

Draft-kurtis-tictoc-itp-req-00

Generally...

- ...says we should work on requirements and then find the best solution from there
- We took a generalised view at what requirements we have for 'timing' over the Internet
 - After all that is what the IETF designs (See RFC3935)
 - The overall goal of TICTOC is to improve the timing distribution over the Internet - compared to today
- Draft requires the protocol time resolution to be better than 1ps
- Draft (will) requires the protocol date format to be MJD stored in 32-bits (11734883 years if they are all leap years)

Backward compatibility

- We don't want to re-do the IPv4 / IPv6 mess...
 - A new server MUST be capable of replying to an NTPv4 query
- It would be advantageous to say that ITP is NTPv5
 - Pros and cons with a new packet format should be evaluated

Topology

- ITP has to be topology agnostic
- We should analyse methods for path characteristic discovery
 - For example variable sized packets (Van Jacobsen pathchar)
 - Intermediate node ITP support
 - Filter algorithms

Hardware support

- Can't be assumed
- Should be possible to use if available
- The ITP should be able to leverage HW support down to the resolution of ITP (1ps)
- Out-of-band timing transfer should also be possible
 - ITP must be able to support the metrology community for primary timing measures
- Define mapping additions to all IPoverPidgeons documents

Resource management

- The protocol must support resource management so that
 - A client can request a certain poll rate
 - A server can either agree, deny, or suggest an acceptable poll rate
- Blind-One-way-transfer
 - Client requests an update rate for a certain time period – without additional queries
- Linked to authentication
 - We get there...

Authentication

- ITP must support authentication of
 - Source of time
 - Server identity
- Validation of authenticity of the transaction

Clock (Master) selection

- Algorithm dependent (implementation)
- Basic clock algorithm **MUST** be specified and supported by all implementations

Timescale

- Drafts goes through the four possible scenarios
 - TAI / GNSS (no leap second in time scale. Has to be handled by client all the time)
 - UTC (leap second in time scale. Client has to handle inserts and deletes)
 - Combinations of above
 - Ignore time scale

External dependencies

- The ITP client must be capable of independently of other sources tell the user ToD in a known time-scale

Handling of additional information

- Server needs to be capable of indicating to the client that there is additional information available from the server that might be required to handle the given information
 - There can be different classes of additional information
 - Informational
 - Required

APIs

- Host API
 - Used to steer time and frequency of host
- OS (Application) API
 - Used by client software that needs to determine ToD and other time related issues
 - The API should be defined so that an ITP server can use it as time source