Sistemas de Tempo Real:

Sincronização de Relógios

Rômulo Silva de Oliveira
Departamento de Automação e Sistemas – DAS – UFSC

romulo.deoliveira@ufsc.br
http://www.das.ufsc.br/~romulo

Setembro/2013

References

- *Cena, G.; Bertolotti, I.C.; Scanzio, S.; Valenzano, A.; Zunino, C., "Synchronize your watches: Part I: General-Purpose Solutions for Distributed Real-Time Control," Industrial Electronics Magazine, IEEE, vol.7, no.1, pp.18,29, March 2013*
  - doi: 10.1109/MIE.2012.2232354
- *Cena, G.; Bertolotti, I.C.; Scanzio, S.; Valenzano, A.; Zunino, C., "Synchronize Your Watches: Part II: Special-Purpose Solutions for Distributed Real-Time Control," Industrial Electronics Magazine, IEEE, vol.7, no.2, pp.27,39, June 2013*
  - doi: 10.1109/MIE.2013.2248431

Summary

Introduction 1/6

- *The precise time of an event is very important*
  - Specially for automated control applications
- *It includes*
  - The knowledge of the exact time of occurrence of a certain event
  - The ability to enforce a given action at a precise instant
- *Example: The execution of transactions in enterprise resource planning (ERP)*
  - Seconds
- *Example: The coordination of movements in multi-axis, numerically controlled machines*
  - Fractions of microseconds

Introduction 2/6

- *Modern distributed control applications:*
  - Many cooperating processes
  - Running on physically disjointed machines
  - Interacting by exchanging information over communication networks
- *Important to share a common view of the elapsing time among the different devices*
- *This concept is often referred to as global time*
  - Also known as *system time* in several protocol specifications

Introduction 3/6

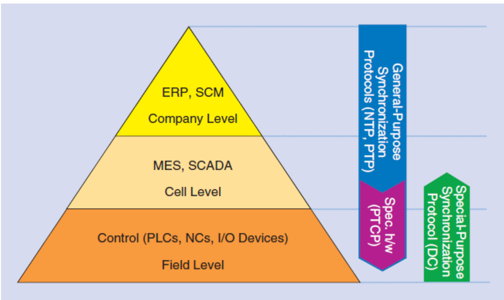
- *clock synchronization protocol*
  - Provides the alignment of device local clocks to a single, system-wide time
  - by the exchange of messages over the network
- *Alternatives:*
  - The time reference provided by global positioning system (GPS)
  - A time code broadcast by a ground station on a longwave radio channel
    - DCF77 station in Europe and WWVB in the United States
- *But there is*
  - Cost
  - Antenna positioning
  - Downtime
  - Time evolution between radio/GPS signals

Introduction 4/6

- *Synchronization protocols for industrial and networked embedded control applications can be broadly divided in two classes*
- *General-purpose solutions*
  - They are not defined specifically for factory automation or distributed control applications
  - Also used in other contexts (e.g., office automation)
  - The network time protocol (NTP) and the precision time protocol (PTP)
- *Special-purpose solutions*
  - They are conceived explicitly for use in real-time control systems
  - Industrial, automotive, embedded, etc.
  - EtherCAT (factory automation), the synchronization of FlexRay (automotive)

Introduction 5/6

- *Synchronization at different levels of the automation pyramid*



Introduction 6/6

- *Goal: to keep the local clocks in a number of networked devices aligned to the same system time*
- *There are 3 main steps*
- *<u>Deviations of local clocks</u> are estimated with respect to the system time*
- *Suitable <u>compensation values</u> are computed using the acquired samples*
- *<u>Local oscillators</u> are corrected*

Measurement of Clock Deviations 1/5

- *Measurement of Clock Deviations*
- *Time stamp:*
  - A time reference that identifies when a certain event occurred
- *The evaluation of clock deviations relies on time stamps that are taken by two or more nodes on "common events"*
- *The transmission and reception of a specific synchronization message, for instance, may constitute such a kind of reference*
- *How messages, on which time stamps are taken, are generated and managed ?*

Measurement of Clock Deviations 2/5

- *How messages, on which time stamps are taken, are generated and managed ?*
- *The master–slave approach*
- *A special node (master) holds the reference time*
  - It sends synchronization messages cyclically
  - to the other devices (*slaves*)

Measurement of Clock Deviations 3/5

- *How messages, on which time stamps are taken, are generated and managed ?*
- *The client–server approach*
- *Every node (client) asks explicitly for synchronization messages*
  - to the special node (*server*)
  - which is in charge of holding the reference time

Measurement of Clock Deviations 4/5

- *How messages, on which time stamps are taken, are generated and managed ?*
- *The distributed approach*
- *Every node broadcasts synchronization messages over the network*
  - which are captured by all the other nodes
  - according to a *producer–consumer* approach
  - Each one derives a common “average” reference time

Measurement of Clock Deviations 5/5

- *The achievable accuracy of time stamps is influenced by many sources of uncertainty*
- *The time of the reference node, to which other nodes must synchronize, may itself be affected by errors due to either its ability to synchronize with an external primary time source*
- *Jitters and propagation delays experienced by packets traveling across the network also may affect the accuracy*
- *The precision of time stamps taken by the different devices is affected by several sources of in-node uncertainty*
  - the network adapter
  - the internal architecture (system buses, northbridge, southbridge, etc.)
  - the operating system

Compensation of Clock Deviations 1/3

- *Clock compensation can be performed through two different kinds of actions*
- *Rate correction: The actual frequency of local oscillators is affected by tolerances and environmental parameters (temperature)*
  - the *rate* of the local clock adjusted accordingly
- *Offset correction: Even in the ideal case of oscillators with exactly the same frequency, misalignmentsmay still occur because devices are not bootstrapped in a synchronous way*
  - The *offset* of every local clock must be corrected

Compensation of Clock Deviations 2/3

- *Popular clock synchronization protocols usually rely on a mix of both rate and offset corrections*
- *To ensure:*
  - Quick convergence times
  - Long-term stability
  - Monotonic behavior
  - While avoiding time warps

Compensation of Clock Deviations 3/3

- *Oscillator frequencies in real clocks may deviate for a number of reasons*
  - They contribute to the drift between the local and the reference clock
- *Production tolerances in quartzes*
  - cause the actual and nominal frequencies to be different
- *Temperature*
- *Components aging*
- *Power supply instability*
- *Mechanical vibrations*
- *Air pressure*
- *There is a substantial stochastic component in the oscillator behavior that cannot be compensated once for all*
  - They have to be corrected through continuous and adaptive mechanisms

Network Latencies 1/4

- *The synchronization of nodes over a network must also take into proper account transmission latencies*
- *Every message exchanged is affected by both*
  - propagation delays over communication media (which depend on the physical distance between nodes)
  - and passthrough delays in network equipment (which are related to the underlying technology)
- *Including messages used to measure deviations between clocks*

Network Latencies 2/4

- *Synchronization protocols provide the means for compensating transmission latencies*
- *In small networks, the overall transmission latency of messages is short and mostly deterministic*
- *In large geographic networks, it can grow noticeably*
  - might be affected by nonnegligible jitters
- *Jitters on the reception of synchronization messages cause unpredictability and worsen the overall precision*

Network Latencies 3/4

- *Propagation delays can be either determined*
- *Right in the design phase, by considering the network topology*
  - Vehicles
  - Ensures the highest degree of reliability
- *Evaluated at run time, through time measurements carried out on the exchanged messages*
  - Controller area network

Network Latencies 4/4

- *Latencies introduced by the network controller and protocol stack in each device affect the quality of synchronization*
- *These delay contributions can often be neglected over a WAN*
- *Importante in small control networks*
  - Hardware mechanisms may be needed to achieve the required quality of synchronization
  - Implementation is more expensive

Summary

- *Introduction*
- *NTP – Network Time Protocol*
- *PTP – Precision Time Protocol*
- *Flexray Clock Synchronization*
- *Final Remarks*

NTP – Network Time Protocol

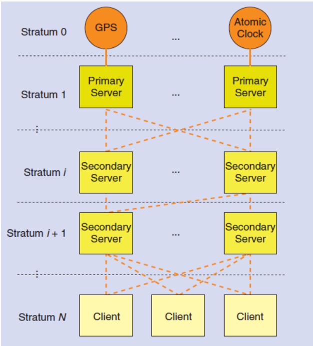
- *The most recent (fourth) version (RFC 5905) was released in June 2010 by the Internet Engineering Task Force (IETF)*
- *Also includes a lightweight variant of NTP known as Simple Network Time Protocol (SNTP)*
- *NTP was mainly designed for WANs*
- *It can be adopted in LAN as well*
- *NTP can be a satisfactory solution for industrial environments*
  - Both at the company and cell/field levels
  - Achieve inexpensive synchronization
  - When timing requirements are not so demanding

NTP – Communication Infrastructure 1/5

- *NTP can be used in every network supporting the UDP/IP protocol suite.*
- *From a logical point of view, NTP nodes are organized as a hierarchy of levels*
- *Levels (strata) are labeled with progressively higher numbers*
  - from 0 (absolute reference clocks) to 16
  - in the order of decreasing accuracy
- *Generally, nodes can then be subdivided into*
  - primary servers
  - Secondary servers
  - clients

NTP – Communication Infrastructure 2/5

- *The typical components of an NTP network*



NTP – Communication Infrastructure 3/5

- *Primary (stratum 1) servers are directly synchronized with absolute (stratum 0) reference clocks*
  - without communicating through the network
- *Devices such as atomic clocks or GPS receivers can act as stratum 0 references*

NTP – Communication Infrastructure 4/5

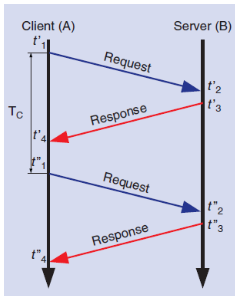
- *Secondary servers are nodes placed in the middle of the NTP hierarchy*
- *They synchronize over the network to the nodes belonging to the upper layer(s)*
- *and act as servers for nodes belonging to the layer(s) below*

NTP – Communication Infrastructure 5/5

- *Clients are the nodes in the bottom stratum of the layered architecture*
- *they are not able to provide a time reference for other entities in the network*

NTP – Synchronization Mechanism 1/4

- *Clients (and secondary servers) send cyclic request messages to nodes in the upper stratum with period $Tc = 2\tau$*
- *The value of τ depends on the difference between the reference and system times, and changes as time elapses*
- *The invoked server replies with a response message as soon as possible*



NTP – Synchronization Mechanism 2/4

- *The protocol is based on four time stamps: t1, t2, t3 and t4*
  - They are handled by the client for every server it is connected to
- *t1 is recorded by the transmitting node before sending the request*
- *t4 is recorded after receiving the response message*
- *In turn, t 2 and t 3 are obtained from the replying server*
  - which determines t 2 as soon as the request is received and
  - t3 just before sending its reply

NTP – Synchronization Mechanism 3/4

- *Time stamps are used to compute the offset and the round-trip delay according to the following equations:*

$$\theta = \frac{(t_2 - t_1) + (t_3 - t_4)}{2},$$
$$\delta = (t_4 - t_1) - (t_3 - t_2).$$

NTP – Synchronization Mechanism 4/4

- *A specific clock filter algorithm is then applied to the eight most recently estimated offsets in order to select the most accurate value*
- *The round-trip delay*
  - depends on the network distance between the client and server nodes
  - is one of the values needed
  - for the proper behavior of the clock filter algorithm
- *Each client can be synchronized with one or more servers*
  - Outliers are discarded
  - The final offset is a weighted average
  - The resulting value is passed to the clock discipline algorithm

NTP – Performance

- *LAN: NTP can reach a synchronization accuracy in the order of a few hundred microseconds*
- *WAN: accuracy can range from few tens to hundreds milliseconds*
- *Since each NTP node can be synchronized with several servers at the same time, the protocol is natively fault tolerant*
- *The unavailability of one server only causes some degradation of the client performance*
- *Client failures only affect the interested node*

Summary

- *Introduction*
- *NTP – Network Time Protocol*
- *PTP – Precision Time Protocol*
- *Flexray Clock Synchronization*
- *Final Remarks*

PTP – Precision Time Protocol

- *The current version of the IEEE 1588 synchronization protocol (PTP) was published by IEEE in 2008*
- *PTP is a master/slave protocol that can be used over LANs supporting multicast transmissions*
  - Ethernet, DeviceNet, ControlNet
- *Several implementations available*
- *PTP is widely used as the preferred synchronization mechanism in several real-time Ethernet solutions*
- *A popular open-source software implementation of IEEE 1588-2002 and IEEE 1588-2008 is PTP daemon (PTPd)*

PTP – Communication Infrastructure 1/8

- *In current real-world industrial applications*
- *PTP typically relies on a communication infrastructure based on switched Ethernet*
  - end nodes (PCs, workstations, I/O devices, etc.)
  - network equipment (switches)
- *Five types of nodes (clocks) are defined in PTP:*
  - Ordinary clocks
  - Boundary clocks
  - End-to-end nodes
  - Peer-to-peer nodes
  - Management nodes

PTP – Communication Infrastructure 2/8

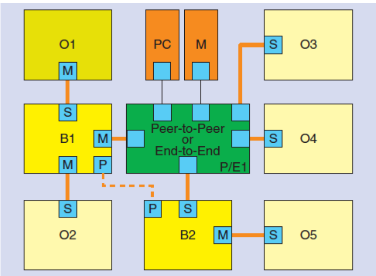
- *Ordinary clocks are nodes with a single network interface (port)*
- *Boundary clocks have two or more ports and are used to build tree-shaped topologies*
- *End-to-end nodes are transparent network elements*
  - They cannot act as masters or even as slaves
  - they are mainly PTP switches that compensate the time spent by packets inside the device by modifying the correctionField in the PTP messages they are relaying
  - This helps prevent nondeterminism introduced by intermediate network elements

PTP – Communication Infrastructure 3/8

- *Peer-to-peer nodes are mostly the same as end-to-end ones*
  - but they are able to compensate propagation delays over network links by using a suitable peer delay mechanism
- *Management nodes are used for administrative purposes*

PTP – Communication Infrastructure 4/8

- *an ordinary clock (O1) is the grandmaster clock*
  - the most precise clock in the network
  - the reference clock at the root of the tree topology



PTP – Communication Infrastructure 5/8

- *B1 is a boundary clock and a slave of O1*
- *B1 is the master for O2, O3, O4, and B2 at the same time.*
- *P/E1 is a peer-to-peer or an end-to-end clock*
  - It is transparent and performs only timing compensations
- *the boundary clock (B2) is a slave of B1 and the master for O5*
- *The dashed line represents a path that is pruned by the underlying protocols in order to obtain a tree topology*
- *A PC and a management node (M) are also connected to the network*

PTP – Communication Infrastructure 6/8

- *PTP selects the grandmaster clock and the state of ports*
  - by means of the best master clock (BMC) algorithm
- *Each port relies on "announce messages"*
  and runs algorithm BMC
  in order to choose the grandmaster and set its own state
- *Nine different port states are possible*
- *The master–slave hierarchy is determined by three of them only*
  - master, slave, and passive
  - A passive port is neither a master nor it synchronizes with a master

PTP – Communication Infrastructure 7/8

- *BMC works by comparing the de scriptions of clocks included in the announce messages*
- *Each clock is associated to a set of attributes*
  - Static: the clock class or the type of time source
  - Dyamic: the jitter of the clock
- *All clock attributes are ranked and compared by BMC using a predefined priority order*
- *Results of the comparisons are used to determine the recommended new state*
- *The BMC algorithm is executed for all ports using the same set of data*
  - The final computed state is the same for all clocks in the network

PTP – Communication Infrastructure 8/8

- *PTP frames exchanged between clocks:*
  - Event
  - General messages
- *A pair of time stamps are recorded before sending and after receiving any event message*
  - Sync
  - Delay_Req
  - Pdelay_Req
  - Pdelay_Resp
- *General messages are used for management functions*
  - transferring timing information (time stamps)
  - reading/writing data sets
  - communicating between nodes
  - Establishing the synchronization hierarchy

PTP – Synchronization Mechanism 1/15

- *PTP deals with four different types of errors that may affect the synchronization between clocks*
  - (1) Asymmetry
  - (2) Residence time
  - (3) Path delay
  - (4) Offset

PTP – Synchronization Mechanism 2/15

- *(1) Assymmetry*
- *Time t_{A2B} is the time to transfer a packet along the path between two nodes A and B*
- *The basic assumption in PTP is that time t_{A2B} is equal to time t_{B2A}*
- *In real situations they are not perfectly equal*
- *The protocol allows for the compensation of the path asymmetry through the correction field mechanism*
- *The **correction field** is inserted in the header of every timing message*
  - can be used by the receiver clock to modify the time-stamp values
- *PTP does not provide any indication about the way asymmetries can be evaluated*

PTP – Synchronization Mechanism 3/15

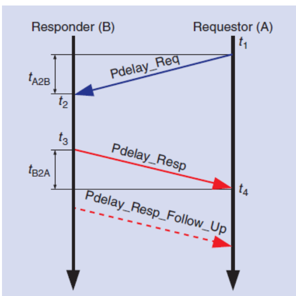
- *(2) Residence Time*
- *Time spent by an event message in traversing an end-to-end or peer-to-peer network element*
- *The update of the **correction field** is based on the difference between time stamps recorded when the message enters and leaves the traversed node*

PTP – Synchronization Mechanism 4/14

- *(3) Path Delay*
- *The propagation delay of the network link*
- *Path delays can be evaluated in two ways*
  - by using the delay request-response
  - the peer delay mechanism
- *Ports of a peer-to-peer node are requested to adopt the peer alternative*
  - Based on updates of the **correction field**,
- *Other types of nodes can choose beetween two methods*
- *Path-delay estimation does not apply to end-to-end nodes*

PTP – Synchronization Mechanism 5/14

- *The **peer delay mechanism** enables a port A (requestor) to estimate the propagation delay with respect to a port B (responder)*
  - without any need to receive incoming Sync messages
  - A is not a slave port



PTP – Synchronization Mechanism 6/14

- *The requesting node (A) records the time stamp t1 before sending a Pdelay_Req message to B*
- *When the request is received, the time stamp t2 is taken.*
- *Then, as quickly as possible, B records a time stamp t3 and replies with a Pdelay_Resp message*
- *A fourth time stamp t4 is finally recorded by the requestor at the response message arrival*
- *The responder B can embed the difference t3-t2*
  - in *Pdelay_Resp*
  - or in a subsequent (optional) *Pdelay_Resp_Follow_Up* message.
- *As an alternative*
  - *t2* can be included in *Pdelay_Resp*
  - *t3* can be included in *P d e l a y_ Re s p_ F o l l ow_ Up*

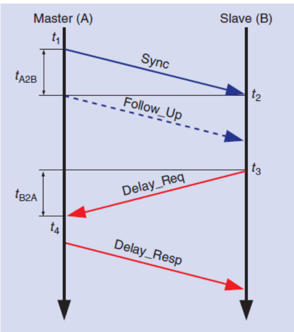
PTP – Synchronization Mechanism 7/14

- *The path delay, dprop, is then computed by A as follows:*
  - *correctionField* is the sum of the corresponding values that are found in the *Pdelay_Resp* and *Pdelay_Resp_ Follow_Up* messages

$$d_{prop} = \frac{(t_4 - t_1) - (t_3 - t_2) - correctionField}{2}$$

PTP – Synchronization Mechanism 8/14

- *The **delay request-response** mechanism is the typical method used by a slave to synchronize its clock with that of the master*



PTP – Synchronization Mechanism 9/14

- *The master (A) records the time stamp t1 before sending a Sync multicast message*
- *If the underlying network does not support multicast transmissions:*
  - a distinct *Sync* message is sent by the master to every slave
- *The value of t1 can be transmitted through either*
  - the *Sync*
  - or a subsequent (optional) *Follow_Up* message
- *The use of Follow_Up messages is recommended*
  - *t*1 cannot be easily included in the leaving *Sync* frame
  - time stamps are recorded immediately before the transmission and/or after the arrival of a packet
- *Obs: the same is valid for the Pdelay_Resp_Follow_Up message used in the peer delay mechanism*

PTP – Synchronization Mechanism 10/14

- *The slave records the time stamp t2 on the Sync reception*
- *t3 is taken before returning a Delay_Req request to the master*
- *The master records the time stamp t4*
  - sends it back immediately using a *Delay_Resp* message
- *At this point, the slave is able to compute the path delay d_{prop}*

$$d_{prop} = \frac{(t_4 - t_1) - (t_3 - t_2) - correctionField}{2}$$

PTP – Synchronization Mechanism 11/14

- *All event messages, with the exception of Delay_Req, are transmitted cyclically*
  - Delay requests, in the case of multicast communications, should be sent according to a random uniform time distribution in order to balance the master load better
  - Requests can be transmitted cyclically when unicast communications are used
- *Delay requests are issued according to a time scheme uniformly distributed between 4 and 60 s*
- *The period for sending other event messages is about 2 s*
- *Synchronization accuracy depends on synchronization intervals*
  - Shorter intervals lead to higher precision

PTP – Synchronization Mechanism 12/14

- *(4) Offset*
- *The estimation of the path delay*
- *The availability of the four time stamps t1, t2, t3 and t4, obtained with the delay request-response mechanism*

- *Enable the slave to compute the offset with respect to the master clock by means of the following equation:*

$$offset = t_2 - t_1 - d_{prop} - correctionField.$$

PTP – Synchronization Mechanism 13/14

- *Offset values are used for synchronizing to the master clock*

- *The IEEE 1588 standard does not specify any particular algorithm to perform synchronization*
  - Filters can be adopted to smoothen the time-stamp jitter
  - A proportional-integral controller allows to correct the slave clock frequency
  - Linear regression and PTP time stamps are used to derive the speed ratio between the slave and master clocks
  - A virtual clock can be assumed as the synchronized time base

PTP – Synchronization Mechanism 14/14

- *PTP is natively a fault-tolerant protocol*
- *Only the faulty nodes are affected in the case of slave failures*

- *The BMC mechanism takes care of the automatic election of a new node when the current master breaks down*
  - But more than 120 s may be required for the recover
  - When the sending period of *Sync* messages is set to 2 s

PTP – Performance 1/2

- *The accuracy of PTP heavily relies on the precision of time stamps*
- *It depends on jitters and channel symmetry*
  - How close t_{A2B} is to t_{B2A}
- *Time stamps can be managed at different points in the communication stack*
  - at the physical layer
  - in the system kernel (interrupt service routine)
  - at the application level
- *Asymmetry and jitters increase with the layer in the protocol stack*
- *Time stamps obtained at the physical level are much more precise than those taken at the application level*

PTP – Performance 2/2

- *In hardware-based solutions, time values are acquired directly by special circuits in network adapters*
- *Possible to have in 99.7% of the samples:*
  - Clock difference between pairs of synchronized nodes from -10 ns to +10 ns

- *In kernel-based solutions, performance is degraded by software time stamps and the statistical distribution of offset values is wider*

- *Time stamps can be taken at the application level*
  - Similar to kernel based when hard real-time OS and protocol stacks are used
  - Offset values in the order of 10 us for 99% of the samples
  - Otherwise, jitters would be significantly larger, especially with other interfering tasks concurrently executed in the system

Summary

- *Introduction*
- *NTP – Network Time Protocol*
- *PTP – Precision Time Protocol*
- *Flexray Clock Synchronization*
- *Final Remarks*

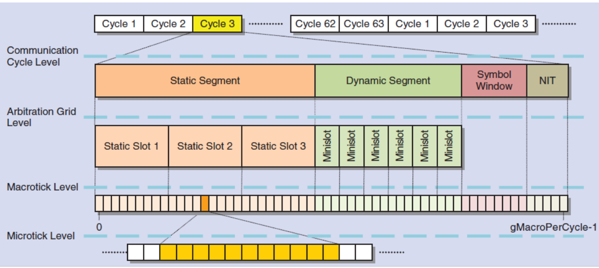
Flexray

- *Flexray is a communication protocol developed for automotive systems*
  - Interconnection of electronic control units in vehicles
- *Time-triggered medium access technique*
- *Native availability of a redundant (double) physical channel*
- *Bit rates as high as 10 Mb/s*

- *Flexray achieves noticeably higher degrees of determinism, fault-tolerance, and performance than CAN*

Flexray – Communication Infrastructure 1/10

- *Access to the shared transmission support is based on a communication cycle that is repeated indefinitely*
  - The duration of the cycle is fixed, though configurable



Flexray – Communication Infrastructure 2/10

- *Each communication cycle is divided into segments (up to four)*
  - The static and dynamic segments
  - The symbol window
  - The network idle time (NIT)

- *In the (mandatory) static segment, a conventional time division multiple access (TDMA) mechanism is adopted*
- *Each transmitting node is assigned its own time slot*
  - Collision free communication
  - Highly deterministic

Flexray – Communication Infrastructure 3/10

- *In the (optional) dynamic segment, a special technique known as a flexible time division multiple access (FTDMA) is used*
- *FTDMA relies on a minislotting approach in which messages are characterized by consecutive identifiers*
- *Each node that does not have a message to transmit generates a period of inactivity on the network (minislot)*
- *The duration of a minislot is much shorter than the static slot length*
  - Flexible data exchanges
  - Schedule is decided at the run time

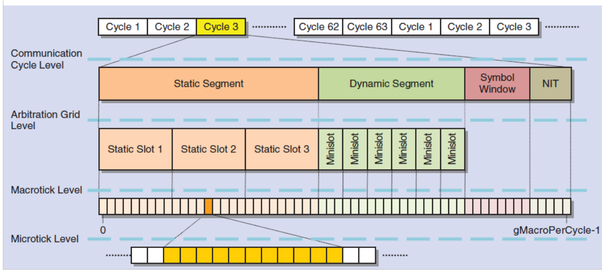
Flexray – Communication Infrastructure 4/10

- *The (optional) symbol window can be used to exchange a single symbol and its arbitration is not managed directly by FR*
- *It is mainly used in the startup phase*

- *NIT is a period at the end of each cycle when the network is kept idle*
- *NIT is used to perform clock corrections*

Flexray – Communication Infrastructure 5/10

- *Precise synchronization of all nodes in the network is a prerequisite for the proper operation of the TDMA and FTDMA mechanisms*
- *A timing hierarchy is defined that consists of four levels*



Flexray – Communication Infrastructure 6/10

- *The arbitration grid level is placed just below the topmost communication cycle level*
- *This is where the static segment is split into a fixed (configurable) number of (static) slots of the same duration*
- *Each slot is assigned to a specific node and can be used to send exactly one frame*
- *Such an assignment is carried out in the configuration phase, before the system is started*
- *It cannot be changed during the normal network operation*

Flexray – Communication Infrastructure 7/10

- *The dynamic segment consists of several identical minislots*
- *Each frame takes an integral number of minislots for its transmission*
  - Messages of different sizes can be accommodated easily
- *This technique permits the transmission of sporadic frames when needed by a device*
- *The medium access technique in this segment is based on message priorities*
  - The resulting behavior someway resembles CAN
  - But the actual mechanisms of the two protocols are very different

Flexray – Communication Infrastructure 8/10

- *The macrotick level is found beneath the arbitration grid*
- *At this level, every node sees the time as a sequence of macroticks with the same duration*
- *Precise alignment of macroticks is maintained by means of a suitable synchronization mechanism*
- *Action points are also defined here:*
  they are global time instants when significant events must/can occur

Flexray – Communication Infrastructure 9/10

- *The concept of global time in Flexray is quite different from other popular solutions*
- *No reference node is present and a true, absolute time is not defined*
- *Nevertheless, at any instant, all nodes share a common view of both*
  the current cycle (vCycleCounter parameter) and
  macrotick (vMacrotick parameter)
  in the cycle
- *This can easily be exploited by the upper levels to perform coordinate actions*

Flexray – Communication Infrastructure 10/10

- *Microtick is the lowest level in the hierarchy*
- *Every node generates microticks from the local oscillator*
- *Microticks in the different nodes neither have the same duration nor are synchronized in any way*
- *This level is mainly related to practical controller implementations*

Flexray – Synchronization Mechanism 1/7

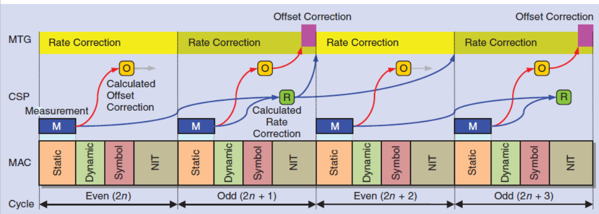
- *Communication and synchronization are tightly interleaved*
- *During the normal operation*
  - Correct FR transmissions require proper synchronization of all the nodes in the network
  - The synchronization mechanism is based on message exchanges between the nodes

Flexray – Synchronization Mechanism 2/7

- *Clock synchronization relies on two processes that operate concurrently*
- *The macrotick generation (MTG)*
  - MTG grants the alignment of macroticks (the arbitration grid) across the whole network by applying suitable rate and offset corrections
- *The CSP*
  - CSP is responsible for both measuring deviations of the local clocks and computing values to correct their rate and offset
  - Nodes measure the difference between the expected and the actual arrival times for every sync frame exchanged in the static segment
  - The expected arrival time is a static slot action point, whereas the actual time is the instant when the frame is received
  - Timestamps are obtained through the local oscillator and are expressed in microticks

Flexray – Synchronization Mechanism 3/7

- *Sync frames are a subset of all static frames*



Flexray – Synchronization Mechanism 4/7

- *Nodes store deviation values*
  - for both channels and separately for even- and odd-numbered cycles
  - in local tables
- *Execute a fault tolerant midpoint (FTM) algorithm*
- *1) A value k is selected based on the number of rows in the table*
  - $k = 0$ for one or two rows, $k = 1$ for three to seven rows, and $k = 2$ otherwise
- *2) The list of measured values is sorted, and the k smallest and largest values are discarded*
- *3) The smallest and largest values in the remaining set are selected*
- *and their average computed*

- *The value obtained is assumed as the deviation of the local clock from the global clock*

Flexray – Synchronization Mechanism 5/7

- *The estimation of the offset deviation occurs on every cycle*
- *The rate deviation is evaluated once for every pair of consecutive cycles*
- *The rate deviation is used to correct the oscillator rate in the following cycle pair*
- *Computed values are checked against suitable limits before applying corrections*
- *If the values exceed the predefined limits, suitable recovery procedures are undertaken*

Flexray – Synchronization Mechanism 6/7

- *The MTG process adjusts the local MTG rate by tuning a parameter that specifies the number of microticks per cycle*
- *This mechanism has been conceived so that*
  - corrections can be applied smoothly
  - abrupt changes in the local view of time avoided

- *No synchronization is possible without communication and vice versa*
- *Network start-up uses a special approach*

Flexray – Synchronization Mechanism 7/7

- *Precisions in the order of several tens of nanoseconds can be obtained*
- *Under realistic conditions, the system precision stays in the order of a few hundreds of nanoseconds*
  - even when the frequencies of some local oscillators deviate from their nominal values in a nonnegligible way

Summary

- *Introduction*
- *NTP – Network Time Protocol*
- *PTP – Precision Time Protocol*
- *Flexray Clock Synchronization*
- *Final Remarks*

Final Remarks 1/2

- *The availability of accurate clock synchronization in distributed control systems is important*
  - Modern industrial plants
  - Energy distribution
  - Networked embedded control systems
- *Basically, all clock synchronization protocols rely on similar techniques*
  - Time measurement
  - Propagation delay evaluation
  - Offset compensation
  - Rate compensation
- *differing mainly in the way the mechanisms are implemented*

Final Remarks 2/2

- *Protocols differ*
  - in the type and size of the underlying network
  - the synchronization approach (centralized versus distributed)
  - the relationships with communication (ranging from complete independence to tight integration)
- *Security aspects and synchronization over wireless networks were not considered*

Summary

- *Introduction*
- *NTP – Network Time Protocol*
- *PTP – Precision Time Protocol*
- *Flexray Clock Synchronization*
- *Final Remarks*