Industrial Internet of Things
Internet-Draft
Intended status: Informational
Expires: April 24, 2019
B. Liu
K. Katsalis
M. Zhang
Huawei Technologies
October 21, 2018

Service Function Chaining Applicability in Industrial Edge Computing
draft-liu-iiot-sfc-edge-computing-applicability-00

Abstract

Decoupling functions from the industrial hardware enables diverse, migratable, cross-industry replicable applications to be deployed with flexibility at the edge and on the cloud. Users should be free to adjust their business policies in industrial IoT and with low cost. Therefore efficient and dynamic orchestration of the applications is critical. This document describes several use cases that demonstrate the applicability of Service Function Chaining in industrial edge computing to organize the applications and provides extra requirements to support this applicability.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Cloudification has become a consensus trend in many domains due to the low cost, high scalability and reliability of the cloud. However, cloudification may not be easy or applicable for all aspects of industrial internet of things. In order to achieve control stability, an input must be given to the system with a bounded latency. For example, the control loop of a robotic arm can be 10ms, in which time the system must acquire the sensors' signals, compute the input and send it to the actuators. Deploying the controller remotely on the cloud is not practical because the round trip of signals is too time consuming, and packet loss will lead to instability. Besides, transmitting all the raw data to the cloud is not economical: VPNs or reserved bandwidth needs much more expenditure. In addition, industrial data is so sensitive that the owners of the data are not willing to expose it on the public Internet. Sending the raw data to the cloud presents such a risk.

The concept of edge computing, i.e. providing networking, compute and storage capabilities close to the data source, is promising to deal with the aforementioned requirements. Time sensitive applications

are deployed at the edge to achieve fast response. The raw data is processed, filtered, or compressed, hence the size could be reduced and the privacy data stays under control of users. A more detailed introduction to edge computing can be found in [I-D.zhang-iiot-edge-computing-gap-analysis] and [I-D.geng-iiot-edge-computing-problem-statement].

Tomorrow will be the era of edge cloud orchestration, where the edge computing and cloud computing work together to meet the various requirements of users. Diverse applications will be deployed at the edge and on the cloud. How to deploy them correctly to realize the industry users' policies, and how to manage the applications efficiently when the policy changes, are currently open questions. Since the edge is the ingress of data, when the data from different sensors arrive at the edge computing device, the set of applications that the data will go through should be indicated according to the preset policies. After the processing by an application, the output data must be forwarded to the correct next hop. Except the applications which have to be deployed at the edge or the cloud due to response time and processing resource requirements, multiple copies of applications can be deployed at different locations, which permit the offload of the tasks to other copies of the application when one copy is busy or over utilized.

Service Function Chaining could be a suitable way to organize the edge and cloud applications. The architecture in [RFC7665] realizes the decoupling of the service plane and forwarding plane, making it easier to add or delete one application or adjust the order of their invocation. In the data plane, the NSH header helps enhance the logical connection between the applications. The classifier in SFC, which decides the path to forward traffic through, matches the requirement of task indication. For the same SFC, different paths can be deployed as backups to each other. When one path is disrupted or fully loaded, the work can be offloaded to another path yet have the same set of functions applied to it.

This document describes the idea of using SFC in industrial edge computing to organize the applications according to user defined policies. Use cases are given as examples to help explain the idea.

2. 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].

ECN

Edge Computing Node: An abstract appellation of devices with edge computing capabilities. In industrial domain, an edge computing device can be a logic controller, a gateway, or a local server cluster, etc. An edge computing node MAY act as the physical carrier of classifier, SFs and/or SFFs.

Service Function Chain
A service function chain defines an ordered set of abstract service functions and ordering constraints that must be applied to packets and/or frames and/or flows selected as a result of classification.

Service Function
An SF in SFC can be mapped to an application in edge computing.

3. Industrial Edge Computing Overview

Industrial edge computing can be deployed in a hierarchical way. For example, in manufacturing industry, the scope of group can refer to a pipeline, and the scope of campus can refer to a factory. The data comes from the end devices, such as sensors, actuators, equipment, assets, etc. The end devices can be edge computing capable or incapable. A group of end devices (e.g. geographical neighbors) are connected to an edge gateway. The edge gateway offers connectivity to the wide area network and may offer connectivity among the end devices. Normally, the resources on an edge gateway are constrained, hence the gateway can only handle relatively simple, deterministic or dedicated tasks:

- Local data processing: aggregation, filtering, translation, consolidation and analytics.

- Local control/reasoning logic: automation control, decision-making, fault detection, and so on. The parameters/models are optimized/trained on the cloud, then assigned to the edge.

- Device management: edge gateway acts as the local assets manager and enables remote assets management via the wide area network.

At the campus level, an edge cloud server with more resources may be deployed. Since more resources are provided, relatively complex tasks can be handled, such as:

- Operation management: translating upper layer abstract commands into operational commands of end devices, updating parameters, organizing new work flows, etc.

- o Data desensitization: users are not willing to expose their sensitive data to the public Internet, thus before uploading the sensitive data must be fuzzified.

- o Data logging: the edge cloud server may have larger storage, hence it is preferred to perform the data logging at the server, especially when the data is large or needs to be stored for long time.

- o Task offloading: when the edge gateway is over loaded, some tasks originally being conducted at the gateways can be transferred to the edge cloud if possible.

The campus network is connected to the cloud via an overlay private network over the public network. The cloud can be private/public which implements the company's or third-party regulatory authority's applications. The applications deployed on the cloud are usually computationally intensive and require mass storage. These applications involve big data analysis, MES, ERP, CRM, etc.

It should be clarified that the aforementioned applications and the hierarchy to deploy them are just examples. The actual deployment depends on the use cases and requirements.

```
                        +------------------+
                        |                  |
                        |       Cloud      |
                        |                  |
                        +---------+--------+
                                  |
                                  |                   ..................
                              _,..+.._                .Campus 2        .
                           ,-'        `-.             .+--------------+.
                          /              `.           .|              |.
                         |     Network    |+-----------+  Edge Cloud  |.
                         `.              /            .|              |.
                           `.__     __,-'             .+--------------+.
                               `''+'                  ..................
   ...............................|........................
   .Campus 1                      |                       .
   .                      +-------+------+                .
   .                      |              |                .
   .                      |  Edge Cloud  |                .
   .                      |              |                .
   .                      +-------+------+                .
   .                              |                       .
   .               +--------------+---------------+       .
   .               |                              |       .
   .*--------------|--------------*       *-------|------*.
   .|       +------+-----+        |       |+------+-----+|.
   .|       |Edge Gateway|        |       ||Edge Gateway||.
   .|       +------+-----+        |       |+------+-----+|.
   .|              |              |       |       |      |.
   .|      +-------+-------+      |       |       |      |.
   .|      |               |      |       |       |      |.
   .|+-----+----+    +-----+----+ |       | +-----+----+ |.
   .||End Device|    |End Device| |       | |End Device| |.
   .|+----------+    +----------+ |       | +----------+ |.
   .|          Group 1            |       |    Group 2   |.
   .*-----------------------------*       *--------------*.
   ........................................................
```

Figure 1: Hierarchical Deployment of Edge and Cloud

4. Function Deployment Constraints

4.1. Node Capability Constraints

The diversity of SFs results in different requirements to capabilities of nodes carrying them. For example, AI training applications may need powerful CPUs and large storage to handle the samples. Therefore, it is not appropriate to deploy such a SF on

gateways or end devices with contrained resources. Besides, some ECNs may have expertise for certain SFs, therefore it will be more efficient to deploy such SFs on these ECNs. For instance, it is better to let a ECN equipped with a GPU to conduct image processing function.

4.2. Performance Constraints

The users may have performance requirements on the SFPs or a certain SF, e.g., the end-to-end delay of the SFP, the response time of SFs, the network bandwidth that the SFs demand, etc. A SF SHOULD be deployed close to the data source if it requires a short response time, since the round-trip to the cloud takes a long time. Data compression/aggregation SHOULD be performed to avoid sending large amounts of data to the cloud, if the users are willing to save their expenditure in network rental.

4.3. Privacy Constraints

Privacy must be considered for industrial data. Industrial professionals are not willing to expose their sensitive data to the public network/cloud. Thus the data must be processed in the portion of the network that the industry can control, e.g. the gateway, the local server. In this case, the related functions MUST be deployed at the edge instead of the cloud.

5. SFC for Edge Computing use case

In order to have an intuitive view for how to implement SFC in edge computing, we use connected elevator as an example. An edge gateway is deployed for each elevator or a group of elevators to process the data from the sensors or cameras. Compared to the raw data, the volume of the data to be uploaded to the cloud is greatly reduced. Besides, the edge gateway can react in a short timeframe when dealing with emergency situations due to the avoidance of a round-trip to the cloud. An edge cloud server at the campus level may also be deployed to perform elevator management, execute commands from the cloud or undertake the tasks offloaded from the edge gateways. In the cloud data center, more complex applications are installed, such as predictive maintenance, machine learning, digital twins, etc. Figure 2 shows the described architecture. All the applications at the edge and on the cloud can be organized in the form of an SFC. Since then, we use the term "SF" to represent the "applications".

```
                           +-------+            +--------+          ---
                           |Sensors|            | Camera |           ^
                           +---+---+            +----+---+           |
                               |                     |               |
                               +----------+----------+               |
                     *- ------------------|------*                   |
                     |  +---+       +-----+----+ |                   |
                     |  |SF1|+--+   |Classifier| |                   |
                     |  +---+   |   +-----+----+ |    *-----------*  |
                     |          |         |      |    |           |  |
                     | +----+   |       +-+-+    |    |   +---+   |  |
     Edge Gateway    | |SF2a|+--+-------+SFF+------------+|SF4|   |  |
                     | +----+   |       +-+-+    |    |   +---+   |  |
                     |          |         |      |    *-----------*
                     | +----+   |         |      |   Video Processor E
                     | |SF3a|+--+         |      |                   d
                     | +----+             |      |                   g
                     *- ------------------|------*                   e
                                          +-------+                  |
                        *-----------------|----*  |                  |
                        |   +----+        |    |  |                  |
                        |   |SF2b|+--+    |    |  |                  |
                        |   +----+   |  +-+-+  |  |                  |
    Edge Cloud          |            +--+SFF|  |  |                  |
   (Campus Server)      |   +----+   |  +-+-+  |  |                  |
                        |   |SF3b|+--+    |    |  |                  |
                        |   +----+        |    |  |                  |
                        *-----------------|----*  |                  v
                                          |       |                 ---
                                          +-------+                  ^
                   *----------------------|----------*               |
                   | +----+               |          |               |
                   | |SF3c|+-------+      |          |
                   | +----+        |      |          |               C
                   | +---+         |    +-+-+        |               l
                   | |SF5|+--------+---+|SFF|        |               o
     Data Center   | +---+         |    +---+        |               u
                   | +---+         |                 |               d
                   | |SF6|+--------+                 |
                   | +---+         |                 |               |
                   | +---+         |                 |               |
                   | |SF7|+--------+                 |               v
                   | +---+                           |              ---
                   *---------------------------------*
```

Figure 2: An example for using SFC in connected elevator

Service Functions	Explaination	Deployment Constraints
SF1	Fault Determination	Edge Gateway
SF2	Data Aggregation	Edge Gateway/Edge Cloud
SF3	Data Logging	Edge Gateway/Edge Cloud/Cloud
SF4	Video Processing	Video Processor
SF5	Predictive Maintenance	Cloud
SF6	AI trainning	Cloud
SF7	Alarm	Cloud

Table 1: The SFs in connected elevator

Path IDs	Paths
SFP1	SF1-->SF2a-->SF3a-->SF5
SFP2	SF1-->SF2b-->SF3b-->SF5
SFP3	SF3-->SF6
SFP4	SF1-->SF7

Table 2: The SFPs in connected elevator

Table 1 shows a list of SFs and their deployment constraints. SF1 must be deployed at the edge gateway, i.e. close to the data source, because the elevator must react in a short timeframe when a fault is detected. The SF2 data aggregation should be deployed at the edge (edge gateway (SF2a) or edge cloud (SF2b)), if the user is willing to save network bandwidth. The aggregated data can be stored at the edge as a distributed database and can be pulled by the cloud when needed, or directly on the cloud as mass storage is provided there. The SF4 video processing should be handled a the dedicated processor to achieve maximum efficiency. The SFs requiring strong computing abilities such as predictive maintenance (SF5) and AI training (SF6) should be deployed on the cloud. Alarms should be triggered on the cloud when faults are detected at the edge, so that the maintenance staff can be informed.

5.1. Building paths from chains

In the example of connected elevator, there are three SFCs. How to instantiate the SFC to actual SFPs depends on the deployment constraints of SFs and the requirements of users. According to the

constraints listed in Table 1, there are 5 possible paths for the chain SF1-->SF2-->SF3-->SF5. The users can decide to use some or all of these paths. Some paths can be prioritized over the others depending on user defined objective functions, such as:

o Maximize the use of the edge: decentralization, deploy the SFs at the edge as many as possible, make full use of the computing power at the edge. This objective function may be preferred by users pursuing timely response and data privacy.

o Maximize the use of the cloud: centralization, deploy the SFs on the cloud as many as possible, so that the edge focuses only on the necessary SFs like timely response tasks. This objective function may be preferred by users that don't have powerful edge computing devices.

o Minimize the traffic between the edge and the cloud: deploy the SFs and SFFs which have relatively large communication traffic at the same place.

In actual deployment, first the users must identify the constraints on which they have concerns. Then the users must find out the paths which meet these constraints, after that order all the possible paths according to the objective function. The users may choose one primary path (e.g. SF1-->SF2a-->SF3a-->SF5), and several paths as backups, e.g. SF1-->SF2b-->SF3b-->SF5 and SF1-->SF2b-->SF3c-->SF5.

5.2. Selecting a path

The Classifier is in charge of filtering the flows which should enter the SFC and deciding which path to follow. The classification is conducted by user-defined policies, such as source/destination port, IP addresses. The initial classification happens at the ingress of the SFC domain. In the case of edge computing, it can be the edge gateway. Related information can be found in the section 4.7 of [RFC7665].

Besides, attaching the traffic to a specific SFP can also depends on the status of the paths. For example, if the primary path is fully loaded, the classifier should direct the subsequent traffic to one of the backup paths. The status of a path can be acquired by iOAM [I-D.brockners-sfc-ioam-nsh] using the trace options. Then the egress of the SFC domain will upload the status to the controller. And the controller will affect the initial classification accordingly.

5.3. Path redirection

As introduced in [RFC7665], a SFP can be redirected to another SFP. For example, in Figure 2, a flow is originally directed to SFP1, however, a fault is detected thus it must be redirected to SFP4 to trigger the alarm (SF7). The redirection can be done by assigning another path ID in the NSH header.

6. Applicability Requirements

The following requirements should be considered when using SFC to organize the applications across the edge and the cloud in industrial IoT:

- The SFs MUST be deployed at qualified places regarding to the deployment constraints.

- Objective functions SHOULD be defined to sort all the possible SFPs, so that the users can find out the optimal path.

- It is RECOMMENDED to build backup paths. When demanded performance is not achieved, the primary path SHOULD be switched to a backup path.

- An orchestrator or controller MAY be required to build the path and detect the status of the path, the SFs and SFFs. Configuration, management interface.

- A control plane is needed to update the forwarding tables accordingly in the network when a SFP is changed.

- Coordination between controllers

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

TBD

9. References

9.1. Normative References

[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>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

9.2. Informative References

[I-D.brockners-sfc-ioam-nsh]
Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "NSH Encapsulation for In-situ OAM Data", draft-brockners-sfc-ioam-nsh-01 (work in progress), March 2018.

[I-D.geng-iiot-edge-computing-problem-statement]
Geng, L., Zhang, M., McBride, M., and B. Liu, "Problem Statement of Edge Computing on Premises for Industrial IoT", draft-geng-iiot-edge-computing-problem-statement-01 (work in progress), March 2018.

[I-D.zhang-iiot-edge-computing-gap-analysis]
Zhang, M., Liu, B., McBride, M., Hu, C., and L. Geng, "Gap Analysis of Edge Computing for Industrial IoT", draft-zhang-iiot-edge-computing-gap-analysis-00 (work in progress), March 2018.

Authors' Addresses

Bing Liu
Huawei Technologies
No. 156 Beiqing Rd. Haidian District
Beijing 100095
China

Email: remy.liubing@huawei.com

Konstantinos Katsalis
Huawei Technologies
No. 12 Riesstrasse
Munich 80992
Germany

Email: kostas.katsalis@huawei.com

Mingui Zhang
Huawei Technologies
No. 156 Beiqing Rd. Haidian District
Beijing 100095
China

Email: zhangmingui@huawei.com