1. Based on Communication Medium

- **Wireless Sensor Networks (WSN)**:
    - Sensors communicate wirelessly using protocols such as Zigbee, LoRa, Wi-Fi, or Bluetooth.
    - These networks offer flexibility in deployment, are easier to install, and are ideal for remote or hard-to-access areas.
    - **Example**: A forest fire detection system where sensors are spread throughout a forest to monitor temperature, humidity, and smoke. They relay data to a central server via wireless protocols, providing early warnings of potential fires.
- **Wired Sensor Networks**:
    - Sensors are connected through physical cables, such as Ethernet or fiber optics, providing high-speed, reliable communication.
    - Often used in environments where interference or data security is a concern, and network stability is essential.
    - **Example**: Industrial machinery monitoring systems in factories, where sensors detect vibration, temperature, and operational status to prevent equipment failure. Wired connections ensure stable data transmission in noisy industrial settings.

2. Based on Deployment Environment

- **Terrestrial Sensor Networks**:
    - Deployed on land for applications like environmental monitoring, agriculture, and defense.
    - They can include fixed sensors in cities, fields, or forests, depending on the target environment.
    - **Example**: Weather stations with temperature, humidity, and air pressure sensors monitor climate changes in various geographic locations.
- **Underground Sensor Networks**:
    - Sensors are buried underground to collect data on soil moisture, seismic activity, and mineral deposits.
    - Used in applications like agriculture, mining, and earthquake prediction.
    - **Example**: Soil moisture monitoring for agriculture, where sensors provide data to optimize irrigation, ensuring crops receive adequate water without waste.
- **Underwater Sensor Networks (UWSN)**:
    - Deployed underwater to monitor aquatic environments or marine life.
    - Designed to withstand water pressure and communicate using acoustic waves, which are more effective underwater.
    - **Example**: Marine life tracking systems that monitor fish migration patterns and ocean conditions to support conservation efforts.
- **Aerial Sensor Networks**:

    - Deployed in the air, often via drones or aircraft, to cover large, hard-to-reach areas.
    - Suitable for disaster response, agricultural health monitoring, and surveillance.
    - **Example**: Drone-based crop health monitoring that uses sensors to assess plant health, soil conditions, and pest presence over large agricultural fields.

3. Based on Mobility

- **Static Sensor Networks**:
    - Sensors remain in fixed locations and continuously monitor specific conditions.
    - Common in stable, predictable environments where coverage consistency is essential.
    - **Example**: Weather monitoring stations in cities or rural areas that measure weather parameters consistently.
- **Mobile Sensor Networks**:
    - Sensors are attached to moving objects like vehicles, drones, or animals.
    - Useful in dynamic environments where sensor mobility can enhance data collection and coverage.
    - **Example**: Mobile healthcare sensors that are attached to patients, allowing doctors to remotely monitor heart rate, oxygen levels, and body temperature.

4. Based on Network Topology

- **Single-Hop Networks**:
    - All sensors communicate directly with a central base station or hub, often in small-scale or local-area networks.
    - This approach reduces latency but limits network range.
    - **Example**: Home automation systems where all devices, like thermostats, lights, and cameras, connect directly to a central hub.
- **Multi-Hop Networks**:
    - Data is relayed through intermediate nodes to reach the destination, ideal for larger networks covering vast areas.
    - This structure conserves power by allowing nodes to transmit over shorter distances.
    - **Example**: Environmental monitoring networks covering large forests, where sensor data on temperature, humidity, and soil composition is passed from one sensor to the next until it reaches the base station.

5. Based on Sensor Type

- **Temperature Sensors**: Measure ambient temperature, commonly used in weather monitoring, HVAC systems, and healthcare.
  - **Example**: In greenhouses, temperature sensors help maintain optimal growing conditions for plants.
- **Humidity Sensors**: Detect moisture levels in the air, used in agriculture, climate control, and HVAC systems.
  - **Example**: Hygrometers in agricultural fields help monitor humidity to optimize crop growth.
- **Pressure Sensors**: Measure variations in atmospheric or fluid pressure, applied in aviation, meteorology, and industrial equipment.
  - **Example**: Barometers in weather stations detect changes in atmospheric pressure to predict weather patterns.
- **Proximity Sensors**: Detect the presence or absence of nearby objects, useful in automation, robotics, and security.
  - **Example**: Proximity sensors in smartphones detect the user's face during calls to prevent accidental touches.
- **Chemical Sensors**: Identify specific chemicals or gas concentrations, essential in pollution monitoring, medical diagnostics, and industrial safety.
  - **Example**: CO2 sensors in indoor air quality systems maintain safe and comfortable levels of carbon dioxide.
- **Optical Sensors**: Measure light intensity or detect infrared radiation, used in environmental studies, health monitoring, and consumer electronics.
  - **Example**: Light sensors in solar panels track sunlight intensity to maximize energy generation.

6. Based on Power Source

- **Battery-Powered Sensor Networks**:
  - Rely on batteries, so energy efficiency is crucial to extend the network’s lifetime.
  - Used in applications where energy harvesting is not feasible.
  - **Example**: Wearable fitness trackers that monitor steps, heart rate, and sleep rely on efficient battery use.
- **Energy-Harvesting Sensor Networks**:
  - Gather energy from renewable sources like solar, thermal, or kinetic energy.
  - Ideal for remote or long-term monitoring applications.
  - **Example**: Solar-powered environmental sensors in remote forests monitor wildlife without the need for battery replacement.

7. Based on Coverage Area

- **Local Sensor Networks**:
    - Cover smaller areas such as homes, offices, or small facilities.
    - Often consist of a limited number of sensors with straightforward communication needs.
    - **Example**: Smart home systems where sensors monitor lighting, temperature, and security in a residential setting.
- **Wide-Area Sensor Networks**:
    - Cover large geographical regions such as cities, forests, or entire ecosystems.
    - Suitable for applications like environmental conservation, disaster management, and urban planning.
    - **Example**: City-wide air pollution monitoring networks that measure air quality across multiple locations for public health analysis.

8. Based on Functionality

- **Event Detection Networks**:
    - Primarily designed to detect and alert on specific events, such as fires, floods, or intrusions.
    - These networks are usually dormant and activate only when an event occurs, conserving energy.
    - **Example**: Fire alarm systems in commercial buildings detect smoke or heat and alert occupants to evacuate.
- **Data Gathering Networks**:
    - Continuously monitor and collect data, which is then analyzed to identify trends or patterns.
    - Often used in research or long-term monitoring applications where consistent data collection is crucial.
    - **Example**: Environmental monitoring stations that continuously record temperature, humidity, and air quality data for climate research.

1 and 2

Running TCP (Transmission Control Protocol) over ad hoc networks—networks without fixed infrastructure where devices connect directly to each other—has challenges because TCP was originally designed for stable, wired networks, not for the dynamic, changing environment of ad hoc networks. Here’s a simplified look at the issues and possible solutions:

Key Challenges

1. **Changing Connections or Route Breaks and Mobility**: In ad hoc networks, devices (nodes) often move around, which causes breaks in the connection paths. TCP was built to interpret missing data as "congestion" (too much data being sent), so it slows down transmission. However, in an ad hoc network, missing data might just mean the connection path is broken, not congested.
2. **Interference or Hidden and Exposed Terminal Problems**: When several nodes try to send data at the same time, their signals can interfere with each other, causing data collisions (like cars at a busy intersection). TCP can misunderstand these collisions as congestion, reducing performance unnecessarily.
3. **Variable Speeds or bandwidth**: The wireless links between nodes in an ad hoc network can vary in speed and quality. TCP expects a steady connection, so these changes can lead to slower or lost data transfers.
4. **Multi-hop Relays**: Data often has to "hop" through several nodes to reach its destination, creating longer delays and increasing the chance of data getting lost along the way.

(or)

. Changing Connections or Route Breaks and Mobility

- In ad hoc networks, devices (or "nodes") are often moving, which means the connection paths between them can break and change frequently.
- When a path breaks, TCP thinks that data loss is happening because of network congestion (too much data being sent). TCP then slows down, assuming it needs to ease congestion.
- However, in an ad hoc network, data loss may just be due to broken paths—not congestion—so TCP's slowdown response is unnecessary.

2. Interference or Hidden and Exposed Terminal Problems

- In ad hoc networks, several nodes might try to send data at the same time. This causes interference, where signals "collide" like cars crashing at a busy intersection.

- TCP sees these data collisions and misinterprets them as a sign of congestion, even though it’s just interference. It slows down data transfer, which lowers network performance unnecessarily.

3. Variable Speeds or Bandwidth

- Wireless links in ad hoc networks can change in speed and quality. For example, when a node moves farther away from another, the signal might weaken, causing slower data transfer.
- TCP was designed for steady, consistent connections, so it doesn’t handle these speed changes well. As a result, TCP might see the slower transfer as a problem and slow down even more, leading to data delays.

4. Multi-hop Relays

- In ad hoc networks, data often has to "hop" through multiple nodes to reach its destination, which is like making several stops on a journey.
- Each hop adds delay and increases the chance of data getting lost. TCP isn’t used to these multiple stops and can become slow or inefficient in transferring data across several hops.

Solutions and Modifications

To help TCP work better in ad hoc networks, several tweaks have been proposed:

1. Feedback-based TCP (TCP-F)

- TCP-F allows devices in the network to tell the sender when a route breaks.
- Instead of assuming there's congestion and slowing down, TCP just pauses, waiting for a new route to be set up. This avoids unnecessary delays or resending of data.

2. Ad Hoc TCP (ATCP)

- ATCP is like a helper layer that sits between TCP and the network. It watches for issues and helps TCP figure out if there’s true congestion (too much data being sent) or if it’s just a broken route.
- This way, TCP only slows down when there’s actual congestion, making the connection more reliable.

3. Split TCP

- This solution breaks up the long journey of data into smaller parts or "segments." An intermediate device, or helper node, takes charge of each segment separately.
- By managing each segment individually, any issues that happen in one part of the journey don’t affect the entire connection, making it easier to handle interruptions.

4. Explicit Congestion Notification (ECN)

- ECN allows network devices to directly inform TCP when they sense congestion is likely to happen.
- This way, TCP slows down only when there's real congestion instead of slowing down for other problems, making it more efficient.

5. Adaptive Congestion Window

- Normally, TCP controls its sending speed with a "congestion window" (the amount of data it sends at once). An adaptive congestion window adjusts this window size based on current network conditions.
- This means TCP can adapt its speed better to changing network quality, like fluctuating speeds or interference.

6. Cross-Layer Solutions

- In ad hoc networks, it can be helpful for different parts of the network (like TCP and routing) to share information.
- For example, if the routing layer detects a path change, it can inform TCP, allowing TCP to adjust its behavior, like pausing until the new path is ready.

Summary

These tweaks let TCP handle route breaks, interference, and speed changes more effectively in ad hoc networks. By using feedback, segmenting data paths, notifying about real congestion, and adapting speed, TCP can perform better in these dynamic environments.

Alternative Protocols

Since TCP isn't perfect for ad hoc networks, researchers have looked into alternatives:

- **AODV-TP**: This protocol combines transport and routing functions to handle path breaks and packet loss better, specifically for ad hoc networks.
- **SCTP (Stream Control Transmission Protocol)**: SCTP can manage multiple paths for data and handle interruptions better, though it's not designed specifically for ad hoc networks.