GE Fanuc
Intelligent Platforms

TTEthernet – A Powerful Network Solution for Advanced Integrated Systems

GE FANUC

TTTech

TTEthernet – A Powerful Time-Triggered Network Solution

As the most widely-installed local area network technology, Ethernet is used as a universal network solution in office and web applications, and production facilities. Engineering, maintenance and training costs for Ethernet-based networks are considerably lower than costs for many proprietary bus systems and Ethernet generally offers higher bandwidths. But when Ethernet was developed over 30 years ago, time-critical, deterministic or safety-relevant tasks were not taken into account.

Time-Triggered Ethernet (TTEthernet) expands classical Ethernet use with powerful services (SAE AS6802) to meet the new requirements of reliable, real-time data delivery in advanced integrated systems. In addition, TTEthernet switches provide ARINC 664 functionality to meet existing requirements of avionics Ethernet networks.

With TTEthernet, critical control systems, audio/video and standard LAN applications can share one network. TTEthernet facilitates design of mixed criticality systems and system-of-systems integration.

In the aviation domain, TTEthernet can be used for high-speed active controls, smart sensor and actuator networks, deterministic avionics and vehicle backbone networks, critical audio/video delivery, reflective memory, modular controls and integrated modular systems such as Integrated Modular Avionics (IMA) or distributed IMA. TTEthernet also targets also critical embedded systems in aerospace and defense, automotive, medical, energy production, and industrial automation.

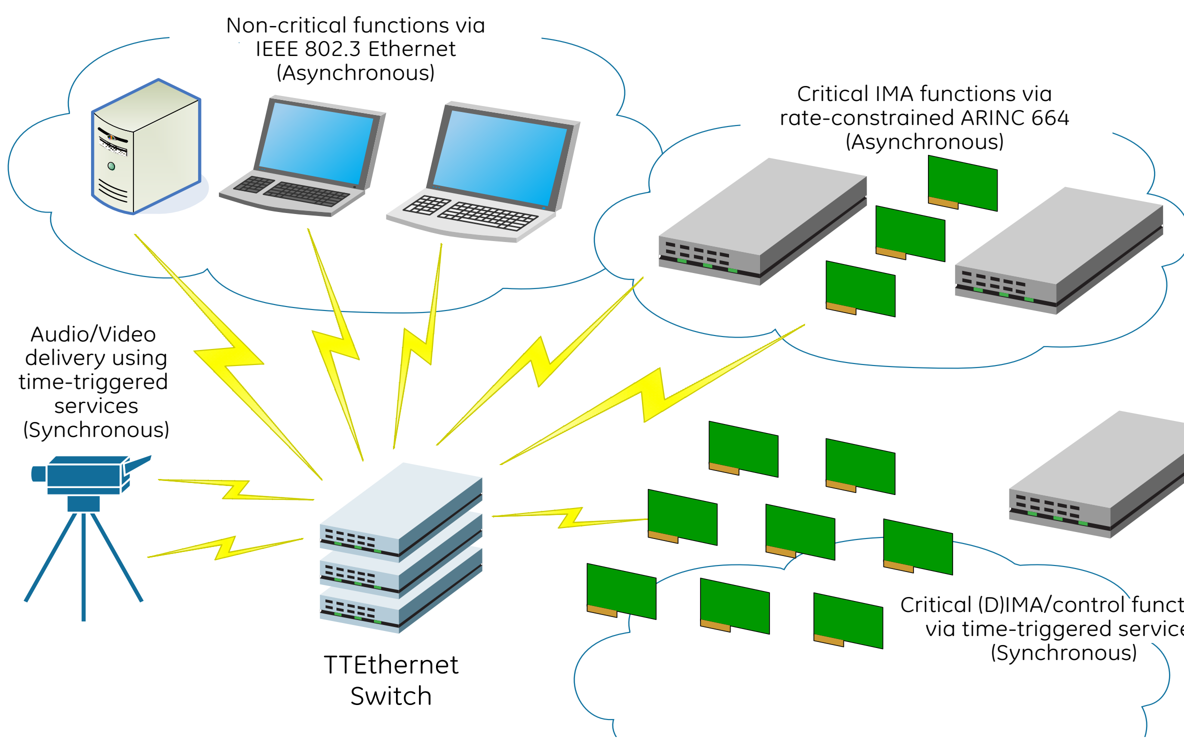


Figure 1: TTEthernet (SAE AS6802) enables design of advanced integrated systems utilizing asynchronous and synchronous communication via IEEE 802.3 Ethernet. It is scalable and supports fault-tolerant (N-redundant), time-critical and mixed criticality functions in one network. TTEthernet supports system-of-systems integration.

Determinism in Critical Ethernet Networks

Determinism is dependant on application domain requirements and represents predictable operation performance. For networking domains, determinism is related to:

- temporal communication behavior (message jitter and latency)
- predictable bandwidth use

An Ethernet network with specified jitter and latency can be seen as deterministic “enough”, assuming that predictable data exchange without data traffic congestion is viable for a given application case. For ARINC 664 networks, the context of determinism is defined as the control of maximum transmission delay (latency) throughout the network.

In the Time-Triggered Ethernet standard, SAE AS6802, Time-Triggered Ethernet provides extensions to standard IEEE 802.3 to support hard real-time, rate-constrained and unconstrained IEEE 802.3 traffic on a common mixed-criticality network. A synchronization strategy based on time-triggered principles is described in the SAE AS6802 standard, defining a fault-tolerant self-stabilizing synchronization strategy. Devices that comply with this standard are capable of synchronizing their local clocks to each other in a fault-tolerant way. The context of determinism is defined as exact system timing throughout the system. Messages are transferred based on a fault-tolerant system time base with microsecond jitter.

Predictable (deterministic) operation in IEEE 802.3 Ethernet networks for critical embedded systems can be achieved by:

- Asynchronous approach (ARINC 664): Constraining the rate (frequency) of data transmissions (e.g. max. jitter 500µs, latency > jitter) with sampling rates of upto 1KHz. Bandwidth partitioning is based on rate-constrained traffic shaping (in end systems) and policing (in the switch)
- Synchronous approach (SAE AS6802): Establishing fault-tolerant synchronized operation using asynchronous Ethernet messaging with sampling rates of upto 50kHz (jitter below few microseconds). The bandwidth partitioning is based on exact (µs) time base and message delivery based on time-triggered services.
- Mixed asynchronous/synchronous approach to satisfy different contexts of determinism in different applications using critical Ethernet networks

TTEthernet supports both asynchronous and synchronous (ARINC 664 and SAE AS6802) approaches to deterministic networking, and enables parallel operation in mixed asynchronous/synchronous networks. It is designed to cover cross-industry application needs and provide deterministic network operation for a broad range of different applications. The primary reason for the integration of both SAE AS6802 and ARINC 664 on the same TTEthernet switch is the ensured availability of time-triggered services. Without those services it would be impossible to define robust network partitioning for asynchronous and synchronous data flows.

System Integration using Synchronous and Asynchronous Ethernet Communication

Distributed functionality in advanced integrated network systems is established by coordinated operation of different functions with different criticality-levels and quality of service. In order to establish coordination among all of the different functions on a network, some sort of “synchronization” is required either at the network, middleware or application layer, or combined at different layers at once.

IEEE 802.3 Ethernet provides asynchronous communication services. By constraining the maximum rate of message delivery, jitter (e.g. 500µs) and latency, the deterministic communication behavior for avionics applications can be accomplished without synchronous communication, as described in the ARINC 664 (Avionics Full Duplex Ethernet) Specification.

At the network level, ARINC 664 operates asynchronously and coordination among distributed functions using ARINC 664 networks is “synchronized” at higher-level OSI layers. In this case, jitter is in Nx100 µs range and latencies are in the order of 1 to 10 milliseconds or higher. High jitter influences the accuracy of point-to-point latency and limits maximum achievable sampling rates, especially in a complex system with several switches (multi-hop networks). Maximum latencies of 10 or more milliseconds are not unusual in complex ARINC 664 networks. Therefore the applications and higher layers should be designed to be robust against latency and high jitter. So the idea of determinism here is not to constrain jitter, but to have a known maximum latency throughout the network.

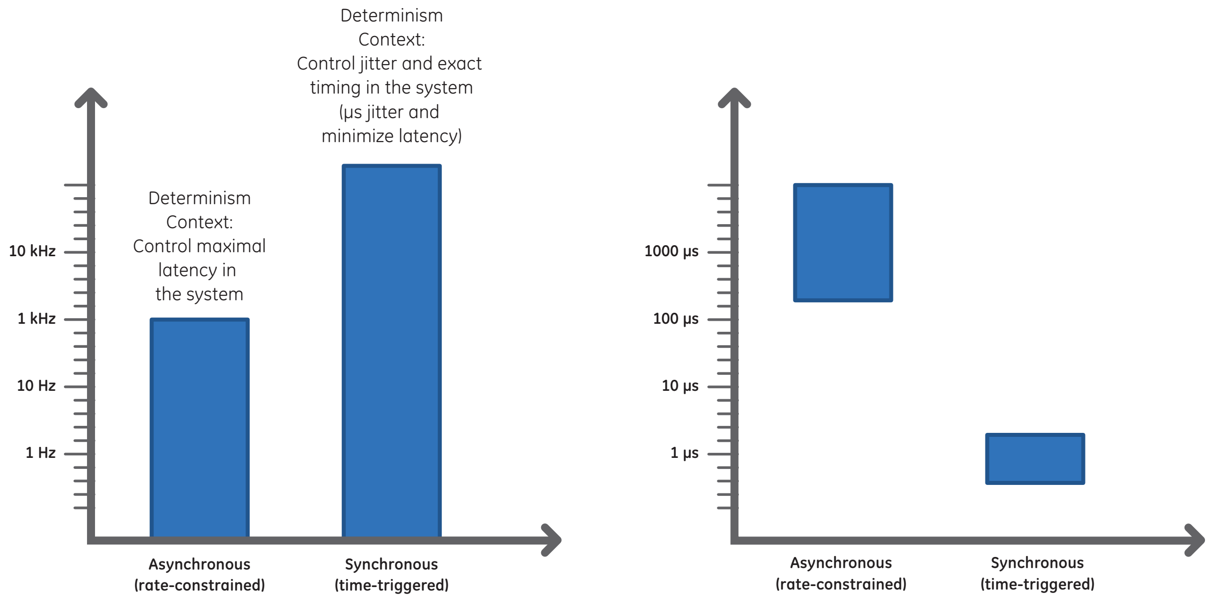


Figure 2 Determinism context in Ethernet networks depends of the application (max. sampling rate) and the approach to system design – asynchronous (coordination and synchronization among functions is conducted at higher layers) or synchronous (control of timing and synchronization at network level).

It is easier to control accurate (µs jitter!) network timing using services which are integrated closer to the communication layer, so it makes sense to add fault-tolerant synchronization at the Ethernet network level as an additional service (SAE AS6802). Higher layers and middleware are therefore simpler and can support efficient use of computing resources.

Due to low jitter and ability to narrowly control latency, it is possible to precisely control network bandwidth use and its allocation to asynchronous or synchronous traffic with robust network bandwidth partitioning. This level of control supports design of well-defined and unambiguous key system interfaces to simplify system integration. So determinism in SAE AS6802 Time-Triggered Ethernet is to minimize jitter and at the same time have well-defined latency. The control of timing throughout a network system also helps to reduce maximum latency. Otherwise the jitter in a system represents an uncertainty which requires additional margins be added to the maximum latency to guarantee deterministic behavior. The positive side-effect of a synchronous approach is that latency can be ruled out from system design considerations if the global time base is available and the jitter is in microseconds.

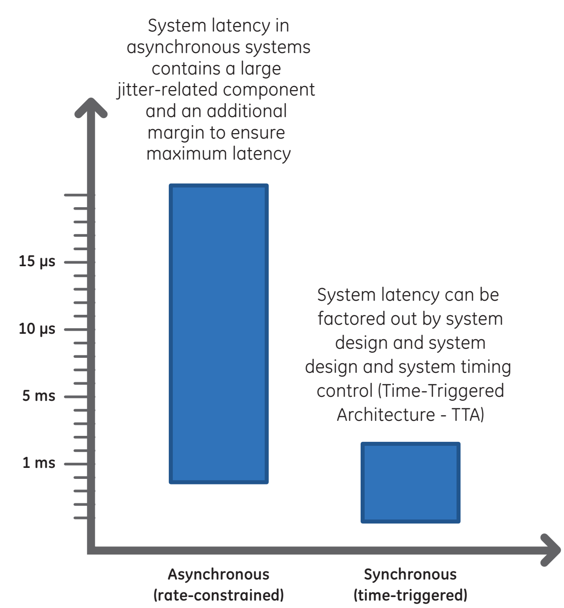


Figure 3 Maximum latency throughout the system is influenced by jitter in asynchronous (rate constrained) and synchronous (time-triggered) systems. The maximum latency in the system represents a trade-off between synchronous and asynchronous design paradigm.

In avionics networks, the SAE AS6802 standard is a synchronous communication technology enabling hard real-time network communication and the transfer of critical audio/video data which is not handled by ARINC 664. This is where time-triggered services as defined by the SAE AS6802 standard complement asynchronous services from the ARINC 664 Specification.

Because TTEthernet switches can handle both synchronous and asynchronous communication in one Ethernet-based network, a network system designer can select a networking approach which makes sense for a specific deterministic application and can deliver a desired level of latency and jitter control throughout the system.

Network Dependability and Critical Ethernet Networks
Network dependability is a term which extends well beyond "system reliability", covering such aspects as availability, reliability, integrity maintainability, confidentiality and safety. Beyond the simplified definition of communication determinism, network dependability also contributes to a systems determinism.

System engineering not only covers dependability, but many other "-ilities", such as survivability, security, adaptability, scalability, upgradeability, and real-time capability. Marrying dependability with any other "-ility" is a non-trivial task. In most cases Ethernet-based networks are tailored and modified for specific application use with focus on one dimension of dependability, but are generally less flexible and hard to integrate with IEEE 802.3 Ethernet systems. Design of dependable distributed network systems can be achieved in one or a few dimensions, but even if a network system covers all dependability criteria, the following questions need to be answered:

- Will the network system be scalable, upgradeable and affordable?
- Will Quality of Service (QoS) be sufficient for time-critical applications?
- Will the system have predictable (deterministic) behavior?

TTEthernet technology exploits synchronous operation and resulting fault-tolerant time base to support different system dependability dimensions at once. It simplifies design challenges for complex distributed systems by robust partitioning, unambiguous definition of key system interfaces, and support for redundant and time-driven system design.

Robust system-level partitioning and distributed computing
With asynchronous communication, it is possible to guarantee the bandwidth use by traffic shaping and policing, without exact timing. With synchronous time-triggered services it is possible to define exact sections of bandwidth to be used for a time-driven function, and the remaining bandwidth slots to be used by asynchronous data traffic (ARINC 664 Part 7 or IEEE 802.3), so hard real-time communication will not be impacted by asynchronous or event-drive functions. The availability of fault-tolerant global time plays a critical role in network bandwidth partitioning and enables parallel operation of multiple streams with different QoS, including audio/video and hard real-time control loops.

There is a similarity of robust network resource partitioning and computing resource partitioning (time/space) on the host microprocessor. The time and space partitioning (via MMU and partitioning OS), as defined by ARINC 653, emerged in Integrated Modular Avionics (IMA) to enable design of many functions utilizing the same computing, housing and power supply resources to take advantage of the availability of common system time (tasks run on one processor with common time anyway!).

If consistent common time can be extended to the whole system, it is possible to take advantage of time/space/communication partitioning to design distributed functions of mixed criticality integrated in one network. This type of resource partitioning in a networked system we call "system-level partitioning". This means all distributed functions can be executed without being influenced by other less critical networked functions. From this perspective, the concept and benefits of Integrated Modular Avionics (IMA) or modular aerospace controls (MAC) can be extended to the whole networked system.

System tasks can be scheduled to take advantage of global time, send and receive data just in-time. With broadcast communication, this would mean that the periodic data transfers play the role of a distributed virtual shared system memory, an equivalent to the reflective memory. At the

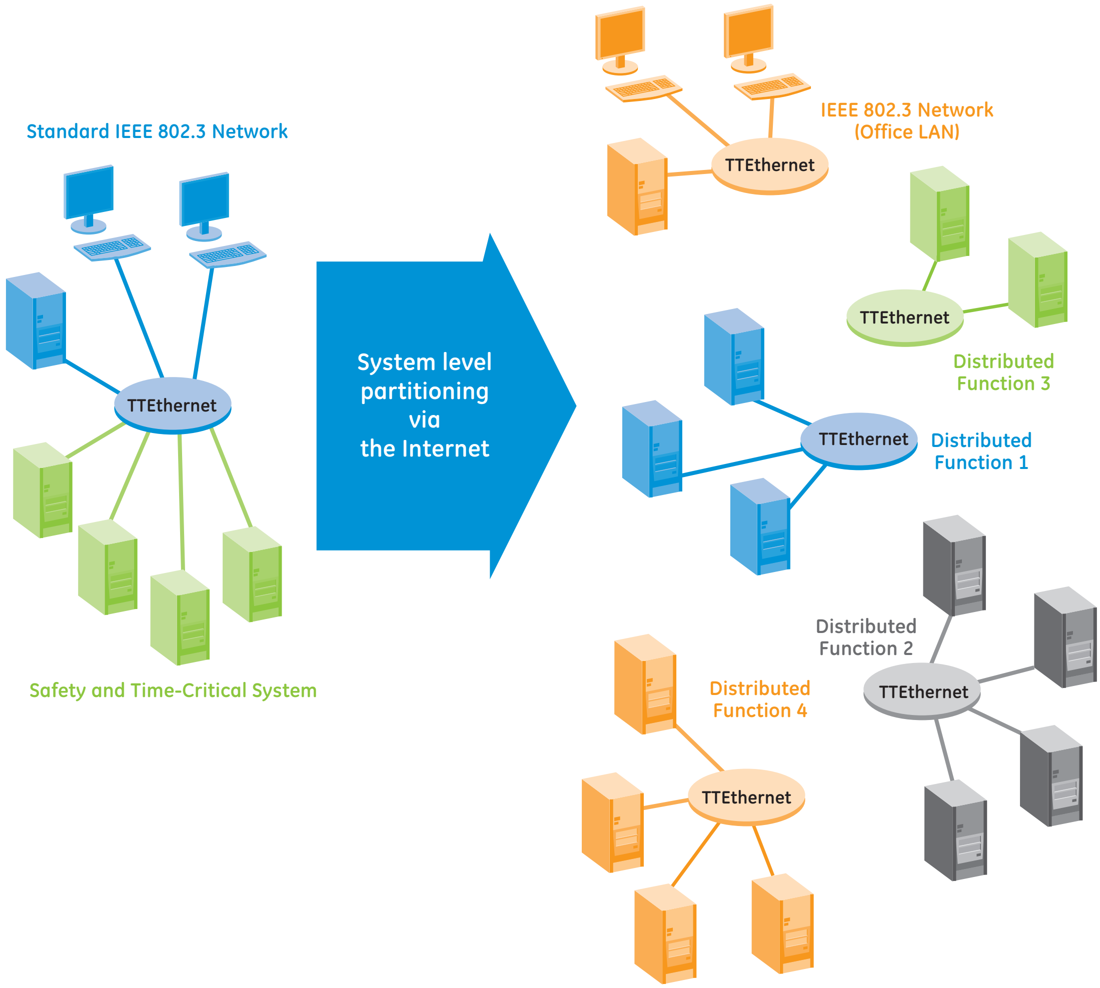


Figure 4 System-level partitioning with TTEthernet. Different distributed critical and non-critical functions and non-critical functions can operate in TTEthernet networks taking advantage of time/space partitioning (host CPU) and network bandwidth partitioning. The failure of one system will not impact other distributed functions.

same time other non-critical data can be shared in the network without impact on reflective memory operation (see "Use case for TTEthernet"). In such a case, from the application perspective, all tasks and functions run on one fault-tolerant distributed embedded computer.

System-level partitioning concept simplifies design of fault-tolerant distributed embedded computing platforms and is very useful for design of centralized and distributed IMA, reflective memories, and smart sensor/actuator networks.

With robust system-level partitioning, TTEthernet enables division of distributed functions in private subsystems with adjustable levels of coupling.

TTEthernet adds a few novel properties, such as synchronization islands which simplify design of logically separate functions in one network. This relaxes synchronization constraints for time-driven distributed functions and simplifies modular contracting strategies (e.g. Modular Open System Acquisition (MOSA)).

Mixed Criticality Systems with TTEthernet

With robust communication partitioning and the availability of global time base in the system all critical distributed functions will be not influenced by other less critical functions running on different endsystems and communicating over the same network. Bandwidth is partitioned (TDMA) to accommodate different types of traffic and QoS.

In avionics or vehicle systems, it is possible to have in the same network the following systems:

- Redundant by-wire controls with hard real-time loop running on a distributed system
- Displays with time-critical datastreams
- Audio/Video Communication Networks
- Integrated Vehicle Health Management (IVHM) taking advantage of existing sensors in the system
- Stores management and controls
- Linux-based embedded computer for diagnosis and collection of maintenance data
- Windows laptop for mission planning

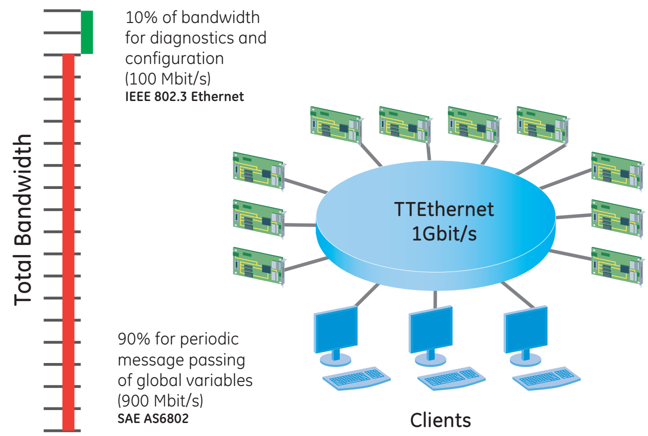
TTEthernet enables design of mixed criticality systems using IEEE 802.3 Ethernet, rate-constrained and time-triggered communication services. This enables more efficient use of total bandwidth (see Table 1). In comparison, an asynchronous network communication is not deterministic with more than 20-25% bandwidth utilization. With both asynchronous and synchronous traffic, the deterministic bandwidth utilization can rise to 70-80%. An overview of capabilities for asynchronous vs synchronous traffic is given here:

	Asynchronous Traffic	Synchronous	Asynchronous/Synchronous
Bandwidth utilization in real-world (mixed criticality) applications	20-25%	>80%	50-70% (assuming 50% sync, 50% async. traffic)
Bandwidth partitioning	Traffic shaping and enforcing	Exact definition of TDMA slots and time base	By using SAE AS6802 services, ARINC 664 / IEEE 802.3 traffic (LAN) can be integrated
Deterministic Communication	Max. sampling rate 1kHz	Max. sampling rate 50kHz at 1Gbit/s	Max sampling rate >Nx10kHz
Hard Real-Time Control Loops	Limited to 1KHz, assuming non-critical functions consume minor part of the bandwidth	Fully supported under any workload	Fully supported under any workload
Audio/Video Delivery	No	Fully supported under any workload	Fully supported under any workload

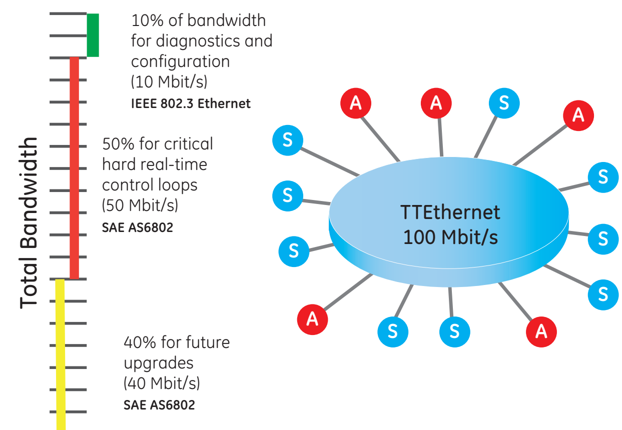
Table 1 Mixed criticality systems and comparison of asynchronous, synchronous and mixed asynch/synch approach to data communication

TTEthernet: A Powerful Network Solution for Advanced Integrated Systems

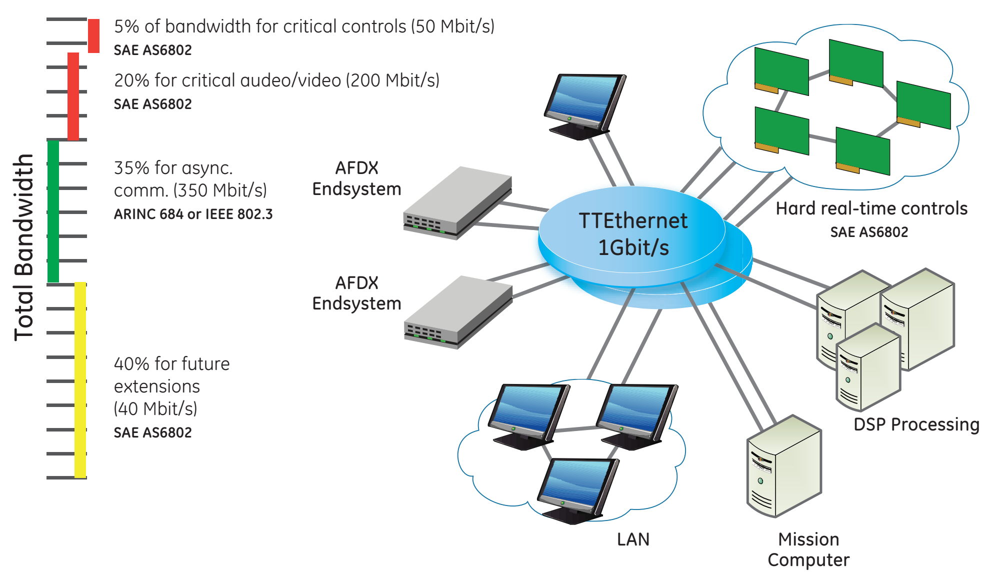
Use Cases for TTEthernet



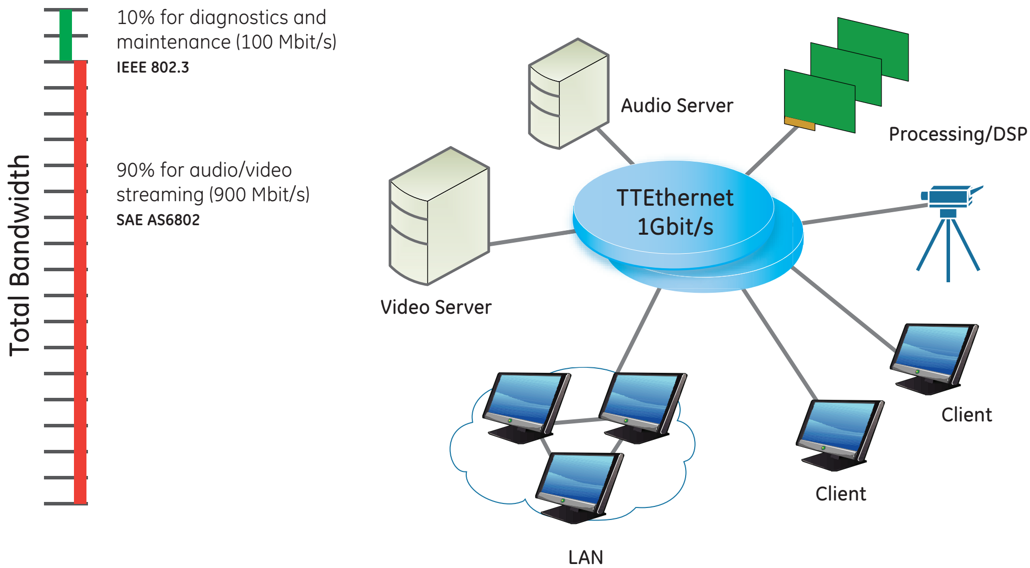
Reflective Memory (configured for 64 units, ~1kByte message/unit, periodic update 5kHz)



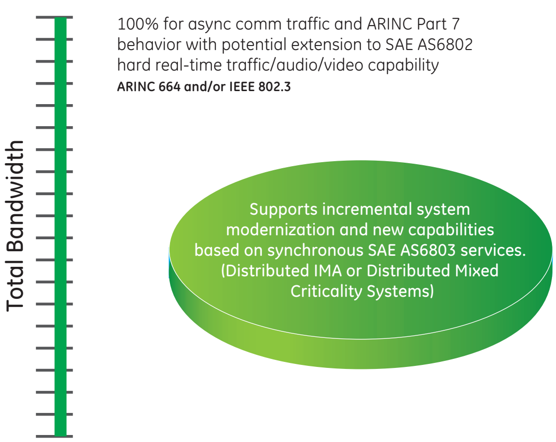
Smart Sensor/Actuator Network without standalone processing units (100 Mbit/s)



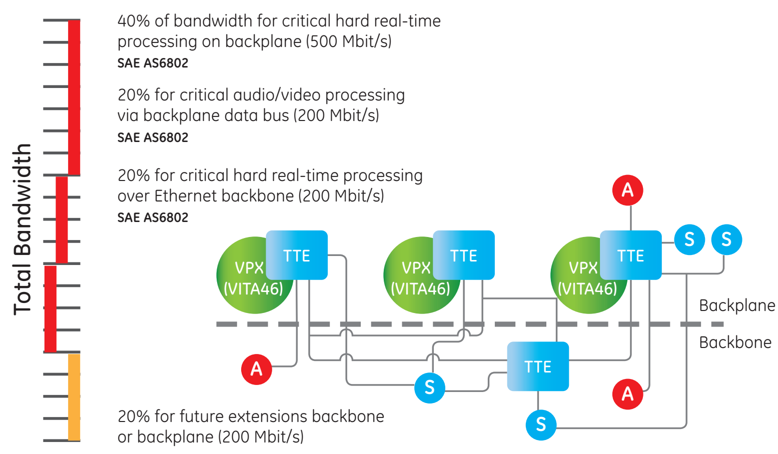
Mixed Criticality Systems/Intgrated Vehicle Network (1 GbE)



Audio/Video Delivery Payload Processing



Use ARINC 664 Part 7 (AFDX) capabilities only



TTEthernet as switched fabrics for VPX (VITA46) systems (blurs difference between backplane and backbone bus)

TTEthernet System Properties

TTEthernet has time-triggered services that enable time triggered communication over Ethernet. These time-triggered services establish and maintain a global time, which is realized by the close synchronization of the local clocks of the devices. The global time forms the basis for system properties such as communication partitioning, precise diagnosis, efficient resource utilization, and unambiguous definition of key system interfaces (composability).

Dataflow Options in TTEthernet

TTEthernet specifies services that enable time-triggered communication on top of Ethernet. Time triggered services can exist in parallel to the usual OSI layers. A communication controller that implements these services is able to synchronize with other communication controllers and switches in the system, and can send messages at points in time derived from this system-wide synchronization. These messages are then called time-triggered messages.

As TTEthernet supports communication among applications with various real-time and safety requirements over a network, three different traffic types are provided: time-triggered (TT) traffic, rate-constrained (RC) traffic, and best-effort (BE) traffic. If required, the corresponding traffic type of a message can be identified based on a message's Ethernet Destination address. The relation of the TTEthernet traffic types to existing standards is depicted in Figure 2.

Messages from higher layer protocols, like IP or UDP, can be "made" time-triggered without modifications of the messages' contents itself. The TTEthernet protocol overhead is transmitted in dedicated messages termed protocol control frames, which are used to establish system-wide synchronization. In short, TTEthernet is only concerned with "when" a data message is sent, not with specific contents within in a message.

TT messages are used for time-triggered applications. All TT messages are sent over the network at predefined times and take precedence over all other traffic types. TT messages are optimally suited for communication in distributed real-time systems. TT messages are typically used for brake-by-wire and steer-by-wire systems that close rapid control loops over the network. TT messages allow designing and testing strictly deterministic distributed systems, where the behavior of all system components can be specified, analyzed and tested with sub-micro second precision.

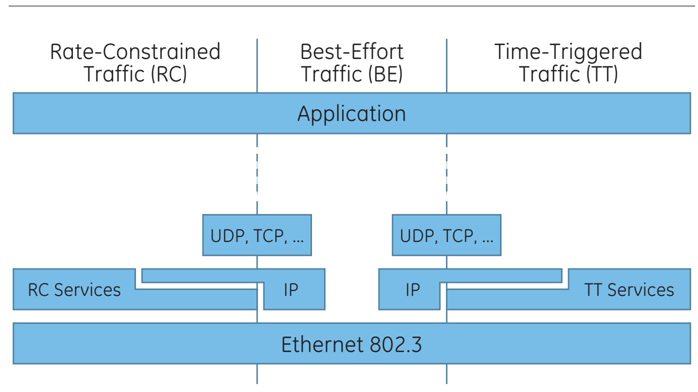


Figure 2 Relation of TTEthernet to existing communication standards

RC messages are used for applications with less stringent determinism and real-time requirements than strictly time-triggered applications. RC messages guarantee that bandwidth is predefined for each application, and delays and temporal deviations have defined limits. RC messages are used for safety-critical automotive and aerospace applications that depend on highly reliable communication and have moderate temporal quality requirements. Typically, RC messages are also used for multimedia systems.

In contrast to TT messages, RC messages are not sent with respect to a system-wide synchronized time base. Hence, different communication controllers may send RC messages at the same point in time to the same receiver. As a consequence, the RC messages may queue up in the network switches, leading to increased transmission jitter. As the transmission rate of the RC messages is bound a priori and controlled in the network switches, an upper bound on the transmission jitter can be calculated off-line and message loss is prevented.

BE messages follow a method that is well-known in classical Ethernet networks. There is no guarantee whether and when these messages can be transmitted, what delays occur and

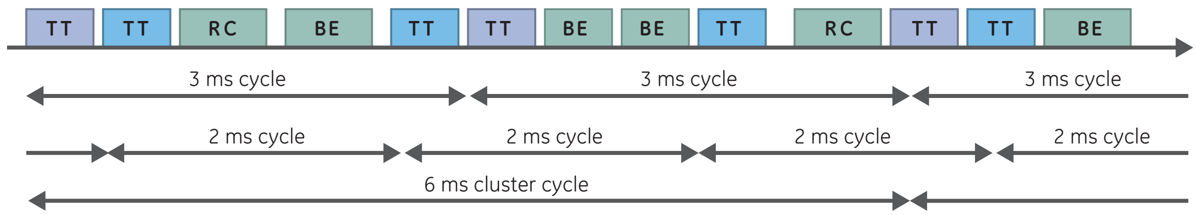


Figure 3 TTEthernet includes TT, RC and BE messages

if BE messages arrive at the recipient. BE messages use the remaining bandwidth of the network and have less priority than TT and RC messages. Typical user of BE messages are web services. All legacy Ethernet traffic (e.g. internet protocols) without any QoS requirement can be mapped to this service class. TTEthernet implements strong partitioning between non-critical BE traffic and all other service classes (see Figure 3).

TTEthernet as Transparent Synchronization Protocol

TTEthernet is a transparent synchronization protocol, i.e., it is able to co-exist with other traffic, potentially legacy traffic, on the same physical communication network. For reasons of fault tolerance a multitude of devices can be configured to generate synchronization messages. The devices generating the synchronization messages may be distributed with a high number of intermediate devices in between each other.

TTEthernet defines basic building blocks that allow the transparent integration of the time-triggered services on top of message-based communication infrastructures such as standard Ethernet. For this, TTEthernet defines a novel application of the transparent clock mechanism that enables the concept of the permanence point in time, which allows re-establishing the send order of messages in a receiver:

- Application of transparent clock mechanism: all devices in the distributed computer network that impose a dynamic delay on the transmission, reception, or relay of a synchronization message add this dynamic delay into a dedicated field in the synchronization messages used for the synchronization protocol.
- Precise calculation of the permanence point in time: the application of transparent clock mechanism allows a precise re-establishment of the temporal order of synchronization messages. In a first step the worst case delay is calculated off-line. In a second step, each synchronization message is delayed for "worst case delay minus dynamic delay" upon reception of the synchronization message, where the dynamic delay is the delay added to the synchronization message, as the synchronization message flows through the communication channel. This point after the reception point in time will be called the permanence point in time. For fault-tolerant algorithms in general, and fault-tolerant synchronization algorithms in particular, the message send order is of highest importance. The re-establishment of the send order of synchronization messages is required for any fault-masking synchronization protocol that ensures synchronization of local clocks in a distributed computer network.

Safety and Fault Tolerance

A high level of safety is provided by the time-triggered method of TTEthernet, which detects failures and irregularities in the network and certain systems. Additional measures need to be taken to achieve maximum safety, availability and fault tolerance.

TTEthernet networks can be set up with multiple redundant end systems, switches and segments. Thus the system will remain in operation even if faults occur. Redundant network paths are always used in fault-tolerant TTEthernet systems so that the failure of a single system or messages can be tolerated without affecting the application. If multiple redundancy is implemented, multiple faults can be tolerated. It is important that the entire system remains in operation without interrupts under the same temporal conditions as defined before.

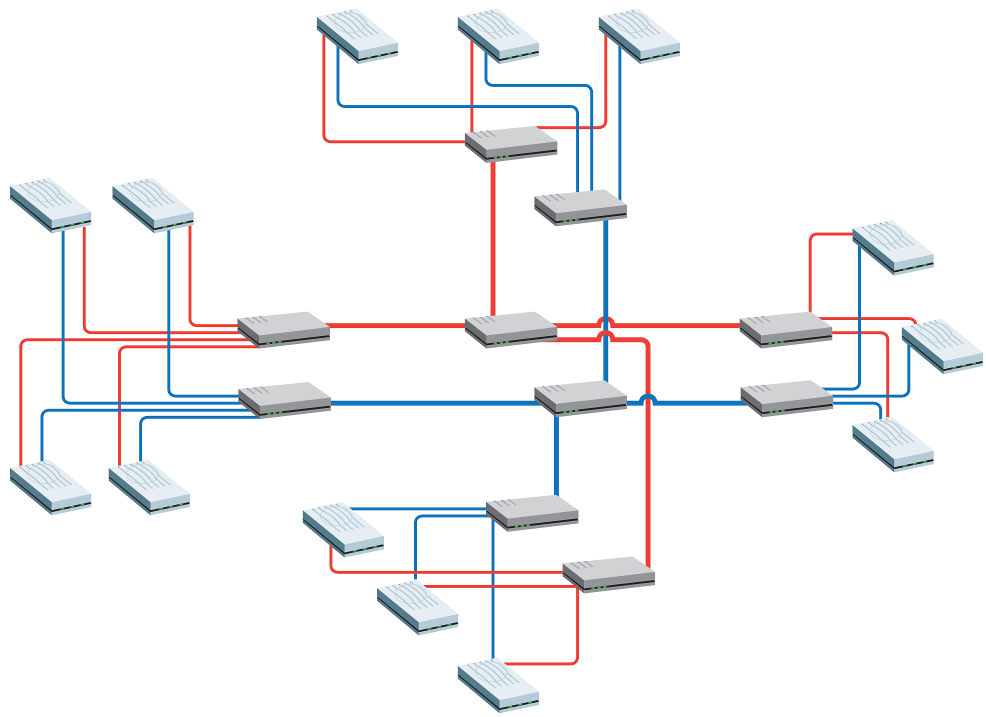


Figure 4 TTEthernet provides implicit fault tolerance mechanisms

TTEhernet allows the integration of guardians in switches and end systems. Guardians check if the communication on the network works in compliance with the predefined parameters. If faulty systems block network segments, the guardian disconnects the network segment or port. Multiple redundant guardians can be implemented to meet the highest safety requirements (see Figure 4).

Fault-Tolerant Capabilities

TTEthernet is designed to scale over a multitude of cross-industry applications. As such, TTEthernet comprises demanding fault-tolerant capabilities:

- TTEthernet is scalable: TTEthernet can be configured to operate as a simple master-slave synchronization protocol for industrial control or a multi-master synchronization protocol for civil avionics. This scalability gives a vast economic benefit because the cost of the realization of TTEthernet throughout different application domains can be decreased significantly. Likewise, the cross-domain usage of TTEthernet increases the probability of latent failure detection in the realization of TTEthernet and contributes to the "service history" of TTEthernet, when deployed in systems with a comparable level of criticality.

- TTEthernet tolerates multiple inconsistent faults: When configured to a multi-master mode,

  TTEthernet tolerates a fully inconsistent-omission faulty communication path and even an inconsistent-omission faulty end system at the same point in time. This failure mode means that each faulty device can arbitrarily drop messages on any of its incoming communication links and on any of its outgoing communication links with potential inconsistent dropping behavior for each message.

  TTEthernet therefore allows a more cost-efficient realization of system architectures that require tolerance of multiple concurrent failures in the system.

- TTEthernet tolerates arbitrary end system failures: The switches in TTEthernet can be configured to execute a central bus guardian function. The central bus guardian function guarantees that even if a set of end systems becomes arbitrarily faulty, the system-wide impact of these faulty end systems is masked. In a distributed hard real-time system based on a broadcast bus for inter-node communication, it is important to prevent a single faulty node from monopolizing the communication bus. The arbitrarily faulty failure mode also includes babbling idiot behavior and similar failure modes in a time-triggered communication system. TTEthernet switches establish fault containment boundaries.

- TTEthernet tolerates arbitrary transient disturbances even in presence of permanent failures: In addition to fault tolerance, TTEthernet also provides self-stabilization properties, i.e., the synchronization will be re-established even after transient upsets in a multitude of devices in the distributed computer system. TTEthernet stabilizes from an arbitrary system state to a synchronized system state. This self-stabilizing property becomes more and more important with decreasing feature sizes of computer chips and, therefore, resulting increase in transient upsets. The design of future reliable distributed computer networks depends on an effective and sound tolerance of multiple transient upsets.

Network Structure

TTEthernet supports all physical layers specified in IEEE 802.3 for switch-based networks. Even subnetworks with different bandwidths (100 Mb/s, 1 Gb/s, etc.) are supported.

Switches in TTEthernet have the central role of organizing the data communication. TT messages are routed in the switch according to a predefined schedule with as little delay as possible. Precise planning at the time of system design precludes resource conflicts at runtime. TT messages have the highest priority level. If the planned transmission time of one of these messages arrives, this message is immediately transmitted. Due to the predefined transmission of the message the switch ensures that the medium is free at the time of transmission and delays are precluded.

RC messages are routed with little delay. If TT messages are to be transmitted via the same outgoing port at the same time, the TT messages take priority over the RC messages. TT messages can delay RC messages. RC messages are transmitted if no planned transmission of TT messages is pending and the sender observes the minimal transmission distance. The switch is responsible for arranging several RC messages at an outgoing port.

BE messages always have the lowest priority. RC and TT messages can delay or discard BE messages at the same outgoing port. The switch uses the remaining bandwidth for BE messages if no TT or RC messages are to be transmitted. BE messages are transmitted after all pending RC messages. This method exploits the bandwidth of the network in an optimal way. Tools are used to design and verify a TTEthernet system in advance. This ensures that the bandwidth for TT, RC and BE messages is always sufficient according to the requirements of the application and interrupts are reduced to a minimum. Later incremental changes of the system configuration are possible.

TTEthernet switches allow the simultaneous distribution of TT messages to groups of end systems or the connection of unsynchronized TTEthernet networks. This is how TTEthernet networks can be divided into smaller application-specific sub-networks and the design can be facilitated.

Supported Topologies

TTEthernet allows synchronizing local clocks in a distributed computer network. Of particular interest are computer networks that exchange information via messages that are sent on communication links between devices in the network. In standard Ethernet, end systems are connected with network switches via bidirectional communication links. An end system will communicate with a second end system or a group of end systems by sending a message to the switch, which then relays the message to the receiving end system or end systems. Also, switches can be connected to each other via bi-directional communication links. The resulting architecture is referred to as a multi-hop architecture and the links between any two switches as the multi-hop link.

Communication links and switches are said to form a communication channel between end systems. End systems can be connected directly to each other via bi-directional communication links, which makes a clear differentiation between end systems and switches in certain configurations difficult. Generally we use the term device to refer to a physical device that can be either end system or switch. Whether a device is regarded as an end system or a switch is determined by its usage rather than its physical appearance.

Synchrony

Events in time-triggered systems occur at predefined times with a precision at the single microsecond level. This also includes the communication of TT (Time-Triggered) messages. The system design specifies when the TT messages are transmitted by which participants and who shall receive them. This ensures that the network processes TT messages without collisions (i.e. without data congestion in the switches) and the recipient can continuously check the quality of the deterministic system if, for example, a message fails to arrive at the predefined time or does not arrive at all. This makes TTEthernet suited for applications of the highest safety integrity level.

Synchronization among all participants is crucial for the transmission of TT messages. TTEthernet always transmits clock synchronization messages to keep the clocks of the end systems and switches in synchrony. For this purpose TTEthernet relies on a redundant hierarchical master-slave method that has a distributed fault-tolerant majority of master nodes and master switches to provide the time in the system. This guarantees both the fail-safe operation and the high quality in synchronization. This method is unique for TTEthernet and can be combined with other mechanisms such IEEE 1588 (see Figure 5).

IEEE 1588 specifies a synchronization protocol for Ethernet. The global time base of TTEthernet can be leveraged to synchronize native IEEE 1588 synchronization clients, too. For this purpose, additional functionality can be realized on top of a TTEthernet device that generates IEEE 1588 clock synchronization frames. TTEthernet provides means to compensate for delays through the TTEthernet network. Outside the TTEthernet network, in a native IEEE 1588 network, the clock synchronization messages can be handled as native IEEE 1588 clock synchronization messages.

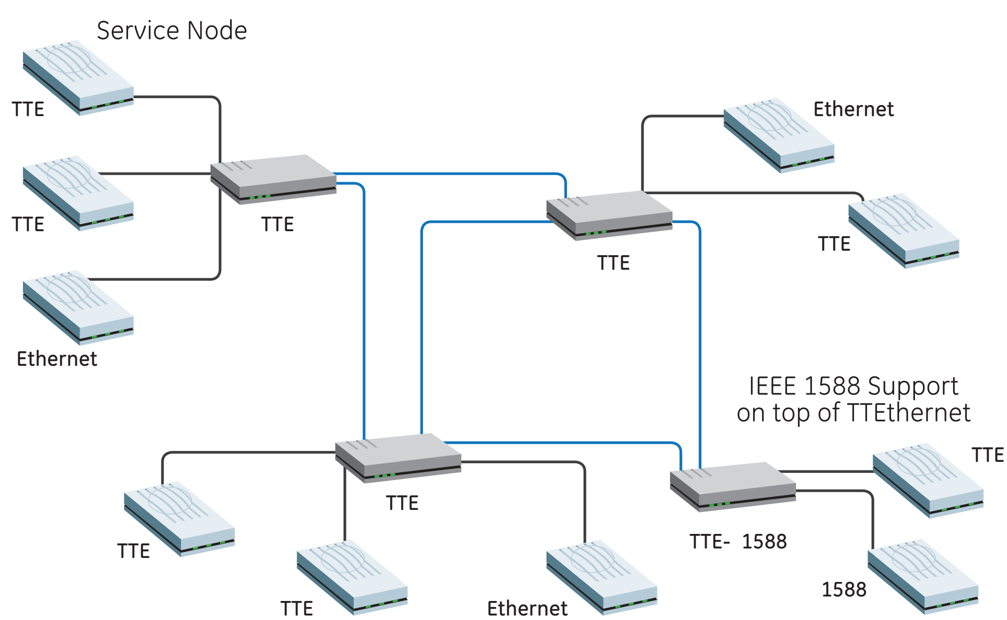


Figure 5 TTEthernet allows for flexible, even redundant network topologies and synchronization with other systems such as IEEE 1588

Synchronization Approach

TTEthernet takes a two-step approach to synchronization. In the first step synchronization masters send protocol control frames to the compression masters. The compression masters then calculate an averaging value from the relative arrival times of these protocol control frames and send out a new protocol control frame in a second step. This new protocol control frame is then also sent to synchronization clients.

The decision on which devices are configured as synchronization masters, synchronization clients, and compression masters arises from the requirements on the system architecture. End systems can be configured as synchronization masters and switches as compression masters. But system configurations with end systems configured as compression masters and switches as synchronization masters are also possible.

Switches and end systems not configured either as synchronization or compression masters will be configured as synchronization clients (see Figure 6).

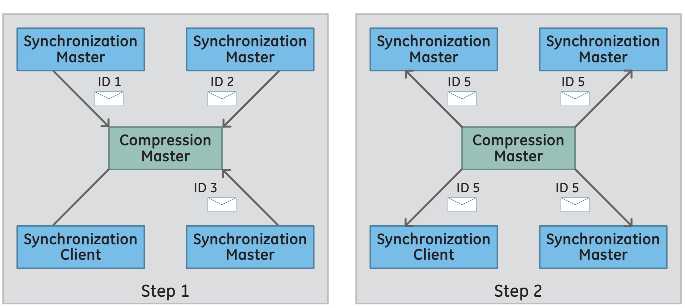


Figure 6 TTEthernet follows a two-step synchronization approach

Synchronization Topology

TTEthernet distinguishes four different levels in synchronization topology. On the lowest level, TTEthernet defines the device level that comprises synchronization masters, synchronization clients, and compression masters. The cluster level groups devices with the same synchronization priority and the same synchronization domain to a single cluster. On the multi-cluster level, several clusters with different synchronization priorities but same synchronization domain are grouped together. Finally, the network level groups different clusters (potentially multi-clusters) with different synchronization priorities and different synchronization domains (see Figure 7).

Network Level	y Synchronization Domains (y, z) Synchronization Priorities
Multi-Cluster Level	One Synchronization Domain x Synchronization Priorities
Cluster Level	One Synchronization Domain One Synchronization Priority
Device Level	Synchronization Masters Synchronization Clients Compression Masters

Figure 7 The TTEthernet synchronization topology has four levels

TTEthernet specifies the concept of a cluster. A TTEthernet cluster is a group of end systems and switches that have the same synchronization priority and synchronization domain. TTEthernet clusters could be used in large TTEthernet networks, where different clusters shall be able to run in isolation, but shall be able to operate in a master-slave mode, once a high priority cluster joins the network or is powered on.

A TTEthernet simple cluster consists of a set of end systems that are connected to each other via an optionally redundant set of communication channels, where each communication channel consists of one switch only (see Figure 8). In a TTEthernet cascaded cluster configuration each communication channel consists of more than one switch (see Figure 9).

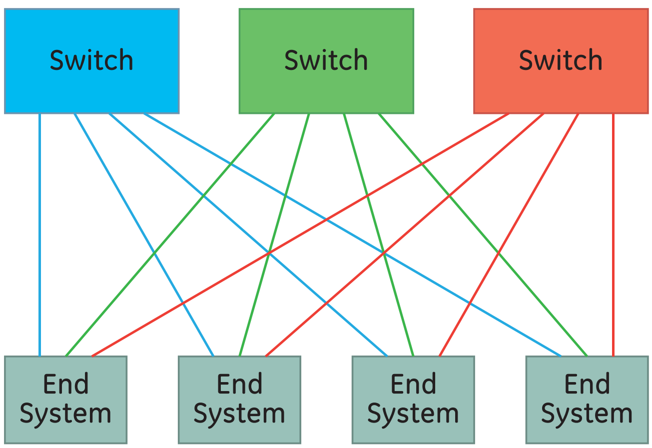


Figure 8 TTEthernet simple cluster with three redundant channels

TTEthernet specifies different synchronization priorities. Synchronization in a multi-cluster system is usually done according a master-slave paradigm, where the devices will synchronize towards the highest synchronization priority. TTEthernet also specifies different synchronization domains. A synchronization domain is a group of TTEthernet clusters that will not synchronize to each other. However, dataflow between two TTEthernet clusters in different synchronization domains can be done using RC or BE traffic.

Variable Implementation

A TTEthernet integration for end systems can be implemented in hardware or software, depending on such requirements as temporal quality, safety and fault tolerance. A TTEthernet system can always be connected to conventional Ethernet systems without affecting the predefined behavior. But there might be a lack of bandwidth for this additional system.

Even standard PCs can participate in a TTEthernet system. These scenarios are possible:

- A PC with a conventional Network Interface Card (NIC) can send and receive BE messages.

  Equipped with dedicated software the PC can also receive and analyze TT and RC messages.
- A PC with conventional NIC and a TTEthernet stack is a software-based end system (SES) that allows the reception and transmission of TTC, RC and BE messages. But the PC software is the limiting factor to the temporal precision.
- A PC with specific TTEthernet NIC can send and receive TT, RC and BE messages with the highest temporal precision.

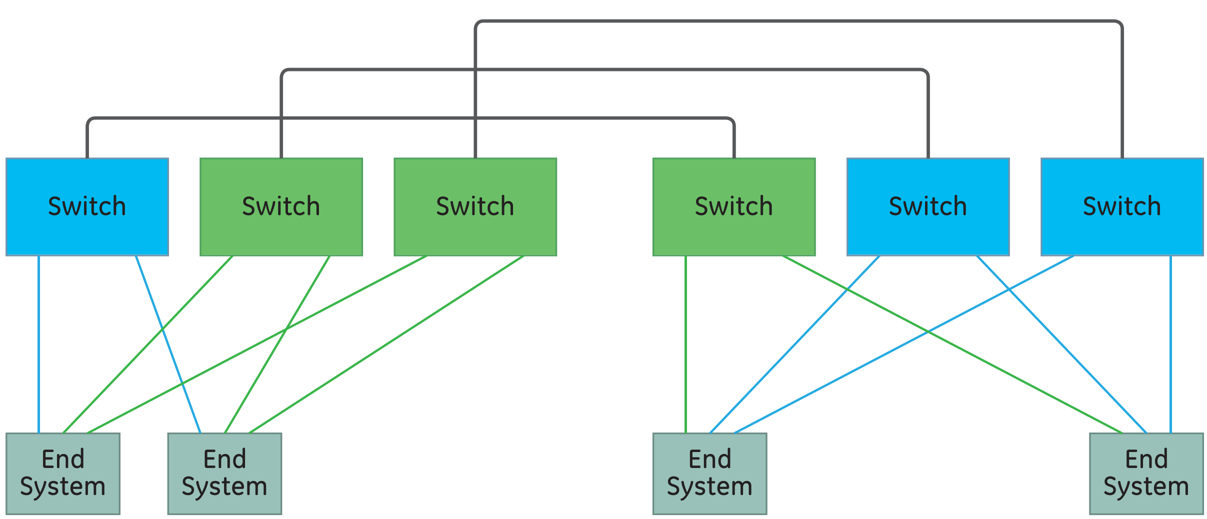


Figure 9 TTEthernet cascaded cluster with three redundant channels

Non-PC-based embedded systems can implement TTEhernet. This can happen in stack-based software on standard Ethernet hardware or in dedicated hardware controllers.

This broad implementation freedom is a result of TTEthernet's compatibility to the Ethernet standard as only Ethernet messages are used in TTEthernet. However, there is a natural trade-off between the implementation options and the temporal quality of TTEthernet. A TTEthernet protocol stack that is implemented on a standard PC with standard NICs may, for example, achieve a precision in the order of hundred microseconds, while a dedicated hardware implementation will come down to a one digit microsecond precision and below. Still, the lower temporal quality arising from standard Ethernet controllers is sufficient for a multitude of real-time control processes. In both cases the deterministic properties of time-triggered systems can be maintained

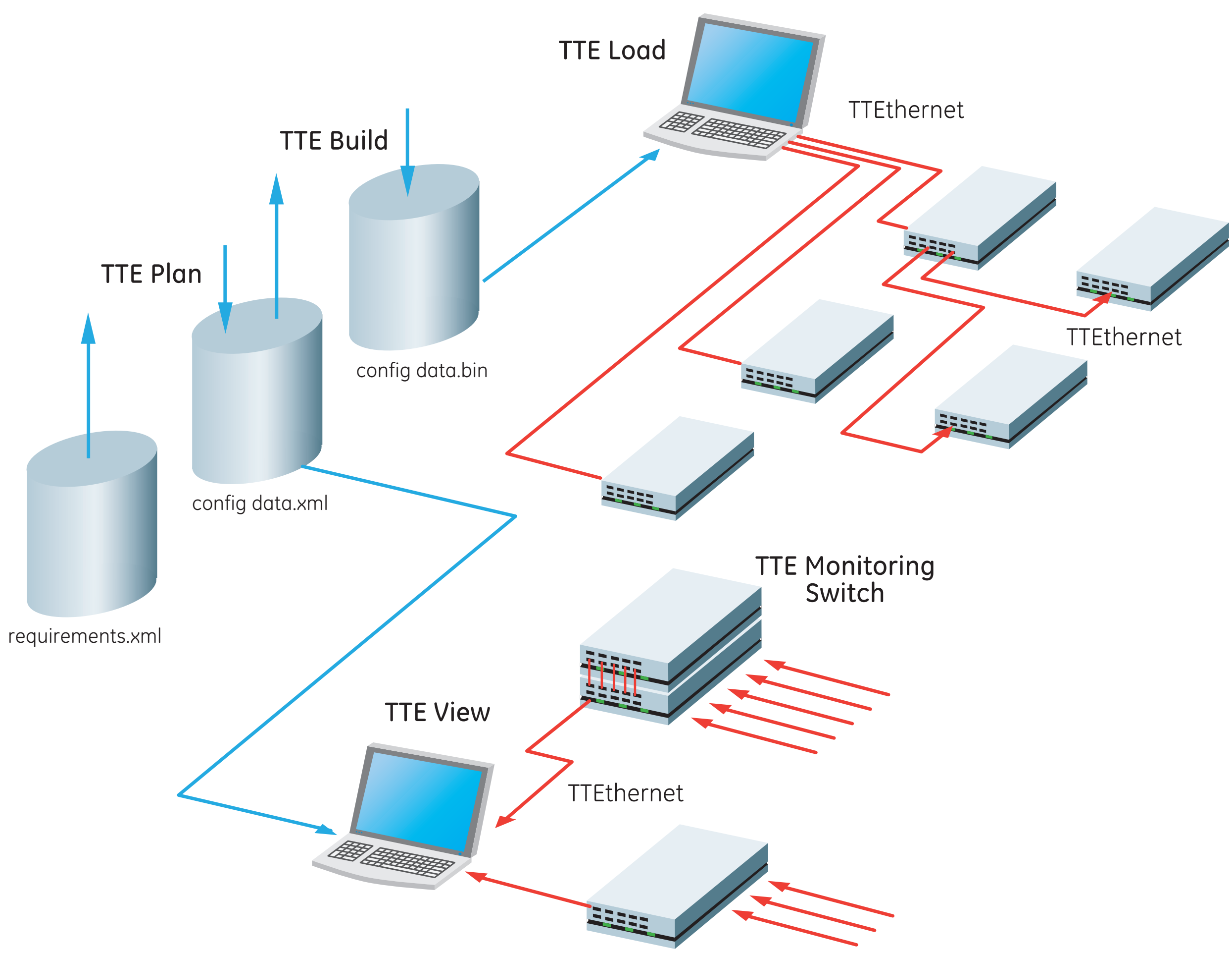


Figure 10 The TTEthernet tool chain covers the entire life cycle of a network

Engineering Tools

GE Fanuc Intelligent Platforms is releasing proven time-triggered tools have been ported for TTEthernet systems. These tools cover the entire lifecycle of the network. Automatic and manual modeling tools allow the intuitive system design in terms of temporal behavior, network and topology. TT-based software applications generate configuration data that comply with the communication schedule and load the data into the involved systems. Monitoring switches can display the network traffic on-line and off-line, and check the accuracy and consistency of a designed system including the temporal behavior of TT and RC messages. These tools can also generate detailed reports for approving a system in compliance with application regulations such as DO-178B in the aerospace industry (see Figure 10). Open XML data exchange formats allow the simple and seamless integration with third-party tools.

Conclusion

Time-Triggered Ethernet enables time-triggered communication over Ethernet networks in all application areas. The network provides all necessary mechanisms for applications as diverse as classical web services and time-critical and safety-critical control system in aircraft. Existing networks can be extended step by step using TTEthernet-capable switches and end systems without the need to change existing applications and end systems. Reducing network solutions to established and recognized Ethernet standards opens up saving potentials that secure major advantages in competitive markets. TTEthernet has great potential not only in extremely demanding aerospace applications but also in completely new cross-industry application areas.

GE Fanuc Intelligent Platforms Information Centers

Headquarters:
1 800 GEFANUC
1 800 322 3616
1 434 978 5100

Global Regional phone numbers are available on our web site www.gefanuc.com

Additional Resources

For more information, please visit the GE Fanuc Intelligent Platforms web site at:

www.gefanuc.com

08.09 GFT-751