NIST Technical Note 1867

Time-Aware Applications, Computers, and Communication Systems (TAACCS)

Marc Weiss
John Eidson
Charles Barry
David Broman
Leon Goldin
Bob Iannucci
Edward A. Lee
Kevin Stanton

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.TN.1867

National Institute of Standards and Technology
U.S. Department of Commerce

NIST Technical Note 1867

Time-Aware Applications, Computers, and Communication Systems (TAACCS)

Marc Weiss
Time and Frequency Division
Physical Measurement Laboratory

John Eidson
Edward A. Lee
University of California, Berkeley
Berkeley, CA

Charles Barry
Jolata Inc.
San Jose, CA

David Broman
KTH Royal Institute of Technology and University of California, Berkeley
Stockholm, Sweden and Berkeley, CA

Leon Goldin
Integrated Device Technology, Inc.
Ottawa, Canada

Bob Iannucci
Carnegie Mellon University, Silicon Valley Campus
Moffet Field, CA

Kevin Stanton
Intel Corporation
Hillsboro, OR

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.TN.1867

February 2015

U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Acting Under Secretary of Commerce for Standards and Technology and Acting Director

National Institute of Standards and Technology Technical Note 1867
Natl. Inst. Stand. Technol. Tech. Note 1867, 25 pages (February 2015)
CODEN: NTNOEF

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.TN.1867

Time-Aware Applications, Computers, and Communication Systems (TAACCS)

Marc Weiss, NIST, Time and Frequency Division, mweiss@nist.gov
John Eidson, University of California at Berkeley,
Charles Barry, CTO, Jolata Inc.,
David Broman, KTH Royal Institute of Technology and UC Berkeley
Leon Goldin, Compliance Manager, IDT Inc.,
Bob Iannucci, Department of Electrical and Computer Engineering, Silicon Valley Campus, Carnegie Mellon University,
Edward A. Lee, EECS Department, University of California at Berkeley,
Kevin Stanton, Sr. Principal Engineer, Intel Corporation

Abstract

A new economy built on the massive growth of endpoints on the internet will require precise and verifiable timing in ways that current systems do not support. Applications, computers, and communications systems have been developed with modules and layers that optimize data processing but degrade accurate timing. State-of-the-art systems now use timing only as a performance metric. Correctness of timing as a metric cannot currently be designed into systems independent of hardware and/or software implementations. To enable the massive growth predicted, accurate timing needs cross-disciplinary research to be integrated into these existing systems. This paper reviews the state of the art in six crucial areas central to the use of timing signals in these systems. Each area is shown to have critical issues requiring accuracy or integrity levels of timing, that need research contributions from a range of disciplines to solve.

Keywords

Internet of Everything (IoE), Internet of Things(IoT), Industrial Internet, Machine to machine, Cyber-physical systems, Time transfer, Clocks, Discrete event systems, Distributed control, Ordering, PTIDES, Real-time systems, Simultaneity, Synchronization, and Timestamps.

CONTENTS

1 INTRODUCTION

We stand at the advent of a revolutionary new economy fueled by the global *Internet of Everything*, IoE, a combination of the traditional telecom system with its growing need for wireless technology, and the emerging *Internet of Things*, IoT, [1] [2], including Machine-to-Machine (M2M) technology [3]. Cisco, among others, predicts that there will be a trillion endpoints connected to the internet by 2022, with $14.4 trillion in value at stake [4]. General Electric, GE, says "about 46% of the global economy or $32.3 trillion in global output can benefit from the Industrial Internet" [5]. The National Institute of Standards and Technology (NIST) has formed a Cyber-Physical Sytems (CPS) Public Working Group to bring together experts to help define and shape key aspects of CPS, and to create a framework and reference architectures to encourage interoperability and appropriate designs [6]. One fundamental enabler of this revolution will be a marriage of timing signals and data that breaks through the existing barriers. Currently, optimal use of data in computing and networking is anathema to optimal use of timing signals. Computer hardware, software and networking all isolate timing processes, allowing the data to be processed with maximum efficiency due in part to asynchrony. Yet, coordination of processes, timestamping of events, latency measurement and optimal use of precious spectrum are enabled by timing.

Timing is critical for the future development and improvements to several current high value applications. For example, smart transportation where the exchange of information between vehicles, highways, and perhaps civil authorities will depend on a robust ubiquitous timing system to ensure the availability and integrity of the data. Similar requirements are found in the operation of the power grid, especially now that wind farms, solar arrays and the like, which will require different control strategies, are becoming an important part of the system. Medical applications such as tele-surgery, and applications in financial systems are other important examples.

There are three different types of timing signals for synchronization: frequency, phase, and time. Frequency can be supplied by an individual clock, such as a commercial cesium standard, though pragmatism drives the use of oscillators that require calibration and active reference signals. By contrast, phase and time synchronization *always* require transport of signals plus data, the transfer of timing signals. Timing signals are physical; they occur on the physical layer of networks. The IoT will have many devices and applications that require frequency, time or phase synchronization. Frequency, time, and phase will all need to cross layers, boundaries, and networks from their sources in accurate clocks. Requirements for these transfer systems will include parameters that can create different, perhaps orthogonal, demands on systems. Accuracy, stability, integrity and security requirements are realized with different demands on systems. Timing is generally regarded today as a performance metric, not as a correctness criterion. Timing accuracy emerges from the details of the implementation: the system topology and the specific equipment used. To facilitate the massive growth of the IoE, data processing and networking will need to converge with timing, making integration of cyber and physical seamless. However, this requires new research at fundamental levels, allowing precise and verifiable timing signals to be designed with correctness criteria.

Research and development will be required in all aspects of timing, computing, and networking to develop time-aware applications, computers, and communications systems (TAACCS). We present needed research by suggesting a layering into six areas, though they are often coupled:

- clock and oscillator designs,
- time and frequency transfer methods,
- the use of timing in networking and communications systems,
- hardware and software architecture,
- design environments,
- and the design of applications.

Achieving innovation in these areas will require collaborative research that combines the resources of three different domains: industry, academia, and government. Each of these three domains has a different orientation for contributing to this research, as well as a different need for the research results. For example,

industry, academia, and government may prioritize very differently the value of short-term versus long-term results, as well the value of standardization. All different priorities are subject to common criteria for classifying the results, which include (1) different levels of accuracy or precision, (2) the extent of the availability of the technique, (3) the cost of the device or implementation, (4) the amount of power available or needed, and (5) the verifiability or security of the timing supplied. Each of these leads to different, yet essential types of challenges. For example, supplying sub-nanosecond timing over a network with best-effort reliability is quite different from supplying millisecond timing for safety-of-life applications.

The following sections will expand upon this brief introduction. Section 2 expands the six-layered research model as listed above. In section 3, we give a brief statement of the state of the art (SOA) for each layer, and most importantly discuss critical issues (CI) requiring research. Section 4 concludes with a brief summary of this needed research. Research problems will often appear as orthogonal challenges tying research in all these areas together, emphasizing the need for cross-discipline research to make solutions viable.

2 A LAYERED VIEW OF TIME-AWARE SYSTEMS

This section proposes a layered model for time-aware applications, computers, and communications systems (TAACCS) starting with the bottom layer of oscillators and clocks and proceeding up- time transfer, time-aware networks and communication systems, hardware and software timing support for applications, design environments, and finally applications that exploit timing. We note that many of these layers are coupled and requirements need to be addressed at all levels.

The layers are:

- Oscillators and clocks: A clock consists of an oscillator and a counter. These devices fundamentally produce a frequency and may or may not have a capability to maintain time. In practice, a clock as frequency supplier must usually be calibrated against a more accurate standard, remain stable, and/or must be locked to a frequency signal. However time, being a human artifact, can only be supplied by a clock after it has been initially set. In addition, frequent updates from a source of time are generally required to maintain time accuracy within the needed limits.
- Time and frequency transfer systems: The function of this layer is to synchronize clocks in different locations in a distributed system. Accurate time and phase fundamentally must be transfered through some physical medium, while frequency transfer is generally required for practical reasons. The most common methods in current use are *Global Navigation Satellite Systems* (GNSS) signals from the sky, and transfer through communication networks using protocols such as the *network time protocol*, NTP, or *precision time protocol*, PTP.
- Time-aware network and communications systems: Here we consider traditional communication technology but with enhanced capability to support time-aware services and communications. This section is subdivided into the following five areas.
  - Network hardware and software will need new designs to support and make use of time-awareness
  - Development of time-aware and time-controlled networks requires research in both propagating and using timing signals
  - Controlling latency in networks is crucial to many real-time applications
  - Performance monitoring is required for maintenance and service-level agreements, and is greatly enhanced by synchronization
  - Precision timing optimizes use of precious spectrum
- Timing support for applications: We need a semantics in programming for precision time and timing, allowing deterministic execution independent of the processing hardware, sensors, or actuating electronics. This will include timing artifacts in microprocessors, FPGAs, and other hardware, and

in the programming languages used to employ them. This section is subdivided into the following three areas.

  - Hardware and software support of predictable execution will need to balance the depth of change in systems with cost and implementation
  - Timing across interfaces will require standards and latency control both among CPUs and in crossing network domains
  - Scale issues need addressing in supplying time to large numbers of systems

- Development environments: Tools are required for designing, simulating, and generating code for time-aware systems. Though there are powerful tools available, incorporating precise timing and deterministic functioning will require a fundamental shift in the design of these tools and their relationship to physical levels of hardware.
- Time-aware applications: These ultimate consumers of timing signals will have diverse requirements. Accuracy, precision, and stability form one type of requirement, along with security and verifiability or integrity providing a rather different direction of needs. Levels of accuracy and the ability to trust the timing will determine the usefulness and safety of this new universe. Finally, operation within the power envelope of the IoT device, especially those that are battery-powered, will need to be achieved without degrading its cost effectiveness.

3 THE STATE OF THE ART AND CRITICAL ISSUES FOR EACH LAYER

In this section, we discuss the state of art (SOA) for each layer, and the critical issues (CI) requiring additional research to enable the most effective use of time in these systems.

3.1 Oscillators and clocks

Timing performance is fundamentally dependent upon an underlying oscillator, or ensemble of oscillators and the clocks constructed based on these oscillators.

3.1.1 State of the art

Time-aware systems depend on clocks. A clock consists of a counter and an oscillator and the performance of the clock depends on the granularity of the counter, e.g. the least significant bit, LSB, and the frequency and stability of the oscillator.

Of the various types of oscillators, the quartz crystal remains the most ubiquitous in electronic systems due to the relatively small size, power, stability and cost for a given performance range. However, a variety of silicon resonators are now being developed, some MEMS-based. The latter offers the possibility of ultra-high volume manufacturing and the potential for integration into chips/CPUs/DSPs/FPGAs. Each oscillator type has its own peculiarities and neither alone may be successful in meeting all of the requirements.

Representative state of the art oscillator attributes are shown in Table 1 [7]. Until recently OCXOs were quite expensive but recent developments have produced small OCXOs with excellent performance for very low cost [8]. The CSACs can be obtained in a $<$17cm^3, 35 grams, $<$120mW form factor but with approximate rubidium performance, hence should lead to new design tradeoffs once the cost is reduced [9].

3.1.2 Critical research issues

Power and cost constraints for the Internet of Things will be primary drivers in the creation of new designs for highly accurate time-keeping devices that can be deployed at the scale of tens-of-billions annually. Near term applications such as the smart grid and transportation will require more stability and accuracy than can be provided by mechanical oscillators. Atomic frequency standards are necessary at key places in telecommunications networks and critical infrastructure in order to meet the required accuracy, maintain holdover and for security requirements.

Oscillator Type	Accuracy	Cost
Quartz crystal (XO)	10^{-5} to 10^{-4}	inexpensive
Temperature compensated quartz crystal (TCXO)	10^{-6}	inexpensive
Oven controlled quartz crystal (OCXO)	10^{-8}	\$5 to \$100
Rubidium oscillator	10^{-9}	\$800
Chip scale atomic clock (CSAC)	10^{-10}	\$1500
Cesium oscillator	10^{-13} to 10^{-12}	\$50000

TABLE 1
Oscillator attributes

To meet these needs, further research will be required in the following areas:

- Increasing accuracy while reducing the cost, size, weight, and power for atomic frequency standards,
- Fundamental cost/performance limits for crystal and/or silicon oscillators over the range of expected operating conditions (temperature, humidity, pressure, voltage, vibration, aging),
- Evaluation of ensembles of low cost oscillators and or hybrid quartz-MEMs ensembles, and the development of hardware and software techniques for ensembling and compensation for varying operating conditions.

3.2 Time and frequency transfer

The transfer of time, phase and frequency all are degraded by the *delay* of the medium carrying the signal. There are two general methods: one-way transfer and two-way transfer. For accurate one-way time transfer, the total delay must be estimated and either physically removed from the received signal, or compensated for using a data technique. In two-way time and phase transfer a signal is sent in both directions from a reference clock to a client. If the delay is the same in both directions, it is possible to use the measurements associated with these two signals to remove the delay. The equality of these path delays is referred to as the symmetry of the two-way signals. Noise in both one-way and two-way systems essentially appears as variations in the transfer delay.

We focus here on time and frequency transfer for large numbers of users and commercial applications. Time and frequency transfer is generally accomplished using GNSS (particularly *Global Positioning Satellites* GPS), and time transfer across communications networks.

3.2.1 State of the art

GNSS has become the most common method of providing precise timing to users across the world. GPS is incredibly accurate, reliable, and inexpensive, resulting in many parts of industry becoming dependent on GPS for time and position. GPS accuracy depends on the quality of the quartz oscillator that determines the time required for averaging GPS signals. GPS receivers are unlikely to achieve better than 100 ns phase accuracy using low-cost quartz oscillators, but a well-designed receiver with an oscillator permitting integration times of $>$100s, can achieve better than 100 ns time and phase accuracy. A receiver using an oscillator permitting integration times of 24 hours can achieve $\pm$10 ns accuracy [10]. The time of all GNSS are derived from national laboratories and are therefore the de facto method of obtaining *coordinated universal time*, UTC, or *international atomic time*, TAI, time or frequency for any precision needs.

The great success of GPS also presents a problem. Unfortunately, the GPS signal is extremely weak and hence vulnerable to interference, either intentional or unintentional [11]. There also have been failure modes from the satellites or in the receivers. Thus, though GPS is now embedded in many systems for time and frequency transfer, there is an urgent need to back it up, or allow for appropriate holdover.

The second most common method of time transfer is across communications networks. Here we include both so-called wired networks (including optical fibers) and wireless networks.

Accurate time transfer to endpoints distributed throughout networks requires compensation, reduction or elimination of delay and *packet delay variation*, PDV, due to physical transmission over the media, media access protocols, switching, routing and queuing in and between the network elements. In networks, time transfer is typically achieved by using a two-way protocol, sending time stamps in both directions between the devices. This process cancels the delay to the extent that it is the same in both directions. The *asymmetry* of delay in networks is a major cause of inaccuracy in time transfer that can not be completely compensated by means of network protocols alone, but requires careful design to minimize at each network layer.

Despite these degradations, substantial efforts have been made to improve time transfer in *wide area networks*, WAN, and *local area networks*, LAN. The *network time protocol*, NTP, came into widespread use around 1985 as a wide area timing protocol thanks to its inventor David Mills [12], and today is implemented in essentially every PC in the world. NTP accuracy is limited by network and operating system PDV as well as the asymmetry of paths through the internet to a few milliseconds. NTP distributes UTC time derived from an ensemble of servers each synchronized to UTC, generally using either GPS or analog telephone lines to link directly to a national lab.

In recent years two additional network-based time distribution protocols designed primarily for LANs have been standardized. The first, SAE6802 [13], Deterministic or Time-Triggered Ethernet, is designed for safety-critical systems and has been implemented in several recently designed aircraft. SAE6802 achieves sub-microsecond accuracy while partitioning IEEE 802.3, i.e. Ethernet, bandwidth using time slots allocated to safety-critical and normal traffic thereby providing latency guarantees for safety-critical traffic and best effort for normal traffic.

The second major protocol is the *precision time protocol*, PTP, defined in standard IEEE 1588 [14]. PTP was designed for LANs but is increasingly being applied in WAN environments by the telecommunications industry where accuracies of $< 3\mu$s have been realized over multiple hops of optical fiber in a 40km long path in a live telecommunication network [15]. PTP specifies upgraded network bridges, so-called boundary and transparent clocks, that greatly, but not completely, compensate for symmetric, and asymmetric network delay and PDV due to queuing in the network elements. With these specialized bridges, IEEE 1588 can easily achieve LAN synchronization of 100 ns and with care <10 ns [16] [17]. In addition, recent work at CERN using a combination of IEEE 1588 and OSI network layer 1 frequency transfer has achieved 100 ps accuracy and 8 ps precision across three 5 km hops of fiber optic cable [18]. The major impediment to high accuracy in PTP (or any other two-way time-transfer protocol) is network asymmetry where the timing packet transmission latencies differ in the forward and backward directions. This is particularly pronounced in WAN environments where, unless care is taken, network routing can cause the forward and backward packets to take different routes through the network. In a LAN, bridge asymmetry is effectively eliminated by boundary or transparent clocks leaving uncorrected PHY[1] chip and link asymmetry as the principal impairments.

In wired networks, the state-of-the-art in network-based time transfer, e.g. NTP and PTP, relies on communication circuits that are mostly reliable yet somewhat difficult to physically access. As mobile computing and so-called connected embedded systems (popularly the Internet of Things) become dominant, the trend in per-communication-link quality will drop due to the increased use of wireless technologies with their inherent physical limitations. Moreover, IoT-style wireless networks are especially vulnerable to attack – at least in their present incarnation – because the physical links are readily "hacked" using off-the-shelf components (e.g., 802.15.4 radios that are built in to sub $10 microcontrollers). It would not be at all difficult for a malicious party to interfere with an IoT-instrumented factory or plant at the level of disrupting system time and device coordination, and the consequences could be disastrous.

1. A PHY chip forms the interface between the digital representation in a microprocessor and the signals on the communication medium

Moreover, the spectrum used by such devices – typically in the industrial, scientific and medical bands – does not have any monitoring or oversight per se. This stands in contrast to today's mobile networks that are operated under license. The licensee has the motivation to monitor for and deal with interferences in their spectrum.

3.2.2 Critical research issues

Except in a closed LAN environment it is unrealistic to expect application designers to implement a complete time distribution protocol. Rather designers should access time as a service provided by others. This service model has resulted in the widespread use of both GPS and NTP in commercial and industrial equipment. However both have their limitations. NTP, as widely implemented in servers, computers and laptops, is limited to millisecond accuracy.

NTP is not limited to wired networks. Smart phones use NTP for time-of-day alignment. The medium for transport of the NTP packets is invisible to the TCP/IP layers. The question is the available precision of the time-transfer mechanism and its demand on the underlying transport network. NTP with hardware time stamping can achieve the same performance as PTP. The issue is a lack of standardization for NTP hardware time-stamping methods. For example, Ericsson used a proprietary mechanism for hardware time stamping NTP to provide 3GPP with 50 ppb frequency accuracy before PTP was standardized.

GPS, while capable of high accuracy, is vulnerable to interference and is unreliable in indoor and some urban environments, e.g. concrete canyons. Obtaining accuracy and hold-over with GPS requires the use of expensive local oscillators. In addition to GPS and NTP, what is needed is the delivery of precision time, for example using PTP, as a service. This might be provided by the telecommunications providers at the WAN and wide area wireless level, by enterprises at the LAN and local wireless level, and by consumers and residences via Cable Modems, PON, XDSL and wireless access devices.

As noted, all current time distribution protocols have their vulnerabilities. The physics of wireless links and the threat of widespread interference poses new challenges. Given this, research aimed at resolving the ease of use, accuracy, reliability and security issues in time transfer is essential. This is particularly true for safety-critical applications. To meet these challenges additional research is needed in the following areas:

- Methods of authenticating GPS (and GNSS) signals, detecting and mitigating spoofing, and overcoming jamming
- Security technology for all relevant time transfer methods, such as verifiability and resilience of timing, particularly for safety-critical applications
- Optimal techniques for combining GNSS with wired and wireless network timing to support a wide variety of requirements, including accuracy, stability, security, reliability and GNSS availability
- Cost-effective and power-appropriate methods of transferring time and frequency through networks, both wired and wireless, as applications demand increasing accuracy in diverse environments, with emphasis on delivering *time as a service*.

3.3 Time-aware networks and communications

The *Open Systems Interconnection* (OSI) model [19] creates seven abstraction layers that remove implementation details of lower and upper layers from designers. This model is now used throughout communications networks with great success, as it allows designs that are implementation independent, *except for timing*. The lowest layer, the physical layer, is required for timing signals. Timing through a communications network can be no better than this layer, while the separation of layers worsens timing signals that rely on upper layers. As noted in section 1 network timing is currently seen as merely a performance metric, not a correctness criterion, thus timing accuracy must emerge from the details of the implementation: the network topology and the specific equipment used. Determinism of network delays and latency is necessary to enable the growth of time-aware applications including the Internet of Everything.

We break up this section into several subsections discussing time-awareness in networks and communications:

- network equipment hardware issues,
- time-aware and time-controlled networks,
- time support of latency control,
- performance monitoring and optimization,
- and wireless spectrum and bandwidth utilization.

3.3.1 Network equipment hardware issues

3.3.1.1 State of the art – network equipment hardware issues: This section discusses issues with networking equipment (NE) such as bridges and routers. Today's wide area communication systems are composed of a variety of technologies, e.g. SONET/SDH, ATM, microwave links, etc. all managed by the world's telecommunications companies. Recently carrier-Ethernet has begun to displace this infrastructure due to perceived cost advantages primarily of the NE. However these companies are still expected to deliver some traditional and newer synchronous services, e.g. fax, streaming video. This is reflected in the design of NE where packet buffers mediate differences in transmission and reception rates and where needed timestamps or adaptive techniques are used to reconstruct source timing at the receivers and packet routing protocols have replaced circuit switched communications.

However, as discussed earlier, timing needs for wireless protocols operating at the local area have radically increased the importance of telecommunications operators providing accurate time throughout the system. This is reflected in standards activities governing NE, e.g. ITU-T recommendations G.8261, G.8262, G.8264, and G.8271 [20]. This work is well underway; network equipment following these standards is available and, as noted earlier in the case of China Mobile and others, rapidly being deployed.

Similar changes are occurring at the LAN level where bridges and end devices incorporating IEEE 1588 to provide accurate local timing are available and are being deployed[2].

3.3.1.2 Critical research issues – network equipment hardware issues: Networking equipment contains data and control parts. Today, timing for these parts is not well linked. Moreover, different parts of a data path use different clocks, with the control path running on low quality oscillators, and often with each CPU having its own independent oscillator. This effectively creates a need for controlled transitions, creating protocols that need to allow various delays in the systems, to support the exchange of acks and requests, etc. If the clocks were better synchronized, there would be a reduction of WAIT states, since data exchanges could be done on a schedule. Delays caused by the crossing of a clock domain would be eliminated, and protocols would be simplified and work in a time coordinated fashion.

We note here that time triggered systems still incorporate WAIT states, albeit primarily for enabling provably correct systems rather than efficiency [21]. There is also a large community that advocates asynchronous protocols rather than time-based protocols [22] [23].

That said, there are promising areas of research including:

- Design requirements for a global clock domain including phase alignment of the physical clock in each component,
- The need to distribute the clock within equipment using buffers with adjustable delays, including negative delays. The board clock needs to be distributed accurately to compensated for delays.
- To evolve change slowly, the time-driven design of boards and network elements' hardware would be the first step. Component variability compensation (similar to what is done in CERN's White Rabbit project [18] but on a larger scale [3]) would be the next step. The CERN project is a significant

2. Discussions of many of these devices and deployments can be found in the proceedings of the ISPCS conferences available from the IEEE Xplore digital library. Note also that IEEE 802.1AS, which contains a profile of IEEE 1588, was designed to serve the timing needs of the audio/video community.

3. The CERN White Rabbit technology is scheduled to be included in the revision of IEEE 1588-2008 currently in progress

step in showing what can be achieved with hand-adjusted delays. While it is not practical in mass production yet, there are many lessons that could be learned from the design.

3.3.2 Time-aware and time-controlled networks

3.3.2.1 State of the art – time-aware and time-controlled networks: In today's networks the primary use of global time and frequency is related to operation of the cellular systems and to emulating existing services as discussed earlier. At the LAN level, safety-critical applications such as aircraft controls are managed using time-triggered architectures that for networks imply a time division multiple access (TDMA) protocol. SAE 6802 is an example of a standard providing this capability [13]. Another example is time-triggered Ethernet [24], [25], an extension to the Ethernet standard that supports real-time data communication. This TDMA-based approach was originally an academic effort [24], but is now also developed in industry [25].

The Audio Video Bridging (AVB) task group within IEEE 802.1 recently completed a suite of standards that synchronize time and bound network latency for packets associated with an accepted reservation, allowing time-sensitive streams to coexist with time-insensitive networking without degrading the performance of either. Building on these standards, the task group recently expanded its scope and changed its name to Time Sensitive Networking (TSN) [26]. With the planned enhancements, future networks will provide extremely low latency (few microseconds or less) by quiescing links at the right time, allowing time-sensitive packets to pass through uninhibited by queues and unimpeded by other traffic. For such a system to operate effectively, the network infrastructure must agree on both a timebase, and a schedule for every set of time-sensitive paths. However it is an open question of whether these efforts can be made robust enough to enable life/safety applications such as tele-surgery that depend on provably robust delay bounds and timing across the application.

The rapid adoption of cloud-based computing and network visualization raises additional complications when the applications are time sensitive. Synchronizing data bases, linking commit order, distributed applications, and other issues will require different levels of timing accuracy and integrity.

3.3.2.2 Critical research issues– time-aware and time-controlled networks: Research projects here include:

- Move toward a fully deterministic network with absolute time alignment
  - Interworking from individual components of a clock tree to network level protocols
  - New protocols and features, with increased security
- Strategize implementing total synchronization given legacy interfaces
  - Identify where the precise time is most useful at different levels of the system
  - Separate accurate time domains and run time alignment between them,
  - Extend to wider areas
- Use time to augment routing protocols. Today's routing protocols are designed for addressing and to average throughput efficiency. Time sensitivity of traffic is reflected in various priority mechanisms. Other mechanisms such as earliest-deadline-first routing need to be investigated
- Approach absolute latency guarantees. Access can be controlled as is done in LANs operating using a time scheduled protocol. Research extending these techniques to wider areas is needed
- Explore the use of computing resources within network equipment for something intermediate between cloud and strictly local computation

3.3.3 Time support of latency control

3.3.3.1 State of the art – time support of latency control: Latency control is essential for many real-time applications including live audio/video, factory automation/control, and financial transactions.

Tele-surgery, live audio/video and computer control of a manufacturing line require bounded latency, while financial transactions require fixed latency for fairness.

There is a strong trend in the professional audio/video industry to move toward the use of standard networks instead of analog or special-purpose interfaces in the transport of time-sensitive media, for reasons of flexibility, scalability, and cost. But if digitized sound from the stage at a concert is delayed too much by the network on its way to the mixer, an audio processor, and then to the speakers, the resulting comb filtering is distracting to the performers and degrades the resulting experience. Audio or video samples that arrive before the deadline cause no problem if an independent time reference is available, but if the latency exceeds a deadline, that media data cannot be used. In this case, use of accurate time is used to control the phase of the rendering. The use of time for bounding latency in industrial control similarly requires accurate, shared time in order to apply actuator values simultaneously as part of a machine control loop.

Consider financial transaction systems. In order to achieve latency control, they must first have the ability to measure the latency. This requires that globally accurate time be distributed to all of the network and computational elements. Financial traders, brokers, and exchanges are collaborating to reduce the latency of transactions within the FIX Inter-party Latency Working Group (FIX IPL)[4]. FIX IPL has defined mandatory and recommended/optional latency monitoring points. These points/times vary from when a transaction was initiated at the seller/buyer in the CPU/Server of origin, to the time the transaction was placed on the internal network of the seller/buyer, when the transaction was placed on the external network, the arrival at the broker, processing/aggregation by the broker, when sent on to the exchange through the brokers network, then through an external network to the exchange, and through the various switches and routers in the exchange and eventually through to the exchange server where the trade is executed. Similarly, there are considerations to measure the latency of execution completion back to the seller/buyer in addition to stock prices via multicast feeds. In order to achieve minimization of delays in this process, it is first necessary to distribute timing, then to accurately timestamp events and measure time intervals among the various process elements and then to reduce the delays via design, architecture or process. Today's state of the art is limited to hundreds of microseconds in global accuracy and tens of microseconds in resolution, whereas the process steps themselves may be sub-microsecond in duration. The result is apparent non-causality of transactions, e.g., a transaction that due to mistimed systems appears to have been completed before the transaction was placed.

Next consider the fairness issue in connection with transaction processing and information delivery. Financial exchanges are required by the SEC to provide fair and equal access/processing in the exchange. Among those parties paying for a certain tier of service, it is required to deliver a *service level agreement*, SLA, that is not just fast, but also fair in terms of latency to the extent possible. For instance, two brokers/banks co-located in the NYSE paying the same price for a service tier expect that their trades will be executed faster than those paying a lower tier, but the exchange must provide a fairness of the trade execution among those at the same service level. This means that the exchange must meet an SLA of a certain latency with a bounded jitter (latency variation). This is a much harder problem to solve than the telecommunications service provider that is just trying to provide service below a certain maximal delay. Not only must the exchange be able to measure the delay of the transactions, they must act, at a packet level, to queue or hold traffic such that the transaction completion times and/or network transit times meet an acceptable probability distribution (mean and jitter). Today's switches and routers are not designed to support forwarding packets at a constant or at least a stationary mean with bounded jitter. Today's routers typically forward packets as soon as they can as opposed to in a time-wise fair manner.

A case in point is multicast traffic, such as price-quote information that is multicast to a set of clients subscribing to a particular stream of trading information, a "feed". Many routers suffer from the "multicast diameter" issue whereby certain router ports have different delays from the incoming port. These routers are extremely problematical for the Financial exchange provider that must provide latency fairness. The fairness

4. Information on FIX IPL can be found at http://www.fixprotocol.org/working_groups/

requirement requires time transfer to the network elements, measurement/calibration/compensation of inter-port delays and potentially deep packet inspection to identify the customer for the purpose of monitoring inter-customer fairness. There may exist queuing processes that can inherently equalize the delays without inherent knowledge of time, but in all cases the transit times must be measured and validated as per the SLA. In addition, it is a common occurrence for many systems to respond to a price change (as received in a multicast feed), resulting in a flurry of transactions queued at nearly the same time. The resulting burst of traffic may be of extremely high volume but last for a very short time, of a millisecond or less, and is called a *microburst*. If one customer has a consistent (though inadvertent) latency edge then it can gain an unfair advantage over the other customers by being the system that is on the front edge, or potentially being the intentional or unintentional cause of the microburst. The financial exchange therefore has the task to find an optimally small latency while minimizing jitter and still providing fairness under all ranges of network loading.

3.3.3.2 Critical research issues – time support of latency control: There are fundamental enhancements that must be made to the network elements that transport data including:

- Use of techniques including cut-through switching and time-sensitive switching to bound latency
- Development of new time-sensitive routing algorithms and scheduling techniques for TSN
- Switch/router/network architectures that measure intra- and inter-element latency and jitter
- Techniques to reduce/control multicast diameter
- Queuing techniques to mitigate and/or de-synchronize microbursts
- Techniques to manage the conflicting requirements of bounded latency and fair distribution of latency

3.3.4 Performance monitoring and optimization

3.3.4.1 State of the art – performance monitoring and optimization: Timing is essential for systems performance monitoring and performance optimization. Service providers are required to meet or exceed *service level agreements* (SLAs) that they have contracted to provide for their customers. The SLA specifies *key performance indicators* (KPIs) that are used to measure and prove performance to the agreement. KPIs always include measurements of throughput, delay (latency), jitter (variation of latency), and loss, all of which are reported as a function of time or over a given time period. In addition, KPIs include *mean time to repair* (MTTR) service in the event of an outage or degradation. MTTR is fundamentally dependent upon *root cause analysis* (RCA) and RCA is based upon an operator's ability to pinpoint the time of an event, the events leading up to the failure event and the events that follow the failure. This process is greatly enhanced when the time of events is very well known. Poor timing results in event sequences that appear "non-causal".

Due in large part to the relatively poor timing for the monitoring and reporting systems in most of today's network equipment, RCA systems require a combination of heuristics and human intelligence to logically arrange the alarms and events according to possible or likely time lines. This costly and time-consuming process of event resequencing is obviated if the timing of all of the monitoring systems is globally accurate. Timing accuracy therefore results in a great reduction in MTTR, better quality of service, operational efficiency, etc. Today, service providers are spending billions of dollars on network probes, data storage, data base systems and analytics platforms to monitor, measure and extract valuable information to operate their networks more efficiently. All of these systems can benefit from better timing than has been available to date.

Todays network probes use timing as a basis for their measurement, for instance Internet Control Message Protocol (ICMP) PING [27], Two-Way Active Measurement Protocol (TWAMP) [28], One-Way Active Measurement Protocol (OWAMP) [29], IETF RFC2544 [30] and Y.1731 [31]. These protocols rely

on timing accuracy that typically is derived from IEEE1588-2008 PTPv2 and/or NTP with or without Synchronous Ethernet.[5]

In many cases the measurements are self-referential, e.g., TWAMP and PING, and the measurement results from one pair of devices cannot be precisely correlated in time with other devices in the network. This is especially the case when NTP timing is used or if the underlying packet transport network is highly asymmetrical as in DOCSIS or PPN. Time offsets between these systems are often in the milliseconds, whereas the communication processes, and indeed the impairments in the processes that need to be monitored, may exist for hundreds of microseconds or less, a potential mismatch of four orders of magnitude! It is therefore necessary to distribute time, to timestamp events and to process these events with far better accuracy than has been available to date.

3.3.4.2 Critical research issues – performance monitoring and optimization: Future networks will be carrying much more traffic and significant portions will be time sensitive, e.g. tele-surgery, smart transportation systems, financial systems. Both latency and determinism will be critical in some of these applications. As noted, users will demand guarantees of key metrics. To meet these needs additional research is required in the following areas:

- Correlation of globally synchronized statistics from point-to-point active protocols, e.g. (PING, TWAMP, etc)
- Time of flight measurement and correlation of data packets that are passively or actively monitored, tapped or port mirrored at more than one observation point
- Development of monitoring probes, optimum deployment patterns to ensure coverage across a distributed network
- Verifying time accuracy and traceability
- The use of enhanced timing accuracy for root cause analysis, and to improve mean time to repair

3.3.5 Wireless spectrum and bandwidth utilization

3.3.5.1 State of the art – wireless spectrum and bandwidth utilization: It is clear that the demand for wireless bandwidth will only continue to grow with mobile phone usage (voice, applications) and with the widespread use of wireless communications for IoT devices (*e.g.*, IEEE 802.15.4 and the protocol families that ride atop it such as ZigBee and WirelessHART). Increasing bandwidth demand within fixed spectrum allocations necessitates increasingly efficient usage of that spectrum. Part of this will come from improved channel coding, up to the Shannon limit (with techniques such as spinal codes [32]). But attention must also be focused on bandwidth lost to poor synchronization that could be harvested with better time synchronization between the users of a given channel. TDMA techniques [33], [34] depend critically on maintaining accurate timing within clusters of IoT devices. These, in turn, depend on the accuracy and drift characteristics of local clocks (often derived from inexpensive quartz oscillators). Improvements in low-power, highly-accurate timekeeping methods in IoT-class devices will translate directly into bandwidth savings and channel efficiency.

In addition to mobile phone usage, there is an increasing demand for lower delay (for real-time interactive communications) augmenting the need for spectrum usage efficiency. It is well known that the performance of wireless communications systems is fundamentally dependent upon phase and frequency [35]. The physical layer provides the best accuracy of time distribution, since timing is fundamentally a physical signal. But the time is needed at the different levels of the node and has to be propagated to them. Among the wireless technologies with the most stringent requirements are CDMA, WCDMA, WiMAX, DVB-H, LTE and variants of Multimedia Broadcast Multicast Service (MBMS) and Coordinated Multi-Point (CoMP). These technologies require accurate inter-tower time synchronization ranging from 10 microseconds down to 100 ns, and frequency synchronization to 50 ppb. Today's state of the art

5. Synchronous Ethernet is the common name for ITU-T Recommendations G.8261, G.8262, and G.8264. These protocols result in frequency transfer at OSI layer 1 and leave questions of phase and time to higher layer protocols, e.g. NTP, PTP

basestation systems are marginally capable of providing such accuracy but suffer from reliability issues (GPS) or network asymmetry (IEEE 1588). Still, telecommunications operators such as China Mobile [15] are aggressively installing these networks.

Better timing available at handsets, femtocells, small cells and macrocells will permit better spectrum utilization via reduced guardbands, CoMP, and denser signal constellations. This in turn permits more users, and more bandwidth per user. There is also the possibility of reduced power and thus energy savings or battery charge longevity.

Similar issues arise for coaxial (DOCSIS) and fiber (G-PON) transmission media. Today these systems use round trip delays to measure the distance and transport time to the PHY and MAC layers, but not to the overlying layers. Accurate timing can reduce the guard band dead times, improving efficiency (bandwidth utilization) of the channel.

3.3.5.2 Critical research issues – wireless spectrum and bandwidth utilization: Needed research topics include:

- Hybrid solutions for better timing of basestations, and handsets
- Packet timing distribution over DOCSIS/GPON with transparent clocks or the equivalent[6]
- Guardband reduction for better spectrum use
- Bandwidth improvement by coordinating multiple Wi-Fi access points

3.4 Timing support for applications

A central tenet of the great progress we have made in computers and networks is due to the ability to design with abstractions that can be realized in hardware. Currently, programming languages and design tools can implement abstractions with great repeatability in terms of function and speed of performance, but the clock speeds and timing accuracy are deliberately discarded. Timing is a performance metric, not a correctness criterion. Timing is implementation dependent. Manufacturers have to stockpile parts to last the lifetime of the product if it has time-sensitive features. We need a semantics of time in software with hardware support that will ensure repeatability of timing in diverse hardware. Determinism at the interface between software and the real-world is necessary to ensure repeatable behavior that is implementation independent.

In this section we examine the state of the art and critical issues for several aspects of timing support for applications: predicable execution times, timing interface hardware support, and issues of scale.

3.4.1 Predictable execution in time-aware systems

3.4.1.1 State of the art-predictable execution: Computing hardware, generally, has evolved toward delivering raw performance and has compromised its suitability as elements in time-based systems. Caches, speculative execution, and other mechanisms that improve aggregate performance in modern processors significantly degrade predictability and, hence, the applicability of these processors for high-precision real-time computation. We believe there is an opportunity to revisit some of these architectural decisions and further develop computational models that are, at their heart, time-respecting. Early work in adapting data flow and multithreaded architectures [37] to real-time systems [38] shows promise but is under-developed. Modern FPGA technology provides the power and flexibility for experimentation and development of these and other new concepts [39].

Timing analysis and application design can be simplified if the underlying hardware has predictable timing behavior, i.e. the size of the machine state is limited. The idea of precision timed machines (PRET) was first advocated by Edwards and Lee [40] in 2007. This idea has then been further extended to include both precision timed compiles and languages, sometime referred to as a PRET infrastructure [41]. Of particular interest here are PRET machines that augment the instruction set architecture (ISA) with timing instructions [42], [43].

6. IEEE 802.1AS, which contains a profile of IEEE 1588, includes specifications for operating over GPON and 802.11 / Wi-Fi [36]

A common way for PRET machines to achieve repeatability is to remove pipeline hazards by means of thread-interleaved pipelines [44] and by replacing caches with scratchpad memories [45]. The latter approach moves the memory management problem from hardware to software, thus introducing the need for time-aware scratchpad allocation techniques [46].

Other predictable processor designs include a reactive processor based on the Xilinxs MicroBlaze processor, [47], a processor designed for executing synchronous programs [48], a predictable Java processor JOP [49], and the XMOS architecture [50].

Predictable hardware platforms may enable more scalable and precise worst-case execution (WCET) analysis. WCET analysis [51] is a fundamental part of hard real-time systems development, that is, development of safety critical systems where the execution time of tasks must be guaranteed to meet deadlines. Static WCET analysis techniques [52] consist typically of three main steps: program flow analysis (finding loop bounds [53] and infeasible paths [54]), microarchitectural analysis, and global bound analysis [55]. There exist techniques to perform microarchitectural analysis of caches [56] and hardware solutions designed for special instruction caches [57], but these techniques cannot today scale to complex modern hardware platforms.

Most research on WCET analysis focuses on processors with one core; analysis of multi-core systems is even harder due to resource sharing. Research on WCET analysis for multicore is currently an active area of research [58], [59].

There exists also work where WCET analysis is part of the compiler tool chain [60], with the purpose of optimizing and minimizing the WCET instead of average case execution time. Few attempts have so far been made to make WCET analysis interactive [61], meaning that the designer is iteratively improving the WCET by updating the design of the program or model.

3.4.1.2 Critical research issues– predictable execution: With such proofs-of-concept in hand, the key remaining challenge will be to motivate adoption of these concepts for time-predictability into mainstream processors. This implies the need for fundamental research in the following areas:

- Hardware mechanisms of time-aware, time-preserving, and timing-predictable computation, starting from a deep understanding of present-day architectures
- Mechanisms that are minimally invasive and, therefore, maximally adoptable by the mainstream
- New or extended programming and modeling languages where time is a first class citizen
- New algorithms and methods in compilers and runtime systems that can take advantages of time-aware hardware mechanisms and languages with timing semantics. In particular, scalable WCET analysis techniques and tool chains that treat timing as a correctness factor are needed.

3.4.2 Timing interface hardware support

3.4.2.1 State of the art-timing interface hardware support: There are four time-critical interfaces between the clock and other features of a time-aware device. These interfaces exist at sensors, actuators, network input and output, and to microprocessor code execution.

- Sensor and actuator interfaces: In most current devices driven directly from firmware the response timing precision is limited by interrupt latency. If tighter timing is required then typically the design will be implemented in hardware, often an FPGA. Indeed it is not uncommon that all critical timing is implemented in an FPGA. An alternative and promising approach is embodied in the design of the Texas Instruments DP83640 [62]. This chip includes an Ethernet PHY, a hardware clock that can be synchronized to its network peers, for example using IEEE 1588, and most importantly, registers that can either capture the value of clock in response to a hardware signal (timestamping) or can generate a hardware signal when the time of the clock matches a timestamp written to a register. This chip allows control of sensor and actuator events to an accuracy of 8ns.
- Network interfaces: The timing of these interfaces is typically asynchronous and can depend on network traffic patterns, local operating system loads and of course on the details of the

communication protocol. The network interfaces used in safety-critical systems typically use a time triggered approach, e.g. SAE6802 [13] which can provide a high degree of determinism but scale poorly to large systems. New Ethernet controllers are becoming available that provide PTP timestamping and hardware event capabilities to commercial off-the-shelf computers [63] but transfering time within the system between various I/O and computational units is lacking.

- Code execution: Except in a pure hardware solution there is interaction between supporting hardware and code executing in a microprocessor. Operating systems typically provide timers based on the microprocessor clock, which in general is not synchronized to its networked peers. Furthermore the latency and accuracy of timer-based interactions is limited by interrupt and other operating system latency and jitter which limits timing precision to microseconds at best and often much worse. Likewise the timing of communications between executing software and the network interfaces and even to devices like the DP83640 is limited by these same factors.
- System interconnects: Recent additions of time to buses such as PCI Express will give hardware support to time-transfer within large systems [64] but the effective use of such features by applications requires further work.

In addition the use of multi-core and multi-CPU designs introduces timing dependent on the distribution of execution between cores or CPUs. These timing dependencies are often non-deterministic and thus difficult to analyze and control.

3.4.2.2 Critical research issues– timing interface hardware support: Needed research in this area includes:

- Support architectures inspired by the DP83640 but with deterministic timing for executing code
- Standardization efforts at the interfaces between sensors and actuators, and between the synchronized clocks and code execution
- Scalable network interfaces with predictable timing
- Low latency support for code execution, e.g. timers and for newer architectures such as Ptides (discussed briefly in Section 3.5.1)
- Develop multi-CPU, and multi-core CPUs with a strict notion of time
- Research is needed to understand the accuracy with which time can be transferred to all threads and cores and the accuracy with which a process can time an event and measure time between events
- Do all of this without slowing progress toward lower power and longer battery life

3.4.3 Issues of scale in time-aware systems

3.4.3.1 State of the art-issues of scale: The Internet of Things represents a fifth generation of computing platforms. It is distinguished from the previous generations by scale (within the scope under consideration here, a conservative estimate is that 10x as many IoT devices will be installed annually as there will be mobile phones sold annually). Issues of scale lead naturally to an only handle it once (OHIO) approach to systems architecture and deployment, and the substantial costs of installing large-scale wired networks (or even providing wired power to such devices) has motivated substantial research over the last 15 years into wireless sensor networks (WSN), primarily aimed at battery-operated (or energy-harvesting [65]) devices. There are a number of Time Synchronization Protocols in WSN. A summary of some of these and a proposal for a new one is by Shannon et. al. [66].

Under the assumptions of constrained power and imperfect wireless networking technologies, IoT devices must be thought of as functioning in a semi-connected fashion in peer-to-peer networks and/or in hierarchical networks connecting to cloud services. Lacking strong guarantees of connectedness, IoT devices providing cyber-physical interfaces (sensing the environment and, through various control processes, actuating devices in the physical world), must necessarily have local access (in-SoC, on-board) to accurate time information so that sensed (input) data can be properly time-labeled for later cross-network correlation. And for those IoT devices that provide actuation, correct local time is essential to carrying out

pre-scheduled physical actions, despite the vagaries of the wireless networks that tie these systems together [67] [68].

For example the Samsung Galaxy S4 smart phone reportedly incorporates an accelerometer, a gyroscope, a hygrometer, a magnetometer, a light sensor, a thermometer, and a barometer as well as a camera. However access to these as components in the IoT is based on current individual ownership of the connection and with timing limited by either the cellular system or GPS with limited holdover.

As discussed previously, IoT devices are especially vulnerable to attack in the present incarnation, both because the radio links can be hacked, and because the spectrum is largely unmonitored.

3.4.3.2 Critical research issues– issues of scale: Issues of scale in time-aware systems suggest the following critical research areas:

- The creation of ultra-low-power, stable local time references
- Robustly maintain cross-IoT-network synchronization in the presence of semi-connectedness
- IoT security, including verifiability of the timing signal
- Distributed computing issues: latency, distribution and optimization of computation locality, communication and storage
- Provide *time as a service*

3.5 Development environments

Though there are powerful tools available for system development, simulation and code generation, incorporating precise timing and deterministic functioning will require a fundamental shift in the design of these tools and their relationship to fundamental physical levels of hardware.

3.5.1 State of the art

There are many design environments supporting various aspects of system design, e.g. Simulink, LabView, Rational, Step7, Modelica, and Ptolemy[7]. They are reasonably sophisticated in terms of simulating system behavior given the proper models. They are much weaker when used to specify system behavior, especially timing behavior, and then generate executables that map the design to a range of commercial systems with varying degrees of support for timing interface hardware (see 3.4.2 above). There exist some attempts to include time as part of programming language semantics, for instance Real-time concurrent C [69] and Real-time Euclid [70], but none of these languages have compilers and tool chains that are widely used in industry. There are few if any tools for adequately visualizing, analyzing, and verifying timing properties.

We need abstractions that can model a system's timing accurately and that can then be translated into implementations that faithfully reproduce the timing design. Such systems would then allow for design abstractions to include timing that would lead to implementation independence.

An example of a development environment that addresses these issues is the Ptides environment, which currently operates only on the Ptolemy framework [71] [72]. This environment allows the user to specify timing within a GUI along with more common development elements. Ptides has been used to produce deterministic timing when compiled and applied to varied hardware. Ptides represents a start toward including time accuracy along with the complex features of modern development environments.

3.5.2 Critical research issues

Key research issues include:

- Design environments that specify the design of system timing independent of the target execution platform
- Develop suitable execution platform support to realize this timing
- Models, languages, design interfaces that allow specification of timing as a first-class artifact
- Analysis and visualization tools to aid the designer in timing specification and reasoning about time.

7. Simulink, LabView, Rational, and Step7 are products respectively of the MathWorks®, National Instruments, IBM, and Siemens. Ptolemy is a project at the University of California, Berkeley.

3.6 Time-aware applications

This section discusses current examples of application domains that incorporate at least some time-aware features or would be enhanced if they did. A number of years ago Barbara Liskov noted that NTP spurred interest in using global time to improve mainstream computer science algorithms and protocols [73]. It is time to revisit this question in the context of advances in global time capability.

3.6.1 State of the art

The design of local safety-critical systems using time-triggered architectures is reasonably well understood [21]. However these techniques are brittle with respect to scaling and at least in their present form are not suitable for the IoT or other large scale applications or for a system of systems. Machine-to-machine (M2M) technologies, expected to grow rapidly as part of the IoT, will be a major consumer of synchronization [3]. As an illustrative example, Cisco talks of "Smart and Connected Passenger Vehicles" [74]. They envision four different connection capabilities in a vehice: 1) within the car, 2) to personal devices, 3) around the car, and 4) to the cloud. Each of these capabilities has aspects that require levels of synchronization to reach their full potential. Current synchronization technologies will need more precision to fully enable various safety and optimization functions. Better integration of timing with computers and networks is essential to facilitate such technologies.

A number of applications require new research to support their use of precise timing. An example of the early applications of time-aware techniques is the time-synchronized and distributed rendering of audio over standard networks. These include implementations of the IEEE 802.1AS profile of IEEE 1588 discussed earlier, [36], over Ethernet for professional audio installations in theaters, concert halls, airports, hospitals, and even homes. This recent industry direction has begun to provide a foundation of commercial off-the-shelf devices on which research into some of the new areas may be performed[8]. Similar applications have appeared in industrial control particularly for motion control, e.g. packaging machines [75].

However in many ways the Google Spanner project is the most interesting from the view of incorporating global time as an integral part of applications. Following Liskov's dictum, Google used an NTP-based global time service to "Transform commit order reasoning to timestamp order reasoning" [76].

Expanding on the example of synchronized audio, there are a number of application areas that would benefit from increasingly network-capable devices sharing physical proximity — if they are synchronized. For example, a group of microphone-equipped cell phones could become an *ad hoc* microphone array [77]; or a phased array wireless communication system (known as cooperative diversity) [78] for improved range, reduced interference, or perhaps lower total power. Actively adaptive A/V media distribution among synchronized home-networked devices has enormous possibility but would benefit from further research. Indeed, when a mobile device interacts with the physical world, whether through acoustic or electromagnetic signals, acceleration or through stimulating any of the other multitude sensors being integrated into smart devices, these devices could act in concert in new interesting ways when such interactions with the physical word are time-coordinated or time-correlated.

Devices operating behind the scenes are not the only type for which precise time and timing is important. Even consumer-oriented devices such as personal computers, tablets, and smart phones interact with one another and also with the physical world as computers become increasingly "social". Historically, such devices communicated "data" to large servers, communicating between themselves only through the "cloud", but peer-to-peer networking is becoming increasingly collaborative in ways that requires accurate time (see Wi-Fi Direct, for example) — and when two devices share proximity in space, they could interact with the physical world in new ways if (when) they also share precise time. But these applications will require research in areas of *ad hoc* cooperative diversity, cooperative noise suppression, dynamic acoustic source localization and tracking, irregular array microphone beam-forming algorithms. And these

8. AVnu Alliance promotes the AVB standards and provides certification of interoperability, see http://www.avnu.org

algorithms must be practical in even a handheld device without sacrificing battery life or adding excessive cost.

3.6.2 Critical research issues

Until adequate design environments are available, creating innovative time-aware applications will be a hand-crafted exercise. At the same time realistic examples, such as Spanner and local techniques for ensembling sensor data, will do much to direct efforts in design environments and the other needs discussed in earlier sections.

It is therefore critically important that research efforts be made to:

- Examine high-value existing applications for potential improvements using time-aware techniques (e.g. what Google did on Spanner)
- Consider the implication of timing on the IoT
  - Coordination
  - Location
  - Determinism
- Produce compelling new or enhanced applications using more accurate, stable, and/or reliable timing.

4 Conclusions

We have attempted to outline the state of the art and critical research areas in the area of time-aware systems. As we have seen with the exception of safety-critical systems, the area is still in its infancy, yet holds great potential to help enable or improve many compelling existing applications, e.g. intelligent transportation, financial industry, and telecommunications. Synchronization research will be needed for longer-term developments such as the IoT.

It is also apparent, for example by analyzing the bibliography, that the researchers and developers of the various technologies described in our layered model of time-aware systems operate pretty much independently of each other. They attend different conferences, read different literature, and in general do not interact sufficiently to achieve the breakthroughs needed. In our minds this calls for a dedicated and collaborative "across the stack" research collaboration focused on two or three comprehensive challenge problems.

Finally, we summarize again the critical research that we have outlined.

- *Oscillators* in the network will require a range of trade-offs among performance, power and cost, as well as ensembling methods, that challenge the state-of-the art,
- *Time Transfer Systems* will need to deliver signals to an exponential increase in endpoints, with various specifications of accuracy and integrity
- *Time-Aware Networks* will need development in a number of areas:
  - Network hardware and software will need new designs to support and make use of time-awareness
  - Development of time-aware and time-controlled networks requires research in both propagating and using timing signals
  - Controlling latency in networks is crucial to many real-time applications
  - Performance monitoring is required for maintenance and service-level agreements, and is greatly enhanced by synchronization
  - Precision timing optimizes use of precious spectrum
- *Timing Support for Applications* will need cross-displine research in the following areas:

  - Hardware and software support of predictable execution will need to balance the depth of change in systems with cost and implementation
  - Timing across interfaces will require standards and latency control both among CPUs and in crossing network domains
  - Scale issues need addressing in supplying time to large numbers of systems
- *Development Environments* will need to evolve to support timing accuracy independent of the hardware that systems are running on
- *Applications* can make innovative use of time, and will further stimulate the development of these other items.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's intranet of things to a future internet of things: a wireless-and mobility-related view," *Wireless Communications, IEEE*, vol. 17, no. 6, pp. 44–51, 2010.

[3] J. O. Smith, "Standardization in machine-to-machine (M2M) communications," 2012. [Online]. Available: http://connectedvehicle.itsa.wikispaces.net/file/view/ITSA_Webinar_February_15_2012_Jeff_Smith_Numerex_GSCMSTF.pdf

[4] J. Bradley, J. Barbier, and D. Handler, "Embracing the internet of everything to capture your share of $14.4 trillion," 2013. [Online]. Available: http://www.cisco.com/web/about/ac79/docs/innov/IoE_Economy.pdf

[5] P. C. Evans and M. Annunziata, "Industrial internet: Pushing the boundaries of minds and machines," 2012. [Online]. Available: http://www.ge.com/docs/chapters/Industrial_Internet.pdf

[6] NIST, "Cyber-physical systems public working group," 2014. [Online]. Available: http://www.cpspwg.org/

[7] J. Vig, "Quartz crystal resonators and oscillators for frequency control and timing applications - a tutorial," April, 2012. [Online]. Available: http://www.ieee-uffc.org/frequency-control/learning-vig-tut.asp

[8] Rakon, "RFPO40 SMD Oven Controlled Crystal Oscillator," 2013. [Online]. Available: http://www.rakon.com/products/families/ocxo-ocso

[9] Symmetricom, "SA.45s CSAC," 2013. [Online]. Available: http://www.symmetricom.com/products/frequency-references/

[10] M. Weiss, "One way GPS time transfer," 2013. [Online]. Available: http://tf.nist.gov/time/oneway.htm

[11] K. Wesson, D. Shepard, and T. Humphreys, "Straight talk on anti-spoofing," *GPS World*, 2012.

[12] D. L. Mills, "A brief history of NTP time: confessions of an internet timekeeper," *ACM Computer Communications Review*, vol. 33, 2003.

[13] As-2d2 Deterministic Ethernet And Unified Networking, "AS6802 Time-Triggered Ethernet," SAE, Standard Specification, November 1, 2011.

[14] IEEE Instrumentation and Measurement Society, "1588: IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," IEEE, Standard Specification, July 24, 2008.

[15] M. Ouellette, K. Ji, S. Liu, and H. Li, "Using IEEE 1588 and boundary clocks for clock synchronization in telecom networks," *Communications Magazine, IEEE*, vol. 49, no. 2, pp. 164–171, 2011.

[16] D. Vook, B. Hamilton, A. Fernandez, J. Burch, and V. Srikantam, "Update on high precision time synchronization," in *Proceedings of the 2005 Conference on IEEE-1588*, Zurich, CH, 2005.

[17] D. Mohl and M. Renz, "Improved synchronization behavior in highly cascaded networks," in *International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Vienna, AT, 2007.

[18] M. Lipinski, T. Wlostowski, J. Serrano, P. Alvarez, J. David, G. Cobas, A. Rubini, and P. Moreira, "Performance results of the first white rabbit installation for cngs," in *International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, San Francisco, CA, 2012.

[19] International Organization for Standardization and International Electrotechnical Commission and others, "ISO/IEC 7498-1 Information Technology - Open Systems Interconnection - Basic Reference Model," ISO/IEC, Standard Specification, 1994.

[20] J.-L. Ferrant, M. Gilson, M. Mayer, L. Montini, M. Ouellette, S. Rodrigues, S. Ruffini *et al.*, *Synchronous Ethernet and IEEE 1588 in Telecoms: Next Generation Synchronization Networks*. John Wiley & Sons, 2013.

[21] H. Kopetz, *Real-Time Systems : Design Principles for Distributed Embedded Applications*. Springer, 1997.

[22] I. Sutherland, "The tyranny of the clock," *Commun. ACM*, vol. 55, no. 10, pp. 35–36, Oct. 2012. [Online]. Available: http://doi.acm.org/10.1145/2347736.2347749

[23] L. Sha, A. Al-Nayeem, M. Sun, J. Meseguer, and P. Ölveczky, "Pals: Physically asynchronous logically synchronous systems," *University of Illinois at Urbana-Champaign, http://hdl. handle. net/2142/11897, Tech. Rep*, 2009.

[24] H. Kopetz, A. Ademaj, P. Grillinger, and K. Steinhammer, "The Time-Triggered Ethernet (TTE) Design," in *Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*. IEEE, 2005, pp. 22–33.

[25] W. Steiner, G. Bauer, B. Hall, M. Paulitsch, and S. Varadarajan, "TTEthernet Dataflow Concept," in *Proceedings of the Eighth IEEE International Symposium on Network Computing and Applications*. IEEE, 2009, pp. 319–322.

[26] I. 802.1, "Time-sensitive networking task group." [Online]. Available: http://www.ieee802.org/1/pages/tsn.html

[27] J. Postel, "Internet control message protocol, STD 5, RFC 792," IETF, Standard Specification, September 1981.

[28] K. Hedayat, R. Krzanowski, A. Morton, K. Yum, and J. Babiarz, "A two-way active measurement protocol (TWAMP), RFC 5357," IETF, Standard Specification, October 2008.

[29] S. Shalunov, B. Teitelbaum, A. Karp, J. Boote, and M. Zekauskas, "A one-way active measurement protocol (OWAMP), RFC 4656," IETF, Standard Specification, September 2006.

[30] S. Bradner and J. McQuaid, "Benchmarking methodology for network interconnect devices, RFC 2544," IETF, Standard Specification, March 1999.

[31] ITU-T, "Recommendation ITU-T Y.1731, OAM functions and mechanisms for ethernet based networks," ITU-T, Standard Specification, July 2011.

[32] J. Perry, P. A. Iannucci, K. Fleming, H. Balakrishnan, and D. Shah, "Spinal Codes," in *SIGCOMM '12*, no. Ml. ACM, 2012, pp. 49–60.

[33] A. Rowe, R. Mangharam, and R. Rajkumar, "RT-Link: A Time-Synchronized Link Protocol for Energy- Constrained Multi-hop Wireless Networks," *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, pp. 402–411, 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4068297

[34] L. Doherty, J. Simon, and T. Watteyne, "Wireless Sensor Network Challenges and Solutions," 2012.

[35] Proakis and Saheli, *Digital Communications (5th Edition)*. McGraw-Hill, 2008.

[36] IEEE Computer Society, "802.1as: IEEE standard for local and metropolitan area networks - timing and synchronization for time-sensitive applications in bridged local area networks," IEEE, Standard Specification, March 30, 2011.

[37] R. A. Iannncci, "Toward a Dataflow / von Neumann Hybrid Architecture," in *ISCA '88 Proceedings of the 15th Annual International Symposium on Computer Architecture*, H. J. Siegel, Ed. IEEE Computer Society Press, 1988, pp. 131–140.

[38] K. Ekanadham, S. L. Gregor, K. Hiraki, R. A. Iannucci, and R. Rajkumar, "An Architecture for Generalized Synchronization and Fast Switching," in *Multithreaded Computer Architecture*, R. Iannucci, G. R. Gao, R. H. Halstead, and B. Smith, Eds. Boston, MA: Kluwer Academic Publishers, 1994, ch. 12.

[39] J. D. Davis, C. P. Thacker, and C. Chang, "BEE3 : Revitalizing Computer Architecture Research," *MSR Technical Report MSR-TR-2009-45*, 2009.

[40] S. A. Edwards and E. A. Lee, "The case for the precision timed (PRET) machine," in *Design Automation Conference (DAC)*, San Diego, CA, 2007.

[41] D. Broman, M. Zimmer, Y. Kim, H. Kim, J. Cai, A. Shrivastava, S. A. Edwards, and E. A. Lee, "Precision Timed Infrastructure: Design Challenges," in *to Appear in the Proceedings of Electronic System Level Synthesis Conference (ESLsyn)*. IEEE, 2013.

[42] I. Liu, J. Reineke, D. Broman, M. Zimmer, and E. A. Lee, "A PRET Microarchitecture Implementation with Repeatable Timing and Competitive Performance," in *Proceedings of the 30th IEEE International Conference on Computer Design (ICCD 2012)*. IEEE, 2012.

[43] M. Zimmer, D. Broman, C. Shaver, and E. A. Lee, "FlexPRET: A Processor Platform for Mixed-Criticality Systems," in *Proceedings of the 20th IEEE Real-Time and Embedded Technology and Application Symposium (RTAS)*. IEEE, 2014.

[44] E. Lee and D. Messerschmitt, "Pipeline interleaved programmable dsp's: Synchronous data flow programming," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 35, no. 9, pp. 1334–1345, 1987.

[45] R. Banakar, S. Steinke, B. Lee, M. Balakrishnan, and P. Marwedel, "Scratchpad memory: A design alternative for cache on-chip memory in embedded systems," in *In Tenth International Symposium on Hardware/Software Codesign (CODES), Estes Park*. ACM, 2002, pp. 73–78.

[46] Y. Kim, D. Broman, J. Cai, and A. Shrivastaval, "WCET-Aware Dynamic Code Management on Scratchpads for Software-Managed Multicores," in *Proceedings of the 20th IEEE Real-Time and Embedded Technology and Application Symposium (RTAS)*. IEEE, 2014.

[47] S. Andalam, P. S. Roop, A. Girault, and C. Traulsen, "PRET-C: A new language for programming precision timed architectures," INRIA Grenoble Rhone-Alpes, Technical Report 6922, June 4 2009.

[48] X. Li and R. von Hanxleden, "Multithreaded Reactive Programming—the Kiel Esterel Processor," *IEEE Transactions on Computers*, vol. 61, no. 3, pp. 337–349, 2012.

[49] M. Schoeberl, "A Java processor architecture for embedded real-time systems," *Journal of Systems Architecture*, vol. 54, no. 1-2, pp. 265 – 286, 2008.

[50] D. May, "The XMOS XS1 architecture," XMOS Limited, White Paper, 2009. [Online]. Available: https://www.xmos.com/zh/download/public/The-XMOS-XS1-Architecture(1.0).pdf

[51] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Staschulat, and P. Stenström, "The Worst-Case Execution-Time Problem - Overview of Methods and Survey of Tools," *ACM Transactions on Embedded Computing Systems*, vol. 7, pp. 36:1–36:53, May 2008.

[52] R. Wilhelm, D. Grund, J. Reineke, M. Schlickling, M. Pister, and C. Ferdinand, "Memory hierarchies, pipelines, and buses for future architectures in time-critical embedded systems," *IEEE Transactions on Computer Aided Design*, vol. 28, no. 7, pp. 966 – 978, 2009.

[53] J. Knoop, L. Kovcs, and J. Zwirchmayr, "Symbolic Loop Bound Computation for WCET Analysis," in *Perspectives of Systems Informatics*, ser. LNCS, E. Clarke, I. Virbitskaite, and A. Voronkov, Eds. Springer, 2012, vol. 7162, pp. 227–242.

[54] J. Gustafsson, A. Ermedahl, C. Sandberg, and B. Lisper, "Automatic derivation of loop bounds and infeasible paths for WCET analysis using abstract execution," in *Proceedings of the 27th IEEE International Real-Time Systems Symposium (RTSS'06)*. IEEE, 2006, pp. 57–66.

[55] Y.-T. Li and S. Malik, "Performance analysis of embedded software using implicit path enumeration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 12, pp. 1477–1487, 1997.

[56] C. Ferdinand and R. Wilhelm, "Efficient and precise cache behavior prediction for real-time systems," *Real-Time Systems*, vol. 17, no. 2, pp. 131–181, 1999.

[57] M. Schoeberl, "A time predictable instruction cache for a Java processor," in *On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops*. Springer, 2004, pp. 371–382.

[58] M. Paolieri, E. Quiñones, F. J. Cazorla, G. Bernat, and M. Valero, "Hardware Support for WCET Analysis of Hard Real-time Multicore Systems," in *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ser. ISCA '09. New York, NY, USA: ACM, 2009, pp. 57–68.

[59] S. Chattopadhyay, L. K. Chong, A. Roychoudhury, T. Kelter, P. Marwedel, and H. Falk, "A Unified WCET Analysis Framework for Multicore Platforms," *ACM Trans. Embed. Comput. Syst.*, vol. 13, no. 4s, pp. 124:1–124:29, Apr. 2014.

[60] H. Falk and P. Lokuciejewski, "A compiler framework for the reduction of worst-case execution times," *Real-Time Systems*, vol. 46, no. 2, pp. 251–300, 2010.

[61] I. Fuhrmann, D. Broman, S. Smyth, and R. von Hanxleden, "Towards Interactive Timing Analysis for Designing Reactive Systems," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2014-26, April 2014.

[62] Texas Instruments, "Precision PHYTER - IEEE 1588 precision time protocol transceiver," 2013. [Online]. Available: http://www.ti.com/product/dp83640

[63] Intel, "Intel ethernet controller i210: Datasheet." [Online]. Available: http://www.intel.com/content/www/us/en/ethernet-controllers/i210-ethernet-controller-datasheet.html

[64] PCI-SIG, "Precision time measurement (PTM), revision 1.0a, amendment to pci express," 2013. [Online]. Available: http://www.pcisig.com/specifications/pciexpress/specifications/ECN_PTM_Revision1a_31_Mar_2013.pdf

[65] "Energy harvesting technologies," 2013. [Online]. Available: http://www.mouser.com/applications/energy_harvesting/

[66] J. Shannon, H. Melvin, and A. G. Ruzzelli, "Dynamic flooding time synchronisation protocol for WSNs," in *Proceedings of the IEEE Global Communications Conference*. IEEE, 2012, pp. 365–371.

[67] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, pp. 147–163, 2002.

[68] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, 1999, pp. 263–270.

[69] N. Gehani and K. Ramamritham, "Real-time concurrent C: A language for programming dynamic real-time systems," *Real-Time Systems*, vol. 3, no. 4, pp. 377–405, 1991.

[70] E. Kligerman and A. D. Stoyenko, "Real-time Euclid: A language for reliable real-time systems," *Software Engineering, IEEE Transactions on*, vol. 12, no. 9, pp. 941–949, 1986.

[71] Y. Zhao, J. Liu, and E. A. Lee, "A programming model for time-synchronized distributed real-time systems," in *13th IEEE Real Time and Embedded Technology and Applications Symposium, 2007. RTAS '07*, April 2007, pp. 259 – 268. [Online]. Available: http://chess.eecs.berkeley.edu/pubs/325.html

[72] J. Zou, S. Matic, and E. Lee, "PtidyOS: A lightweight microkernel for Ptides real-time systems," in *Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, April 2012.

[73] B. Liskov, "Practical uses of synchronized clocks in distributed systems," *Distributed Computing*, vol. 6, no. 4, pp. 211–219, 1993.

[74] D. Schlesinger and M. Girardot, "Smart and connected passenger vehicles, pervasive connectivity will fundamentally change the automotive landscape," 2010. [Online]. Available: http://www.cisco.com/web/about/ac79/docs/pov/S_CPV_POV_0719FINAL.PDF

[75] K. Harris, "An application of IEEE 1588 to industrial automation," in *Precision Clock Synchronization for Measurement, Control and Communication, 2008. ISPCS 2008. IEEE International Symposium on*. IEEE, 2008, pp. 71–76.

[76] J. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, and D. Woodford, "Spanner: Google's Globally-Distributed Database," in *10th USENIX Symposium on Operating Systems Design and Implementation*, Hollywood, CA, 2012.

[77] S. Golan, S. Gannot, and I. Cohen, "Performance analysis of a randomly spaced wireless microphone array," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 121–124.

[78] N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 50, no. 12, pp. 3062–3080, 2004.