

TICTOC	Y(J). Stein	
Internet-Draft	RAD Data Communications	
Intended status: Informational	S. Bryant	
Expires: October 11, 2008	Cisco Systems	
	K. O'Donoghue	
	US Navy	
	April 09, 2008	

[TOC](#)

## TICTOC Modules

**draft-stein-tictoc-modules-01.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/ietf/1id-abstracts.txt>.

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

This Internet-Draft will expire on October 11, 2008.

### Abstract

This Internet draft proposes the modularization of TICTOC (Transmitting Timing over IP Connections and Transfer of Clock) work. In particular, it breaks the work down into individual modules (deliverables) that need to be developed.

---

## Table of Contents

- [1.](#) Introduction
- [2.](#) Frequency Layer

- [2.1.](#) Local Oscillator and Digital Synthesizers
  - [2.2.](#) Frequency Distribution Protocols
  - [2.3.](#) Packet Select/Discard Algorithms
  - [2.4.](#) Frequency Acquisition Algorithms
  - [2.5.](#) Frequency Presentation
- [3.](#) Time Layer
  - [3.1.](#) Time Distribution Protocols
  - [3.2.](#) Ranging
  - [3.3.](#) Time Presentation
- [4.](#) Other Modules
  - [4.1.](#) Security Mechanisms
  - [4.2.](#) Autodiscovery and Master Clock Selection
  - [4.3.](#) OAM, SSM, and Performance Monitoring
  - [4.4.](#) Path Selection
  - [4.5.](#) On-path Support
  - [4.6.](#) Network Management
- [5.](#) Security Considerations
- [6.](#) IANA Considerations
- [7.](#) Acknowledgements
- [8.](#) Informative References
- [§](#) Authors' Addresses
- [§](#) Intellectual Property and Copyright Statements

[Editor's Note: The present version of this draft contains references to work to be performed by the TICTOC working group. This language has been included in order to help with the chartering of this working group, and will be removed in the next revision.]

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 [\[RFC2119\] \(Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.\)](#).

---

## 1. Introduction

[TOC](#)

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

The most general TICTOC system can be logically decomposed into two layers, corresponding to the distribution and presentation of frequency and time, respectively. In Figure 1 we depict these layers, and the top-level modules in these layers, for a TICTOC client. Specific TICTOC systems may consist of either or both layers. For example, if time is required but frequency is available from a physical layer mechanism, then the frequency layer is omitted. On the other hand, if time is not required for the application, then only the frequency layer may be present.

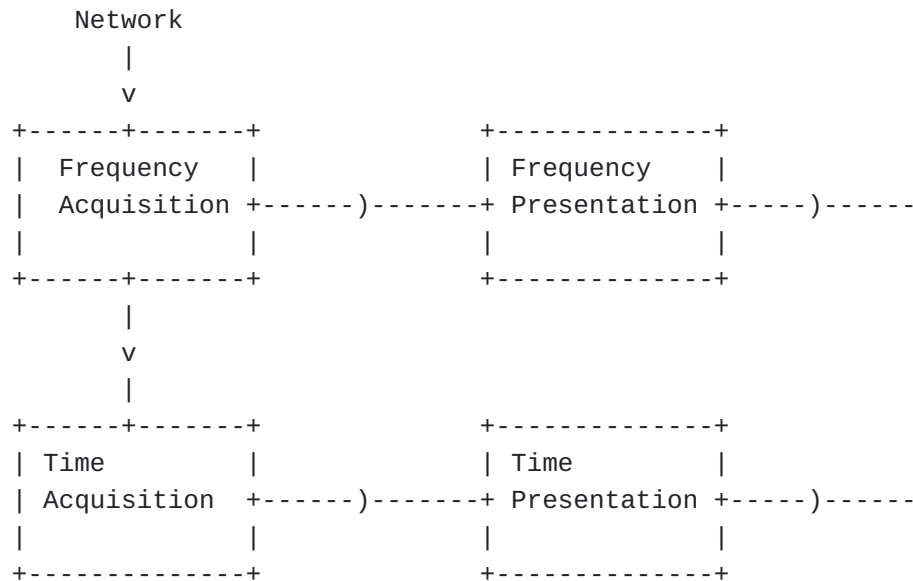


Figure 1. Generic TICTOC client

Referring to Figure 1, the frequency acquisition module is responsible for recovery of frequency information distributed over a packet switched network (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. The remainder of this document further subdivides these modules and identifies the submodules needed for timing distribution. Submodule may require one or more protocols, a set of processing algorithms, and implementations. Descriptions of these protocols, algorithms, and reference implementations are the TICTOC deliverables. The frequency acquisition module may be subdivided into the following submodules. Not all need to be present in all implementations. Those that need not be developed by TICTOC are marked by an asterisk.

- \*local oscillator (\*)
- \*digital synthesizer (\*)
- \*timestamp generation (not required if packets are sent at constant rate)
- \*frequency distribution protocol
- \*packet selection/discard algorithms
- \*frequency acquisition algorithms
- \*disciplining/adaptation/clock adjustment

The time acquisition module may be subdivided into the following submodules. Not all need to be present in all implementations.

- \*time distribution protocol
- \*ranging
- \*timestamp generation (already mentioned)

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

- \*security
- \*autodiscovery and master clock selection
- \*OAM, synchronization status messaging, and performance monitoring
- \*path selection and/or self-organization
- \*on-path support

## 2. Frequency Layer

[TOC](#)

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 [\[G8262\] \(ITU-T, "G.8262 - Timing characteristics of synchronous ethernet equipment slave clock \(EEC\)," .\)](#)). 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 Digital Synthesizers

[TOC](#)

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

[TOC](#)

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, while unicast transmission is often more desirable.

Ideally the frequency distribution protocol should be decoupled from the algorithm used to derive the local reference frequency, and the packets sent should only contain information that is physically meaningful without reference to a particular algorithm. For example, if the master sends packets at a constant rate as measured by the frequency reference, then the existence of the packet itself is the only physically meaningful information, and the packet should contain nothing but headers required for packet delivery. If the packets are not sent at a constant rate, they should contain nothing but a sufficiently precise timestamp. The use of extension mechanisms for carrying additional information may be considered for specific cases. The TICTOC Working Group will select a suitable network frequency transfer protocol. This may be based on an existing protocol, although that is not to be considered a requirement.

The basic TICTOC architecture itself should not be strongly bound to the frequency distribution protocol. It should be possible to upgrade or completely replace this protocol (e.g., to improve accuracy), or to optimize the transfer for a different network environment, without redesigning the TICTOC architecture.

---

### 2.3. Packet Select/Discard Algorithms

[TOC](#)

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).

---

### 2.4. Frequency Acquisition Algorithms

[TOC](#)

Observed arrival times of frequency distribution protocol packets are influenced by two phenomena: time behavior of the local oscillator as compared to the master oscillator, and network delay behavior (PDV). The former behavior needs to be tracked, 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. 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.

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.

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).

The TICTOC Working Group may specify a reference algorithm, but the TICTOC architecture will enable vendors to employ proprietary algorithms. It will also be possible to upgrade the reference algorithm in the future.

---

## 2.5. Frequency Presentation

[TOC](#)

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, a digital synthesizer is ideal for this).

---

[TOC](#)



### 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:

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

In such cases the frequency input in Figure 1 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

[TOC](#)

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. When this is not possible, the stochastic delay should be minimized. The IEEE 1588 protocol has a follow-up message, to indicate the actual injection time of the previous packet.

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. The IEEE 1588 protocol 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 [\[RFC1305\] \(Mills, D., "Network Time Protocol \(Version 3\) Specification, Implementation," March 1992.\)](#), and IEEE 1588 [\[1588\] \(IEEE, "1588-2002 Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," .\)](#). NTP is the product of the IETF, and is currently undergoing revision to version 4. IEEE 1588 is a product of the IEEE Test and Measurement community. It is presently specified in a limited first version, while the second version (1588v2) is soon to be a standard. IEEE 1588v2 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 optimised for their purpose, and not compromised by the need to fulfill a dual role.

The TICTOC Working Group will select a suitable time distribution transfer protocol or protocols. There are three options here:

- \*a new or alternative version of NTP, to be called NTPv5

- \*a profile of 1588

- \*a completely new protocol.

The selection will be made by producing a set of specific requirements for the TICTOC timing distribution protocol, and by a disinterested evaluation of each of these options in light of these requirements. The requirements will be based on the applications listed in the TICTOC problem statement. If neither of the first two options fulfill the requirements, then the third option will be chosen. Even if the first two options equally fulfill the requirements one of them will be chosen as the mandatory protocol, and the use of the other will be permissible under specific circumstances.

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

---

### 3.2. Ranging

[TOC](#)

To transfer time it is necessary to know the time of flight of time of packets from the time server to the client. The accuracy of time at the client can be no better than the accuracy provided by the ranging mechanism. Some ranging mechanisms require or can exploit special hardware assistance by intermediate network elements, such as IEEE 1588 transparent clocks [\[1588\] \(IEEE, "1588-2002 Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," .\)](#) .

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 (for example using MPLS RSVP-TE paths, or the use of the peer-to-peer mechanism described in IEEE1588v2) are useful, as is the ability to correct asymmetry in the physical connection through the use of external calibration.

---

### 3.3. Time Presentation

[TOC](#)

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. Other Modules

[TOC](#)

---

### 4.1. Security Mechanisms

[TOC](#)

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

[TOC](#)

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 an important TICTOC deliverable.

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.

The TICTOC Working Group will specify the required clock characteristics and identify the set of clock and client characteristics and the application characteristics that influence the optimum method of clock discovery. It will then specify one or more methods of clock autodiscovery together with associated applicability information.

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:

- \*claims made by the potential masters as to the accuracy of its local clock (see OAM and SSM below)

\*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. OAM, SSM, and Performance Monitoring

[TOC](#)

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.

---

[TOC](#)

#### 4.4. Path Selection

In infrastructure applications it may be possible for TICTOC 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 that are explicitly chosen to be symmetric, or paths may be selected 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 the master to inform the client of the change in operating conditions.

---

#### 4.5. On-path Support

[TOC](#)

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 IEEE 1588 transparent clock) 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.

---

#### 4.6. Network Management

[TOC](#)

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.

---

#### 5. Security Considerations

[TOC](#)

This document proposes modularization of the work of a proposed Working Group, and thus does not, in itself, raise security concerns. Security aspects of TICTOC have been addressed above.

---

#### 6. IANA Considerations

[TOC](#)

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

---

#### 7. Acknowledgements

[TOC](#)

The authors wish to thank Alon Geva, Gabriel Zigelboim, and Laurent Montini for invaluable comments.

---

#### 8. Informative References

[TOC](#)

[1588]	IEEE, "1588-2002 Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems."
[G8262]	ITU-T, "G.8262 - Timing characteristics of synchronous ethernet equipment slave clock (EEC)."
[RFC1305]	<a href="#">Mills, D.</a> , " <a href="#">Network Time Protocol (Version 3) Specification, Implementation</a> ," RFC 1305, March 1992 ( <a href="#">TXT</a> , <a href="#">PDF</a> ).
[RFC2119]	<a href="#">Bradner, S.</a> , " <a href="#">Key words for use in RFCs to Indicate Requirement Levels</a> ," BCP 14, RFC 2119, March 1997 ( <a href="#">TXT</a> , <a href="#">HTML</a> , <a href="#">XML</a> ).

---



## Authors' Addresses

[TOC](#)

	Yaakov (Jonathan) Stein
	RAD Data Communications
	24 Raoul Wallenberg St., Bldg C
	Tel Aviv 69719
	ISRAEL
Phone:	+972 3 645-5389
Email:	<a href="mailto:yaakov_s@rad.com">yaakov_s@rad.com</a>
	Stewart Bryant
	Cisco Systems
	250 Longwater Ave., Green Park
	Reading RG2 6GB
	United Kingdom
Email:	<a href="mailto:stbryant@cisco.com">stbryant@cisco.com</a>
	Karen O'Donoghue
	US Navy
	Code B35, NSWCDD, 17320 Dahlgren Rd.
	Dahlgren, VA 22448
Email:	<a href="mailto:karen.odonoghue@navy.mil">karen.odonoghue@navy.mil</a>

---

## Full Copyright Statement

[TOC](#)

Copyright © 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

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).