Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

E. Nordmark
Zededa

October 22, 2018

Computing at the edge
draft-nordmark-t2trg-computing-edge-00

Abstract

There has been some discussion about edge computing in the T2TRG. This note explores the edge from a computing perspective, and from that suggests aspects of networking that relate to edge computing. It includes some potential research problems for networking edge computing.

Status of This Memo

This Internet-Draft is submitted 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 April 25, 2019.

Copyright Notice

Table of Contents

1. Introduction 2
2. What is computing? 2
3. What is edge? 3
4. Why edge computing? 4
5. Networking for Edge Computing 5
6. Application connectivity 5
7. Potential research topics 6
8. Security Considerations 6
9. IANA Considerations 6
10. Informative References 7
Author's Address 7

1. Introduction

Edge computing has gained increased interest industry over the last few years and there has already been some discussion in the IETF and IRTF as exemplified by [I-D.zhang-iiot-edge-computing-gap-analysis], [I-D.hong-iot-edge-computing], and [I-D.geng-iiot-edge-computing-problem-statement]. This note builds on that work while taking a step back from the networking aspects to look at how computing would happen at the edge and what is needed to satisfy that computing. The note also tries to separate out the motivations for computing edge from the attributes of the edge.

2. What is computing?

The general notion of computing is likely to be clear; some programmable ability in the form of CPU/GPU plus some memory, with some ability to interact with the external word (I/O, networking), with optional ability to store data locally with persistence.

However, it might be useful to try to separate the flexibility of edge computing from fixed function devices which do not have the capacity or flexibility to perform other functions than envisioned prior to their deployment. Such fixed function devices still require a software/firmware update capability as discussed in [RFC8240], but they do not require handling new application deployment and associated new communication patterns.

These more flexible devices are likely to be larger than the class 2 devices defined in [RFC7228], however if applications are sufficiently small, constrained devices might very well be edge computing devices. But in general it makes sense to think about devices of the Raspberry Pi class and larger.

If the set of applications which might be deployed on a device isn't known prior to deployment it might be difficult to determine a utility cycle and resulting upper bound on power consumption. As such it is hard to envision this flexibility for devices which need to run for years on a battery or using energy harvesting.

One way to envision edge computing is to think of a developer or a devops person wanting to run computation closer to the sensors and actuators, but with the same ease as running computation in the cloud (e.g, docker, kubernetes). In that vein one can think of existing applications with a new deployment target; deploy to all the light-poles in Palo Alto as opposed to a cloud availability zone. Of course the industry will also develop new applications and new applications for the edge, but some applications are likely to migrate from the cloud or be existing standalone applications.

In some cases it is useful to make a distinction between a device and an (IoT) gateway in this context. However, the term gateway might mean very different things. In some cases it is a product category, referring to compact and passively cooled PCs with rich physical connectivity e.g., RS485 ports and multiple Ethernet ports, etc. That generalizes to an architectural node which has a router and protocol translators e.g., from Modbus to MQTT. In other cases it refers to nodes which run software to translate one data model to another one as in [I-D.iab-iotsi-workshop].

We might see architectural patterns in the future which separate fixed function devices (in particular those with hardware crypto implementations) with long deployment lifetimes from the larger Internet and its threats and need for crypto agility by interposing a "gateway" device which limits the security exposure for those fixed function devices. However, we have yet to see that architectural pattern develop.

3. What is edge?

As edge computing is gaining popularity the term seems to be applied to refer to a large range of things:

- In the context of Industry 4.0 the edge is where the actions are to be taken based on the analyzed data, thus in or near the machines in the factory.
- The Industrial Internet Consortium (IIC) refers to the edge as customer premise equipment i.e., equipment deployed in the enterprise.
- ETSI MEC refers to the edge of provider network, i.e., the provider edge.

- Some datacenter operators with many smaller datacenters offer edge computing solutions since they claim being closer to the users than the large cloud operators.
- There is also the Internet Edge (where the large Content Delivery Networks tend to be deployed) and the datacenter edge.

From an architectural perspective what matters is not the term but what the characteristics are. As you move further and further out towards the premises and enterprises some new characteristics appear compared to running inside a datacenter, large or small. These apply to varying degrees whether that edge device is in a light pole in a smart city, on a factory floor, on an oil rig, or on a truck.

- Less physical security - not the same physical access control.
- Different network security - there might not be a firewall between it and the Internet
- (Physical) location is key in many cases; need to be in BTLE range or have the camera pointing at the right road intersection.
- Scale is likely to be much larger than the cloud datacenters because devices need to exist in all the locations which matter.
- Less network transparency; NATs and/or firewalls could exist between the elements of the computation at the edge.
- Richer uplink networking (such as Ethernet, LTE, WiFi, etc). Might have redundant uplinks using different technologies.
- Might require specific downlink connectivity (6lo, BTLE, RS485, etc)
- Intermittent connectivity more likely than inside datacenter.
- Normally no fallback network such as serial console, management network, or remote power control to handle software updates gone wrong.

4. Why edge computing?

The benefits of moving computing closer the sensors and actuators seems to fall in three categories:

- Lower latency, for instance to handle process control where decisions need to be made locally.
- Cost of bandwidth. In many cases sending all the data to the cloud to perform data aggregation and run analytics can get quite expensive compared to local aggregation and analytics.
- Autonomy and failure resiliency. A machine on the factory floor needs to continue to work even if the Internet/cloud connectivity is temporarily out.

Some documents also mention data jurisdiction as a key benefit. That seems to be more an issue with keeping the data in the same

enterprise and same country and the data origin, and not about keeping it as close as possible to the sensors and actuators.

5. Networking for Edge Computing

From the above list several of the different attributes between the datacenter and the edge are about connectivity. In general the connectivity is a lot more diverse at the edge than in the datacenter.

Solutions need to handle at least Ethernet, WiFi, LTE, IPv4/IPv6, redundant/multihomed connectivity, mobility, and NATs. Ideally the applications which are deployed at the edge should not be required to handle this diversity but instead operate the same as in the cloud where the applications see DNS and IP connectivity.

In addition, if the applications are structured as communicating (micro) services when deployed in the cloud, they are likely to assume some level of network isolation (security groups etc) from the infrastructure. In order to be able to deploy such applications at the edge the infrastructure needs to provide at least the same level of isolation, and due to the more challenging security environment at the edge, it probably needs to be stronger.

6. Application connectivity

Earlier work [RFC7452] outlines three different communication patters:

- Device-to-Device Communication Pattern
- Device-to-Cloud Communication Pattern
- Device-to-Gateway Communication Pattern

For the purposes of this note we separate the Device-to-Device pattern into a topology-specific pattern and a topology independent.

The topology-specific D2D pattern is where a set of devices are e.g., on the same link hence can make assumptions about discovery and security that are related to the link's properties. Such deployments are common today for certain applications.

The topology-independent D2D pattern is examplified by an application deployment pattern where one microservice needs e.g., a GPU for running a model and the GPU might exist in the same building or site but on a different network perhaps separated by NATs. To enable the promise of flexibility for edge computing the infrastructure needs to be able to support such a communication pattern, which places new

requirements on discovery and security even when all of the edge infrastructure is under common administrative control.

The Device-to-Cloud Communication Pattern [RFC7452] is commonly being deployed today, but does not necessarily handle the applications with real-time considerations.

The Device-to-Gateway Communication Pattern [RFC7452] can mean different things depending on what type of gateway is at play. In current deployment it seems to imply that the application topology is the same as the network topology. For instance, the devices connect over one protocol to a gateway, and that physical gateway is the only one which can run the applications (be they simple protocol translators or analytics/AI) which serve those devices. Over time one would expect to see the application/micro-service topology to be unrelated to the network topology; that is how the datacenter and cloud has evolved.

7. Potential research topics

As discussed in the previous section there are likely to be additional needs to enable micro-services at the edge which has the different attributes we have identified. However, most of that might be an engineering exercise.

That assumes a single asset owner controlling some set of devices, gateways, and compute elements. In the case of that asset owner leasing space for VMs or containers those technologies as used in the cloud can be reused for multi-tenancy. However, orchestration might need to be different due to the importance of location for edge computing.

If we envision a future where we want to enable more flexible resource sharing, e.g., shop owners on a street in Sao Paulo be able to offer their spare CPU and GPU capacity to their neighbors with some compensation/tokens, there will be additional issues around trust, compensation, discovery, etc.

8. Security Considerations

This note touches on both system security and communication security.

9. IANA Considerations

There are no IANA actions needed for this document.

10. Informative References

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-editor.org/info/rfc7228>.

[RFC7452] Tschofenig, H., Arkko, J., Thaler, D., and D. McPherson, "Architectural Considerations in Smart Object Networking", RFC 7452, DOI 10.17487/RFC7452, March 2015, <https://www.rfc-editor.org/info/rfc7452>.

[RFC8240] Tschofenig, H. and S. Farrell, "Report from the Internet of Things Software Update (IoTSU) Workshop 2016", RFC 8240, DOI 10.17487/RFC8240, September 2017, <https://www.rfc-editor.org/info/rfc8240>.

[I-D.iab-iotsi-workshop] Jimenez, J., Tschofenig, H., and D. Thaler, "Report from the Internet of Things (IoT) Semantic Interoperability (IOTSI) Workshop 2016", draft-iab-iotsi-workshop-02 (work in progress), July 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.

[I-D.hong-iot-edge-computing] Hong, J., Hong, Y., and J. Youn, "Problem Statement of IoT integrated with Edge Computing", draft-hong-iot-edge-computing-01 (work in progress), October 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.

Author's Address

Erik Nordmark
Zededa
Santa Clara, CA
USA

Email: nordmark@sonic.net