T2TRG Edge Computing Break Out Discussion at IETF-98

Dirk Kutscher and Eve Schooler (on behalf of breakout session team at T2TRG meeting)

We discussed three main topics: 1) Motivation for Edge Computing, 2) Terminology, and 3) Research Questions.

The Motivation for Edge Computing

IoT deployments generate an increasing amount of data at the network edge, which reverses today's dominant data flow direction compared to how the network was designed and provisioned. Data is no longer primarily sent downstream (i.e., from the cloud to the edge), but increasingly is generated at the network edge, then processed and/or consumed locally, before possibly being transmitted upstream in the direction toward the cloud, to other Internet hosts, including cloud-based consumers.

Much of the data is generated by applications with requirements unmet by the current cloud infrastructure: support for high data volume, time sensitivity, trust sensitivity, intermittent (dis)connectivity, and the need for energy efficiency and reduced costs. As a result, cloud functionality (compute, storage, networking, control, etc.) is migrating to be more proximate to the data, leading to what is referred to as "edge computing" or "edge clouds".

With the expected **high data volume** and limited uplink bandwidth to the Internet, it may not be possible to transmit data in its original format, particularly streaming data, which is generated continuously. In these scenarios, local processing (e.g., filtering, compression, transcoding, aggregation, analytics, etc.) may be required to enable meaningful data and connectivity management. Such filtering and aggregation could be seen as an application-independent edge computing capability in a data-oriented IoT approach.

Moreover, certain IoT deployments, for example factory automation, may exhibit strong **time sensitivity and/or trust sensitivity** with respect to data communication and processing. For example, the back-end cloud might simply be" too far away" (i.e., the roundtrip time is too long) for the timing constraints of the control loop for certain industrial IoT applications. Furthermore, it may not be desirable nor legal to transmit and process data remotely because of data privacy policies. For instance, in the healthcare industry and in manufacturing scenarios that are sensitive to the leakage of intellectual property, the data may be too sensitive for it to leave the local premises. In such cases, local data processing and analytics at the edge are required.

Depending on the nature of the interaction between the IoT edge cloud and the properties of access networks, keeping data and processing local also helps with **energy efficiency and cost reduction**.

Finally, there are usages where the connectivity to the cloud is prohibitively expensive (e.g., satellite communications for an oil rig or an airplane), rendering the network effectively disconnected. Still other use cases, where device mobility, radio interference, or energy conservation render the network only **intermittently connected**. Additionally there are IoT deployments that require autonomy and as such cannot rely on Internet connectivity at all. Thus Edge computing would be necessary when the cloud is inaccessible.

Terminology

We raised many questions, discussed relevant terms, and posed ideas.

- **What is the edge**?
    - What is the edge a boundary between? For example, the edge might differ in definition for the Telcos versus other service providers
    - Moving from edge computing to fog computing (a multi-tiered cloud of clouds) creates additional edges
- **Edge computing is a first step toward re-imagining the data center**
    - Edge computing moves cloud computing functionality closer to the edges of the network and closer to the Things that comprise that edge
    - Compute, storage, networking, control, actuation - all will be distributed beyond the confines of a back-end cloud
    - What will the data center look like as it moves out of the back-end cloud and closer to the edge?
- **Edge dynamics supports (mobile) edge computing**
    - How dynamically can edges be created?
    - How dynamically would we like or need to distribute computation, storage, etc.?
- **Edge computing is more than computation on a gateway**
    - Edge computing is often equated with the first-hop gateway in the direction from Things to the cloud
    - Edge computing however also could be viewed as the ensemble of resources that are willing to logically defined as an “edge cloud”
    - Not limited to specific platforms and execution environments

Research Questions

We discussed the following research questions:

- **Programming models**
    - How would people develop applications that can leverage edge computing?

    - What distributed constructs require support?
    - How to steward, curate, route, cache, process, migrate, archive the edge device data?
- **Networking and operations**
    - Compute function description
    - Compute function discovery
    - Assembly of individual functions into larger blocks, applications & services
    - Orchestration of edge computing systems
    - Managed vs. unmanaged edge computing
- **Isolation**
    - How would individual tenants and compute functions be isolated in a decentralized cloud environment?
- **What would be granularity levels for edge compute functions?**
    - Containers
    - Step functions
    - Stateless functions
    - Named Function Networking as in ICN
- **Multi-X**
    - Multi-application, multi-user, multi-tenancy
    - Edge Computing in multi domain networks

Discussion

Finally, we discussed the following additional topics and ideas:

- **Difference between Edge Computing and Data Center Computing**
    - Would edge computing rely on completely new abstractions and mechanisms or would it be compatible with existing data center distributed computing platforms?
    - As in, would edge computing re-use existing cloud service provider APIs?
- **Usability of Edge Computing**
    - With increased levels of dynamics, scalability, and group data sharing, how to extend existing eco-system components (e.g., data/meta-data registries) to support?
    - How to make distributed system interfaces intuitive and consistent?
- **From “Pet” to “Cattle model”**
    - In the presence of ubiquitous, cheap IoT deployments, how carefully should/can Edge Computing deployments be crafted?
    - What would be security and availability implications?
- **“Rackscale for Edge Computing”**
    - Will there be established models for disaggregating network, storage, compute for the edge?
    - Will Edge computing rely on similar automation and operations support functions (infrastructure management, telemetry)?

    - Will Edge computing rely on standards for software-defined networking to dynamically configure and reconfigure resource pools?
- **Networking Edge Computing**
    - What would be appropriate communication models to support Edge Computing?
    - How will Edge Computing affect existing protocols?
    - If edge computing and cloud computing represent two ends of an evolving cloud computing spectrum, how to seamlessly evolve edge computing to fog computing?
    - Will there be a difference between intra-cloud and inter-cloud communication in Edge/Fog computing?
    - Will different technologies be needed to support upstream vs downstream data flows?
    - Might Information-Centric concepts be helpful (cf. Named Function Networking)? Since ICNs already combine routing with native caching in the network, could they be extended to support processing for data in-flight as well (e.g., at the aggregation points in the reverse data flow paths)?