

Real-time Protocol Requirements

Status of this memo

This document is an Internet Draft. 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. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a ``working draft'' or ``work in progress``.

To learn the current status of any Internet-Draft, please check the `1id-abstracts.txt` listing contained in the Internet-Drafts Shadow Directories on `ds.internic.net`, `nic.nordu.net`, `ftp.isi.edu`, or `munari.oz.au`.

A revised version of this draft document will be submitted to the IESG as a Proposed Standard for the Internet Community. Discussion and suggestions for improvement are requested. This document will expire six months after publication. Distribution of this draft is unlimited.

1. Abstract:

The purpose of this document is to create a framework for selecting the appropriate solution for a real-time protocol designed to allow calendaring and scheduling systems from different vendors to interoperate. To that end, it describes the assumptions about the context in which this protocol will operate, and the requirements that the protocol must meet.

These requirements are not intended to apply to a companion protocol which will use E-mail based transport to achieve interoperability. The E-mail based protocol, which is the subject of a different document, will not make any assumptions about transport latency.

2. Assumptions:

2.1. iCalendar

It is assumed that events will be represented as iCalendar objects encoded using MIME.

2.2. iCalendar Profiles

It is assumed that specific profiles have been defined for iCalendar so that each iCalendar object completely describes how a receiving calendar system should interpret the received object. The mechanism used to deliver the iCalendar object to the receiving system **MUST** not contain data that will affect the interpretation of the iCalendar object.

2.3. iCalendar Attendee identification

It is assumed that iCalendar attendees are identified as [RFC822](#) addresses. Further, for an individual we assume that this address is their e-mail address.

3. Requirements:

3.1. Bounded latency

The protocol must be capable of returning a response to the requester (client) within a definite limited time (<60 seconds ???). A mechanism needs to exist to inform the client that a well formed request has been received within a limited time. This will be necessary to support exchanges where processing time on the receiver would exceed the definite limited time.

3.2. Simple

The protocol should be simple to understand, debug and to implement on a variety of devices capable of connection to a IP internetwork.

3.3. Leverages existing, widely deployed Internet technologies

Wherever possible, the protocol should reuse existing, widely deployed Internet technologies or popular conventions rather than invent a new technology unnecessarily.

3.4. Efficient for the particular task

The protocol should efficiently handle the task it is designed to handle, establishing interoperability between calendaring and scheduling systems. As such, generality is not a goal for this protocol.

3.5. Scaleable

The protocol should be designed to handle communication from any Internet node to any other Internet node, that is, it should support an unbounded number of users and servers.

The protocol should support an unbounded number of servers per domain. A domain is a part of the address as defined in [RFC822](#).

The protocol should be designed to handle a large number of calendars (i.e. users, resources, etc.) per server.

3.6. Secure

The protocol should be designed to ensure that the communicating nodes are who they are supposed to be.

The protocol should be designed to enable private communication between nodes.

The protocol should be designed to work well with and be deployed in conjunction with existing firewall technology.

3.7. Server address resolution

The protocol should define the mechanism that the client will use to locate the appropriate servers corresponding to the attendee addresses contained in the iCalendar object that is submitted for transport.

3.8. Easy deployment and administration

The protocol should be easily deployed within the existing Internet infrastructure. That is, it should not require substantial new technology or administrative overhead or cause incompatibilities with existing technology. In fact, it should fit well into the existing infrastructure.

4. Acknowledgments:

Thanks to the following people for helpful comments on an earlier draft: Einar Stefferud, Derik Stenerson, Frank Dawson and Steve Hanna.

5. Author's Address:

Anik Ganguly

Campbell Services, Inc.

21700 Northwestern Highway, 10th Floor

Southfield, MI 48075 USA

Email: anik@ontime.com

Document expires: 5-Oct-1997

Ganguly

[Page 3]