

TICTOC  
INTERNET-DRAFT  
Intended Status: Informational

Tim Frost  
Greg Dowd  
Symmetricom

Karen O'Donoghue  
US Navy

Expires: 03 May 2009

03 Nov 2008

**Architecture for the Transmission of Timing over Packet Networks**  
**draft-stein-tictoc-modules-03.txt**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on May 03, 2009.

Copyright Notice

Copyright (C) The IETF Trust (2008)

Abstract

This Internet draft proposes an architecture for the transmission of timing over packet networks. It identifies the major functional components, and maps the current solutions onto this framework.



## Table of Contents

<b><a href="#">1. Introduction</a></b>	3
1.1. TICTOC Layers	3
1.2. Generic TICTOC Client	4
1.3. TICTOC Functional Modules	5
<b><a href="#">2. Frequency Layer</a></b>	7
2.1. Local Oscillator and Frequency Synthesizers	7
2.2. Frequency Distribution Protocols	8
2.3. Frequency Acquisition Algorithms	8
2.3.1. Packet Selection and Discard Algorithms	9
2.3.2. Filtering and Control Servos	9
2.3.3. Server Contribution	10
2.3.4. Reference Algorithm	10
2.4. Frequency Presentation	10
<b><a href="#">3. Time Layer</a></b>	11
3.1. Time Distribution Protocols	11
3.2. Ranging	13
3.3. Time Presentation	13
<b><a href="#">4. Generic Modules</a></b>	15
4.1. Security Mechanisms	15
4.2. Autodiscovery and Master Clock Selection	15
4.3. Redundancy and Fail-Over Mechanisms	17
4.4. OAM, SSM, and Performance Monitoring	17
4.5. Network Management	17
4.6. Application profiles	17
<b><a href="#">5. Network Support</a></b>	18
5.1. Path Selection	18
5.2. Network and Traffic Engineering	18
5.3. Service Level Specifications and QoS settings	19
5.4. Routing considerations	19
5.5. On-path Support	20
<b><a href="#">6. Security Considerations</a></b>	20
<b><a href="#">7. IANA Considerations</a></b>	20
<b><a href="#">8. Acknowledgements</a></b>	20
INFORMATIVE REFERENCES	21
AUTHORS' ADDRESSES	21

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].



## **1. Introduction**

In this document the term "timing" will be used to refer to recovery of frequency and/or time (wall-clock).

There is an emerging need to distribute highly accurate time and frequency information over IP and over MPLS packet switched networks (PSNs). A variety of applications require time and/or frequency information to a precision which existing protocols cannot supply.

Several other groups are working on related issues. For example, the NTP Working Group in IETF is working on an upgrade of the long-standing Network Time Protocol [[RFC1305](#)]. The IEEE has recently revised its Precision Time Protocol (PTP) [[IEEE1588](#)]. The ITU-T has defined Synchronous Ethernet, a physical layer technology for transfer of frequency across native Ethernet [[G.8261](#), [G.8262](#)].

However, none of these efforts are sufficient by themselves to create a complete and robust time and frequency transfer ecosystem in the IP and MPLS network environment. The TICTOC (Timing over IP Connections and Transmission of Clock) Working Group was created to develop solutions that meet the needs of the various applications for timing over packet networks.

This draft attempts to define an architecture for a TICTOC system, identifying the various functional components, and considering the support required from the network in order to assist the timing recovery.

### **1.1. TICTOC Layers**

The most general TICTOC system can be logically decomposed into two layers, corresponding to the distribution of frequency and time, respectively, carried over the network by a timing protocol as shown in Figure 1. Examples of such timing protocols include Network Time Protocol (NTP) [[RFC1305](#)] and Precision Time Protocol (PTP) [[IEEE1588](#)].

Specific TICTOC systems may consist of either or both layers. For example, if time is not required for the application, then only the frequency layer may be present.

In some cases there may be additional support from the network to assist the timing distribution. For example, it may be possible to use a physical layer technology such as Synchronous Ethernet to distribute an accurate frequency. In this case, the timing protocol is only required to distribute time. Other examples of network support may include specific QoS settings for the timing protocol flow, and routing constraints to ensure an optimum path.



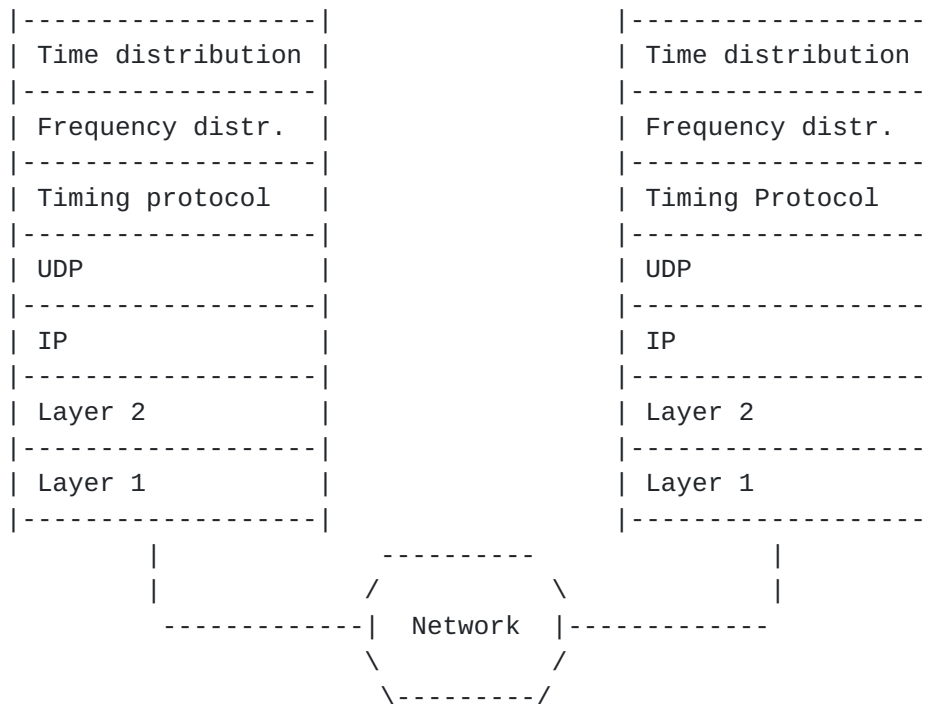


Figure 1. TICTOC Layers

### 1.2. Generic TICTOC Client

Figure 2 shows a generic TICTOC client, consisting of separate frequency and time acquisition modules, and the presentation modules for each.

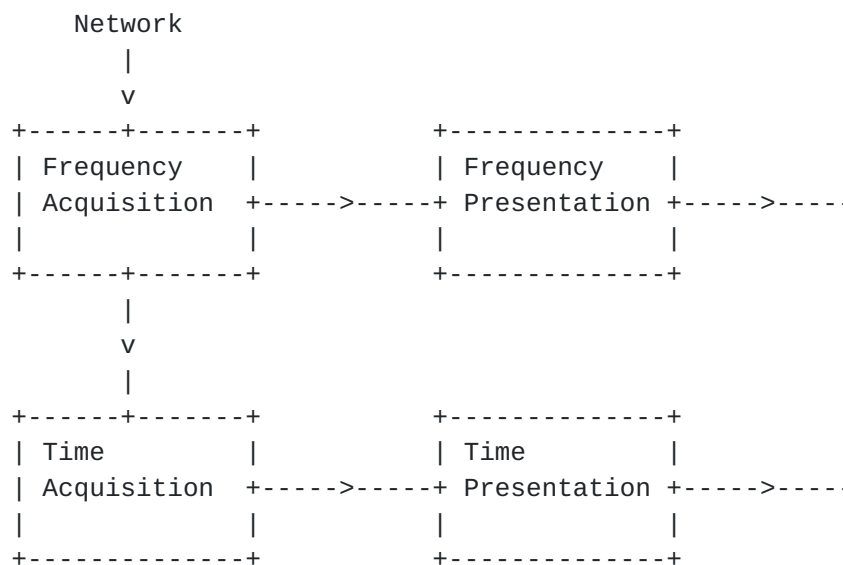


Figure 2. Generic TICTOC client

The frequency acquisition module is responsible for recovery of

frequency information distributed over a packet switched network

Frost/O'Donoghue/Dowd Expires 14 January 2009

[Page 4]



(PSN), such as IP or MPLS. This distribution is accomplished by use of a frequency distribution protocol. The frequency distributed may be derived from an international standard, or may be an arbitrary frequency needed at some remote location. If the value of the frequency itself is needed, then the output of this module is input to the frequency presentation module that formats the frequency according to application needs. Note that this format may be numeric prepared for display, or may take analog form and be used for locking or disciplining other analog frequencies. When time is needed, the frequency information is sent to the time acquisition module.

Even if the frequency itself is not needed, time acquisition almost always requires a stable frequency reference. This reference may be acquired from the frequency acquisition layer, or may be obtained via some other means, such as an accurate local frequency reference, or from a non-PSN mechanism for frequency distribution (e.g., GPS, SONET/SDH).

From the acquired or otherwise available frequency, we may derive arbitrary time (also known as "phase") by mathematical means. If the frequency reference is traceable to an international standard, then arbitrary time means a sequence of time values that are synchronous with true (wall-clock, UTC) time, but with an arbitrary offset. The purpose of the time acquisition module is to enable multiple TICTOC clients to share a single offset, and thus to agree as to marking of phase values. Such marked phases values are all that is required for certain applications, such as where multiple time-aware agents need to interact (e.g., systems employing Time Division Multiple Access systems).

When the time markings distributed by the time distribution protocol are traceable to an international standard, the TICTOC clients are said to have acquired wall-clock time. These wall-clock values are formatted for use by the intended application by the time presentation module.

### 1.3. TICTOC Functional Modules

The remainder of this document further subdivides these modules and identifies the sub-modules needed for timing distribution. Sub-modules may require one or more protocols, a set of processing algorithms, and implementations.

The frequency acquisition module may be subdivided into the following sub-modules. Not all need to be present in all implementations.

- o local oscillator
- o frequency generator (typically a digital synthesizer)
- o timestamp generator  
(not required if packets are sent at constant rate)
- o frequency distribution protocol

- o frequency acquisition algorithms

Frost/O'Donoghue/Dowd Expires 14 January 2009

[Page 5]

(may include packet selection algorithms, oscillator disciplining and frequency adjustment techniques)

The time acquisition module may be subdivided into the following additional sub-modules. Not all need to be present in all implementations.

- o time distribution protocol
- o ranging algorithms
- o clock adjustment techniques
- o timestamp generation (already mentioned)

In addition, various generic modules may be needed, for either frequency distribution, time distribution, or both.

- o security
- o autodiscovery and master clock selection
- o redundancy and fail-over mechanisms
- o OAM, synchronization status messaging, and performance monitoring
- o network management
- o application profiles

Any timing protocols should operate over the general internet without requiring specific support from the network. However, when operated over a specific, managed network where support is available, the accuracy of the time and frequency distribution will be enhanced.

Examples of network support include:

- o path selection and/or self-organization
- o network and traffic engineering for optimum transport of timing protocols
- o service level specifications and QoS settings
- o routing considerations
- o on-path support, e.g. PTP boundary or transparent clocks



## **2. Frequency Layer**

There are applications that require an accurate frequency reference, but do not require knowledge of absolute time. In addition, it is often wise to acquire frequency for applications that require absolute time but do not directly use frequency.

TICTOC is concerned with the network frequency transfer using packet technology. Frequency may be transferred across a network using the physical layer, such as through the use of a synchronous network interface (for example Synchronous Ethernet [[G.8262](#)]). The use of the physical layer may produce a higher quality frequency reference than achievable using packet based network frequency transfer, but is outside the scope of this document.

### **2.1. Local Oscillator and Frequency Synthesizers**

TICTOC masters and clients both need local sources of frequency. For masters, this may be provided by high quality Cesium clock with extremely good long term accuracy, or by an atomic clock with somewhat lower accuracy, but which is disciplined by comparison with a better clock (e.g., by GPS).

Note that a pure frequency master that does not need to know time can be standalone.

For clients, the local oscillator is usually a quartz crystal oscillators. Such oscillators may be stable, special cut crystals with double oven and temperature compensation, or lowly inexpensive crystals. In either case the frequency derived from these oscillators needs to be adjusted by the TICTOC frequency acquisition module. This adjustment can be electronic (e.g. changing the control voltage of a voltage controlled oscillator).

However, it is common practice not to directly adjust the physical oscillator, or even to directly use the oscillator's frequency. Instead, a digital synthesizer fed by the oscillator provides the required output frequency. Changing parameters of the digital synthesizer changes the output frequency in the required way. It is important for the digital synthesizer not to introduce too much jitter, and for the changing of parameters to be done in such fashion as to minimize transients.

Disciplining of the local oscillator, or setting the parameters of the digital synthesizer, is based on the arrival times of received frequency distribution protocol packets, and the information contained in them. When, for whatever reason, frequency distribution packets are not received, the frequency layer is said to be in "holdover mode". In holdover mode the local oscillator or

synthesizer is still used as usual, but it is no longer disciplined

based on new information from the distribution protocol (although it may still be adapted, if the acquisition module has models that have been trained during periods that the frequency distribution packets were received).

## 2.2. Frequency Distribution Protocols

The protocol used to distribute frequency across the packet network may be the same protocol used for time distribution, or may be an independent protocol. The important design consideration is that the protocol used is optimized for that purpose, and not compromised by the need to fulfill a dual role.

Frequency distribution protocols are "one way", i.e. only require information transfer from the master (where the frequency is known) to the client. In particular, frequency distribution protocols can be broadcast or multicast to many clients, without the master needing to know the client identities. Frequency distribution protocols may even operate when there is no return path available. Exceptions to this rule are control messages sent by the client requesting commencing or termination of the frequency distribution service, or changing its parameters (e.g., update rate).

Frequency distribution protocols can be broadcast, multicast or unicast. Multicast transmission conserves network bandwidth since timing packets may be distributed to multiple clients or slaves. However, unicast transmission is often more desirable, since it avoids the multicast replication operation in each switch or router, which may add variable delay.

The TICTOC architecture is modular, and not bound to the frequency distribution protocol. It is possible to use different frequency distribution methods for different applications and environments, e.g. the use of physical layer frequency distribution protocols such as Synchronous Ethernet.

## 2.3. Frequency Acquisition Algorithms

A TICTOC client needs to be able to recover the original timebase (or frequency reference) of the master or server. In general it achieves this by comparing the arrival time of packets with the original sending time. From this it can determine the relationship between the local timebase and the master timebase.

Observed arrival times of frequency distribution protocol packets are influenced by two phenomena:

- o frequency drift of the local oscillator relative to the master oscillator (typically caused by temperature and voltage fluctuations, and by aging phenomena)

o variation in delay of timing packets through the packet network



(PDV).

The former behavior needs to be tracked and compensated, while the latter needs to be filtered out. In order to eliminate the effects of PDV, frequency acquisition algorithms perform some sort of averaging and/or filtering, in an attempt to recover the frequency at which packets were sent. Typically this involves two steps:

- o packet selection and discard algorithms
- o filtering and control servos to "lock onto" the sending frequency

#### 2.3.1. Packet Selection and Discard Algorithms

In the simplest networks all frequency distribution packets received are used by the frequency acquisition algorithms to derive corrections to the local oscillator. A major problem with frequency acquisition in complex networks is the elimination of frequency distribution packets that, if used by the acquisition algorithms, would degrade the quality of the recovered frequency. Such packets are variously called outliers, falsetickers, etc.

There are several reasons for such unreliable packets. First, malfeasants may deliberately inject false packets in order to impair the TICTOC client's frequency recovery. We will discuss security mechanisms below. Second, PDV distributions for complex networks can be long tailed, leading to extremely misleading results. Third, various network degradations, e.g., congestion and reroute events, can lead to bizarre information being received.

Furthermore, even typical packets may have experienced significant delay variation, making them less suitable for exploitation than the small fraction of packets that may have experienced almost no delay (and thus minimal delay variation). When there is a significant fraction of packets that experience essentially no delay, it is reasonable to discard all others (assuming that after this decimation the sampling rate is still sufficiently high).

#### 2.3.2. Filtering and Control Servos

Since simple averaging would take too long to sufficiently eliminate the stochastic components, by which time the frequency difference between local oscillator and master oscillator would have changed, more efficient "control loops" are employed. Control loops are nonlinear mechanisms, and are thus able to "lock onto" frequencies, rather than simply removing stochastic noise. Conventional control loops include the Phase Locked Loop (PLLs), the Frequency Locked Loops (FLL), and combinations of the two.



Delay variation effects can be quite complex and traffic dependent. There are obvious effects such as queuing indeterminacy leading to stochastic PDV, and congestion leading to packet loss. There are also more subtle effects, for example multiple plesiochronous flows (e.g. TDM pseudowires) traversing forwarding elements can cause slow "beating" effects, generating low frequency wander that is difficult to eliminate. Another example is the batch processing effect introduced by some forwarders in which the first of a series of packets experiences greater latency than later packets.

Modern algorithms attempt to model the time behavior of local oscillator as compared to the master oscillator, and/or the network behavior. More complex algorithms may attempt blind separation of the contribution of the two effects.

#### 2.3.3. Server Contribution

Traditionally the focus on frequency acquisition algorithm design has been on the client. However it is possible to mitigate some of the aforementioned effects by modulating the packet generation rate of the master. Thus TICTOC algorithm modules may describe the client and the server components, and may require matching of these algorithms to achieve optimal performance.

#### 2.3.4. Reference Algorithm

Current technologies differ in how much of the algorithm is defined. NTP defines the clock recovery algorithm completely, while PTP only defines the on-the-wire protocol, leaving the clock recovery algorithms undefined. Ultimately, multiple algorithms may be required to suit different network environments and application requirements.

Next-generation frequency acquisition algorithms need to be optimized such that network bandwidth consumed by the frequency distribution protocol is minimized. Furthermore, these algorithms are required to be robust to network degradations such as packet delay, packet loss, packet delay variation (PDV) and infrequent stepwise changes in network traversal latencies (e.g., due to reroute events or network loading changes).

It may be necessary to specify a reference algorithm for comparison and validation purposes, but the TICTOC architecture will enable vendors to employ proprietary algorithms. It will also be possible to upgrade the reference algorithm in the future.

### 2.4. Frequency Presentation



In general, the output of the frequency acquisition module needs to be reformatted before use. In some cases the reference frequency distributed across the network is different from the frequency needed by the local application; in such cases a digital synthesizer can be employed to convert the acquired frequency to the desired one. Sometimes it is not frequency itself that is required, but a sequence of equally separated times; in such cases the sequence of time "ticks" is generated from the acquired frequency (once again, digital synthesizer is ideal for this).

### **3. Time Layer**

In general the time layer is fed accurate frequency from the frequency layer. There are applications that require accurate time, but do not need to acquire frequency over the PSN. For example:

- o if the update rate is high and the time resolution required is low
- o if highly accurate frequency is available locally
- o if frequency is distributed by means external to the PSN (e.g. GPS, LORAN, WWV)
- o if frequency is available by means of the network physical layer (e.g. PoS, SyncE).

In such cases the frequency input in Figure 2 is provided by the alternative frequency source, rather than the frequency layer. The TICTOC two layer decomposition is a conceptual one, and may not be easily identifiable in all implementations. For example, when using a pure PLL frequency acquisition module, true "frequency" is never derived, only relative phase. Similarly, tracking mechanisms may simultaneously model frequency and time, reducing convergence time by adjusting both simultaneously, rather than first acquiring frequency, and afterwards time. However, even in these cases the decomposition is a useful one.

#### **3.1. Time Distribution Protocols**

The purpose of the time distribution protocol is to calibrate a sequence of phase events generated by the local oscillator or digital synthesizer, thus converting arbitrary offset phase values into wall-clock values. This is done by sending packets containing timestamps from the master (where the time is known) to the TICTOC client. In order for the timestamp to accurately indicate the time that the packet was actually injected into the network (rather than some earlier time when the packet was formed, separated by a stochastic time delay from the injection time), the timestamp should be generated by the networking hardware. Providing any checksums or CRCs can be updated on-the-fly, this hardware-generated timestamp may be directly inserted into the packet. Alternatively, a subsequent

packet can be used to carry the measured injection time, as in the

PTP follow-up message. Where hardware assistance to measure the injection time accurately is not available, the stochastic delay should be minimized.

The timestamp in the time distribution protocol packet indicates the time that the master injected this packet into the network. On the other hand, the client receives this same packet after the propagation delay, i.e. the time taken to traverse the packet network. Were this time to be accurately known, the timestamp could be corrected, and the precise time known. Estimation of the traversal time is called ranging, and will be discussed below. PTP has an optional mechanism (transparent clocks) whereby forwarding delays, and even individual link delays are measured and compensated, thus leading to precise knowledge of the traversal delay. However, this mechanism requires upgrading of all intermediate network elements.

Due to the requirement of traversal delay measurement, time distribution protocols are generally bidirectional, that is, they require a bidirectional channel, require both master and client to send and receive messages, and usually assume symmetric (or known asymmetry) propagation times. Unlike frequency distribution, time distribution can not be entirely multicast, due to the ranging requirement.

There are two well-known protocols capable of running over a general-purpose packet network, NTP [[RFC1305](#)], and PTP [[IEEE1588](#)]. NTP is the product of the IETF, and is currently undergoing revision to version 4. PTP is a product of the IEEE Test and Measurement community, and has recently been revised to better accommodate telecommunication applications. The new version has a profiling mechanism enabling organizations to specify required and prohibited options, ranges and defaults of attribute values, and optional features, while maintaining interoperability.

The protocol used to transfer the reference frequency across the network may be the same protocol that is used to transfer time. The overriding design consideration is that frequency and time distribution protocols be optimized for their purpose, and not compromised by the need to fulfill a dual role.

The TICTOC architecture itself is not bound to a specific time distribution protocol. It is possible to upgrade, or replace this protocol, for example to improved the quality of the transfer, or to optimize the transfer for a different network environment, without redesigning the entire TICTOC architecture.





### 3.2. Ranging

To transfer time it is necessary to know the time of flight of packets from the time server to the client. The accuracy of time at the client can be no better than the accuracy provide by the ranging mechanism. Some ranging mechanisms require or can exploit special hardware assistance by intermediate network elements, such as PTP transparent clocks [[IEEE1588](#)].

A typical ranging mechanism functions by exchanging timestamps between master and client. For example, the master sends an initial timestamped packet. When this packet arrives at the client its arrival time (in terms of the local clock) is recorded. After some amount of time the client sends a response packet back to the master, containing the initial timestamp (in terms of the master clock), and the packet arrival and retransmission times (in terms of the client clock). When this packet is received by the master a fourth timestamp is generated (in terms of the master clock). Since the second and third timestamps are both in terms of the client clock, their difference - representing the amount of time the client hesitated between receipt of the initial packet and sending the response packet - is readily computed. This difference can be subtracted from the difference between the fourth and first timestamps, to yield the sum of propagation times. Under the assumption of symmetry, the one-way time is half this sum. Note that since the difference between third and fourth timestamps is short, possible frequency inaccuracy of the client has little deleterious effect.

Various variations on this basic four-timestamp method are possible. In the three timestamp method the client sends the time difference (e.g., generated by resetting a timer upon packet arrival) rather than the two timestamps. When symmetry is not a valid assumption, additional information (e.g., ratio or difference between expected propagation delays) can be used to extract the one-way delay.

Ranging accuracy is reduced by deviation from expected asymmetry in the network. Mechanisms to avoid asymmetry at the network layer are useful (for example using MPLS RSVP-TE paths, or the use of the peer-to-peer mechanism described in IEEE1588), as is the ability to correct asymmetry in the physical connection through the use of external calibration.

### 3.3. Time Presentation

In general, the output of the time acquisition module needs to be reformatted before use. Formatting includes translation between different representations (e.g., from seconds after a given date to

date and time), changing time zones, etc. When the time needs to be

displayed, e.g., on a computer or mobile device, further processing is often required, such as identification of day of week, translation to other calendars (e.g., Hebrew, Islamic, Chinese), conversion between TAI and UTC, and flagging of holidays and special dates.

## **4. Generic Modules**

### **4.1. Security Mechanisms**

Time and frequency services are a significant element of network infrastructure, and are critical for certain emerging applications. Hence time and frequency transfer services **MUST** be protected from being compromised. The most significant threats are false timing servers distributing purposely misleading timing packets, and man in the middle tampering with valid timing packets. Both threats would enable a malfeasant to mislead TICTOC clients, and to bring down critical services.

Based on the aforementioned threats, one can conclude that confidentiality and replay protection are usually not needed (and in many cases not possible), but source authentication and integrity protection are vital. In general, it is the master that needs to be authenticated to the server, although in certain cases it may be needed to authenticate the client to the master.

Applying IPsec Authentication Header (AH) mechanisms to all timing packets is not feasible, due to the computational resources consumed by public key cryptography algorithms. Adoption of IPsec would impact time acquisition accuracy, and would lead to new methods for malfeasants to disrupt the timing distribution protocols by exhausting resources and denial of service from TICTOC clients. Furthermore, certificates used to authenticate packets have lifetimes that must be enforced for secure use. Certification of time packets themselves can thus lead to circular dependence and to attacks that invalidate valid time packets and substitute invalid ones. NTP implementations provided an authentication protocol called Autokey. Autokey utilizes public key algorithms to encrypt cookies, symmetric key message digests for integrity, and digital signatures for source authentication.

Any security mechanism must be designed in such a way that it does not degrade the quality of the time transfer. Such a mechanism **SHOULD** also be relatively lightweight, as client restrictions often dictate a low processing and memory footprint, and because the server may have extensive fan-out. The mechanism also **SHOULD** not require excessive storage of client state in the master, nor significantly increase bandwidth consumption.

### **4.2. Autodiscovery and Master Clock Selection**

The topology of presently deployed synchronization networks is universally manually configured. This requires manual or management-plane configuration of master-client relationships, as well as preconfigured fallback behaviors. This strategy is workable for SDH

networks, where there are a relatively small number of network

elements that require such configuration, and all elements are controlled by a single entity. In packet switched network scenarios there may be a very large number of independent network elements requiring timing, making automatic mechanisms important. Such autodiscovery, automatic master selection algorithms, and perhaps control plane protocols to support these features, are important features of the TICTOC architecture.

There are application scenarios where it is desirable for clocks to advertise their service, and for clients to automatically select a clock with the required accuracy and path characteristics. The optimum advertising method may depend on the environment and the application. For example a host may be best served by finding a suitable clock (or set of clocks) through a DHCP parameter, whilst a provider edge may find it more convenient to discover the available clock through BGP.

Once a TICTOC client has discovered potential TICTOC masters, it must choose how to derive its clock from one or more of them. This choice can be aided using two types of information:

- o claims made by the potential masters as to the accuracy of its local clock (see OAM and SSM below)
- o analysis of the stability of frequency recovered from the potential masters.

One tactic that a TICTOC client may employ is to individually choose which master to use based on this information. Even if statically configured to use a particular master, a client may be allowed to make independent decisions when the master becomes unavailable or sends synchronization status messages indicating a fault condition.

A second tactic is not to make a hard decision at all, but rather to perform weighted ensemble averaging of frequencies recovered from all available masters. The weightings, once again, may be based on claims made by the masters or by monitoring of frequency stability.

Using either tactic, it is important to ensure that "timing loops" are not formed. A timing loop is an erroneous topology that causes locking onto frequencies not traceable to a high quality source. Loops are avoided by ensuring that the frequency distribution paths form a tree.

Yet another method is not to decide on a timing distribution tree at all, but rather to enable self-organization of independently acting intelligent agents. In such cases the meaning of master and client becomes blurred, as all agents exchange timing information with a subset neighbors that have been discovered. This method may be driven by timing acquisition algorithms that guarantee global convergence of the timing agents, meaning that over time the average



timing error decreases, although a given agent's error may sometimes increase.

#### 4.3. Redundancy and Fail-Over Mechanisms

Since synchronization is a critical function in many network applications, operators conventionally protect it from single points of failure. Redundant sources of synchronization are normally provisioned, connected via diverse paths to prevent loss of synchronization at the end station.

This is also required in packet synchronization. It may be necessary to provision alternative paths, or techniques such as fast re-route to ensure that connection between a time server and client devices is not lost for too long.

#### 4.4. OAM, SSM, and Performance Monitoring

In order to ensure continuity and connectivity of the path from the master to the client, and the reverse path when needed, an Operations, Administration, and Maintenance protocol may be needed. When the master sends timing distribution packets at a constant rate, faults are detected by the client after a few interpacket times; when the rate is variable, the problem is detected after a few times the maximum interpacket interval. However, in either case the master may be unaware of the problem.

Synchronization Status Messages (SSMs) were traditionally used in TDM networks to indicate the accuracy of the source upon which an incoming TDM link based its timing, and to notify the remote site of clock failures. Timing distribution protocols generally have similar functions.

Performance monitoring is important to ensure the proper operation of timing systems, and may be integrated into OAM mechanisms.

#### 4.5. Network Management

As for all network infrastructure mechanisms, timing distribution systems SHOULD be managed. This implies provision of management agents in TICTOC masters and clients, and definition of the appropriate MIBs.

#### 4.6. Application profiles

PTP defines the concept of "profiles", defining the attributes and options of the protocol required to support a given application. The concept is extensible to other timing protocols, to ensure interoperable components of a timing system.





Profiles are developed by standards organizations or other industry bodies. For example, a "Telecom Profile" is currently under definition by the ITU-T. The TICTOC working group will need to specify the profile for the particular applications under its remit.

## **5. Network Support**

### **5.1. Path Selection**

In infrastructure applications it may be possible for TICTOC devices to seek co-operation from routing elements to optimize the path through the network in order to obtain a better quality of time and frequency transfer that is possible when the timing distribution protocol operates independent of the network layer.

At the simplest level this is use of paths that can support the required quality of service, and have the lowest congestion impact. However it is also possible for the network layer to be a proactive partner in the transfer process. For example paths may be selected on the basis of their symmetry, or such that all are capable of supporting physical frequency transfer (for example Synchronous Ethernet), or nodes may be selected such that they are all capable of supporting transparent clock.

In addition to having to deal with degradation of service due to congestion, TICTOC must not add to the problem. Thus, TICTOC timing distribution protocols MUST be able to act in a TCP friendly way, even at the expense of minor degradation of performance. In such cases, it may be required for client to be informed of the change in operating conditions.

### **5.2. Network and Traffic Engineering**

Every network element (e.g. router or switch) in a packet network adds varying amounts of delay to the packet. This variation is typically correlated to the load on that network element at the time, and especially to the shared load on the output port.

Therefore, there is some maximum limit on both the traffic load and the number of network elements that a packet timing flow can traverse before being degraded beyond the point at which the recovered time or frequency is outside of the specification for the given application.

This maximum limit will change depending on:

- o stringency of the application requirements
- o stability and environment of the local oscillator at the time/frequency client device
- o traffic load in the network
- o topology of the network



- o physical layer aspects of the network.

When planning a time/frequency distribution service over a managed network, the network planner must take these considerations into account. These will influence the appropriate locations for the time/frequency servers and any on-path support elements that may be deployed.

Traffic engineering may be used to control the traffic load in the network on the routes used by the timing flows. This may help to control the delay variation in the timing flow, and hence improve the performance of the clock recovered by the client.

### 5.3. Service Level Specifications and QoS settings

The traditional approach to specifying network performance has been to define a series of metrics such as packet loss and packet delay variation. However, these metrics are not good predictors of how a packet timing scheme is going to perform. For instance, the packet delay variation metric is too abstract, since it doesn't take into account the distribution of delays, or the correlation of delays over a longer interval.

Recent research has examined the application to PDV of new metrics borrowed from synchronization standards such as TDEV and a modified version called minTDEV [[minTDEV](#)]. The goal is to define a metric that is both measurable and capable of continuous monitoring. This can then form the basis of a service level specification for a network capable of running time/frequency distribution to a given performance level.

Further, the network operator needs to know how to tune the network to stay within the specified limit, since no operator is going to sign up to a service level specification that he has no control over. Appropriate QoS settings and techniques must be deployed to ensure the timing packets are forwarded through the routers in the optimum way.

### 5.4. Routing considerations

The basic method for calculating a time offset of a remote slave connected over a packet network makes an assumption that the network delays in each direction are the same. Any asymmetry between the forward and reverse directions yields an error in the time offset estimation.

While there may be various inferences that an algorithm can make concerning the size of that asymmetry, there are no currently known techniques for directly calculating its magnitude. Therefore it is important that the routing is constrained to make the packet flow



take the same path in each direction. This in itself will not guarantee that the time delay is the same in each direction, but it should minimize any difference.

#### 5.5. On-path Support

The TICTOC architecture will be commensurate with the public Internet, and the timing distribution protocol chosen will be able to function over general packet switched networks.

In some cases on-path support (for example PTP transparent clocks) may be needed in order to obtain the required frequency or time accuracy. Such on-path support cannot be expected to be universally available over the public Internet, and it is not a goal of the TICTOC working group to propose augmentation of basic forwarding elements. However, such on-path support may be provided on service provider or enterprise networks, and in such cases TICTOC protocols should be able to exploit it.

In networks with on-path support, it may be that the optimum path for time transfer is not congruent with the optimum path for data transfer. By using special-purpose IP addresses, and making routing aware of the required path characteristics and address attributes, it is possible to construct paths within the network that maintain the required time transfer characteristics.

The TICTOC Working Group will specify the required path characteristics and work with the ISIS and OSPF Working Groups to specify the necessary routing extensions to support these requirements.

TICTOC traffic may need to traverse NATs and firewalls.

## **6. Security Considerations**

Security aspects of TICTOC have been addressed above.

## **7. IANA Considerations**

No IANA actions are required as a result of the publication of this document.

## **8. Acknowledgements**

The authors wish to thank Yaakov Stein and Stewart Bryant for their contributions in the development of this document, and also Alon Geva, Gabriel Zigelboim, and Laurent Montini for invaluable comments.



## INFORMATIVE REFERENCES

- [IEEE1588] IEEE, "Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE1588-2008.
- [G.8261] ITU-T, "Timing and Synchronization Aspects in Packet Networks", G.8261, April 2008
- [G.8262] ITU-T, "Timing characteristics of synchronous Ethernet equipment slave clock (EEC)", G.8262, August 2007
- [RFC1305] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", [RFC 1305](#), March 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [minTDEV] G. Zampetti and L. Cosart, "Definition of minTDEV", COM15-C363-E, Contribution to ITU-T Study Group 15, Question 13, May 2007

## AUTHORS' ADDRESSES

Tim Frost,  
Symmetricom Inc.,  
Tamerton Road,  
Roborough,  
Plymouth, PL6 7BQ,  
United Kingdom.  
Email: tfrost@symmetricom.com

Greg Dowd,  
Symmetricom Inc.,  
[2300 Orchard Parkway](#),  
San Jose,  
CA 95112,  
USA.  
Email: gdowd@symmetricom.com

Karen O'Donoghue  
US Navy  
Code B35, NSWCDD,  
[17320 Dahlgren Rd.](#)  
Dahlgren,  
VA 22448,  
USA.  
Email: karen.odonoghue@navy.mil





## Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgement

Funding for the RFC Editor function is provided by the Internet Society.

