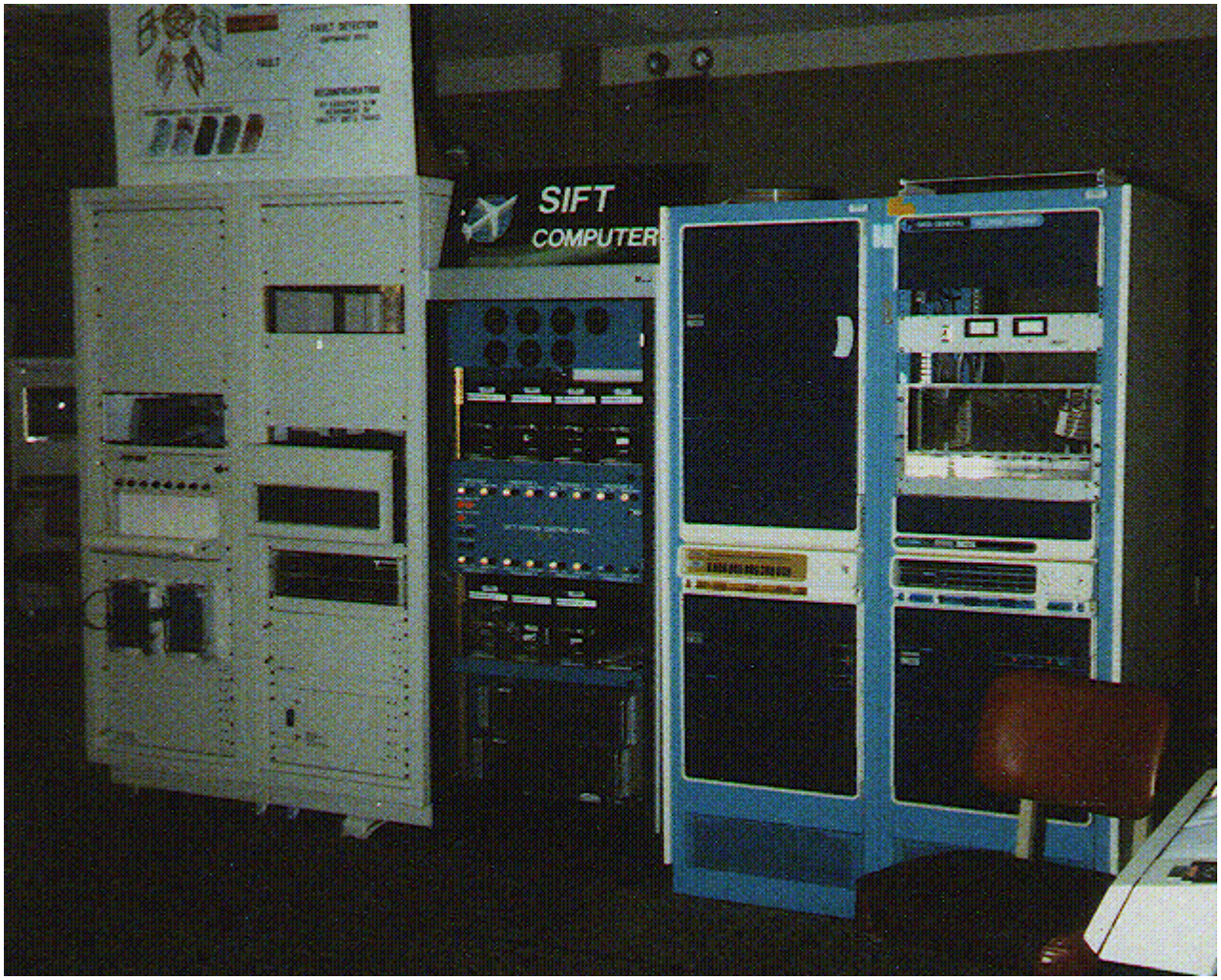
MAFT

- Background
  - SIFT
    - fault-tolerance is achieved under the control of executives
    - penalty: 70-80% overhead
  - FTMP
    - fault tolerant multiprocessor
    - provided hardware assistance for functions such as synchronization, voting and control functions
    - still 60% overhead of executive functions
  - Neither SIFT nor FTMP were designed to be true distributed systems => proof-of-concept systems

MAFT

"nostalgic" picture of SIFT

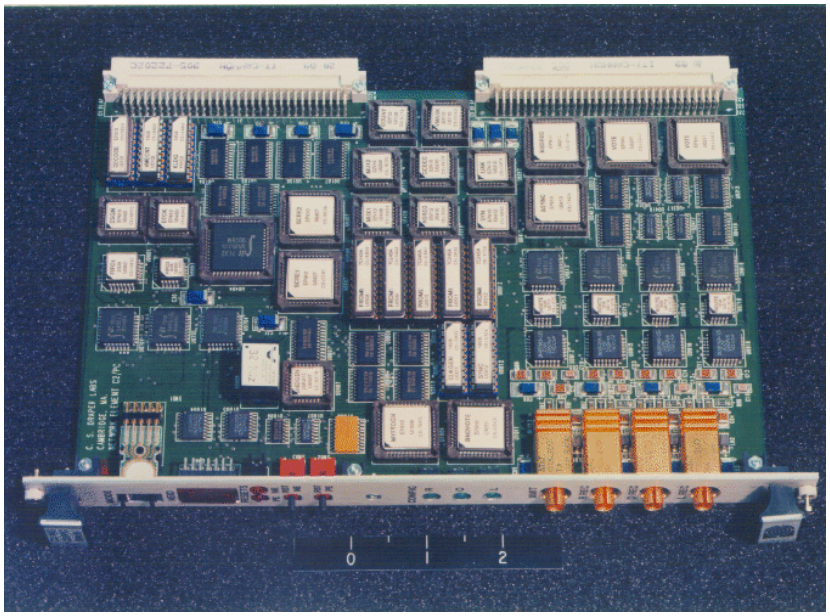


MAFT

- FTP
  - fault tolerant processor
  - was targeted towards enhancing system efficiency
  - FTP is a uniprocessor employing redundant processing channels
  - throughput of redundant system is equal to uniprocessor throughput
  - FTP was used in AIPS
- AIPS
  - Advanced Information Processing System
  - AIPS concept has been demonstrated using a dynamic simulation of the Blackhawk helicopter
  - The demonstration system consists of a quadruple redundant parallel processor, known as the AIPS/Army Fault Tolerant Architecture (AFTA), and a Silicon Graphics, Inc., workstation. The helicopter simulation executes on the workstation.
  - AIPS was developed by Draper Labs under a NASA contract.

MAFT

- FTPP
  - » fault tolerant parallel processor emphasized performance issues
  - » did not provide same level of Byz.-resilient operation as FTMP



MAFT

- MAFT (Kie88)
  - Multicomputer Architecture for Fault-Tolerance
  - Design objectives
    - reliability of 10^{-9} over 10 hours
    - minimum performance requirements
      - 200 Hz. Max Task Iteration Rate
      - 5.5 MIPS Max Computational Capacity
      - 1.0 MBPS Max I/O Transfer Rate
      - 5.0 ms Min Transport Lag (Input to Output)
  - Achieving reusability trough functional partitioning
    - Application Specific Functions
    - Standard Executive Functions

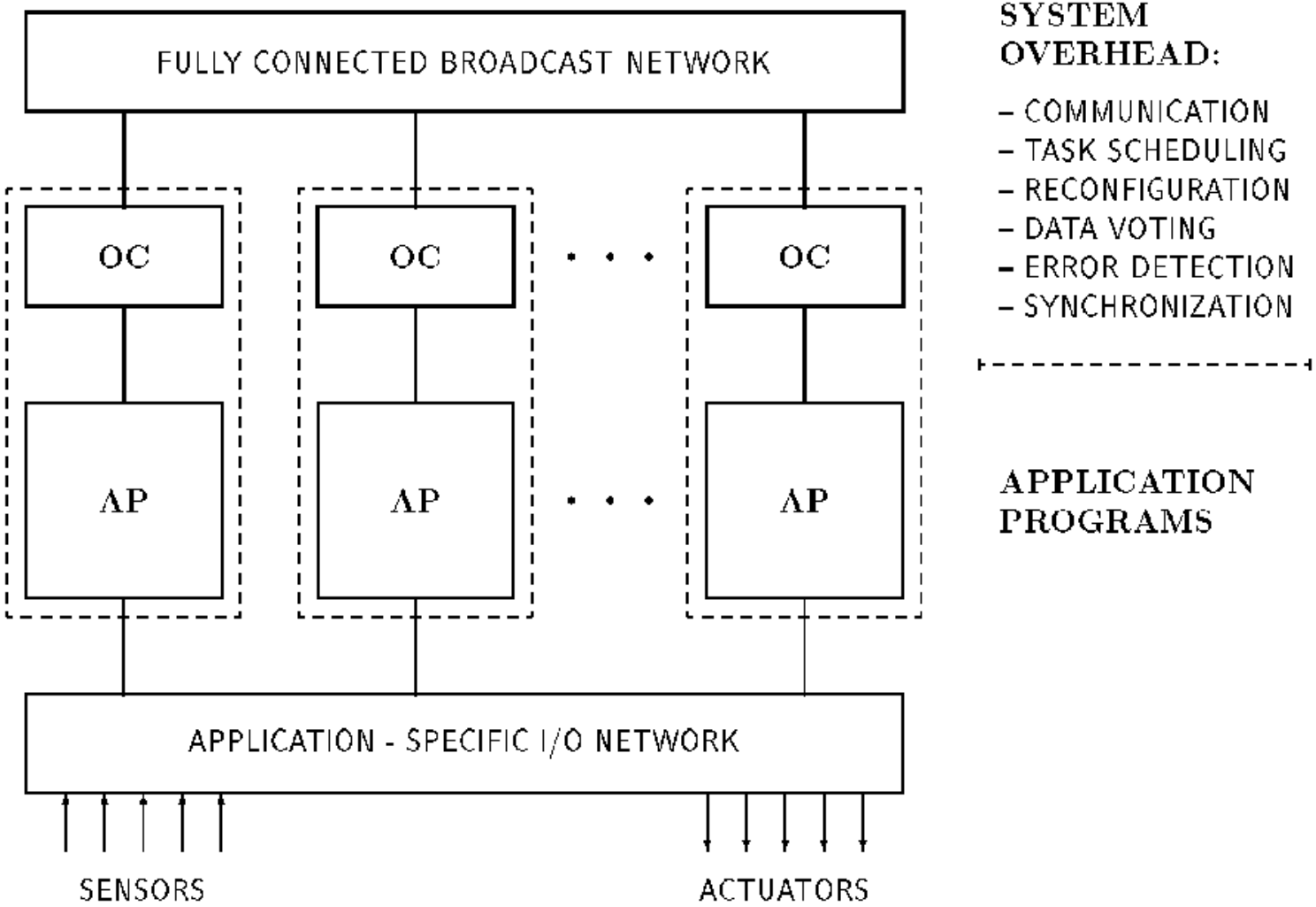
MAFT

- References

- [Dar88] Darwiche, A.A., and F.M. Doerenberg, Application of the Bendix/King Multicomputer Architecture for Fault-Tolerance in a Digital Fly-By-Wire Control System", Midcon, Aug 1988.
- [Glu86] Gluch, D.P., and M.J. Paul, Fault-Tolerance in Distributed Digital Fly-by-Wire Flight Control Systems", AIAA/IEEE Seventh Digital Avionics Systems Conference, Oct 1986.
- [Kie87] Kieckhafer, R.M., Task Reconfiguration in a Distributed Real-Time System," Eighth IEEE Real-Time Systems Symposium, Dec 1987.
- [Kie88] Kieckhafer, R.M., et al, The MAFT Architecture for Distributed Fault-Tolerance", IEEE Trans. Computers, V. C-37, No. 4, pp. 398-405, Apr 1988.
- [Kie89] Kieckhafer, R.M., Fault-Tolerant Real-Time Task Scheduling in the MAFT Distributed System," Proc, 22nd Hawaii International Conference on System Sciences, Jan 1989.
- [McE88] McElvany, M.C., Guaranteeing Deadlines in MAFT," Proc. IEEE Real-Time Systems Symp., pp. 130-139, Dec 1988.
- [Tha88] Thambidurai, P.M., and Y.K. Park, Interactive Consistency with Multiple Failure Modes", Proc. Seventh Reliable Dist Systems Symp., Oct 1988.
- [Tha88a] Thambidurai, P.M. Critical Issues in the Design of Distributed, Fault-Tolerant, Hard Real-Time Systems, Ph.D. Dissertation, Dept. of Electrical Engineering, Duke University, 1988.
- [Tha89] Thambidurai, P.M., et al., Clock Synchronization in MAFT", Nineteenth Fault-Tolerant Computing Symposium, pp. 142-151, Jun 1989.
- [Wal88] C.J. Walter, MAFT: An Architecture for Reliable Fly-by-Wire Flight Control," Proc. AIAA/IEEE Eighth Digital Avionics Systems Conference, pp. 415-421, Oct 1988.

MAFT

– System Architecture

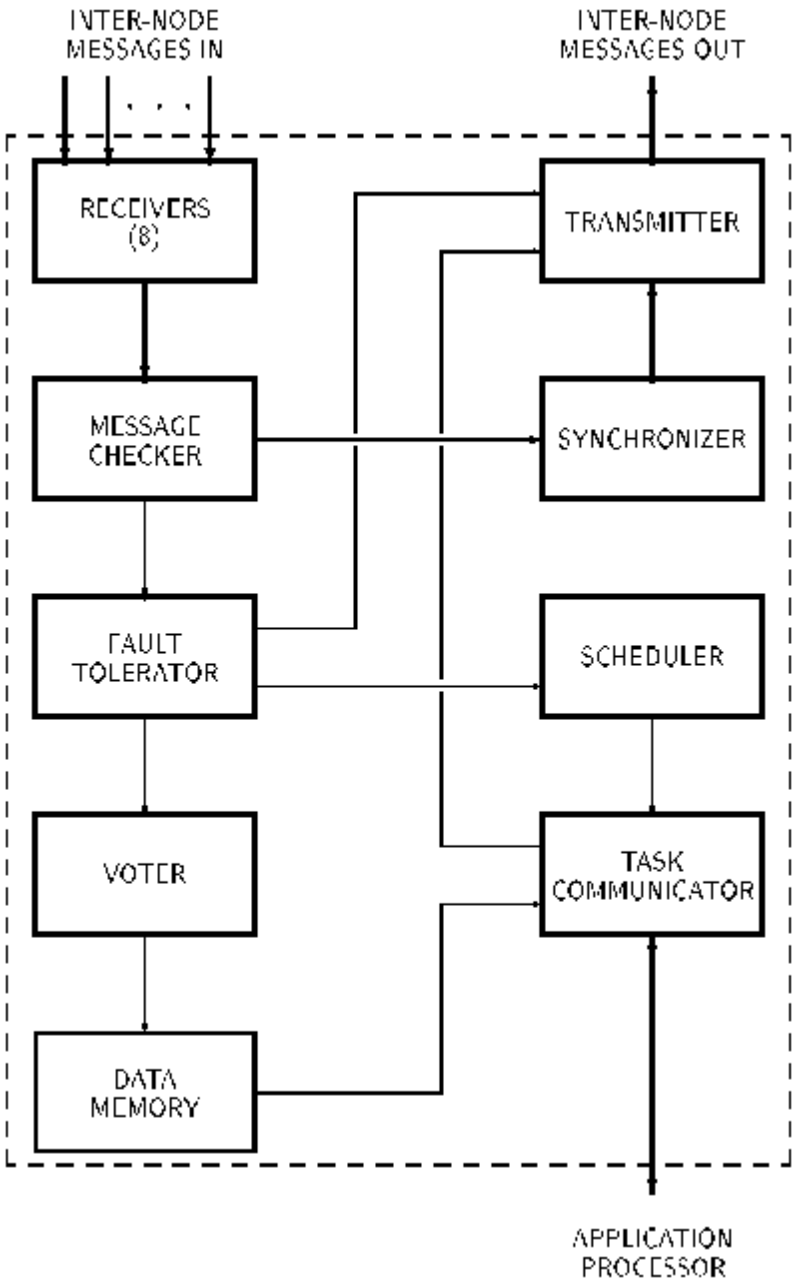


MAFT

- Application Processor (AP)
  - free to executes application program
    - e.g. reading sensors, performing control law computations, sending commands to actuators
  - flexibility to select processor suitable for the application
- Operations Controller (OC)
  - special purpose device common to all MAFT systems
  - performs overhead functions including
    - internode communication
    - synchronization
    - data voting
    - error detection
    - task scheduling
    - system reconfiguration
  - as a result the OS on application processor is extremely simple

MAFT

- Operation Controller block diagram
  - communication
    - with other OCs
    - with local AP



MAFT

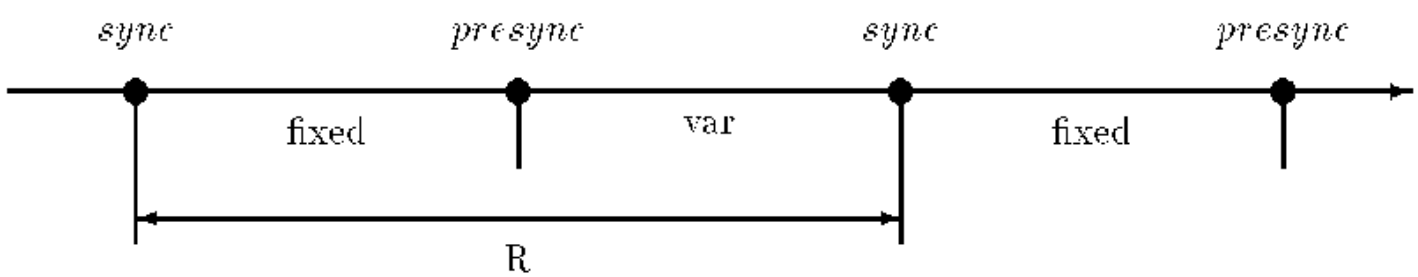
- Communication
  - transmitter
    - formats message, message type, framing, ECC
    - broadcasts message
  - receiver
    - one per incoming link
    - accepts properly framed bytes
    - buffer byte for message checker
  - message checker
    - poll receiver at a cycle rate of 6.4 μs
    - physical and logical checks
    - forward good messages to other subsystems
    - dump bad messages

MAFT

- synchronization
  - two major synchronization functions
    - steady-state operation
      - maintain synchronization in the *operating set*
      - similar to SIFT
      - loose frame based synchronization achieved using system state (SS) messages
      - accuracy depends on length of sync. interval, clock drift and message delivery delay. Given aircraft geometry, max skew is 18 μs.
    - startup
      - *cold start mode* for initialization of system or midmission event
      - *warm start mode* for synchronizing a node to an existing operating set

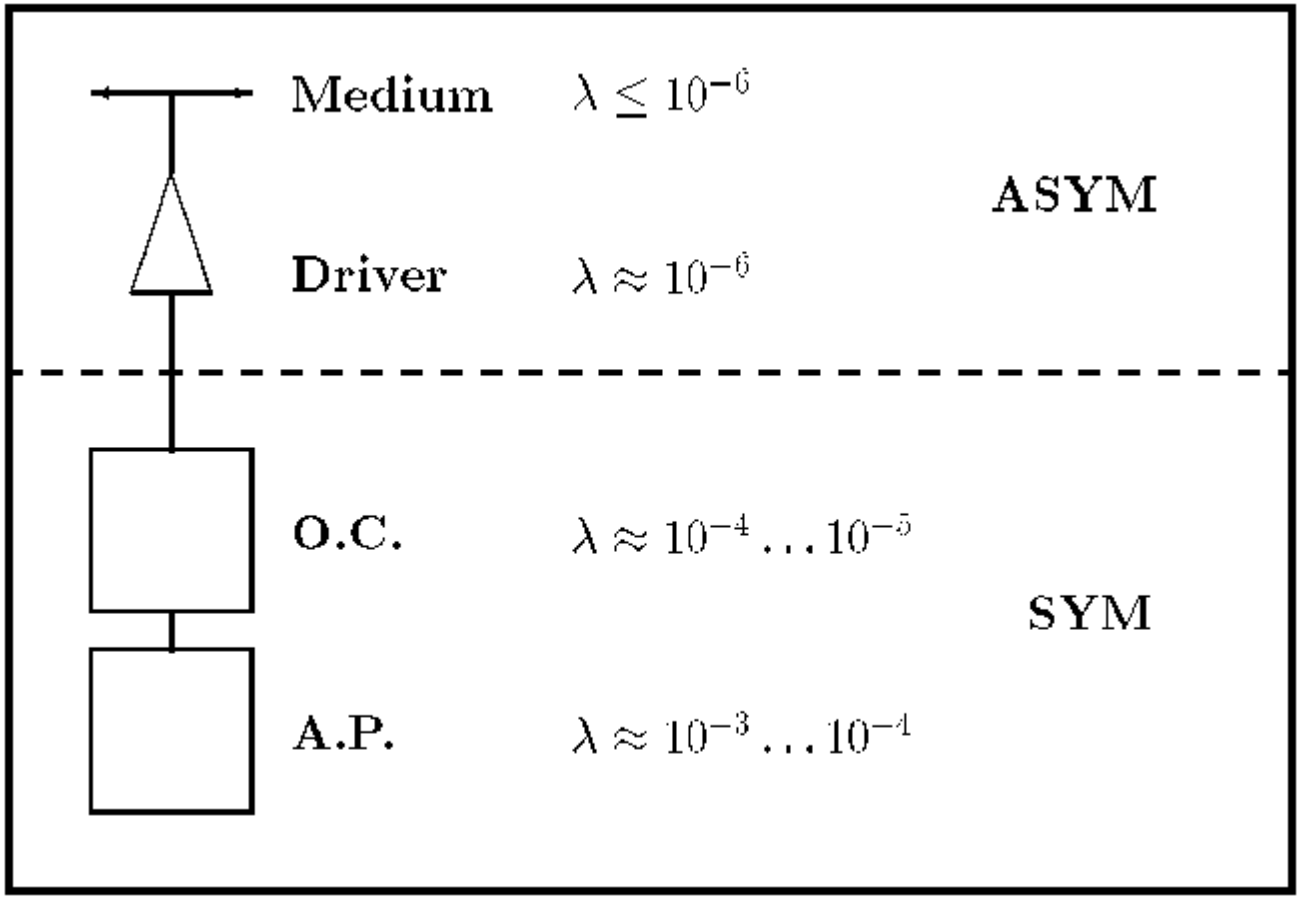
MAFT

- Steady-State Synchronization
  - each iteration has 2 phases:
    - phase 1 (fixed length)
      - count a fixed number of local clock ticks
      - broadcast "presync" SS (system state) message
    - phase 2 (variable length)
      - receive message during "presync window"
      - locally timestamp all received messages
      - vote: compute fault tolerant midpoint of timestamps
      - correction = midpoint - own timestamp
      - count to corrected "sync" value
      - issue "sync" SS message



MAFT

- Fault classes by source



MAFT

- Data Management
  - OCs store copy of all shared data in local memory
    => N-way redundant storage
  - data identification descriptor (DID) uniquely labels data
  - OC performs reasonableness checks to filter out data which is outside of a predefined interval.
  - size of deviance window is individually defined for each DID
- Voting
  - transparent to the AP
  - "on-the-fly" voting uses new copy and any previously received copies
    => voted value still available in the case of omissions

MAFT

- Two voting algorithms are available:
  - median select (MS)
    - select center value for odd number of inputs
    - average two center values for even number of inputs
  - mean of the medial extremes (MME)
    - discard the μ most extreme values from both sides of the sorted multiset of values
    - compute mean the two remaining extremal values
  - difference between MS and MME
    - computationally MS and MME vary only by respective μ

$$\mu_{MS} = \left\lfloor \frac{N-1}{2} \right\rfloor \qquad \mu_{MME} = \left\lfloor \frac{N-1}{3} \right\rfloor$$

  - MS is more robust than MME since it discards more values, but malicious faults can prevent convergence
  - MME is convergent for any fault model with convergence rate 1/2

- $c(T)$ = Value of "Real Time" when clock time T.

- ρ = Max peak-to-peak variation in clock rate.

$$\frac{\rho}{2} = \frac{dc(T)}{dT} - 1 \ \leq \ 5 \cdot 10^{-5}$$

- ϵ = Max delay between transmission and time-stamping of an SS message, including:

  - ϵ_p = Propagation through medium ($\leq 2\ \mu sec$).

  - ϵ_r = Msg. proc. in receiving node ($= 6.4\ \mu sec$).

- $\delta_{pq}(T) = |c_p(T) - c_q(T)|$

- $\delta(T) = \max_{\forall(p,q)}(\delta_{pq})$ at time T.

MAFT

- Typical Synchronization Values
  - $\epsilon = 7\ \mu sec$ – 600 ft. separation
  - $\rho = 5 \cdot 10^{-5}$
  - $R = 20\ msec \Rightarrow 10\ msec$ Atomic Pd. $\Rightarrow$ 100 Hz.
  - $\rho R = 1\ \mu sec$
  - No Faults: Max $\delta = 8.5\mu\ sec$
  - With Faults: Max $\delta = 16.5\mu\ sec$

MAFT

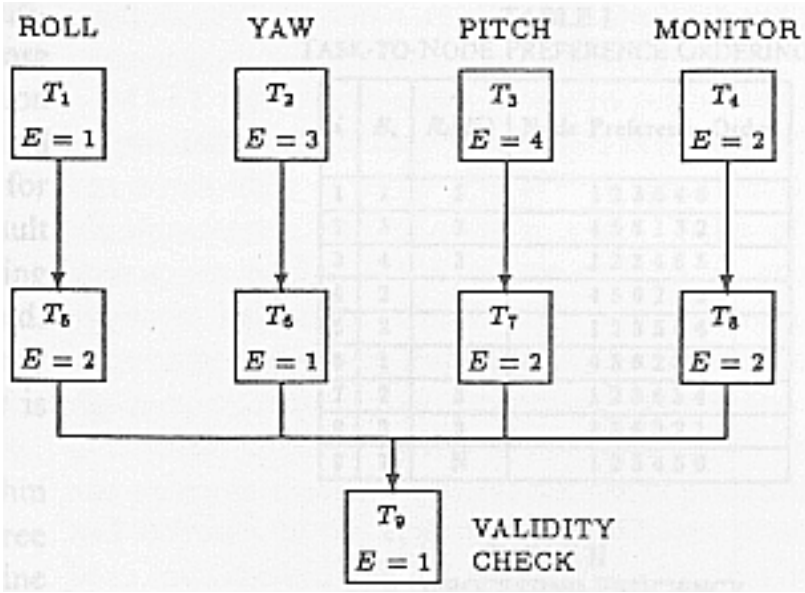
- Scheduling
  - tasks on a single AP are non-preemptive
  - task properties
    - iteration frequency
    - relative priority
    - desired redundancy
    - intertask dependencies
    - dependencies also include AND/OR-FORKS and AND/OR-JOINS
  - shortest iteration period *atomic period*
  - boundaries of atomic periods coincide with an SS (system state) message
  - no task can execute more than once per atomic period
  - however, several tasks may execute during one atomic period
  - low-frequency tasks may run for several atomic periods

MAFT

- longest period is called *master period*
- 1024 atomic periods = 1 master period
- scheduling algorithm is based on non-preemptive static priority list scheduling
- task-to-processor allocation is determined by task reconfiguration process and is static for any given operating set
- scheduler is fully replicated, i.e. the scheduler of each node selects tasks for each node in the system
  - local scheduler determines task for own processor
  - all other nodes monitor the actions of the other schedulers
- to maintain Byzantine agreement on scheduling MAFT implements one round of rebroadcast => 1 asym. fault
- synchronization of rebroadcasts is accomplished by defining *subatomic periods* via task interactive consistency TIC messages

MAFT

» scheduling example



PERIOD	1	2	3	4	5	6	7	8	9	10	11	12
GROUP 1	1	3				5		7		ϕ		9
GROUP 2	2			4		6	ϕ	8		ϕ		9