Network Working Group
Internet-Draft
Intended status: Experimental
Expires: June 16, 2014

W. Ivancic
NASA GRC
W. Eddy
MTI Systems
A. Hylton
D. Iannicca
J. Ishac
NASA GRC

December 13, 2013

Store, Carry and Forward Requirements and Expectations
draft-ivancic-scf-requirements-expectations-01

Abstract

This document describes the requirements for a Store, Carry and Forward (SCF) protocol, and the expectations placed upon the SCF agents and SCF applications.

The Requirements and Expectations document is one of three that fully describe the SCF system. The other two are the problem statement and the testing requirements document.

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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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 June 16, 2014.

Table of Contents

1. Terminology

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 RFC 2119 [RFC2119].

In developing this document, we have intentionally avoided some terminology used by other protocols - particularly store-and-forward protocols - to avoid biases and confusion that may otherwise ensue.

- Container - metadata describing the characteristics of application /user data to be transported over the network and its forwarding requirements as well as a mandatory checksum of that information (Shipping Label) and the application/user data to be transported over the network as well as an optional checksum of that information (Shipping Container). Containers may consists of one or more sub-containers.

- Container Aggregation - The process of organizing one or multiple containers as sub-containers inside another larger container.

- Container Deaggregation - The process of removing one or more sub-containers from a larger container. This differs from

fragmentation because rather than creating new containers, deaggregation operates on existing sub-containers.

- Container Fragmentation - The process of dividing a single container's contents into multiple new containers which will need to be eventually reassembled back into the original container before delivery to the application.

- Container Reassembly - The process of recombining the contents of multiple containers into a single container that they were originally part of, and that needs to be delivered to the application intact.

- Delay - propagation delay between SCF agents. Delay does not include disconnection time.

- Disruption - a relatively short period of disconnection within an otherwise well-connected network (e.g. a loss of connectivity in the range of seconds to perhaps minutes)

- Disconnection - a relatively long period where communication between a significant proportion of hosts is not possible for various reasons (e.g. due to the inability to close a radio link)

- Metadata - synonymous with a Container's Label

- SCF - Store, Carry and Forward

- SF - Store-and-Forward, or "store and forward" as used generically in other literature (where the presence of hyphenation varies)

- SCF Agent - a protocol instance providing SCF services to an upper-layer user/application

- Shipping Container - the application/user data to be transported over the network as well as an optional checksum of that information.

- Shipping Label - metadata describing the characteristics of a container and its forwarding requirements, as well as a mandatory checksum of that information.

- Sub-Container - A small container residing inside a larger container.

- Transport Capacity - (as a first order approximation) the combination of bandwidth and contact time.

2. Introduction

This document describes the requirements for a Store, Carry and Forward (SCF) protocol, and the expectations placed upon the SCF agents and SCF applications.

This Requirements and Expectations document is one of three that fully describe the SCF system. The other two are the problem statement and the testing requirements document.

As background, the SCF Problem Statement [I-D.ivancic-scf-problem-statement] is suggested reading. The SCF Problem Statement describes the core SCF problem and gives an assessment of the potential to use existing technologies as solutions. In addition, it provides a number of SCF deployment scenarios.

3. Design Considerations

The following design considerations are explicitly stated with a goal of keeping the protocol simple. (Anyone can make things more complicated!)

- Do not overload the relaying protocol. Keep It Simple.
  - Keep network management functions separate from the relaying protocol.
  - Content Based Networking is different than SCF. SCF can be used to move content, but should not be considered an in-network content store.
  - Rationale: Separation allows for independent development and optimizations.
- The SCF protocol MUST NOT rely on time synchronization between applications or relaying agents.
  - It is very difficult, if not impossible, to synchronize disconnected networks. Furthermore, if the protocol requires synchronization to work, it can never be used to synchronize a system - even for coarse synchronization. In addition, reliance on absolute time creates security vulnerabilities.
- Protocol options make interoperability hard. Options are often used as a placeholder for fixing a bad design.
- Naming and addressing are key to security and scalability.

- Addressing should be topological (location dependent). This enables aggregation of the routing locator space and improves scalability for the routing system.

- Strive to limit the size of the forwarding table. Large forwarding tables place a great burden on the SCF processing system. There is always a limit for any CPU. The further one is removed from reaching that limit, the better.

4. Protocol Requirements

The following are a list of requirements for a SCF Protocol. The requirements are specifically written in general terms. The intent is to identify what is required, not how to solve the requirement. The requirements are in no particular order of precedence, but are numbered in order to aid in referencing for discussion.

SCF Agent Requirements and Agent operation expectations have been intentionally separated, as the SCF Agent requirements are more policy-based than protocol-based. However, one needs to understand both in order to effectively implement the SCF protocol.

PROTOCOL-REQ1: The SCF relaying protocol MUST be able to handle data sets that are very small (several bytes) and very large (several gigabytes).

* Rationale: SCF is useful for very small, simple, low-power, low-processing minimally-capable sensor systems, as well as for more capable high-end data mules. In a simple sensor-web one may be moving extremely small containers of information on the order of bytes; whereas later onward delivery by a data mule may be moving containers containing gigabytes of data.

PROTOCOL-REQ2: The SCF protocol MUST permit SCF agents to be able to aggregate containers.

* Rationale: Aggregation will reduce forwarding table size and enable pre-processing of forwarding queues. Without aggregation, the SCF agent processing capabilities can be quickly overwhelmed - particularly for a large number of small containers - even if those containers are destined for the same location. Aggregation and Deaggregation enable efficient shipping of information through a SCF network from a variety sources to a common destination by continually

recombining containers as the information moves through the relay network.

PROTOCOL-REQ3: The SCF protocol MUST permit SCF agents to be able to deaggregate containers.

* Rationale: Deaggregation allows subcontainers to be removed from larger aggregated containers and either shipped separately due to contact limitations, or spread out to multiple other relaying SCF agents in parallel.

PROTOCOL-REQ4: The SCF protocol MUST permit SCF agents to be able to reactively fragment a container.

* Rationale: It is often not possible to determine how long a contact time will be between SCF agents. In such instances, which may be the norm, one cannot determine the transport capacity and may only be able to transfer a portion of a container before the contact ends. In order to improve transport efficiency and effectively utilize the radio link, one should not have to retransmit what has already been received.

PROTOCOL-REQ5: The SCF protocol SHOULD permit SCF agents to be able to proactively fragment a container.

* Rationale: It may be possible to a priori know the transport capacity between SCF agents. In such instances, one may determine that an entire container could only be transfer between agents if it is divided into smaller units. In other cases, a SCF agent may wish to limit the size of containers as a matter of policy. In either of these cases, proactive fragmentation would be useful. However, it would be more desirable for the application to limit the size of the container if at all possible, rather than having this done by the SCF agent.

PROTOCOL-REQ6: An SCF protocol MUST implement reliability on the Shipping Label, and a damaged Shipping Label MUST NOT propagate to further SCF agents or have its container further propagated or delivered to applications.

* Rationale: An SCF agent needs to be able to determine if the shipping label is damaged in order to prevent misdelivery of data, waste of resources (storage, battery, network capacity, etc.), and other suboptimal results of operating on flawed forwarding information.

PROTOCOL-REQ7: An SCF protocol MUST be capable of implementing reliability on the Shipping Container.

* Rationale: An SCF agent must be able to determine if the shipping container is damaged. A damaged Shipping Container MAY be discarded along with the associated Shipping Label. Note, user of reliability on the Shipping Container is not mandatory, but the ability to have such capability is.

PROTOCOL-REQ8: The SCF protocol security implementation MUST authenticate the Shipping Label independent of the Shipping Container.

* Rationale: The ability to authenticate data sources and control resource usage early on is critical to reducing vulnerabilities to denial-of-service attacks, whether intentional or unintentional. For large containers, if the entire container had to be received and processed before a determination could be made as the the source of the Container, multiple resources would be wasted including bandwidth, processing cycles and storage.

PROTOCOL-REQ9: The SCF protocol security implementation MUST work with reactive fragmentation.

* Rationale: For medium- and high-end SCF systems, the ability to authenticate data sources and control resource usage is critical. Likewise, reactive fragmentation may be quite common and has been shown to be invaluable in transporting large data sets [Multi-Terminal][Multi-Terminal].

PROTOCOL-REQ10: The SCF protocol security implementation MUST have a security policy database to control resources.

* Rationale: Once a data source is authenticated, the security policy will determine what type and amount of resources that source can use, as well as possibly the forwarding priorities. It is anticipated that SCF systems will have different peering arrangements with different entities (e.g. peers, groups, or organizations).

PROTOCOL-REQ11: The SCF protocol MUST be able to separate the Shipping Label from the Shipping Container

* Rationale: The SCF agent must be able to determine whether or not it wishes to receive or store a container prior to receiving the entire container. This reduces denial-of-service vulnerabilities and enables efficient use of radio and system resources.

PROTOCOL-REQ12: The SCF protocol MUST have a mechanism that enables data to die naturally.

* Rationale: Data should die naturally to avoid routing loops at the SCF layer. Routing loops at the SFC layer cannot be eliminated by lower-layer mechanisms (i.e. and IPv6 hop count will not correct an SCF routing loop).

PROTOCOL-REQ13: The SCF protocol MUST have a naming mechanism that specifies the application and instance to which the content is bound.

* Rationale: This naming mechanism is necessary in order for the SCF agent end system to pass the Shipping Container content to the proper instance of a given application since multiple instances may be invoked at any given time.

PROTOCOL-REQ14: The SCF protocol MUST have a Quality-of-Service (QOS) mechanism to expedite forwarding and to handle storage lifetimes.

* Rationale: Past experiences with other store-and-forward technologies such as DTN [RFC5050] have shown that it is very difficult for many applications to determine how long the useful life of data is. Rather, bundle lifetimes have been set either arbitrarily or rather coarsely (e.g. short, medium, forever) - see Bundle

Lifetimes [1]. QOS will enable and SCF to expedite, store and purge data on a much more coarse scale than the use of absolute or relative time. Such QOS policies could be a configuration setting within individual SCF agents, or within an SCF network. This greatly simplifies the protocol processing as well as aggregation and deaggregation of containers.

PROTOCOL-REQ15: The SCF protocol MUST have a mechanism for a receiving system to acknowledge reception of a container from the sending system (i.e. hop-by-hop acknowledgements).

* Rationale: This allows a sending system to release the container if it so desires, thereby improving resource usage.

PROTOCOL-REQ16: The SCF protocol MUST have a mechanism to notify a sender that the container will not be processed.

* Rationale: If the agent's policy states "Do not accept" for any possible reason, it is important to inform the sender as soon as possible (ASAP) that the container will not be accepted, to allow the sender to stop transmission and determine a different route for that container. Note, there may be security reasons not to provide this information, but in generally such a response SHOULD be sent.

PROTOCOL-REQ17: The SCF protocol SHOULD have a mechanism that enables one to identify fresh versus stale content for a given flow.

* Rationale: Fresh data is often of far greater value than stale data. The ability to identify fresh data and either replace the stale data with fresh, or send the fresh data first, is highly desirable in order to optimize resource usage - particularly storage and bandwidth.

* Comment: There appears to be a desire in many instances to proactively create fixed bundle sizes in DTN and then what the application to put them back in order. With proactive fragmentation, this is possible and there is a mechanism to allow reordering. With straight

bundling, this is problematic as there is no such formalized standard sequencing (i.e. sequence numbers).

5. Agent Requirements and Expectations

The following are a list of requirements for an SCF Agent. The requirements are in no particular order of precedence, but are numbered in order to aid in referencing for discussion.

AGENT-REQ1: An SCF agent MUST NOT be required to implement SCF security. Security must be optional.

* Rationale: Simple devices such as sensors may wish to utilize the SCF protocol, but have neither the need for security, nor the processing capability to implement SCF security.

AGENT-REQ2: An SCF agent MAY implement reliability on the Shipping Container.

* Rationale: An application may or may not care if the contents of the container arrive without modification. For example, protecting a large image file from a bit flip may not be considered as important as reducing the processing overheard of creating and checking reliability on the Shipping Container.

AGENT-REQ3: An SCF agent MUST hold onto a container until it can either be transferred or QOS policy indicates its useful lifetime has expired or storage resources reach a level that requires some purging of containers based on policy.

* Rationale: The sender expects the receiver to do its best to forward the container, and MAY release the container upon notification from the receiver that the container has been received. If the receiver does not plan to hold onto the container, it SHOULD send a notification to the sender stating such.

AGENT-REQ4: An SCF agent MAY remove a container once it receives notification from the next hop SCF that the container has been delivered.

* Rationale: The ability to release containers enables efficient use of storage resources. Note, some

deployments and some routing protocols MAY mandate that the agent retain a container even after a successful transfer. In such deployments, containers would likely be removed based on a retention policy which may be based on QOS.

AGENT-REQ5: An SCF agent SHOULD NOT accept a container if it has no intention of giving a best effort to forward the container.

* Rationale: The sending SCF's default expectation is that, if accepted, the receiving SCF will do its best to forward the container. This allows the sending SCF, if so desired, to purge the container from its storage with some confidence that the container will be delivered.

AGENT-REQ6: An SCF agent SHOULD implement a policy system that controls resources. Such a policy system MAY include the filters described below.

* Rationale: Resources including bandwidth and storage storage are precious commodities that need to be controlled. Various SCF deployments are expected to have vastly different capabilities and needs. For example, an SCF science sensorweb may have not need for security, while implementing a policy that basically says "Accept Everything", because all containers are know a priori to be small and the deployment is a closed network. Other deployments may consist of high-end SCF agents supporting multiple organizations and transferring and storing Gigabytes or more of information. The ability to tune the policies to fit the deployment makes such deployments realizable.

(a) What volume (size) container will be accepted.

+ Rationale: Storage resources are not infinite. It is likely policy will limit container size and/or overall memory allocations per source, address range, or other filters. In addition, some SCF agents may have limited processing and not be willing or capable of handling extremely large containers

(b) What sources are permitted to use resources and how much resource?

+ Rationale: Resources including bandwidth and storage are precious. It is anticipated that peering arrangements will exist to populate this database. Not every source may be permitted to utilize the resources.

(c) What destinations are permitted to use resources and how much resource?

+ Rationale: Resources including bandwidth and storage are precious. It is anticipated that peering arrangements will exist to populate this database. Not every destination may be permitted to utilize the resources.

(d) Prioritized container delivery.

+ Rationale: It is anticipated that peering arrangements will exist to populate this database. Some peers are likely to be given preferential treatment, while others may be serviced only after all commitments have been met, regardless of QoS (e.g. the general's containers are processed before the private's; the organization who owns the SCF agent gets preferential treatment over all other organizations.

(e) A mapping of QoS to retention lifetime and forwarding priority.

+ Rationale: A coarse-grained retention policy is anticipated. Such granularity may be minutes, hours, days, forever (until resources become scarce and memory must be released). This alleviates the need for actual lifetime settings within the SCF protocol and allows various deployments to be uniquely configured.

AGENT-REQ7: If security is implemented, when coming in contact with one another, adjacent SCF agents MUST minimally be able to identify one another securely and prove that they can be trusted as relays for a given destination application.

* Rationale: Such a mechanism is necessary to prevent hijacking of information. Also, aggregation and deaggregation may be implemented along a container's route. Trust between forwarding agents must be established to enable this.

AGENT-REQ8: SCF Agents MUST be able to indicate (or deny) forwarding of individual containers, based on exchanging their shipping labels only.

* Rationale: This allows for efficient use of RF resources as well as reducing DOS vulnerabilities. It the SCF Agent had to process an entire Container prior to denying acceptance, a malicious entity could easily perform a DOS attack by sending extremely large containers which would have to be stored and processed by the receiving SCF prior to rejection.

AGENT-REQ9: SCF Agents MAY notify applications of pending received data.

* Rationale: If the SCF agent knows it is bound to an application and can notify the application of pending received data, this could improve the application's operations.

6. Application Requirements and Expectations

APPLICATION-REQ1: Applications SHOULD be designed to operate in a disconnected systems.

* Rationale: Applications that have been designed assuming a connected network are likely to break.

* Rationale: Streaming may work, but should not be encouraged as streaming applications with reasonably significant volumes of traffic are likely to only work for connected systems or very short fades. Such systems probably do not need SCF.

APPLICATION-REQ2: Applications MUST be able to select their own globally-unique identifiers and notify SCF agents of them, along with providing proof of ownership.

* Rational:

APPLICATION-REQ3: Applications MUST be able to poll a SCF agent for pending received data.

* Rationale: Applications are the only ones that can keep track of the shared state between sender and receiver. The Application cannot

expect lower layers such as the SCF agent to fully understand its needs.

* Rationale: This eliminates putting undue burden on the SCF, and ensures interoperability to specify a known operational expectation.

7. Security Considerations

This document does not specify a protocol or implementation, only the requirements set. The rationale for individual requirements related to security includes discussion of the security considerations that motivate them.

8. IANA Considerations

This document neither creates nor updates any registries or codepoints, so there are no IANA Considerations.

9. Acknowledgements

Much work builds on lessons learned from the work performed by the IRTF DTN Research Group.

Work on this document at NASA's Glenn Research Center was funded by the NASA Glenn Research Center Innovation Funds.

10. References

10.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[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.ivancic-scf-problem-statement] Ivancic, W., Eddy, W., Iannicca, D., and J. Ishac, "Store, Carry and Forward Problem Statement", draft-ivancic-scf-problem-statement-00 (work in progress), July 2012.

[I-D.ivancic-scf-testing-requirements] Ivancic, W., Eddy, W., Iannicca, D., and J. Ishac, "Store, Carry and Forward Testing Requirements", draft-ivancic-scf-testing-requirements-00 (work in progress), July 2012.

[Multi-Terminal]
Ivancic, W., Paulsen, P., Stewart, D., Taylor, J., Lynch, S., Heberle, J., Northam, J., Jackson, C., and L. Wood, ""Large File Transfers from Space using Multiple Ground Terminals and Delay-Tolerant Networking"", IEEE Global Telecommunications Conference (GLOBECOM 2010), , December 2010.

[RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, April 2007.

[RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, November 2007.

Authors' Addresses

William Ivancic
NASA Glenn Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
United States

Phone: +1-216-433-3494
Email: william.d.ivancic@nasa.gov

Wesley M. Eddy
MTI Systems

Email: wes@mti-systems.com

Alan G. Hylton
NASA Glenn Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
United States

Phone: +1-216-433-6045
Email: alan.g.hylton@nasa.gov

Dennis C. Iannicca
NASA Glenn Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
United States

Phone: +1-216-433-6493
Email: dennis.c.iannicca@nasa.gov

Joseph A. Ishac
NASA Glenn Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
United States

Phone: +1-216-433-6587
Email: jishac@nasa.gov