## <https://docs.google.com/document/d/1ecPf3h1Q6g7bMA2dxf2CKTzPQ6HFVv8JIdjRgm5bCRU/edit>

Internet of things (IoT) is the network of physical devices, vehicles, home appliances, and other items embedded with **electronics** ,**software**, **sensors, actuators** , and **connectivity** which enables these things to **connect, collect exchange data.**

## **IoT Enablers:**

## Enablers in IoT are fundamental capabilities that make the Internet of Things possible and allow it to function effectively.

**1. Portability:**

* **Enables:** Compact and lightweight devices for easy deployment and mobility.
* **Benefits:** Enables wider application range like wearable health trackers, asset tracking, and remote monitoring.

**2. Miniaturization:**

* **Enables:** Integration of sensors, processors, and communication modules into small form factors.
* **Benefits:** Allows for applications in space-constrained environments like smart home devices, industrial sensors, and healthcare implants.

**3. Low Power and Low Heat:**

* **Enables:** Efficient operation with minimal power consumption and heat generation.
* **Benefits:** Extends battery life for applications like remote sensing, smart agriculture, and wearables.

**4. Connectivity:**

* **Enables:** Communication between devices and central systems for data exchange and control.
* **Benefits:** Facilitates real-time monitoring and control in smart cities, industrial automation, and vehicle telematics.

**5. Convergence:**

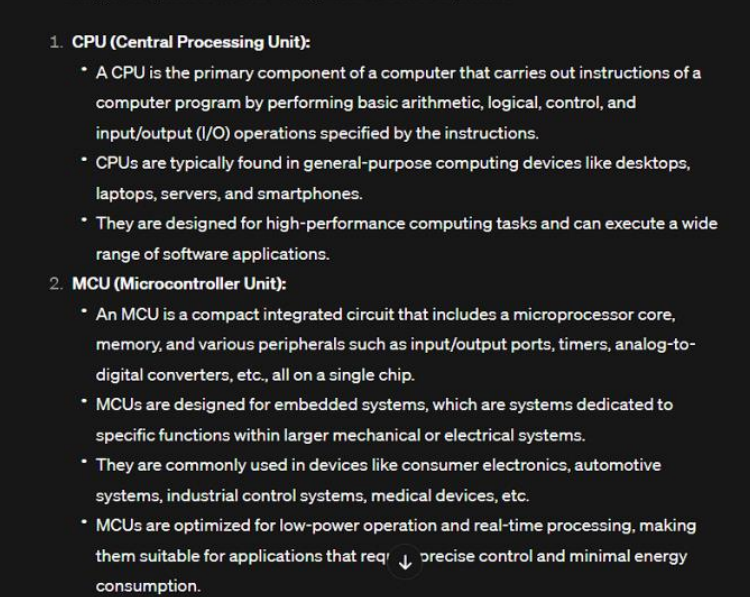
* **Enables:** Integration of disparate technologies and data into unified platforms.
* **Benefits:** Optimizes resource utilization and improves operational efficiency in smart buildings, energy management, and healthcare.

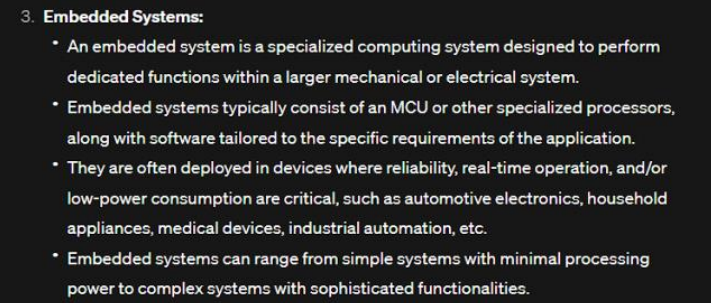
**6. Divergence:**

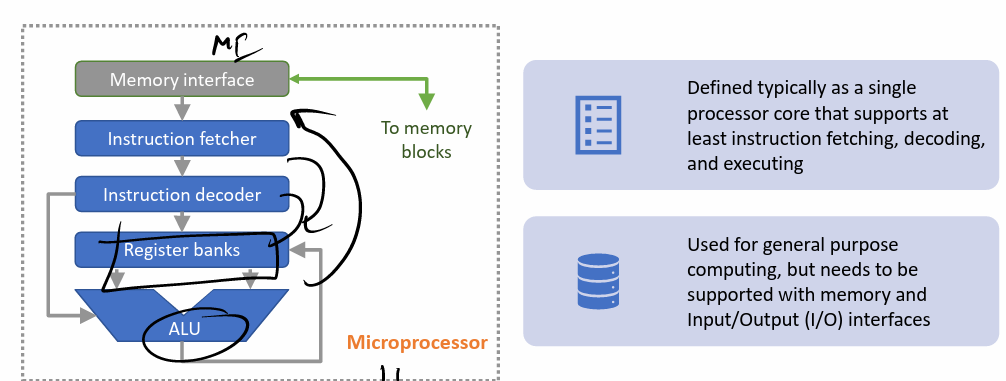
* **Enables:** Customization of solutions to address specific requirements and use cases.
* **Benefits:** Caters to diverse user needs and preferences in applications like personalized healthcare, smart home automation, and industrial IoT.

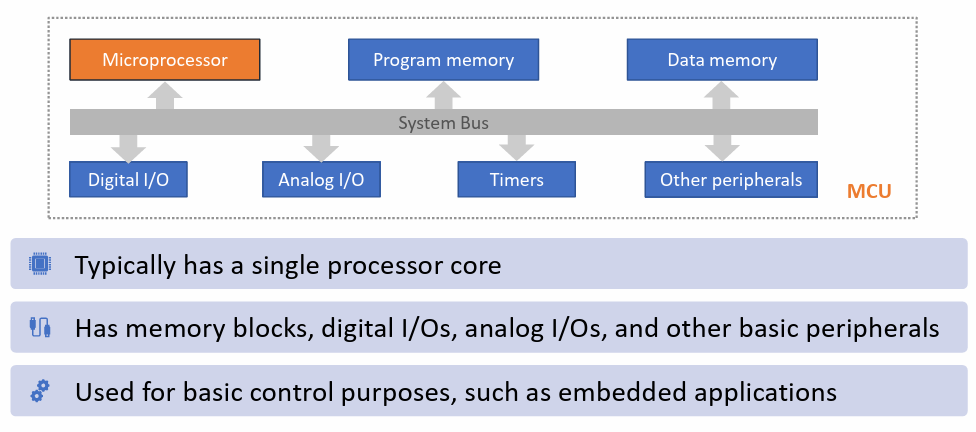
**7. Ecosystem:**

* **Enables:** Collaboration between devices, platforms, and services for value-added solutions.
* **Benefits:** Drives innovation and creates new business opportunities in areas like smart cities, agriculture, and supply chain management.









**HCI**

Why is HCI Important? Expanding on the Key Points

**Positive Impacts:**

* Effectiveness: Well-designed interfaces allow users to achieve their goals quickly and efficiently, leading to improved results and satisfaction.
* Productivity: Intuitive interfaces reduce learning curves, minimize errors, and streamline workflows, ultimately boosting productivity.
* Morale: Frustrating interfaces lead to user fatigue and dissatisfaction. Good HCI promotes positive user experiences, fostering engagement and motivation.
* Safety: In critical applications like medical devices or aviation systems, clear and intuitive interfaces can help prevent errors and accidents, ensuring safety for both users and others.

**Negative Impacts of Bad Interfaces:**

* Confusion: Complex or poorly organized interfaces leave users unsure of what to do, leading to frustration and wasted time.
* Cumbersome: Interfaces that require excessive clicks or steps to accomplish tasks become tedious and hinder user flow.
* Time-consuming: Inefficient interfaces force users to spend unnecessary time navigating and searching for information, impacting productivity.
* Uninformative: Lack of clear feedback or instructions leaves users guessing, hindering understanding and decision-making.
* Errors: Poorly designed interfaces can lead to misinterpretations and mistakes, which can have real-world consequences.

**Additional Considerations:**

* Accessibility: HCI goes beyond traditional users. Interfaces should be inclusive and cater to diverse needs, including those with disabilities.
* Cultural understanding: Interfaces should account for cultural differences and adapt to user preferences for language, symbols, and interaction styles.
* Ethical considerations: HCI should be used responsibly, prioritizing user privacy, security, and preventing manipulation or bias.

**Examples:**

* Banking apps: A user-friendly app allows transferring funds, checking balances, and managing finances easily, improving financial literacy and user confidence.
* Educational platforms: Engaging and interactive interfaces make learning fun and effective, boosting student engagement and knowledge retention.
* Self-service kiosks: Intuitive interfaces enable users to complete tasks like check-in, ordering, or bill payment quickly and efficiently.

**Traditional Interfaces:**

* **Keyboard/mouse/screen/speakers:** The classic desktop setup, still widely used for its precision and efficiency in tasks like writing, editing, and data manipulation.
* **Pen input:** Offers natural handwriting feel for drawing, note-taking, and signing documents on tablets and touchscreens.

**Emerging Interfaces:**

* **Touch:** Ubiquitous on smartphones and tablets, providing intuitive interaction through gestures and taps.
* **Speech/audio/sound:** Voice commands and voice assistants like Siri or Alexa enable hands-free control and interaction with devices.
* **Gesture, eye movement:** Advanced interfaces track hand and eye movements for more immersive and natural control in gaming, design, and presentations.
* **Tangible interfaces:** Physical objects or manipulatives used to interact with digital data, offering a more tactile and engaging experience.
* **Haptic interfaces:** These interfaces provide users with a sense of touch, allowing them to feel virtual objects and textures, adding a layer of realism and engagement to interactions. Examples include vibrating game controllers, texture-simulating touchscreens, and even exoskeletons providing physical feedback
* **Augmented reality (AR):** AR overlays digital information onto the real world, enhancing user experience and providing context-sensitive information. Imagine viewing repair instructions overlaid on a physical object or navigating with directions projected onto your surroundings.
* **Wearable computing:** These devices, like smartwatches and glasses, are worn on the body, offering constant access to information and interaction. Wearables can track health data, provide notifications, and even control smart home devices, seamlessly integrating technology into everyday life.

**Multi-modal Interfaces:**

These combine multiple input and output channels for richer and more intuitive interactions.

* **Examples:** Using speech commands and touch gestures to control a smart home, combining keyboard input with eye tracking for augmented reality design, or using both voice and facial expressions to interact with virtual assistants.
* **Example:** Imagine a surgeon using a VR headset for remote surgery, receiving haptic feedback through gloves while simultaneously viewing vital signs displayed in their AR glasses.

.

**Current Incarnations of IoT: A Look at Today's Landscape**

**What they look like now:**

Modern IoT devices encompass a wide range:

* **Smart home devices:** Connected thermostats, lights, speakers, appliances, and security systems.
* **Wearable technology:** Smartwatches, fitness trackers, and health monitors.
* **Smart appliances:** Connected refrigerators, ovens, and washing machines.
* **Industrial IoT (IIoT) sensors and actuators:** Monitoring and controlling equipment in factories, power grids, and transportation systems.
* **Connected vehicles:** Cars with telematics systems, self-driving capabilities, and in-vehicle communication.
* **Smart cities:** Connected infrastructure for traffic management, waste collection, and environmental monitoring.

**Specialized hardware:**

While some devices leverage existing hardware (e.g., smartphones), dedicated IoT devices often utilize specialized components:

* **Low-power microcontrollers:** Optimized for low energy consumption and long battery life.
* **Wireless communication protocols:** Bluetooth, Wi-Fi, Zigbee, cellular networks, depending on range and data needs.
* **Sensors:** Temperature, pressure, motion, light, sound, and various other types to gather data from the environment.
* **Actuators:** Control physical systems like motors, valves, and lights based on sensor data or commands.

**Domestic setting:**

The home remains a major player in IoT, but its reach has expanded:

* **Beyond automation:** From simple lighting control to complex security systems, entertainment hubs, and personalized experiences.
* **Healthcare and wellness:** Remote patient monitoring, medication reminders, and fitness tracking are becoming increasingly common.
* **Connected appliances:** Offering convenience, energy efficiency, and remote control capabilities.

**Mostly used for home entertainment:**

While entertainment remains a significant segment, IoT's applications are much broader:

* **Industrial automation:** Optimizing manufacturing processes, predictive maintenance, and remote monitoring.
* **Smart cities:** Improving traffic flow, energy efficiency, and public safety.
* **Agriculture:** Precision farming, irrigation management, and livestock monitoring.
* **Retail:** Inventory management, personalized customer experiences, and real-time data insights.

**All open to 3rd parties:**

Openness varies depending on the device and platform, but many offer:

* **APIs:** Application Programming Interfaces allow developers to create custom apps and integrations.
* **Developer platforms:** Tools and resources for building and deploying IoT applications.
* **Standardization efforts:** Initiatives to promote interoperability and compatibility between devices and platforms.

**Overall:**

The current incarnations of IoT showcase its immense potential beyond home automation. Specialized hardware, diverse applications, and openness to third-party development are driving its rapid growth and transformation. As the technology evolves, we can expect even more innovative and impactful applications across various domains.

## 

## **7 Design Principles: In-Depth with Examples**

**1. Equitable Use:**

* **Description:** Ensures everyone, regardless of abilities or background, can access and benefit from the design.
* **Focus:** Accessibility, inclusivity, diverse needs.
* **Example:** Public library with various sized fonts, audio books, and accessible seating.

**2. Flexibility in Use:**

* **Description:** Offers options and adapts to different needs and preferences.
* **Focus:** User choice, customization, adaptability.
* **Example:** Smartphone with touch, swipe, and voice commands for different interaction styles.

**3. Simplicity and Intuitiveness of Use:**

* **Description:** Makes things easy to understand and use, avoiding unnecessary complexity.
* **Focus:** Clarity, minimal learning curve, intuitive navigation.
* **Example:** Traffic lights with simple color codes, universally understood.

**4. Perceptible Information:**

* **Description:** Ensures information is easily noticed and understood by everyone.
* **Focus:** Different sensory modalities, clear communication, redundancy.
* **Example:** Crosswalk with both visual (lights) and auditory (beeps) signals.

**5. Tolerance for Error:**

* **Description:** Minimizes the impact of mistakes, allowing for easy recovery.
* **Focus:** Error prevention, clear feedback, user-friendly correction mechanisms.
* **Example:** Microwave with a timer that can be easily stopped if accidentally set too long.

**6. Low Physical Effort:**

* **Description:** Minimizes physical strain and discomfort during use.
* **Focus:** Ergonomics, comfort, reduced physical demands.
* **Example:** Door with a lever handle instead of a knob, easier to use for people with arthritis.

**7. Size and Space for Approach and Use:**

* **Description:** Provides adequate space and accessibility for everyone, regardless of physical abilities.
* **Focus:** Universal design, accessibility standards, clear circulation paths.
* **Example:** Public restroom with spacious stalls and grab bars, accommodating wheelchair users.

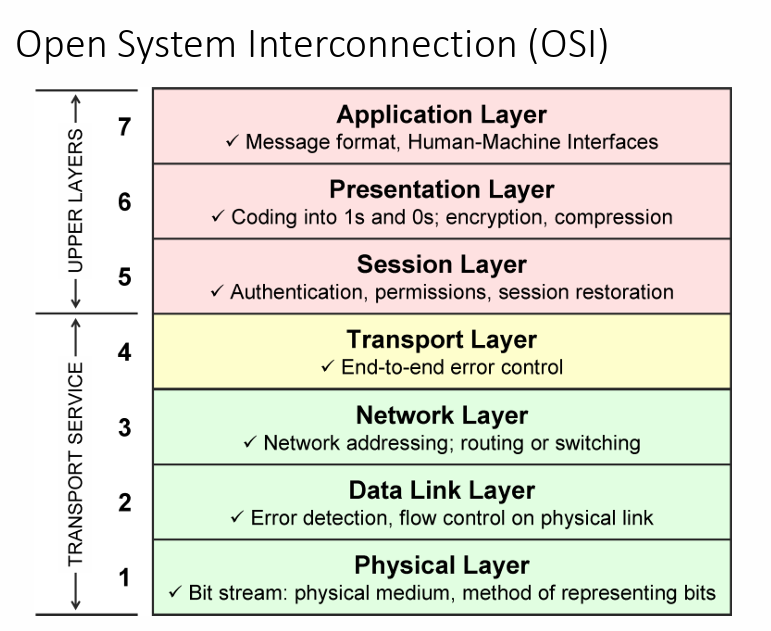
**Remember:** These principles are interconnected and should be considered holistically in any design process. By applying them thoughtfully, you can create solutions that are inclusive, user-friendly, and beneficial for everyone.

## **Cloud, Fog, and Edge Computing: A Tabular Comparison**

| Feature | Cloud Computing | Fog Computing | Edge Computing |
| --- | --- | --- | --- |
| **Location of Data Processing** | Centralized data centers | Network edge (gateways, local servers) | Devices themselves or small on-site devices |
| **Latency and Real-time Needs** | Suitable for tasks with some latency tolerance | Ideal for faster response times and lower latency | Best for real-time applications with critical latency requirements |
| **Processing Power and Storage** | High processing power and storage capacity | Moderate processing power and storage | Limited processing power and storage |
| **Security and Privacy** | Requires robust security measures due to centralized data storage | Offers more control over data location and processing | Potentially highest level of security and privacy control |
| **Cost** | Cost-effective for on-demand resources, but long-term costs can accumulate | Higher upfront costs due to additional edge infrastructure | May require specialized hardware and ongoing maintenance, but can reduce reliance on cloud resources |
| **Applications** | Big data analytics, enterprise applications, software development, content delivery networks | Smart cities, industrial automation, connected vehicles, video surveillance | Wearable devices, medical devices, drones, autonomous robots, remote sensor monitoring |
| **Advantages** | Scalability, high availability, cost-effectiveness for some scenarios | Reduced latency, improved privacy, local control | Real-time processing, low latency, reduced reliance on cloud |
| **Disadvantages** | High latency for real-time applications, security concerns | Higher upfront costs, increased complexity | Limited processing power and storage, potential security vulnerabilities |

**The OSI Model: 7 Layers of Network Communication**

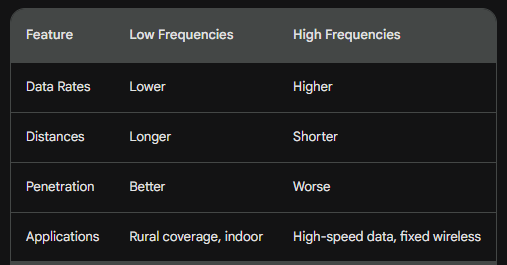
The Open Systems Interconnection (OSI) model is a conceptual framework that defines network communication into seven distinct layers. Each layer performs specific functions, allowing data to be transmitted efficiently and reliably between devices. Here's a breakdown of each layer:



**Layer 1: Physical Layer**

* Deals with the physical transmission of data bits over a network medium like cables or fiber optics.
* Handles tasks like converting digital data into electrical signals, managing signal strength, and ensuring physical connectivity.
* Examples of protocols: Ethernet, Token Ring, Wi-Fi.

**Frequencies for Mobile Communication**

****

**Layer 2: Data Link Layer**

* Responsible for reliable data transmission between network devices on the same physical segment.
* Detects and corrects errors introduced during transmission using error-checking codes.
* Addresses devices on the network and controls data flow.
* Examples of protocols: Ethernet, HDLC, PPP.

**Ethernet (802.3): The LAN Workhorse**

* **Widely used:** Indeed, Ethernet is the most popular LAN technology due to its simplicity, affordability, and reliability.
* **Bus architecture:** This means all devices share a single cable, similar to a bus route. Each device listens to the traffic and transmits only when the bus is free, using protocols like Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to avoid collisions.
* **Easy to install and manage:** Ethernet cables are readily available and relatively simple to connect. Setting up an Ethernet network typically involves plugging devices into a switch or router without complex configuration.
* **Packet-based communication:** Data is broken down into smaller units called packets, each containing addressing information, control data, and the actual payload. This enables efficient transmission and routing across the network.

**MAC Protocol:**

* The protocol within the Data Link Layer (Layer 2) of the OSI model that governs access to the shared medium in Ethernet networks.
* Defines how devices share the cable and avoid collisions during data transmission.

**CSMA/CD Protocol (Carrier Sense Multiple Access with Collision Detection):**

* The specific MAC protocol used in traditional Ethernet implementations (e.g., 10BASE-T, 100BASE-TX).
* Works as follows:
  + **Carrier Sense:**
    - The act of listening to the cable to detect the presence of a signal before transmitting.
    - Prevents multiple devices from transmitting at the same time, which would cause data corruption.
  + **Multiple Access:**
    - The ability of multiple devices to share the same physical medium for data transmission.
    - CSMA/CD allows for efficient utilization of the shared medium, but collisions can occur under high network load.
  + **Collision Detection:**
    - The ability of devices to detect collisions while transmitting data.
    - Allows devices to quickly abort transmission and re-try later, minimizing data loss.

**Wi-Fi (802.11):**

* A wireless LAN technology that allows devices to connect to a network without the need for cables.
* Uses the radio spectrum to transmit data between devices and access points.

**CSMA/CA Protocol (Carrier Sense Multiple Access with Collision Avoidance):**

* The MAC protocol used in Wi-Fi networks, designed to avoid collisions and improve efficiency compared to CSMA/CD in wired Ethernet.
* Works as follows:
  + **Carrier Sense:** Similar to Ethernet, devices listen to the channel before transmitting to check if it's busy.
  + **Multiple Access:** If the channel is free, devices can still potentially collide.
  + **Collision Avoidance:** Before transmitting data, devices send a short "request-to-send" (RTS) frame to the intended receiver.
  + **Channel Reservations:** If the receiver is available, it responds with a "clear-to-send" (CTS) frame, reserving the channel for the upcoming transmission.
  + **Transmission:** The data is then transmitted, and surrounding devices hearing the RTS/CTS frames stay quiet, preventing collisions.

**Channel Reservations:**

* The key difference from CSMA/CD is the use of RTS/CTS frames to reserve the channel before transmitting data.
* This reduces the chance of collisions, as surrounding devices know to avoid transmitting until the CTS frame is received.

**Advantages:**

* **Reduced collisions:** Channel reservations significantly decrease the likelihood of data loss due to collisions compared to CSMA/CD.
* **Improved efficiency:** By avoiding collisions, CSMA/CA allows for more efficient use of the shared wireless medium.
* **Lower power consumption:** Collisions can drain battery life on mobile devices. By avoiding them, CSMA/CA can improve battery life.

**Additional points:**

* While CSMA/CA offers advantages, it can still introduce overhead with RTS/CTS frames, especially in low-traffic situations.

**Layer 3: Network Layer**

* Handles routing data packets across different network segments, regardless of the physical topology.
* Determines the best path for data packets to reach their destination, considering factors like network congestion and available bandwidth.
* Assigns logical addresses (IP addresses) to devices.
* Examples of protocols: IP, ICMP, BGP.

## 

## 

## An IP address, short for Internet Protocol address, is essentially a unique identifier assigned to each device connected to the internet. It acts like a digital address, allowing other devices to locate and communicate with it.

## **Key Differences Between IPv4 and IPv6**

IPv4 and IPv6 are the two main versions of the Internet Protocol (IP), the core addressing system for devices on the internet. Here are some key differences between them:

Address Space:

* IPv4: Uses 32-bit addresses, allowing for approximately 4.3 billion unique addresses. This limited space is becoming increasingly scarce.
* IPv6: Uses 128-bit addresses, offering a virtually infinite number of addresses (2^128).

Address Format:

* IPv4: Uses decimal numbers separated by dots (e.g., 192.168.1.1).
* IPv6: Uses eight groups of hexadecimal digits separated by colons (e.g., 2001:db8:85a3:0000:0000:8a2e:0370:7334).

Security:

* IPv4: Has no built-in security features. Encryption and authentication are typically implemented at higher layers.
* IPv6: Includes built-in security features for data encryption and authentication, offering improved security.

Header Structure:

* IPv4: Header is more complex, containing fields for options and checksums.
* IPv6: Header is simpler and smaller, improving processing efficiency.

Other Differences:

* Mobility: IPv6 is better suited for mobile networks due to its simpler autoconfiguration features.
* QoS: IPv6 offers better support for Quality of Service (QoS), allowing for prioritization of different types of traffic.
* Fragmentation: IPv4 allows packets to be fragmented for transmission on smaller networks, while IPv6 generally avoids fragmentation.

**Layer 4: Transport Layer**

* Provides reliable and ordered data delivery between applications on different devices.
* Segments data into packets for efficient transmission and reassembles them at the receiving end.
* Offers flow control and error correction mechanisms.
* Examples of protocols: TCP, UDP.

**Key Differences between TCP and UDP:**

| Feature | UDP | TCP |
| --- | --- | --- |
| **Purpose** | Fast, efficient data transfer | Reliable, ordered data transfer |
| **Reliability** | Unreliable, packets can be lost or arrive out of order | Reliable, guarantees delivery and order of packets |
| **Header** | Simpler, smaller (8 bytes) | More complex, larger (20 bytes) |
| **Ports** | Uses ports to identify applications | Uses ports and sequence numbers for tracking data flow |
| **Error Correction** | No built-in error correction | Uses checksums and acknowledgments to verify data integrity |
| **Flow Control** | No congestion control | Uses flow control and congestion control mechanisms |
| **Applications** | Streaming media, online gaming, DNS lookups | Web browsing, file transfers, email |

**Additional Points:**

* **Speed:** UDP is generally faster than TCP due to its simpler structure and lack of error correction overhead. However, this speed comes at the cost of reliability.
* **Complexity:** TCP adds more complexity to data transmission due to its sequencing, acknowledgments, and flow control mechanisms.
* **Applications:** Choose UDP when speed is crucial and minor data loss is acceptable (e.g., streaming media). Choose TCP when data integrity and order are essential (e.g., downloading files).

**Example:** Imagine downloading a large file. You'd want to use TCP to ensure all data arrives in the correct order and without errors. However, for watching a live stream, UDP might be preferable due to its lower latency, even if some frames are lost or arrive slightly out of order.

**Layer 5: Session Layer**

* Establishes, manages, and terminates communication sessions between applications.
* Handles tasks like authentication, authorization, and synchronization.
* Examples of protocols: RPC, NetBIOS.

**Layer 6: Presentation Layer**

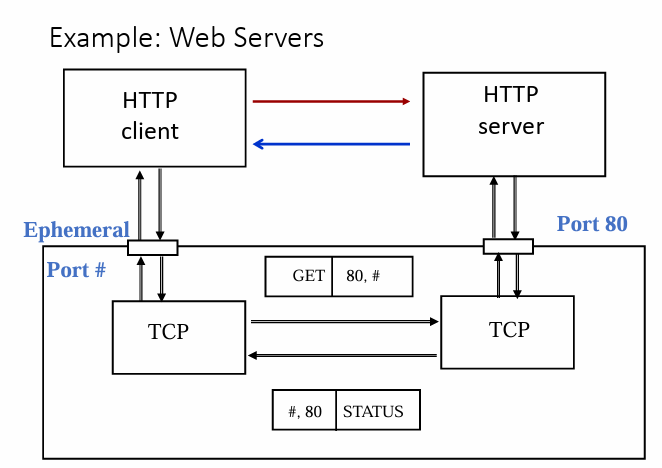
* Deals with data format and encryption/decryption.
* Translates data between different formats used by applications on different devices.
* Ensures compatible data representation across platforms.
* Examples of protocols: ASCII, EBCDIC, JPEG, MPEG.

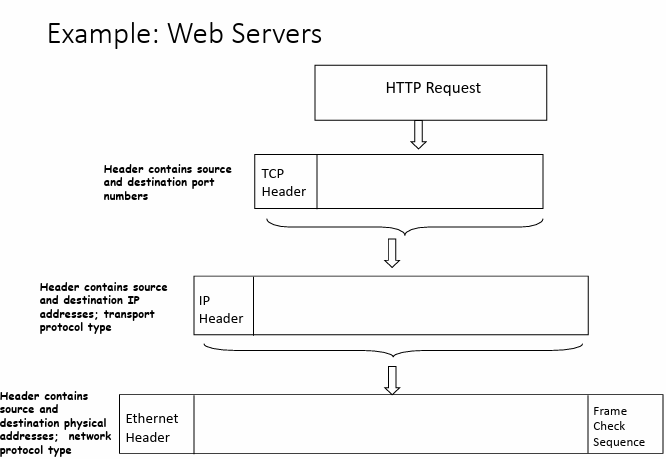
**Layer 7: Application Layer**

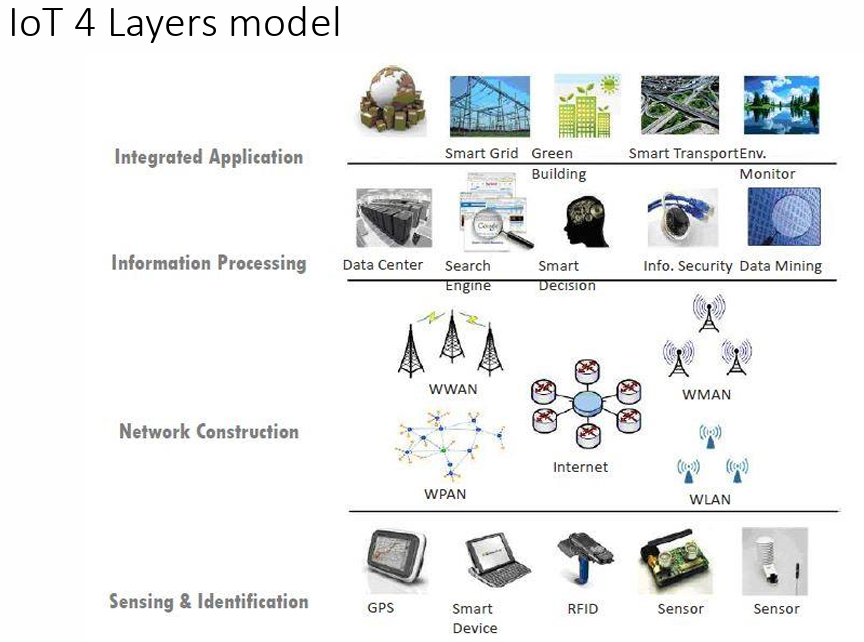
* Provides network services to user applications like email, file transfer, web browsing.
* Defines protocols specific to different applications.
* Examples of protocols: HTTP, FTP, SMTP.

**Importance of the OSI Model:**

* Provides a common reference point for understanding network communication.
* Helps in troubleshooting network issues by isolating problems to specific layers.
* Facilitates the development of interoperable network devices and protocols.







The IoT architecture can be visualized using a 4-layer model, each layer with its specific functionalities:

**1. Device Layer (Sensor Layer):**

* This layer consists of the physical devices that interact with the physical world.
* These devices include sensors, actuators, and gateways.
* Sensors collect data from the environment, such as temperature, pressure, humidity, or motion.
* Actuators control physical devices based on received data or commands, such as turning on lights or adjusting thermostats.
* Gateways aggregate data from multiple devices and provide connectivity to the network layer.

**2. Network Layer:**

* This layer is responsible for transmitting data between the device layer and the application layer.
* It involves communication protocols, network security, and data routing.
* Various network technologies can be used, including cellular networks, Wi-Fi, Bluetooth, and Low-Power Wide-Area Networks (LPWANs).

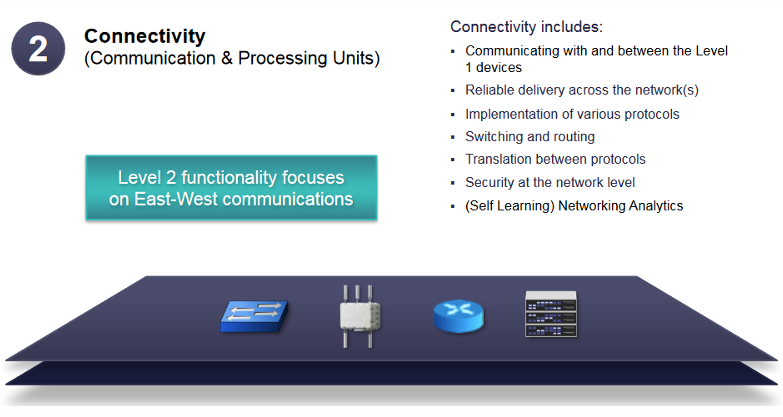
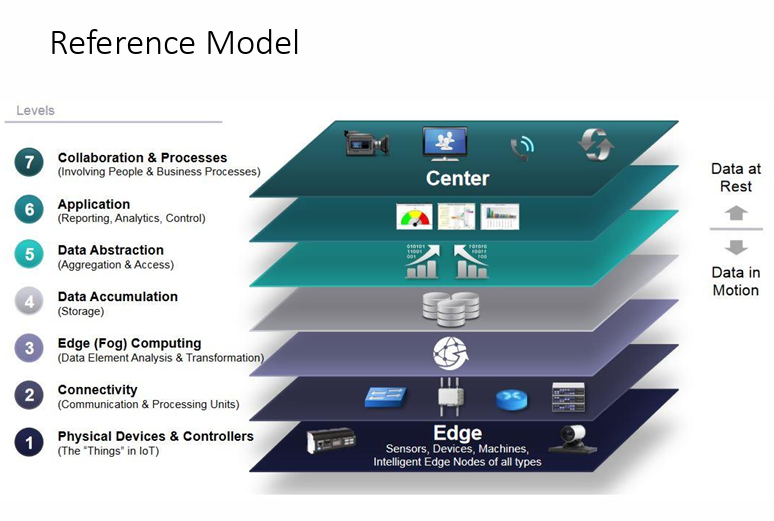
**3. Processing Layer:**

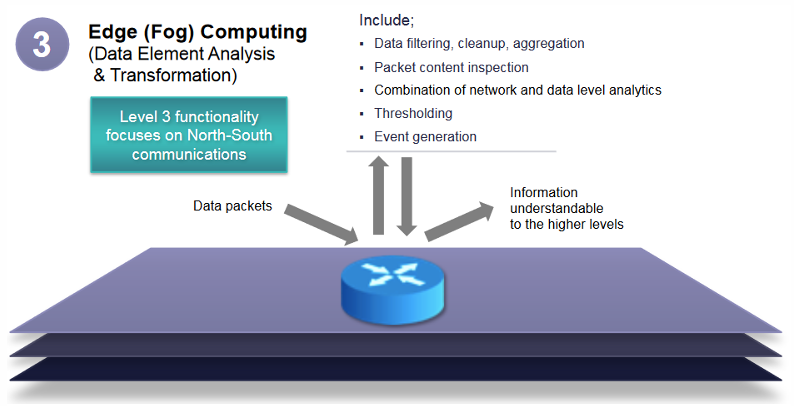
* This layer processes, stores, and analyzes the data collected from the device layer.
* It may involve filtering, aggregation, and transformation of data for further analysis.
* This layer might also employ cloud computing platforms for data storage and processing.

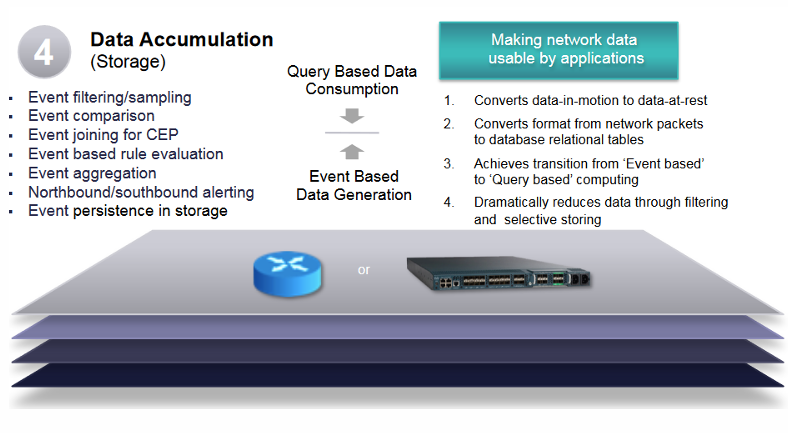
**4. Application Layer:**

* This layer provides the user interface and functionalities for interacting with the collected data and controlling devices.
* It can include mobile apps, web dashboards, and other software applications.
* This layer analyzes the data to gain insights, make decisions, and automate tasks based on the collected information.

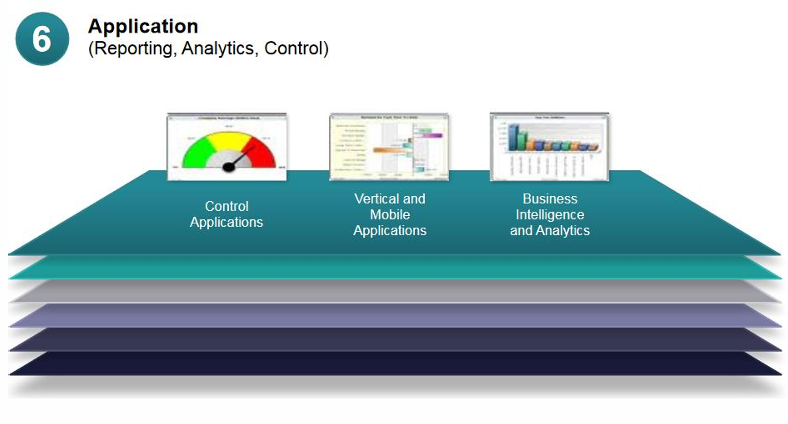
**Overall, the 4-layer model provides a structured way to understand the complex ecosystem of the IoT. Each layer plays a crucial role in enabling data collection, transmission, processing, and utilization, ultimately leading to valuable insights and automated actions in the physical world.**

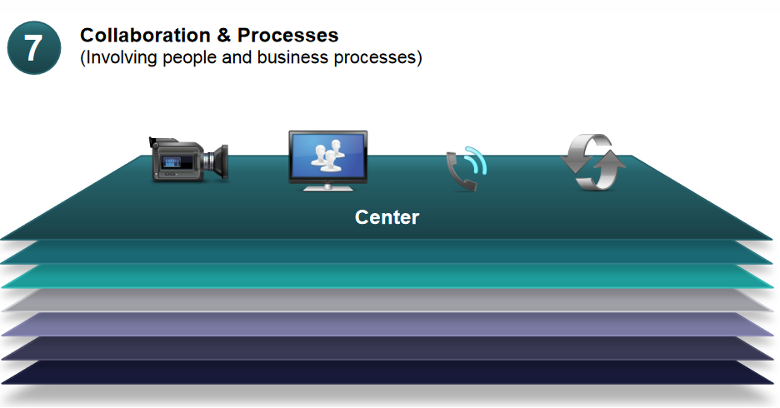


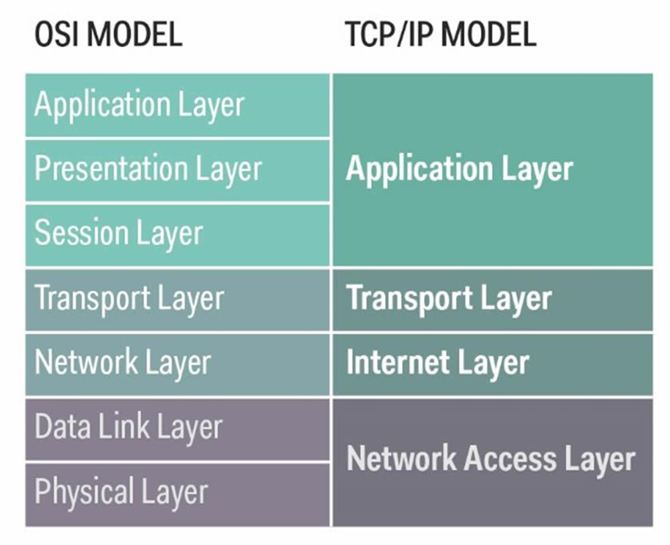


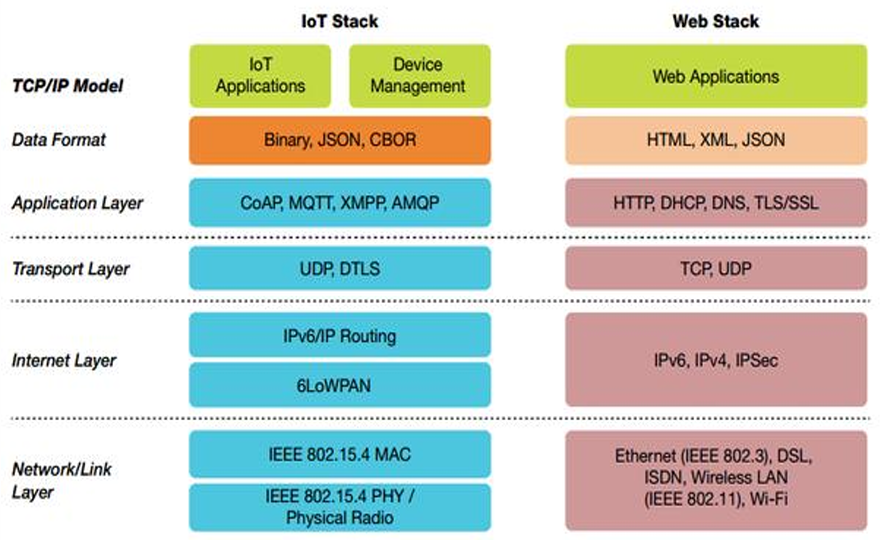


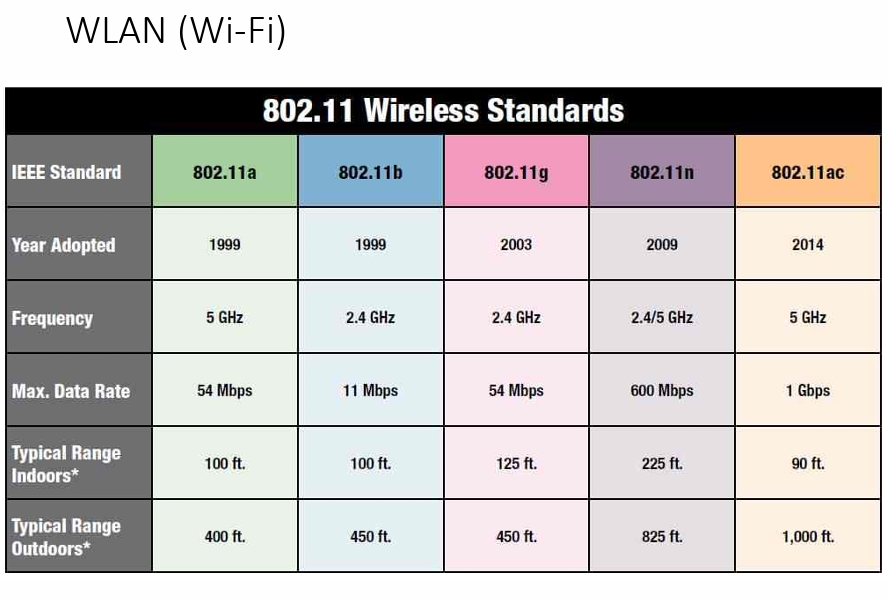












## **802.11 Infrastructure Network Architecture: A Deeper Dive**

The 802.11 standard defines the architecture for wireless local area networks (WLANs) based on the infrastructure model. Here's a breakdown of the key components and their roles:

**Station (STA):**

* Represents any device with wireless capabilities, such as laptops, smartphones, and smart TVs.
* STAs have access mechanisms to the wireless medium and can communicate with Access Points (APs) using radio waves.
* They rely on APs for network access and utilize protocols defined in the 802.11 standard for communication.

**Basic Service Set (BSS):**

* A group of STAs and an AP that share the same radio frequency channel and form a logical network segment.
* Communication within a BSS happens directly between STAs and the AP.
* One AP can only manage a single BSS, although multiple BSSes can co-exist in the same physical space using different channels.

**Access Point (AP):**

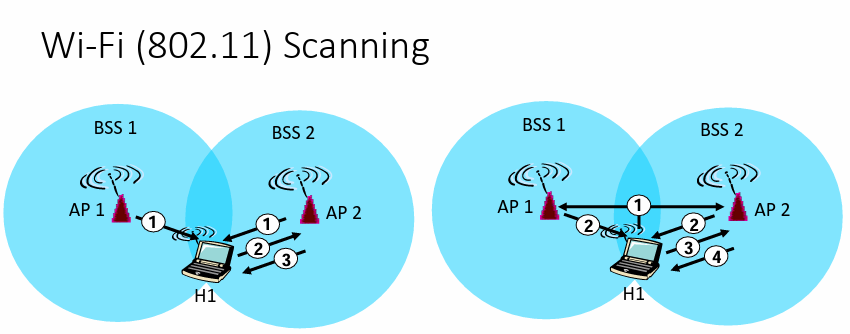
* Acts as a central hub for a BSS, providing wireless connectivity and network access to STAs.
* It connects to the wired network (e.g., Ethernet) and acts as a bridge between wireless and wired devices.
* APs manage communication within the BSS, regulating access to the shared wireless medium and forwarding data between STAs and the wired network.

**Portal:**

* Not explicitly defined in the 802.11 standard but often used interchangeably with AP.
* Refers specifically to the function of an AP as a gateway or bridge between the wireless BSS and other wired networks.
* Portals provide internet connectivity and access to network resources for STAs within the BSS.

**Distribution System (DS):**

* Connects multiple BSSes to form a larger logical network called an Extended Service Set (ESS).
* DS can be a wired network (e.g., Ethernet) or a wireless network using a different frequency band for inter-BSS communication.
* Allows roaming between BSSes within the ESS, enabling seamless connectivity for STAs as they move around.



Wi-Fi scanning refers to the process of a device searching for available wireless networks in its vicinity. This process can be carried out in two main ways: passive scanning and active scanning.

**Passive Scanning:**

1. **Beacons sent from APs:** Access Points (APs) periodically broadcast beacons containing information about their network, including the SSID, security settings, and signal strength.
2. **Association Request sent from H1 to selected AP:** When a device (H1) wants to connect to a network, it listens for beacons and chooses the desired one. It then sends an Association Request frame to the chosen AP.
3. **Association Response sent from AP to H1:** If the AP is willing to accept the connection, it sends an Association Response frame confirming the connection and providing further details.

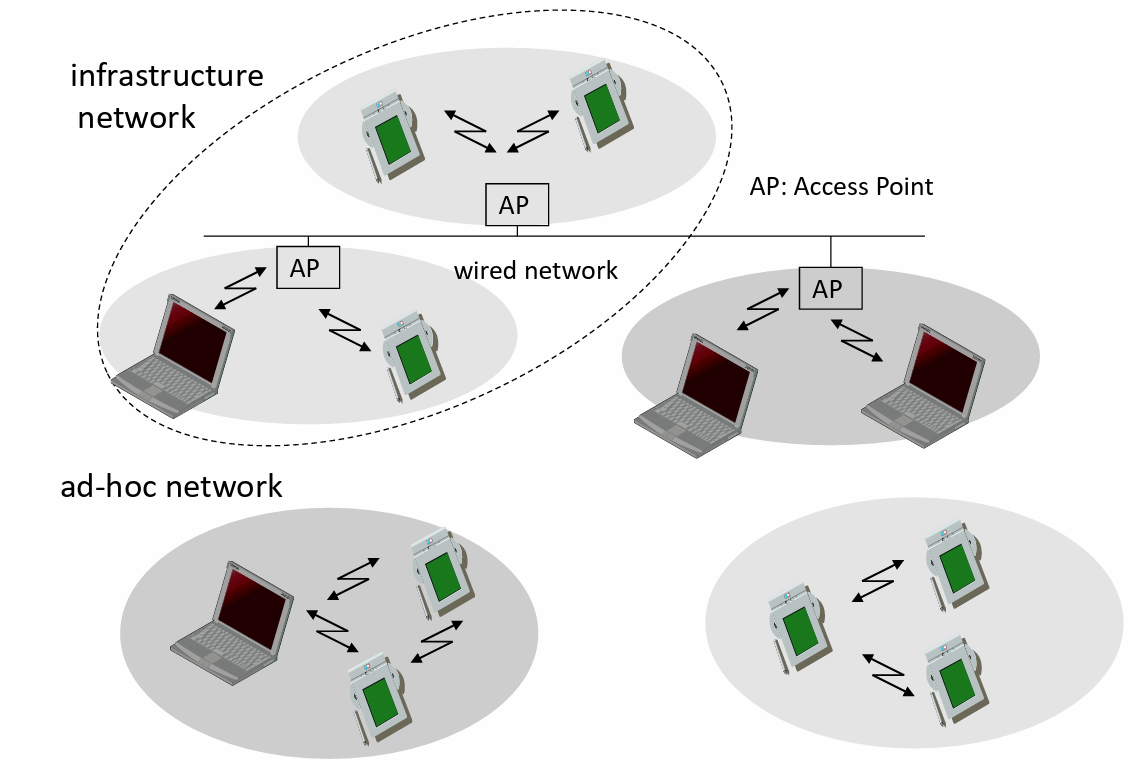
**Active Scanning:**

1. **Probe Request (broadcast) sent from H1:** The device (H1) sends a Probe Request frame, which is a broadcast message asking for information about nearby networks.
2. **Probe Response sent from APs:** Any APs within range that receive the Probe Request respond with a Probe Response frame, containing their network information like SSID, security settings, and signal strength.
3. **Association Request sent from H1 to selected AP:** Based on the received Probe Responses, H1 chooses a suitable network and sends an Association Request to the desired AP.
4. **Association Response sent from AP to H1:** Similar to passive scanning, the AP responds with an Association Response frame if it agrees to the connection.

In some cases, devices might combine both passive and active scanning for faster discovery and improved accuracy.

## **Infrastructure vs. Ad-Hoc Networks: Key Differences**

Both infrastructure and ad-hoc networks are types of wireless networks, but they differ in their structure, security, performance, and use cases. Here's a breakdown of their key differences:



**Infrastructure Network:**

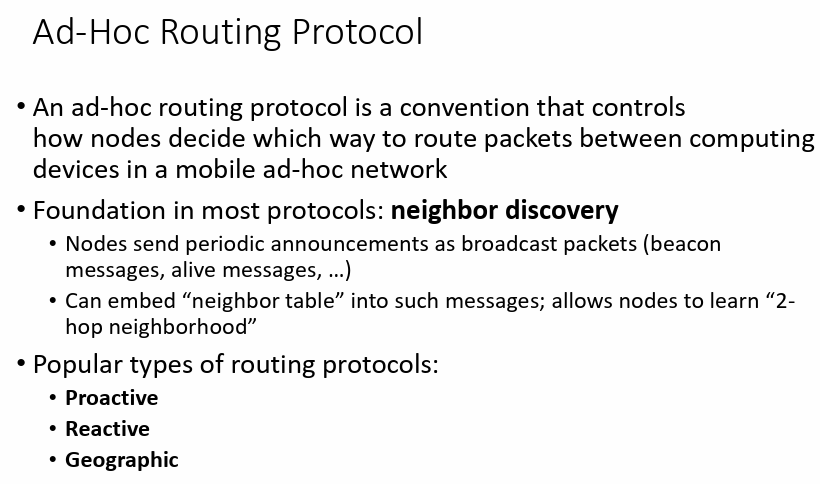
* **Structure:** Relies on a central device called an access point (AP). Devices connect to the AP to access the network and the internet.
* **Security:** Generally more secure with central authentication and encryption mechanisms.
* **Performance:** Typically faster and more stable due to dedicated bandwidth and centralized management.
* **Scalability:** More scalable as you can add more APs to cover larger areas and accommodate more devices.
* **Use cases:** Ideal for homes, offices, public spaces, and any situation where centralized control, security, and performance are important.

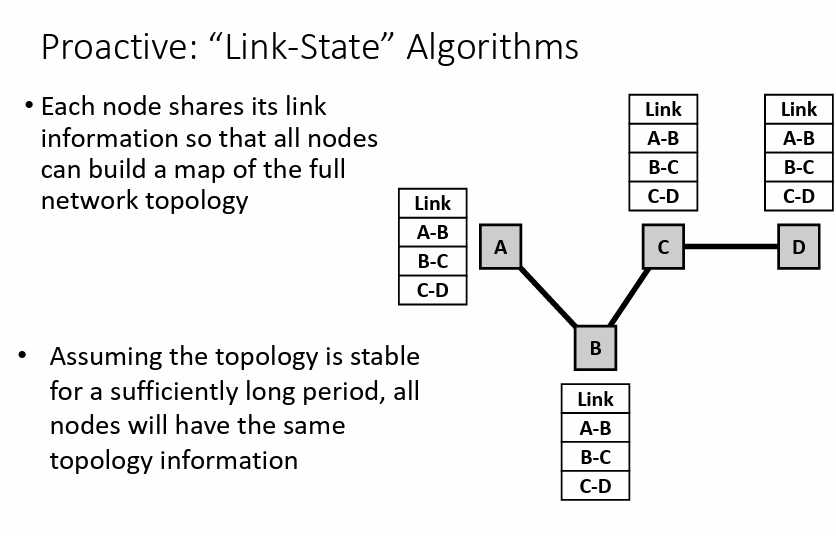
**Ad-Hoc(for this purpose) Network:**

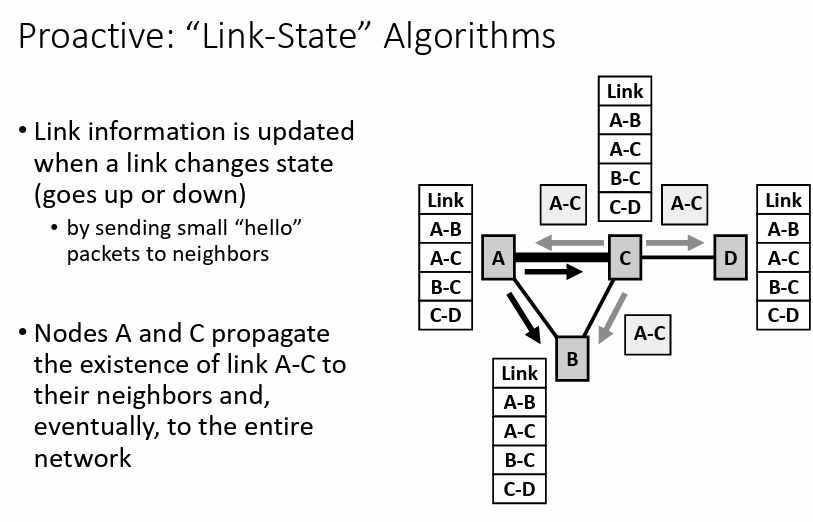
* **Structure:** Devices connect directly with each other without an AP.
* **Security:** Less secure as there's no central authentication and encryption.
* **Performance:** Can be slower and less stable due to peer-to-peer communication and limited bandwidth.
* **Scalability:** Limited scalability as the network size is restricted by the range of individual devices.
* **Use cases:** Good for temporary connections, file sharing among a small group of devices, or situations where centralized infrastructure is unavailable.
* • MANET: Mobile Ad-Hoc Network

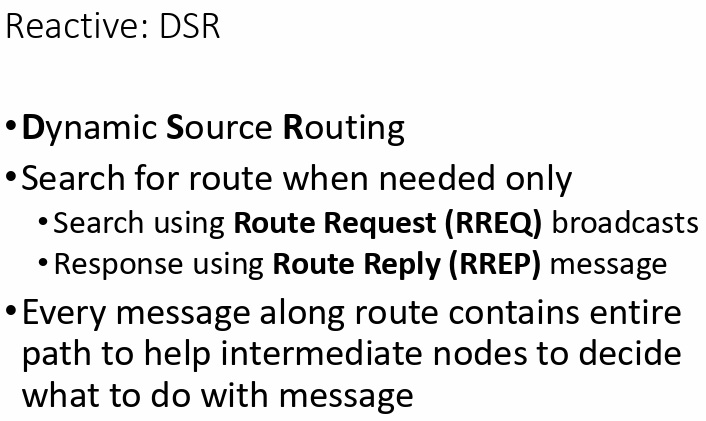
**Additional Differences:**

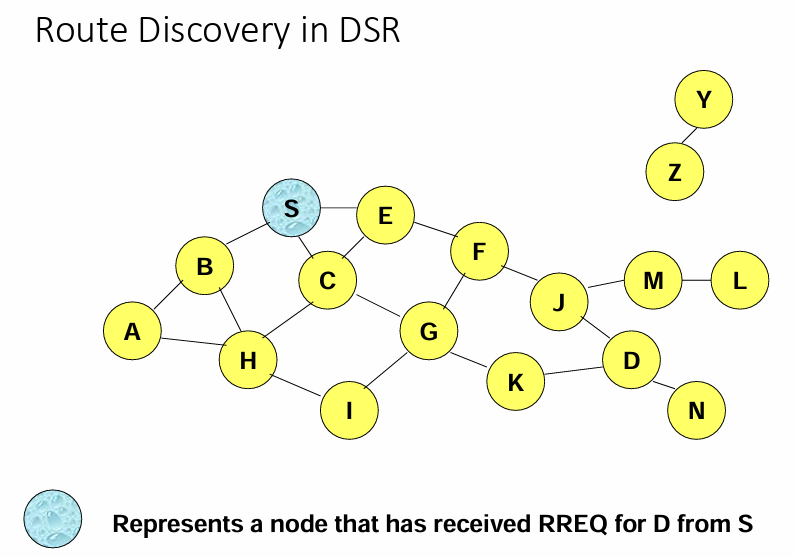
* **Configuration:** Setting up an ad-hoc network is generally simpler, while infrastructure networks require configuring the AP.
* **Power consumption:** Ad-hoc networks may use less power as they don't need an AP.
* **Mobility:** Easier to move ad-hoc networks as they don't depend on a fixed AP location.

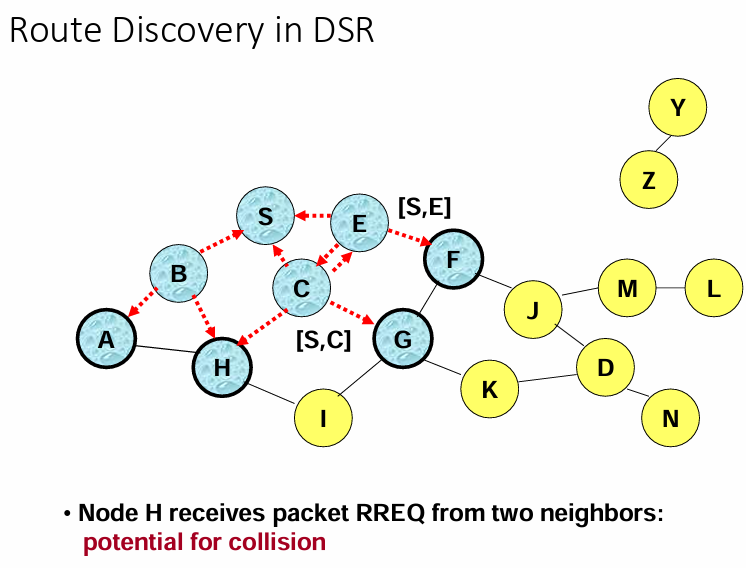


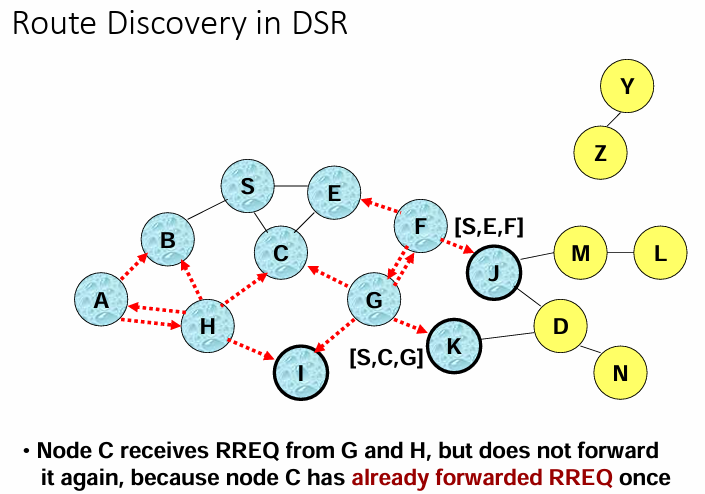


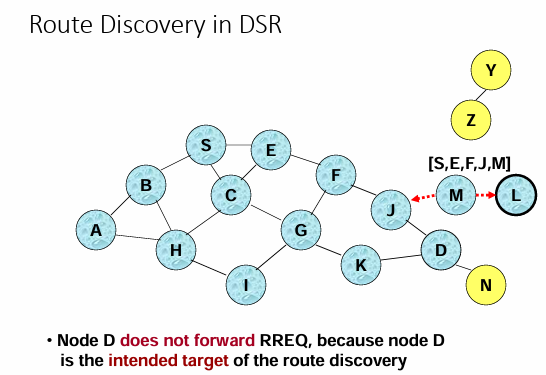
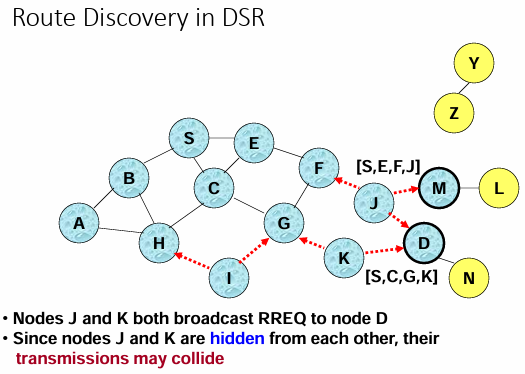


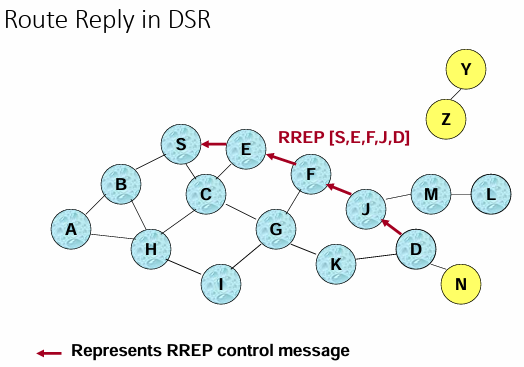


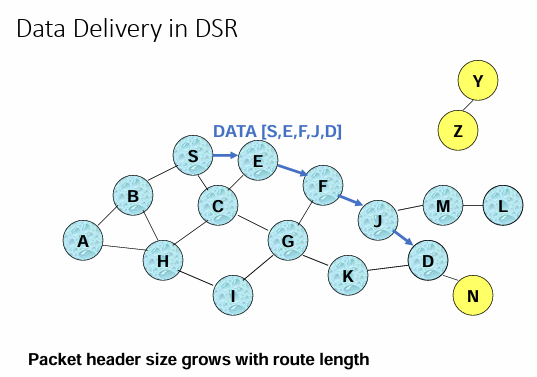












## **Reactive vs. Proactive: Routing Strategies in Ad-hoc Networks**

Imagine you need to send a message to a friend in a crowded park. How do you find them?

**Reactive:** You ask nearby people if they've seen your friend (flooding), and keep asking until someone points you in the right direction. This is like **reactive routing**. It's efficient for short trips, but asking everyone can be noisy and slow.

**Proactive:** You have a map of the park (route information) and know exactly how to reach your friend. This is like **proactive routing**. It's fast and reliable, but your map might be outdated if your friend moved since you last saw it.

| Feature | Reactive | Proactive |
| --- | --- | --- |
| Route discovery | On-demand (when needed) | Always known |
| Message size | Larger (includes entire route) | Smaller |
| Network traffic | High (due to flooding) | Lower |
| Route information | May be outdated | Up-to-date (ideally) |
| Suitable for | Short, dynamic networks | Stable networks with frequent updates |

**Geographic routing in ad-hoc networks:**.

**Key points:**

* Devices use their own location and the destination's location to choose the next hop for a message.
* They might need to know the location of nearby devices (neighbors) to choose the best path.
* This location information can be obtained from GPS, other positioning systems, or even shared within the network.

**Pros and cons:**

* **Pros**: Fast and efficient, especially in networks where devices move around frequently.
* **Cons**: Requires accurate location information, which might not always be available. May not work well in complex environments with obstacles.

**Think of it like this:** Geographic routing is like using a map with landmarks to navigate, while other routing methods rely on pre-defined paths or asking for directions. It's a powerful tool for dynamic networks, but its effectiveness depends on the specific environment and available information.

## **Unicast Location-Based Routing**

Imagine you're trying to deliver a message to a specific friend in a bustling city, but you only have a map with landmarks and no street names. That's essentially how **unicast location-based routing** works in ad-hoc networks.

**The key idea:**

* Each device uses its location and knowledge of its neighbors (nearby devices) to decide the best next hop for the message.
* It basically points the message in the "right direction" based on the destination's location, hoping it gets closer with each hop. This is called **greedy forwarding**.

**Benefits:**

* Simple and efficient, especially when devices move around frequently.
* No need for complex route calculations or map memorization.

**The challenge: getting stuck:**

Sometimes, the message might reach a point (a "void" or "hole") where none of the neighbors are closer to the destination. It's like ending up in a dead-end alley!

**How to overcome it:**

* Some protocols use additional mechanisms like perimeter routing or location-aided flooding to escape these "voids" and find a new path.
* Other approaches involve cooperation among devices, where they share information to create a more complete picture of the network and avoid dead ends.

## **Geocasting in ad-hoc networks**

Imagine you're in a forest and need to alert all researchers within a specific area about a rare bird sighting. Geocasting is like shouting out your message, but only within that specific part of the forest.

**The key idea:**

* Send messages to all devices (or a subset) within a defined geographic region, like a circle or a polygon.
* This is useful for situations where you need to reach multiple devices in a particular area, like sending emergency alerts or sharing local sensor data.

**Think of it like this:**

* Geocasting combines two approaches:
  + **Unicast-like**: Initially, it gets the message "close" to the target region, like pointing people in the right direction.
  + **Flooding-like**: Once within the region, it forwards the message to all devices there, like shouting out loud.

**The challenge:**

* Striking a balance: How to get the message to the target region efficiently (like unicast) while also ensuring it reaches everyone within that region (like flooding)?

**Solutions:**

* Some protocols use special forwarding techniques based on location information to optimize distribution within the region.
* Others leverage the existing unicast infrastructure to reach the region and then rely on local flooding for wider spread.

**UART:** Simple, low-cost, slow, asynchronous, 2 wires. Ideal for basic tasks.

**SPI:** Faster, full-duplex, master-slave, 4 wires. Great for high-speed data transfer.

**I2C:** Low-power, multi-master, addressing, 2 wires. Efficient for multiple devices.

**GPIO:** Flexible, user-controlled, software-based communication. Useful for custom hardware and low-level control.