Design Examples of Distributed Applications Based on the Use of Synchronized Clocks

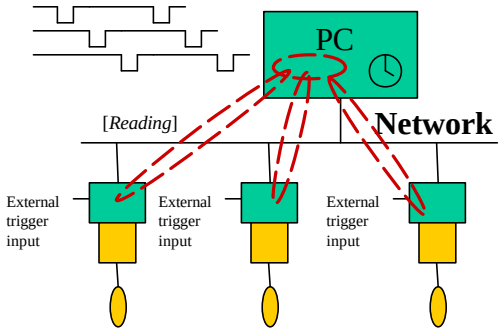
Randy Coverstone and John C. Eidson
Agilent Laboratories

Ptolemy Conference, Berkeley,
March 22-23, 2001

Agenda

- Measurement and control based on explicit time
- Techniques for synchronizing clocks
- Example systems based on synchronized clocks
- Simulation of distributed, synchronized clock based systems

Centralized Systems



Scalability of such a system is poor: as the number of channels is increased,

- Processing load on the controller increases,
- Complexity of the control program increases,
- Maximum channel sampling rate decreases,
- Ability to sample channels "simultaneously" degrades, and
- The worst-case network load increases.

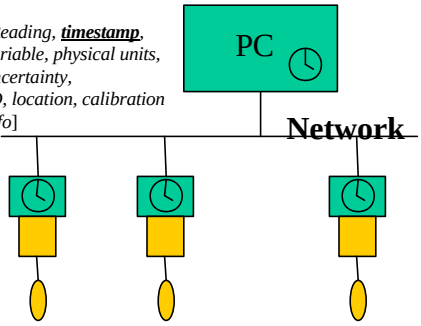
- Master/slave protocol
- Usually message based design
- Slaves are sequentially polled in a loop.

BUT: We are familiar with this style!

Agilent Technologies

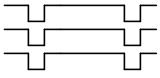
Distributed Systems With Clocks

[*Reading, **timestamp**, variable, physical units, uncertainty, ID, location, calibration info*]



Improved scalability:

- Complexity of the control program remains the same,
- Maximum rate at which each channel can be sampled depends on each node, not the controller,
- The minimum time to sample all the channels is reduced to the amount of time to sample one channel.
- To add more channels you simply add more nodes,
- Simultaneous acquisition is possible.



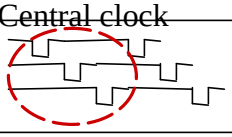
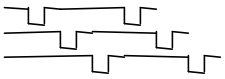
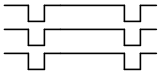
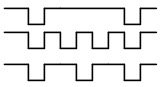
- Master/slave protocol permitted, but not required.
- Increased flexibility in sampling schedule.
- Network messages may now contain accurate time-stamps generated at the point of sensing/actuation

Agilent Technologies

Use of Explicit Time in Measurement and Control

- Basic uses of time (relative or absolute)
  - Time stamping system events for debugging
  - Time stamping data to allow correlation with other data or events
- Coordination of measurement: (sampling and triggering)
- Coordination of action (time based behaviors)
  - Timed execution scripts
  - Temporal mutual exclusion: (time slots)

Sampling/triggering Options

	Central clock	With distributed clocks
Sequential		
Simultaneous	Requires simultaneous sample and hold or ext trigger wires.	
Different rates for each node	Requires oversampling and decimation.	

Timed Execution Script

network message: start_experiment(time stamp = a)

```
Script_1:                    Script_2:
When(t= =a){                 Sample(100khz,now);
Initialize_node( );          Start_circular_buffer( );
Sample(10khz,a);}            When(t= =a+0.001){
When(t= =a+0.0005){          Freeze_circular_buffer( );
Setpoint_1 = 27;}            Send_data(a-
                               0.001,a+0.001);}
```

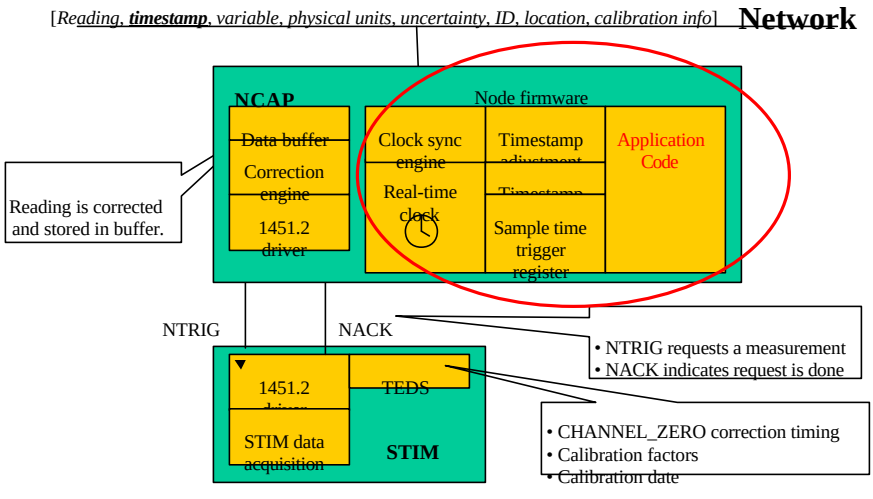
A start into the literature

- Atul Adya, Robert Gruber, Barbara Liskov, Umesh Maheshwari: Efficient Optimistic Concurrency Control Using Loosely Synchronized Clocks. SIGMOD Conference 1995: 23-34
- **Barbara Liskov: Practical Uses of Synchronized Clocks in Distributed Systems. Distributed Computing 6(4): 211-219 (1993)**
- Barbara Liskov, Liuba Shrira, John Wroclawski: Efficient At-Most-Once Messages Based on Synchronized Clocks. TOCS 9(2):125-142 (1991)
- **Real-Time Systems: Design Principles for Distributed Embedded Applications, Hermann Kopetz, Kluwer Academic Publishers, 1997.**

Use of explicit time representation requires:

- A real time clock and associated application devices in each distributed device
- The local time of all clocks to agree to within some system defined uncertainty
- Clock time is monotonic and ‘continuous
- The process synchronizing the clocks to be a provided system service

Node Block Diagram



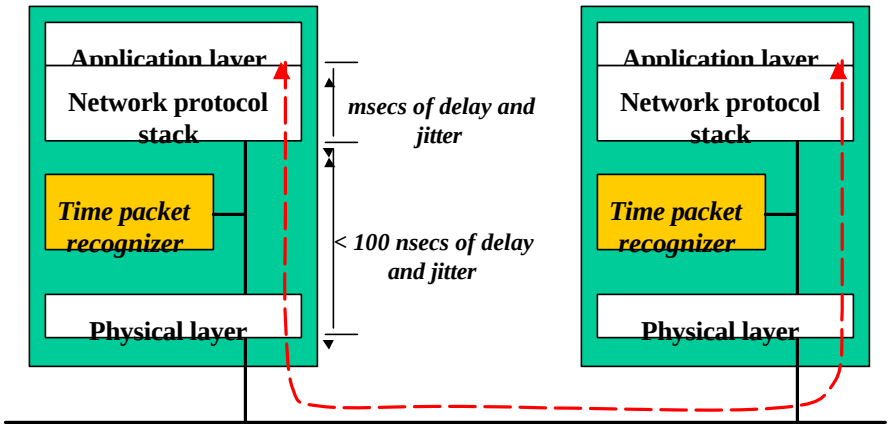
Degrees of Time Synchronization

- Day
- Hour
- Minute
- Second
- 10 msec
- millisecond
- microsecond
- < microsecond

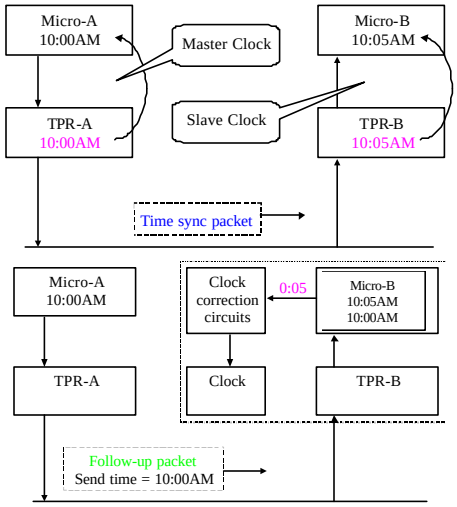
No hardware assist required
e.g. Network Time Protocol (NTP)

Requires hardware assist
e.g. Agilent PTP, TTP/C, GPS,IRIG

Removing jitter



PTP Time Synchronization



Step 1:

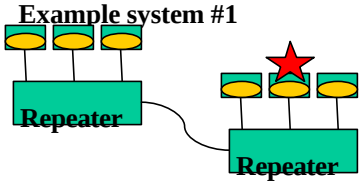
- Time sync packet from master to slaves
- All TPR's note the time this packet appears

- Step 2:

- Follow-up packet from master to slaves
- All slaves compute offset and correct the slave clock

Selecting best master clock

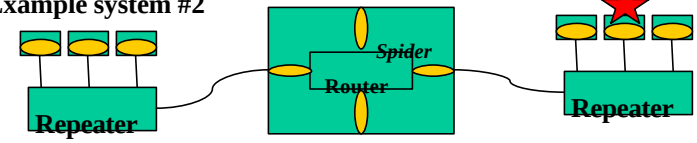
Example system #1



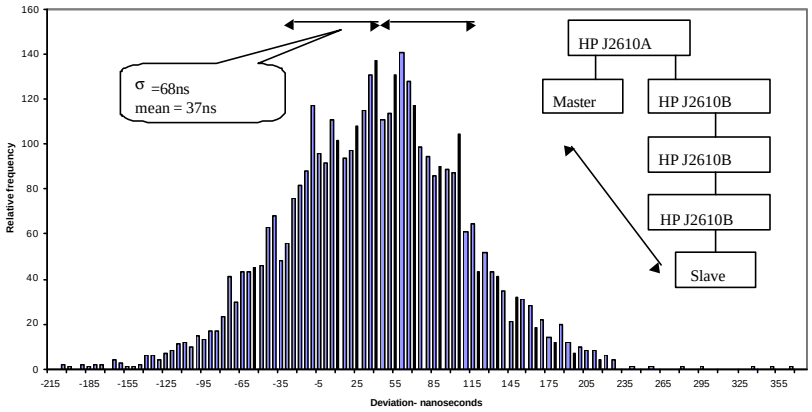
Selection criteria

- **Tracability to UTC**
- **Precision/jitter**
- **Path length**
- **Preferred status**

Example system #2



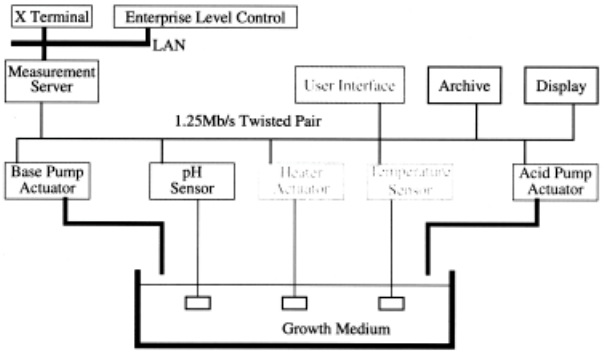
Seconds tick deviations between the clocks of two system instruments



Agilent Technologies

4/3/2001 15

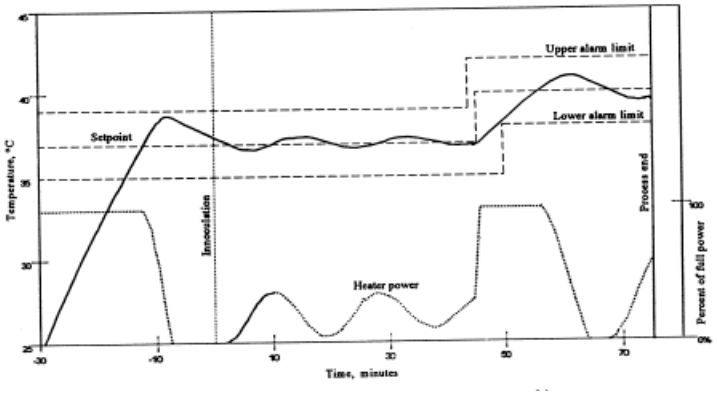
Example: Fermentation Control



Agilent Technologies

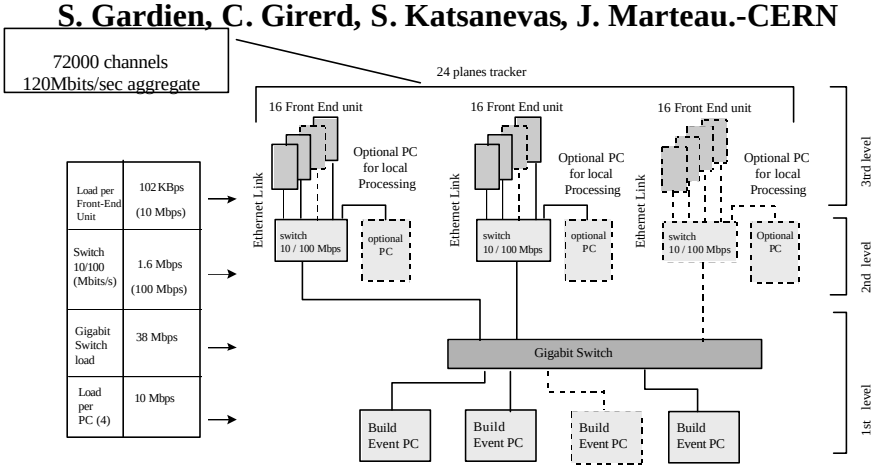
4/3/2001 16

Fermentation Control



Ethernet Network-based DAQ and smart sensors for the neutrino long base line OPERA experiment.

S. Gardien, C. Girerd, S. Katsanevas, J. Marteau.-CERN



CERN-front end

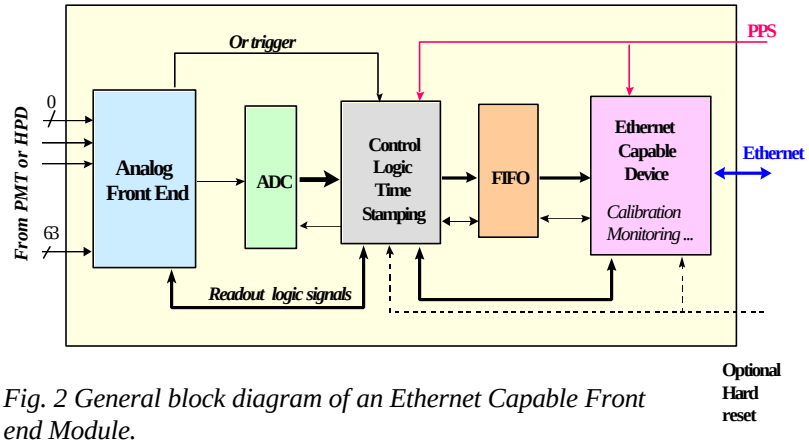
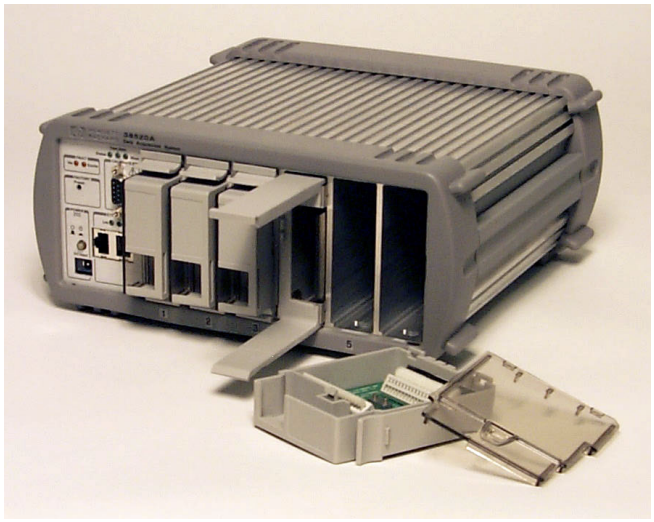
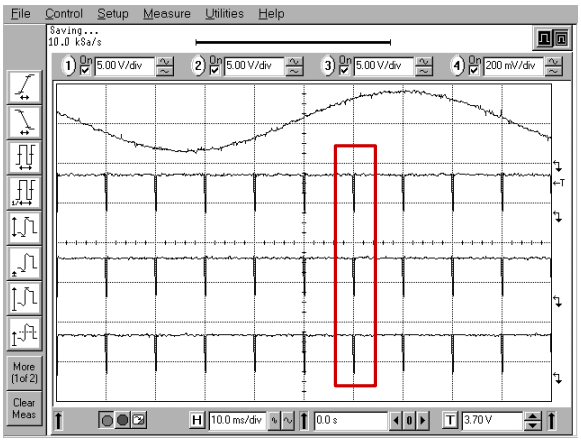


Fig. 2 General block diagram of an Ethernet Capable Front end Module.

Example: Data Acquisition



Sample Triggers From Several Instruments



100 triggers/sec
10 msec/division

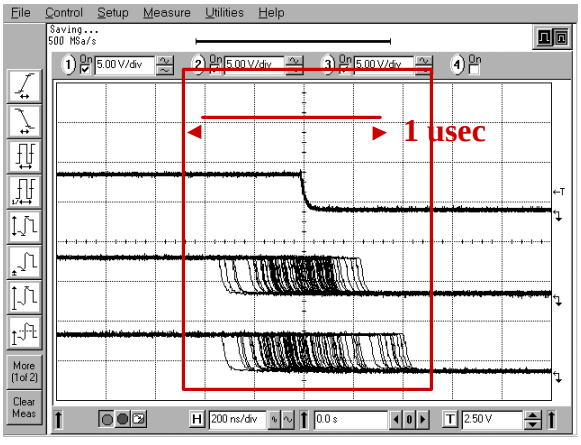
12 Hz sine wave

Clock master

Clock slave

Clock slave

Sample Triggers From Several Instruments



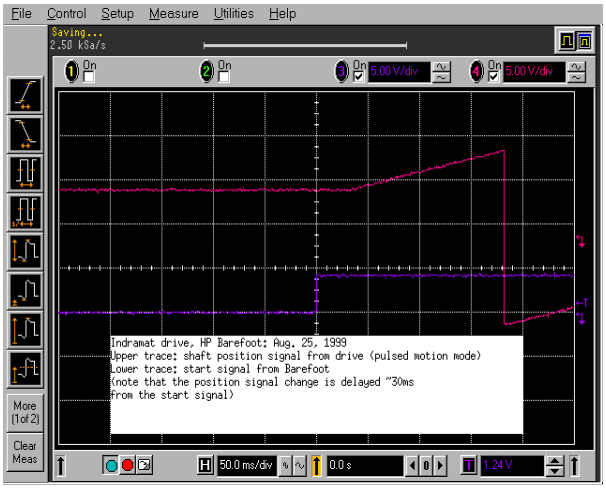
0.48 triggers/sec
200 nsec/division

Clock master

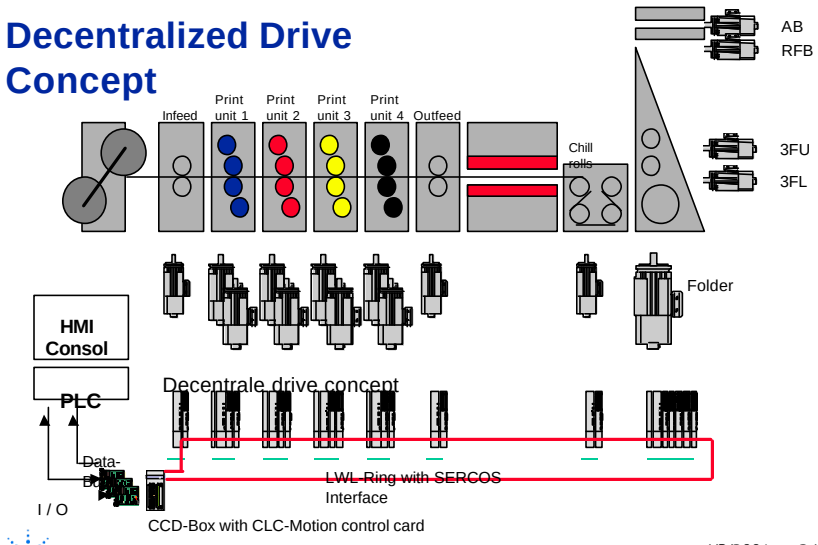
Clock slave

Clock slave

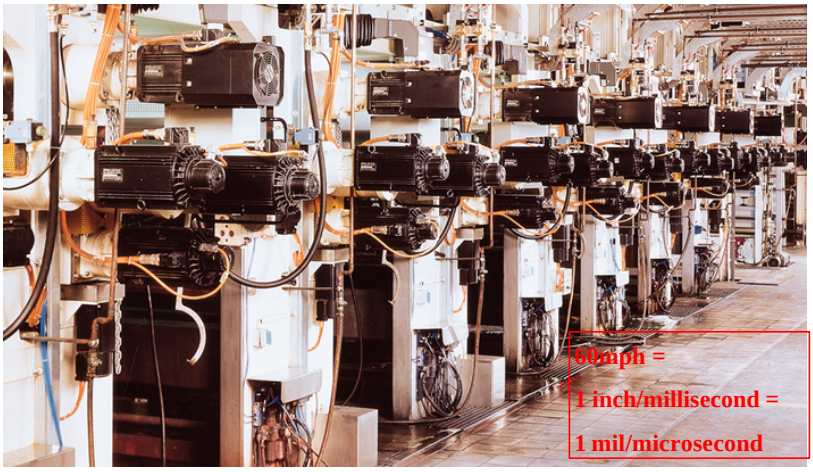
Example: Motor Control
courtesy Indramat



Commercial printing press
(slide courtesy of Indramat)



Flexo Printing Press(Unit Construction)
(slide courtesy of Indramat)



Simulation of Time Based Systems