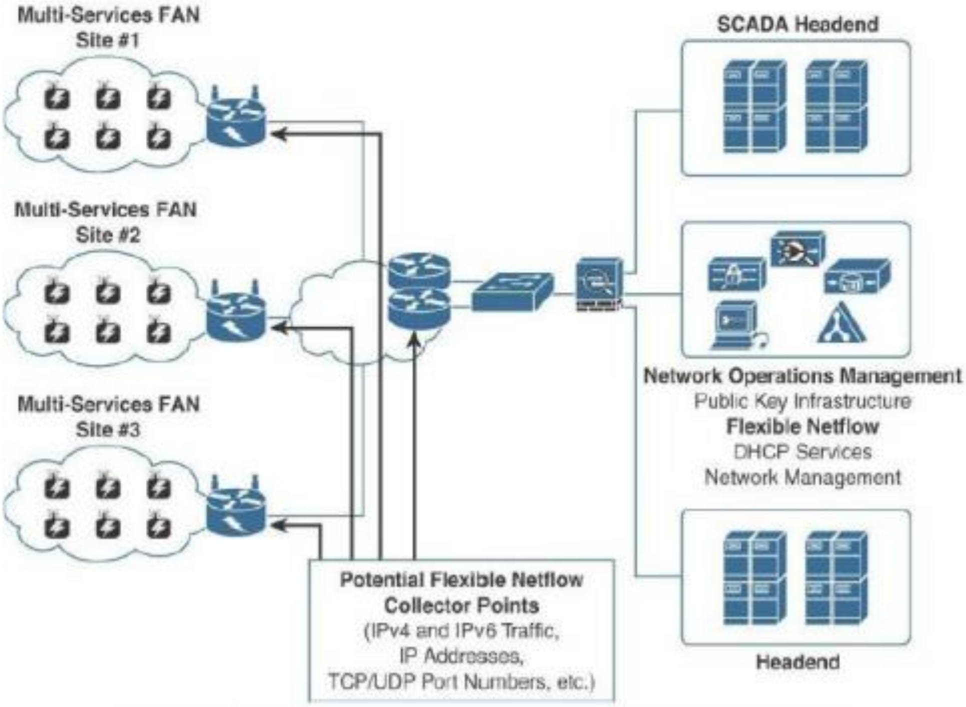
CMR
INSTITUTE OF
TECHNOLOGY

Internal Assesment Test – III - Answer Key				
Sub:	**Internet of Things**	**Code:**	**22MCA32**	
Answer Any One FULL Question from each part.		Marks	OBE	
			CO	RBT
1)	**PART I** Explain Network analytics in detail with a neat diagram.	10	CO 4	L2
	OR			
2)	With a neat diagram explain FNF architecture.	10	CO 4	L2
3)	**PART II** Explain IoT security challenges	10	CO 3	L2
	OR			
4)	Describe MQTT framework message format in detail.	10	CO 3	L3
5)	**PART III** Describe the Data analytics in detail	10	CO4	L2
	OR			
6)	Explain tunneling legacy SCADA over IP networks and SCADA protocol transition with a neat diagram.	10	CO3	L2
7)	**PART IV** Explain Formal Risk analysis structures	10	CO4	L3
	OR			
8)	Explain Purdue model for control hierarchy.	10	CO4	L2
9)	**PART V** Explain IoT Data analytics overview.	10	CO3	L2
	OR			
10)	Explain the advantages of IP as a network layer.	10	CO3	L2

1. Explain Network analytics in detail with a neat diagram.

Network analytics has the power to analyze details of communications patterns made by protocols and correlate this across the network. It allows you to understand what should be considered normal behavior in a network and to quickly identify anomalies that suggest network problems due to suboptimal paths, intrusive malware, or excessive congestion. Analysis of traffic patterns
is one of the most powerful tools in an IoT network engineer's troubleshooting arsenal. This behavior represents a key aspect that can be leveraged when performing network analytics: Network analytics offer capabilities to cope with capacity planning for scalable IoT deployment as well as security monitoring in order to detect abnormal traffic volume and patterns (such as an
unusual traffic spike for a normally quiet protocol) for both centralized or distributed architectures, such as fog computing.
Consider that an IoT device sends its traffic to specific servers, either directly to an application or an IoT broker with the data payload encapsulated in a given protocol. This represents a pair of source and destination addresses, as well as application layer–dependent TCP or UDP port numbers, which can be used for network analytics.


Multi-Services FAN Site #1
Multi-Services FAN Site #2
Multi-Services FAN Site #3
SCADA Headend
Network Operations Management
Public Key Infrastructure
Flexible Netflow
DHCP Services
Network Management
Headend
Potential Flexible Netflow Collector Points
(IPv4 and IPv6 Traffic,
IP Addresses,
TCP/UDP Port Numbers, etc.)

This behavior represents a key aspect that can be leveraged when performing network analytics: Network analytics offer capabilities to cope with capacity planning for scalable IoT deployment as well as security monitoring in order to detect abnormal traffic volume and patterns (such as an unusual traffic spike for a normally quiet protocol) for both centralized or distributed architectures, such as fog computing.

Consider that an IoT device sends its traffic to specific servers, either directly to an application or an IoT broker with the data payload encapsulated in a given protocol. This represents a pair of source and destination addresses, as well as application layer–dependent

TCP or UDP port numbers, which can be used for network analytics. One of the drivers of the adoption of an IP architectural framework for IoT is to leverage tools and processes largely known and deployed by

Internet service providers (ISPs) as well as private corporate enterprise networks. To monitor network infrastructure, de facto industry standards and protocols allow pervasive characterization of IP traffic flows, including identification of source and/or destination addresses, data timing and volume, and application types within a network infrastructure. Flow statistics can be collected at different locations in the network. For example, centralized routers or switches that aggregate subnet works as well as nodes that are highly distributed and connect the last mile of the infrastructure can be used to collect flow information. After data is collected in a known format, it can be sent to an external network analytics tools that delivers unique services to network managers, like security and performance monitoring and capacity planning.

The benefits of flow analytics, in addition to other network management services, are as follows:

■ **Network traffic monitoring and profiling:** Flow collection from the network layer provides global and distributed near-real-time monitoring capabilities. IPv4 and IPv6 network wide traffic volume and pattern analysis helps administrators proactively detect problems and quickly troubleshoot and resolve problems when they occur.

■ **Application traffic monitoring and profiling:** Monitoring and profiling can be used to gain a detailed time-based view of IoT access services, such as the application layer protocols, including MQTT, CoAP, and DNP3, as well as the associated applications that are being used over the network.

■ **Capacity planning:** Flow analytics can be used to track and anticipate IoT traffic growth and help in the planning of upgrades when deploying new locations or services by analyzing captured data over a long period of time. This analysis affords the opportunity to track and anticipate IoT network growth on a continual basis.

■ **Security analysis:** Because most IoT devices typically generate a low volume of traffic and always send their data to the same server(s), any change in network traffic behavior may indicate a cyber security event, such as a denial of service (DoS) attack. Security can be enforced by ensuring that no traffic is sent outside the scope of the IoT domain. For example, with a LoRaWAN gateway, there should be no reason to see traffic sent or received outside the LoRaWAN network server and network management system. Such traffic could indicate an attack of some sort.

■ **Accounting:** In field area networks, routers or gateways are often physically isolated and leverage public cellular services and VPNs for backhaul. Deployments may have thousands of gateways connecting the last-mile IoT infrastructure over a cellular net-work. Flow monitoring can thus be leveraged to analyze and optimize the billing, in complement with other dedicated applications, such as Cisco Jasper, with a broader scope than just monitoring data flow.

■ **Data warehousing and data mining:** Flow data (or derived information) can be ware-housed for later retrieval and analysis in support of proactive analysis of multiservice IoT infrastructures and applications.

2. With a neat diagram explain FNF architecture.

Flexible NetFlow Architecture

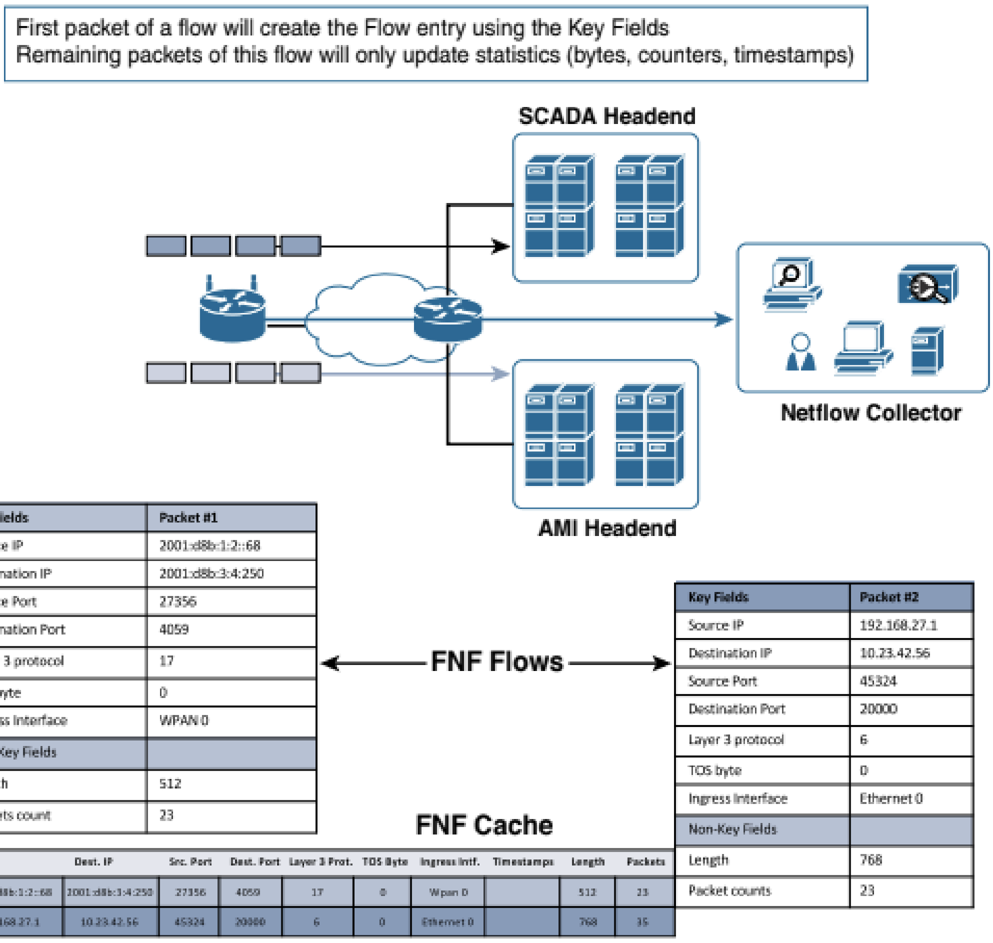
Flexible NetFlow (FNF) and IETF IPFIX (RFC 5101, RFC 5102) are examples of protocols that are widely used for networks. This section examines the fundamentals of FNF and how it may be used in an IoT deployment.

FNF is a flow technology developed by Cisco Systems that is widely deployed all over the world. Key advantages of FNF are as follows:

- Flexibility, scalability, and aggregation of flow data
- Ability to monitor a wide range of packet information and produce new information about network behavior
- Enhanced network anomaly and security detection
- User-configurable flow information for performing customized traffic identification and ability to focus and monitor specific network behavior
- Convergence of multiple accounting technologies into one accounting mechanism

FNF Components

FNF has the following main components, as shown in Figure 7-17:



First packet of a flow will create the Flow entry using the Key Fields
Remaining packets of this flow will only update statistics (bytes, counters, timestamps)

Key Fields	Packet #1
Source IP	2001:d8b:1:2::68
Destination IP	2001:d8b:3:4:250
Source Port	27356
Destination Port	4059
Layer 3 protocol	17
TOS byte	0
Ingress Interface	WPAN 0
Non-Key Fields	
Length	512
Packets count	23

Key Fields	Packet #2
Source IP	192.168.27.1
Destination IP	10.23.42.56
Source Port	45324
Destination Port	20000
Layer 3 protocol	6
TOS byte	0
Ingress Interface	Ethernet 0
Non-Key Fields	
Length	768
Packet counts	23

Src. IP	Dest. IP	Src. Port	Dest. Port	Layer 3 Prot.	TOS Byte	Ingress Intf.	Timestamps	Length	Packets
2001:d8b:1:2::68	2001:d8b:3:4:250	27356	4059	17	0	Wpan 0		512	23
192.168.27.1	10.23.42.56	45324	20000	6	0	Ethernet 0		768	35

Figure 7-17 *Flexible NetFlow overview*

- **FNF Flow Monitor (NetFlow cache):** The FNF Flow Monitor describes the NetFlow cache or information stored in the cache. The Flow Monitor contains the flow record definitions with key fields (used to create a flow, unique per flow record: match statement) and non-key fields (collected with the flow as attributes or characteristics of a flow) within the cache. Also, part of the Flow Monitor is the Flow Exporter, which contains information about the export of NetFlow information, including the destination address of the NetFlow collector. The Flow Monitor includes various cache characteristics, including timers for exporting, the size of the cache, and, if required, the packet sampling rate.

Note Each packet that is forwarded within a router or switch is examined for a set of IP packet attributes. These attributes are the IP packet identity, or *key fields*, for the flow and determine whether the packet information is unique or similar to other packets. If packet key fields are unique, a new entry in the flow record is created. The first packet of a flow creates the flow entry, using the key fields. Remaining packets of this flow only update statistics (bytes, counters, timestamps). This methodology of flow characterization is scalable because a large amount of network information is condensed into a database of NetFlow information called the *NetFlow cache*.

Additional information (non-key fields) can be added to the Flow Record and exported. The non-key fields are not used to create or characterize the flows but are exported and just added to the flow. If a field is non-key, normally only the first packet of the flow is used for the value in this field. Examples include flow timestamps, next-hop IP addresses, subnet masks, and TCP flags.

- **FNF flow record:** A flow record is a set of key and non-key NetFlow field values used to characterize flows in the NetFlow cache. Flow records may be predefined for ease of use or customized and user defined. A typical predefined record aggregates flow data and allows users to target common applications for NetFlow. User-defined records allow selections of specific key or non-key fields in the flow record. The user-defined field is the key to Flexible NetFlow, allowing a wide range of information to be characterized and exported by NetFlow. It is expected that different network management applications will support specific user-defined and predefined flow records based on what they are monitoring (for example, security detection, traffic analysis, capacity planning).
- **FNF Exporter:** There are two primary methods for accessing NetFlow data: Using the **show** commands at the command-line interface (CLI), and using an application reporting tool. NetFlow Export, unlike SNMP polling, pushes information periodically to the NetFlow reporting collector. The Flexible NetFlow Exporter allows the user to define where the export can be sent, the type of transport for the export, and properties for the export. Multiple exporters can be configured per Flow Monitor.

- **Flow export timers:** Timers indicate how often flows should be exported to the collection and reporting server.
- **NetFlow export format:** This simply indicates the type of flow reporting format.
- **NetFlow server for collection and reporting:** This is the destination of the flow export. It is often done with an analytics tool that looks for anomalies in the traffic patterns.

Figure 7-18 illustrates the analysis reported from the FNF records on a smart grid FAN. In this example, the FNF collector is able to see the patterns of traffic for various applications as well as management traffic on the FAN.

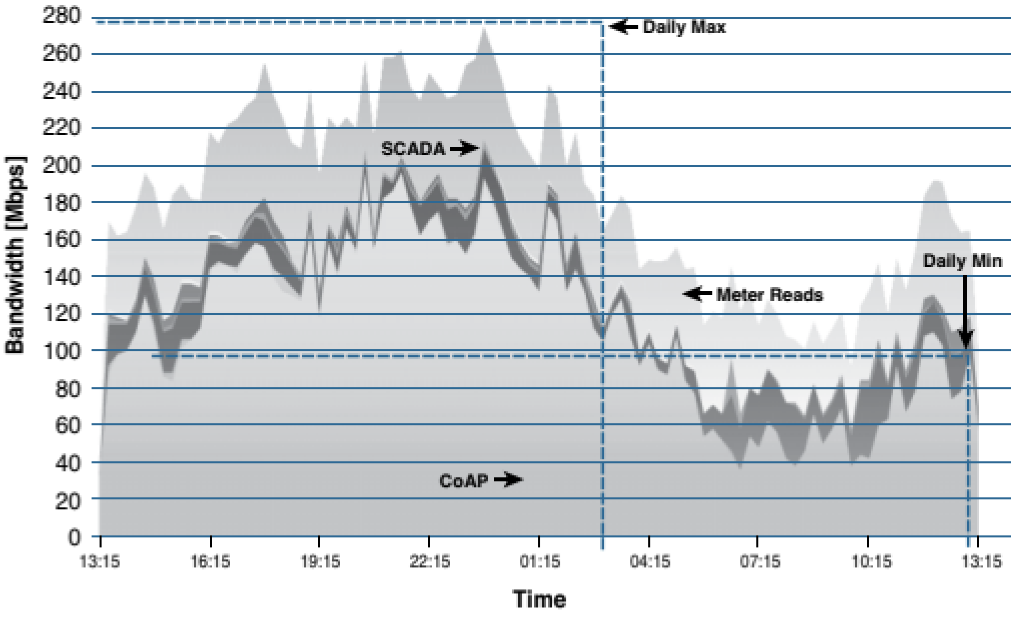
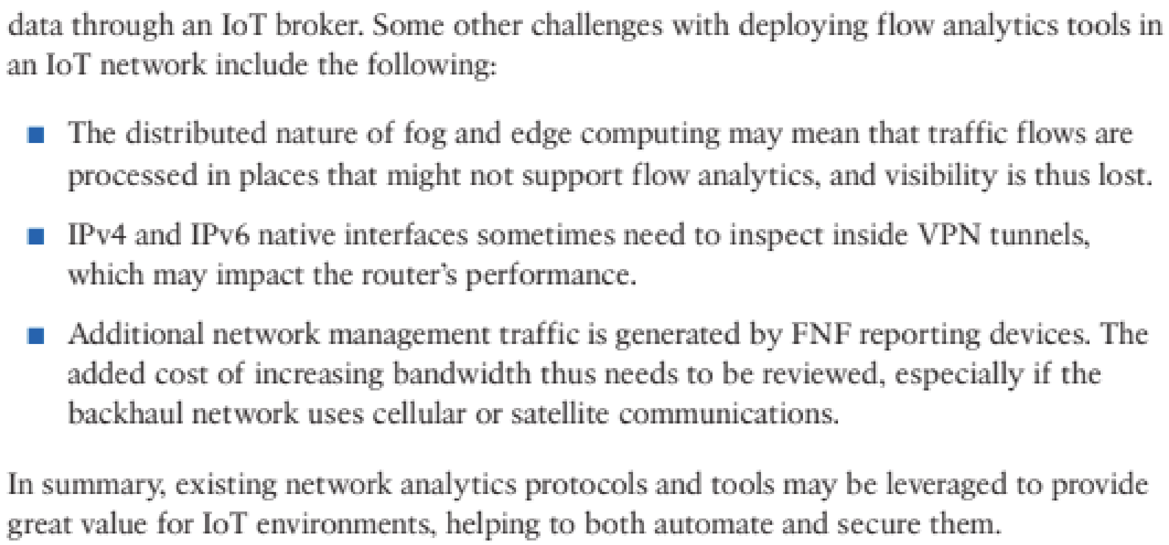


Figure 7-18 *FNF Report of Traffic on a Smart Grid FAN*

Flexible NetFlow in Multiservice IoT Networks

In the context of multiservice IoT networks, it is recommended that FNF be configured on the routers that aggregate connections from the last mile's routers. This gives a global view of all services flowing between the core network in the cloud and the IoT last-mile network (although not between IoT devices). FNF can also be configured on the last-mile gateway or fog nodes to provide more granular visibility. However, care must be taken in terms of how much northbound data is consumed through reporting.

However, flow analysis at the gateway is not possible with all IoT systems. For example, LoRaWAN gateways simply forward MAC-layer sensor traffic to the centralized LoRaWAN network server, which means flow analysis (based on Layer 3) is not possible at this point. A similar problem is encountered when using an MQTT server that sends



data through an IoT broker. Some other challenges with deploying flow analytics tools in an IoT network include the following:

- The distributed nature of fog and edge computing may mean that traffic flows are processed in places that might not support flow analytics, and visibility is thus lost.
- IPv4 and IPv6 native interfaces sometimes need to inspect inside VPN tunnels, which may impact the router's performance.
- Additional network management traffic is generated by FNF reporting devices. The added cost of increasing bandwidth thus needs to be reviewed, especially if the backhaul network uses cellular or satellite communications.

In summary, existing network analytics protocols and tools may be leveraged to provide great value for IoT environments, helping to both automate and secure them.

3. Explain IoT security challenges

The Internet of Things (IoT) refers to the network of interconnected devices that communicate and exchange data with each other over the internet. While IoT offers numerous benefits in terms of efficiency, convenience, and automation, it also introduces a host of security challenges due to its interconnected nature and the diversity of devices involved. Here are some key IoT security challenges:

1. **Data Privacy**: IoT devices often collect and transmit sensitive data, such as personal information or confidential business data. Ensuring the privacy of this data throughout its lifecycle, including storage, transmission, and processing, is a significant challenge.
2. **Device Authentication and Access Control**: Many IoT devices lack robust authentication mechanisms, making them vulnerable to unauthorized access. Weak or default credentials, improper authentication protocols, and insufficient access controls can allow attackers to compromise devices and networks.
3. **Firmware and Software Vulnerabilities**: IoT devices frequently run on outdated or unpatched firmware and software, leaving them susceptible to known vulnerabilities and exploits. Manufacturers may not provide regular updates or support, leaving devices unprotected against emerging threats.
4. **Network Security**: IoT devices often communicate over unsecured networks, such as Wi-Fi or Bluetooth, which can be vulnerable to interception and eavesdropping. Additionally, the sheer number of connected devices increases the attack surface, making it challenging to monitor and secure every entry point.
5. **Physical Security**: IoT devices deployed in uncontrolled environments, such as industrial settings or public spaces, are susceptible to physical tampering or theft. Physical security measures, such as tamper-resistant enclosures and secure installation practices, are essential to prevent unauthorized access.
6. **Supply Chain Risks**: The complex supply chain involved in IoT device manufacturing introduces various security risks, including counterfeit components, supply chain attacks, and third-party dependencies. Ensuring the integrity and security of the entire supply chain is crucial to prevent compromise at any stage.
7. **Data Integrity and Authenticity**: As IoT devices generate and exchange vast amounts of data, ensuring the integrity and authenticity of this data is essential. Without proper encryption and integrity verification mechanisms, data transmitted between devices and servers may be vulnerable to tampering or spoofing.
8. **Regulatory Compliance**: Compliance with privacy and security regulations, such as GDPR, HIPAA, or industry-specific standards, adds another layer of complexity to IoT security. Ensuring compliance across diverse geographical regions and regulatory frameworks can be challenging for IoT deployments.
9. **Scalability and Management**: Managing security across a large-scale IoT deployment with thousands or even millions of devices can be overwhelming. Effective device provisioning, monitoring, and management strategies are essential to maintain security posture and respond promptly to security incidents.

4. Describe MQTT framework message format in detail.

At the end of the 1990s, engineers from IBM and Arcom (acquired in 2006 byEurotech) were looking for a reliable, lightweight, and cost-effective protocol to monitor and control a large number of sensors and their data from a central server location, as typically used by the oil and gas industries.
Their research resulted in the development and implementation of the Message Queuing Telemetry Transport (MQTT) protocol that is now standardized by the Organization for the Advancement of Structured Information Standards (OASIS).

Considering the harsh environments in the oil and gas industries, an extremely simple protocol with only a few options was designed, with considerations for constrained nodes, unreliable WAN backhaul communications, and bandwidth constraints with variable latencies. These were some of the rationales for the selection of a client/server and publish/subscribe framework based on the TCP/IP architecture, as shown in Figure 6-10.

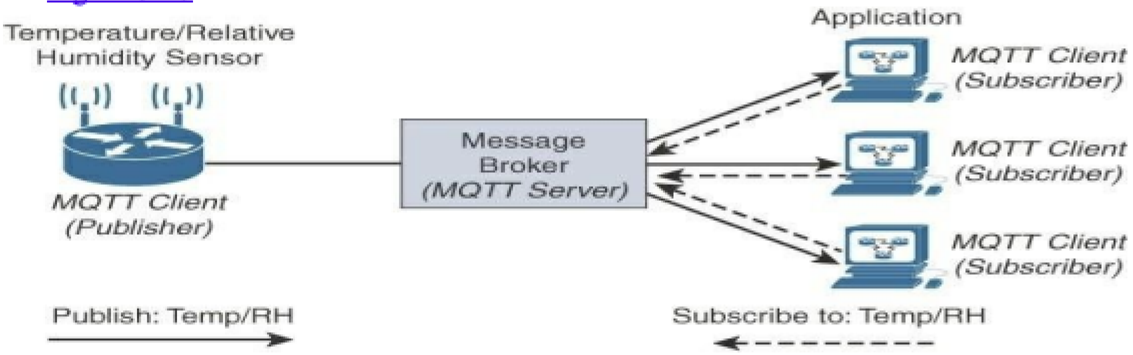


Figure 3.20 *MQTT Publish/Subscribe Framework*

An MQTT client can act as a publisher to send data (or resource information) to an MQTT server acting as an MQTT message broker. In the example illustrated in Figure 3.20, the MQTT client on the left side is a temperature (Temp) and relative humidity (RH) sensor that publishes its Temp/RH data. The MQTT server (or message broker) accepts the network connection along with application messages, such as Temp/RH data, from the publishers. It also handles the subscription and unsubscription process and pushes the application data to MQTT clients acting as subscribers.

The application on the right side of Figure 3.20 is an MQTT client that is a subscriber to the Temp/RH data being generated by the publisher or sensor on the left. This model, where subscribers express a desire to receive information from publishers, is well known. A great example is the collaboration and social networking application Twitter.

With MQTT, clients can subscribe to all data (using a wildcard character) or specific data from the information tree of a publisher. In addition, the presence of a message broker in MQTT decouples the data transmission between clients acting as publishers and subscribers. In fact, publishers and subscribers do not even know (or need to know) about each other. A benefit of having this decoupling is that the MQTT message broker ensures that information can be buffered and cached in case of network failures. This also means that publishers and subscribers do not have to be online at the same time.

MQTT control packets run over a TCP transport using port 1883. TCP ensures an ordered, lossless stream of bytes between the MQTT client and the MQTT server. Optionally, MQTT can be secured using TLS on port 8883, and WebSocket (defined in RFC 6455) can also be used.

MQTT is a lightweight protocol because each control packet consists of a 2-byte fixed header with optional variable header fields and optional payload. Control packet can contain a payload up to 256 MB.

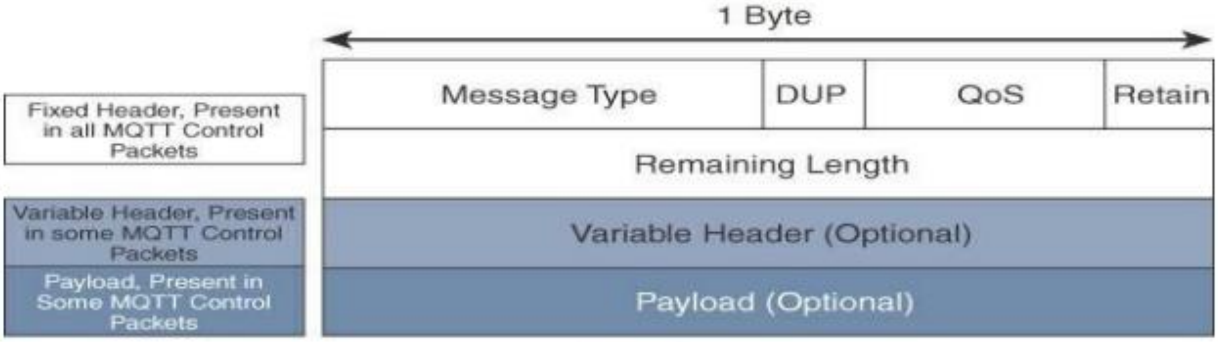
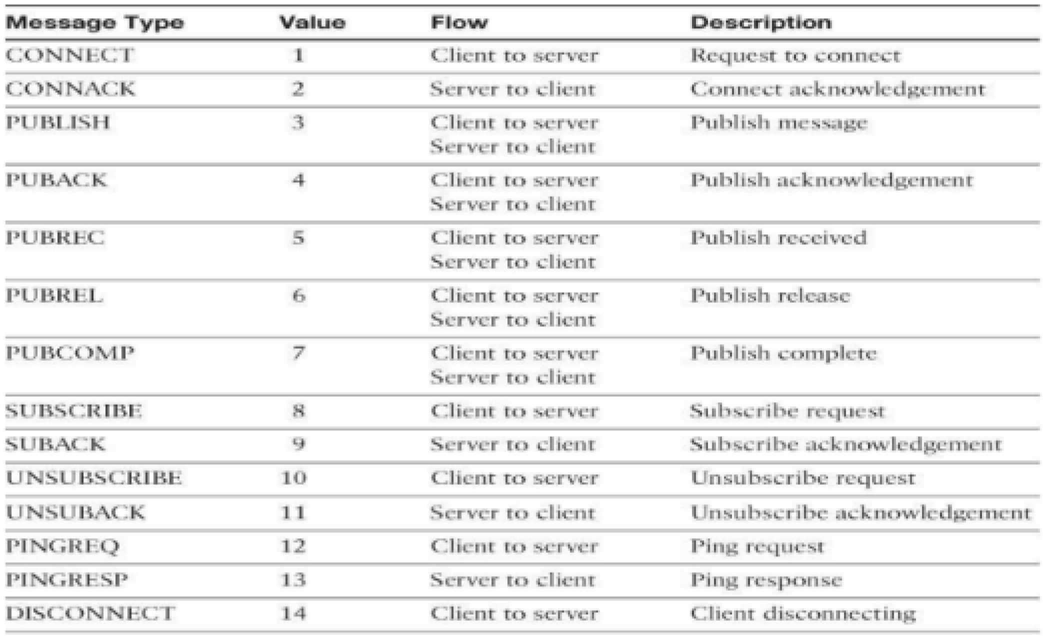

1 Byte
Message Type
DUP
QoS
Retain
Remaining Length
Variable Header (Optional)
Payload (Optional)
Fixed Header, Present in all MQTT Control Packets
Variable Header, Present in some MQTT Control Packets
Payload, Present in Some MQTT Control Packets

Figure 3.21 *MQTT Message Format*

Compared to the CoAP message format in Figure 3.17, MQTT contains a smaller header of 2 bytes compared to 4 bytes for CoAP. The first MQTT field in the header is Message Type, which identifies the kind of MQTT packet within a message. Fourteen different types of control packets are specified in MQTT version 3.1.1. Each of them has a unique value that is coded into the Message Type field. Note that values 0 and 15 are reserved. MQTT message types are summarized in Table 3.2.

The next field in the MQTT header is DUP (Duplication Flag). This flag, when set, allows the client to notate that the packet has been sent previously, but an acknowledgement was not received. The QoS header field allows for the selection of three different QoS levels. The next field is the Retain flag. Only found in a PUBLISH message, the Retain flag notifies the server to hold onto the message data. This allows new subscribers to instantly receive the last known value without having to wait for the next update from the publisher. The last mandatory field in the MQTT message header is Remaining Length. This field specifies the number of bytes in the MQTT packet following this field.

Table 6-2 *MQTT Message Types*

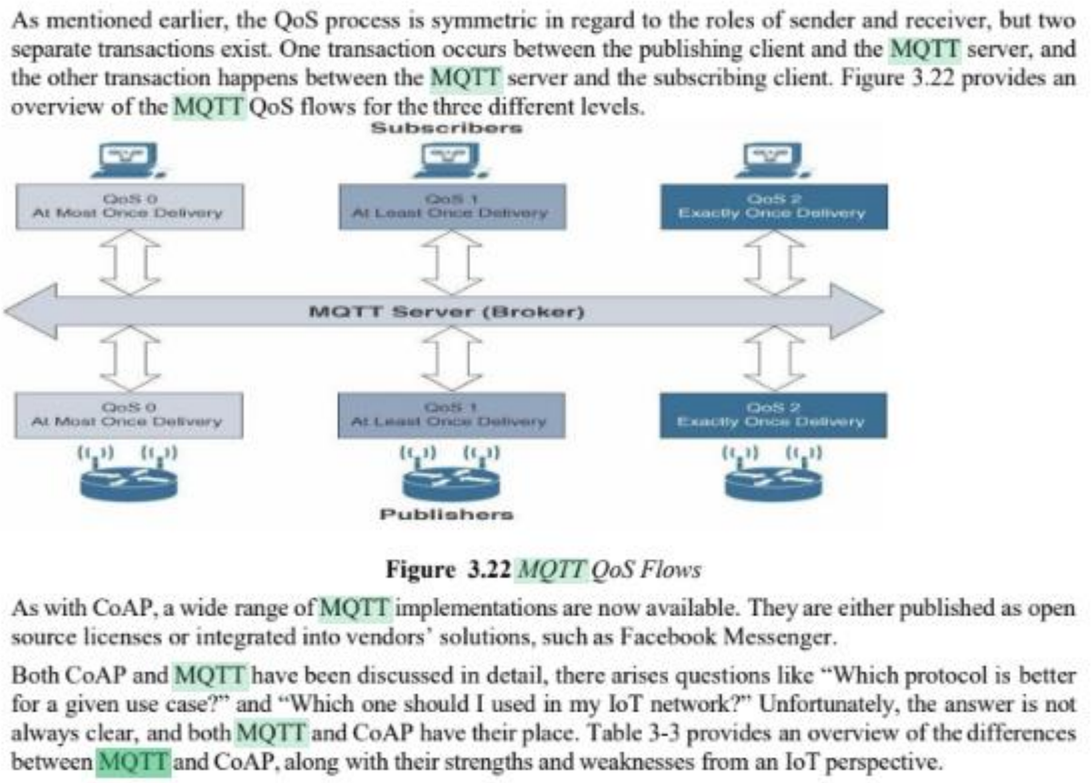


Message Type	Value	Flow	Description
CONNECT	1	Client to server	Request to connect
CONNACK	2	Server to client	Connect acknowledgement
PUBLISH	3	Client to server Server to client	Publish message
PUBACK	4	Client to server Server to client	Publish acknowledgement
PUBREC	5	Client to server Server to client	Publish received
PUBREL	6	Client to server Server to client	Publish release
PUBCOMP	7	Client to server Server to client	Publish complete
SUBSCRIBE	8	Client to server	Subscribe request
SUBACK	9	Server to client	Subscribe acknowledgement
UNSUBSCRIBE	10	Client to server	Unsubscribe request
UNSUBACK	11	Server to client	Unsubscribe acknowledgement
PINGREQ	12	Client to server	Ping request
PINGRESP	13	Server to client	Ping response
DISCONNECT	14	Client to server	Client disconnecting

MQTT sessions between each client and server consist of four phases: session establishment, authentication, data exchange, and session termination. Each client connecting to a server has a unique client ID, which allows the identification of the MQTT session between both parties. When the server is delivering an application message to more than one client, each client is treated independently.

The MQTT protocol offers three levels of quality of service (QoS). QoS for MQTT is implemented when exchanging application messages with publishers or subscribers, and it is different fromthe IP QoS that most people are familiar with. The delivery protocol is symmetric. This means the client and server can each take the role of either sender or receiver. The delivery protocol is concerned solely with the delivery of an application message from a single sender to a single receiver. These are the three levels of MQTT QoS:

- **QoS 0:** This is a best-effort and unacknowledged data service referred to as "at most once" delivery. The publisher sends its message one time to a server, which transmits it once to the subscribers. No response is sent by the receiver, and no retry is performed by the sender. The message arrives at the receiver either once or not at all.
- **QoS 1:** This QoS level ensures that the message delivery between the publisher and server and then between the server and subscribers occurs at least once. In PUBLISH and PUBACK packets, a packet identifier is included in the variable header. If the message is not acknowledged by a PUBACK packet, it is sent again. This level guarantees "at least once" delivery.
- **QoS 2:** This is the highest QoS level, used when neither loss nor duplication of messages is acceptable. There is an increased overhead associated with this QoS level because each packet contains an optional variable header with a packet identifier. Confirming the receipt of a PUBLISH message requires a two-step acknowledgement process. The first step is done through the PUBLISH/PUBREC packet pair, and the second is achieved with the PUBREL/PUBCOMP packet pair. This level provides a "guaranteed service" known as "exactly once" delivery, with no consideration for the number of retries as long as the message is delivered once.



As mentioned earlier, the QoS process is symmetric in regard to the roles of sender and receiver, but two separate transactions exist. One transaction occurs between the publishing client and the MQTT server, and the other transaction happens between the MQTT server and the subscribing client. Figure 3.22 provides an overview of the MQTT QoS flows for the three different levels.

Subscribers
QoS 0 At Most Once Delivery
QoS 1 At Least Once Delivery
QoS 2 Exactly Once Delivery
MQTT Server (Broker)
QoS 0 At Most Once Delivery
QoS 1 At Least Once Delivery
QoS 2 Exactly Once Delivery
Publishers

Figure 3.22 *MQTT QoS Flows*

As with CoAP, a wide range of MQTT implementations are now available. They are either published as open source licenses or integrated into vendors' solutions, such as Facebook Messenger.

Both CoAP and MQTT have been discussed in detail, there arises questions like "Which protocol is better for a given use case?" and "Which one should I used in my IoT network?" Unfortunately, the answer is not always clear, and both MQTT and CoAP have their place. Table 3-3 provides an overview of the differences between MQTT and CoAP, along with their strengths and weaknesses from an IoT perspective.

5 . Describe the Data analytics in detail

An Introduction to Data Analytics for IoT
In the world of IoT, the creation of massive amounts of data from sensors is common and one of the biggest challenges—not only from a transport perspective but also from a data management standpoint. A great example of the deluge of data that can be generated by IoT is found in the commercial aviation industry and the sensors that are deployed throughout an aircraft.
Modern jet engines are fitted with thousands of sensors that generate a whopping 10GB of data per second. For example, modern jet engines, similar to the one shown in Figure 7-1, may be equipped with around 5000 sensors. Therefore, a twin engine commercial aircraft with these engines operating on average 8 hours a day will generate over 500 TB of data daily, and this is just the data from the engines! Aircraft today have thousands of other sensors connected to the airframe and other systems. In fact, a single wing of a modern jumbo jet is equipped with 10,000 sensors.

Structured Versus Unstructured Data

Structured data and unstructured data are important classifications as they typically require different toolsets from a data analytics perspective. Figure 7-2 provides a high-level comparison of structured data and unstructured data.

Data in Motion Versus Data at Rest

As in most networks, data in IoT networks is either in transit ("data in motion") or being held or stored ("data at rest"). Examples of data in motion include traditional client/server exchanges, such as web browsing and file transfers, and email. Data saved to a hard drive, storage array, or USB drive is data at rest.
From an IoT perspective, the data from smart objects is considered data in motion as it passes through the network en route to its final destination. This is often processed at the edge, using fog computing. When data is processed at the edge, it may be filtered and deleted or

forwarded on for further processing and possible storage at a fog node or in the data center. Data does not come to rest at the edge.

Data at rest in IoT networks can be typically found in IoT brokers or in some sort of storage array at the data center. Myriad tools, especially tools for structured data in relational databases, are available from a data analytics perspective. The best known of these tools is Hadoop. Hadoop not only helps with data processing but also data storage.

IoT data places two specific challenges on a relational database:

- **Scaling problems:** Due to the large number of smart objects in most IoT networks that continually send data, relational databases can grow incredibly large very quickly. This can result in performance issues that can be costly to resolve, often requiring more hardware and architecture changes.
- **Volatility of data:** With relational databases, it is critical that the schema be designed correctly from the beginning. Changing it later can slow or stop the database from operating. Due to the lack of flexibility, revisions to the schema must be kept at a minimum.I oT data, however, is volatile in the sense that the data model is likely to change and evolveover time. A dynamic schema is often required so that data model changes can be made daily or even hourly.

To deal with challenges like scaling and data volatility, a different type of database, known as NoSQL, is being used. Structured Query Language (SQL) is the computer language used to communicate with an RDBMS. As the name implies, a NoSQL database is a database that does not use SQL. It is not set up in the traditional tabular form of a relational database. NoSQL databases do not enforce a strict schema, and they support a complex, evolving data model. These databases are also inherently much more scalable.

6. Explain tunneling legacy SCADA over IP networks and SCADA protocol transition with a neat diagram.

SCADA

In the world of networking technologies and protocols, IoT is relatively new. Combined with the fact that IP is the de facto standard for computer networking in general, older protocols that connected sensors and actuators have evolved and adapted themselves to utilize IP.

A prime example of this evolution is supervisory control and data acquisition (SCADA). Designed decades ago, SCADA is an automation control system that was initially implemented without IP over serial links, before being adapted to Ethernet and IPv4.

A Little Background on SCADA

For many years, vertical industries have developed communication protocols that fit their specific requirements. Many of them were defined and implemented when the most common networking technologies were serial link-based, such as RS-232 and RS-485. This led to SCADA networking protocols, which were well structured compared to the protocols described in the previous section, running directly over serial physical and data link layers.

At a high level, SCADA systems collect sensor data and telemetry from remote devices, while also providing the ability to control them. Used in today's networks, SCADA systems allow global, real-time, data-driven decisions to be made about how to improve business processes.

SCADA networks can be found across various industries, but you find SCADA mainly concentrated in the utilities and manufacturing/industrial verticals. Within these specific industries, SCADA commonly uses certain protocols for communications between devices and applications. For example, Modbus and its variants are industrial protocols used to monitor and program remote devices via a master/slave relationship. Modbus is also found in building management, transportation, and energy applications. The DNP3 and International Electrotechnical Commission (IEC) 60870-5-101 protocols are found mainly in the utilities industry, along with DLMS/COSEM and ANSI C12 for advanced meter reading (AMR). Both DNP3 and IEC 60870-5-101 are discussed in more detail later in this chapter.

As mentioned previously, these protocols go back decades and are serial based. So, transporting them over current IoT and traditional networks requires that certain accommodations be made from both protocol and implementation perspectives. These accommodations and other adjustments form various SCADA transport methods that are the focus of upcoming sections.

Adapting SCADA for IP

In the 1990s, the rapid adoption of Ethernet networks in the industrial world drove the evolution of SCADA application layer protocols. For example, the IEC adopted the Open System Interconnection (OSI) layer model to define its protocol framework. Other protocol user groups also slightly modified their protocols to run over an IP infrastructure. Benefits of this move to Ethernet and IP include the ability to leverage existing equipment and standards while integrating seamlessly the SCADA subnetworks to the corporate WAN infrastructures.

To further facilitate the support of legacy industrial protocols over IP networks, protocol specifications were updated and published, documenting the use of IP for each protocol. This included assigning TCP/UDP port numbers to the protocols, such as the following:

- DNP3 (adopted by IEEE 1815-2012) specifies the use of TCP or UDP on port 20000 for transporting DNP3 messages over IP.
- The Modbus messaging service utilizes TCP port 502.
- IEC 60870-5-104 is the evolution of IEC 60870-5-101 serial for running over Ethernet and IPv4 using port 2404.
- DLMS User Association specified a communication profile based on TCP/IP in the DLMS/COSEM Green Book (Edition 5 or higher), or in the IEC 62056-53 and IEC 62056-47 standards, allowing data exchange via IP and port 4059.

supporting these legacy protocols with modern IP networks. Let's dig deeper into how these legacy serial protocols have evolved to use IP by looking specifically at DNP3 as a representative use case.

Like many of the other SCADA protocols, DNP3 is based on a master/slave relationship. The term *master* in this case refers to what is typically a powerful computer located in the control center of a utility, and a *slave* is a remote device with computing resources found in a location such as a substation. DNP3 refers to slaves specifically as *outstations*.

Outstations monitor and collect data from devices that indicate their state, such as whether a circuit breaker is on or off, and take measurements, including voltage, current, temperature, and so on. This data is then transmitted to the master when it is requested, or events and alarms can be sent in an asynchronous manner. The master also issues control commands, such as to start a motor or reset a circuit breaker, and logs the incoming data.

The IEEE 1815-2012 specification describes how the DNP3 protocol implementation must be adapted to run either over TCP (recommended) or UDP. This specification defines connection management between the DNP3 protocol and the IP layers, as shown in Figure 6-2. Connection management links the DNP3 layers with the IP layers in addition to the configuration parameters and methods necessary for implementing the network connection. The IP layers appear transparent to the DNP3 layers as each piece of the protocol stack in one station logically communicates with the respective part in the other. This means that the DNP3 endpoints or devices are not aware of the underlying IP transport that is occurring.

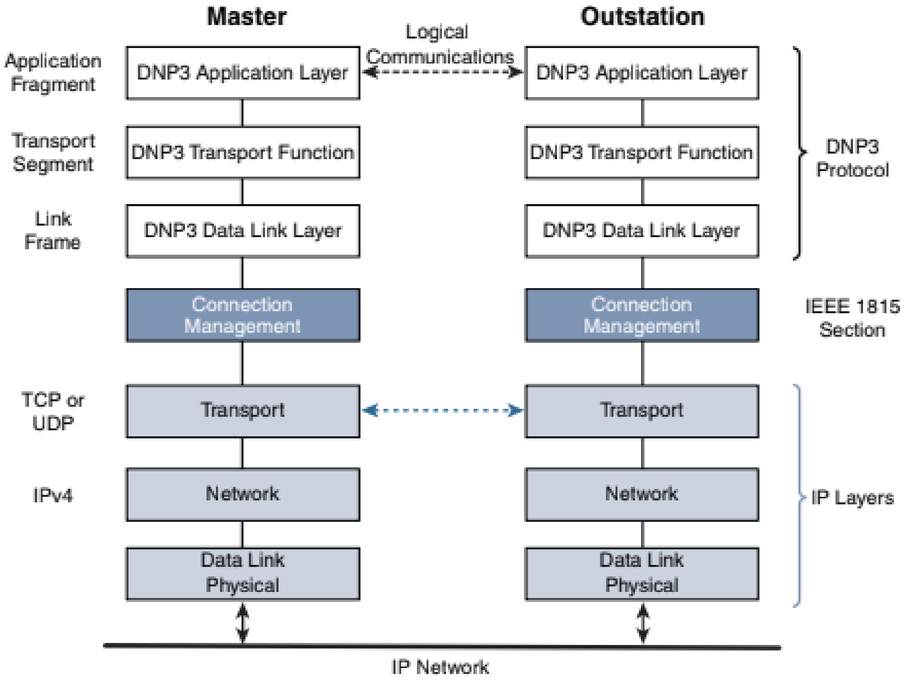


Figure 6-2 *Protocol Stack for Transporting Serial DNP3 SCADA over IP*

In Figure 6-2, the master side initiates connections by performing a TCP active open. The outstation listens for a connection request by performing a TCP passive open. *Dual endpoint* is defined as a process that can both listen for connection requests and perform an active open on the channel if required.

Master stations may parse multiple DNP3 data link layer frames from a single UDP datagram, while DNP3 data link layer frames cannot span multiple UDP datagrams. Single or multiple connections to the master may get established while a TCP keepalive timer monitors the status of the connection. Keepalive messages are implemented as DNP3 data link layer status requests. If a response is not received to a keepalive message, the connection is deemed broken, and the appropriate action is taken.

Tunneling Legacy SCADA over IP Networks

Deployments of legacy industrial protocols, such as DNP3 and other SCADA protocols, in modern IP networks call for flexibility when integrating several generations of devices or operations that are tied to various releases and versions of application servers. Native support for IP can vary and may require different solutions. Ideally, end-to-end native IP support is preferred, using a solution like IEEE 1815-2012 in the case of DNP3. Otherwise, transport of the original serial protocol over IP can be achieved either by tunneling using raw sockets over TCP or UDP or by installing an intermediate device that performs protocol translation between the serial protocol version and its IP implementation.

A raw socket connection simply denotes that the serial data is being packaged directly into a TCP or UDP transport. A socket in this instance is a standard application programming interface (API) composed of an IP address and a TCP or UDP port that is used to access network devices over an IP network. More modern industrial application servers may support this capability, while older versions typically require another device or piece of software to handle the transition from pure serial data to serial over IP using a raw socket. Figure 6-3 details raw socket scenarios for a legacy SCADA server trying to communicate with remote serial devices.

SCADA Protocol Translation

As mentioned earlier, an alternative to a raw socket connection for transporting legacy serial data across an IP network is protocol translation. With protocol translation, the legacy serial protocol is translated to a corresponding IP version. For example, Figure 6-4 shows two serially connected DNP3 RTUs and two master applications supporting DNP3 over IP that control and pull data from the RTUs. The IoT gateway in this figure performs a protocol translation function that enables communication between the RTUs and servers, despite the fact that a serial connection is present on one side and an IP connection is used on the other.

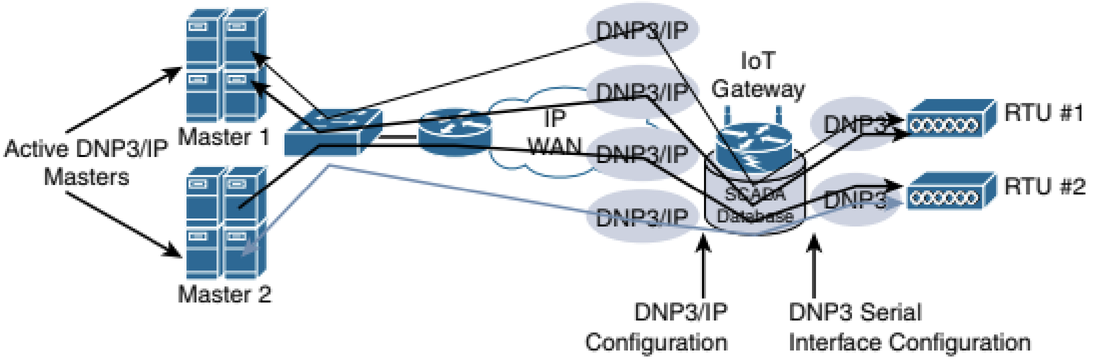


Figure 6-4 *DNP3 Protocol Translation*

By running protocol translation, the IoT gateway connected to the RTUs in Figure 6-4 is implementing a computing function close to the edge of the network. Adding computing functions close to the edge helps scale distributed intelligence in IoT networks. This can be accomplished by offering computing resources on IoT gateways or routers, as shown in this protocol translation example. Alternatively, this can also be performed directly on a node connecting multiple sensors. In either case, this is referred to as fog computing. (For more information on fog computing, see Chapter 2, “IoT Network Architecture and Design.”)

7. Explain Formal Risk analysis structures

Formal Risk Analysis Structures: OCTAVE and FAIR

Within the industrial environment, there are a number of standards, guidelines, and best practices available to help understand risk and how to mitigate it. IEC 62443 is the most commonly used standard globally across industrial verticals. It consists of a number of parts, including 62443-3-2 for risk assessments, and 62443-3-3 for foundational requirements used to secure the industrial environment from a networking and communications perspective. Also, ISO 27001 is widely used for organizational people, process, and information security management. In addition, the National Institute of Standards and Technology (NIST) provides a series of documents for critical infrastructure, such as the NIST Cybersecurity Framework (CSF). In the utilities domain, the North American Electric Reliability Corporation's (NERC's) Critical Infrastructure Protection (CIP) has legally binding guidelines for North American utilities, and IEC 62351 is the cybersecurity standard for power utilities.

The key for any industrial environment is that it needs to address security holistically and not just focus on technology. It must include people and processes, and it should include all the vendor ecosystem components that make up a control system.

In this section, we present a brief review of two such risk assessment frameworks:

- OCTAVE (Operationally Critical Threat, Asset and Vulnerability Evaluation) from the Software Engineering Institute at Carnegie Mellon University
- FAIR (Factor Analysis of Information Risk) from The Open Group

These two systems work toward establishing a more secure environment but with two different approaches and sets of priorities. Knowledge of the environment is key to determining security risks and plays a key role in driving priorities.

OCTAVE

OCTAVE has undergone multiple iterations. The version this section focuses on is OCTAVE Allegro, which is intended to be a lightweight and less burdensome process to implement. Allegro assumes that a robust security team is not on standby or immediately at the ready to initiate a comprehensive security review. This approach and the assumptions it makes are quite appropriate, given that many operational technology areas are similarly lacking in security-focused human assets. Figure 8-5 illustrates the OCTAVE Allegro steps and phases.

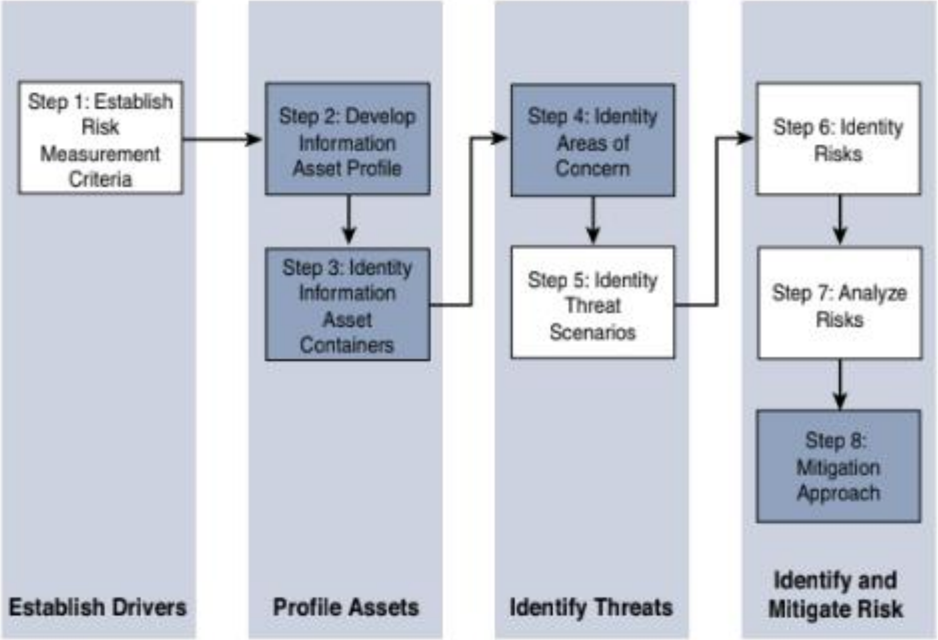

Step 1: Establish Risk Measurement Criteria
Step 2: Develop Information Asset Profile
Step 3: Identity Information Asset Containers
Step 4: Identity Areas of Concern
Step 5: Identity Threat Scenarios
Step 6: Identity Risks
Step 7: Analyze Risks
Step 8: Mitigation Approach
Establish Drivers
Profile Assets
Identify Threats
Identify and Mitigate Risk

Figure 8-5 *OCTAVE Allegro Steps and Phases (see https://blog.compass-security.com/2013/04/lean-risk-assessment-based-on-octave-allegro/).*

The first step of the OCTAVE Allegro methodology is to establish a risk measurement criterion. OCTAVE provides a fairly simple means of doing this with an emphasis on impact, value, and measurement. The point of having a risk measurement criterion is that at any point in the later stages, prioritization can take place against the reference model. (While OCTAVE has more details to contribute, we suggest using the FAIR model, described next, for risk assessment.)

The second step is to develop an information asset profile. This profile is populated with assets, a prioritization of assets, attributes associated with each asset, including owners, custodians, people, explicit security requirements, and technology assets. It is important to stress the importance of process. Certainly, the need to protect information does not disappear, but operational safety and continuity are more critical.

Within this asset profile, process are multiple substages that complete the definition of the assets. Some of these are simply survey and reporting activities, such as identifying the asset and attributes associated with it, such as its owners, custodians, human actors with which it interacts, and the composition of its technology assets. There are, however, judgment-based attributes such as prioritization. Rather than simply assigning an arbitrary ranking, the system calls for a justification of the prioritization. With an understanding of the asset attributes, particularly the technical components, appropriate threat mitigation methods can be applied. With the application of risk assessment, the level of security investment can be aligned with that individual asset.

The third step is to identify information asset containers. Roughly speaking, this is the range of transports and possible locations where the information might reside. This references the compute elements and the networks by which they communicate. However, it can also mean physical manifestations such as hard copy documents or even the people who know the information. Note that the operable target here is information, which includes data from which the information is derived.

In OCTAVE, the emphasis is on the container level rather than the asset level. The value is to reduce potential inhibitors within the container for information operation. In the OT world, the emphasis is on reducing potential inhibitors in the containerized operational space. If there is some attribute of the information that is endemic to it, then the entire container operates with that attribute because the information is the defining element. In some cases this may not be true, even in IT environments. Discrete atomic-level data may become actionable information only if it is seen in the context of the rest of the data. Similarly, operational data taken without knowledge of the rest of the elements may not be of particular value either.

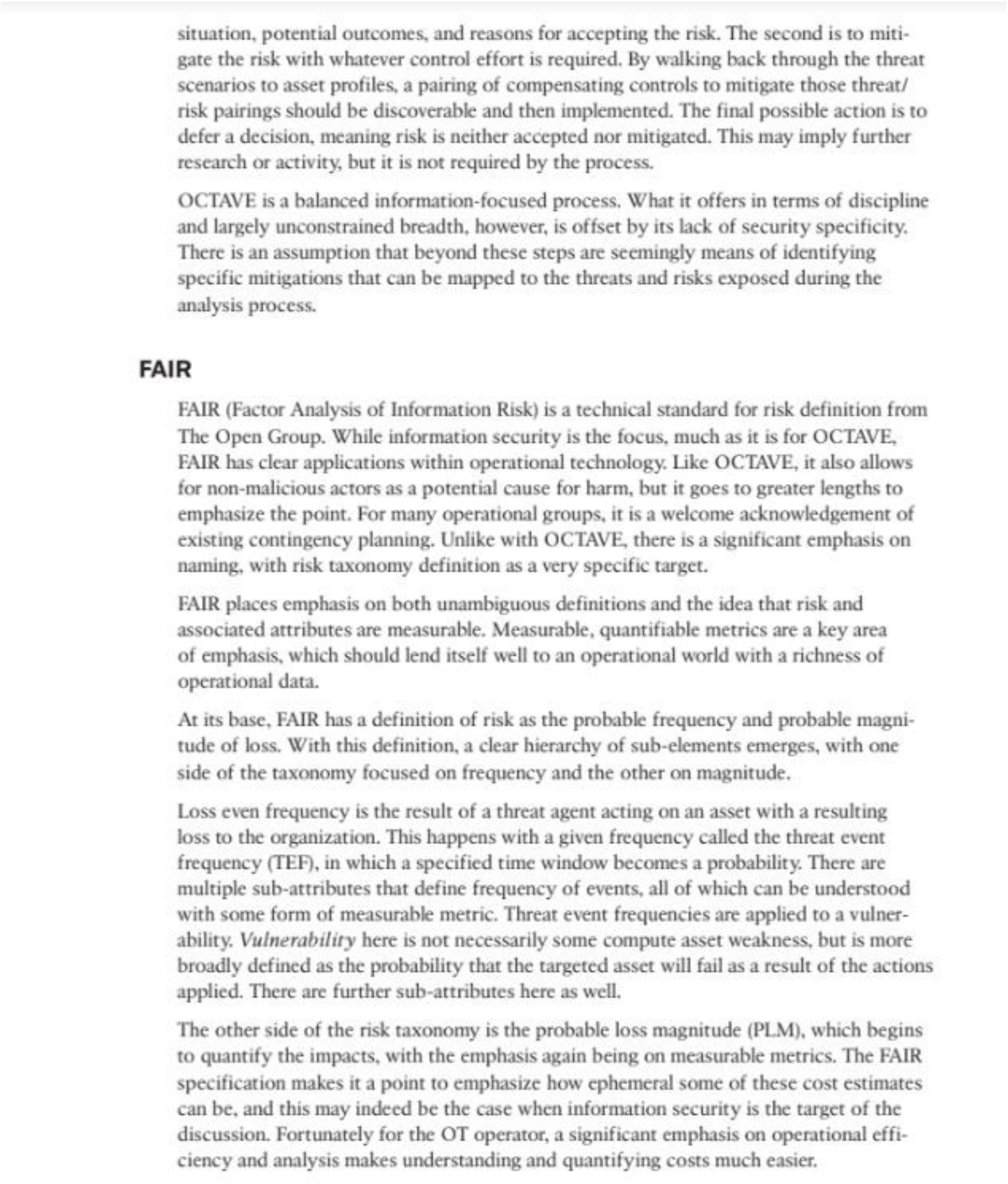
The fourth step is to identify areas of concern. At this point, we depart from a data flow, touch, and attribute focus to one where judgments are made through a mapping of security-related attributes to more business-focused use cases. At this stage, the analyst looks to risk profiles and delves into the previously mentioned risk analysis. It is no longer just facts, but there is also an element of creativity that can factor into the evaluation. History both within and outside the organization can contribute. References to similar operational use cases and incidents of security failures are reasonable associations.

Closely related is the fifth step, where threat scenarios are identified. Threats are broadly (and properly) identified as potential undesirable events. This definition means that results from both malevolent and accidental causes are viable threats. In the context of operational focus, this is a valuable consideration. It is at this point that an explicit identification of actors, motives, and outcomes occurs. These scenarios are described in threat trees to trace the path to undesired outcomes, which, in turn, can be associated with risk metrics.

At the sixth step risks are identified. Within OCTAVE, risk is the possibility of an undesired outcome. This is extended to focus on how the organization is impacted. For more focused analysis, this can be localized, but the potential impact to the organization could extend outside the boundaries of the operation.

The seventh step is risk analysis, with the effort placed on qualitative evaluation of the impacts of the risk. Here the risk measurement criteria defined in the first step are explicitly brought into the process.

Finally, mitigation is applied at the eighth step. There are three outputs or decisions to be taken at this stage. One may be to accept a risk and do nothing, other than document the



situation, potential outcomes, and reasons for accepting the risk. The second is to mitigate the risk with whatever control effort is required. By walking back through the threat scenarios to asset profiles, a pairing of compensating controls to mitigate those threat/risk pairings should be discoverable and then implemented. The final possible action is to defer a decision, meaning risk is neither accepted nor mitigated. This may imply further research or activity, but it is not required by the process.

OCTAVE is a balanced information-focused process. What it offers in terms of discipline and largely unconstrained breadth, however, is offset by its lack of security specificity. There is an assumption that beyond these steps are seemingly means of identifying specific mitigations that can be mapped to the threats and risks exposed during the analysis process.

FAIR

FAIR (Factor Analysis of Information Risk) is a technical standard for risk definition from The Open Group. While information security is the focus, much as it is for OCTAVE, FAIR has clear applications within operational technology. Like OCTAVE, it also allows for non-malicious actors as a potential cause for harm, but it goes to greater lengths to emphasize the point. For many operational groups, it is a welcome acknowledgement of existing contingency planning. Unlike with OCTAVE, there is a significant emphasis on naming, with risk taxonomy definition as a very specific target.

FAIR places emphasis on both unambiguous definitions and the idea that risk and associated attributes are measurable. Measurable, quantifiable metrics are a key area of emphasis, which should lend itself well to an operational world with a richness of operational data.

At its base, FAIR has a definition of risk as the probable frequency and probable magnitude of loss. With this definition, a clear hierarchy of sub-elements emerges, with one side of the taxonomy focused on frequency and the other on magnitude.

Loss even frequency is the result of a threat agent acting on an asset with a resulting loss to the organization. This happens with a given frequency called the threat event frequency (TEF), in which a specified time window becomes a probability. There are multiple sub-attributes that define frequency of events, all of which can be understood with some form of measurable metric. Threat event frequencies are applied to a vulnerability. *Vulnerability* here is not necessarily some compute asset weakness, but is more broadly defined as the probability that the targeted asset will fail as a result of the actions applied. There are further sub-attributes here as well.

The other side of the risk taxonomy is the probable loss magnitude (PLM), which begins to quantify the impacts, with the emphasis again being on measurable metrics. The FAIR specification makes it a point to emphasize how ephemeral some of these cost estimates can be, and this may indeed be the case when information security is the target of the discussion. Fortunately for the OT operator, a significant emphasis on operational efficiency and analysis makes understanding and quantifying costs much easier.

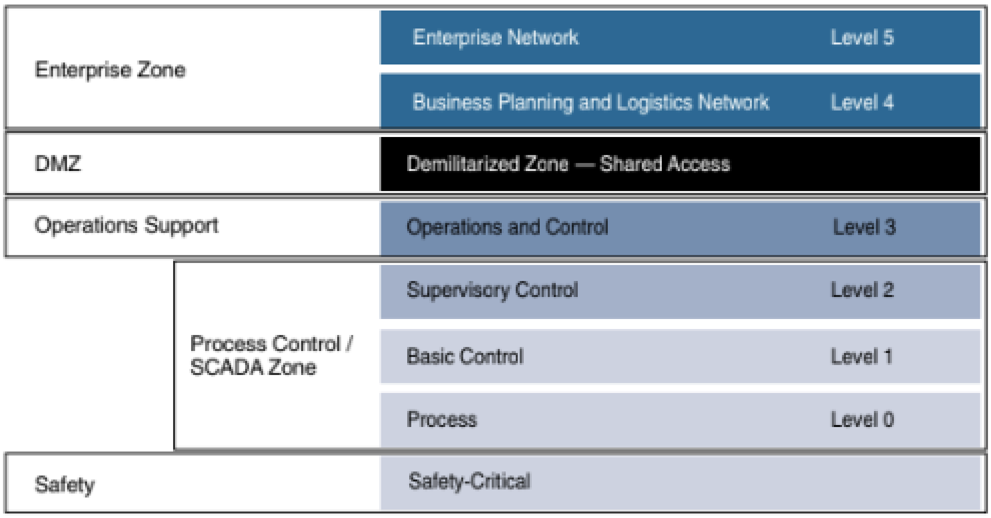
8. Explain Purdue model for control hierarchy.

The Purdue Model for Control Hierarchy

Regardless of where a security threat arises, it must be consistently and unequivocally treated. IT information is typically used to make business decisions, such as those in process optimization, whereas OT information is instead characteristically leveraged to make physical decisions, such as closing a valve, increasing pressure, and so on. Thus, the operational domain must also address physical safety and environmental factors as part of its security strategy—and this is not normally associated with the IT domain. Organizationally, IT and OT teams and tools have been historically separate, but this has begun to change, and they have started to converge, leading to more traditionally IT-centric solutions being introduced to support operational activities. For example, systems such as firewalls and intrusion prevention systems (IPS) are being used in IoT networks.

As the borders between traditionally separate OT and IT domains blur, they must align strategies and work more closely together to ensure end-to-end security. The types of devices that are found in industrial OT environments are typically much more highly optimized for tasks and industrial protocol-specific operation than their IT counterparts. Furthermore, their operational profile differs as well.

Industrial environments consist of both operational and enterprise domains. To understand the security and networking requirements for a control system, the use of a logical framework to describe the basic composition and function is needed. The Purdue Model for Control Hierarchy, introduced in Chapter 2, is the most widely used framework across industrial environments globally and is used in manufacturing, oil and gas, and many other industries. It segments devices and equipment by hierarchical function levels and areas and has been incorporated into the ISA99/IEC 62443 security standard, as shown in Figure 8-3. For additional detail on how the Purdue Model for Control Hierarchy is applied to the manufacturing and oil and gas industries, see Chapter 9, "Manufacturing," and Chapter 10, "Oil and Gas."



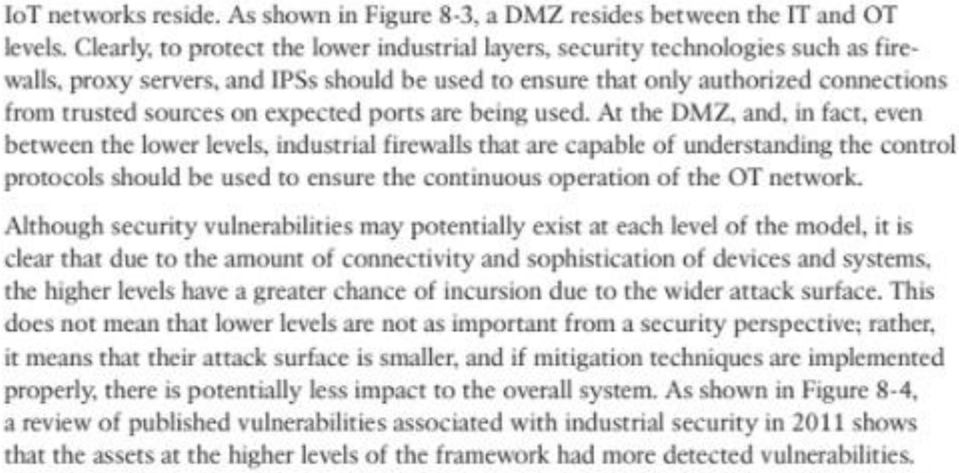
Enterprise Zone	Enterprise Network	Level 5
	Business Planning and Logistics Network	Level 4
DMZ	Demilitarized Zone — Shared Access	
Operations Support	Operations and Control	Level 3
Process Control / SCADA Zone	Supervisory Control	Level 2
	Basic Control	Level 1
	Process	Level 0
Safety	Safety-Critical	

Figure 8-3 *The Logical Framework Based on the Purdue Model for Control Hierarchy*

This model identifies levels of operations and defines each level. The enterprise and operational domains are separated into different zones and kept in strict isolation via an industrial demilitarized zone (DMZ):

- Enterprise zone
  - **Level 5: Enterprise network:** Corporate-level applications such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), document management, and services such as Internet access and VPN entry from the outside world exist at this level.
  - **Level 4: Business planning and logistics network:** The IT services exist at this level and may include scheduling systems, material flow applications, optimization and planning systems, and local IT services such as phone, email, printing, and security monitoring.
- Industrial demilitarized zone
  - **DMZ:** The DMZ provides a buffer zone where services and data can be shared between the operational and enterprise zones. It also allows for easy segmentation of organizational control. By default, no traffic should traverse the DMZ; everything should originate from or terminate on this area.
- Operational zone
  - **Level 3: Operations and control:** This level includes the functions involved in managing the workflows to produce the desired end products and for monitoring and controlling the entire operational system. This could include production scheduling, reliability assurance, systemwide control optimization, security management, network management, and potentially other required IT services, such as DHCP, DNS, and timing.
  - **Level 2: Supervisory control:** This level includes zone control rooms, controller status, control system network/application administration, and other control-related applications, such as human-machine interface (HMI) and historian.
  - **Level 1: Basic control:** At this level, controllers and IEDs, dedicated HMIs, and other applications may talk to each other to run part or all of the control function.
  - **Level 0: Process:** This is where devices such as sensors and actuators and machines such as drives, motors, and robots communicate with controllers or IEDs.
- Safety zone
  - **Safety-critical:** This level includes devices, sensors, and other equipment used to manage the safety functions of the control system.

One of the key advantages of designing an industrial network in structured levels, as with the Purdue model, is that it allows security to be correctly applied at each level and between levels. For example, IT networks typically reside at Levels 4 and 5 and use security principles common to IT networks. The lower levels are where the industrial systems and



IoT networks reside. As shown in Figure 8-3, a DMZ resides between the IT and OT levels. Clearly, to protect the lower industrial layers, security technologies such as firewalls, proxy servers, and IPSs should be used to ensure that only authorized connections from trusted sources on expected ports are being used. At the DMZ, and, in fact, even between the lower levels, industrial firewalls that are capable of understanding the control protocols should be used to ensure the continuous operation of the OT network.

Although security vulnerabilities may potentially exist at each level of the model, it is clear that due to the amount of connectivity and sophistication of devices and systems, the higher levels have a greater chance of incursion due to the wider attack surface. This does not mean that lower levels are not as important from a security perspective; rather, it means that their attack surface is smaller, and if mitigation techniques are implemented properly, there is potentially less impact to the overall system. As shown in Figure 8-4, a review of published vulnerabilities associated with industrial security in 2011 shows that the assets at the higher levels of the framework had more detected vulnerabilities.

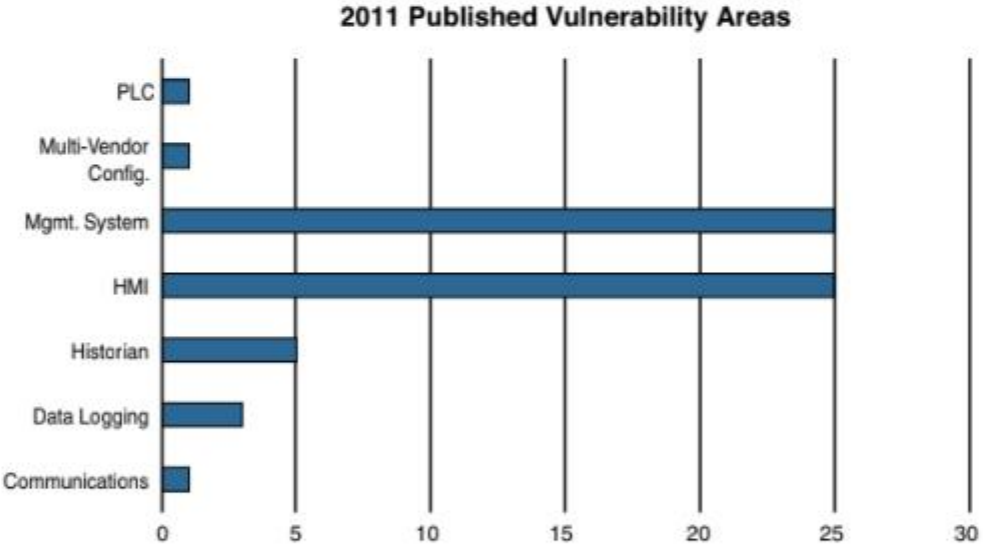
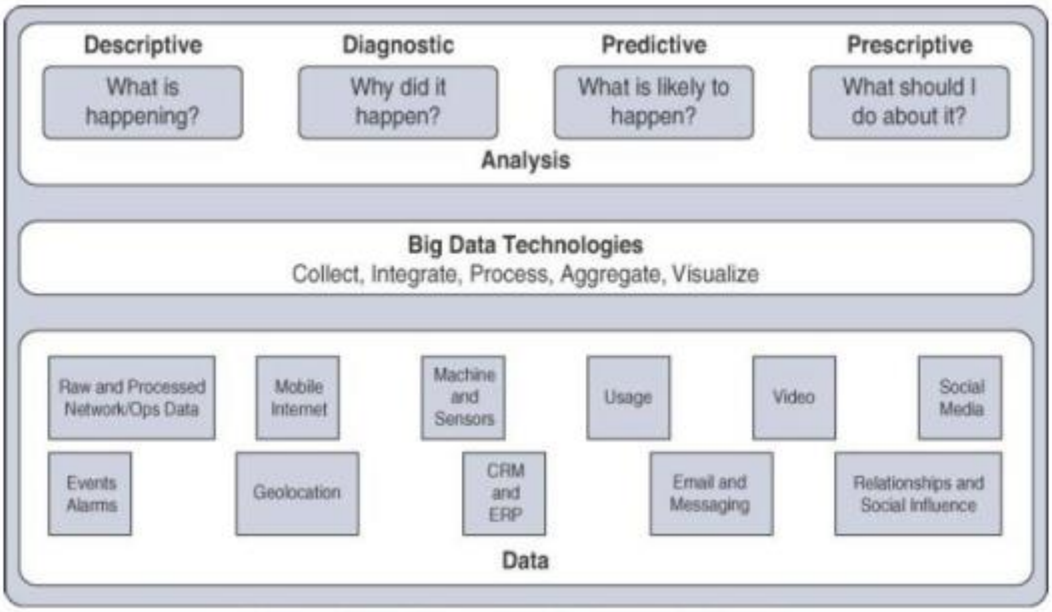


Figure 8-4 *2011 Industrial Security Report of Published Vulnerability Areas (US Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) https://ics-cert.us-cert.gov).*

9 . Explain IoT Data analytics overview.

The true importance of IoT data from smart objects is realized only when the analysis of the data leads to actionable business intelligence and insights. Data analysis is typically broken down by the types of results that are produced. As shown in Figure 7-3, there are four types of data analysis results



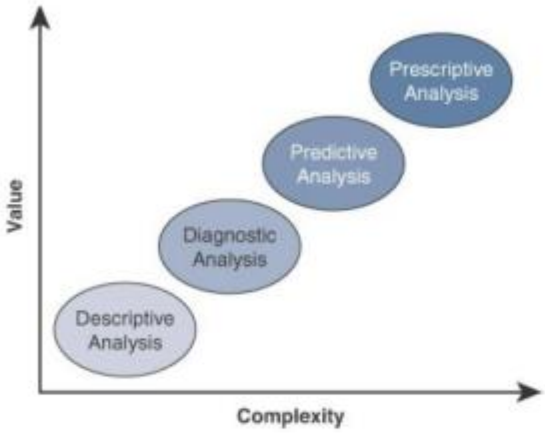
Descriptive: Descriptive data analysis tells you what is happening, either now or in the past. For example, a thermometer in a truck engine reports temperature values every second. From a descriptive analysis perspective, you can pull this data at any moment to gain insight into the current operating condition of the truck engine. If the temperature value is too high, then there may be a cooling problem or the engine may be experiencing too much load.

Diagnostic: When you are interested in the "why," diagnostic data analysis can provide the answer. Continuing with the example of the temperature sensor in the truck engine, you might wonder why the truck engine failed. Diagnostic analysis might show that the temperature of the engine was too high, and the engine overheated. Applying diagnostic analysis across the data generated by a wide range of smart objects can provide a clear picture of why a problem or an event occurred.

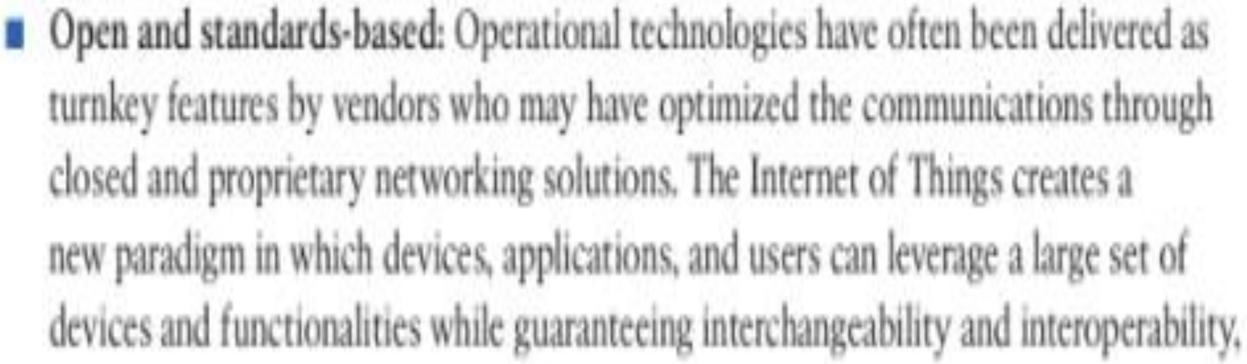
Predictive: Predictive analysis aims to foretell problems or issues before they occur. For example, with historical values of temperatures for the truck engine, predictive analysis could provide an estimate on the remaining life of certain components in the engine. These components could then be proactively replaced before failure occurs. Or perhaps if temperature values of the truck engine start to rise slowly over time, this could indicate the need for an oil change or some other sort of engine cooling maintenance.

Prescriptive: Prescriptive analysis goes a step beyond predictive and recommends solutions for upcoming problems. A prescriptive analysis of the temperature data from a truck engine might calculate various alternatives to cost-effectively maintain our truck. These calculations could range from the cost necessary for more frequent oil changes and cooling maintenance to installing new cooling equipment on the engine or upgrading to a lease on a model with a more powerful engine. Prescriptive analysis looks at a variety offactors and makes the appropriate recommendation.

Both predictive and prescriptive analyses are more resource intensive and increase complexity, but the value they provide is much greater than the value from descriptive and diagnostic analysis. Figure 7-4 illustrates the four data analysis types and how they rank as complexity and value increase. You can see that descriptive analysis is the least complex and at the same time offers the least value. On the other end, prescriptive analysis provides the most value but is the most complex to implement. Most data analysis in the IoT space relies on descriptive and diagnostic analysis, buta shift toward predictive and prescriptive analysisis understandably occurring for most businessesand organizations.



10 Explain the advantages of IP as a network layer.



- **Open and standards-based:** Operational technologies have often been delivered as turnkey features by vendors who may have optimized the communications through closed and proprietary networking solutions. The Internet of Things creates a new paradigm in which devices, applications, and users can leverage a large set of devices and functionalities while guaranteeing interchangeability and interoperability, security, and management. This calls for implementation, validation, and deployment of open, standards-based solutions. While many standards development organizations (SDOs) are working on Internet of Things definitions, frameworks, applications, and technologies, none are questioning the role of the Internet Engineering Task Force (IETF) as the foundation for specifying and optimizing the network and transport layers. The IETF is an open standards body that focuses on the development of the Internet Protocol suite and related Internet technologies and protocols.

- **Versatile:** A large spectrum of access technologies is available to offer connectivity of "things" in the last mile. Additional protocols and technologies are also used to transport IoT data through backhaul links and in the data center. Even if physical and data link layers such as Ethernet, Wi-Fi, and cellular are widely adopted, the history of data communications demonstrates that no given wired or wireless technology fits all deployment criteria. Furthermore, communication technologies evolve at a pace faster than the expected 10- to 20-year lifetime of OT solutions. So, the layered IP architecture is well equipped to cope with any type of physical and data link layers. This makes IP ideal as a long-term investment because various protocols at these layers can be used in a deployment now and over time, without requiring changes to the whole solution architecture and data flow.

- **Ubiquitous:** All recent operating system releases, from general-purpose computers and servers to lightweight embedded systems (TinyOS, Contiki, and so on), have an integrated dual (IPv4 and IPv6) IP stack that gets enhanced over time. In addition, IoT application protocols in many industrial OT solutions have been updated in recent years to run over IP. While these updates have mostly consisted of IPv4 to this point, recent standardization efforts in several areas are adding IPv6. In fact, IP is the most pervasive protocol when you look at what is supported across the various IoT solutions and industry verticals.

- **Scalable:** As the common protocol of the Internet, IP has been massively deployed and tested for robust scalability. Millions of private and public IP infrastructure nodes have been operational for years, offering strong foundations for those not familiar with IP network management. Of course, adding huge numbers of "things" to private and public infrastructures may require optimizations and design rules specific to the new devices. However, you should realize that this is not very different from the recent evolution of voice and video endpoints integrated over IP. IP has proven before that scalability is one of its strengths.

- **Manageable and highly secure:** Communications infrastructure requires appropriate management and security capabilities for proper operations. One of the benefits that comes from 30 years of operational IP networks is the well-understood network management and security protocols, mechanisms, and toolsets that are widely available. Adopting IP network management also brings an operational business application to OT. Well-known network and security management tools are easily leveraged with an IP network layer. However, you should be aware that despite the secure nature of IP, real challenges exist in this area. Specifically, the industry is challenged in securing constrained nodes, handling legacy OT protocols, and scaling operations.

- **Stable and resilient:** IP has been around for 30 years, and it is clear that IP is a workable solution. IP has a large and well-established knowledge base and, more importantly, it has been used for years in critical infrastructures, such as financial and defense networks. In addition, IP has been deployed for critical services, such as voice and video, which have already transitioned from closed environments to open IP standards. Finally, its stability and resiliency benefit from the large ecosystem of IT professionals who can help design, deploy, and operate IP-based solutions.
- **Consumers' market adoption:** When developing IoT solutions and products targeting the consumer market, vendors know that consumers' access to applications and devices will occur predominantly over broadband and mobile wireless infrastructure. The main consumer devices range from smart phones to tablets and PCs. The common protocol that links IoT in the consumer space to these devices is IP.
- **The innovation factor:** The past two decades have largely established the adoption of IP as a factor for increased innovation. IP is the underlying protocol for applications ranging from file transfer and e-mail to the World Wide Web, e-commerce, social networking, mobility, and more. Even the recent computing evolution from PC to mobile and mainframes to cloud services are perfect demonstrations of the innovative ground enabled by IP. Innovations in IoT can also leverage an IP underpinning.

In summary, the adoption of IP provides a solid foundation for the Internet of Things by allowing secured and manageable bidirectional data communication capabilities between all devices in a network. IP is a standards-based protocol that is ubiquitous, scalable, versatile, and stable. Network services such as naming, time distribution, traffic prioritization, isolation, and so on are well-known and developed techniques that can be leveraged with IP. From cloud, centralized, or distributed architectures, IP data flow can be developed and implemented according to business requirements. However, you may wonder if IP is an end-to-end requirement; this is covered in the next section.