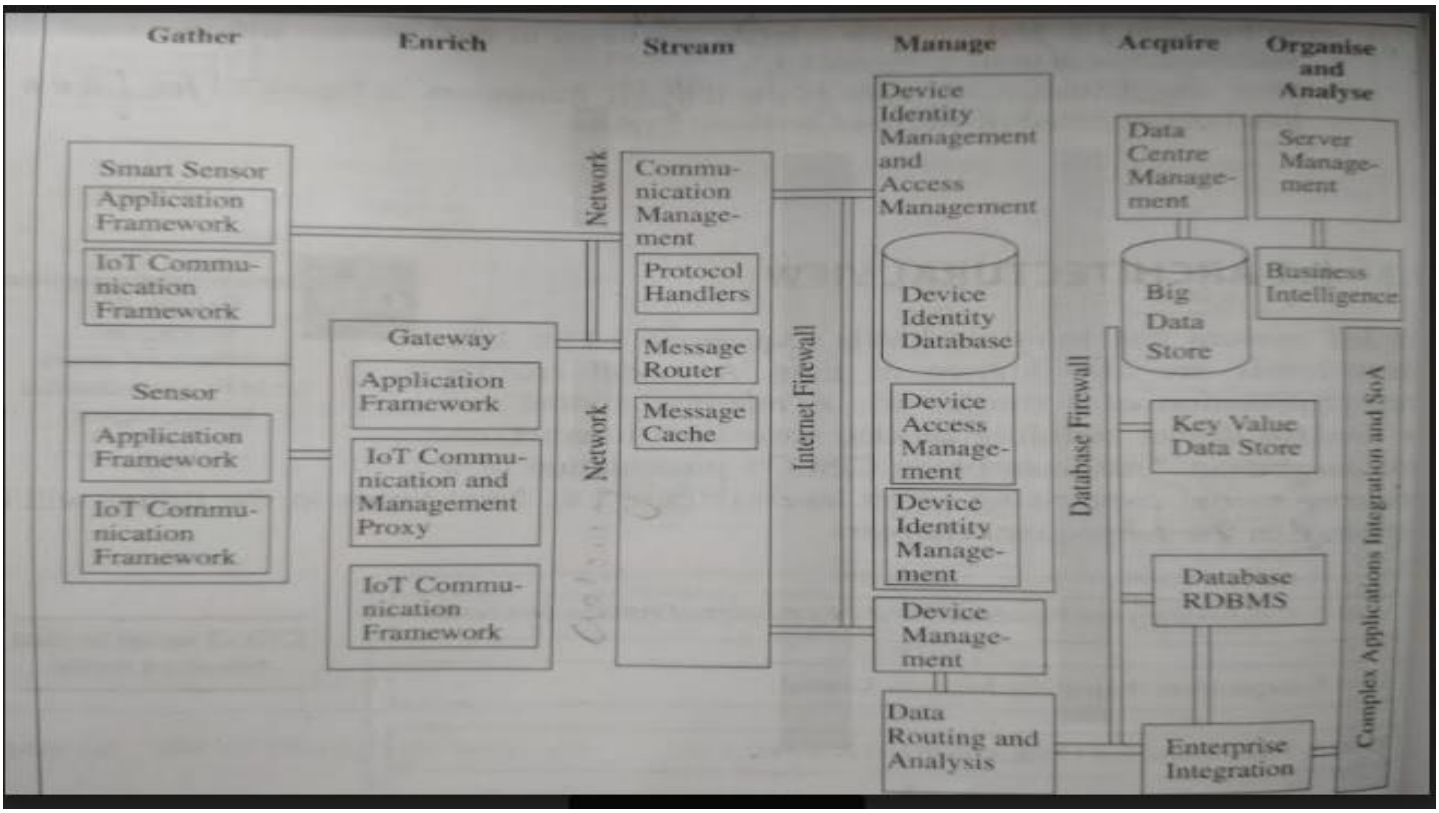
Q1 a) IOT Conceptual framework

ORACLE defines the IOT Conceptual framework by using an equation as shown below:Gather + Enrich + Stream + Manage + Acquire + Organize and Analyze

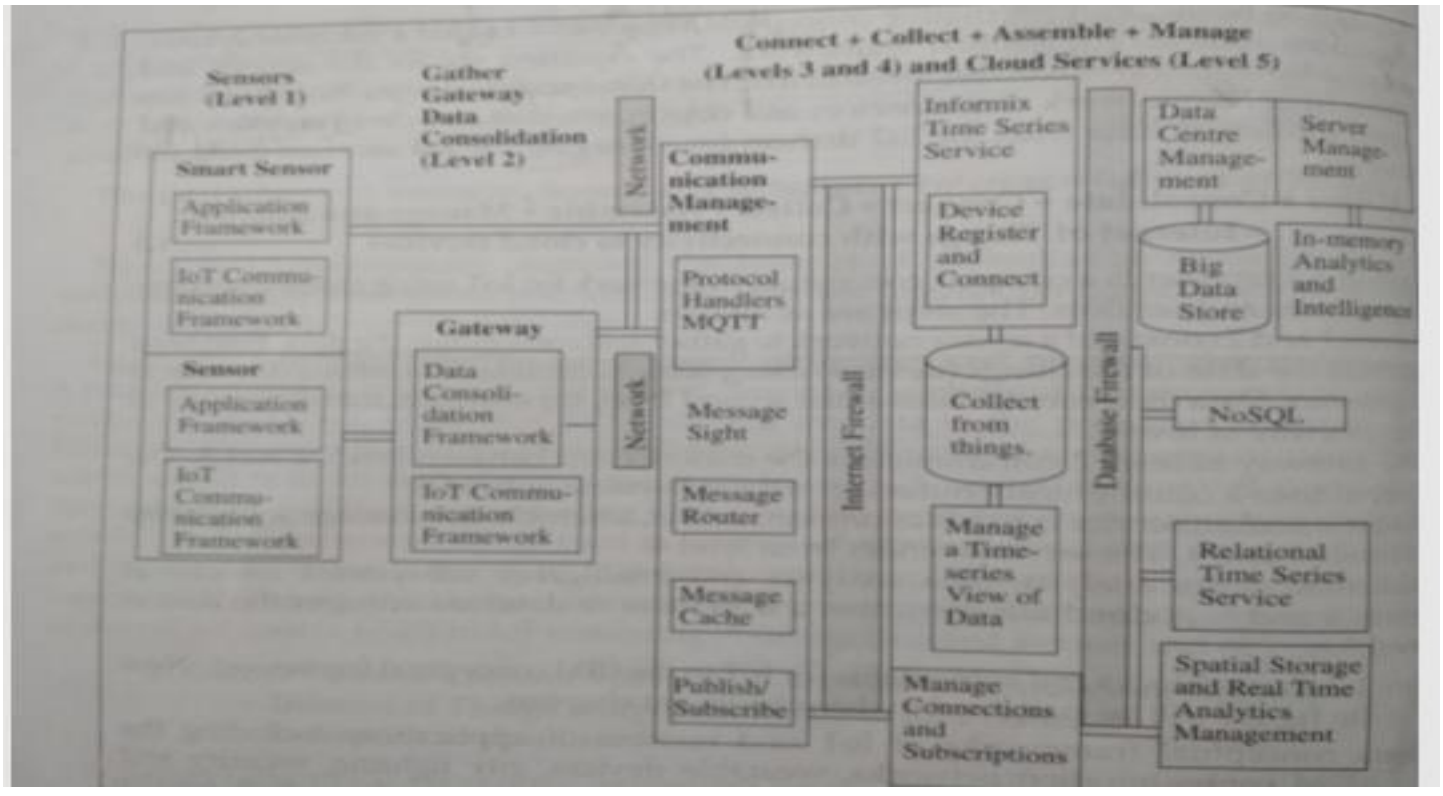
The equation represents the action and communication of the data through successive layers in IOT through interconnected devices and objects.



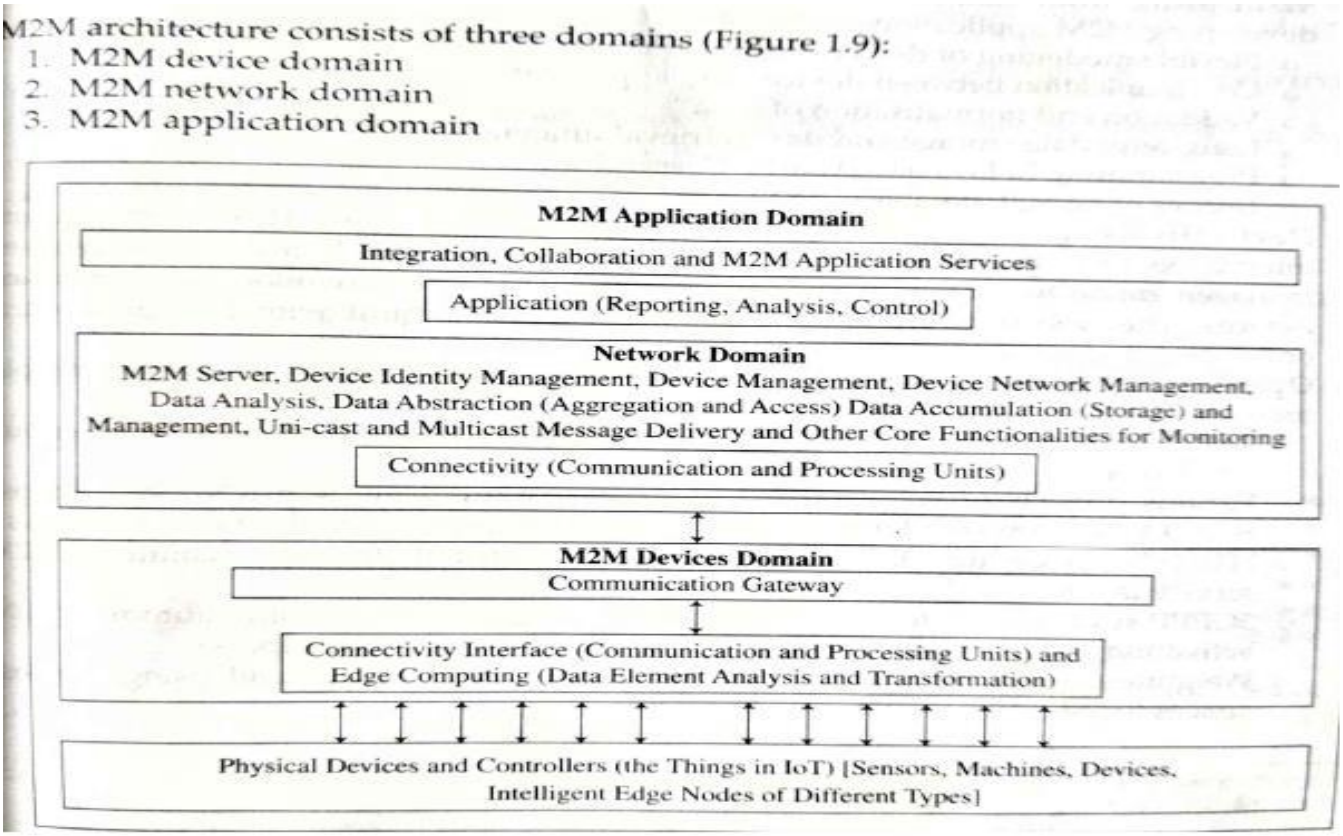
ORACLE IOT ARCHITECTURE OR FRAME WORK

IBM CONCEPTUAL FRAMEWORK

Gather + Consolidate + Connect+ Collect + Assemble + Manage and analyse= IOT



M2M archietecture:



M2M device domain

It consists of three subparts

a)Physical devices and controllers

b)Communication interface

c)Gateway (BS)

a)Physical devices and controllers:

They are **sensors, physical devices, controllers, machines which are capable of transmitting data autonomously.**

Two types of devices –ones capable of directly connecting to the network and the others requires an M2M gateway in order to connect to the network

Generally, devices can connect directly to an M2M devices via embedded SIM, TPM and radio stack or fixed line access but some connect through gateway.

a)Communication interface:

It is the port or processing unit that receives data from one interface and transmit it to other interface.

c)Gateway

Gateways and routers are the endpoints of the operator's network in scenarios where sensors and M2M devices

do not connect directly to the network

2) **M2M Network Domain (Communication Networks)**

It consists of ***M2M core and M2M service*** *capabilities.*

M2M core covers the communications between the M2M Gateway(s) and M2M application(s), e.g. LTE, WiMAX, and WLAN.

M2M service capabilities include network functions to support M2M applications. It also deals with management functions like device identity management, data storage, data collection, analysis, aggregation etc.

3)M2M application domain

Two types of applications -**M2M applications and client applications**

M2M applications: These applications are located on the servers, interacts with M2M devices.

Client applications: These used to serve end-users; either receive services from M2M applications or directly from **M2M devices**.

Here the received data is further analyzed, reports are generate and based on the analysis further actions are taken for controlling the network.

Q1 c) MQTT protocol forM2M/IoT connectivity

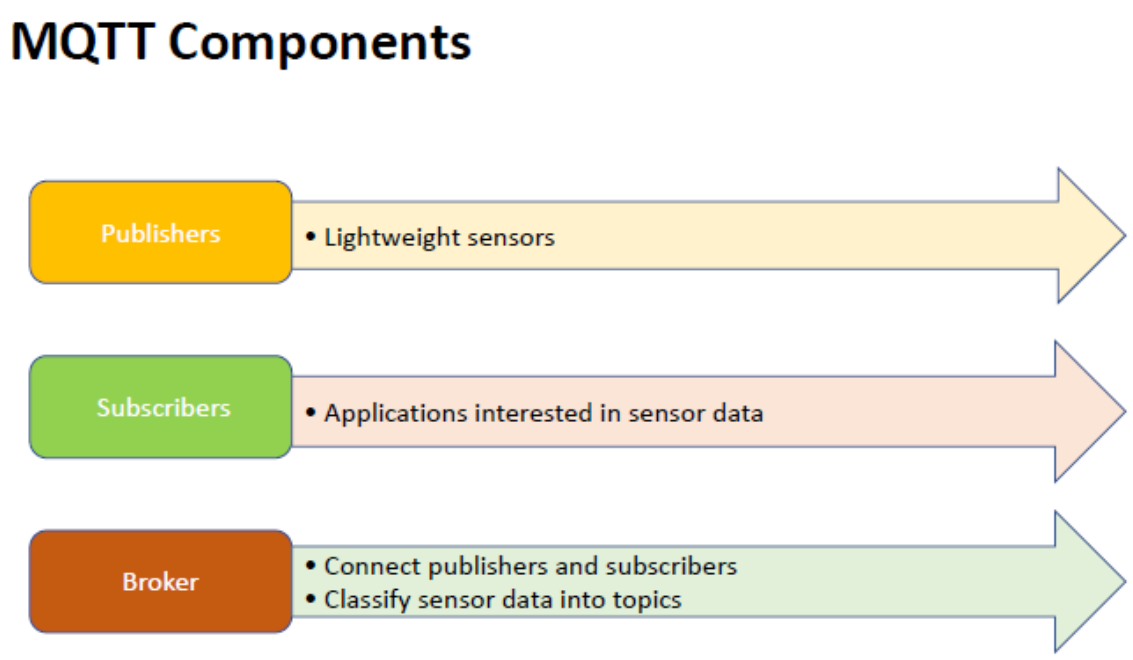
MQTT (Message Queuing Telemetry Transport)

•An **open source protocol for machine-to-machine (M2M**)/"Internet of Things" connectivity

•(Telemetry dictionary meaning is measuring and sending values or messages to far off places by radio or other mechanism)

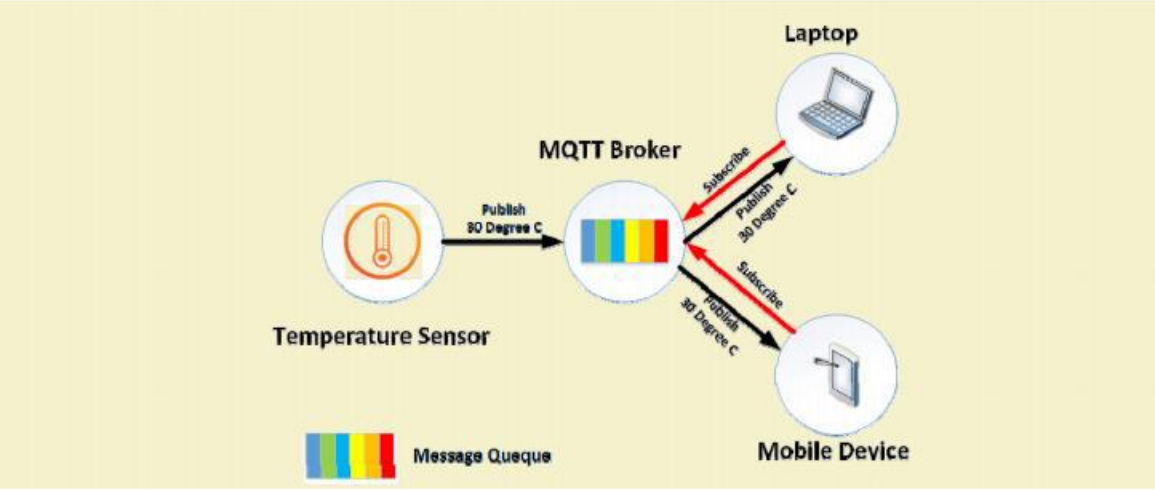
•Created by IBM IN 1999, as a constrained environment protocol.

•Designed to **provide connectivity** (mostly embedded) **between applications** and middle-wares **(M2M/IOT objects)** on one side **and networks and communications (WEB Objects)**on the other side.



(Pub/Sub) Publish/Subscribe method is used for communication in place of request-response client-server architecture

Communication



1) Publisher:

•They are the light weight sensors also called as clients

•These clients first make connections to the Broker and then publish a message to the broker.

•The message include the topic. The topic is the routing information for the broker.

2) Broker:

•Perform store and forward operation

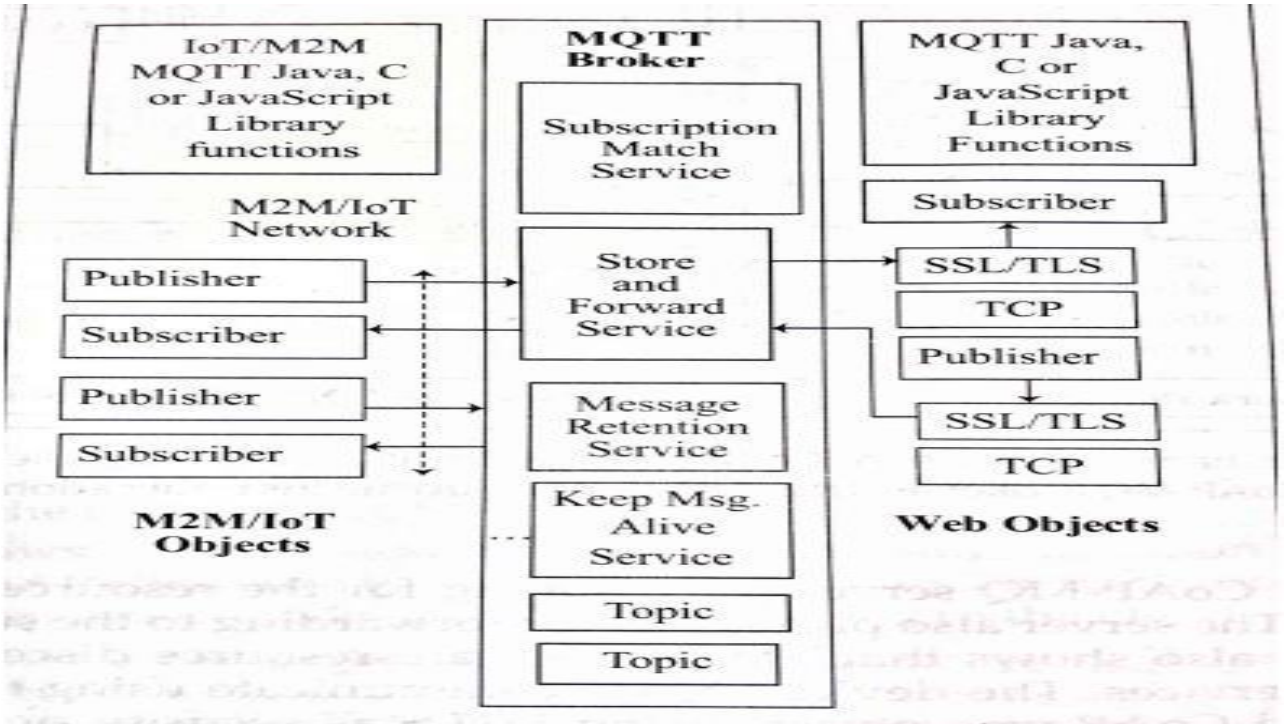
•Receives the topics from publishers

•Each client that wants to receive messages first subscribes to a certain topic and then the broker delivers all messages with the matching topic to the client.

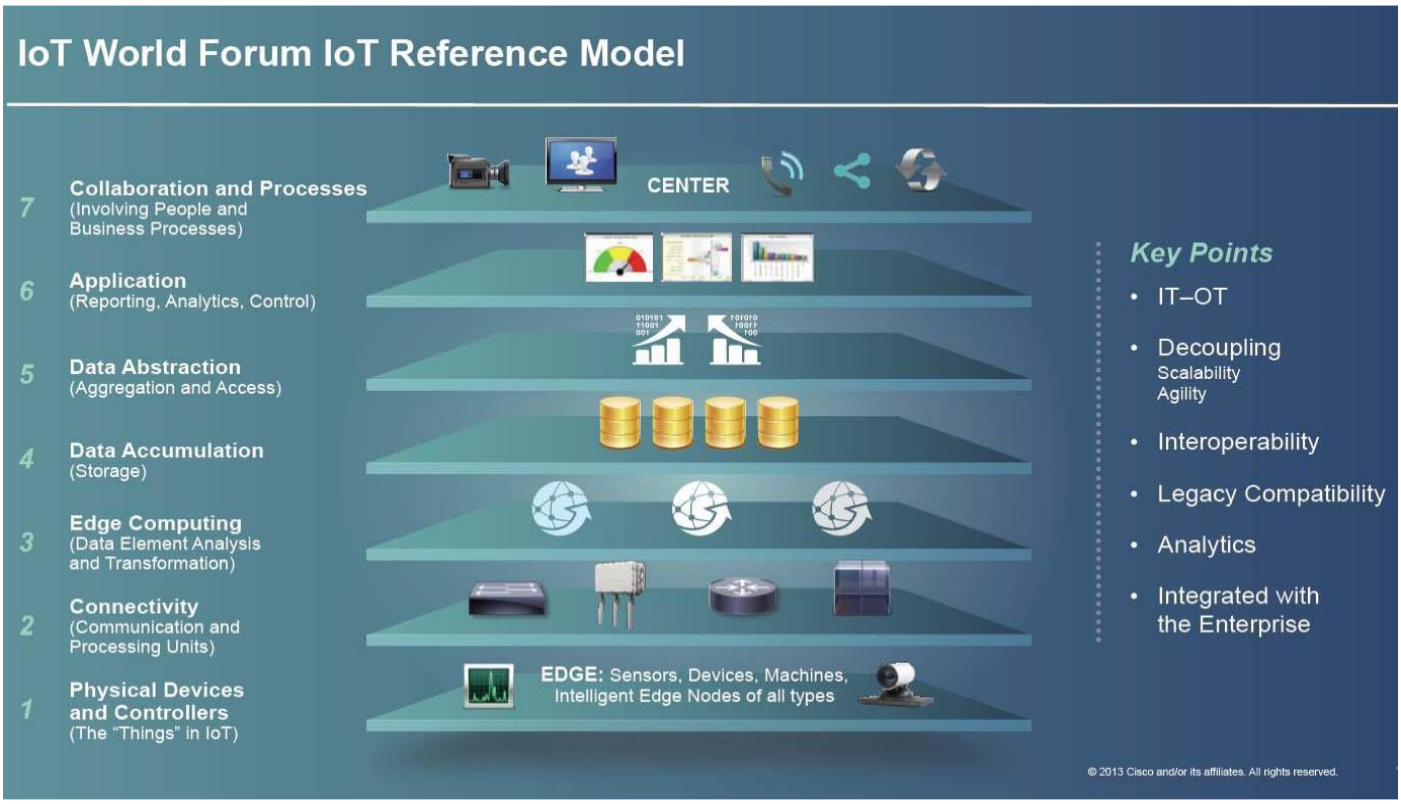
Therefore the clients don't have to know each other. They only communicate over the topic.

3) Subscribers:

•They are the clients that require the information from publishers



Q2 a) IOT Architectural view and IOT reference model suggested by CISCO



IEEE suggested P2413standard for Architecture of IoT

•A reference architecture of IoT

•The IOT reference model has 7 levels called “LAYERS” OR “TIERS”.

•Each level is defined with some terminology.

•Each level perform some specific function.

•The model describes how the task at each layer should be handled to maintain simplicity and scalability.

•IN IOT the data flows in both directions i.e.

From top to bottom (LAYER 7 to LAYER 1) –control pattern.

From bottom to top (LAYER 1 to LAYER 7) –monitoring pattern

But Basically follows top-down approach (means consider top layer design first and then move to the lowest).

Architecture of IoT

•It defines basic architectural building blocks and their integration capability into multi-tiered systems.

•The reference model defining relation-ships among various IoT verticals, for example, transportation and healthcare

•Gives a blueprint for data abstraction

•Recommends quality ‘quadruple’ trust

•“Protection, Security, Privacy, and Safety”

•Defines no new architecture and no reinvent but existing architectures congruent with it

LAYER 1. Physical devices and controllers.

•They are the **physical devices**, also called as "**THINGS**" in IOT .

•Basically they are **Embedded Devices**, Embedded hardware/software like Sensors/Actuators , RFID, Hardware (Arduino, Raspberry Pi, Intel Edison, Beagle Bone Black and Wireless SoC…).

•They are **ready to send and receive the information**.

•Devices are **unlimited, diverse and no rules about the size**, location etc…. For example..

•Devices are capable of **Analog to digital conversion** and vice versa.

•Devices are capable of **generating data and being queried**.

LAYER 2. Connectivity (Communication and processing units)

Processing Units:

Contains **Routers and Gateways**

Main task is to deliver the right information at right time and to right machine i.e. reliable transmission.

Communication:

Includes **protocol handlers, message routers, message cache**

It can be between smart device and network/ internet directly

It can be through gateways then to network

Therefore main task involves switching and routing, enriching, transcoding, translation between protocols, security and self learning etc.

Communication occurs networks –EAST-WEST communication

Layer 3 [Edge Computing Or Fog Computing]

Edge computing is an architecture that uses**edge devices** / **network edge** like **routers, gateways, switches**, multiplexers, integrated access devices to do some **preprocessing of data**.

Preprocessingincludes data aggregation, storage, data filtering, cleanup, analysis, transformation (formatting, decoding, distillation) , Threshold(alert), event generation etc.

Finally the data is routed to web servers/cloud.

Layer 4 [Data Accumulation and storage]

Data management is done at backend server/cloud or data base centres

Main roles of layer 4 are:

Convert data in motion to data at rest.

Convert format from network packets to database relational tables.

Convert Event based data to query based data (it bridges the gap between real time networking and non real time)

The concept of BIG DATA is used at layer 4.

Layer 5 Data Abstraction

Data abstraction is done at backend server/cloud or data base centres

Abstraction **means providing the essential and relevant information**of the data by hiding the irrelevant one.

Main roles are:

1)Provide multiple storage systems to accommodate data from different IOT devices.

2)Reconciling multiple data formatfrom different sources.

3)Combining data from multiple sources and simplifying the application i.econsolidating the data into one place.

4)Filtering, selecting, projecting and reformatting the data to serve client application.

5)Protecting the data with appropriate authentication and authorization.

Layer 6 Application

Layer 6 deals with reporting, analysis and control

i.e. the data is analysed and then send to controlling device like actuator.

And then the data is passed to specific application like mobile application or webpage or to the business enterprise which require that data.

Layer 7 Collaboration and processes.

It means involving people and business process.

Basically multiple people are using same applications for a range of different purpose

But in IOT the main objective is to empower people to do their work better, not the application.

UNIT-1

1	a	An IoT system has multiple levels. These levels are also known as tiers. Explain in detail about an IoT reference model suggested by CISCO that gives a conceptual framework for a general IoT system.	10
	b	Explain Various functional units in an MCU that are embedded in an IoT device or a physical object.	10
		OR	
2	a	Machine-to-machine refers to the process of communication of a physical object or device at machine with others of the same type, mostly for monitoring but also for control purposes. Explain the M2M Architecture in detail.	10
	b	OSI protocols mean a family of information exchange standards developed jointly by the ISO and the ITU-T. Explain in detail about modified OSI model for the IoT/M2M Systems.	10

UNIT-2

3	a	Explain in detail about Lightweight Machine-to-Machine Communication Protocol with a Suitable diagram.	10
	b	Explain the different terminologies used in Message Communication Protocols for Connected devices.	10
		OR	
4	a	Explain the following: i) MQTT Protocol ii) XMPP	10
	b	Define the following: i) Communication protocol ii) Application Programming Interface iii) Web service iv) Communication gateway v) Universal Resource Locator	10

UNIT-3

5	a	The Internet of Things (IoT) envisages new security challenges, including in the area of access control that can hardly be met by existing security solutions. Explain in detail about IoT@Work Capability Based Access Control System.	10
	b	From a security and privacy perspective, the developments in GAMBAS are centred on a secure distributed architecture in which data acquisition, data storage and data processing are tightly controlled by the user. What are the elements involved in security band privacy perspectives?	10
		OR	
6	a	Explain the following with respect to objectives and usage: i) Smart Transportation ii) Smart campus	10
	b	Explain in detail about trust and Quality-of-Information in an Open Heterogeneous Network.	10

UNIT-4

7	a	FEC provides a complement to the cloud in IoT by filling the gap between cloud and things towards providing service continuum. Explain the advantages by mentioning how FEC provides these advantages along with the diagram.	10
	b	Explain in detail about the Hierarchy of Fog and Edge Computing.	10
		OR	
8	a	Discuss about the Business models of FEC and also the opportunities and challenges involved in FEC.	10
	b	What can be done to overcome the limitation of current cloud-centric architecture? Explain with a neat diagram.	10

UNIT-5

9	a	Explain in detail about 5G Slicing Framework with a generic architectural diagram.	10
	b	Explain the following: i) Network-aware Virtual Machines Management ii) Virtual Network Functions Management	10
		OR	
10	a	With a neat diagram explain the Taxonomy of network-aware VM/VNF Management in software–defined Clouds.	10
	b	Write the short notes on: i) Mobile Edge Computing ii) Edge and Fog Computing	

Unit4

Illustrate BLUR challenges faced with Cloud-centric Internet of Things (CIoT)

Although the CIoT model is a common approach to implement IoT systems, it is facing the growing challenges in IoT. Specifically, CIoT faces challenges in BLURS—Bandwidth, Latency, Uninterrupted, Resource-constraint and Security [3] .

Bandwidth. The increasingly large and high-frequent rate data produced by objects in IoT will exceed the bandwidth availability. For example, a connected car can generate tens of megabytes' data per second for the information of its route, speeds, car operating condition, driver's condition, surrounding environment,

weather etc. Further, a self-driving vehicle can generate gigabytes of data per second due to the need for real-time video streaming. Therefore, fully relying on the distant cloud to manage the things becomes impractical.

 Latency. Cloud faces the challenges to achieve the requirement of controlling the end-to-end latency within tens of milliseconds. Specifically, industrial smart grids systems, self-driving vehicular networks, virtual and augmented reality applications, real-time financial trading applications, healthcare and eldercare applications cannot afford the causes derived from the latency of CIoT.

 Uninterrupted. The long distance between cloud and the front-end IoT devices can face issues derived from the unstable and intermittent network connectivity. For example, a CIoT-based connected vehicle will be unable to function properly due to the disconnection occurred at the intermediate node between the vehicle and the distant cloud.

 Resource-Constrained. Commonly, many front-end devices are resource-constraint in which they are unable to perform complex computational tasks and hence, CIoT systems usually require front-end devices to continuously stream their data to the cloud. However, such a design is impractical in many devices that operate with battery power because the end-to-end data transmission via the Internet can still consume a lot of energy.

 Security. A large number of constraint front-end devices may not have sufficient resources to protect themselves from the attacks. Specifically, outdoor-based front-end devices, which rely on the distant cloud to keep them updated with the security software, can be attackers' targets, in which the attackers are capable of performing a malicious activity at the edge network where the front-end devices are located and the cloud does not have full control on it. Furthermore, the attacker may also damage or control the front-end device and send false data to the cloud.

FEC offers five main advantages

Security. FEC supports additional security to IoT devices to ensure safety and trustworthiness in transactions. For example, today's wireless sensors deployed in outdoor environments often require the remote wireless source code update in order to resolve the security-related issues. However, due to various dynamic environmental factors such as unstable signal strength, interruptions, constraint bandwidth etc., the distant central backend server may face challenges to perform the update swiftly and hence, increases the chance of cyber security attack. On the other hand, if the FEC infrastructure is available, the backend can configure the best routing path among the entire network via various FEC nodes in order to rapidly perform the software security update to the wireless sensors.

Cognition. FEC enables the awareness of the objectives of their clients towards supporting autonomous decision making in terms of where and when to deploy computing, storage and control functions. Essentially, the awareness of FEC, which involves a number of mechanisms in terms of self-adaptation, self-organization, self-healing, self-expression and so forth [16] , shifts the role of IoT devices from passive to active smart devices that can continuously operate and react to customer requirements without relying on the decision from the distant cloud.

Agility. FEC enhances the agility of the large scope IoT system deployment. In contrast to the existing utility cloud service business model, which relies on the large business holder to establish, deploy and manage the fundamental infrastructure, FEC brings the opportunity to individual and small businesses to participate in providing FEC services using the common open software interfaces or open Software Development Kits (SDKs). For examples, the MEC standard of ETSI and the Indie Fog business model [18] will hasten the deployment of large scope IoT infrastructures.

Latency. The common understanding of FEC is to provide rapid responses for the applications that require ultra-low latency. Specifically, in many ubiquitous applications and industrial automation, the system needs to collect and process the sensory data continuously in the form of the data stream in order to identify any event and to perform timely actions. Explicitly, by applying FEC, these systems are capable of supporting time-sensitive functions. Moreover, the softwarization feature of FEC, in which the behavior of physical devices can be fully configured by the distant central server using software abstraction, provides a highly flexible platform for rapid re-configuration of the IoT devices.

Efficiency. FEC enhances the efficiency of CIoT in terms of improving performance and reducing the unnecessary costs. For example, by applying FEC, the ubiquitous healthcare or eldercare system can distribute a number of tasks to the Internet gateway devices of the healthcare sensors, and utilize the gateway devices to perform the sensory data analytics tasks. Ideally, since the process happens near the data source, the system can generate the result much faster. Further, since the system utilizes gateway devices to perform most of the tasks, it highly reduces the unnecessary cost of outgoing communication bandwidth.

advantages provided by FEC leads

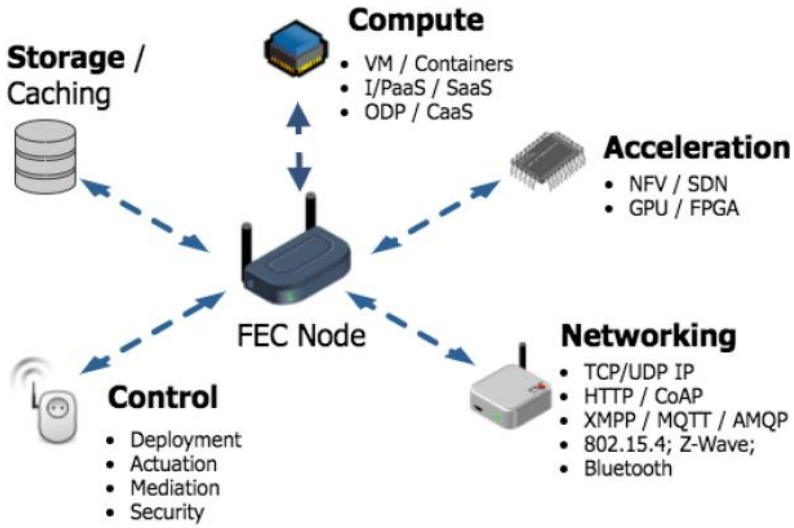


Figure 1.2. FEC nodes supports five basic mechanisms—storage, compute, acceleration, networking and control.

Storage. The mechanism of storage in FEC corresponds to the temporary data storing and caching at the FEC nodes in order to improve the performance of information or content delivery. For example, content service providers can perform multimedia content caching at the FEC nodes that are most close to their customers in order to improve the quality of experience [19] . Further, in connected vehicle scenarios, the connected vehicles can utilize the roadside FEC nodes to fetch and to share the information collected by the vehicles continuously.

Compute. FEC nodes provide the computing mechanisms mainly in two models—Infrastructure or Platform as a Service (I/PaaS) and Software as a Service (SaaS). In general, FEC providers offer I/PaaS based on two approaches—hypervisor Virtual Machines (VMs) or Containers Engines (CEs), which enable flexible platforms for FEC clients to deploy the customized software they need in a sandbox environment hosted in FEC nodes. Besides the I/PaaS, the SaaS is also promising in FEC service provision [3] . To enumerate, SaaS providers can offer two types of services—On-demand Data Processing (ODP) and Context as a Service (CaaS). Specifically, an ODP-based service has pre-installed methods that can process the data sent from the client in the request/response manner. Whereas, the CaaS-based service provides a customized data provision method in which the FEC nodes can collect and process the data to generate meaningful information for their clients.

Acceleration. FEC provides acceleration with a key concept

“programmable”. Fundamentally, FEC nodes support acceleration in two aspects

networking acceleration and computing acceleration.

 Networking acceleration. Initially, most network operators have their own configuration in message routing paths in which their clients are unable to request for the customized routing tables. For example, an Internet Service Provider (ISP) in East Europe may have two routing paths with different latency to reach a Web server located in Central Europe, and the path a client will be on is based on the ISP's load balancing setting, which in many cases, is not the optimal option for the client. On the other hand, FEC supports network acceleration mechanism based on network virtualization technology, which enables FEC nodes to operate multiple routing tables in parallel and to realize Software Defined Network (SDN). Therefore, the clients of the FEC nodes can configure customized routing path for their applications in order to achieve optimal network transmission speed.

 Computing acceleration. Researchers in fog computing have envisioned that the FEC nodes will provide computing acceleration by utilizing advanced embedded processing units such as Graphics Processing Units (GPUs) or Field Programmable Gate Arrays (FPGA) units [8] . Specifically, utilizing GPUs to enhance the process of complex algorithms has become a common approach in general cloud computing field. Therefore, it is foreseeable that FEC providers may also provide the equipment that contains middle- or high-performance independent GPUs. Further, FPGA units allow users to re-deploy program codes on them in order to improve or update the functions of the host devices. Particularly, researchers in sensor technologies [20] have been utilizing FPGA for runtime reconfiguration of sensors for quite some time. Further, in comparison with GPUs, FPGA is potential to be a more energy efficient approach for the need of acceleration based on allowing clients to configure their customized code on the FEC nodes.

Networking. Networking of FEC involves vertical and horizontal connectivities. Vertical networking interconnects things and cloud with the IP networks; whereas, horizontal networking can be heterogeneous in network signals and protocols depending on the supported hardware specification of the FEC nodes.

 Vertical networking. FEC nodes enable vertical network using IP network-based standard protocols such as the request/response-based TCP/UDP sockets, HTTP, Internet Engineering Task Force (IETF)---Constraint Application Protocol (CoAP) or publish-subscribe-based Extensible Messaging and Presence Protocol (XMPP), OASIS---Advanced Message Queuing Protocol (AMQP; ISO/IEC 19464), Message Queue Telemetry Transport (MQTT; ISO/IEC PRF 20922) and so forth. Specifically, the IoT devices can operate server-side function (e.g. CoAP server) that allows FEC nodes, which act as the proxy of cloud, to collect data from them and then forward the data to the cloud. Further, FEC nodes can also operate as the message broker of publish-subscribe-based protocol that allows the IoT devices to publish data stream to the FEC nodes and enable the cloud backend to subscribe the data stream from the FEC nodes.

 Horizontal networking. Based on various optimization requirements such as energy efficiency or the network transmission efficiency, IoT systems are often using heterogeneous cost-efficient networking approaches. In particular, smart home, smart factories, connected vehicles are commonly utilizing Bluetooth, ZigBee (based on IEEE 802.15.4), Z-Wave on the IoT devices and connect them to an IP network gateway towards enabling the connectivity between the devices and the backend cloud. In general, the IP network gateway devices are the ideal entities to host FEC servers since they have the connectivity with the IoT devices in various signals. For example, the cloud can request an FEC server hosted in a connected car to communicate with the roadside IoT equipment using ZigBee in order to collect the environmental information needed for analyzing the real-time traffic situation.

Control. The control mechanism supported by FEC consists of four basic types---deployment, actuation, mediation, and security.

 Deployment control allows clients to perform customizable software program deployment dynamically. Further, clients can configure FEC nodes to control which program the FEC node should execute and when it should execute it? Further, FEC providers can also provide a complete FEC network topology as a service that allows clients to move their program from one FEC node to another. Moreover, the clients may also control multiple FEC nodes to achieve the optimal performance for their applications.

 Actuation control represents the mechanism supported by the hardware specification and the connectivities between the FEC nodes and the connected devices. Specifically, instead of performing direct interaction between the cloud and the devices, the cloud can delegate certain decisions to FEC nodes to directly control the behavior of IoT devices.

 Mediation control corresponds to the capability of FEC in terms of interacting with external entities owned by different parties. In particular, the connected vehicles supported by different service providers can communicate with one another though they may not have a common protocol initially, with the softwarization feature of FEC node, the vehicles can have on-demand software update towards enhancing their interoperability.

 Security control is the basic requirement of FEC nodes that allows clients to control the authentication, authorization, identify and protection of the virtualized runtime environment operated on the FEC nodes.

1.5 Business Models

While the common discussions of FEC are focusing on the advantages and applications, a fundamental question regards to how the business models of FEC will be like, has usually not been elaborated. Thereupon, here we discuss the three basic business models derived from the recent works [3] [10] [18] .

1.5.1 X as a Service

Here, the 'X' of the X as a Service (XaaS) corresponds to infrastructure, platform, software, networking, cache or storage and many other types of resources mentioned in general cloud services. Specifically, XaaS providers of FEC allow their clients to pay to use the hardware equipment that supports SCANC mechanisms described in the previous section. Further, XaaS model does not limit to major business providers such as ISPs or the large cloud providers. Ideally, individuals and small businesses can also provide XaaS in the form of IndieFog [18] that is based on the popular Consumer as Provider (CaP) service provisioning model in multiple domains. For example, the MQL5 Cloud Network distributed computing project (cloud.mql5.com) utilizes Customer-Premises Equipments (CPEs) to perform various distributed computing tasks. Further, Fon (fon.com) utilizes CPEs to establish a global Wi-Fi network. These examples indicate that many individuals are willing to let application service providers pay to use their equipment for offering services.

1.5.2 Support Service

The support service of FEC is similar to the software management support services in general information systems in which the clients who own the hardware equipment can pay the support service provider to provide them the corresponding software installation, configuration, and updates on the clients' equipment based on the requirements of the clients. Further, the clients may also pay for monthly or annual support services to the provider for assisting them with the maintenance and technical support. In general, support service providers offer their clients the highly customized solutions to achieve the optimal operation of their FEC-integrated systems. In general, a typical example of the support service provider is how Cisco provides the fog computing solution, in which the clients purchase Cisco's IOX-enabled equipment then pay the

additional service fee to gain access to the software update and technical support for configuring their FEC environments. It is foreseeable that in near future, such a model will not be constrained to the single provider's hardware and software in which the support service provider will be decoupled from the hardware equipment vendors just like today's enterprise information systems support service providers such as RedHat, IBM or Microsoft.

1.5.3 Application Service

Application service providers provide application solutions to help their clients in processing the data within or outside of the client's operation environments. For example, the recent Digital Twinning technologies create real-time virtualized 'twin' that clone the real-world behavior of a broad range of physical entities, from industrial facilities, equipment to the entire factory plane and the involved production lane and supply chains. Explicitly, such technology can provide the insight of the efficiency and performance towards optimizing and improving the industrial activities. Accordingly, an FEC application service provider can provide the Digital Twinning solution configured across all the involved entities at the edge networks in order to provide the analysis in an ultra-low latency manner (less than tens of milliseconds) towards helping the industrial system with the rapid reactions. Similarly, the FEC application service providers can also provide the service to local government in real-time traffic control system that assists the self-driving, connected vehicles. Further, IndieFog providers can also provide various application services to assist the Ambient Assisted Living (AAL) service providers in providing a certain edge analytics mechanisms for the clients of the AAL service providers. For example, an IndieFog provider who has installed Apache Edgent can offer the built-in stream data classification function as an application service for the mobile AAL clients in the close proximity.

1.6 Opportunities and Challenges

1.6.1 Out-of-Box Experience

Industrial marketing research forecasts that the market value of FEC hardware components will reach $7,659 million by the year 2022 [10] , which indicates that more FEC-ready equipment such as routers, switches, IP gateway or hubs will be available in the market. Further, it is foreseeable that many of these products will feature with the Out-of-Box Experience (OOBE) in two forms—OOBE-based equipment and OOBE-based software.

OOBE-based equipment represents that the product vendors have integrated the FEC runtime platform with their products such as routers, switches or other gateway devices in which the consumers who purposed the equipment can easily configure and deploy FEC applications on the equipment via certain user interfaces, which is similar to the commercial router products that have graphical user interfaces for users to configure customised settings.

OOBE-based software is similar to the experience of Microsoft Windows in which the users who own FEC-compatible devices can purchase and install OOBE-based FEC software to their devices towards enabling FEC runtime environment and the SCANC mechanisms without any extra low-level configuration.

The OOBE-based FEC faces challenges in defining standardization for software and hardware. First, OOBE-based equipment raises a question to the vendors in what FEC platform and the related software packages should be included in their products? Second, OOBE-based software raises a question to the vendors regarding compatibility. Specifically, users may have devices in heterogeneous specification and processing units (e.g. x86, ARM etc.) in which the vendor may need to provide a version for each type of hardware. Moreover, developing and maintaining such an OOBE-based software can be extremely costly unless a corresponding common specification or standard for hardware exist.

1.6.2 Open Platforms

At this stage, besides the commercial platforms such as Cisco IOX for fog computing, there is a few number of open platforms for supporting FEC. However, most of the platforms are in the early stage in which they have limited support in deployment. Below, we summarize the characteristics of each platform.

OpenStack++ [25] is a framework developed by Carnegie Mellon University Pittsburgh for providing VM-based cloudlet platform on regular x86 computers for mobile application offloading. Explicitly, since the recent trend intends to apply lightweight virtualization technology-based FEC, OpenStack++ is less applicable to most use cases such as hosting FEC servers on routers or hubs. Further, it also indicates that the virtualization technology used in FEC is focusing more on containerization such as Docker Containers Engine.

WSO2—IoT Server (wso2.com/iot) is an extension of the popular open source enterprise service-oriented integration platform—WSO2 server that consists of certain IoT-related mechanisms such as connecting a broad range of common IoT devices (e.g. Arduino Uno, Raspberry Pi, Android OS devices, iOS devices, Windows 10 IoT Core devices etc.) with the cloud using standard protocols such as MQTT and XMPP. Further, WSO2—IoT server includes the embedded Siddhi 3.0 component that allows the system to deploy real-time stream processes in embedded devices. In other words, WSO2—IoT server provides the FEC computing capability on outer-edge devices.

Apache Edgent (edgent.apache.org), formerly known as Quarks, is an open source runtime platform contributed by IBM. Generally, the platform provides distributed stream data processing between cloud and edge devices. Specifically, the cloud-side supports most major open platforms in the stream data processing field such as Apache Spark, Apache Storm, Apache Flink and so forth. Further, at the outer-edge, Edgent supports common open operating systems such as Linux and Android OS. In summary, by utilizing Edgent, a system can dynamically migrate the stream data processing between cloud and edge, which ideally fulfils the need in most use cases that involve edge analytics.

Current open platforms lack capability in deploying and managing FEC across all the hierarchy layers of edge networks. However, it is likely due to the inflexibility of existing commercial devices in supporting the need for FEC mechanisms configuration. On the other hand, it also indicates an opportunity for product vendors to provide the enhanced devices that support FEC.

1.4 Hierarchy of Fog and Edge Computing

In general, from the perspective of central cloud in the core network, CIoT systems can deploy FEC servers at three edge layers—inner-edge, middle-edge, and outer-edge (see Figure 1.3). Here, we summarize the characteristics of each layer

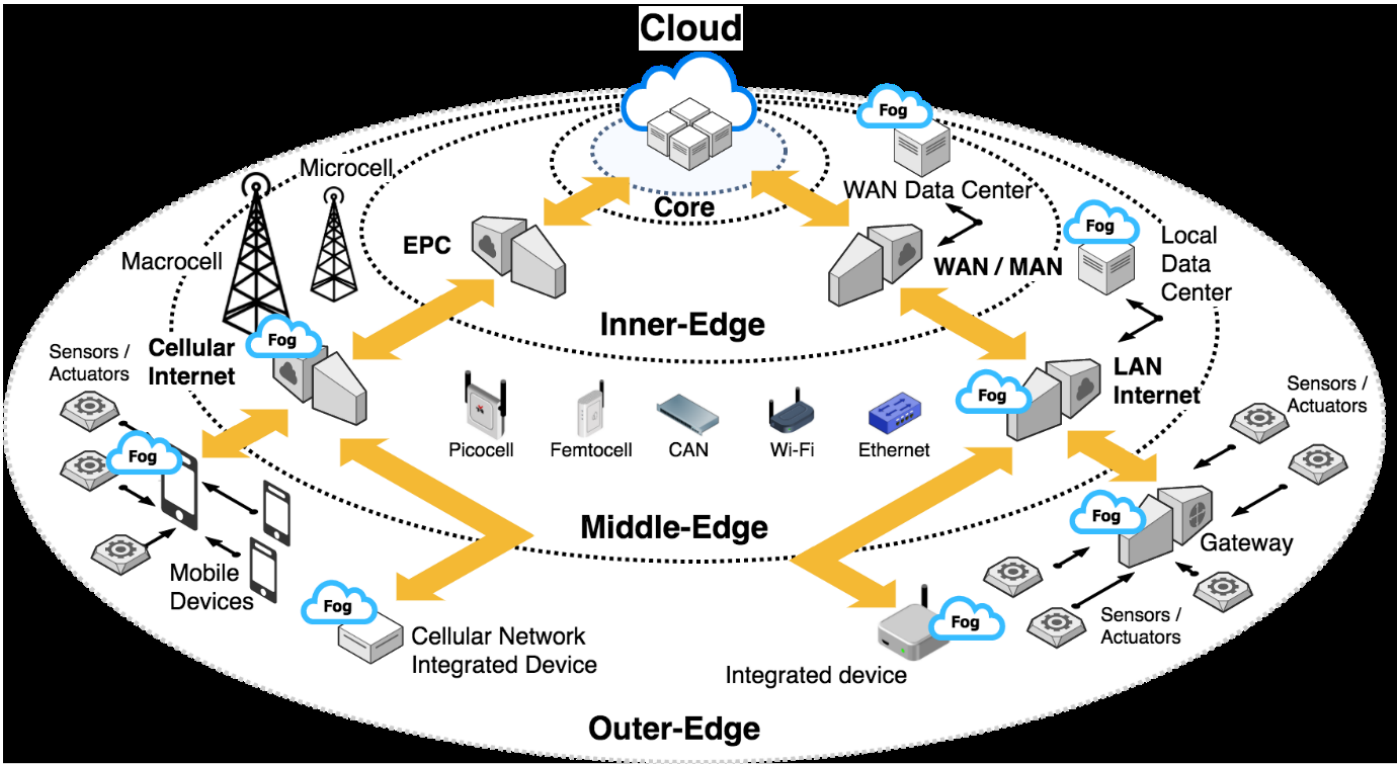


Figure 1.3. Hierarchy of fog and edge computing.

1.4.1 Inner-Edge

Inner-edge (also known as near-the-edge [4]) corresponds to countrywide, statewide and regional Wide Area Network (WAN) of enterprises, Internet Service Providers (ISPs), the data center of Evolved Packet Core (EPC) and Metropolitan Area Network (MAN). Initially, service providers at inner-edge only offer the fundamental infrastructures for connecting local networks to the global Internet. However, the recent needs in improving the Quality of Experience (QoE) of Web services have motivated the geo-distributed caching and processing mechanism at the network data centers of WAN. For example, in the commercial service aspect, Google Edge Network (peering.google.com) collaborates with ISPs to distribute data servers at the ISPs' data centers in order to improve the response speed of Google's cloud services. Further, many ISPs (e.g. AT&T, Telstra, Vodafone, Deutsche Telekom etc.) are aware that many local businesses require low latency cloud and hence, they have offered local cloud within the country. Based on the reference architecture of fog computing [8] , the WAN-based cloud data centers can be considered as the fog of inner-edge.

1.4.2 Middle-Edge

Middle-edge corresponds to the environment of the most common understanding of FEC, which consists of two types of networks—Local Area Network (LAN) and cellular network. To summarize, LAN includes ethernet, Wireless LAN (WLAN) and Campus Area Network (CAN). Whereas, the cellular network consists of the macrocell, microcell, picocell, and femtocell. Explicitly, middle-edge covers a broad range of equipment to host FEC servers.

Local Area Network. The emerging fog computing architecture introduced by Cisco's research [7] was utilizing Internet gateway devices (e.g. Cisco IR829 Industrial Integrated Router) to provide the similar model as utility cloud services in which the gateway devices provide virtualization technologies that allow the gateway devices to support FEC mechanisms mentioned previously. Further, it is also an ideal solution to utilize the virtualization technology-enabled server computers located within the same subnet of LAN or CAN (i.e. within the one-hop range between the IoT device and the computer) with the FEC nodes. Ordinarily, such an approach is also known as local cloud, local data center or cloudlet.

Cellular Network. The idea of providing FEC mechanisms derived from the existing network virtualization technologies that have been used in various cellular networks. In general, most developed cities have wide coverage of cellular networks provided by numerous types of Base Transceiver Stations (BTSs), which are the ideal facilities to serve as roadside FEC hosts for various mobile IoT use cases such as connected vehicles, mobile healthcare, virtual or augmented reality, which require rapid process and response on the real-time data stream. Therefore, major telecommunication infrastructure and equipment providers such as Nokia, ADLink or Huawei have started providing MEC-enabled hardware and infrastructure solutions. Accordingly, it is foreseeable that in near future, cellular network-based FEC will be available in a broad range of related equipment from macrocell BTS, microcell BTS to the indoor cellular extension equipment such as picocell and femtocell [21] base stations.

1.4.3 Outer-Edge

Outer-edge, which is also known as extreme-edge, far-edge or mist [14] [15] [16] , represents the front-end of the IoT network where consists of three types of devices—constraint devices, integrated devices and IP gateway devices.

Constraint devices such as sensors or actuators are usually operated by microcontrollers that have the very limited processing power and memory. For example, Atmel ATmega328 single-chip microcontroller, which is the CPU of Arduino Uno Rev3, has only 20 MHz processing power and 32kB flash memory. Commonly, IoT administrators would not expect to deploy complex tasks to this type of devices. However, due to the "field programmable" ability of today's wireless sensors and actuators, the IoT system can always update or re-configure the program code of the devices dynamically and remotely. Explicitly, such a mechanism grants the constraint IoT devices with self-awareness feature and motivated the mist computing discipline [14] , which emphasizes the abilities of IoT devices in self-management of the interaction and collaboration among IoT devices themselves towards achieving a highly autonomous Machine-to-Machine (M2M) environment without relying on the distant cloud for all their activities.

Integrated devices are the devices operated by the processors that have the decent processing power. Further, the integrated devices have many embedded capabilities in networking (e.g. Wi-Fi and Bluetooth connectivities), embedded sensors (e.g. gyroscope, accelerator) and decent storage memory. Typically, Acorn RISC Machine (ARM) CPU-based smartphones and tablets (e.g. Android OS, iOS devices) are the cost-efficient commercial products of integrated devices that can perform sensing tasks and also can interact with the cloud via the middle-edge facilities. Although the integrated devices may have constraint OS environment that reduces the flexibility of deploying virtualization platform on them; considering the swiftly evolved ARM CPUs and the embedded sensors embedded in the integrated devices, it is foreseeable that in near future, virtualization-based FEC will be available on the integrated devices. Overall, at this stage, a few platforms such as Apache Edgent (edgent.apache.org) or Termux (termux.com) are promising approaches towards realizing FEC on the integrated devices.

IP gateway devices are also known as hubs, which act as the mediator between the constrained devices and the middle-edge devices. Commonly, because of the need for energy efficient wireless communication, many constraint devices do not operate in IP network, which usually requires the energy-intensive Wi-Fi (e.g. IEEE 802.11g/n/ac). Instead, the constraint devices are communicated using the protocols that consume less energy, such as Bluetooth Low Energy, IEEE 802.15.4 (e.g. ZigBee) or Z-Wave. Further, since the low energy communication protocols do not directly connect with the IP network, the system would use IP gateway devices to relay the communication messages between the constraint devices and the Internet gateway (e.g. routers). Hence, the backend cloud is capable to interact with the frontend constraint devices. In general, the Linux OS-based IP gateway devices such as Prota's hub (prota.info), Raspberry Pi or ASUS Tinker Board can easily host virtualization environment such as Docker Containers Engine. Hence, it is common to see that research projects [22] [23] [24] have been utilizing IP gateway devices as FEC node.

Unit5

4.3 Network Slicing in 5G

In recent years, numerous research initiatives are taken by industries and academia to explore different aspects of 5G. Network architecture and its associated physical and MAC layer management are among the prime focuses of current 5G research works. The impact of 5G in different real-world applications, sustainability, and quality expectations are also getting predominant in the research arena. However, among the ongoing researches in 5G, network slicing is drawing more attractions since this distinctive feature of 5G aims at supporting diverse requirements at the finest granularity over a shared network infrastructure [6] [7].

Network slicing in 5G refers to sharing a physical network's resources to multiple virtual networks. More precisely, network slices are regarded as a set of virtualized networks on the top of a physical network [8]. The network slices can be allocated to specific applications/services, use cases or business models to meet their requirements.

Each network slice can be operated independently with its own virtual resources, topology, data traffic flow, management policies, and protocols. Network slicing usually requires implementation in an end-to-end manner to support co-existence of heterogeneous systems [9].

The network slicing paves the way for customized connectivity among a high number of inter-connected end-to-end devices. It enhances network automation and leverages the full capacity of SDN and NFV. Also, it helps to make the traditional networking architecture scalable according to the context. Since network slicing shares a common underlying infrastructure to multiple virtualized networks, it is considered as one of the most cost-effective ways to use network resources and reduce both capital and operational expenses [10]. Besides, it ensures that the reliability and limitations (congestion, security issues) of one slice do not affect the others. Network slicing assists isolation and protection of data, control and management plane that enforce security within the network. Moreover, network slicing can be extended to multiple computing paradigms such as Edge [11], Fog [14] and Cloud that eventually improves their interoperability and helps to bring services closer to the end user with less Service Level Agreement (SLA) violations [12].

Apart from the benefits, the network slicing in current 5G context is subjected to diversified challenges, however. Resource provisioning among multiple virtual networks is difficult to achieve since each virtual network has a different level of resource affinity and it can be changed with the course of time. Besides, mobility management and wireless resource virtualization can intensify the network slicing problems in 5G. End-to-End slice orchestration and management can also make network slicing complicated.

Recent researches in 5G network slicing mainly focus on addressing the challenges through efficient network slicing frameworks. Extending the literature [12] [13], we depicted a generic framework for 5G network slicing in Figure 4.1 The framework consists of three main layers: *Infrastructure layer*, *Network Function layer*, and *Service layer*.

Infrastructure layer: The infrastructure layer defines the actual physical network architecture. It can be expanded from Edge Cloud to remote Cloud through radio access network and the core network. Different software defined techniques are encapsulated to facilitate resource abstraction within the core network and the radio access network. Besides, in this layer, several policies are conducted to deploy, control, manage andborchestrate the underlying infrastructure. This layer allocates resources (compute, storage, bandwidth, etc.) to network slices in such way that upper layers can get access to handle them according to the context.

Network Function and Virtualization Layer: The network function and virtualization layer executes all the required operations to manage the virtual resources and network function's life cycle. It also facilitates

optimal placement of network slices to virtual resources and chaining of multiple slices so that they can meet specific requirements of a particular service or application. SDN, NFV and different virtualization techniques are considered as the significant technical aspect of this layer. This layer explicitly manages the functionality of core and local radio access network. It can handle both coarse-grained and fine-grained network functions efficiently.

Service and Application Layer: The service and application layer can be composed by connected vehicles, virtual reality appliances, mobile devices, etc. having a specific use case or business model and represent certain utility expectations from the networking infrastructure and the network functions. Based on requirements or high-level description of the service or applications, virtualized network functions are mapped to physical resources in such way that SLA for the respective application or service does not get violated.

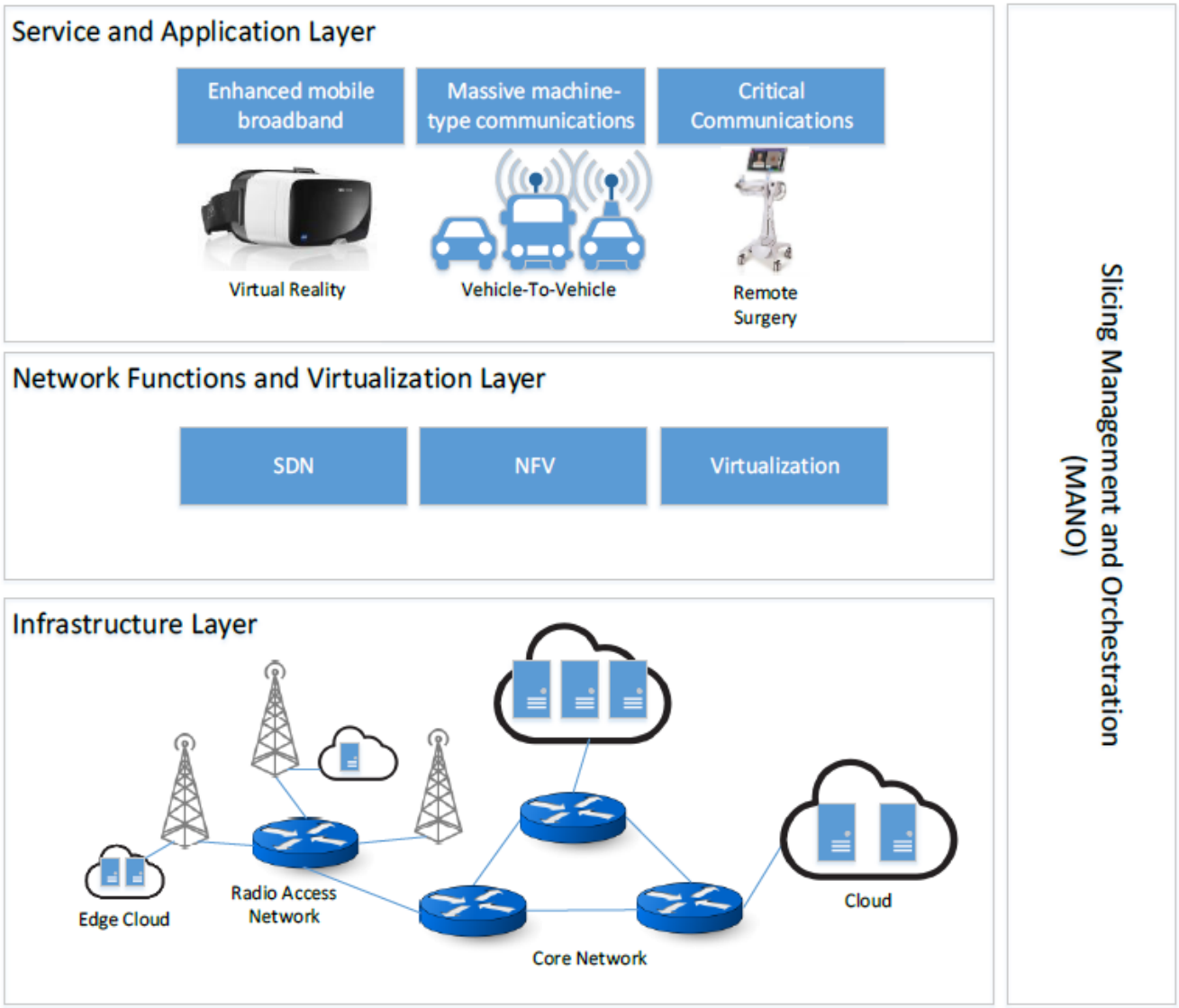


Figure 4.1: Generic 5G Slicing Framework.

Slicing Management and Orchestration (MANO): The functionality of the above layers are explicitly monitored and managed by the slicing management and orchestration layer. The main task of this layer includes;

1. Creation of virtual network instances upon the physical network by using the functionality of the infrastructure layer.

2. Mapping of network functions to virtualized network instances to build a service chain with the association of network function and virtualization layer.

3. Maintaining communication between service/application and the network slicing framework to manage the lifecycle of virtual network instances and dynamically adapt or scale the virtualized resources according to the changing context.

The logical framework of 5G network slicing is still evolving. Retaining the basic structure, extension of this framework to handle the future dynamics of network slicing can be a potential approach to further standardization of 5G.

According to Huawei high-level perspective of 5G network [42], Cloud-Native network architecture for 5G has the following characteristics: 1) it provides Cloud data center based architecture and logically independent network slicing on the network infrastructure to support different application scenarios. 2) It uses Cloud-RAN1 to build radio access networks (RAN) to provide a substantial number of connections and implement 5G required on-demand deployments of RAN functions. 3) It provides simpler core network architecture and provides on-demand configuration of network functions via user and control plane separation, unified database management, and component-based functions, and. 4) In automatic manner, it implements network slicing service to reduce operating expenses.

In the following section, we intend to review the state-of-the-art related work on network slice management happening in Cloud computing literature. Our survey in this area can help researcher to apply advances and innovation in 5G and Clouds reciprocally.

.4 Network Slicing in Software Defined Clouds

Virtualization technology has been the cornerstone of the resource management and optimization in Cloud data centers for the last decade. Many research proposals have been expressed for VM placement and Virtual Machine (VM) migration to improve utilization and efficiency of both physical and virtual servers [15]. In this section, we focus on the state of the art network-aware VM/VNF management in line with the aim of the report, i.e., network slicing management for SDCs. Figure 4.2 illustrates our proposed taxonomy of network-aware VM/VNF management in SDCS. Our taxonomy classifies existing works based on the objective of the research, the approach used to address the problem, the exploited optimization technique, and finally the evaluation technique used to validate the approach. In the remaining parts of this section, we cover network slicing from three different perspectives and map them to the proposed taxonomy: Network-aware VM management, Network-aware VM migration, and VNF management.

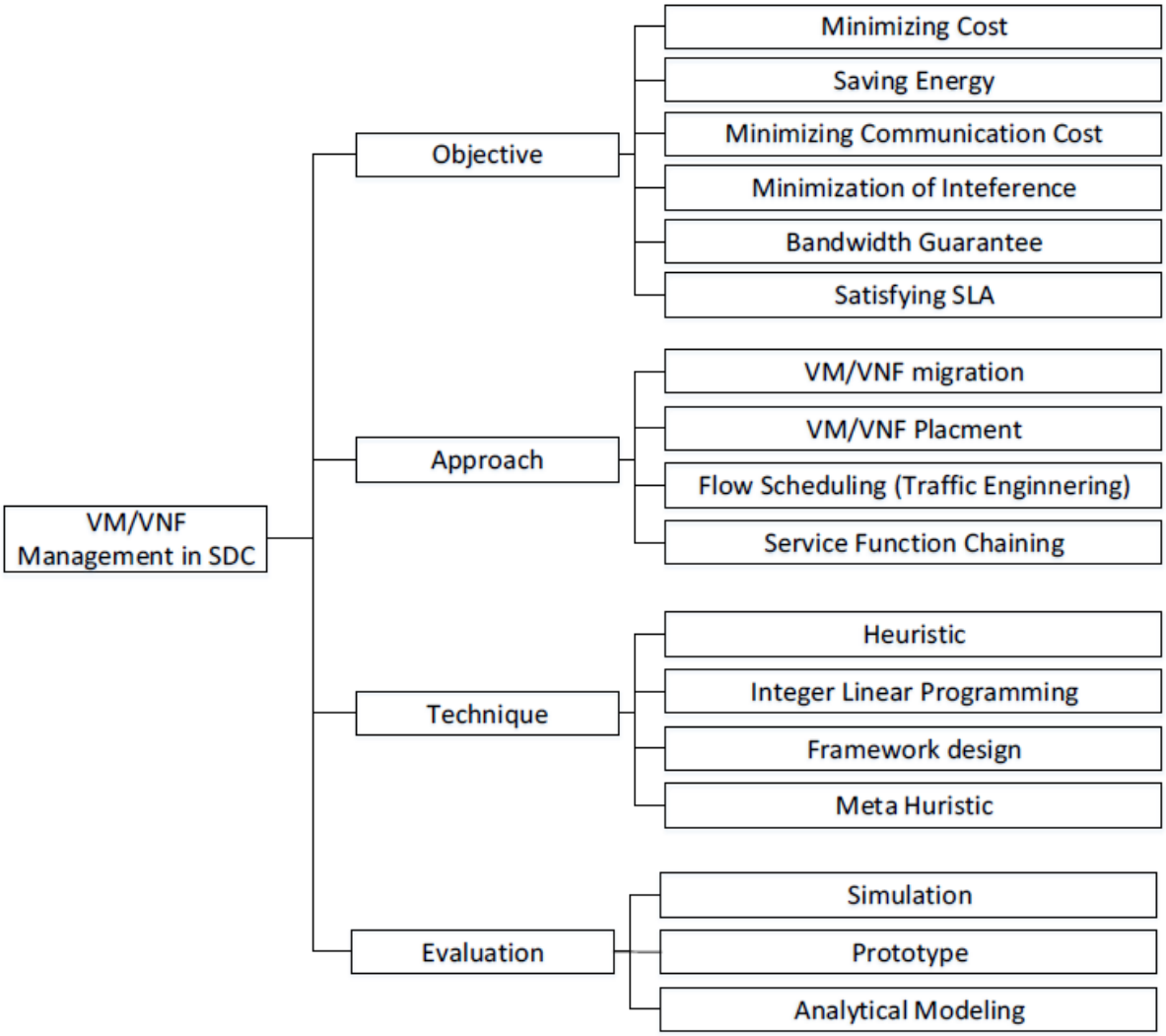


Figure 4.2: Taxonomy of network-aware VM/VNF Management in software-defined Clouds

4.4.1 Network-aware Virtual Machines Management

Cziva et al. [15] present an orchestration framework to exploit time-based network information to live migrate VMs and minimize the network cost. Wang et al. [16] propose a VM placement mechanism to reduce the number of hops between communicating VMs, save energy, and balance the network load. Remedy [17] relies on SDN to monitor the state of the network and estimate the cost of VM migration. Their technique detects congested links and migrates VMs to remove congestion on those links.

Jiang et al. [18] worked on joint VM placement and network routing problem of data centers to minimize network cost in real-time. They proposed an online algorithm to optimize the VM placement and data traffic routing with dynamically adapting traffic loads. VMPlanner [19] also optimizes VM placement and network routing. The solution includes VM grouping that consolidates VMs with high inter-group traffic, VM group placement within a rack, and traffic consolidation to minimize the rack traffic. Jin et al. [21] studied joint host-network optimization problem. The problem is formulated as an integer linear problem which combines VM placement and routing problem. Cui et al. [20] explore the joint policy-aware and network-aware VM migration problem and present a VM management to reduce network-wide communication cost in data center networks while considering the policies regarding the network functions and middleboxes. Table 4.2 summarizes the research projects on network-aware VM management

4.4.2 Network-aware Virtual Machine Migration Planning

A large body of literature focused on improving the efficiency of VM migration mechanism [22]. Bari et al. [23] propose a method for finding an efficient migration plan. They try to find a sequence of migrations to move a group of VMs to their final destinations while migration time is minimized. In their method, they monitor residual bandwidth available on the links between source and destination after performing each step in the sequence. Similarly, Ghorbani et al. [24] propose an algorithm to generate an ordered list of VMs to migrate and a set of forwarding flow changes. They concentrate on imposing bandwidth guarantees on the links to ensure that link capacity is not violated during the migration. The VM migration planning problem is also tackled by Li et al. [25] where they address the workload-aware migration problem and propose methods for selection of candidate virtual machines, destination hosts, and sequence for migration. All these studies focus on the migration order of a group of VMs while taking into account network cost. Xu et al. [26] propose an interference-aware VM live migration plan called *iAware* that minimizes both migration and co-location interference among VMs. Table 4.3 summarizes the research projects on VM migration planning.

4.4.3 Virtual Network Functions Management

Network Functions Virtualization (NFV) is an emerging paradigm where network functions such as firewalls, Network Address Translation (NAT), Virtual Private Network (VPN), etc. are virtualized and divided up into multiple building blocks called Virtualized Network Functions (VNFs). VNFs are often chained together and build Service Function Chains (SFC) to deliver a required network functionality. Han et al. [27] present a comprehensive survey of key challenges and technical requirements of NFV where they present an architectural framework for NFV. They focus on the efficient instantiation, placement and migration of VNFs and network performance. VNF-P is a model proposed by Moens and Turck [28] for efficient placement of VNFs. They propose a NFV burst scenario in a hybrid scenario in which the base demand for network function service is handled by physical resources while the extra load is handled by virtual service instances. Cloud4NFV [29] is a platform following the NFV standards by European Telecommunications Standards Institute (ETSI) to build Network Function as a Service using a Cloud platform. Their VNF Orchestrator exposes RESTful APIs allowing VNF deployment. A Cloud platform such as OpenStack supports management of virtual infrastructure at the background. vConductor [30] is another NFV management system proposed by Shen et al. for the end-to-end virtual network services. vConductor has simple graphical user interfaces (GUIs) for automatic provisioning of virtual network services and supports the management of VNFs and existing physical network functions. MORSA [31] proposed as part of vConductor to perform virtual machine (VM) placement for building NFV infrastructure in the presence of conflicting objectives of involving stakeholders such as users, Cloud providers, and telecommunication network operators.

Service chain is a series of VMs hosting VNFs in a designated order with a flow goes through them sequentially to provide desired network functionality. Tabular VM migration (TVM) proposed by [32] aims at reducing the number of hops in service chain of network functions in Cloud data centers. They use VM migration to reduce the number of hops (network elements) the flow should traverse to satisfy Service level agreements (SLAs). SLA-driven Ordered Variable-width Windowing (SOVWin) is a heuristic proposed by Pai et al. [33] to address the same problem, however, using initial static placement. Similarly, an orchestrator for the automated placement of VNFs across the resources proposed by Clayman et al. [34].

Cloud Computing: Cloud computing is expected to be an inseparable part of 5G services for providing an excellent backend for applications running on the accessing devices. During last decade, Cloud has evolved as a successful computing paradigm for delivering on-demand services over the Internet. The Cloud data centers adopted virtualization technology for efficient management of resources and services. Advances in

server virtualization contributed to the cost-efficient management of computing resources in the Cloud data centers.

Recently, the virtualization notion in Cloud data centers, thanks to the advances in SDN and NFV, has extended to all resources including compute, storage, and networks which formed the concept of Software Defined Clouds (SDC) [2]. SDC aims to utilize the advances in areas of Cloud computing, system virtualization, SDN, and NFV to enhance resource management in data centers. In addition, Cloud is regarded as the foundation block for *Cloud Radio Access Network (CRAN)*, an emerging cellular framework that aims at meeting ever-growing end-users demand on 5G. In CRAN, the traditional base stations are split into radio and baseband parts. The radio part resides in the base station in the form of Remote Radio Head (RRH) unit and the baseband part in placed to Cloud for creating a centralized and virtualized Baseband Unit (BBU) pool for different base stations.

Mobile Edge Computing (MEC): Among the user proximate computing paradigms, Mobile Edge Computing (MEC) is considered as one of the key enablers of 5G. Unlike CRAN [48], in MEC, base stations and access points are equipped with Edge servers that take care of 5G related issues at the edge network. MEC facilitates a computationally enriched distributed RAN architecture upon the LTE-based networking. Ongoing researches on MEC targets real-time context awareness [49], dynamic computation offloading [50], energy efficiency [51] and multi-media caching [52] for 5G networking.

Edge and Fog Computing: Edge and Fog computing are coined to complement remote Cloud to meet the service demand of a geographically distributed large number of IoT devices. In Edge computing, the embedded computation capabilities of IoT devices or local resources accessed via ad-hoc networking are used to process IoT data. Usually, Edge computing paradigm is well suited to perform light computational tasks and does not probe global Internet unless intervention of remote (core) Cloud is required. However, not all the IoT devices are computationally enabled, or local Edge resources are computational-enriched to execute different large-scale IoT applications simultaneously. In this case, executing latency sensitive IoT applications at remote Cloud can degrade the QoS significantly [60]. Moreover, a huge amount of IoT workload sent to remote Cloud can flood the global internet and congest the network. Therefore, Fog computing is coined that offers infrastructure and software services through distributed Fog nodes to execute IoT applications within the network [54].

In Fog computing, traditional networking devices such as routers, switches, set-top boxes and proxy servers along with dedicated Nano-servers and Micro-datacenters can act as Fog nodes and create a wide area Cloud-like services both in independent or clustered manner [55]. Mobile Edge servers or Cloudlets [53] can also be regarded as Fog nodes to conduct their respective jobs in Fog enabled Mobile Cloud Computing and MEC. In some cases, Edge and Fog computing are used interchangeably although, in a broader perspective, Edge is considered as a subset of Fog Computing [56]. However, in Edge and Fog computing, the integration of 5G has already been discussed in terms of bandwidth management during computing instance migration [57] and SDN-enabled IoT resource discovery [58]. The concept of Fog radio access network (FRAN) [59] is also getting attention from both academia and industry where Fog resources are used to create BBU pool for the base stations.

Working principle of these computing paradigms largely depends on virtualization techniques. The alignment of 5G with different computing paradigms can also be analyzed through the interplay between network and resource virtualization techniques. Network Slicing is one of the key features of 5G network virtualization. Computing paradigms can also extend the vision of 5G network slicing into data center and Fog nodes. By the latter, we mean that the vision of network slicing can be applied to the shared data center network infrastructure and Fog networks to provide an end-to-end logical network for applications by establishing a

full-stack virtualized environment. This form of network slicing can also be expanded beyond a data center networks into multi-Clouds or even cluster of Fog nodes [14]. Whatever the extension may be, this creates a new set of challenges to the network, including Wide Area Network (WAN) segments, cloud data centers (DCs) and Fog resources.