DECADE
Internet-Draft
Intended status: Informational
Expires: April 21, 2011

Y. Gu
Huawei
D. Bryan
Cogent Force, LLC / Huawei
Y. Yang
Yale University
R. Alimi
Google

October 18, 2010

DECADE Requirements
draft-ietf-decade-reqs-00

Abstract

The target of DECoupled Application Data Enroute (DECADE) is to provide an open and standard in-network storage system for applications, primarily P2P applications, to store, retrieve and manage their data. This draft enumerates and explains requirements, not only for store and retrieve, but also for data management, access control and resource control, that should be considered during the design and implementation of a DECADE system. These are requirements on the entire system; some of the requirements may eventually be implemented by an existing protocol with/without some extensions (e.g., the data transport level). A user of DECADE as a complete architecture would be guaranteed complete functionality.

Requirements Language

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].

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and 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."

This Internet-Draft will expire on April 21, 2011.

Copyright Notice

Table of Contents

1. Introduction

The object of DECoupled Application Data Enroute (DECADE) is to provide an open and standard in-network storage system for applications, primarily applications that could be implemented using a content distribution paradigm, where data is broken in to one or more chunks and then distributed. This may already include many types of applications including P2P applications, IPTV, and VoD. Instead of always transferring data directly from a source-owner peer to a requesting peer, the source-owner peer can store and manage its content on its in-network storage. The requesting peer can get the address of the in-network storage pertaining to the source-owner peer and retrieve data from the storage.

This draft enumerates and explains the rationale behind SPECIFIC requirements on the protocol design and on any data store implementation that may be used to implement DECADE servers that should be considered during the design and implementation of a DECADE system. As such, it DOES NOT include general guiding principals. General design considerations, explanation of the problem being addressed, and enumeration of the types of applications to which DECADE may be suited is not considered in this document. For general information, please see the problem statement [I-D.ietf-decade-problem-statement] and architecture drafts.

This document enumerates the requirements to enable target applications to utilize in-network storage. In this context, using storage resources includes not only basic capabilities such as storing and retrieving data, and managing data, but also (1) controlling access by peers with which it is sharing data and (2) controlling the resources used by remote peers when accessing data.

This document will be updated to track revisions to the problem statement.

Editors Note: Currently the Architecture document is a WG milestone, but not yet a WG item. We have made the assumption that there will be a WG item meeting this milestone going forward.

2. Terminology and Concepts

This document uses terms defined in [I-D.ietf-decade-problem-statement]. In particular, IAP refers to the In-network storage Access Protocol, which is the protocol used to communicate between a DECADE client and DECADE server (in-network storage) for access control and resource control.

This document also defines additional terminology:

Target Application: An application (typically installed at end-hosts) with the ability to explicitly control usage of network and/or storage resources to deliver contents to a large number of users. Such applications distribute large contents (e.g., a large file, or video stream) by dividing the contents into smaller blocks for more flexible distribution (e.g., multipath). The distributed content is typically immutable (though it may be deleted). We use the term Target Application to refer to the type of applications that are explicitly (but not exclusively) supported by DECADE.

3. Requirements Structure

The DECADE protocol is intended to sit between P2P applications and a back-end storage system. In the development of DECADE, it must be made clear that the intention is to NOT develop yet another storage system, but rather to create a protocol that enables P2P applications to make use of storage within the network, leaving specific storage system considerations to the implementation of the DECADE servers as much as possible. For this reason, we have divided the requirements into three categories:

- General Principles: Overall requirements that a DECADE system must conform to.

- Protocol Requirements: Protocol requirements for Target Applications to make use of in-network storage within their own data dissemination schemes. Development of these requirements is guided by a study of data access, search and management capabilities used by Target Applications.

- Storage Requirements: Functional requirements necessary for the back-end storage system employed by the DECADE server. Development of these requirements is guided by a study of the data access patterns used by Target Applications.

It should also be made clear that the approach is to make DECADE a simple protocol, while still enabling its usage within many P2P applications. For this reason, and to further reinforce the distinction between DECADE and a storage system, in some cases we also highlight the non-requirements of the protocol. These non-requirements are intended to capture behaviors that will NOT be assumed to be needed by DECADE's Target Applications and hence not present in the DECADE protocol.

Finally, some implementation considerations are provided, which while

strictly are not requirements, are intended to provide guidance and highlight potential points of concern that need to be considered by the protocol developers, and later by implementors.

4. Protocol Requirements

4.1. Requirements

4.1.1. Overall Protocol Requirements

4.1.1.1. Application-independent API

REQUIREMENT(S): The DECADE IAP MUST provide a simple, application-independent API for P2P applications to access in-network storage.

RATIONALE: Since the majority of existing P2P application APIs don’t support in-network storage management, new application-independent API must be introduced. The API should be simple to encourage adoption, as well as to ensure that a minimum set of functions, and not an entire network storage system is implemented.

4.1.1.2. Cross-platform Access

REQUIREMENT(S): If DECADE supports the ability to store metadata associated with data objects, the DECADE protocol(s) MUST transmit any metadata using an operating system-independent and architecture-independent format.

RATIONALE: If DECADE supports the possibility for storing metadata (e.g., a description, uploaded date, or object size), a possible use for the metadata is to help a DECADE client locate a desired data object. Data objects may be stored by DECADE clients running on various platforms. To enable metadata to be readable regardless of its source it must be transmitted to and from the DECADE server in a standard format.

4.1.1.3. Connectivity Concerns

4.1.1.3.1. NATs and Firewalls

REQUIREMENT(S): DECADE SHOULD be usable across firewalls and NATs without requiring additional network support (e.g., Application-level Gateways).

RATIONALE: Firewalls and NATs are widely used in the Internet today, both in ISP networks and within households. Deployment of DECADE must not require modifications to such devices (beyond, perhaps, reconfiguration).

4.1.1.3.2. Connections to Clients

REQUIREMENT(S): DECADE SHOULD NOT require that network connections be made to DECADE clients (e.g., from a server to a DECADE client or from a DECADE client to another DECADE client).

RATIONALE: Many household networks and operating systems have firewalls and NATs configured by default. To ease deployment by avoiding configuration changes and help mitigate security risks, DECADE should not require clients to listen for any incoming network connections (beyond what is required by any other already-deployed application).

4.1.1.4. Error and Failure Conditions

4.1.1.4.1. Overload Condition

REQUIREMENT(S): In-network storage, which is operating close to its capacity limit (e.g., too busy servicing other requests), MUST be able to reject requests.

RATIONALE: When in-network storage is operating at a limit where it may not be able to process additional requests, it should not be required to generate responses to such additional requests. Forcing the in-network storage to do so can impair its ability to service existing requests.

4.1.1.4.2. Insufficient Resources

REQUIREMENT(S): DECADE MUST support an error condition indicating to a DECADE client that resources (e.g., storage space) were not available to service a request (e.g., storage quota exceeded when attempting to store data).

RATIONALE: The currently-used resource levels within the in-network storage are not locally-discoverable, since the resources (disk, network interfaces, etc) are not directly attached. In order to allocate resources appropriately amongst peers, a client must be able to determine when resource limits have been reached. The client can then respond by explicitly freeing necessary resources or waiting for such resources to be freed.

4.1.1.4.3. Unavailable and Deleted Data

REQUIREMENT(S): DECADE MUST support error conditions indicating that (1) data was rejected from being stored, (2) deleted, or (3) marked unavailable by a storage provider.

RATIONALE: Storage providers may require the ability to (1) avoid storing, (2) delete, or (3) quarantine certain data that has been identified as illegal (or otherwise prohibited). DECADE does not indicate how such data is identified, but applications using DECADE should not break if a storage provider is obligated to enforce such policies. Appropriate error conditions should be indicated to applications.

4.1.2. Transfer and Latency Requirements

4.1.2.1. Low-Latency Access

REQUIREMENT(S): DECADE SHOULD provide "low-latency" access for application clients. DECADE MUST allow clients to specify at least two classes of services for service: lowest possible latency and latency non-critical.

RATIONALE: Some applications may have requirements on delivery time (e.g., live streaming). The user experience may be unsatisfactory if the use of in-network storage results in lower performance than connecting directly to peers over a low-speed, possibly congested uplink. Additionally, the overhead required for control-plane operations in DECADE must not cause the latency to be higher than for a low-speed, possibly congested uplink. While it is impossible to make a guarantee that a system using in-network storage will always outperform a system that does not for every transfer, the overall performance of the system should be improved compared with direct connections, even considering control overhead.

4.1.2.2. Indirect Transfer

REQUIREMENT(S): DECADE MUST allow a user’s in-network storage to directly fetch from other user’s in-network storage.

RATIONALE: As an example, a requesting peer may get the address of the storage pertaining to the source-owner peer and then tell its own in-network storage to fetch the content from the source-owner’s in-network storage. This helps to avoid extra transfers across ISP network links where possible.

4.1.2.3. Data Object Size

REQUIREMENT(S): DECADE MUST allow for efficient data transfer of small objects (e.g., 16KB) between a DECADE client and in-network storage with minimal additional latency required by the protocol.

RATIONALE: Though P2P applications are frequently used to share large amounts of data (e.g., continuous streams or large files), the data itself is typically subdivided into smaller chunks that are transferred between peers. Additionally, the small chunks may have requirements on delivery time (e.g., in a live-streaming application). DECADE must enable data to be efficiently transferred amongst peers at this granularity.

4.1.2.4. Communication among In-network Storage Elements

REQUIREMENT(S): DECADE SHOULD support the ability for two in-network storage elements in different administrative domains to store and/or retrieve data directly between each other. If such a capability is supported, this MAY be the same (or a subset or extension of) as the IAP used by clients to access data.

RATIONALE: Allowing server-to-server communication can reduce latency in some common scenarios. Consider a scenario when a DECADE client is downloading data into its own storage from another client’s in-network storage. One possibility is for the client to first download the data itself, and then upload it to its own storage. However, this causes unnecessary latency and network traffic. Allowing the data to be downloaded from the remote in-network storage into the client’s own in-network storage can alleviate both.

4.1.3. Data Access Requirements

4.1.3.1. Reading/Writing Own Storage

REQUIREMENT(S): DECADE MUST support the ability for a DECADE client to read data from and write data to its own in-network storage.

RATIONALE: Two basic capabilities for any storage system are reading and writing data. A DECADE client can read data from and write data to in-network storage space that it owns.

4.1.3.2. Access by Other Users

REQUIREMENT(S): DECADE MUST support the ability for a user to apply access control policies to users other than itself for its storage. The users with whom access is being shared can be under a different administrative domain than the user who owns the in-network storage. The authorized users may read from or write to the user's storage.

RATIONALE: Peers in a P2P application may be located across multiple ISPs under multiple administrative domains. Thus, to be useful by P2P applications, DECADE allows a user to specify access control policies for users that may or may not be known to the user's storage provider.

4.1.3.3. Negotiable Data Protocol

REQUIREMENT(S): DECADE MUST support the ability for a DECADE client to negotiate with its In-network storage about which protocol it can use to read data from and write data to its In-network storage.

RATIONALE: Since typical data transport protocols (e.g., NFS and WebDAV) already provide read and write operations for network storage, it may not be necessary for DECADE to define such operations in a new protocol. However, because of the particular application requirements and deployment considerations, different applications may support different protocols. Thus, a DECADE client must be able to select an appropriate protocol also supported by the in-network storage. This requirement also follows as a result of the requirement of Separation of Control and Data Operations (Section 4.1.3.4).

4.1.3.4. Separation of Data Operations from Application Control

REQUIREMENT(S): The DECADE IAP MUST only provide a minimal set of core operations to support diverse policies implemented and desired by Target Applications.

RATIONALE: Target Applications support many complex behaviors and diverse policies to control and distribute data, such as (e.g., search, index, setting permissions/passing authorization tokens). Thus, to support such Target Applications, these behaviors must be logically separated from the data transfer operations (e.g., retrieve, store). Some minimal overlap (for example obtaining credentials needed to encrypt or authorize data transfer using control operations) may be required to be directly supported by DECADE.

4.1.4. Data Management Requirements

4.1.4.1. Agnostic of reliability

REQUIREMENT(S): DECADE SHOULD remain agnostic of reliability/ fault-tolerance level offered by storage provider.

RATIONALE: Providers of a DECADE service may wish to offer varying levels of service for different applications/users. However, a single compliant DECADE client should be able to use multiple DECADE services with differing levels of service.

4.1.4.2. Time-to-live for Stored Data

REQUIREMENT(S): DECADE MUST support the ability for a DECADE client to indicate a time-to-live value (or expiration time) indicating a length of time until particular data can be deleted by the in-network storage element.

RATIONALE: Some data stored by a DECADE client may be usable only within a certain window of time, such as in live-streaming P2P applications. Providing a time-to-live value for stored data (e.g., at the time it is stored) can reduce management overhead by avoiding many 'delete' commands sent to in-network storage. The in-network storage may still keep the data in cache for bandwidth optimization. But this is guided by the privacy policy of the DECADE provider.

4.1.4.3. Offline Usage

REQUIREMENT(S): DECADE MAY support the ability for a user to provide authorized access to its in-network storage even if the user has no DECADE applications actively running or connected to the network.

RATIONALE: If an application desires, it can authorize peers to access its storage even after the application exits or network connectivity is lost. An example use case is mobile scenarios, where a client can lose and regain network connectivity very often.

4.1.5. Resource Control

4.1.5.1. Multiple Applications

REQUIREMENT(S): DECADE SHOULD support the ability for users to define resource sharing policies for multiple applications being run/managed by the user.

RATIONALE: A user may own in-network storage and share the in-network storage resources amongst multiple applications. For example, the user may run a video-on-demand application and a live-streaming (or even two different live-streaming applications) application which both make use of the user's in-network storage. The applications may be running on different machines and may not directly communicate. Thus, DECADE should enable the user to determine resource sharing policies between the applications.

One possibility is for a user to indicate the particular resource sharing policies between applications out-of-band (not using a standard protocol), but this requirement may manifest itself in passing values over IAP to identify individual applications. Such identifiers can be either user-generated or server-generated and do not need to be registered by IANA.

4.1.5.2. Per-Peer, Per-Data Control

REQUIREMENT(S): A DECADE client MUST be able to assign resource quotas to individual peers for reading from and writing particular data to its in-network storage within a particular range of time. The DECADE server MUST enforce these constraints.

RATIONALE: P2P applications can rely on control of resources on a per-peer or per-data basis. For example, application policy may indicate that certain peers have a higher bandwidth share for receiving data. Additionally, certain data (e.g., chunks) may be distributed with a higher priority. As another example, when allowing a remote peer to write data to a user's in-network storage, the remote peer may be restricted to write only a certain amount of data. Since the client may need to manage multiple peers accessing its data, it should be able to indicate the time over which the granted resources are usable. For example, an expiration time for the access could be indicated to the server after which no resources are granted (e.g., indicate error as access denied).

4.1.5.3. Server Involvement

REQUIREMENT(S): A DECADE client MUST be able to indicate, without contacting the server itself, resource control policies for peers' requests.

RATIONALE: One important consideration for in-network storage elements is scalability, since a single storage element may be used to support many users. Many P2P applications use small chunk sizes and frequent data exchanges. If such an application employed resource control and contacted the in-network storage element for each data exchange, this could present a scalability challenge for the server as well as additional latency for clients.

An alternative is to let requesting users get the resource control policies and users can then present the policy to the storage directly. This can result in reduced messaging handled by the in-network storage.

4.1.6. Authorization

4.1.6.1. Per-Peer, Per-Data Read Access

REQUIREMENT(S): A DECADE Client MUST be able to authorize individual peers to read particular data stored on its in-network storage.

RATIONALE: A P2P application can control certain application-level policies by sending particular data (e.g., chunks) to certain peers. It is important that peers not be able to circumvent such decisions by arbitrarily reading any currently-stored data in in-network storage.

4.1.6.2. Per-User Write Access

REQUIREMENT(S): A DECADE Client MUST be able to authorize individual peers to store data into its in-network storage.

RATIONALE: The space managed by a user in in-network storage can be a limited resource. At the same time, it can be useful to allow remote peers to write data directly to a user's in-network storage. Thus, a DECADE client should be able to grant only certain peers this privilege.

Note that it is not (currently) a requirement to check that a peer stores a particular set of data (e.g., the check that a remote peer writes the expected chunk of a file). Enforcing this as a requirement would require a client to know which data is expected (e.g., the full chunk itself or a hash of the chunk) which may not be available in all applications. Checking for a

particular hash could be considered as a requirement in the future that could optionally be employed by applications.

4.1.6.3. Authorization Checks

REQUIREMENT(S): In-network storage MUST check the authorization of a client before it executes a supplied request. The in-network storage MAY use optimizations to avoid such authorization checks as long as the enforced permissions are the the same.

RATIONALE: Authorization granted by a DECADE client are meaningless unless unauthorized requests are denied access. Thus, the in-network storage element must verify the authorization of a particular request before it is executed.

4.1.6.4. Credentials Not IP-Based

REQUIREMENT(S): Access MUST be able to be granted on other credentials than the IP address

RATIONALE: DECADE clients may be operating on hosts without constant network connectivity or without a permanent attachment address (e.g., mobile devices). To support access control with such hosts, DECADE servers must support access control policies that use information other than IP addresses.

4.1.6.5. Server Involvement

REQUIREMENT(S): A DECADE client MUST be able to indicate, without contacting the server itself, access control policies for peers' requests.

RATIONALE: See discussion in Section 4.1.5.3.

5. Storage Requirements

5.1. Requirements

5.1.1. Explicit Deletion of Stored Data

REQUIREMENT(S): DECADE MUST support the ability for a DECADE client to explicitly delete data from its own in-network storage.

RATIONALE: A DECADE client may continually be writing data to its in-network storage. Since there may be a limit (e.g., imposed by the storage provider) to how much total storage can be used, some data may need to be removed to make room for additional data. A

DECADE client should be able to explicitly remove particular data. This may be implemented using existing protocols.

5.1.2. Multiple writing

REQUIREMENT(S): DECADE MUST NOT allow multiple writers for the same object. Implementations raise an error to one of the writers.

RATIONALE: This avoids data corruption in a simple way while remaining efficient.

5.1.3. Multiple reading

REQUIREMENT(S): DECADE MUST allow for multiple readers for an object..

RATIONALE: One characteristic of P2P applications is the ability to upload an object to multiple peers. Thus, it is natural for DECADE to allow multiple readers to access the content concurrently.

5.1.4. Reading before completely written

REQUIREMENT(S): DECADE MAY allow readers to read from objects before they have been completely written.

RATIONALE: Some P2P systems (in particular, streaming) can be sensitive to latency. A technique to reduce latency is to remove store-and-forward delays for data objects (e.g., make the object available before it is completely stored). Appropriate handling for error conditions (e.g., a disappearing writer) needs to be specified.

5.1.5. Writing model

REQUIREMENT(S): DECADE MUST provide at least a writing model (while storing an object) that appends data to data already stored.

RATIONALE: Depending on the object size (e.g., chunk size) used by a P2P application, the application may need to send data to the DECADE server in multiple packets. To keep implementation simple, the DECADE must at least support the ability to write the data sequentially in the order received. Implementations MAY allow application to write data in an object out-of-order (but MAY NOT overwrite ranges of the object that have already been stored).

5.1.6. Storage Status

REQUIREMENT(S): A DECADE client MUST be able to retrieve current resource usage (including list of stored data) and resource quotas on its in-network storage.

RATIONALE: The resources used by a client are not directly-attached (e.g., disk, network interface, etc). Thus, the client cannot locally determine how such resources are being used. Before storing and retrieving data, a client should be able to determine which data is available (e.g., after an application restart). Additionally, a client should be able to determine resource availability to better allocate them to remote peers.

5.2. Non-Requirements

5.2.1. No ability to update

REQUIREMENT(S): DECADE SHOULD NOT provide ability to update existing objects. That is, objects are immutable once they are stored.

RATIONALE: Reasonable consistency models for updating existing objects would significantly complicate implementation (especially if implementation chooses to replicate data across multiple servers). If a user needs to update a resource, it can store a new resource and then distribute the new resource instead of the old one.

6. Implementation Considerations

The intent of this section is to collect discussion items and implementation considerations that have been discovered as this requirements document has been produced. The content of this section will be migrated to an appropriate place as the document and the Working Group progress.

6.1. Resource Scheduling

The particular resource scheduling policy may have important ramifications on the performance of applications. This document has explicitly mentioned simultaneous support for both low-latency applications and latency-tolerant applications.

Denial of Service attacks may be another risk. For example, rejecting new requests due to overload conditions may introduce the ability to perform a denial of service attack depending on a particular DECADE server’s scheduling implementation and resource

allocation policies.

6.2. Removal of Duplicate Records

There are actually two possible scenarios here. The first is the case of removing duplicates within one particular DECADE server (or logical server). While not a requirement, as it does not impact the protocol and is technically not noticeable on message across the wire, a DECADE server may implement internal mechanisms to monitor for duplicate records and use internal mechanisms to prevent duplication of internal storage.

The second scenario is removing duplicates across a distributed set of DECADE servers. This is a more difficult problem, and if the group decides to support this capability, it may require protocol support. See Section 7.2 for more details.

7. Discussion and Open Issues

7.1. Discussion

Sometimes, several logical in-network storages could be deployed on the same physical network device. In this case, in-network storages on the same physical network device can communicate and transfer data through internal communication messages. However in-network storages deployed on different physical network devices SHOULD communicate with in-network storage Access Protocol (IAP).

To provide fairness among users, the in-network storage provider should assign resource (e.g., storage, bandwidth, connections) quota for users. This can prevent a small number of clients from occupying large amounts of resources on the in-network storage, while others starve.

7.2. Open Issues

Gaming of the Resource Control Mechanism: There has been some discussion of how applications may be able game the scheduling system by manipulating the resource control mechanism, for example by specifying many small peers to increase total throughput. This is a serious concern, and we need to identify specific requirements on the protocol (hopefully independent of particular scheduling/resource control schemes) to help address this.

Discovery: There needs to be some mechanism for a user to discover that there is a DECADE service available for their use, and to locate that server. This is particularly important in the case of mobile applications, since the actual servers that are available at any given time may differ. However, the specifics of what mechanisms (DHCP, HTTP page, etc.) have not been discussed, or even if the protocol should specify one or leave it as an implementation detail. This needs to be defined, and specific requirements formulated if needed.

Removal of Duplicate Records Across Servers: If the group wishes to allow for automated mechanisms to remove duplicates across a number of separate servers, some protocol support may need to be added. In the case of removing duplicates within a single (logical) DECADE server, this is simply an implementation concern. See Section 6 for more details.

8. Security Considerations

Authorization for access to in-network storage is an important part of the requirements listed in this document. Authorization for access to storage resources and the data itself is important for users to be able to manage and limit distribution of content. For example, a user may only wish to share particular content with certain peers.

If the authorization technique implemented in DECADE passes any private information (e.g., user passwords) over the wire, it MUST be passed in a secure way.

9. IANA Considerations

There are no IANA considerations with this document.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

10.2. Informative References

[I-D.ietf-decade-problem-statement] Yongchao, S., Zong, N., Yang, Y., and R. Alimi, "DECoupled

Application Data Enroute (DECADE) Problem Statement", draft-ietf-decade-problem-statement-00 (work in progress), August 2010.

[LLSB08] Dave Levin, Katrina LaCurts, Neil Spring, Bobby Bhattacharjee., "BitTorrent is an Auction: Analyzing and Improving BitTorrent's Incentives", In SIGCOMM 2008.

[PPLive] "PPLive", http://www.pplive.com.

Appendix A. Acknowledgments

We would also like to thank Haibin Song for substantial contributions to earlier versions of this document. We would also like to thank Reinaldo Penno, Alexey Melnikov, Rich Woundy, Ning Zong, Roni Even, David McDysan, Boerje Ohlman and Dirk Kutscher for contributions (including some text used verbatim) and general feedback.

Authors' Addresses

Yingjie Gu
Huawei
No. 101 Software Avenue
Nanjing, Jiangsu Province 210012
P.R.China

Phone: +86-25-56624760
Email: guyingjie@huawei.com

David A. Bryan
Cogent Force, LLC / Huawei

Email: dbryan@ethernot.org

Yang Richard Yang
Yale University

Email: yry@cs.yale.edu

Richard Alimi
Google

Email: ralimi@google.com