Fog and IoT: An Overview of Research Opportunities

Mung Chiang, *Fellow, IEEE*, and Tao Zhang, *Fellow, IEEE*

Abstract— **Fog is an emergent architecture for computing, storage, control and networking that distributes these services closer to end users along the Cloud-to-Things continuum. It covers both mobile and wireline scenarios, traverses across hardware and software, resides on network edge but also through access networks and among end users, includes both data plane special cases like cloudlets and control plane special cases such as crowd-sensing. As an architecture, it supports a growing variety of applications, including those in the Internet of Things (IoT), Fifth-Generation (5G) wireless systems, and embedded artificial intelligence (AI). This survey article summarizes the opportunities and challenges of Fog, focusing primarily on the networking context of IoT.**

Index Terms—**fog, fog computing, fog networking, fog storage, fog control, edge computing, edge storage, edge networking, IoT, Internet of Things.**

I. INTRODUCTION

Over the past decade, moving computing, control, and data storage into the Cloud has been the trend. In particular, computing, storage, and network management functions are shifted to centralized data centers, backbone IP networks, and cellular core networks. Today, however, Cloud computing is encountering growing challenges in meeting many new requirements in the emerging Internet of Things (IoT).

At the same time, there has been a surging number and variety of powerful end-user, network edge, and access devices: smartphones, tablets, smart home appliances, small cellular base stations, edge routers, traffic control cabinets along the roadside, connected vehicles, smart meters and energy controllers in a smart power grid, smart building controllers, manufacturing control systems, just to name a few. Many more smart clients and edge devices, such as information-transmitting light-bulbs, computers on a stick, and button-sized Radio Frequency tuners, are following right behind.

It has therefore become feasible and interesting to ask: "What can be done close to the end users?" Can your car become your primary data store? Can a single appliance in your house integrate the different services and applications that have been provided by separate systems such as TV set-boxes, home media centers, Internet access routers, and smart energy control boxes? What if smartphones themselves can collectively perform radio network control functions that are performed by gateways in the LTE core networks today? What can a crowd of nearby smart endpoints and network edge devices collectively accomplish through a distributed and self-organized network on the edge? Can smart edge devices collectively enable ultra-low or even deterministic latency to support delay-sensitive applications such as real-time data analytics on the edge, mining of streaming data, and industrial control functions?

What these questions point to is a pendulum swinging now back from "click" toward "brick," from "more centralization" to "more immersive distribution," from clouds "bigger and farther away" to not just smaller clouds but computation and control closer to sensors, actuators and users. The pendulum between centralization and distribution is decades-old, with two distinct flavors of "distribution": first is the end-to-end principle as exemplified by TCP congestion control and perhaps Peer-to-Peer (P2P) multicast overlay, and second is leveraging local proximity as in Ethernet and sensor networks. Fog embodies and further accelerates this click-to-brick swing-back from the second angle.

This paper starts with the range of new challenges in the emerging IoT and the difficulty to address these challenges with today's computing and networking models. The paper then discusses why we will need a new architecture – Fog computing, Fog networking, Fog storage, Fog control, or collectively Fog, for simplicity – and how it can fill the technology gaps and create new business opportunities.

Architecture is about functionality allocation: deciding who does what and how to "glue" them back together. Unlike the more mature technology fields such as serial computation, digital communication, and the Internet, where strong and solid architectural foundation has been laid, we are still searching for architectural principles for many emerging systems and applications such as IoT, cyber-physical systems, and embedded AI. We need to make fundamental decisions ranging from where to compute and where to store data along the Cloud-to-Things continuum to how to ma computation tasks into a substrate of heterogeneously capable and variably available nodes. Fog provides a direction for us to explore such an architecture; and this paper pays particular attention to IoT as a large application domain over the Fog architectural foundation.

This article was first submitted on April 28, 2016.

Mung Chiang is with the Department of Electrical Engineering, Princeton University (chiangm@princeton.edu).

Tao Zhang is with Cisco Systems (tazhang2@cisco.com).

II. NEW CHALLENGES IN IOT REQUIRES NEW ARCHITECTURE

The emerging IoT introduces many new challenges that cannot be adequately addressed by today's Cloud and host computing models. Here, we discuss several such fundamental challenges.

Stringent latency requirements. Many industrial control systems, such as manufacturing systems, smart grids, oil and gas systems, and goods packaging systems, often demand that end-to-end latencies between the sensor and the control node stay within a few milliseconds [10]. Many other IoT applications, such as vehicle-to-vehicle communications, vehicle-to-roadside communications, drone flight control applications, virtual reality applications, gaming applications, and real-time financial trading applications, may require latencies below a few tens of milliseconds. These requirements fall far outside what mainstream Cloud services can achieve.

Network bandwidth constraints. The vast and rapidly growing number of connected things is creating data at an exponential rate [11]. A connected car, for example, can create tens of megabytes of data per second. This will include data about 1) the car's mobility such as its routes and speeds, 2) the car's operating conditions such as the wear and tear on its components, 3) the car's surrounding environment such as road and weather conditions, and 4) videos recorded by the car's safety cameras. An autonomous vehicle will generate even more data, which was estimated to be about one gigabyte per second [12]. The US smart grid is expected to generate 1000 petabytes of data each year. By comparison, the US Library of Congress generated about 2.4 petabytes of data a month, Google trafficked about one petabyte a month, and AT&T's network consumed 200 petabytes a year in 2010 [13].

Sending all the data to the Cloud will require prohibitively high network bandwidth. It is often unnecessary or sometimes prohibited due to regulations and data privacy concerns. ABI Research estimates that 90% of the data generated by the endpoints will be stored and processed locally rather than in the Cloud [11].

Resource-constrained devices. Many IoT devices will have severely limited resources. Examples include sensors, data collectors, actuators, controllers, surveillance cameras, cars, trains, drones, and medical devices embedded in patients.

Many resource-constrained devices will not be able to rely solely on their own limited resources to fulfill all their computing needs. Requiring all of them to interact directly with the Cloud will be unrealistic and cost prohibitive as well, because such interactions often require resource-intensive processing and complex protocols. For example, the multitude of microcomputers on a modern vehicle need firmware updates, but requiring each of these resource-constrained devices to perform the heavy cryptographic operations and sophisticated procedures required to obtain firmware updates from Cloud services will be impractical.

Cyber-physical systems. As more cyber-physical systems are connected to the IoT, the pendulum between the "brick" versus the "click" is starting to swing back toward the "brick" again, where interactions, and often times close integrations, between cyber systems and physical systems are becoming increasingly important and bring new business priorities and operational requirements. Examples of cyber-physical systems include industrial control systems, smart cities, and connected cars and trains. In such systems, uninterrupted and safe operation is often the top priority. Taking a system offline for any reason can cause significant business loss or intolerable customer inconvenience, and therefore must be planned days, weeks, and even months in advance in some cases [17]. For example,

- Requiring cars to be brought to repair shops just to install software update packages can cause intolerable inconvenience and result in heavy cost to both car owners and carmakers.
- A nuclear reactor typically runs on 18-month cycles and any downtime can cause tens of thousands of dollars [15].
- Many other industrial control or manufacturing systems, such as car assembly plants and electrical power generators in the energy grids, have similar requirements for uninterrupted safe operations and require weeks to months lead times to plan for system down times.

As a result, unlike the routers, switches, personal computers, and smartphones in today's Internet, the timings and opportunities for updating the hardware and software in such cyber-physical systems can be severely limited. Many time-critical control applications, which need to be updated over time, cannot be moved to the Cloud due to delay, bandwidth, or other constraints. Therefore, a new computing and networking architecture will be needed to reduce the needs for the hardware and software in mission-critical systems to be updated over time.

Uninterrupted services with intermittent connectivity to the Cloud. Cloud services will have difficulty providing uninterrupted services to devices and systems that have intermittent network connectivity to the Cloud. Such devices include vehicles, drones, and oil rigs. For example, an oil rig in the ocean and far away from shore may have only satellite communication channels to connect to the Cloud. These satellite channels can suffer widely fluctuating quality and intermittent availability. However, applications such as data collection, data analytics, and controls for the oil rig have to be available even when the rig does not have network connectivity with the Cloud. As another example, when a car traverses an area where it loses Internet connectivity, many services and applications for the devices and people in the car must continue to be available. When a car breaks down in such an area and needs to have one of its electronic control unit (ECU) replaced before it can run again, the new ECU should be authenticated to prevent any unauthorized and potentially malware-infected ECUs from being installed on the vehicle. However, Cloud-based authentication services will not be available in this scenario.

New security challenges. Existing cyber security solutions for today's Internet, designed primarily for protecting enterprise networks, data centers, and consumer electronics, have focused on providing perimeter-based protections. In particular, a system or an individual device under protection is placed behind firewalls that work with intrusion detection and prevention systems to prevent security threats from breaking through the protected perimeters. Some resource-intensive security functions are also being moved to the Cloud. Existing Cloud-based security services continue to focus on providing perimeter-based protection, such as redirecting email and web traffic to the Clouds for threat detection, and redirecting access control requests to the Clouds for authentication and authorization processing. Should threats penetrate these protections, the common responses have been for human operators to take the system offline, clean up or replace compromised files and devices, and then put the system back online.

This existing security paradigm will no longer be adequate for addressing many new security challenges in the emerging IoT. Here, we discuss several such challenges.

Keeping security credentials and software up to date on a large number of devices. As the number and variety of the connected devices increase, a growing challenge will be how to manage the security credentials on these devices and how to keep the security credentials and security software on the devices up to date. Requiring every device to connect to the Cloud to update its security credentials and software will be impractical.

Protecting resource-constrained devices. Many resource-constrained devices in the IoT will not have sufficient resources to protect themselves adequately. These devices may have very long lifespans, and the hardware and software on them can be impractical to upgrade. Yet, these devices will need to remain secure over their long lifespans. For example, replacing any hardware on cars, which have already been sold to consumers, can create significant inconvenience to vehicle owners and result in heavy costs and reputation damages to carmakers. However, over a car's long lifespan that averages about 11.4 years [16], security threats will become significantly more advanced, many new threats will appear, and the mechanisms required to combat the growing threats will need to be enhanced and upgraded accordingly. Therefore, a fundamental question arises: How to protect a very large number of resource-constrained devices from security attacks?

Assessing the security status of large distributed systems in a trustworthy manner. IoT will support many large distributed systems. A connected transportation system, for example, may have thousands of devices deployed throughout a city to control traffic signals and communicate with vehicles. A large carmaker will need to ensure the security of tens of millions of cars on the road in a large country such as the USA. An oil and gas company may need to interconnect hundreds of remote sites such as oil rigs, exploration sites, refineries, and pipelines. A smart grid will consist of networked subsystems for metering, data collection, data aggregation, energy distribution, and demand response in multiple geographical areas.

Therefore, the ability to tell, in a trustworthy manner, whether a large number of distributed devices and systems are operating securely, will be essential. However, conventional approaches have difficulty meeting both the scalability and the trustworthy monitoring requirements at the same time.

Today's security health monitoring systems rely on collecting security status messages and log data from devices. These systems, however, can often generate untrustworthy results when applied in some IoT systems. For example,

- Many devices operating in physically unprotected environments can be compromised and used to send false information [21][22][23]. Adversaries can also easily use these compromised devices to form a local majority in many IoT scenarios. For example, they may compromise the majority of the smart meters in a house, a building, or even an entire region. As a result, existing mechanisms for detecting false information, which typically rely on the majority of the data sources to be honest (i.e., uncompromised and not malfunctioning), will no longer be adequate.
- Attackers can compromise a cyber-physical system and damage the physical equipment while keeping the messages to and from the system appear normal. A prime example is the Stuxnet attack on the Iranian nuclear facility – the Stuxnet worm masqueraded the attack by sending normal status messages to the system administers while spinning the nuclear reactor out of control [18][19][20].

To increase the trustworthiness of security status monitoring, remote attestation mechanisms allow a device to cryptographically prove its trustworthiness to a remote verifier [24][25]. A device makes a claim about certain properties of its hardware, software, or runtime environment to the verifier and uses its security credentials (e.g., a hardware-based root of trust and public key certificates) to vouch for these properties. The verifier then cryptographically verifies these claims.

However, existing remote attestation methods have focused on enabling an individual device to attest to its own trustworthiness. Many resource-constrained devices in the IoT will not be able to support processing-intensive remote attestation. Even when they can, forcing a large number of devices to perform remote attestation can result in prohibitively high cost and management complexity. Furthermore, existing remote attestation technology alone cannot handle the case where a device itself is not compromised but its sensory input is.

Responding to security compromises without causing intolerable disruptions. Today's incident response solutions rely predominately on brute-force mechanisms such as shutting down a potentially compromised system, reinstalling and rebooting its software, or replacing its components and subsystems. Such highly disruptive responses, which largely disregard how severe the compromises actually are, can cause intolerable disruptions to mission-critical systems. However, maintaining uninterrupted and safe operation, even when the system is compromised, is often the highest priority for mission-critical systems such as industrial control systems, manufacturing plants, connected vehicles, drones, and smart grids. For example:

- An electric power generator may be infected by a malware that merely seeks to steal power for unauthorized use.

Shutting down the power generator could cause severe disruptions to the smart grid and excessive power outages.

- Industrial control systems often have little tolerance for down time. Manufacturing operations can also have critical safety implications. As a result, manufacturers usually value uninterrupted operation and safety over system integrity. This means that hardware and software updates can only be installed during a system's scheduled down times, which have to be short and far between, rather than every time any security compromise is detected.
- A connected car can be infected by malware that can become active while the car is in motion. While the malware can do a range of damages to the vehicle and can put the driver and passengers in harm's way, abruptly shutting down the engine each time any malware is detected could be an even quicker and surer way to cause deadly traffic accidents.
- If a drone flying midair is abruptly turned off just because a security compromise is detected, it can crash from the sky onto people, houses, and other properties to cause serious damages. Instead, safe landing or safe return-home mechanisms will be essential for responding to such security threats that can compromise a drone's flight.
- A server in a data center may be infected by a spyware that seeks to steal commercial secrets. While allowing such a compromised server to continue to operate could give the attacker access to some sensitive data, it may not directly impact the data-center's mission-critical services. If we shut down the server, or halt the execution of the malware-infected files to wait for the malware to be removed, the system downtime could cause significantly more damage, including causing vast economic losses to the data center operator, business disruptions to those who count on the data centers to operate their businesses, and inconvenience to other users of the data center.

Therefore, today's highly disruptive incident response paradigm will no longer be adequate for securing the many mission-critical systems in the emerging IoT.

III. THE EMERGING ERA OF FOG

Filling the technology gaps in supporting IoT will require a new computing and networking architecture—Fog—that distributes computing, control, storage, and networking functions closer to end user devices.

Compared to the Cloud, Fog stands out along the following three dimensions:

- Carry out a substantial amount of data *storage* at or near the end user (rather than storing data only in remote data centers).
- Carry out a substantial amount of *computing* and *control* functions at or near the end user (rather than performing all these functions in remote data centers and cellular core networks). Such computing and control functions can include, for example,
    - Applications for end users and their devices.
    - Functions for controlling and operating end-user systems such as manufacturing systems, vehicles, and smart grids.
    - Services for managing end-user networks, systems, and applications.
    - Services for supporting Cloud-based applications, such as collecting and preprocessing data to be sent to the Cloud.
- Carry out a substantial amount of *communication* and *networking* at or near the end user (rather than routing all network traffic through the backbone networks). This can include, for example, ways to improve the performance and scalability of local peer-to-peer networks, intelligent control of radio access networks (RANs), organize and manage local mobile ad-hoc networks, and integrate local ad-hoc networks with the infrastructure networks.

Fog is a natural extension of Cloud: Fog and Cloud complement each other to form a mutually beneficial and inter-dependent service continuum between the Cloud and the endpoints to make computing, storage, control, and communication possible anywhere along the continuum.

- Fog enables a service continuum: For example, to the wearable devices, a mobile phone may become the Fog to provide local control and analytics applications to the wearable devices. When the user is inside her vehicle, the vehicle can become the Fog for her mobile phone to allow many smartphone functions, such as display, user interface, audio, phone book, to be moved to the vehicle. Roadside traffic control equipment can in turn serve as the Fog for the vehicle to provide traffic information to the vehicle.
- Fog and Cloud are inter-dependent: For example, Cloud services may be used to manage the Fog. Fog can act as the proxy of the Cloud to deliver Cloud services to endpoints, and act as the proxy of the endpoints to interact with the Cloud. Furthermore, Fog can be the beachheads for collecting and aggregating data for the Cloud.
- Fog and Cloud are mutually beneficial: Some functions are naturally more advantageous to be carried out in the Fog while others in the Cloud. Determining which functions should be carried out in the Fog and how the Fog should interact with the Cloud will be key aspects of Fog research and development.

Traditionally, services and applications are provided with large, centralized, expensive, and hard-to-innovate "boxes" such as the Service Gateways (S-GW) and packet Data Network Gateways (PDN-GW) in the LTE core, large servers in a data center, and the core gateways and routers in a wide-area-network backbone. The traditional view is that the edge uses the core networks and data centers. The Fog view is that the edge *is* part of the core network and a data center.

Table 1 outlines the main characteristics of Fog with a comparison to Cloud.

Table 1: Main characteristics of Fog as compared to Cloud

	Cloud	**Fog**

Location and Model of Computing	Centralized in a small number of big data centers.	Often distributed in many locations, potentially over large geographical areas, closer to users along the Cloud-to-Thing continuum. Distributed Fog nodes and systems can be controlled in centralized or distributed manners.
Size	Cloud data centers are very large in size, each typically contain tens of thousands of servers.	A Fog in each location can be small (e.g., one single fog node in a manufacturing plant or onboard a vehicle) or as large as required to meet customer demands. A large number of small Fog nodes may be used to form a large Fog system.
Deployment	Require sophisticated deployment planning.	While some Fog deployments will require careful deployment planning, Fog will enable ad-hoc deployment with no or minimal planning.
Operation	Operate in facilities and environments selected and fully controlled by Cloud operators. Operated and maintained by technical expert teams. Operated by large companies.	May operate in environments that are primarily determined by customers or their requirements. A Fog system may not be controlled or managed by anyone and may not be operated by technical experts. Fog operation may require no or little human intervention. May be operated by large and small companies, depending on size.
Applications	Support predominately, if not only, cyber-domain applications. Typically support applications that can tolerate round-trip delays in the order of a few seconds or longer.	Can support both cyber-domain and cyber-physical systems and applications. Can support significantly more time-critical applications that require latencies below tens of milliseconds or even lower.
Internet Connectivity and Bandwidth Requirements	Require clients to have network connectivity to the Cloud for the entire duration of services. Long-haul network bandwidth requirements grow with the total amount of data generated by all clients.	Can operate autonomously to provide uninterrupted services even no or intermittent Internet connectivity. Long-haul network bandwidth requirements grow with total the amount of data that need to be sent to the Cloud after being filtered by the Fog.

A. *Fog Architectural Advantages*

A common denominator underlying Fog is that Fog distributes the resources and services of computation, communication, control, and storage closer to the users. A Fog architectures may be fully distributed, mostly centralized, or somewhere in between. The Fog architecture and the applications it supports ("Fog applications") may be virtualized and implemented completely in software. They may also be implemented in dedicated hardware and software.

A Fog architecture will allow the same application to run anywhere, reducing the need for specialized applications dedicated just for the Cloud, just for the endpoints, or just for the edge devices. It will enable applications from different suppliers to run on the same physical platform without mutual interference. It will provide a common lifecycle management framework for all applications, offering capabilities for composing, configuring, dispatching, activating and deactivating, adding and removing, and updating applications. It will further provide a secure execution environment for Fog services and applications. Fog will with integrate with Cloud to enable seamless end-to-end services.

Fog's main advantages can be summarized as CEAL:

1. **Cognition**: Awareness of client-centric objectives. A Fog architecture, aware of customer requirements, can best determine where to carry out the computing, storage, and control functions along the Cloud-to-Thing continuum. Fog applications, being close to the end users, can be built to be better aware of and closely reflect customer requirements.

2. **Efficiency**: Pooling resources along the Cloud-to-Thing continuum. Fog can distribute computing, storage, and control functions anywhere between the Cloud and the endpoint to take full advantage of the resources available along this continuum. It can also allow applications to leverage the otherwise idling computing, storage, and networking resources abundantly available on network edge and end-user devices such as tablets, laptops, smart home appliances, connected vehicles and trains, and network edge routers. Fog's closer proximity to the endpoints will enable it to be more closely integrated with the end-user systems to enhance overall system efficiency and performance. This is especially important for performance-critical cyber-physical systems.

3. **Agility**: Rapid innovation and affordable scaling. It is usually much faster and cheaper to experiment with client and edge devices. Rather than waiting for vendors of large network and Cloud boxes to initiate or adopt an innovation. Fog will

make it easier to create an open market place for individuals and small teams to use open APIs (Application Programming Interfaces), open SDKs (Software Development Kits), and the proliferation of mobile devices to innovate, develop, deploy, and operate new services.

4. **Latency**: Real-time processing and cyber-physical system control. Fog enables data analytics at the network edge and can support time-sensitive control functions for local cyber-physical systems. This is essential for not only commercial applications but also for the Tactile Internet vision to enable embedded AI applications with millisecond reaction times.

These advantages in turn enable new services and business models, and may help broaden revenues, reduce cost, or accelerate product rollouts.

B. *Fog Helps Address IoT Challenges*

Fog can provide effective ways to overcome many limitations of the existing computing architectures that rely only on computing in the Cloud and on end-user devices. Table 1 shows, as an example, how Fog can help address the IoT challenges we have discussed in Section II.

Table 2: Fog provides effective ways to address IoT challenges.

IoT Challenges	How Fog Can Help
Latency Constraints	Fog, performing data analytics, control, and other time-sensitive tasks close to end users, is the ideal and often the only option to meet the stringent timing requirements of many IoT systems.
Network Bandwidth Constraints	Fog enables hierarchical data processing along the Cloud-to-Things continuum, allowing processing to be performed where it can balance between application requirements and available networking and computing resources. This also reduces the amount of data that needs to be sent to the Cloud.
Resource-Constrained Devices	Fog can carry out resource-intensive tasks on behalf of resource-constrained devices when such tasks cannot be moved to the Cloud due to any reason, hence reducing these devices' complexity, lifecycle costs, and energy consumption.
Uninterrupted Services with Intermittent Connectivity to the Cloud	A local Fog system can operate autonomously to ensure non-interrupted services even when it has intermittent network connectivity to the Cloud.
New IoT Security Challenges	A Fog system can, for example, 1) act as the proxies for resource-constrained devices to help manage and update the security credentials and software on these devices, 2) perform a wide range of security functions, such as malware scanning, for the resource-constrained devices to compensate the limited security functionality on these devices, 3) monitor the security status of nearby devices, and 4) take advantage of local information and context to detect threats on a timely manner.

Proof-of-Concept (POC) trials are demonstrating the business value and technology necessity of Fog. For example, in late 2015, Cisco conducted a successful POC in Barcelona, where Fog made smart city applications more cost-effective and manageable. Barcelona envisions deploying thousands of roadside cabinets throughout the city to optimize traffic management, energy management, and water and waste management. Before they could turn this vision into reality, the city faced two major challenges. First, the traditional way of adding new applications by adding dedicated new gateways and servers in every roadside cabinet is no longer feasible due to limited cabinet space. Second, the siloed applications have been using siloed application management systems, which made the system excessively expensive to deploy, operate, and maintain. Fog provided a solution. A single Fog node provided a common platform at each cabinet for all services, and allowed applications from different suppliers to coexist without interfering with each other. It provided a unified platform to support networking, security, and lifecycle management for all applications, which reduced the systems costs and allowed application providers to focus on developing applications rather than providing specialized hardware and software to host and manage their applications.

C. *Fog Enables New and Disruptive Business Models*

Fog will enable new, and potentially highly disruptive, business models for computing and networking. For example,

- With Fog, routers, switches, application servers, and storage servers will converge into Fog nodes. Such a transformation can significantly reshape the networking, server, and software industry landscape.
- Fog-as-a-Service (FaaS) will enable new business models to deliver services to customers. Unlike the Clouds that are mostly operated by large companies who can afford to build and operate huge data centers, FaaS will enable companies, big and small, to deliver private or public computing, storage, and control services at different scales to meet the needs of a wide variety of customers.
- Fog also provides a new way for network service providers to add value to customers in a new net-neutrality world. Consider, for example, the impact of the United States Federal Communications Commission (FCC) Title II Ruling. The FCC vote in February 2015 to classify Internet services, including mobile services, as a "utility" under Title II regulatory mandate, may further push network innovation to the edge in the US. A new regulatory environment does not mean networks cannot be engineered and managed anymore, but we may need different vantage points of control: not from inside the network but from around the end users. For example, today network operators can pick which lane (WiFi, Macro-cellular, and Femtocell) a user device should be in. Since different lanes have different speeds and different payment

system/amount, such practice may not be allowed any more in the US. Instead, we need to better design system where each user device must choose which lane to be in for itself. The challenge resulting from Title II regulation is a "hanging sword" that chills the deployment of network infrastructure innovations, as risk-Return balance now tips towards "keep the network as is." However, as long as the government does not prohibit end-user choices, then we can run networking from the edge, through client/home-driven control/configuration.

IV. FOG USE CASE STUDIES

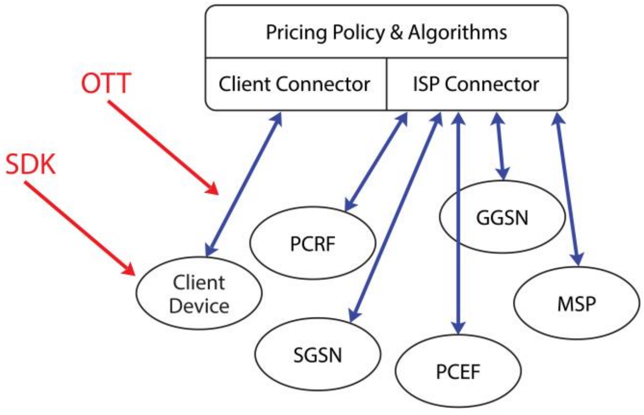

Fig. 2. SDK sitting inside clients can enable network inference and configuration.

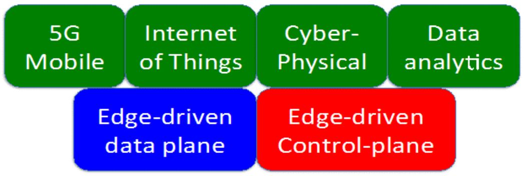

Figure 1: Data plane and control plane of Fog enable different applications

Architectural R&D asks the question of "who does what, at what timescale, and how to put the modules back together?" As an architecture, Fog supports a variety of applications, including those typically associated with IoT and those often viewed as part of 5G or data analytics and data management. Fog is an architecture for computing, storage, as well as for networking. In particular, Fog architecture consists of both data plane and control plane, each with a rapidly growing number of examples across protocol layers from the physical layer to the application layer:

- *Examples of Data plane of Fog:*
  - Pooling of clients idle computing/storage/bandwidth resources and local content
  - Content caching at the edge and bandwidth management at home
  - Client-driven distributed beam-forming
  - Client-to-client direct communications (e.g., FlashLinQ, LTE Direct, WiFi Direct, Air Drop)
  - Cloudlets and micro data-centers
- *Examples of Control plane of Fog:*
  - Over the Top (OTT) content management
  - Fog-RAN: Fog driven radio access network
  - Client-based HetNets control
  - Client-controlled Cloud storage
  - Session management and signaling load at the edge
  - Crowd-sensing inference of network states
  - Edge analytics and real-time stream-mining

Data-plane of Fog has been more extensively studied, e.g., [2]. In the following, we highlight a few particular cases that illustrate the potential and challenges of Fog control plane, such as the inference, control, configuration and management of networks:

While some of these case studies are core topics in what many people imagine would partially define "5G:" HetNets/small cell/densification, over the top service provisioning, cognitive radio and crowd-sensing, other case studies point toward architectural thinking for IoT services, questions about ownership, control and visibility of personal area networks, such as "should Apple Watch and the like have their own data plan?" that will help define the balance of power between the "AT&T"s and the "Apple"s of the world. If the network in or around the end users have a logical topology that looks like a star, with a fixed gateway (e.g., iPhone), the visibility, control, and value-added by network operators will be drastically different than in the alternative scenario where the gateways are dynamically chosen or the Things can sometimes have direct communication paths without a gateway.

Case 1: Crowd-sensing LTE states (in commercial deployment). Through a combination of passive measurement (e.g., RSRQ), active probing (e.g., packet train), application throughput correlation and historical data mining, a collection of client devices may be able to, in real-time and useful accuracy, infer the states of an eNB such as the number of Resource Blocks used [3].

Case 2: OTT network provisioning and content management (in commercial deployment). The traditional approach to innovating networks is to introduce another box inside the network, possibly a virtualized box but a box nonetheless. Fog directly leverages the “things” and phones instead, and removes the dependence on boxes-in-the-network altogether. With SDKs sitting behind apps on client devices, through tasks such as URL wrapping, content tagging, location tracking, behavior monitoring, network services can be innovated much faster. In this case, the client SDKs collectively work through a controller (in the cloud as hosted say by Amazon) but bypass most of the cellular core network (a second type of cloud).

Case 3: Client-based HetNets control (in 3GPP standards). Coexistence of heterogeneous networks (e.g., LTE, femto, WiFi) coexistence is a key feature in cellular networks today. Rather than through network operator control, each client can observe its local conditions and make decision on which network to join. Through randomization and hysteresis, such local actions may emerge globally to converge to a desirable configuration [4]. In the case of hybrid control of HetNets, the fog-cloud interface allows real-time network configuration be carried out by the clients themselves, while over longer timescale parameters like RAT stability attribute or hysteresis values can pass from the cloud (wireless core network) to the clients.

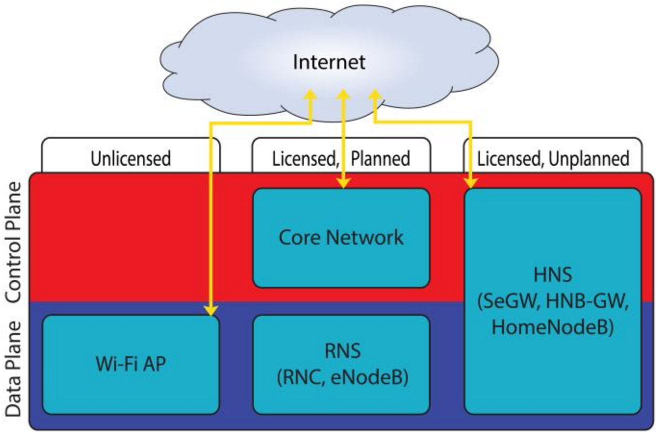

Figure 3: Co-existence of heterogeneous networks may be managed in part by clients

Case 4: “Shred and Spread” Client-controlled Cloud storage (in beta trial). By decoupling massive cheap storage (in the Cloud) from client side control of privacy (in the Fog), we can achieve the best of both worlds. For example, by shredding a file on the fog side and then spreading the bytes across multiple public clouds, in a client shim layer, of a given file across multiple Cloud storage providers, it can be assured that privacy of the data is maintained even if encryption key is leaked by any given Cloud provider [5].

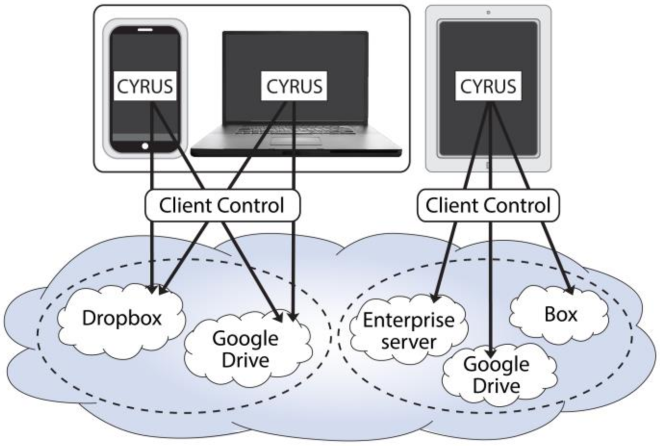

Figure 4: Shred and Spread (CYRUS project) stores in Cloud but controls in Fog

Case 5: Real-time stream mining for embedded AI (in beta trial). Consider virtual reality tasks associated with Google Glass. Some of the information retrieval and computation tasks may be carried out on the Glass (a “wearable thing”), some on the associated phone (a client device), some on the home storage (an edge device), and the rest in the Cloud. An architecture of successive refinement may leverage all of these devices at the same time, with an intelligent division of labor across them [6].

Case 6: Borrowing bandwidth from neighbors in D4D (in beta trial). When multiple devices belonging to the same person, to relatives or to employees of the same company are next to each other, one can ask the others to share their LTE/WiFi bandwidth by downloading other parts of the same file and transmitting, via WiFi Direct, client to client [7].

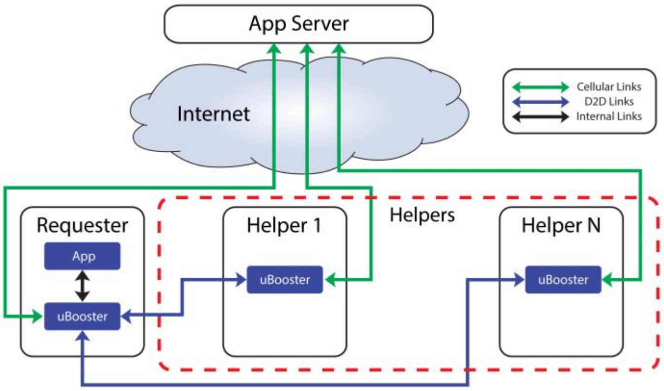

Figure 5: Idle resources in client devices can be pooled in D4D for more efficient use

Case 7: Bandwidth management at home gateway (in beta trial). By adapting the home set-top box/gateway, the limited broadband capacity is allocated among competing users and application sessions, according to each session’s priority and individual preferences. A prototype on a commodity router demonstrates a scalable, economical and accurate control of capacity allocation on the edge [8].

Case 8: Distributed beam-forming (in lab demonstration). Fog can also happen in the physical layer, for example, by exploiting multi-user MIMO to improve throughput and reliability when a client can communicate with multiple WiFi access points. For uplink, we can use multi-user beam-forming so that the client can send multiple data streams to multiple APs simultaneously. For downlink, we can use interference nulling so that the client can decode parallel packets from multiple APs. These can be done entirely on the client side [9].

For more references for these examples and more, please see an initial list of over 100 recent publications on eight different topics under Fog at http://Fogresearch.org

V. Open Questions and Research Challenges

As is typical of any emergent area of R&D, many themes in Fog are not completely new, and instead are evolved versions of accumulated transformations in the past decade or two:

- Compared to peer-to-peer (P2P) networks in the mid-2000s, Fog is not just about content sharing (or data plane as a whole), but also network measurement, network management, service enablement, and real-time control of cyber-physical systems.
- Compared to mobile ad-hoc network (MANET) research, Fog will build upon much more powerful, diverse, and often off-the-shelf edge devices, applications, and end-to-end hierarchical networks enabled by broadband wireless and wired networks.
- Compared to the generic edge-networking work in the past, Fog adds a new layer of meaning to the end-to-end principle: in addition to optimizing among themselves, edge devices, collectively measuring and controlling the rest of the network, will collaborate with the Cloud to enable end-to-end services along the Cloud-to-Thing continuum.

Along with several other network architecture themes with longer histories, Information-Centric Networks (ICN), Software-Defined Networks (SDN), Network Function Virtualization (NFV), Fog is revisiting the foundation of how to *architecture computing and networking:* who does what and how to glue them back together:

- ICN: *Redefine* functions (to operate on digital objects rather than just bytes)
- SDN: *Separate* control plane from data plane, and allow the control plane to be implemented in software.
- NFV: *Virtualize* functions (through centralized control plane).
- Fog: *Relocate* functions (to the network edge and along the Cloud-to-Things continuum).

While Fog does not have to rely on virtualization or to be information-centric or software-defined, one can envision an information-centric and virtualized Fog since these branches are complementary to each other and can be enablers for Fog.

Fog also includes both mobile and wireline networks, and traverses edge, access and the wearables. Supporting mobile edge computing inside a RAN will require many of the same functions of an end-to-end Fog architecture to, for example, distribute, orchestrate, manage, and secure the applications and application enablement platforms. Fog, however, is broader than just supporting mobile edge computing. Fog is an architecture for distributing computing, storage, control, and networking services anywhere along the Cloud-to-Thing continuum, over and inside wireless and wireline networks, and supporting both mobile and wireline network applications.

As in any emergent area in its infant age, there is no shortage of challenging questions in Fog, some of which continue from earlier study of P2P, MANET and Cloud, while others are driven by a confluence of recent developments in network engineering, user devices, and user experience. Next, we discuss several categories of Fog research challenges.

Fog interfaces with Cloud, other Fogs, Things, and end users: The fundamental question of architecture is "who does what, at what timescale, and how to put them back together?" In the case of Fog, the question becomes: 1) which tasks should go to the Fog (e.g., those requiring real-time processing, end user objectives or low-cost leverage of idle resources), 2) which go to the Cloud (e.g., massive storage, heavy-duty computation, or wide-area connectivity), 3) which go to the Things, and 4) how the Fog, the Cloud, and the Things should interact with each other. The Fog architectures should allow computing, storage, and networking tasks to be dynamically relocated among the Fog, the Cloud, and the Tings.

Therefore, the interfaces for Fog to interact with the Cloud, other Fogs, and the Things and users, as illustrated in Figure 6, must 1) facilitate flexible, and in some cases dynamic, relocation of the computing, storage, and control functions among these different entities, 2) enable convenient user access to Fog services, and 3) allow efficient and effective lifecycle management of the system and services.

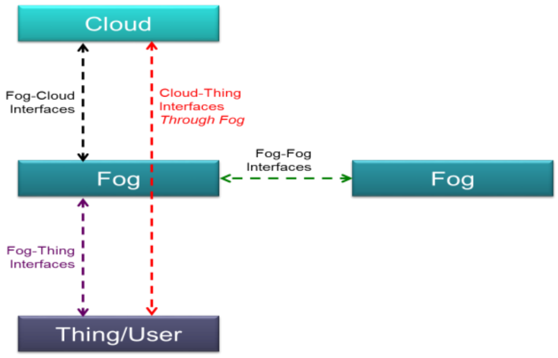

Figure 6: Fog interfaces.

- *Fog-Cloud Interfaces*: The Fog-Cloud interfaces will be needed to support Fog-Cloud collaborations to provide end-to-end services. It will support functions to, for example, allow:
  - Fog to be managed from the Cloud.
  - Fog and Cloud to send data to each other.
  - Cloud to distribute services onto Fog.
  - Cloud services to be provided to Fog
  - Cloud services to be provided through Fog to Things and end users.
  - Fog services to be provided to Cloud.
  - Fog and Cloud to collaborate with each other to deliver end-to-end services.

It is essential to determine what information should be passed across the Fog-Cloud interface, the frequency and granularity of such information, and how the Fog and the Cloud should react to the information.

- *Fog-to-Fog Interfaces:* Different Fog nodes or systems may collaborate with each other to jointly support an application. For example, multiple Fog systems can share the data storage and computing tasks for one or multiple users or applications. Different Fog nodes or systems can also collaborate to serve as backups for each other. An important question is therefore how to design the interface and protocols to enable different Fog nodes in the same Fog system, and different Fog systems, to collaborate.
- *Fog-to-Thing/User Interfaces*: Fog will provide services to a wide range of end users and devices with widely varying capabilities. The Fog-Thing interface and Fog-User interface will be essential to allow Things and end users to access Fog services in user-friendly, resource-efficiently, and secure ways.

Fog-enabled edge and access networking: Fog can be used to support networking at the edge. For example, Fog can provide services to help network edge devices and end-user devices (e.g., vehicles, drones, industrial and consumer robots, smartphones, and virtual reality gaggles) form local networks, providing temporary security credentials to these local devices to help them establish trustworthy communications, and act as local application servers and data storage servers for the edge networks. Some Fog functions for supporting such edge networking may be implemented on the end-user devices. In such cases, how Fog functions interface with the operating systems and hardware of the end-user devices becomes essential. More than just using D4D for pooling idle edge resources as discussed in previous sections, new protocol stacks for end-user devices to support Fog-enabled edge networking may be needed.

Security: Compared to Cloud, Fog presents new security challenges. Distributed systems, such as distributed Fog, are in general more vulnerable to attacks than centralized systems, such as Clouds. While Cloud operates in heavily protected facilities selected and controlled by Cloud operators, Fog often needs to operate in more vulnerable environments – where they can best meet customer requirements and often wherever users want them to. Many Fog systems will be significantly smaller than Clouds (e.g., a Fog node on a vehicle) and hence may not have as much resources as the Clouds to protect themselves. Furthermore, each Fog system may not have the global intelligence necessary for detecting threats.

At the same time, however, Fog's proximity to end users and locality on the edge enable it to help address certain new IoT security challenges as discussed in previous sections. Fog can, for example, act as the first nodes for access control and traffic encryption, provide contextual integrity and isolation, serve as the aggregation and control points for privacy-sensitive data before the data leaves the edge, and act as the proxies of resource-constrained devices to carry out selected security functions for these resource-constrained devices.

Incentivization of client participation: In some IoT use cases, it is not too many un-trustworthy clients that create concerns but too few clients willing to participate. This can be the case when, for example, clients are expected to voluntarily contribute their computing or storage resources or to collaborate with each other to support applications. Market systems and incentive mechanisms will become useful.

Convergence and consistency: Local interactions could lead to divergence, oscillation, and inconsistency of global system states, which are typical issues in distributed systems and can become more acute in a massive, under-organized, possibly mobile crowd with diverse capabilities and virtualized pool of resources shared unpredictably. Use cases in edge analytics and stream mining provide additional challenges on this recurrent challenge in distributed systems.

End-to-end architectural tradeoffs: Fog will create new opportunities for us to design end-to-end systems to achieve better tradeoffs between *distributed and centralized* architectures, between what stays local and what goes global, and between careful deployment planning and resilience through redundancy. Logical Fog system topologies, statically or dynamically established, over the same underlying physical Fog network can be used to support a spectrum of architectures from completely centralized to fully distributed.

To address the above challenges, we need both

- Fundamental research, across networking, device hardware/OS, pricing, HCI and data science, and
- Industry-academia interactions, as exemplified in the Open Fog Consortium, a global, non-profit consortium launched in November 2015 with founding members from ARM, Cisco, Dell, Intel, Microsoft and Princeton University EDGE Lab.

What will be the fundamental technology enabler(s) for Fog: When we examine the past significant advancements of the state of the art in computing and networking, we may often be able to point to one or a small set of technologies that form the fundamental enablers for the advancements. For example, the TCP/IP protocols started the Internet. Virtualization has been powering Cloud computing. What will be *the* fundamental technology enabler, or that small set of fundamental technology enablers, that will power Fog?

Indeed, Fog is starting to reshape the future landscape of multiple industries, driving innovation across the entire industry food chain, including the following:

- End user experience providers (e.g., GE, Toyota, …)
- Network operators (e.g., AT&T, Verizon, Comcast, …)
- Network equipment vendors (e.g., Cisco, Nokia, Ericsson, Huawei, …)
- Cloud service providers (e.g., VMWare, Amazon, …)
- System integrators (e.g., IBM, HP, …)
- Edge device manufacturers (e.g., Linksys, …)
- Client and IoT device manufacturers (e.g., Dell, Microsoft, Apple, Google, …)
- Computer chip suppliers (e.g., Intel, ARM, Qualcomm, Broadcom, …)

2016 is an interesting year to start systematically exploring what Fog might look like and the differences it will bring to the world of networking and computing in the next 15 years.

ACKNOWLEDGEMENTS

The authors are grateful for the inspiring conversations with many colleagues in industry and academia, especially Flavio Bonomi, Russell Hsing, Bharath Balasubramanian, Sangtae Ha, Junshan Zhang, Raj Savoor, John Smee, Chonggang Wang, and representatives of ARM, Cisco, Dell, Intel, Princeton University, and Microsoft in the Open Fog Consortium.

REFERENCES

[1] F. Bonomi, R. Milito, P. Natarajan and J. Zhu, "Fog computing: A Platform for Internet of Things and Analytics," in *Big Data and Internet of Things*: *A Roadmap for Smart Environments*, Springer, March 2014, pp 169-186.

[2] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based Cloudlets in mobile computing," *IEEE Transactions on Pervasive Computing*, 2009.

[3] A. Chakraborty, V. Navda, V. N. Padmanabhan and R. Ramjee, "Coordinating cellular background transfers using LoadSense," *Mobicom* 2013.

[4] E. Aryafar, A. Keshavarz-Haddard, M. Wang and M. Chiang, "RAT selection games in HetNets," *IEEE INFOCOM* 2013.

[5] J. Chong, C. Joe-Wong, S. Ha and M. Chiang, "CYRUS: Toward client-defined Cloud storage," Proceedings of EuroSys 2015.

[6] L. Canzian and M. van der Schaar, "Real time stream mining: Online knowledge extraction using classifier networks," *IEEE Networks Special Issue on Networking for Big Data* 2014.

[7] Z. Zhang, J. Zhang and L. Ying, "Multimedia streaming in cooperative mobile social networks," Preprint.

[8] F. M. F. Wong, S. Ha, C. Joe-Wong, Z. Liu and M. Chiang, "Mind Your Own Bandwidth: Adaptive Traffic Management on Network Edge," *IEEE IWQoS 2015*.

[9] Y. Du, E. Aryafar, J. Camp and M. Chiang, "iBeam: Intelligent client-side multi-user beamforming in wireless networks," Proceedings of IEEE INFOCOM 2014.

[10] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikoli´c, "Design of a Low-Latency, High-Reliability Wireless Communication System for Control Applications", 2014 IEEE International Conference on Communications (ICC), pp. 3829–3835. IEEE (2014).

[11] R. Kelly, "Internet of Things Data to Top 1.6 Zettabytes by 2022" <https://campustechnology.com/articles/2015/04/15/internet-of-things-data-to-top-1-6-zettabytes-by-2020.aspx> [Available: April 7, 2016].

[12] L. Mearian, <http://www.computerworld.com/article/2484219/emerging-technology/self-driving-cars-could-create-1gb-of-data-a-second.html> [Available: April 7, 2016].

[13] N. Cochrane, "US smart grid to generate 1000 petabytes of data a year" <http://www.itnews.com.au/news/us-smart-grid-to-generate-1000-petabytes-of-data-a-year-170290#ixzz458VaITi6> [Published: March 23, 2010] [Available: April 7, 2016].

[14] National Institute of Standards and Technology (NIST), U.S. Department of Commerce, Special Publication 800-82, "Guide to Industrial Control Systems (ICS) Security", June 2011.

[15] W. Ashford, "Industrial Control Systems: What Are the Security Challenges?", http://www.computerweekly.com/news/2240232680/Industrial-control-systems-What-are-the-security-challenges [Published: October 15, 2014], [Available: January 28, 2016].

[16] U.S. Department of Transportation, Bureau of Transportation Statistics. <http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national_transportation_statistics/html/table_01_26.html_mfd> [Available: March 2, 2016].

[17] G. Gan, Z. Lu, and J. Jiang, "Internet of Things Security Analysis", *2011 International Conference on Internet Technology and Applications* (iTAP), August 16-18, 2011.

[18] N. Falliere, L. O Murchu, and E. Chien, "W32.Stuxnet Dossier", Symantec Security Response, Version 1.4, February 2011.

[19] K. Zetter, "Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon", ISBN-13: 978-0770436179, November 11, 2014.

[20] <https://en.wikipedia.org/wiki/Stuxnet> [Available: March 2. 2016].

[21] L. Delgrossi and T. Zhang, "Vehicle Safety Communications: Protocols, Security, and Privacy", published by John Wiley & Sons, 2012, ISBN-10: 1118132726, ISBN-13: 978-1118132722.

[22] T. Zhang, H. Antunes, and S. Aggarwal, "Defending Connected Vehicles against Malware: Challenges and a Solution Framework", *IEEE Internet of Things Journal*, Vol. 1, No. 1, February 2014.

[23] T. Zhang, H. Antunes, and S. Aggarwal, "Securing Connected Vehicles End to End", *SAE 2014 World Congress and Exhibition*, Detroit, Michigan, USA, April 8 – 10, 2014.

[24] R. Chen, L. Wei, H. Zou, and M. Zhai, "A TCM-Based Remote Anonymous Attestation Protocol for Power Information System", *International Power, Electronics and Materials Engineering Conference 2015 (IPEMEC 2015)*, May 16-17, 2015, Dalian, China.

[25] A. Francillon, Q. Nguyen, K. B. Rasmussen, and G. Tsudik, "A Minimalist Approach to Remote Attestation", *Conference on Design, Automation & Test in Europe (DATE)*, 2014.

Mung Chiang is the Arthur LeGrand Doty Professor of Electrical Engineering at Princeton University. His research on networking received the 2013 Alan T. Waterman Award, the highest honor to US young scientists and engineers. His textbook "Networks: Friends, Money and Bytes" and online course reached 250,000 students since 2012. He founded the Princeton EDGE Lab in 2009, which bridges the theory-practice gap in edge networking research by spanning from proofs to prototypes. He co-founded a few startups in mobile, IoT and big data areas and co-founded the Open Fog Consortium. Chiang is the Director of Keller Center for Innovations in Engineering Education at Princeton University and the inaugural Chairman ofPrinceton Entrepreneurship Council.

Tao Zhang (F'00) received the B.S. and the M.S. degrees in electrical engineering from Northern Jiaotong University in Beijing, China, and the Ph.D. degree in electrical and computer engineering from University of Massachusetts, Amherst, MA, USA.

He joined Cisco Systems in 2012 as the Chief Scientist for Smart Connected Vehicles. Since then, he has also been leading initiatives to develop strategies, architectures, technology, and eco-systems for the Internet of Things (IoT) and Fog Computing. From 1995 to 2012, he was with Telcordia Technologies (formerly Bell Communications Research or Bellcore), where he was Chief Scientist and Director of Mobile and Vehicular Networking. For over 25 years, he has been in various technical and executive positions, directing research and product development. He holds over 50 US patents and has co-authored two books "Vehicle Safety Communications: Protocols, Security, and Privacy" (John Wiley & Sons, 2012) and "IP-Based Next Generation Wireless Networks" (John Wiley & Sons, 2004).

Dr. Zhang is a co-founder and a Board Director of the Open Fog Consortium, and the CIO and a Board Governor of the IEEE Communications Society. He was a founding Board Director of the Connected Vehicle Trade Association (CVTA).