Compute-First Networking

Summary of Jan 24th -25th 2019 Workshop

IETF 104, Prague

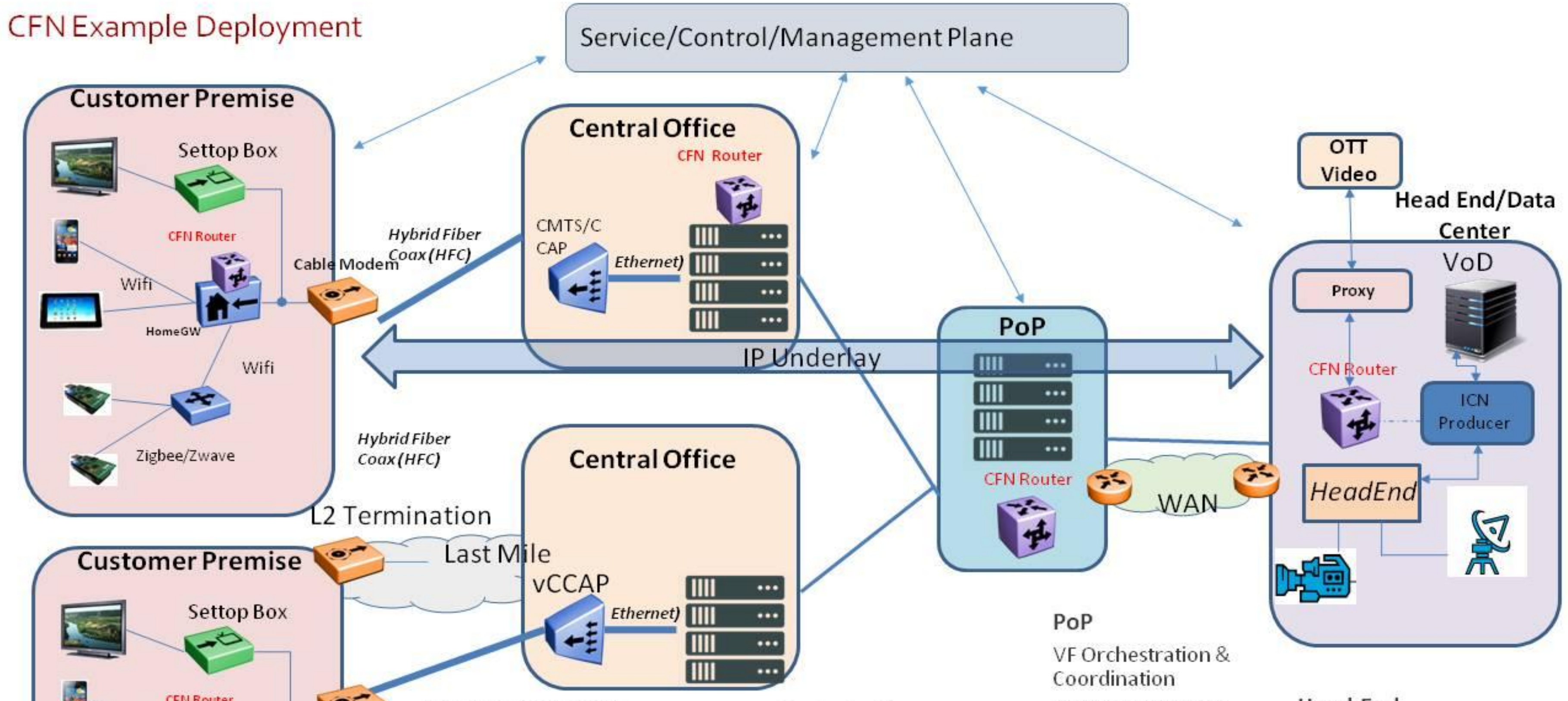
HUAWEI TECHNOLOGIES CO., LTD.

Berkeley Workshop on 1/24-1/25:

- A discussion of Compute-First Networking, trying to answer questions:
  - How CFN will enable new business models and new service scenarios (Who will sell what new services to whom?)
  - Gap Analysis/problem statement (Why the incremental extension of current practice is not enough?)
  - Technical proposals (what architecture/solution is required?)

  - Other questions that you would like to discuss?

CFN: the big picture



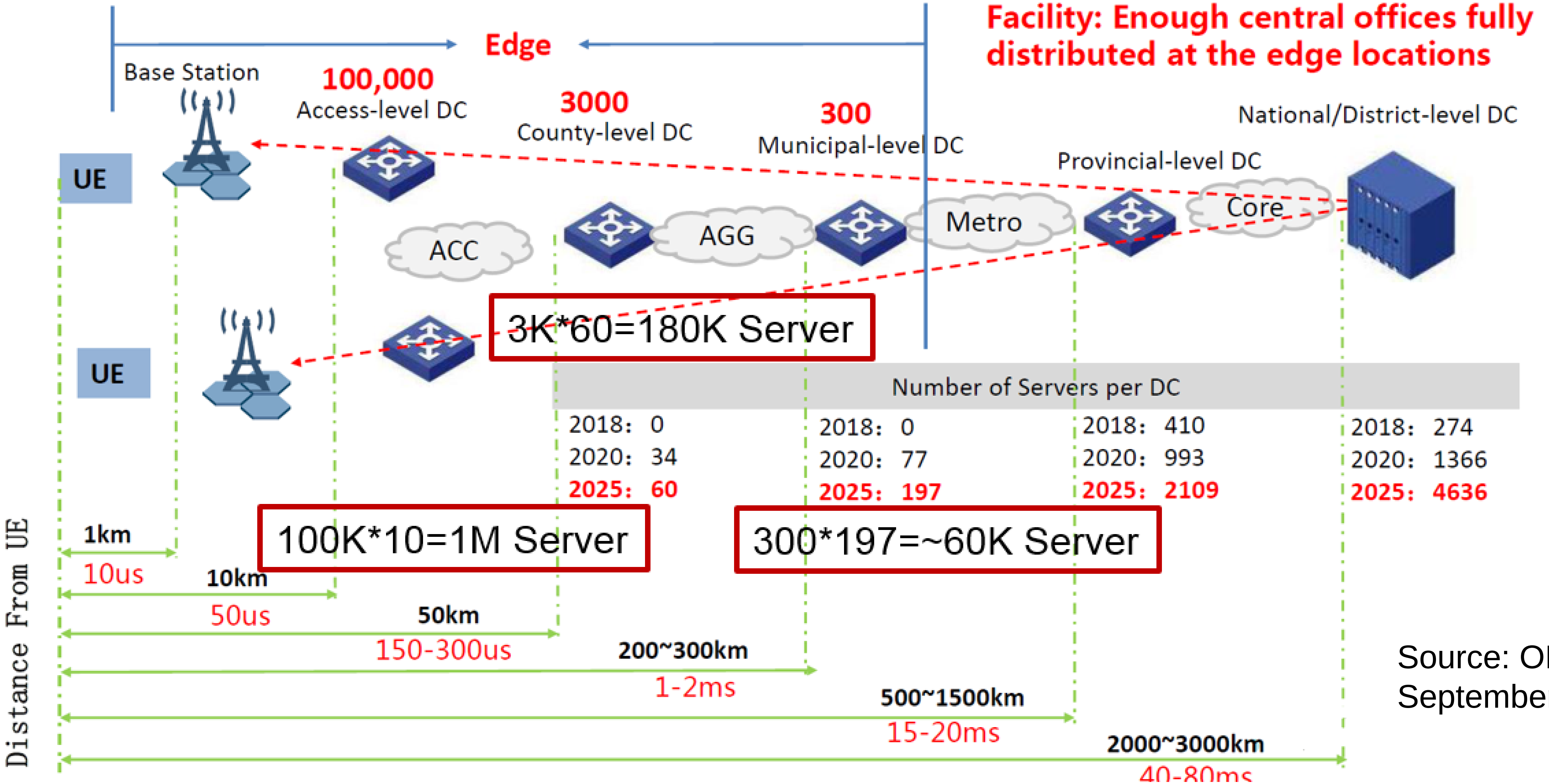
Customer premise
Programmable caching, pre-fetching, content aggregation; Programmable platform for new applications (IoT, AR)
Service discovery

Central office
Programmable caching, Multicast distribution
Enhanced funtionality through smart analytics, prediction etc.

PoP
VF Orchestration & Coordination
Caching, multicast

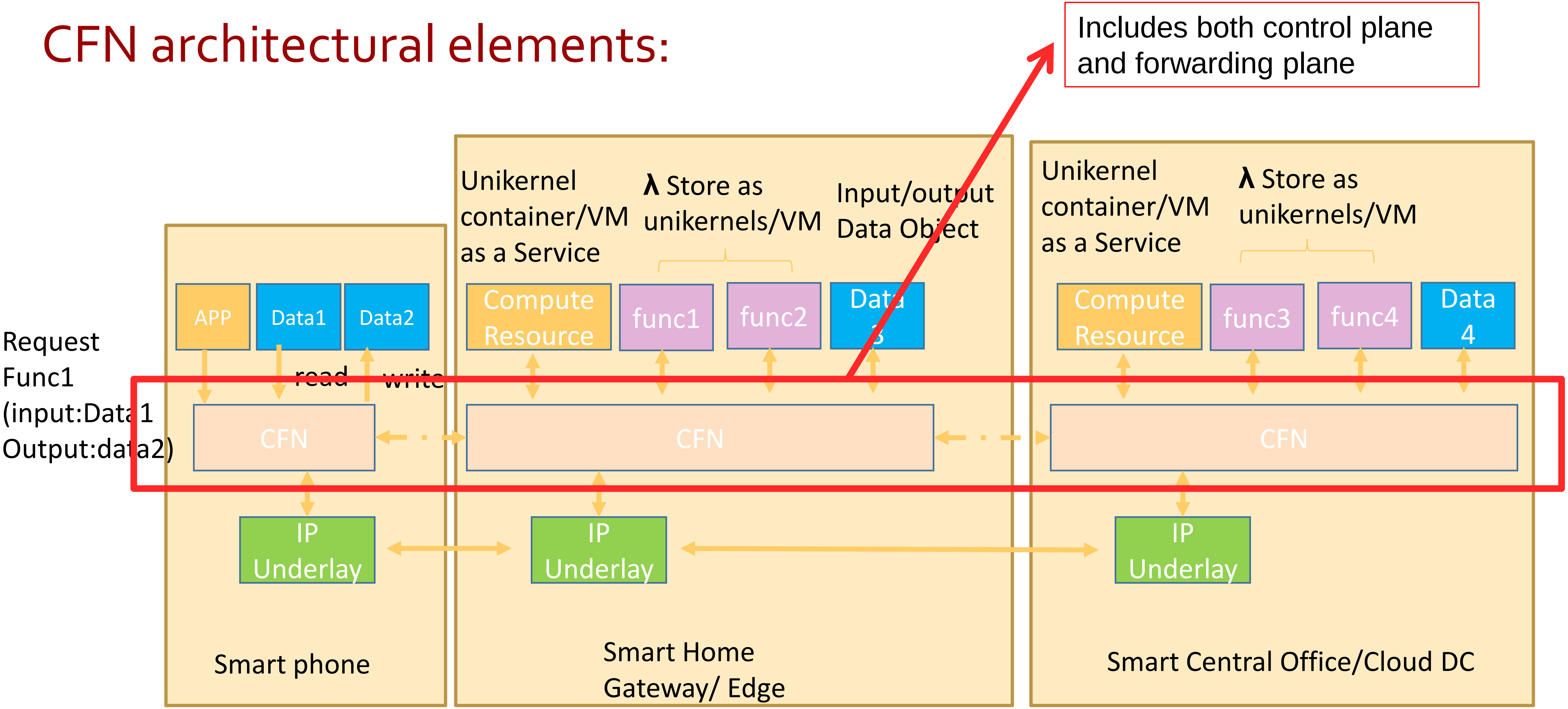
Head End
- Data-oriented provisioning
- Encoding, encryption
- Root of multicast distribution

Edge Computing location in the eye of China Mob



Source: ONS/Eu
September 201

CFN architectural elements:



CFN: an in-network function deployment layer?

- Goal of CFN is to run function in a way that maximizes the end user's QoE
  - Select the better location so that the response time is optimized
  - Taking into account: processing time, network delay, availability of the resources, availability of the function
  - Need to support a wide range of functions and deployment scenarios
- Key question: how to instantiate CFN in a practical, efficient, economical way?

What are the requirements for CFN?

- Expressivity of the service request - Ability to instantiate a wide range of services/programmability - Security (for user, for network) - Optimization of the resource allocation (optimization objective set by operator) - Support for a wide range of applications - Scale (applications/users)	- Dynamic provisioning - Agnostic to network technology and to computing platform technology - Incremental deployment/migration mechanism - Mobility support - Meets delay requirements - Distributed deployement

* Can we design a solution that meets all these requirements?
* Can we build upon a solution that partially meets these requirements and expand it to meet the missing ones?
    - if we build upon an existing solutions, do we need to add other requirements (existing customer base, open source, ease of implementation)

CFN: top down or bottom up?

Is CFN

- a *network solution* that is enhanced with the ability to deploy functions

Or

- a *service layer* that is enhanced with network awareness

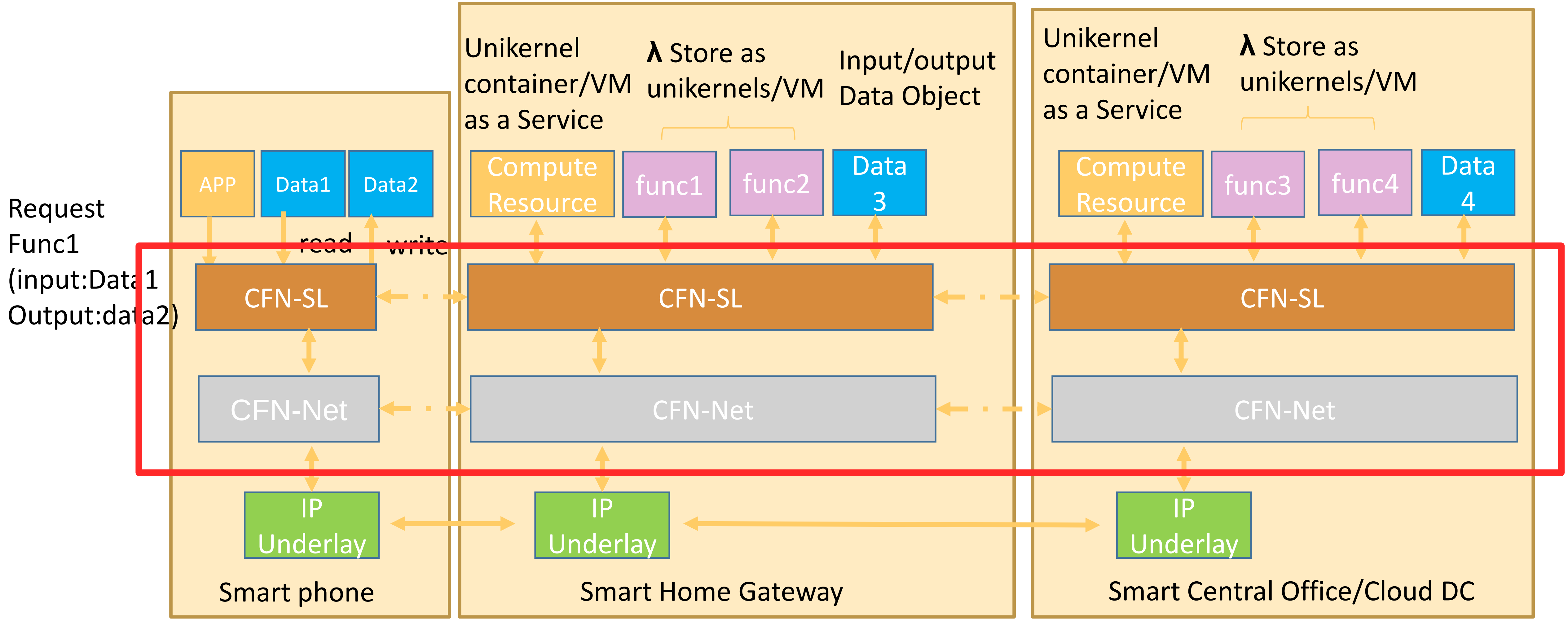
?

(or something else altogether)

Is it build by adding function execution environment to network router, or by adding network APIs to a managed function deployment environment

CFN architectural elements:

CFN-Net: monitors network delay and congestion; updates network controller; route to service instance;
CFN-SL: service layer monitors server/CPU allocations, function performance, server load



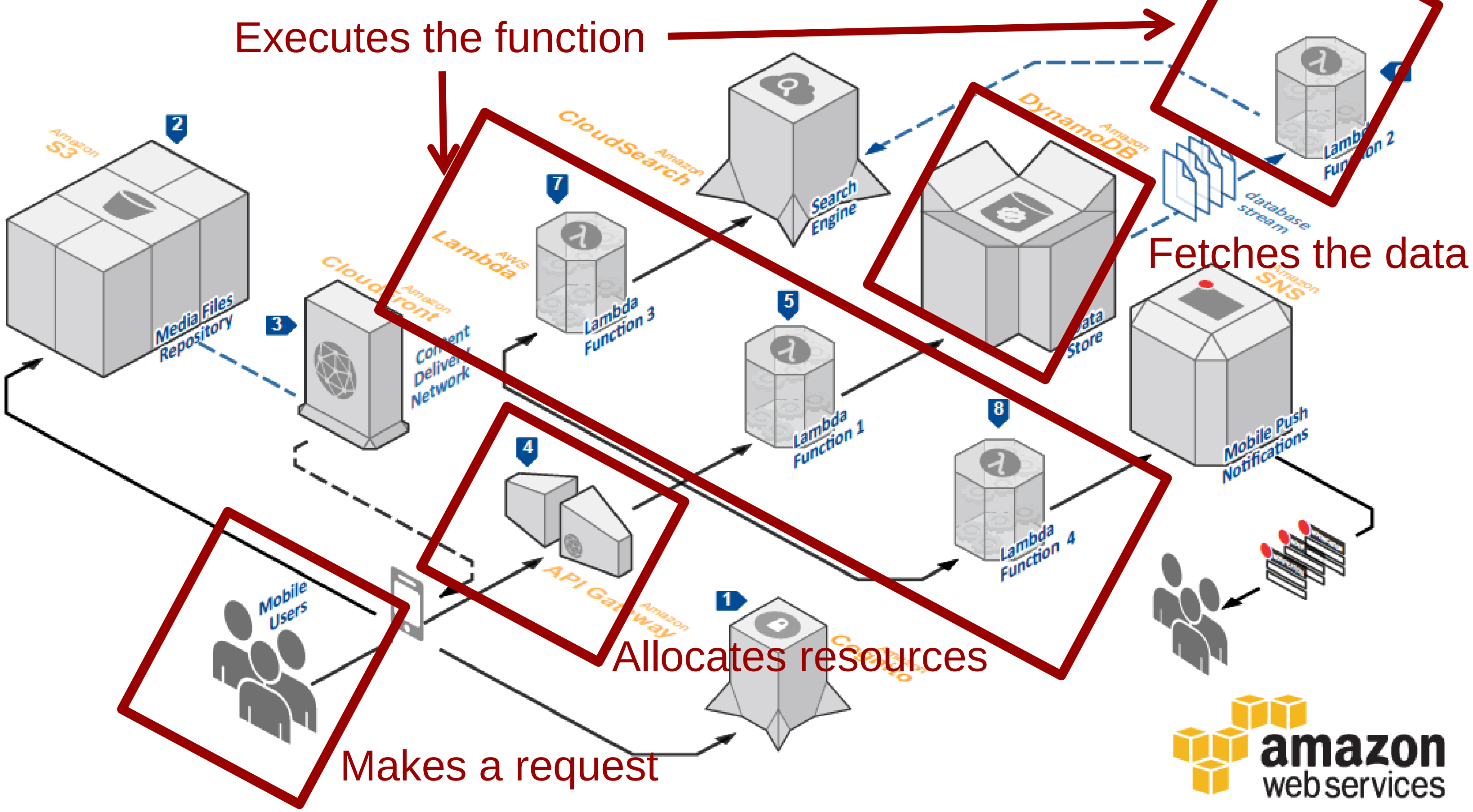
Top-down approach: Serverless architectures?

- Current widely deployed architecture to run services on demand in the cloud

Source: https://aws.amazon.com/lambda/resources/reference-architectures/

Lambda Use-cases: Mobile Backend

Is this CFN-SL?



1 Mobile users retrieve an identity from **Amazon Cognito**, which offers mobile identity management and data synchronization across mobile devices. Once a mobile user has received an identity, the user is granted access to other AWS services.

2 User-generated media files are stored in **Amazon Simple Storage Service** (Amazon S3), a highly available and durable storage service.

3 Mobile users can access their uploaded digital assets stored in **Amazon S3** through **Amazon CloudFront**, a low latency, content delivery network.

4 Mobile users send requests to **Amazon API Gateway** to access application logic and dynamic data. API Gateway acts as an entry point for mobile applications to access functionality from code running on AWS Lambda.

5 Mobile applications require a highly scalable backend infrastructure to support the variable usage created by mobile users. **AWS Lambda** runs code in response to requests and automatically manages and scales the underlying resources. Lambda Function 1 provides a synchronous endpoint for users to store and retrieve unstructured data from Amazon DynamoDB.

6 Lambda Function 2 uses **Amazon DynamoDB Streams** to retrieve changes made by users, creates a searchable document, and inserts it into **Amazon CloudSearch**.

7 Lambda Function 3 provides a synchronous interface for users to search for data from **CloudSearch**. CloudSearch manages and scales the search solution for the mobile backend.

8 Lambda Function 4 provides an asynchronous endpoint for mobile users to communicate with each other within a mobile application. The function formats each communicaton request and sends a push notification to specific users with **Amazon SNS**.

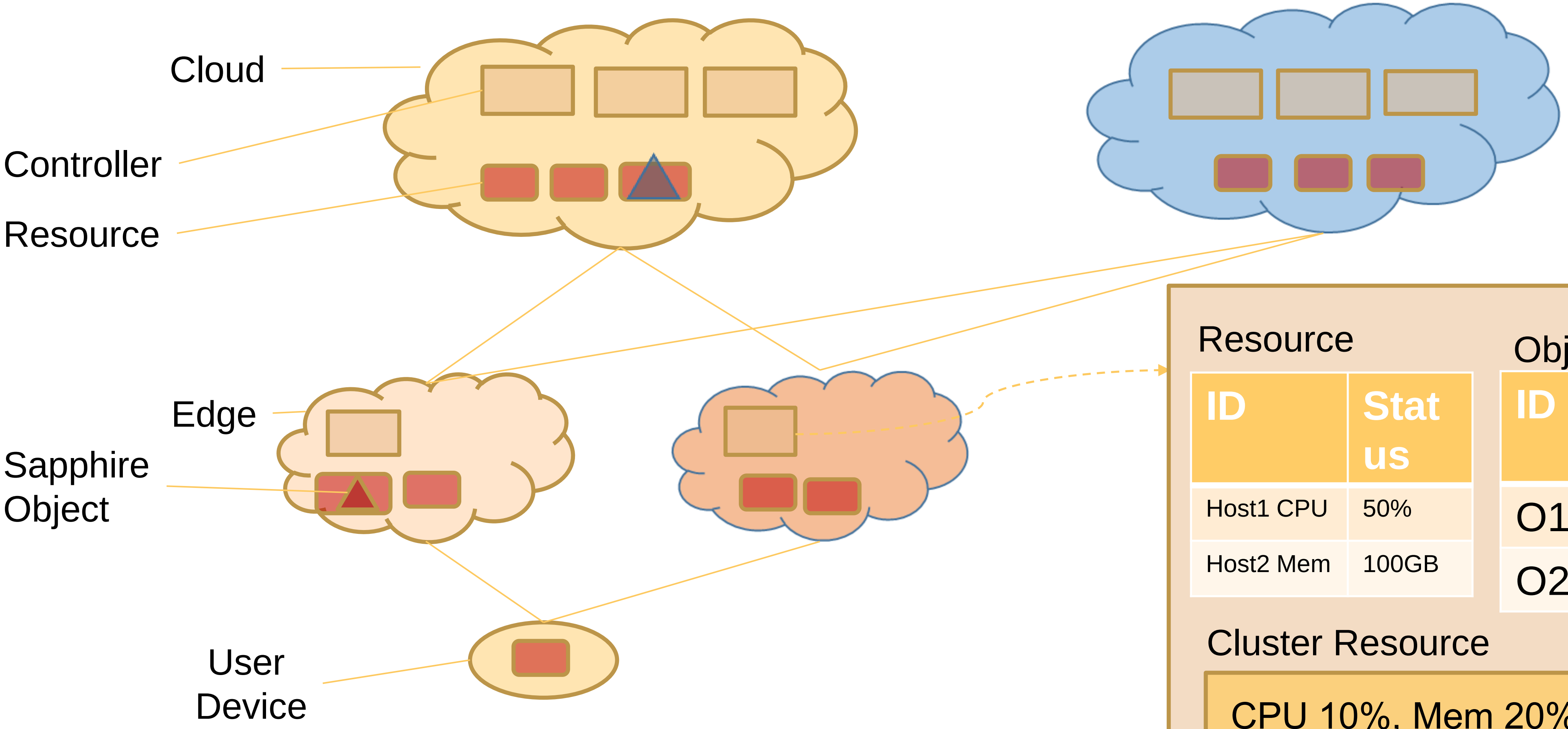
Bottom-Up Approach: Named-Function Networking

- Leverages the abstraction of Information-Centric Networking, where the networking layer is able to route content by name
- Named Data Networking exchanges data using the data name as the network handle
- NFN generalizes this to include function names as well, so that a network request carries multiple names:
  - compute(/name/of/fct, /name/of/arg)
- Up to the *network* to forward request (for *fct* and *arg*) to the node that will execute the function

Our preliminary attempt at implementing some form of CFN

- **Built on top of Sapphire, a distributed programming platform built for mobile/cloud applications**
- **Focuses on application developers need to specify application logic and deployment logic in separate but structured way**
- **Deployment scenarios range from replication and client-side caching to dynamic code offloading**
- **Any node in the cloud or edge, along with user devices can provide execution environment – provided they run sapphire components**
- **Research prototype level code publically available**
- **Implementation by Aziz Albalawi (UCSC) and Asit Chakraborti**

CFN controller architecture



Resource	
ID	**Status**
Host1 CPU	50%
Host2 Mem	100GB

Object	
ID	**Status**
O1	IP1
O2	IP2

Cluster Resource

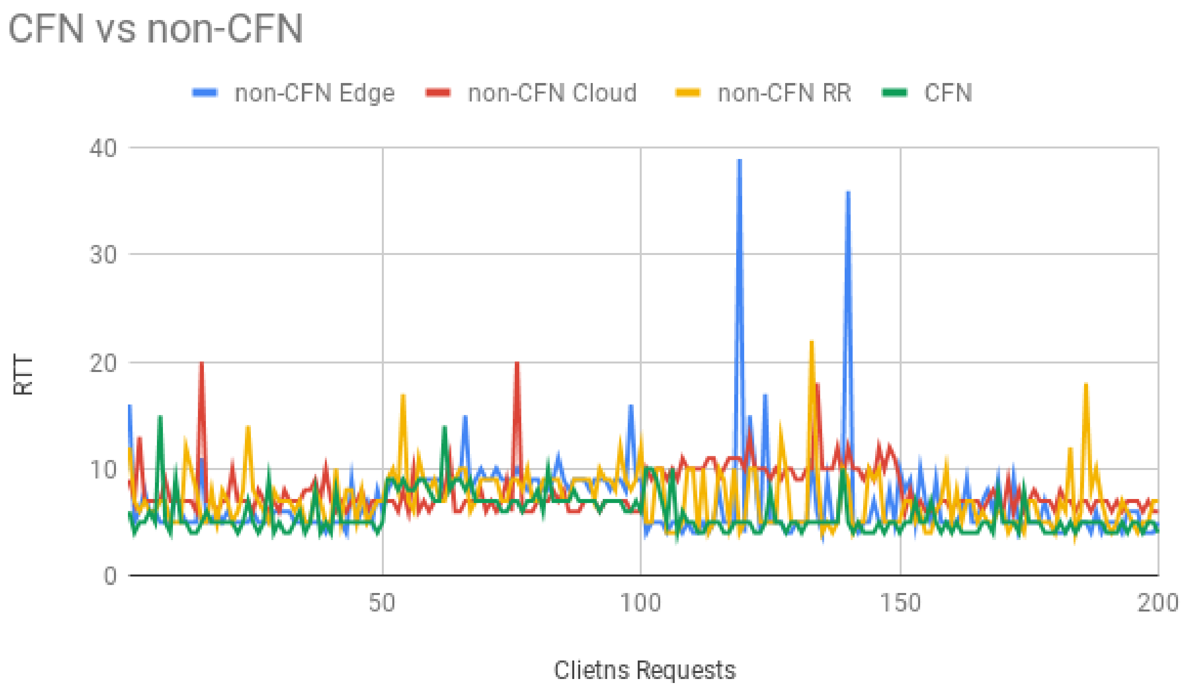
CPU 10%, Mem 20%, …

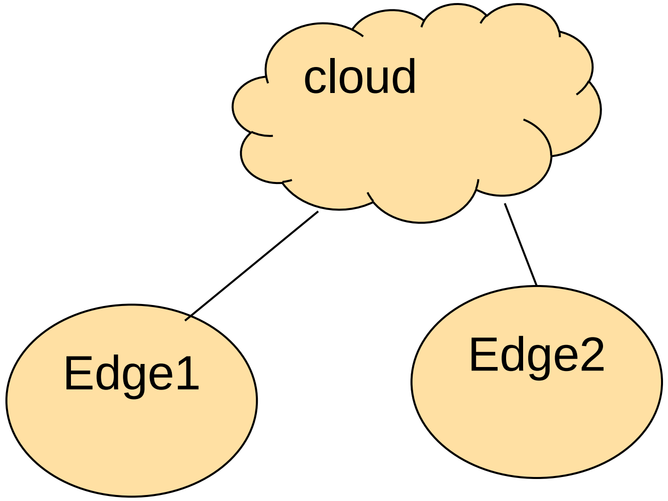
Network performance

BW utilization, RTT

CFN controller design choices…

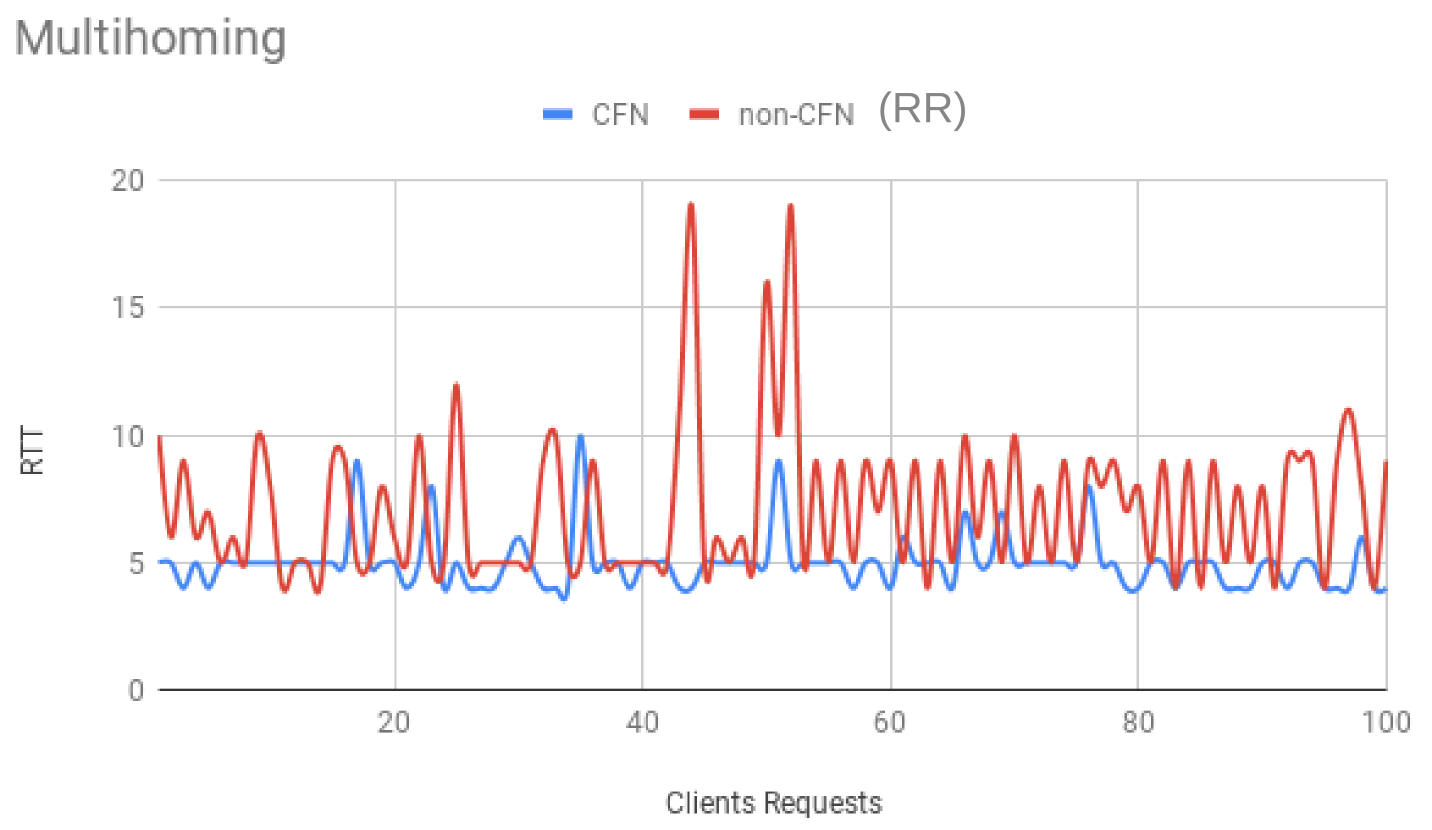
- How to allocate requests based upon service requirements and availability.
- For instance, for a latency sensitive application CFN can place the workload based upon network latency
- CFN: latency-aware, vs other workload placement policy:

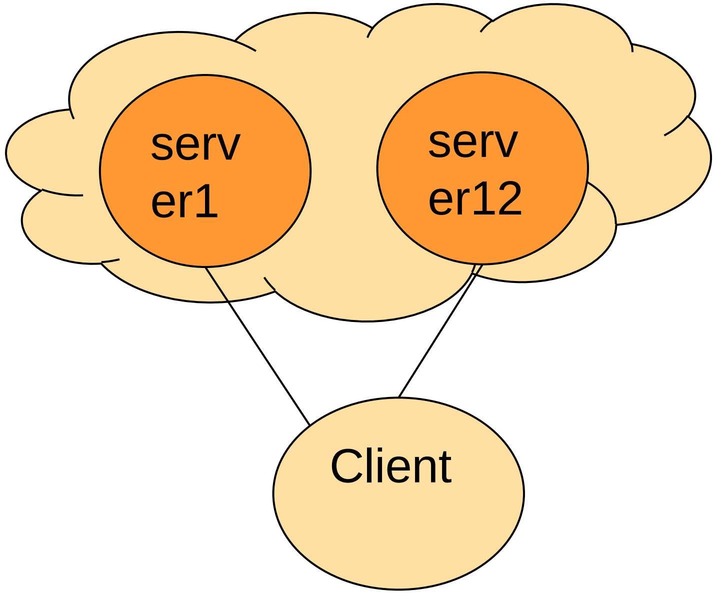




CFN controller design choices…

- Multihoming scenario:
- CFN can decide which server to allocate based upon performance





HUAWEI

Questions to consider

- How to deploy services at the edge (heterogeneous devices within different administrative domains): is there a simple and robust way to deploy applications?
- What compatibility with the current network layer?
- How to abstract the network so that application developers do not worry about it?
- Security models, distributed trust, delegation of security?
- Mobility – of user and of function?
- What are the next steps: towards standardization?

Agenda:

Thursday:
9am-10am: Breakfast, welcome, intro, motivation– Cedric Westphal (Huawei) & Dirk Kutscher (Emden University)
10-am-12pm: Use-case session – What are the business models? What are the applications for CFN? Marie-Jose Montpetit - COIN Ignacio Solis (LinkedIn) – Stream processing Alexey Tumanov (UC, Berkeley) - Ray software project and prediction serving work and latency constraints Khachik Sahakyan (Grovf) – Accelerated computing using FPGA on premise, at edge or in the cloud
12pm-1:30pm: Lunch
1:30pm-3:30pm: Gaps - What is missing today? Where are we now? Quinton Hoole (Huawei) – Amino open source project Zack Butcher (Tetrate/CNCF) - Overview of Envoy and application to operators Igor Tarasenko (Bayware) – Network Microservices
3:30pm-4pm: Coffee Break
4pm-5pm: Discussion Dave Oran, moderator
5pm-6pm: Reception

Agenda:

Friday:
9am-11am: Technical solutions – Going forward – What is the research community looking at? Stratis Ioannidis (Northeastern University) – In-Network Caching Alex Afanasayev (Florida International University) – NDN as Compute-First Networking KK Ramakrishnan (UC, Riverside) – Scheduling in NFV - OpenNetVM Dirk Kutscher (Emden University) – Remote Method Invocation in ICN
11am-12pm: Discussion moderated by Dave Oran
12pm-1pm: Lunch

Talks at the workshop: Use case session

- **COIN and its motivations (M-J. Montpetit)**

Go to the BOF!

- **Stream Processing at LinkedIn (Ignacio Solis)**

how LinkedIn relies on stream processing, and on Kafka in particular. The numbers are staggering: 4.5B messages. Stream processing touches pretty much everything, including the end user's activities, security, monitoring, infrastructure, etc.

- **Ray from the RISE Lab (Alexey Tumanov)**

Ray: a library of tools for distributed machine learning. It is one system that makes it possible to implement functionality as libraries over a distributed eager dynamic task graph.

- **Accelerated computing using FPGA (Khachik Sahakian, Grovf)**

use cases for FPGA acceleration at the edge significantly overlap with CFN use cases: 1- autonomous driving ; 2- edge AI; 3- industrial IoT; 4- M2M communication; 5- RF/DSP; 6- in memory caching; 7- network edge firewalls; 8- High Frequency Trading; 9- real-time network monitoring/filtering; 10- real-time control systems

Session 2:

- **Amino Open Source Project (Quinton Hoole, Huawei)**

  Open source product-grade version of Sapphire, distributed programmable platform

- **Network Micro Services (Igor Tarasenko, Bayware)**

  an API to program network flows that express the intent of application services across an overlay network. This is done by micro-service packaging into IPv6 extension header. Permissions to run microprogram are granted via 100-byte certificate; a graph description supports 12 and 20-bit segment identifiers and 8-bit program counters

Session 3: Research

- **Stratis Ioannidis (Northeastern University) – In-Network Caching**

  routing and caching are jointly optimized; mathematical framework to formulate the optimization problem and can solve the optimization within (1-1/e) of the optimal in a distributed manner

- **KK Ramakrishnan (UC, Riverside) – Network scheduling of workloads**

  OpenNetVM: an open source project to implement NFV; NFVnice: an NF scheduler that combines hardware packet schedule, and software process scheduling. While this was in the context of NFV, service chaining (and scheduling of functions) would be required in CFN

- **Alex Afanasayev (Florida International University) – NDN support for CFN**

  NDN as a substrate for CFN. Application naming at the network layer can re-use the content naming semantics. The recursivity of the content recovery gives NDN an advantage.

Thank You