HYBRID FAULT MODELS

- The material in the following slides is courtesy of Dr. Azad Azadmanesh (UNO), i.e., they were prepared using many slides he had given to me.
  - Thanks Azad!
- This discussion of fault models uses the "dependability" fault assumptions and does not focus on malicious act. We will discuss what that really means in class.

REFERENCES

- Leslie Lamport, R. Shostak, M. Pease, "The Byzantine Generals Problem", ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, pp. 382-401, Jul 1982.
- F.J. Meyer, D.K. Pradhan, Consensus with Dual Failure Modes, 17'th Int'l Symposium on Fault Tolerant Computing (FTCS-17), pp. 48-54, July 1987.
- P. Thambidurai, Y-K. Park, "Interactive Consistency with Multiple Failure Modes", 7th Reliable Distributed Systems Symposium, pp. 1-8, Oct. 1988.
- A. Azadmanesh, R.M. Kieckhafer, "New Hybrid Fault Models for Asynchronous Approximate Agreement", IEEE Transactions on Computers, Vol. 45, No. 4, pp. 439-449, April 1996.
- A. Azadmanesh, R.M. Kieckhafer, "Exploiting Omissive Faults in Synchronous Approximate Agreement", IEEE Transactions on Computers, Vol. 49, No. 10, pp. 1031-1042, Oct. 2000.

OVERVIEW

- The Byzantine Fault Model (BYZ-1)
- Hybrid Fault Models
  - MPH-2
  - TPH-3
  - OTH-5
- Taxonomy of Fault Models
- Advantages of Hybrid Fault Models
- Summary of Fault-Tolerance

BYZANTINE FAULT MODEL

- Assumes that every fault is Byzantine (asymmetric)
- Oral t-fault-tolerance requires
  - $N \geq 3t+1$ --independent processes
  - $r \geq t$ -- rounds of re-broadcast
- Observations (in the classic dependability sense):
  - Asymmetry is hard to achieve (especially with busses).
  - Asymmetric faults are very rare.
  - The majority of faults are very likely to be less severe.
  - Designs based on this model are likely to be strongly over-conservative.
  - Designs based on this model are normally used for applications which can not afford to tolerate any form of faults.

HYBRID FAULT MODELS

- Objective of Hybrid Fault Modeling is:
  - A fault model with fault "modes" of different severities.
  - Fault modes must be:
    - Mutually Exclusive
    - Collectively Exhaustive
- Motivation:
  - Allow a set of coincident faults to be of mixed modes (severities).
  - Design systems which are not so over-conservative.
  - Analyze agreement algorithms in the presence of mixed-mode faults.

MPH-2 FAULT MODEL (MEYER/PRADHAN)

- Partitions all faults into:
  - Benign faults
  - Malicious faults
- Benign fault: Immediately self-evident to all receivers.
  - Can not send an erroneous value that can not be detected.
  - Can not cause the system to undergo incorrect state transition.
  - Also called self-incriminating, fail-notify, dormant, manifest.
  - Must be pre-agreed upon by all non-faulty nodes:
    - Not immediately evident in all authors' definitions
    - Embedded in assumptions of proofs.
    - All nodes simply ignore benign faults.

MPH-2 FAULT MODEL

- Examples of Benign faults:
  - Out of bound data
  - Timing-fault
  - Crash fault
- Malicious fault: Does not identify itself as faulty:
  - Any faulty behavior that is not considered benign.
  - Not self-evident to all nodes.
  - Can behave symmetrically or asymmetrically.
  - Must be handled as Byzantine.

MPH-2 FAULT MODEL

- Total Number of Faults
  - m = number of malicious faults
  - b = number of benign faults
  - Thus total number of faults t = m+b
- Oral Message Fault-Tolerance: OM(m)
- $N \geq 3m + b + 1$ -- independent processes
  - $r \geq m$ -- rounds of re-broadcast

TPH-3 FAULT MODEL

- Thambidurai & Park model
- BenignFault: same as MPH-2
- Symmetric Malicious:
  - Can send an erroneous value.
  - Must broadcast the same error to all receivers.
- Asymmetric Malicious:
  - Can send different errors to different receivers.

TPH-3 FAULT MODEL

- Total Number of Faults:
  - a = Number of asymmetric faults
  - s = Number of symmetric faults
  - b = Number of benign faults
  - Thus: Total number of faults $t = a + s + b$
- Oral Message Fault-Tolerance: OM(a)
  - $N \geq 2a + 2s + b + r + 1$
  - $r \geq a$

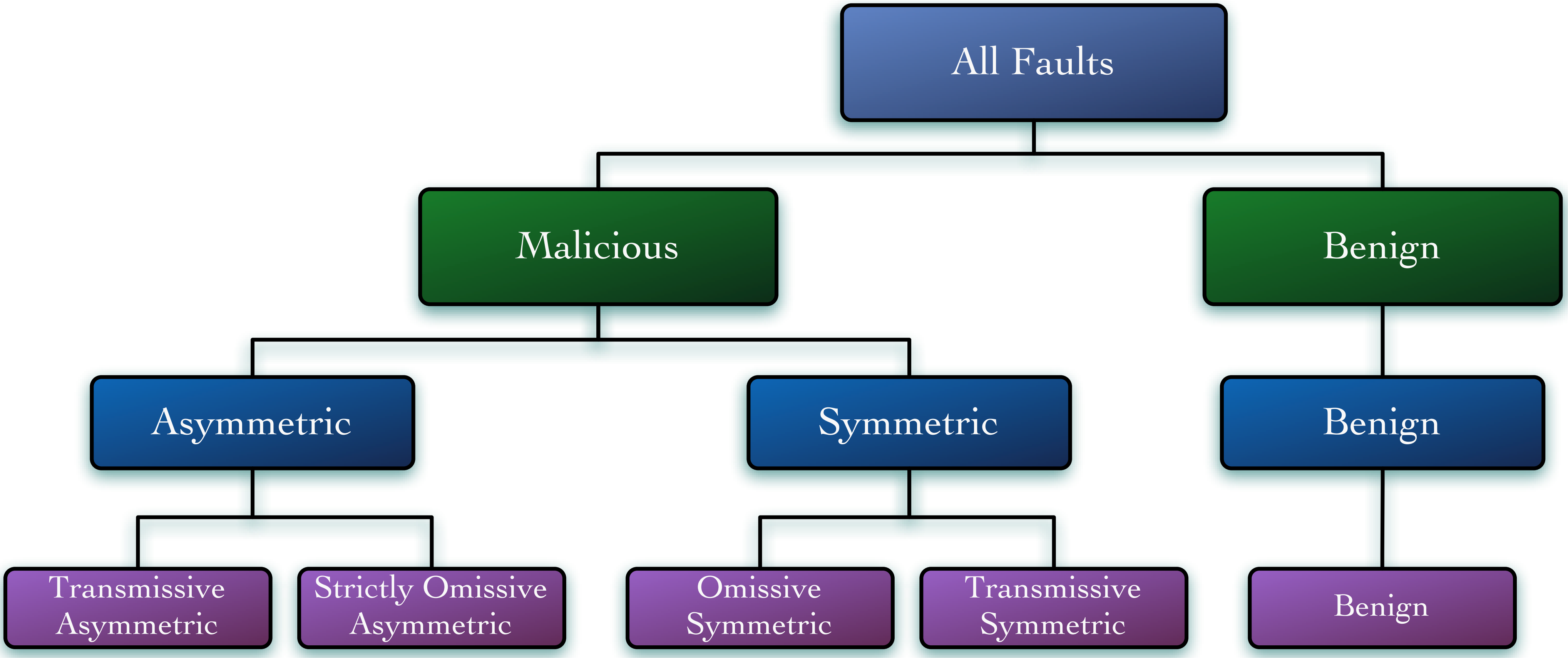
OTH-5 FAULT MODEL

- OTH stands for Omissive & Transmissive Hybrid model
- Benign: Same as TPH-3 and MPH-2 benign
- Omissive Symmetric:
  - Does not send a value to any receiver.
  - Unlike a benign fault, it is not previously diagnosed and agreed to by all nodes.
- Transmissive Symmetric: same as TPH-3 symmetric.
- Strictly Omissive Asymmetric: Can transmit only:
  - Correct value to some receivers.
  - And <u>No</u> value to other receivers.
- Transmissive Asymmetric: Same as TPH-3 asymmetric.

OTH-5 FAULT MODEL

- Total Number of Faults
  - a‘ = Number of transmissive asymmetric faults
  - ωa = Number of strictly omissive asymmetric faults
  - s‘ = Number of transmissive symmetric faults
  - ωs = Number of omissive symmetric faults
  - b = Number of benign faults
  - Thus total number of faults: $t = (a' + \omega a) + (s' + \omega s) + b$
- Oral Message Fault-Tolerance
  - OTH has not been applied to Byzantine Agreement
  - Has been applied to "Approximate" Agreement:
    - $N \geq 3a' + 2s' + \omega a + \omega s + b + 1$

TAXONOMY OF FAULT MODELS



FT ADVANTAGES OF HYBRID FAULT MODELS

1) More accurate model of the system: Less "overly" conservative

2) Resulting reliability estimates are better.

Consider N = 6:
OM is capable of tolerating 1 fault.
Whereas TPH-3 can tolerate: 1 asymmetric and 1 symmetric fault, or
1 asymmetric and 2 benign faults.

Also, with N=4, OM is capable of still tolerating 1 fault.
Therefore the system reliability when N=6 is less, and

Thus, it is better to turn off the additional nodes.

FT ADVANTAGES OF HYBRID FAULT MODELS

3) Smarter degradation

Consider the incompatible fault scenarios:

Scenario1: a=1, s=0, b=0

Scenario2: a=0, s=2, b=0

Consider a 5-node system:

r=0 can handle scenario 2, but not scenario1

(simple majority vote), (see N for TPH-3).

r=1 can handle scenario1, but not scenario 2.

Thus: By specifying the number of rounds, the algorithm can be ‘‘tuned’ for the most likely scenarios.

FT ADVANTAGES OF HYBRID FAULT MODELS

- Requirements for success:
  - To have a good estimate of fault rates λa , λs , λb
    - Typically $\lambda a << \lambda s << \lambda b$
  - To have a good estimate of recovery rates ρa , ρs
    - Typically $\rho a < \rho s$

SUMMARY

- BYZ-1
  - $N \geq 3t+1$
  - $r \geq t$
- MPH-2
  - $N \geq 3m + b + 1$
  - $r \geq m$
- TPH-3
  - $N \geq 2a + 2s + b + r + 1$
  - $r \geq a$
- OTH-5
  - Byzantine Agreement: Unknown at this time. ApproximateAgreement: $N \geq 3a' + 2s' + \omega a + \omega s + b + 1$

DATA AGGREGATION

- There are different yet similar definitions:
  - Ability to provide global information for purposes of network management and user services.
  - A set of functions that provide components of a distributed system access to global information.
- Reasons to do Data Aggregation (DA)
  - To coordinate tasks.
  - The need for node/component duplication for higher performance.
  - The need for redundancy for higher fault tolerance.

DATA AGGREGATION

- Data Aggregation has gone by other names:
  - Data Fusion
  - Approximate Agreement
  - Consensus
  - Distributed Agreement

DATA AGGREGATION

- General Scenario
  - Each node in the network holds a value.
  - Nodes exchange values to decide on values.
- Agreement conditions:
  - Agreement: The decided values, one value by each node, are within a predefine tolerance of each other.
  - Validity: The decided values are within the range of initial values held by non-faulty nodes.

A ROUND OF COMMUNICATION

- Agreement is reached executing multiple rounds.
- Each round consists of:
  - 1. Broadcast: Each node broadcasts its value to others.
  - 2. Collect: Each node forms a multiset of values.
  - 3. Filter: Select values to vote with.
  - 4. Average: Average the selected values to vote.

MAJOR EVOLUTIONS

- Static networks
  - Multiprocessor/multiprocessing systems
    - Mostly concentrated on Fully Connected Networks (FCN)
    - Small scale critical systems: power plants, aircrafts, automobiles
  - Distributed Networks
    - Limited focus on Partially Connected Networks (PCN)
    - Large scale distributed systems (Internet)

MAJOR EVOLUTIONS

- Mobile/Adhoc networks
  - Include both FCN and PCN
  - Growing PCN applications: tactical military operations, tracking endangered species, Unmanned Autonomous Vehicles (UAV), etc.

FULLY CONNECTED NETWORKS

- Most agreement algorithms are devised for FCNs.
- Full exchange of values in a round is immediate.
- Diameter of values shrinks in each round (single-step).
- Agreement is reached gradually by shrinking diameter in each round.

EXAMPLE: MEDIAN SELECT ALGORITHM

- 5 nodes with real values in [0,1]
- 4 (of 5) correct nodes posses the values: <0,0,1,1>
- The faulty node behaves asymmetrically.
- After broadcasting, collecting, filtering, and selecting:
  - Node i: <0,0,0,1,1> thus median = 0
  - Node j: <0,0,1,1,1> thus median = 1
- Result:
  - Validity: voted values are within range of correct values.
  - Not convergent: values are still within [0,1].

EXAMPLE: MIDPOINT SELECT ALGORITHM

- 4 nodes with real values in [0,1]
- 3 (of 4) correct nodes posses the values: <0,1,1>
- The faulty node behaves asymmetrically.
- After broadcasting, collecting, filtering, and selecting:
  - Node i: <0,0,1,1> thus midpoint = 0.5
  - Node j: <0,1,1,1> thus midpoint = 1
- Result:
  - Validity: voted values are within range of correct values.
  - Convergent: voted values are < diameter in [0,1].

HOW TO ACHIEVE AA CONDITIONS

- Three general approaches:
  - Mean Select Reduce (MSR)
  - MSR algorithms: Remove the extreme-end values.
  - Egocentric algorithms: Replace the extreme-end values with your own value.
  - Egophobic algorithms: Replace the extreme-end values with your own plus a predefined threshold.

MSR: REMOVE THE EXTREME--END VALUES

- Example:
  - Assume 5 nodes with values in the range: 0…1
  - 4 of the nodes posses the values: <0,0,1,1,>
  - The one faulty node (t=1) behaves asymmetrically.
  - Use "fault tolerant mean" function.
  - Case 1: Assume, the faulty node transmits a value outside of the range.

0] 0 1 1 [1.5	-1] 0 0 1 [1
0 1 1 → (0+1+1)/3 = 0.66	0 0 1 → (0+0+1)/3 = 0.33

- Average is convergent [(0.66-0.33) < (1-0)] & valid.

MSR: EXAMPLE CONT.

- Example, Cont.:
  - Case 2: Assume, the faulty node transmits a value within the range.

0] 0 0.5 1 [1	0] 0 0.9 1 [1
0 0.5 1 → (0+0.5+1)/3=0.5	0 0.9 1 → (0+0.9+1)/3=0.63

- Average is convergent [(0.63-0.5) < (1-0)] & valid.

EGOCENTRIC: REPLACE THE EXTREME--END VALUES WITH YOUR OWN

- Example:
  - Assume 5 nodes with values in the range: 0… 1
  - 4 of the nodes posses the values: <0,0,1,1,>
  - The one faulty node (t=1) behaves asymmetrically.
  - Assume two correct nodes i & j performing the voting hold the values 1 & 0 respectively.
  - Use Fault Tolerant Mean function.

EGOCENTRIC: REPLACE THE EXTREME--END VALUES WITH YOUR OWN, CONT.

- Example, Cont.:
  - Case 1: Assume, the faulty node transmits the values (-0.5 & 1.5) outside of the range.
    - node i receives: 0 0 1 1 1.5
      - 1 0 1 1 1 => (1+0+1+1+1)/5 = 0.8
    - node j receives: -0.5 0 0 1 1
      - 0 0 0 1 0 => (0+0+0+1+0)/5 = 0.2
    - The result is both valid and convergent.

EGOCENTRIC: REPLACE THE EXTREME--END VALUES WITH YOUR OWN, CONT.

- Example, Cont.:
  - Case 2: Assume, the faulty node transmits values (0.2 & 0.7) within the range.
    - node i receives: 0 0 0.2 1 1
      - 1 0 0.2 1 1 => (1+0+0.2+1+1)/5 = 0.64
    - node j receives: 0 0 0.7 1 1
      - 0 0 0.7 1 0 => (0+0+0.7+1+0)/5 = 0.34
    - The result is both valid and convergent.

EGOCENTRIC: REPLACE THE EXTREME--END VALUES WITH YOUR OWN, CONT.

- Example, Cont.:
  - Case 3: Assume, the faulty node transmits value 0.2 (within range) to i and 1.5 (outside of range) to j.
    - node i receives: 0 0 0.2 1 1
      - 1 0 0.2 1 1 => (1+0+0.2+1+1)/5 = 0.64
    - node j receives: 0 0 1 1 1.5
      - 0 0 1 1 0 => (0+0+1+1+0)/5 = 0.4
    - The result is both valid and convergent.

SINGLE STEP CONVERGENCE

$\mathbf{V} = \langle v_1, \ldots, v_V \rangle$, where $V = |\mathbf{V}|$, is the multiset of values received and sorted in ascending order.

$\rho(\mathbf{V}) = [\,\min(\mathbf{V}), \max(\mathbf{V})\,]$, is the range of values spanned by $\mathbf{V}$.

$\delta(\mathbf{V}) = [\,\max(\mathbf{V}) - \min(\mathbf{V})\,]$, is the diameter of $\mathbf{V}$.

$\mathbf{U}_{all}$ = The multiset of all correct values received by non-faulty nodes.

During each round of voting, each non-faulty node *i* broadcasts its initial value to all nodes including itself.
Then it collects all the values received into the voting multiset V_i.
Node *i* then applies a function *F* to V_i to attain its latest estimate (voted value) for the round, which it uses as its initial value in the next round.

SINGLE STEP CONVERGENCE

- An algorithm is single-step-convergent if both of the following conditions are true following every round of voting:

C1: VALIDITY—For each nonfaulty process i, the voted value is within the range of correct values, i.e., $F(\mathbf{V}_i) \in \rho(\mathbf{U}_{all})$.

C2: CONVERGENCE—For each pair of nonfaulty processes, i and j, the difference between their voted values is strictly less than the diameter of the correct values received, i.e., $|F(\mathbf{V}_i) - F(\mathbf{V}_j)| \leq C\delta(\mathbf{U}_{all})$, where $0 \leq C < 1$.

SINGLE STEP CONVERGENCE

In convergence condition C2 above, i.e.,

$$|F(\mathbf{V}_i) - F(\mathbf{V}_j)| \leq C\delta(\mathbf{U}_{all})$$

parameter C, called the Convergence Rate, is the primary performance measure of a voting algorithm. The constraint $0 \leq C < 1$ ensures that the algorithm is indeed single-step convergent. The smaller the values of C, the faster the voted values converge.

MSR ALGORITHMS

- Mean Select Reduce (MSR)
- MSR algorithms can handle the following fault modes:
  - Benign (b)
  - Symmetric (s)
  - Asymmetric (a)
  - Thus: $t = a + s + b$

MSR ALGORITHMS

N = The total number of processes in the system.

τ = The maximum number of malicious errors that could be received by a process. This number is known a priori and is identical for all nonfaulty processes.

$\mathbf{V}_i$ = The multiset of values received in a given round by nonfaulty process i. The number of elements in $\mathbf{V}_i$ is $V_i = |\mathbf{V}_i|$.

$\mathbf{M}_i$ = $Red^{\tau}(\mathbf{V}_i)$, the *Medial Multiset* of $\mathbf{V}_i$. The number of elements in $\mathbf{M}_i$ is $M_i = |\mathbf{M}_i| = V_i - 2\tau$.

$\mathbf{S}_i$ = $Sel_{\sigma_i}(\mathbf{M}_i) = Sel_{\sigma_i}(Red^{\tau}(\mathbf{V}_i))$, the *Selected Multiset* generated by $F(\mathbf{V}_i)$. The number of elements in $\mathbf{S}_i$ is $\sigma_i = |\mathbf{S}_i|$.

MSR ALGORITHMS

The voting function:

$$F(V)=\text{mean}[Sel_{\sigma}(Red^{\tau}(V))]$$

where

$\tau = (a + s)$, maximum number of erroneous values

Red^{τ} = The subsequence M remaining after removing the τ extreme values

Sel_{σ} = The subsequence S of size σ , after selecting σ elements from M

EXAMPLE

$$
\begin{array}{lcccccccccc}
\boldsymbol{V} = & v_1, & v_2, & v_3, & v_4, & v_5, & v_6, & v_7, & v_8, & v_9, & v_{10} \\
 & & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \\
\boldsymbol{M} = & & & m_1, & m_2, & m_3, & m_4, & m_5, & m_6 & & \\
 & & & \downarrow & & \downarrow & & & \downarrow & & \\
\boldsymbol{S} = & & & s_1, & & s_2, & & & s_3 & & \\
 & & & \downarrow & & \downarrow & & & \downarrow & & \\
 & & & m_{k(1)}, & & m_{k(2)}, & & & m_{k(3)} & &
\end{array}
$$

$$\Rightarrow \quad s_i = m_{k(i)}$$

TPH-3 MODEL

- With TPH-3 we have N ≥ 3a + 2s + b + 1
- With Lamport we have N ≥ 3t +1
- It was shown [Kieckhafer & Azadmanesh 1994] that single-step convergent MSR voting algorithms can exist if

$$N \geq 3a + 2s + b + 1 = 3t - (s + 2b) + 1$$

- Examples of algorithms that achieve this bound are
  - Fault-Tolerant Midpoint (if a>0)
  - Simple Median Select (if a=0)

DEFINITION

$g, h \in \{1, \ldots, \sigma\}$, where $g \le h$:

$(h - g) \equiv$ the distance between s_g and s_g,

$k(h) - k(g) \equiv$ the distance between $m_{k(g)}$ and $m_{k(h)}$,

Define:

$$\gamma_\alpha \equiv \min(h - g) \text{ such that } k(h) - k(g) \ge \alpha, \quad \forall\, g, h \in \{1, \ldots, (\sigma - \gamma_\alpha)\}$$

$\gamma_\alpha \equiv$ the minimum distance between any two elements of $\boldsymbol{S}$ which guarantees that the distance between the corresponding elements of $\boldsymbol{M}$ is at least α.

EXAMPLE

Let $\boldsymbol{S} = \ < s_1, s_2, s_3, s_4 > \ = \ < m_1, m_3, m_6, m_8 >$

$$
\begin{array}{lcccccccc}
\boldsymbol{S} = & s_1, & & s_2, & & & s_3, & & s_4 \\
 & \downarrow & & \downarrow & & & \downarrow & & \downarrow \\
\boldsymbol{M} = & m_1, & m_2, & m_3, & m_4, & m_5, & m_6, & m_7, & m_8
\end{array}
$$

If α = 2:

- A distance of 1 in ***S*** always yields a distance of at least 2 in ***M.***
- Since $k(g+1) - k(g) >= 2$, $\gamma_2 = (g+1) - g = 1$.

If α = 3:

- A distance of 1 in ***S*** can yield a distance of less than 3 in ***M***.
- A distance of 2 in ***S*** always yields a distance of 5 in ***M***.
- Since $k(g+2) - k(g) >= 3$, $\gamma_3 = (g+2) - g = 2$.

DEFINITIONS

$\boldsymbol{V}_i \equiv$ Multiset of values received by node i.

$\boldsymbol{U}_i \equiv$ Multiset of values received by node i from non faulty nodes.

$\boldsymbol{U}_{i\cap j} \equiv \boldsymbol{U}_i \cap \boldsymbol{U}_j$

$\delta(\boldsymbol{U}) \equiv \max(\boldsymbol{U}) - \min(\boldsymbol{U})$, for some multiset $\boldsymbol{U}$.

$$C \equiv \text{Convergence Rate} = \frac{\max\left|F(\boldsymbol{V}_i) - F(\boldsymbol{V}_j)\right|}{\delta(\boldsymbol{U}_{i\cap j})}, \quad \delta(\boldsymbol{U}_{i\cap j}) > 0$$

If $0 \le C < 1$, then the algorithm is convergent.

MSR PERFORMANCE FOR FCN

$$C = \frac{\gamma_a}{\sigma}$$

$$N \geq 3a + 2s + b + 1$$

$$V = N - b$$

$$\tau \geq a + s$$

SUMMARY

- Hybrid Fault Models were discussed with the 5-mode model being offering special value, as they consider the impact of omissive behavior on symmetric and asymmetric faults,
  - transmissive
  - omissive
- Approximate agreement algorithms were considered in the context of their convergence rates
  - allows to see how fast algorithms converge
  - some algorithms may work fast, but convergence is not guaranteed, e.g., simple median with a >0