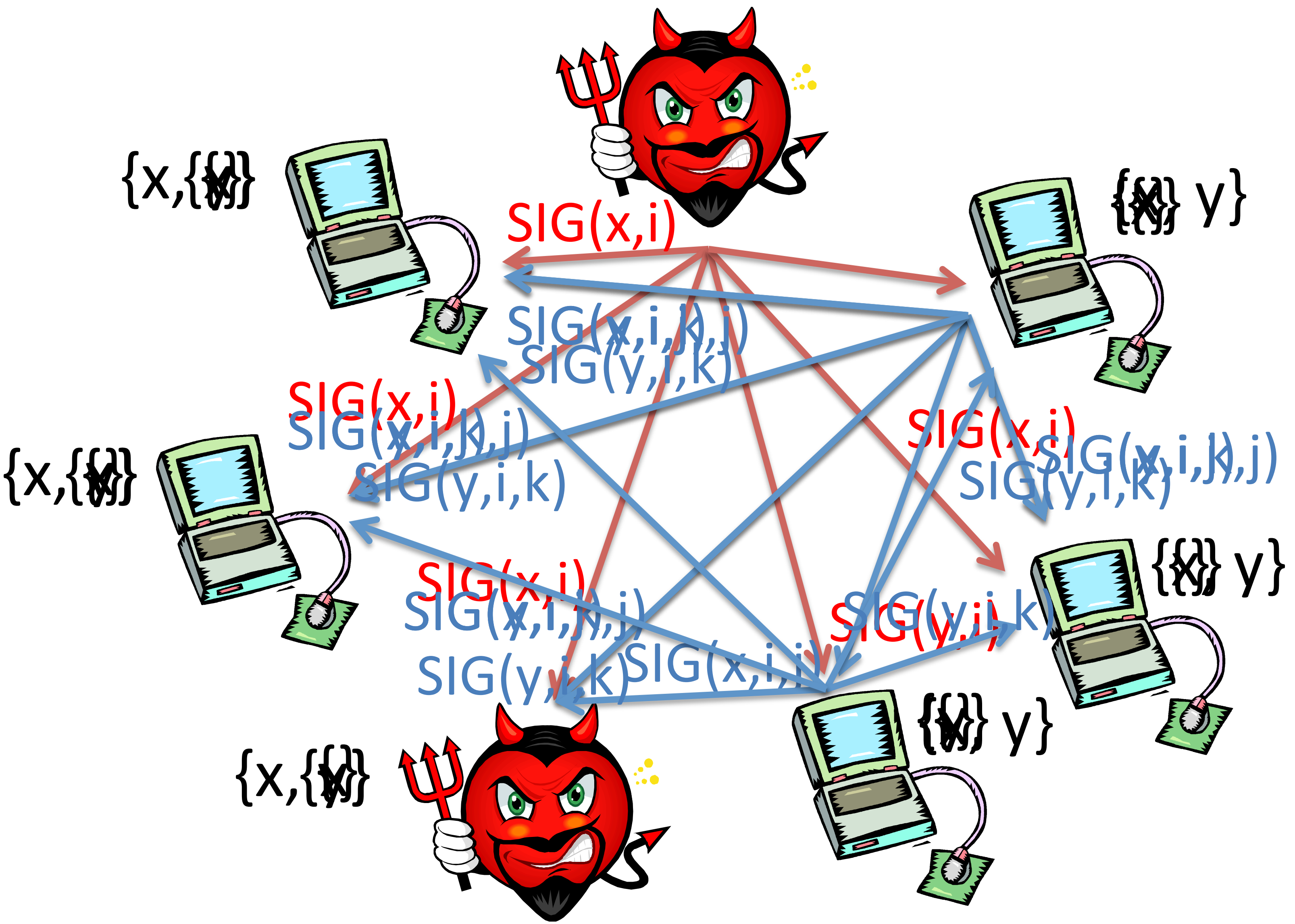
Byzantine Fault Tolerance (Results)

(m = traitors, n = total)	Synchronous	Semi-Sync	Asynchronous
Oral Messages: Negative	$n \leq 3m$ [LSP80]		$m \geq 1$ [FLP82]
Positive	$n \geq 3m+1$ [LSP80]		
Authenticated: Negative			$m \geq 1$ [FLP82]
Positive			

BFT with Auth. Messages

- (A1) Every message is delivered correctly
- (A2) The receiver knows who sent the message
- (A3) The absence of a message can be detected
- *(A4) A loyal general's signature cannot be forged, alterations are detected, authenticity can be verified by all*

A Solution with Signed Messages



A Solution with Signed Messages

- *SM (m):*
  - $V_i = \{\}$
  - Com. Gen. *i* sends $v_{i,j}:0$ to each Lt. *j*
  - If Lt. *j* receives $v:0:k_1 : \ldots : k_\ell$ and $v \notin V_j$, then
    - Lt. *j* adds *v* to V_j
    - If $k < m$, then he sends the message $v:0:k_1: \ldots :k_\ell:i$ to all Lt. $s \neq 0,k_1, \ldots, k_\ell$
  - When Lt. *j* will receive no more messages, he follows $MAJ(V_j)$

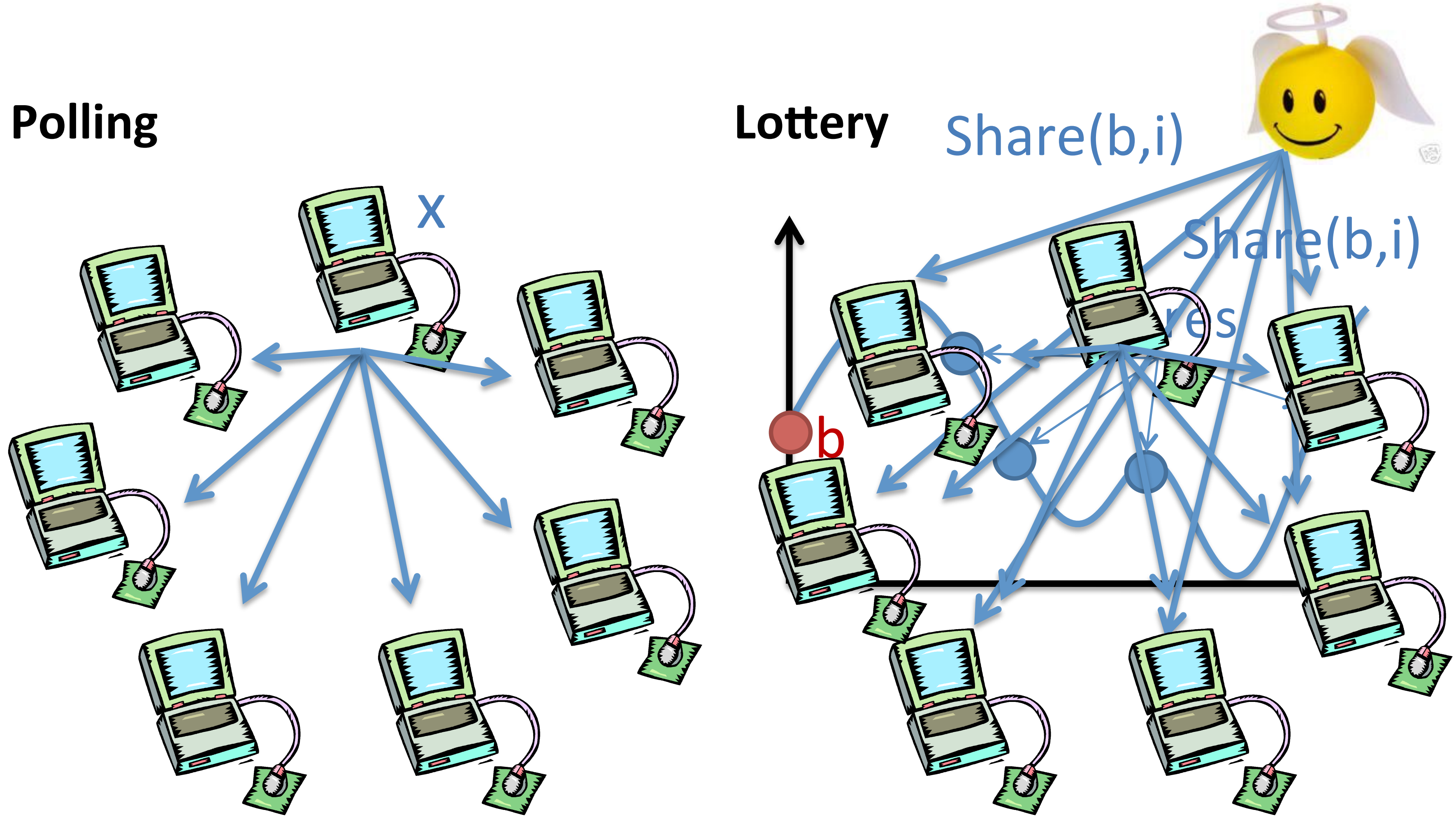
So what’s wrong?

- Synchronous
- Unscalable
- (Inefficient)

M. Rabin
Randomized Byzantine Generals (1983)

- PhD Princeton (1956)
- Professor: MIT, Hebrew University, Harvard
- Nondeterminism, primality testing, encryption, oblivious transfer, string search, auctions
- Turing Award 1976

A Randomized Solution



Temp = MAJ({x,x,y,y,x,z,x})

b = 0 & count(Temp) ≥ n/2
b = 1 & count(Temp) ≥ n – 2m

So what is wrong?

- [LSP80]
  - Synchronous
  - Unscalable
- [Rabin83]
  - Still too inefficient
- Rampart
- SecureRing

Fifteen years later…

M. Castro and B. Liskov
Practical Byzantine Fault Tolerance (1999)

- Miguel Castro
  - PhD MIT 2001
  - MSR Cambridge
- Barbara Liskov
  - PhD Stanford 1968
  - MIT
  - Distributed systems, fault tolerance, prog. languages (OOP)
  - Turing Award 2008

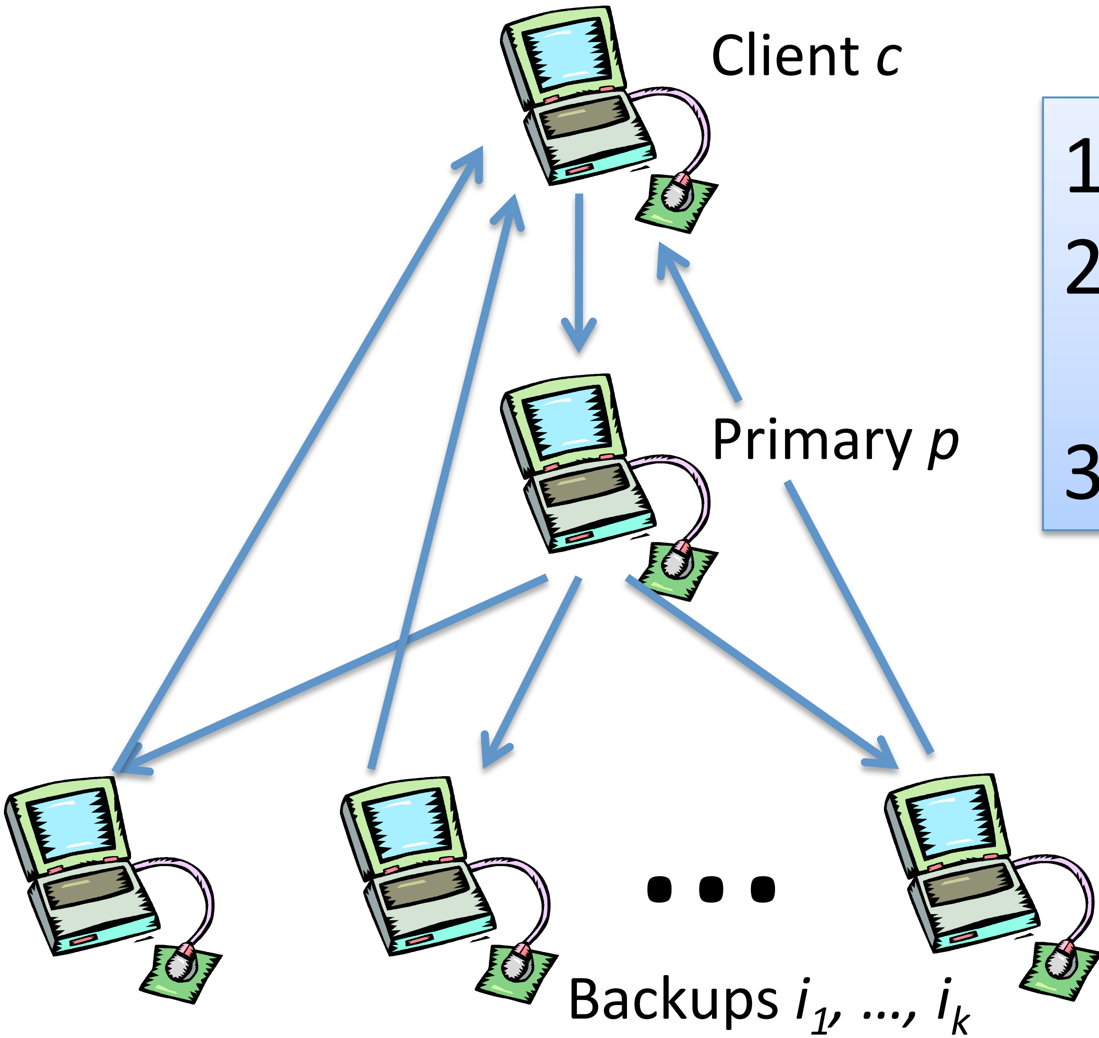
PBFT Assumptions

- Asynchronous environment/ communication
  – delay(t) doesn't grow faster than t indefinitely
- Independent, Byzantine node failures
  – At most n-1/3 faulty
- Authenticated messages
  – Adversary can't break signatures/ MACs

Byzantine Fault Tolerance (Results)

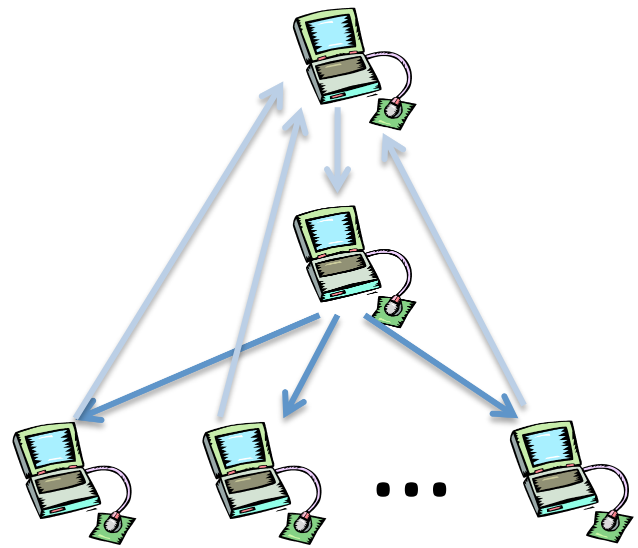
(m = traitors, n = total)	Synchronous	Semi-Sync	Asynchronous
Oral Messages: Negative	n ≤ 3m [LSP80]		m ≥ 1 [FLP82]
Positive	n ≥ 3m+1 [LSP80]		
Authenticated: Negative			m ≥ 1 [FLP82]
Positive	n ≥ 1 [LSP80]	n ≥ 3m+1 [CL99]	

State Machine Replication



1) $\langle \text{Request}, o, t, c \rangle_{\sigma_c}$
2) Multicast Request (3-phase protocol)
3) $\langle \text{Reply}, v, t, c, i, r \rangle_{\sigma_i}$

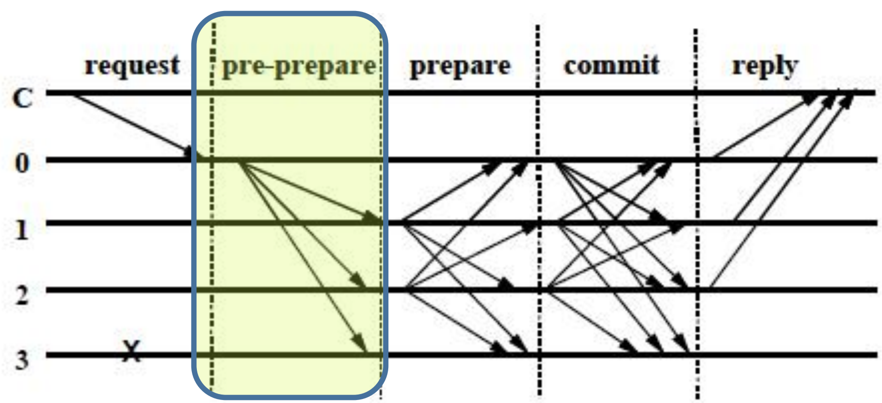
Multicast (3-phase)



1) $\langle \langle \text{Pre-prepare, v, n, d} \rangle_{\sigma p}, \text{m} \rangle$
2) $\langle \text{Prepare, v, n, d, i} \rangle_{\sigma i}$

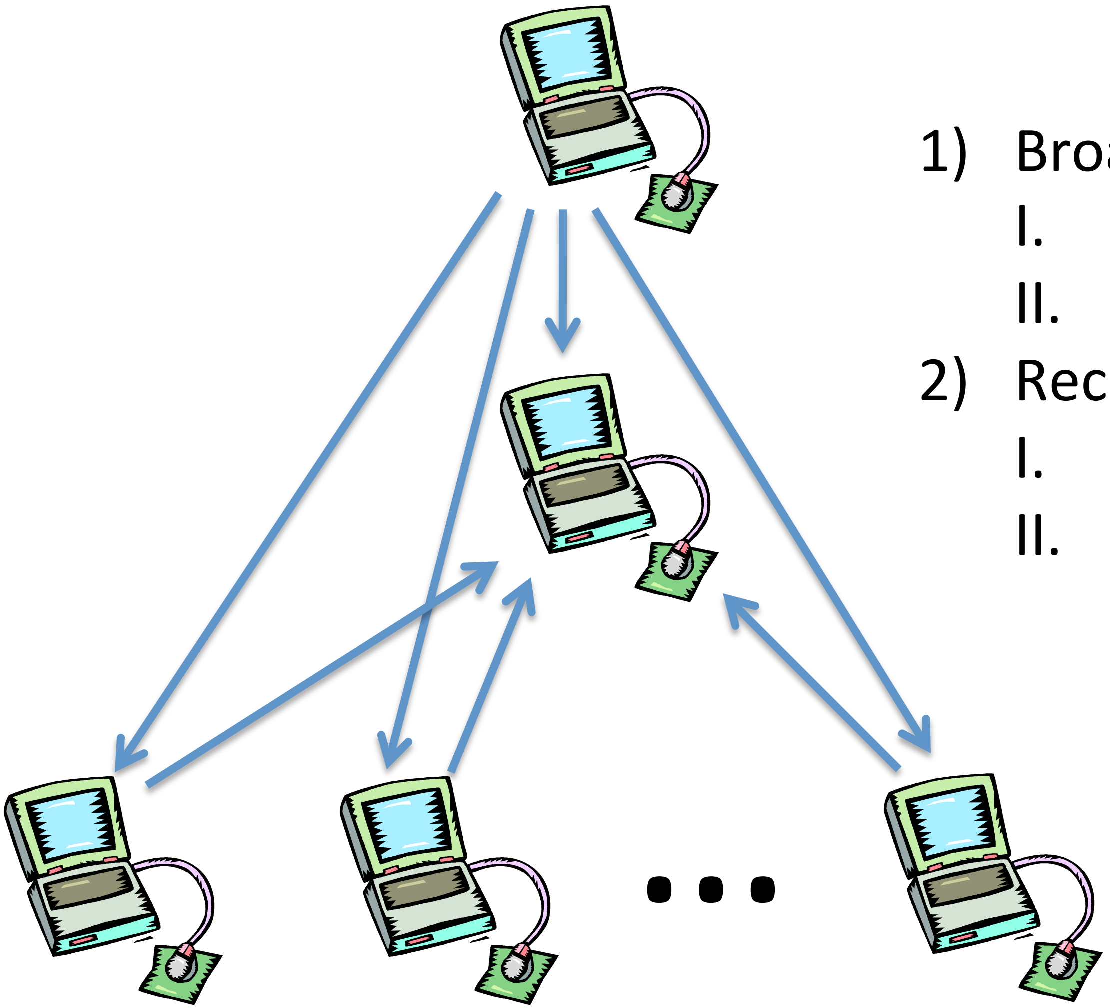
Successfully prepared if received 2m different prepared copies (⇒ honest agree on total ordering)

3) $\langle \text{Commit, v, n, D(m), i} \rangle_{\sigma i}$



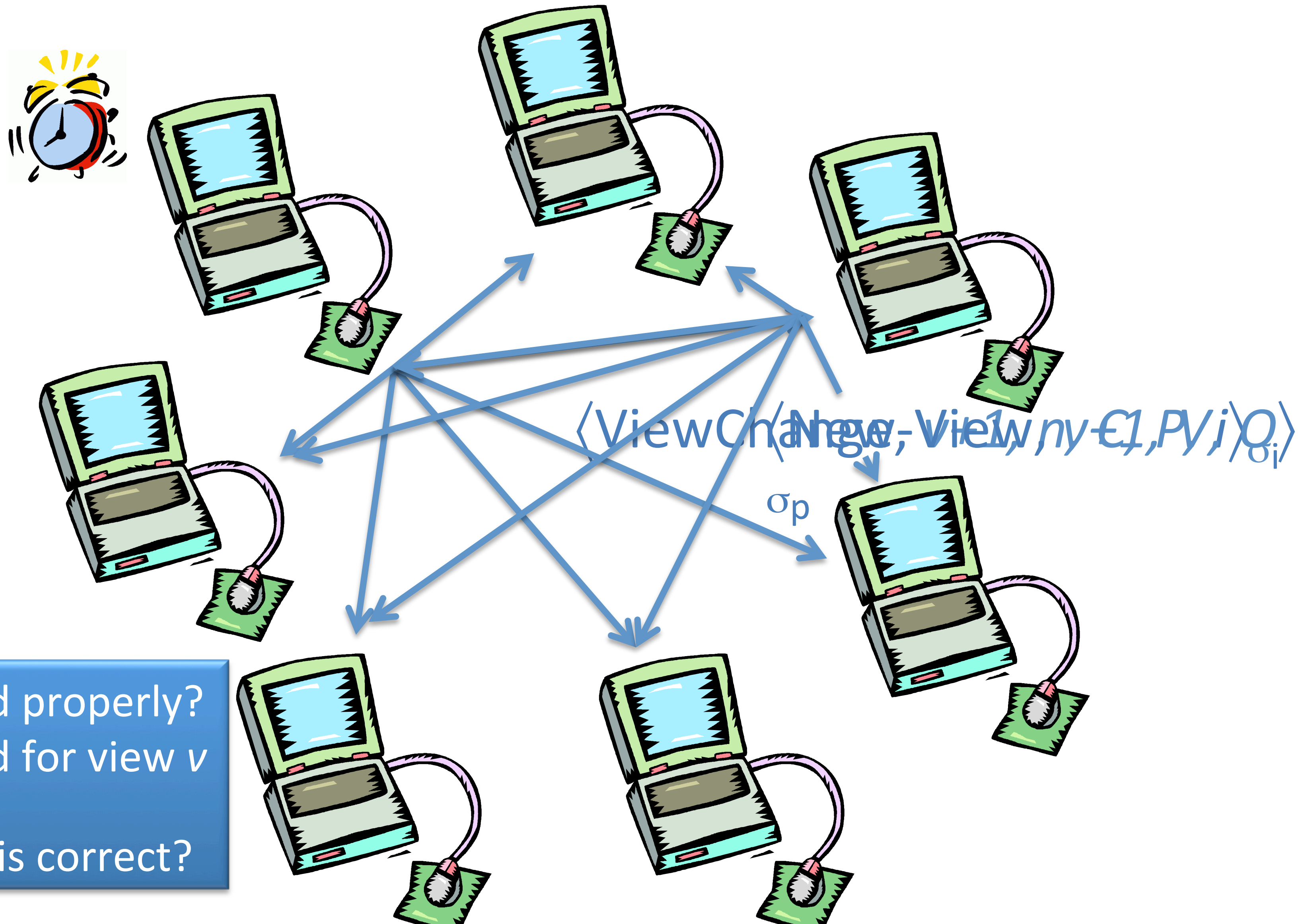
Backup Plan

(*c* doesn't receive *m+1* replies)



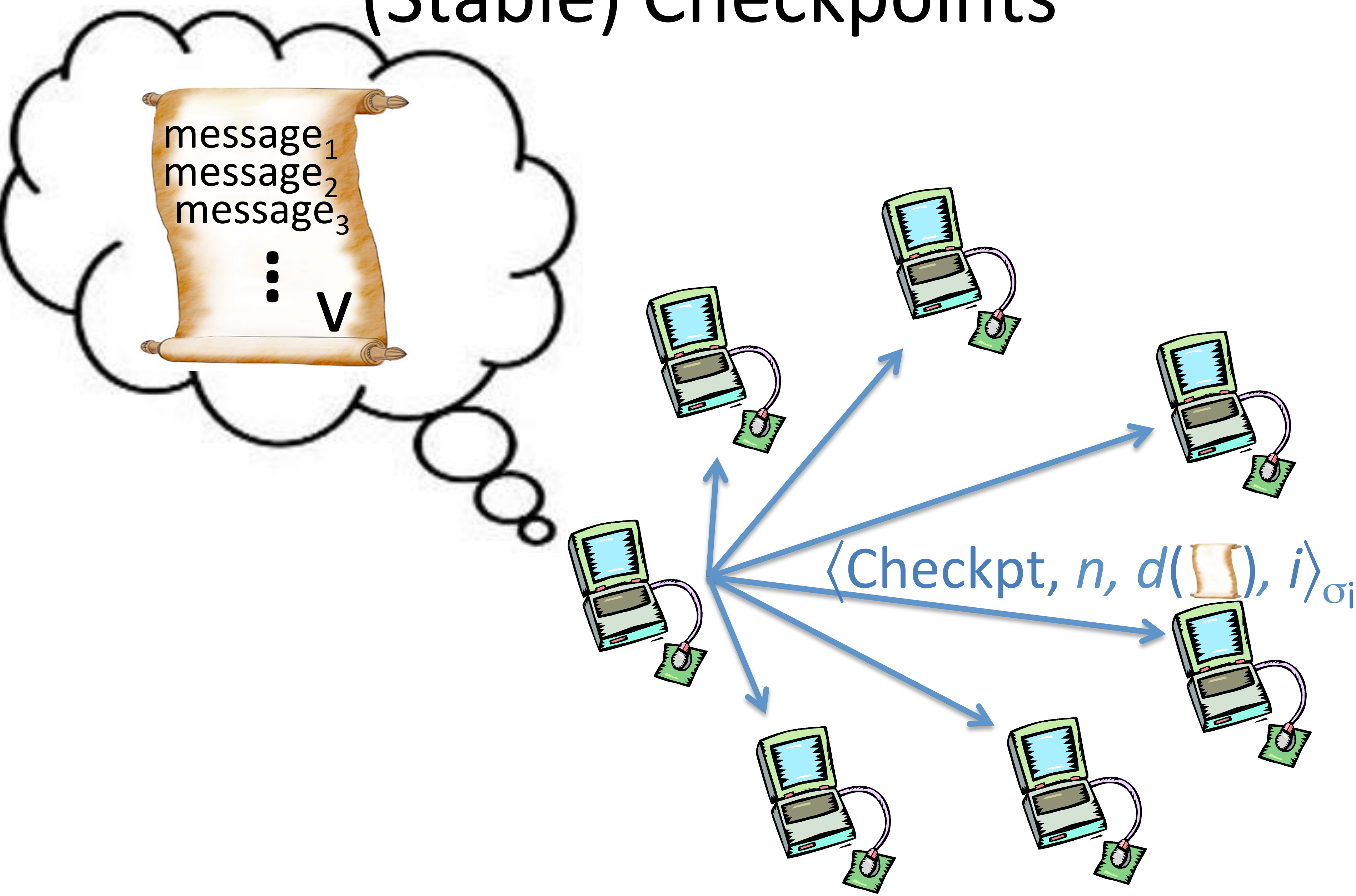
1) Broadcast $\langle \text{Request}, o, t, c \rangle_{\sigma_c}$
   I. Resend or
   II. Relay request to *p*
2) Recover
   I. If p multicasts continue
   II. Else Change View

View Change



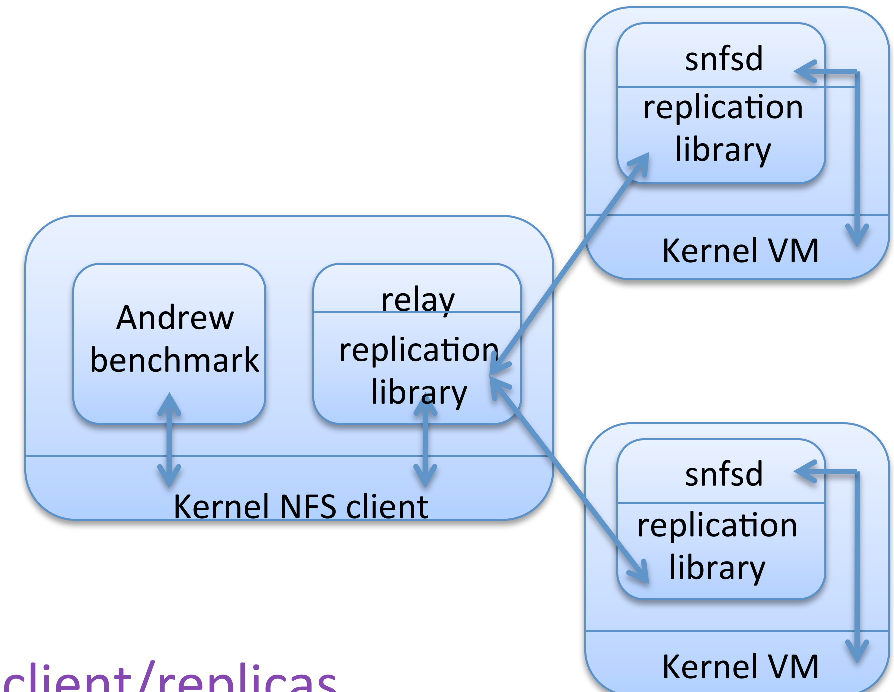
- Signed properly?
- *V* valid for view *v +1*?
- Set *O* is correct?

(Stable) Checkpoints



BFS: A Byzantine-Fault-Tolerant File System

- Replication Library
  - Client: *invoke*
  - server: *execute*
    make_checkpoint
    delete_checkpoint
    get_digest
    get_checkpoint
    set_checkpoint



- Relay
  - Mediate comm. b/n NFS and client/replicas
- snfsd
  - NFS v2 daemon
  - Implemented using fixed-size memory-mapped file

Performance

phase	BFS strict	BFS r/o lookup	NFS-std
1	.55 (-69%)	.47 (-73%)	1.75
2	9.24 (-2%)	7.91 (-16%)	9.46
3	7.24 (35%)	6.45 (20%)	5.36
4	8.77 (32%)	7.87 (19%)	6.60
5	38.68 (-2%)	38.38 (-2%)	39.35
total	64.48 (3%)	61.07 (-2%)	62.52

Table 3: Andrew Benchmark (BFS vs. NFS-std)

Thoughts?