Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 23, 2021

M. McBride
J. Guichard
Y. Qu
Futurewei
T. Hardjono
MIT
CJ. Bernardos
UC3M
February 19, 2021

Data Discovery Use Cases
draft-mcbride-data-discovery-use-cases-00

Abstract

There needs to be a solution for locating and capturing data in a standardized way. Data may be cached, copied and/or stored at multiple locations in the network on route to its final destination. With an increasingly high volume of devices connecting to the Internet, support for network caching and replication is critical for continuous data availability. There are data repositories throughout a modern network and there needs to be a standardized way to locating the repositories and discovering the desired data within.

There are several use cases which illustrate a need for a data discovery solution. An application might need to query the network to discover resources (program, service, resource) that can help the local application perform a particular task. Additionally, there could be volumes of data which needs to be searched and discovered in order to provide a result to be acted upon by the application. These are a couple of the use cases being addressed in this document.

Status of This Memo

Table of Contents

1. Introduction

An application might need to query the network to discover resources that can help the local application perform a particular task. There could be volumes of data which needs to be searched and discovered in order to provide a result to be acted upon.

Data discovery might involve an application requesting data. It might involve a device looking to store data or to request the processing from a data store and then gather the result. Or it could be execution of a set of instructions at an appropriate device in the network. Another possible area is service chaining where an

application needs to run its data through a firewall but the selected firewall must have a particular rule set applicable to this particular application. Perhaps the service function has to be located within a particular environment (security level). Or a particular device must be found that is capable of executing upon a set of instructions provided in the data packet. This document focuses on various data discovery use cases.

2. 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 RFC 2119 [RFC2119].

3. Terminology

o SFC: Service Function Chaining

o APN: Application-Aware Networking

o DLT: Distributed Ledger Technologies

4. Problem Statement

As discussed in [I-D.mcbride-data-discovery-problem-statement], there are many proprietary and standardized ways of discovering networking devices and hosts. There are many solutions for discovering data within a database. There are proprietary, non-standardized, ways of discovering the data that may be stored throughout an environment of networking devices. We can discover information about the devices but can't locate and capture stored data (resource, program, service, etc) in a standard way. With more networking devices storing collected data there needs to be a standard way of discovering the specific data needed amongst a potentially huge lake of databases.

This data discovery problem is particularly true for use cases where it will be important to have the capability to express a data request within the data packets and have the network route the traffic accordingly. This might be an application requesting data. It might be a device looking to store data or to request the processing, and result, from a data store. It could be execution of a set of instructions at an appropriate device in the network. An application may need to run its data through a firewall but the selected firewall must have a particular rule set applicable to this particular application. Perhaps a service function needs to be located within a particular environment (security level). Or a particular device must be found that is capable of executing upon a set of instructions

provided in the data packet. This document focuses on data discovery use cases.

4.1. Types of Data

Discoverable data can be a resource, program, service etc. And an infinite amount, and types, of data can be discoverable including statistics, measurements, temperature, location, metadata, health, transactions and so on.

Program: applets, graphics, games, spreadsheets, database systems, browsers, etc

Service: firewalls, load balancers, spam filters, header manipulators, etc

Resource: CPU, memory, etc

5. Use Cases

Here are some use cases to illustrate the need for data discovery:

5.1. Application-Aware Service Function Chaining

Application Aware Networking (APN), as described in [I-D.li-apn-problem-statement-usecases], allows applications to specify finer granularity requirements to the network operator by providing application knowledge to the network layer. This granularity includes the ability to convey the characteristics of an application's traffic flow and program the network infrastructure accordingly to provide service assurance.

An application might need to query the network to discover resources that can help the local application perform a particular task. Additionally, there could be volumes of data which needs to be searched and discovered in order to provide a result to be acted upon by the application.

End-to-end service delivery often needs to go through various service functions, including traditional network service functions such as firewalls, DPIs as well as new application-specific functions, both physical and virtual. APN provides assigning a given traffic flow to a specific service function chain (SFC) but also specifically allows the subsequent steering according to the application information carried in the APN packets.

When an application needs to run its data through a firewall, but the selected firewall must have a particular rule set applicable to this

particular application, then the application can leverage data discovery functionality. The service function may be required to be located within a particular environment such as a with a certain security level. Data discovery is needed to find that particular rule set (amongst the various firewalls) and then steer the packet accordingly. Or a particular device, along the SFC, may need to be found that is capable of executing upon a set of instructions provided in the data packet. The data capabilities of devices needs to be discoverable in order to steer the application packets towards them along a SFC.

5.2. Available CPU and Memory Resources

An application, or service, may need to discover the available server memory and compute resources from the network. A certain amount of CPU resources may be required to support a particular application workload. And the application may need to know the maximum CPU utilization threshold available on a compute device. Gathering info on available clock speeds and amount of cores can help determine how quickly servers load and interact with a set of applications. The network can provide the discoverability of the necessary data (cpu, memory) in order for applications to properly execute. A network planning app can also utilize this information to help predict future resource demands in order to meet applications performance requirements.

5.3. Data Dependency

There may be scenarios where it's critical to find X type of data that can help a local application, or service, successfully perform a particular task. Perhaps an industrial application needs real time measurement data, such as temperature, in order to execute a process. This required data may be cached, copied and/or stored at multiple locations in the network on route to its final destination. With an increasing percentage of devices connecting to the Internet being mobile, support for in-the-network caching and replication is critical for continuous data availability, not to mention efficient network and battery usage for endpoint devices. In order for some applications to properly execute, we need to find a way for the network to provide support for data discovery.

5.4. Distributed Ledgers

DLT Gateways, as discussed in [I-D.sardon-blockchain-gateways-usecases], will be given a permissioned view of assets/transactions, that they are requested to transfer, within their attached DLT domain. GW's may also need to discover assets/transactions, not explicitly provided, within the DLT

domain. It may become necessary for the GW (or other network element.. if permitted) to discover the data (asset, resource, service...) in order to transfer the required asset. Discovery of the data parts is also needed to validate the transfer after the asset movement. The ledger in the DLT will not hold all the relevant information pertaining to a previous asset transfer. So there needs to be ways to search/discover these. The data parts, to be discovered, include:

Relevant DLT transaction public-keys of the involved entities (i.e. public-keys (addresses) used on both DLTs.

Relevant entity public-keys and X.509 certs (Originator, owner of gateway G1, owner of gateway G2, Beneficiary). This is similar to the X.509 certs and cert-profiles used in the SWIFT banking network.

Relevant asset-related JSON documents (e.g. asset profiles).

5.5. Edge Computing

As described in [I-D.mcbride-edge-data-discovery-overview], the required data may be distributed across thousands of edge computing devices. Edge computing is motivated by the sheer volume of data that is being created by endpoint devices (sensors, cameras, lights, vehicles, drones, wearables, etc.) at the very network edge. In dense IoT deployments (e.g., many video cameras are streaming high definition video), where multiple data flows collect or converge at edge nodes, data is likely to need transformation (transcoded, subsampled, compressed, analyzed, annotated, combined, aggregated, etc.) to fit over the next hop link, or even to fit in memory or storage. This data, distributed across the edge, will need to be discovered in order to perform any number of functions such as an IoT application needing elevator vibration data in order to execute a process.

6. IANA Considerations

7. Security Considerations

8. Acknowledgements

9. Normative References

[I-D.li-apn-problem-statement-usecases]
Li, Z., Peng, S., Voyer, D., Xie, C., Liu, P., Qin, Z., Ebisawa, K., Previdi, S., and J. Guichard, "Problem Statement and Use Cases of Application-aware Networking (APN)", draft-li-apn-problem-statement-usecases-01 (work in progress), September 2020.

[I-D.mcbride-data-discovery-problem-statement]
McBride, M., Kutscher, D., Schooler, E., Bernardos, C., and D. Lopez, "Data Discovery Problem Statement", draft-mcbride-data-discovery-problem-statement-00 (work in progress), July 2020.

[I-D.mcbride-edge-data-discovery-overview]
McBride, M., Kutscher, D., Schooler, E., Bernardos, C., Lopez, D., and X. Foy, "Edge Data Discovery for COIN", draft-mcbride-edge-data-discovery-overview-05 (work in progress), November 2020.

[I-D.sardon-blockchain-gateways-usecases]
Sardon, A. and T. Hardjono, "Blockchain Gateways: Use-Cases", draft-sardon-blockchain-gateways-usecases-00 (work in progress), October 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

Authors' Addresses

Mike McBride
Futurewei

Email: michael.mcbride@futurewei.com

Jim Guichard
Futurewei

Email: james.n.guichard@futurewei.com

Yingzhen Qu
Futurewei

Email: yingzhen.qu@futurewei.com

Thomas Hardjono
MIT

Email: hardjono@mit.edu

Carlos J. Bernardos
Universidad Carlos III de Madrid
Av. Universidad, 30
Leganes, Madrid 28911
Spain

Phone: +34 91624 6236
Email: cjbc@it.uc3m.es
URI: http://www.it.uc3m.es/cjbc/