# CHAPTER 4

# CONGESTION CONTROL AND FRAGMENTATION IN IOT

## 4.1 IoT - network of networks

IoT offers a high level of value in its ability to collect, distinguish, analyze, and distribute data that represent messages, information, knowledge and generally wisdom. In this contest, IoT becomes immensely important. For example, vehicles with multiple networks can benefit to control all functions like engine function, communications systems, safety features and so on. Similarly Commercial and residential buildings can have various types of control systems for the purpose of heating, venting and air conditioning (HVAC), security service, telephone connection and lighting the environment. The different set of network connected together with added function of analytics, management and security capabilities is shown in Figure 4.1.

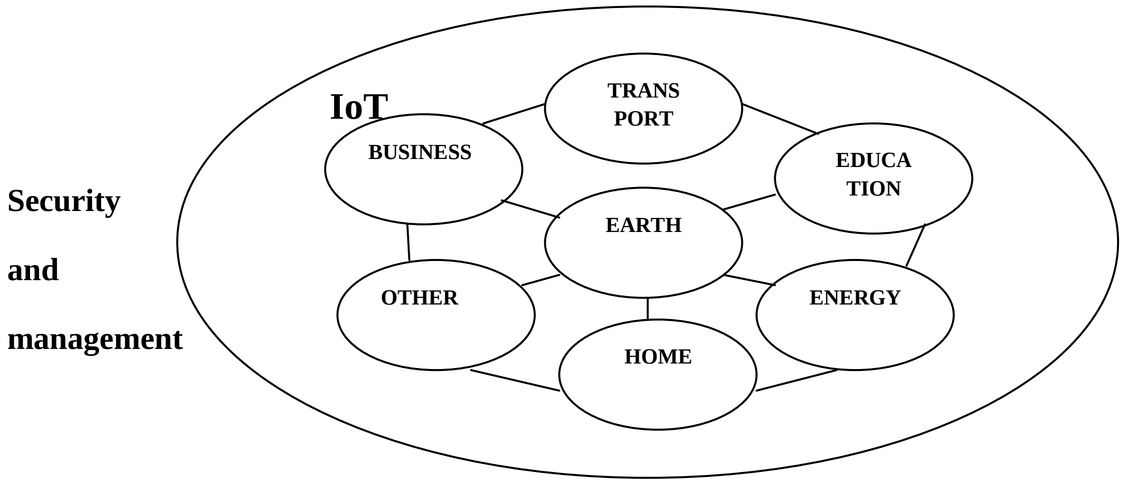

**Figure 4.1 Internet of Things- Network of Networks**

### 4.1.1 The Internet and the World Wide Web

The Internet and the World Wide Web are two terms that are used interchangeably. The Internet is considered as the physical layer or network and it is generally made up of switches, routers, and other devices. Its important work is to communicate information from one place to another place in quick and reliable manner as well as in secure environment. The World Wide Web (web) is an application layer which is operating generally on top of the Internet. Its main role is to provide an interconnection which makes the information transferring across the Internet connected devices.

### 4.1.2 Internet Protocol

Internet Protocol is a set of technical rules that describes how computers communicate over a defined network.There are currently two protocol versions used in the internet connection. IP version 4 (IPv4) and IP version 6 (IPv6). The function of Internet is transferring data between hosts in the form of packets which are routed across networks as specified by routing protocols. These packets will require an addressing scheme, either IPv4 or IPv6, to specify their source and destination addresses. Every host, it may be computer or any other device on the Internet requires an IP address in order to communicate between them. The huge growth of the Internet has necessitated for more addresses than are possible with IPv4.

### 4.1.3 IPv4 Protocol

IPv4 address size is only 32 bit number. IPv4 has address format of dotted Decimal Notation (192.149.252.76). The name of IPv4 shows that it's the fourth stage of the key Internet Protocol, but IPv4 was the first protocol that was widely used in modern TCP/IP.This protocol provides the basic datagram delivery functions and all of TCP/IP functions. Main reason for IPV4 exhaustion is tremendous growth in number of internet users and smart phones devices using Internet connection. This condition changes the TCP/IP environment to further development and introduction of IPV6 as an alternate protocol to provide endless internet connection. The header detail of IPv4 is shown in Figure 4.2.

| Version | IHL | Type of service | Total length | |
|---|---|---|---|---|
| Identification | | | Flags | Fragment offset |
| Time to live | Protocol | | Header checksum | |
| Source address | | | | |
| Destination address | | | | |
| Options | | | | Padding |

$\longleftarrow$ **32 bits** $\longrightarrow$

**Figure 4.2 IPv4 header structure**

### 4.1.4 IPv6 Protocol

IPv6 represents for Internet Protocol version 6 (also known asIpng (IP next generation)) is the second version of the Internet Protocol to be used widely across the virtual world. Its address size is 128 bits. IPv6 has address format of Hexadecimal Notation (3FFE:F200:0234:AB00: 0123:4567:8901: ABCD). Main aim of IPv6′s address space expansion is to remove the use of NAT (network address translation) and improving total connectivity of the network with improved Reliability and flexibility of the network. End-to-end traffic and transparency across the Internet is reestablished by IPv6. A second major goal of IPv6 is to reduce the total time spent to configure and manage systems. Both IPv4 and IPv6 are using an internet-layer protocol for packet switched internetworking communication and also provide end-to-end datagram transmission across multiple IP networks. The header structure of IPv6 is shown in Figure 4.3.

| **Version** | **Traffic class** | **Flow label** | |
|---|---|---|---|
| **Payload length** | | **Next header** | **Hop limit** |
| **Source IP address** | | | |
| **Destination IP address** | | | |

**128 bits**

**Figure 4.3 IPv6 header structure**

## 4.1.5 Tunneling Concepts

As IPv6 uses 128 bit addresses and IPv4 uses the 32 bit addresses, the incompatibility problem naturally exists between these two protocols. Therefore IETF (The Internet Engineering Task Force) suggested that IPv4 and IPv6 protocols need to co-exist for a certain amount of time until complete migration from IPv4 to IPv6 takes place**.** Adopting the new protocol requires changes to dozens of earlier network protocols like DNS, PPP, NAT, DHCP etc. associated with IPv4 protocol, Since these protocols were developed based on the protocol IPv4(32 bit addresses).

Tunneling concepts are used on top of an existing IPv4 protocol and uses IPv4 protocol to route the IPv6 packets between IPv6 based networks by transporting these encapsulated packets in IPv4.Tunnelling is used by all the networks and capable of offering native IPv6 functionality. Tunneling is the main concept for deploying IPv6 protocol to create global connectivity. Figure.4.4 indicates the tunneling concept.Tunneling concept used as main technique by which one transport protocol is encapsulated as the payload of another protocol. Tunneling techniques includes manual, semi-automatic, automatic configured tunnels.Tunneling technique can be used by routers and hosts in the network.The tunneling concept are of various categories depending upon the type of system that connect with the type of protocol. The general ways are Host to Host, Router to Host, Router to Router, Host to Router Tunnels. This is shown in Figure 4.4.

There are two steps of procedure followed when tunneling concept is applied in the network.

a) Encapsulation of IPv6 packets to IPv4 packets&

b) Decapsulation of IPv4 packets to IPv6 packets.

Post Office Protocol 3(POP3) is used as an important toolin Internet electronic mail transfer. Simple Network Management Protocol (SMTP) is designed to deliver mail into the mailbox from the Internet. POP3 is designed to collect mail from the mailbox and deliver them to their recipients even if they are located in remote area.

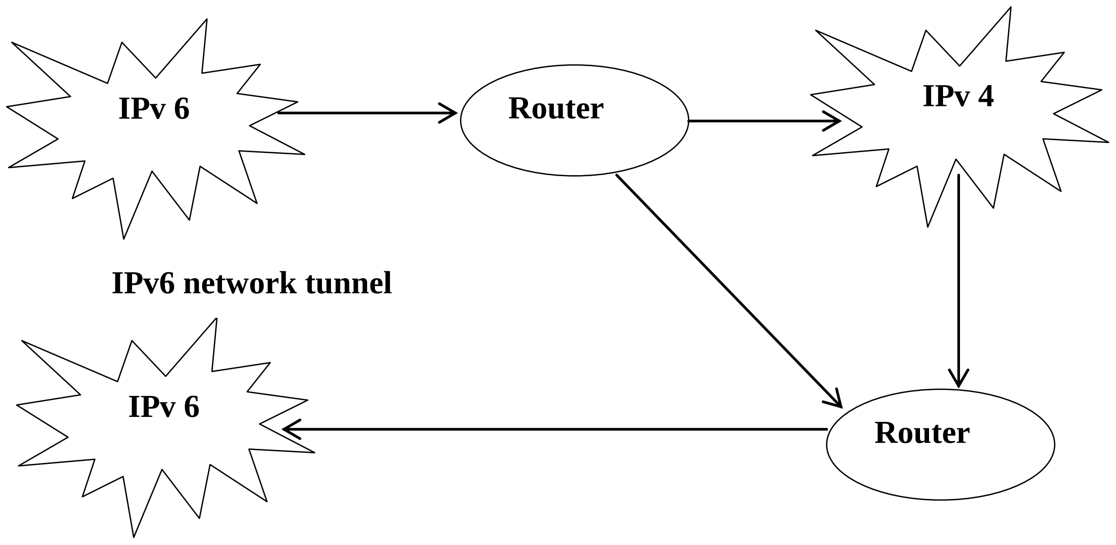

**Figure 4.4 Tunneling concept**

### 4.1.6 Simple Network Management Protocol (SNMP)

SNMP is one of the most popular protocol used to manage all the devices which are in network. One of typical use of SNMP is for management of network devices in remote condition. The SNMP protocol is used to enable network as well as system administrators to remotely monitor and configure devices on their network, such as servers, hubs, switches and routers. SNMP is not only monitoring the usage of CPU, but also the voltage and attributes of the particular device and surrounding conditions other than the main work of Network traffic throughput. This is shown in Figure 4.5. and Figure 4.6.

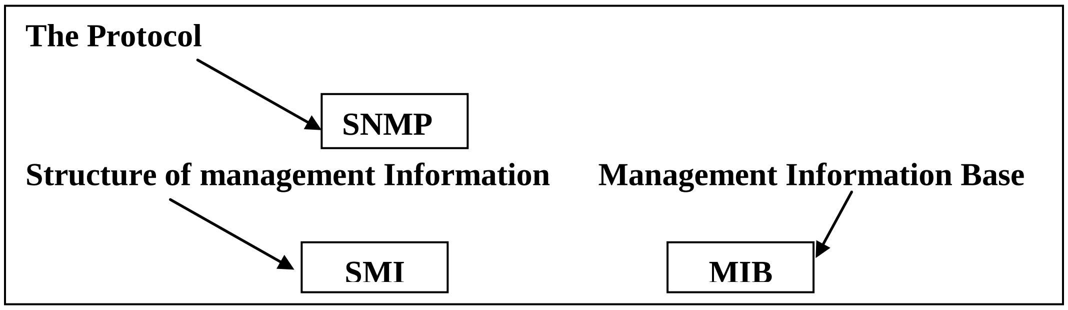
The Protocol
SNMP
Structure of management Information
Management Information Base
SMI
MIB

**Figure 4.5 Components of TCP/IP Network Management**

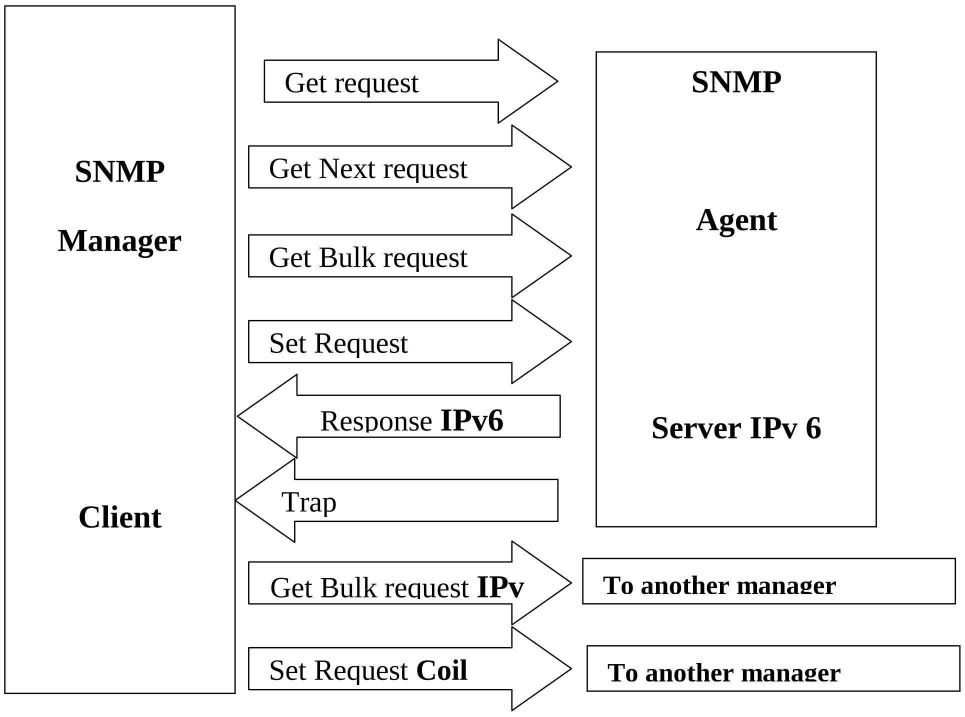

**Figure 4.6 SNMPMessages**

## 4.1.7 Post Office Protocol 3 (POP 3)

This protocol is generally used  for receiving email over the Internet. POP3 server must be connected  to read email from an email account. After connected with the server, all the email messages in an email account are numbered from 1 and the particular account is locked. This specified number identifies email in the given account. The account number is not changed even if the mail is deleted but the account number can be changed when the connection is closed. Always POP3 account uses only one connection at any one particular time. So that only when the present connection is closed, the new incoming email can be posted in the account.

All the Email delivery models show how email servers as well as email clients are connected with each other. Three different models are used in the internet to deliver email namely; Online, Offline and Disconnected models. In this research, protocol POP3 is designed to work using the offline model. In this offline model, a client node connects online with an email server, downloads the mail, and node goes to offline. Now all mail and its attachments are stored on the client node and completely deleted from the email server. The user can do all the work related with the mail like reads, prints, deletes the mail while the client node is offline. Figure 4.7 shows the messages transferred to the client node in one way when the client node is connected to the server.

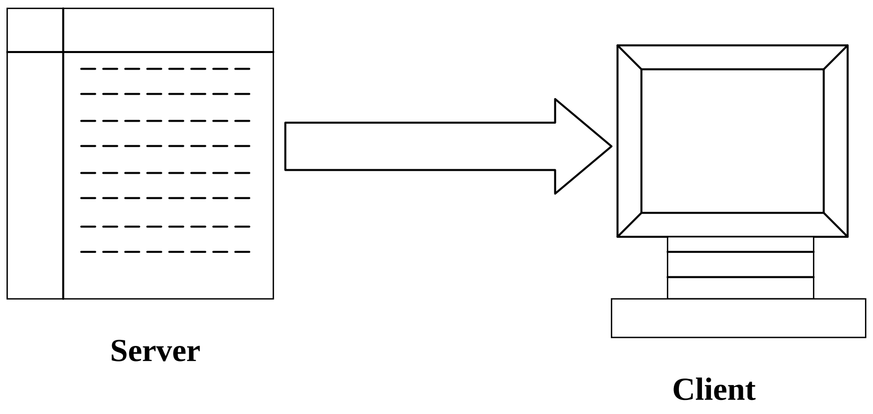

**Figure 4.7 Post Office Protocol 3 (POP 3)**

## 4.2 Intrinsic location information

Generally IP addresses are assigned for one address per network interface rather than per host. Thus, even with nonunique or null addresses, IP can tolerate point to point interfaces. However, interfaces on different networks should have the address of different network portions. This is achieved in this work by assigning location information in addresses itself. This eliminates any duplicate address assigned to reach the endpoint (point to point communications), such that it cannot be addressed by other hosts in the given network. The offset field use 13 bits in the IPv4 header and packets length varies upto $2^{16} – 1$. Maximum sized fragmentation is achieved by using the multiples of $2^3$ bytes ($2^{16-13}$) for count offsets. The tradeoff is increased space in a network with MTU (=8n+7bytes) and inconvenient in alignment on 8-byte boundaries.

### 4.2.1 Intrinsic Error detection

In this work, the internet checksum is protected in case of any signaling error and this has the advantage of not altering the version and HLen fields and thus retaining IPV4 packets. In conventional case, these two fields are changed due to this error and non-IPV4 packets are created.

### 4.2.2 Optimal Fragmentation

In this research work, optimal fragmentation is introduced to increase the load on the network and thus lesser complication of packet handling both at the endpoints and along the way. Consider two networks are interconnected by a router and first & second network has MTU of $N_1$ bytes and $N_2$ bytes respectively. The IP header size is static and is 20 bytes. The size of the TCP message is also $N_1$ bytes. Then, Largest IP datagram size carried by first network’s MTU is given by

MTU=largest IP datagram size+ IP header

Largest IP datagram size=MTU-IP header

Largest IP datagram size=$N_1$-20

To make largest IP datagram size to be a multiple of 8, N1-20 is approximated as nearest

8 Mod ($N_1$-20) =$N_3$.

Total data size=TCP message size+ IP header =$N_1$+20.

Then this data is fragmented into the data size of $N_3$& ($N_1$+20-$N_3$).

Similarly, largest IP datagram size carried by second network's MTU is given by,

Largest IP datagram size=$N_2$-20

To make largest IP datagram size to be a multiple of 8, $N_2$-20 is approximated as nearest

8 Mod ($N_2$-20) =$N_4$.

Total data size=TCP message size+ IP header =N1+20.

Then this data is fragmented into the data size of $N_4$ & ($N_1$+20-$N_4$).

For illustration, the values are tabulated in Table 4.1.

**Table: 4.1 Optimal Fragmentations**

| N1 | N2 | Measured values | | | |
|---|---|---|---|---|---|
| | | N3 | ($N_1$+20-$N_3$) | $N_4$ | ($N_1$+20-$N_4$) |
| 1024 | 576 | 1000 | 44 | 552 | 448 |

Thus, the number of fragment optimally depends on the router MTU value. Consider the source node transmit (A) $N_x$ IP data size to the destination node (B). In between A and B, two intermediate node act as router and it has different MTU value such as $N_{R1}$ and $N_{R1}$+152 ($NR_2$). The block diagram for optimal fragmentation is shown in Figure 4.8.

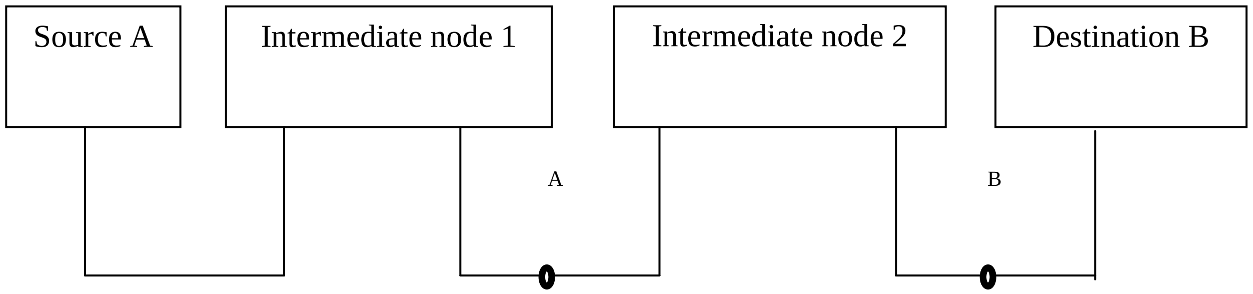

**Figure 4.8 Block diagram for optimal fragmentation**

## Fragmentation at point A

The maximum fragment size =$N_{R1}$-20 (IP header size) = $N_M$ bytes

Then the fragmentation is shown in Table 4.2.

**Table 4.2 Fragmentation at point A**

| M | Offset | Bytes data | Source |
|---|---|---|---|
| 1 | 0 | $N_M$ | 1st original fragment |
| 1 | 64 | $N_M$ | 2nd original fragment |
| 1 | 128 | $Nx-(2N_M)$ | 3rd original fragment |

## Fragmentation at point B

The maximum fragment size =$NR_2$-20 (IP header size) = $N_N$bytes

Then the fragmentation is shown in Table 4.3

**Table 4.3 Fragmentation at point B**

| M | Offset | Bytes data | Source |
|---|---|---|---|
| 1 | 0 | $N_N$ | Ist original fragment |
| 1 | 360 | $N_M$-$N_N$ | Ist original fragment |
| 1 | 512 | $N_N$ | 2nd original fragment |
| 1 | 872 | $N_M$-$N_N$ | 2nd original fragment |
| 1 | 1024 | $N_N$ | 3rd original fragment |
| 0 | 1384 | (Nx-(2$N_M$)) - $N_N$ | 3rd original fragment |

### 4.2.3 Fragmentation reassembly based on link

Typically, Fragment reassembly of IPv4 has done at endpoint only and not in the adjacent router. Alternatively, if reassembly is done at the downstream router, it will be done at the link layer and transparent to IPv4. But, in cases where link layer fragmentation is not practical, this requires IP layer fragmentation which is an overhead and also performance problems of IPv6 fragmentation. In this research work, both IPv4 & IPv6 can avoid link layer fragmentation by doing fragmentation based on the nature of the link.

**Decreased probability of loss**

In conventional method $P_1$% probability of loss has occurred independently when 'N' individual fragments of given IP packet were transmitted. So ($P_1$*N) % of data is lost in the whole packet approximation. In this research work, decreased probability of loss is achieved by using Ident field. The effect of Ident field in decreasing the probability of loss analyzed by considering the whole packet is transmitted twice.

a) If all fragments received from part of the transmission then the probability of net loss in both the transmission will be $P_1$ % (i.e. ($P_1$*N) × ($P_1$*N)).
b) If fragments received from part of either transmission then the probability of loss of any particular fragment in both transmissions is $10^{-4}$(i.e. $P_1$ % × $P_1$ % = $10^{-4}$). If this happens only once in 10 different fragments, the loss is 0.001(10 times this). In this research work, this is achieved by using same value for ident field in retransmission. Received packet contains fragments from each transmission when reassembly timeout is higher than retransmission timeout.The probability of loss is shown in Figure 4.9. Scheme 1 and scheme 2 are the existing method.

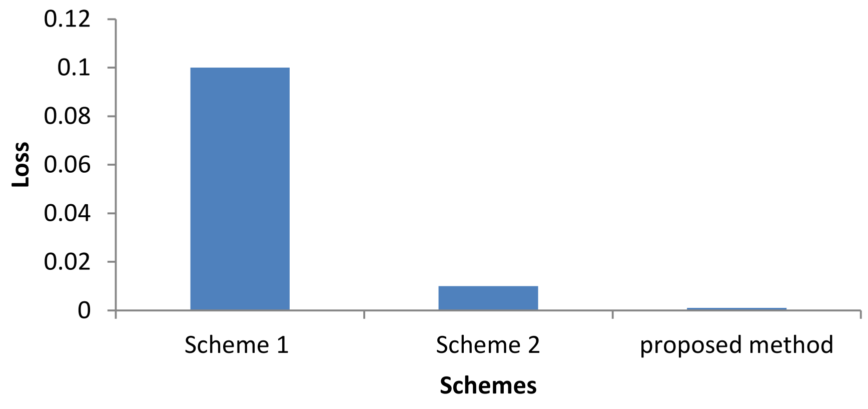

**Figure 4.9 Probability of loss Vs various schemes**

### 4.2.4 Maximizing bandwidth

The maximal bandwidth is determined by Ident field. For example if the Ident field is 16 bits, 5 Mbps data (i.e.576 (data size) ×$2^{16}$bytes per 60 seconds) can be sent. This puts a limitation on the maximum data and if more data will be sent than this data value, fragments of one packet can have the same Ident value as fragments of another packet.

### 4.2.5 Self-ARP and its benefits

- In existing methods, routing table updates their values in default timeout manner (10 to 15 minutes). If timeout value is small, network will be flooded with unnecessary rerequests and transmission is halted until all the re-request are answered. If the timeout value is too long, network waiting time increased. In this research work, self – ARP is implemented to solve the problem of too long and small ARP timeout by design hosts should update their caches, finally ARP cache timeout is set as high and it is necessary as a backup.

- In network, initially IP uses ‘x’ (32-bit) address. At time $t_1$instead of ‘x’ (32 bit) address IP use ‘y’ (6-byte MAC) address. Each hardware interface has its own MAC address in this research work, self-ARP mapping enables the direction from IP addresses to MAC addresses of its hardware. So when hardware moves to another network, ARP mapping reallocates dynamically IP addresses. Example for this is when mobile devices are allowed to another network access. If IP addresses assigned as MAC addresses, static IP addresses is needed. Routing is less efficient when use static IP addresses because internet routing use address space hierarch that is it uses higher bits address for network and lower bit address for host.

- In this research work, IP address confliction is identified by self-ARP. Assume host A & B have same IP address on same Ethernet on which ARP is used. If B starts after A, what happens to A’s connections when B starts after A.All the data traffic to ‘A’ will be halt when B broadcasts ARP query to allstations. Since all stations will switch to send B’s address. If B uses its own self-ARP on startup, it willindicate that its IP address is already in use. So the B cannot continue on the network until the problem is resolved by using self-ARP.

In conventional method, IP implementation follows this algorithm on receipt of a packet ‘P’, destination for IP address ‘D’.

```
If (Ethernet address for D is in ARP cache)
```

```
        (send PT)
    else
    (send out an ARP query for D)
        (put PT into a queue until the response comes back)
```

If multiple packets arrive at the IP layer before response from ARP comes back, the unnecessary multiple ARP packets are send. This will interrupt every host and consume bandwidth. If cache lookup fails after sending query, it will lead to excessive and frequent packet loss in new connections. In this research work, this problem is overcome by maintaining the database of ARP queries at all time. Before sending query, this database is checked and then retransmitted after some suitable timeout.

### 4.2.6 Global distance vector table

In this research work, routing function operates independent of fragmentation requirements by using global distance vector table. The global distance vector table formation is illustrated with the network shown in Figure 4.10. Three different cases are explained in Table 4.4 to 4.6.

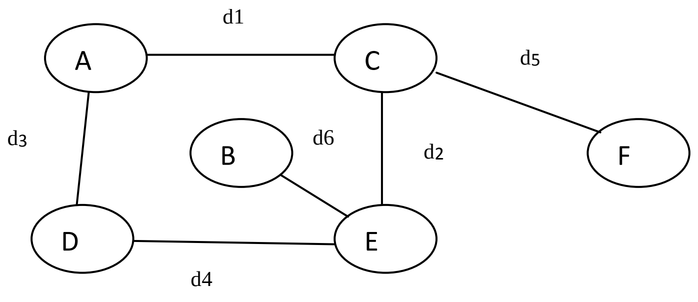

**Figure 4.10 Global distance vector table**

**Table 4.4 Case i: Immediate neighbors distance only known to each node**

| Node | Distance to Reach Node | | | | | |
|---|---|---|---|---|---|---|
| | A | B | C | D | E | F |
| A | 0 | ∞ | Dist[A to C] | Dist[A toD] | ∞ | ∞ |
| B | ∞ | 0 | ∞ | ∞ | Dist [B to E] | ∞ |
| C | Dist [C to A] | ∞ | 0 | ∞ | Dist [C to E] | Dist [B to E] |
| D | Dist [D to A] | ∞ | ∞ | 0 | Dist[D to E] | ∞ |
| E | ∞ | Dist [E to B] | Dist [E to C] | Dist[E to D] | 0 | ∞ |
| F | ∞ | ∞ | Dist [F to C] | ∞ | ∞ | 0 |

All diagonal elements are zero.

**Table 4.5 Case ii: Each node gets information in the preceding one to immediate neighbors**

| Node | Distance to Reach Node | | | | | |
|---|---|---|---|---|---|---|
| | A | B | C | D | E | F |
| A | 0 | ∞ | Dist[A to C] | Dist [A to D] | Dist[A to C]+ Dist[C to E] | Dist[A to C]+ Dist[C to F] |
| B | ∞ | 0 | Dist[B to E]+ Dist[E to C] | Dist[B to E]+ Dist[E to D] | Dist [B to E] | ∞ |
| C | Dist [C to A] | Dist[C to E] + Dist[E to B] | 0 | Dist[C to A]+ Dist[A to D] | Dist [C to E] | Dist [B to E] |
| D | Dist [D to A] | Dist[D to E]+ Dist[E to B] | Dist[D to E] + Dist[E to C] | 0 | Dist[D to E] | ∞ |
| E | Dist[E to D]+ Dist[D to A] | Dist [E to B] | Dist [E to C] | Dist[E to D] | 0 | Dist[E to C]+ Dist[C to F] |
| F | Dist[F to C]+ Dist[C to A] | ∞ | Dist [F to C] | ∞ | Dist[E to C]+ Dist[C to F] | 0 |

**Table 4.6 Case iii: Second time above case (ii) happens**

| Node | Distance to Reach Node | | | | | |
|---|---|---|---|---|---|---|
| | A | B | C | D | E | F |
| A | 0 | Dist[B to E]+ Dist[E to C]+ Dist[C to A] | Dist[A to C] | Dist [A to D] | Dist[A to C]+ Dist[C to E] | Dist[A to C]+ Dist[C to F] |
| B | Dist[B to E]+ Dist[E to C]+ Dist[C to A] | 0 | Dist[B to E]+ Dist[E to C] | Dist[B to E]+ Dist[E to D] | Dist [B to E] | Dist[B to E]+ Dist[E to C]+ Dist[C to F] |
| C | Dist [C to A] | Dist[C to E]+ Dist[E to B] | 0 | Dist[C to A]+ Dist[A to D] | Dist [C to E] | Dist [B to E] |
| D | Dist [D to A] | Dist[D to E]+ Dist[E to B] | Dist[D to E]+ Dist[E to C] | 0 | Dist[D to E] | Dist[D to E]+ Dist[E to C]+ Dist[C to F] |
| E | Dist[E to D]+ Dist[D to A] | Dist [E to B] | Dist [E to C] | Dist[E to D] | 0 | Dist[E to C]+ Dist[C to F] |
| F | Dist[F to C]+ Dist[C to A] | Dist[F to C]+ Dist[C to E]+ Dist[E to B] | Dist [F to C] | | Dist[E to C]+ Dist[C to F] | 0 |

### 4.2.7 Subnet Masking & Routing issue

A router forms a routing table as shown in Table 4.7. The router can directly deliver the packets using interfaces 0 and 1 or these packets can be delivered to routers R2, R3 or R4. The condition of the router when the following destinations are assigned to the router is explained in the flowchart of Figure 4.11 along with the routing table of Table 4.7. Destinations are 128.96.39.10, 128.96.40.12, 128.96.40.15, 192.4.153.17 and 192.4.153.90

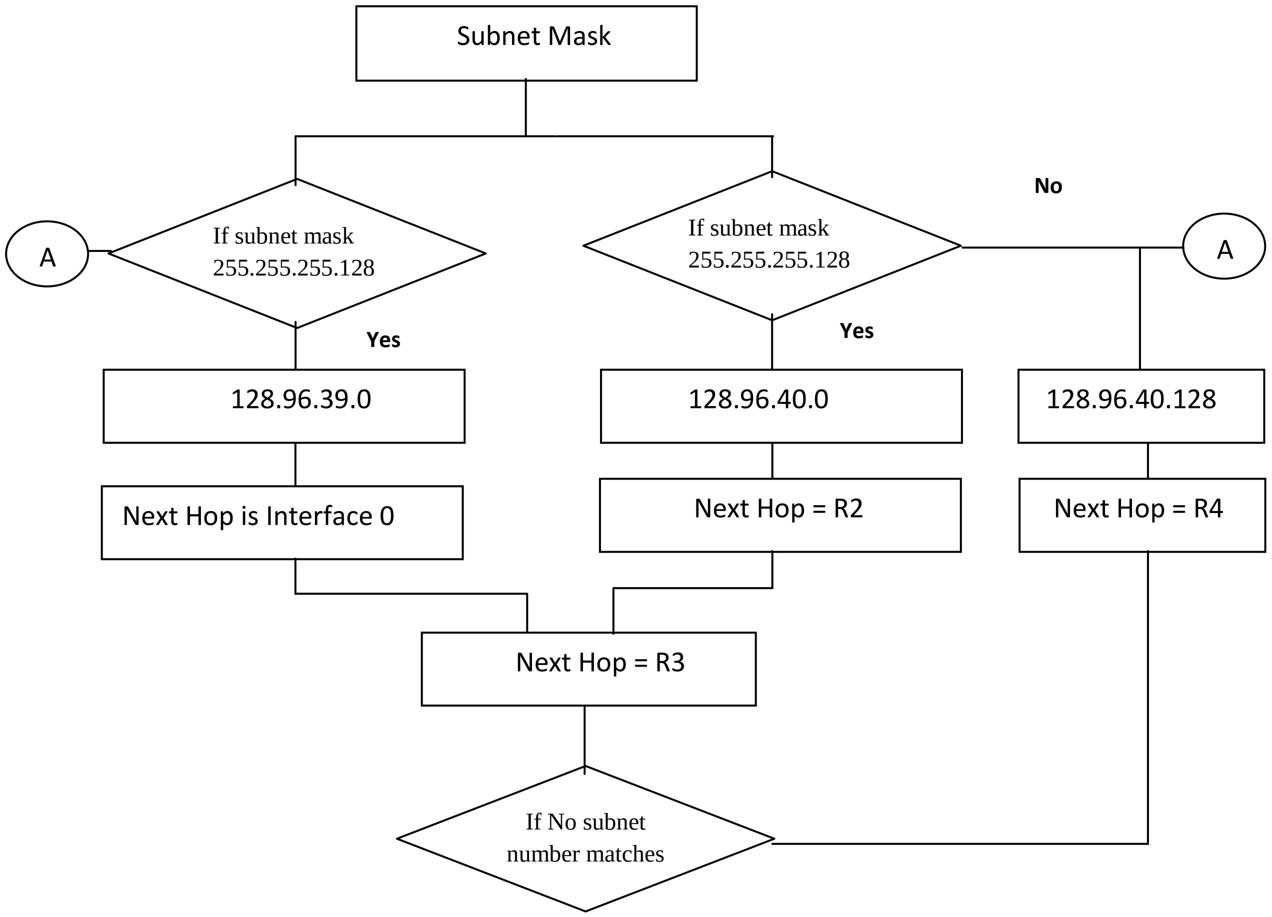

**Figure 4.11 Flow chart for Subnet Masking & Routing issue**

**Table 4.7 Routing table**

| Subnet Number | Subnet Mask | Next Hop |
|---|---|---|
| 128.96.39.0 | 255.255.255.128 | Interface 0 |
| 128.96.39.128 | 255.255.255.128 | Interface 1 |
| 128.96.40.0 | 255.255.255.128 | R2 |
| 192.4.153.0 | 255.255.255.192 | R3 |
| (default) | | R4 |

### 4.2.8 Concurrent update process

When A – E link goes down, All the update process of A, B & C to destination E is listed. In this work, it is assumed that all the updates are done concurrently and split horizon technique is observed by all the nodes (Refer Figure 4.12). For example A sends report to B about unreachability of E. The concurrent update process is explained as follows:

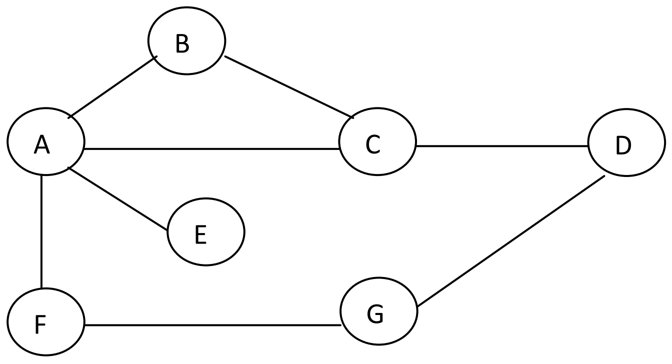

**Figure 4.12 Example network for Concurrent update process**

Case 1: A → B about E

B → C or C → B updates will not change E

By split horizon method, B → A & C →A are not allowed.

First relevant report isA → B

C’s entry for E is (E, 2, A)

Case 2: Now A→ C, C →B & B→C; A → B

Exchanges will not change E entries and C →A is disallowed by split horizon.

If A →C (or) B→C the loop formation is halted.

So, C → B is performed.

C’s table entry is (E, 2, A)

B’s Table entry is (E, 3, C)

Case 3: Possible updatesare, B → A (or) A→ C

A’s table entry (E, 4, C) & loop is complete.

If A→ C, then B becomes sole believer.

Possible update for E is B → A, now exchange A→ C from the loop.

When C→ B, A is sole believer.

1. A loop can be formed at some point.
2. At some point all belief in E are eliminated.
3. C → B →A→ C →....., sole believer status changes around the loop.

### 4.2.9 Forwarding table for Smallest Network

Forward tables are given for nodes A and F. If all links in a given network have unit cost the network diagram that has smallest network consistent with these forwarding tables is given in Table 4.8 and 4.9.

**Table 4.8 Forwarding tables – A**

| A | | |
|---|---|---|
| Node | Cost | Next hop |
| B | 1 | B |
| C | 2 | B |
| D | 1 | D |
| E | 2 | B |
| F | 3 | D |

**Table 4.9 Forwarding tables – F**

| F | | |
|---|---|---|
| Node | Cost | Next hop |
| A | 3 | E |
| B | 2 | C |
| C | 1 | C |
| D | 2 | E |
| E | 1 | E |

The minimum network is calculated as

A connects to B and D; F connects to C and E.

F reaches B through C at cost 2, so B & C must connect.

F reaches D through C at cost 2, so D & E must connect.

A reaches E at cost 2 through B, so B & E must connect.

These give the network of Figure 4.13. It gives unique minimal solution and this network also is consistent with the tables. These are explained in the flowchart in Figure 4.14 and 4.15.

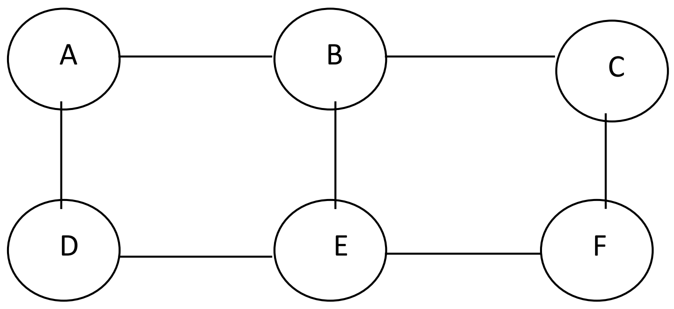

**Figure 4.13 Example for smallest network**

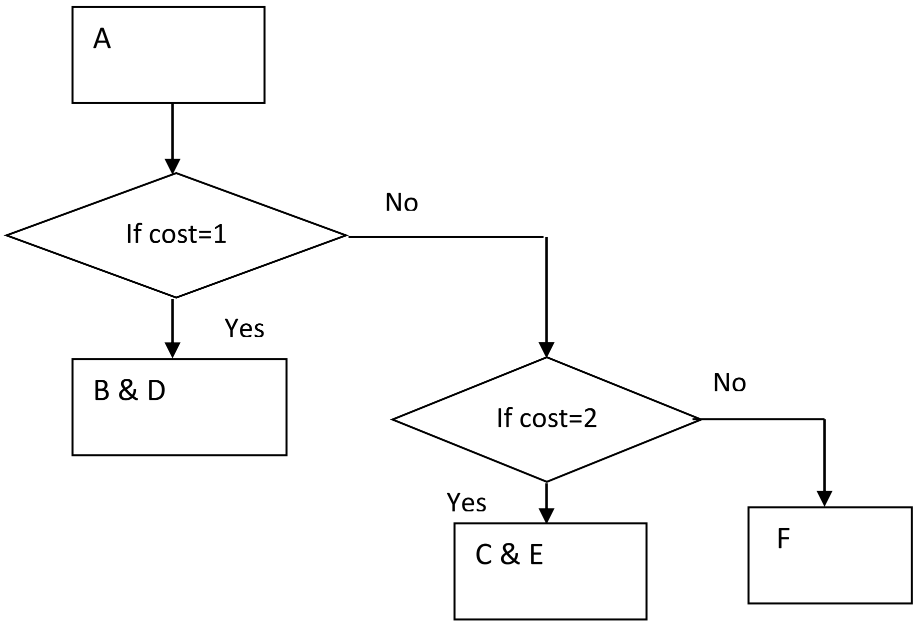

**Figure 4.14 Flow chart for Forwarding tables – A**

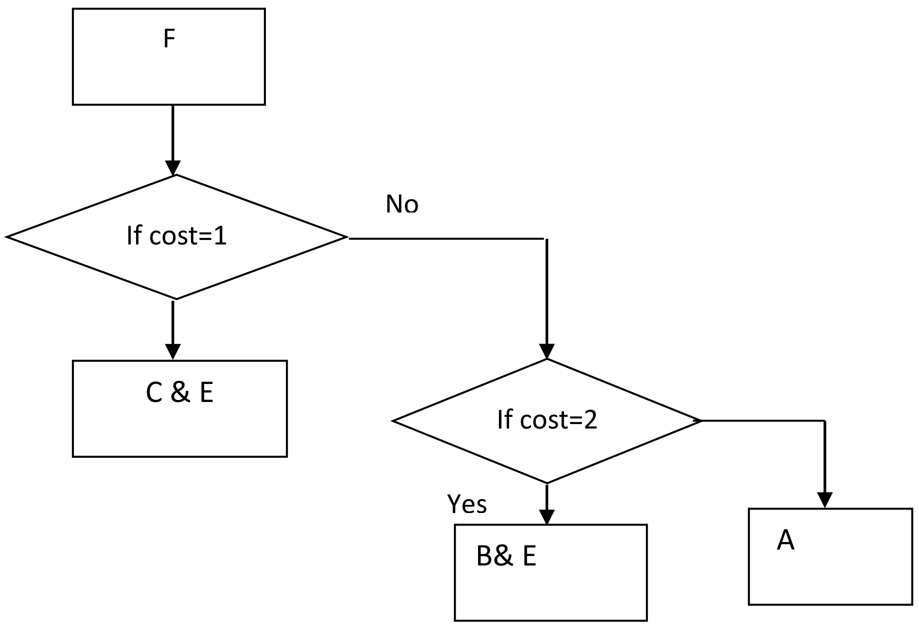

**Figure 4.15 Flow chart for Forwarding tables - F**

### 4.2.10 Link reliability issues

Alternate to the split – horizon technique, the routers can also use the poison reverse in addition. However, When the A-E link fails, whether or not poison reverse is used it is observed that the following sequence happen depends on the timing and the oscillation of information continues between the routers A, B &C as shown in figure 4.16. The advantage of poison reverse technique is that it eliminates false routers to unmentioned destinations, otherwise the routers A and B send updates to each other that did not mention the particular destination X and false routes to X would form until they aged out eventually. With the poison reverse technique, such kind of loop would be eliminated away on the initial table exchange itself. Slow links are also accommodated and for slow links

1. The routers B and A send out the information of their route to destination X via C to each other.

2. The router C announces to A and B that it cannot reach X. The informations given by the routers A and B have not arrived yet.
3. The routers A and B receive information from step 1 each other and adapt according to them.

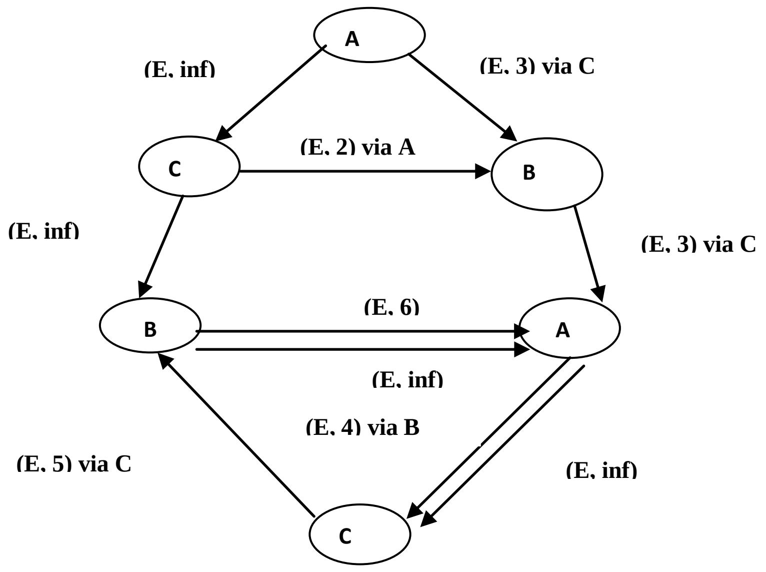

**Figure 4.16 Link reliability issues**

## 4.2.11 Fair Queuing (FQ)

The main issue with FIFO queuing is that it cannot separate different traffic sources or packets. Such kind of situation affects the internet's routers with flood of its own packets to be discarded. Fair queuing is to form a separate queue for each flow being handled by the particular router. As an illustration, a router with three input and one output flow is considered. The packets are received by the router is given in Table 4.10. As a result, the output port is busy and all queues are empty.

The finishing times is calculated as

$$F_i = F_{i-1} + P_i \quad \rightarrow \quad (4.1)$$

**Table 4.10 Example for Fair Queuing** [Note P=200 & $F_1$=1]

| Packet | Size | Flow |
|---|---|---|
| 1 | P/2 | $F_1$ |
| 2 | P/2 | $F_1$ |
| 3 | P/2 | $F_1$ |
| 4 | P/2 | $F_1$ |
| 5 | 0.95P | 2 $F_1$ |
| 6 | P | 2$F_1$ |
| 7 | 0.55P | 3 $F_1$ |

Where clock speed $A_i$ = 0 for all packets. So $F_i$ is calculated using eqn.4.1 and tabulated in Table 4.11:

**Table 4.11 Solution for Fair Queuing Example** [Note P=200 & $F_1$=1]

| Packet | Size | Flow | $F_i$ |
|---|---|---|---|
| 1 | P/2 | $F_1$ | 100 |
| 2 | P/2 | $F_1$ | 200 |
| 3 | P/2 | $F_1$ | 300 |
| 4 | P/2 | $F_1$ | 400 |
| 5 | 0.95P | 2 $F_1$ | 190 |
| 6 | P | 2$F_1$ | 390 |
| 7 | 0.55P | 3 $F_1$ | 110 |

The packets are sent in increasing order of $F_i$ as

Packet 1, Packet 7, Packet 5, Packet 2, Packet 3, Packet 6, Packet 4.

### 4.2.12 Weighted Fair queuing

In this work, flow 2 is assigned weight 4 and other two flows are assigned weight 1.

$$F_i = F_{i-1} + P_i / 4 \quad \rightarrow (4.\ 2)$$

Now Fi is calculated using eqn.4.2 and tabulated in Table 4.12:

**Table 4.12 Solution for Weighted Fair Queuing Example** [Note P =200 & $F_1$=1]

| Packet | Size | Flow | Weighted $F_i$ |
|---|---|---|---|
| 1 | P/2 | $F_1$ | 100 |
| 2 | P/2 | $F_1$ | 200 |
| 3 | P/2 | $F_1$ | 300 |
| 4 | P/2 | $F_1$ | 400 |
| 5 | 0.95P | 2 $F_1$ | 47.5 |
| 6 | P | $2F_1$ | 97.5 |
| 7 | 0.55P | 3 $F_1$ | 110 |

### 4.2.13 Random Early Detection (RED)

RED implicitly notifies congestion source by dropping packets rather than sending explicitly congestion notification message to the particular source. Average queue length is computed using a weighted average as same as TCP timeout computation and is calculated as

$$\text{AvgLen} = (1 - \text{Weight}) \times \text{AvgLen} + \text{Weight} \times \text{SampleLen}$$

**Case i:**

1. RED gateway with MaxP = 0.01

(where MaxP is the maximum probability of drop of packets)

2. Average queue length is halfway between two thresholds.

3. For example Pcountfor count = 1 and count = 100

TemP = MaxP × [(AvgLen–MinThreshold) ÷ (MaxThreshold – MinThreshold)]

Here, AvgLen is halfway between MinThreshold and the fraction is ½.

TemP = MaxP/2 = 0.005

$P_{count}$ = TemP / (1-count × TemP)

= 1/ (200- count)

For count = 1 it is 1/199

For count = 100 this is 1/100

**Case ii:**

With initial 'M' packets are assumed as dropped. Probability for this case is calculated as follows.

$(1-P_{i)}$ × - - - - - - - - - - × $(1-P_M)$

For M = 50; 198/199 × 197/198 × 196/197 × - - - - - - - - - × 150/151 × 149/150

Probability is 149/199 or 0.7487

### 4.2.14 Slow start

A router R is in between A and B. Here A-R bandwidth is infinite but R-B link has bandwidth delay of 1 packet per second. From B to R, acknowledgement is sent instantly. Slow start with large window size is followed when A sends data to B over a TCP connection R has a queue size of one. In some cases, once timeout and retransmission is pending, subsequent later packets timeouts are ignored until acknowledgement is received. But no action is taken even though it is shown in Table 4.13 and 4.14.

**Case i:** Fixed Timeout period is 2 seconds. There is no idle time in R-B link.

**Table 4.13 Slow start process for fixed timeout of 2 seconds**

| Time | A receives | A sends | R sends | cwnd size |
|---|---|---|---|---|
| 0 | | Data0 | Data0 | 1 |
| 1 | Ack0 | Data1,2 | Data1 | 2 |
| 2 | Ack1 | Data3,4(4 dropped) | Data2 | 3 |
| 3 | Ack2 | Data5,6(6 dropped) | Data3 | 4 |
| 4 | Ack3 timeout 4 | Data4 | Data4 | 1 |
| 5 | Ack3 timeout 5 & 6 | Data6 | Data5 | 1 |
| 6 | Ack5 | | Data6 | 1 |
| 7 | Ack6 | Data7,8(slow start) | Data7 | 2 |

**Case ii:** Fixed Timeout period = 3 sec. No message is transmitted at T =6 because ack 4 has not yet been received.

**Table 4.14 Slow start process for fixed timeout of 3 seconds**

| Time | A receives | A sends | R sends | cwnd size |
|---|---|---|---|---|
| 0 | | Data0 | Data0 | 1 |
| 1 | Ack0 | Data1,2 | Data1 | 2 |
| 2 | Ack1 | Data3,4(4 dropped) | Data2 | 3 |
| 3 | Ack2 | Data5,6(6 dropped) | Data3 | 4 |
| 4 | Ack3 | Data7,8(8 dropped) | Data5 | 5 |
| 5 | Ack3 timeout 4 | Data4 | Data7 | 1 |
| 6 | Ack3 timeout 5 & 6 | | Data4 | 1 |
| 7 | Ack5 timeout 7 & 8 | Data6 | Data6 | 1 |
| 8 | Ack7 | Data8 | Data8 | 1 |
| 9 | Ack8 | Data9,10(slow start) | Data9 | 2 |

## 4.3 IoT Architecture

IOT architecture consists of different suite of technologies supporting IOT. It serves to illustrate how various technologies relate to each other and to communicate. This is shown in Figure 4.17. The functionality of each layer is described below:

### Sensor Layer

The lowest layer is made up of smart objects integrated with sensors. The sensors enable the interconnection of the digital and physical worlds allowing real time information to be collected and processed.

## Management Service Layer

The management service renders the processing of information possible through analytics, security controls, process modeling and management of devices. IOT brings connection and interaction of objects and systems together providing information in the form of events or contextual data such as temperature of goods, current location and traffic data.

## Application Layer

There are various types of applications from industry sectors that can leverage on IOT. Applications can be verticalised ones that are specific to a particular industry sector, and other applications.

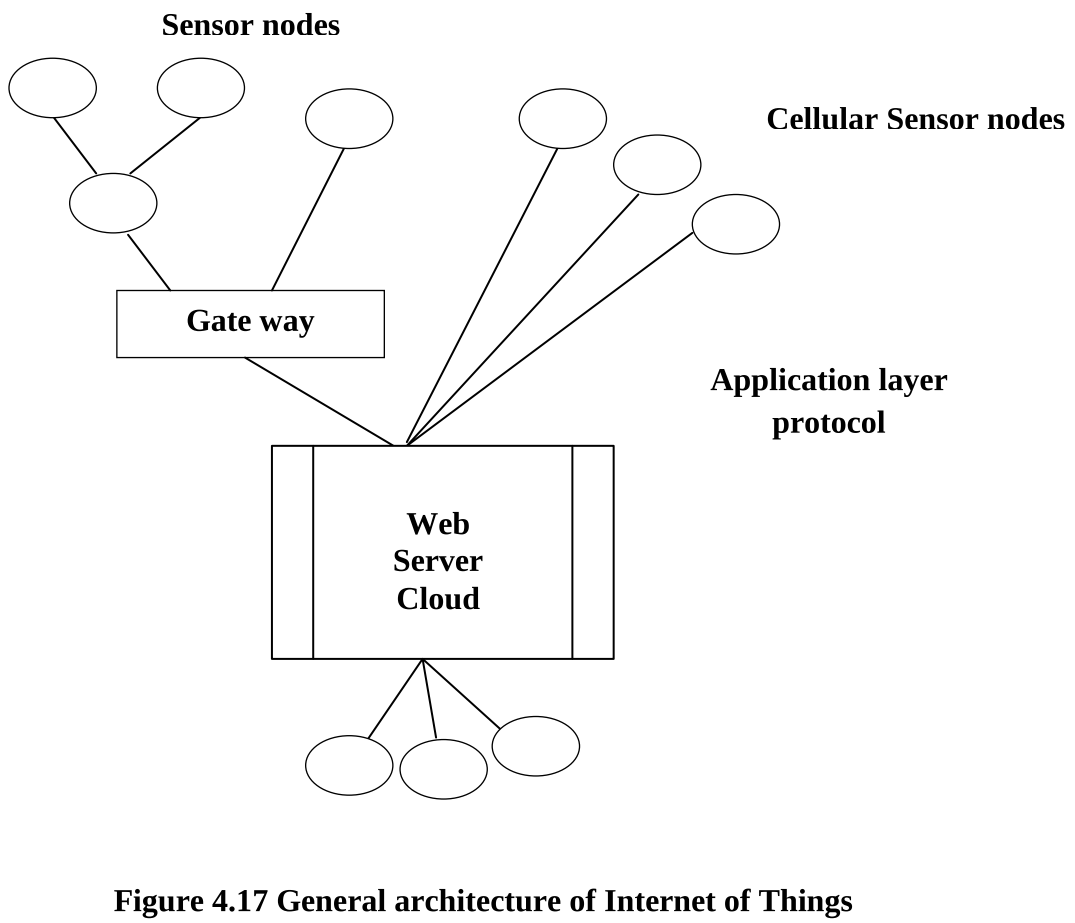

**Figure 4.17 General architecture of Internet of Things**

### 4.3.1 Internet of Things (IoT) and Cloud computing Integration

Cloud computing and Internet of Things (IoT), two different technologies are both already in use. Their massive adoption and use is expected to increase further, making them important components of the Future Internet. The features of IoT and Cloud for a better integrated operation are listed in Table 4.15.

**Table 4.15 General features of IoT and Cloud computing**

| IoT | Cloud |
|---|---|
| All devices are pervasive in nature | Cloud is Ubiquitous in nature |
| IoT is generally characterized by real world and small things with limited storage and processing capacity. | Cloud computing has virtually unlimited capabilities in terms of storage and processing power. |
| IoT can benefit from the virtually unlimited capabilities and resources of Cloud to compensate its technological constraints. | Cloud can benefit from IoT by extending its scope to deal with real world things in a more distributed and dynamic manner. |
| Limited or less computational capabilities: | Unlimited virtually computational capabilities. |
| No storage capabilities or limited storage : | Unlimited virtually storage capabilities: |
| Internet as a point of convergence :<br>IoT is characterized by a very high heterogeneity of devices, technologies, and protocols.. | Internet for service delivery:<br>The integration with the Cloud solves most of these problems also providing additional features such as ease-of-access, ease of-use, and reduced deployment costs |
| Big data source :<br>IoT devices have limited processing resources that do not allow on-site data processing. | To manage big data:<br>The unlimited processing capabilities of Cloud and its on-demand model allow IoT processing needs to be properly satisfied. |

### 4.4 Implementation of applications based on IoT

(i) Personal parking slot management device that automatically detects the availability of vacant slots for parking vehicles.
(ii) Weather information reporting device (useful in crisis such as a landslide).
(iii) Smart Irrigation
(iv) Air Pollution Monitoring

The above mentioned IoT applications are implemented in this research. Here, the embedded controller ATMEGA Arduino with the sensors is connected to the on-board diagnostics through Internet of Things. The open source clouds services like Xively, Django (with 24x7 backend support) and Twitter are used to monitor the state of the process sent from the controller and update the same automatically to the vehicle owner. This work can convert processes that are presently non real-time in an existing control system into real-time. The non-requirement of GPS or GSM device and the integration of the cloud service are the merits of this work.

#### 4.4.1 Smart parking system

The main purpose of the smart parking system is used to detect the available number of empty parking slots and send the relevant information over the internet to smart parking application back ends. These applications can be accessed easily by corresponding drivers from tablets, Smartphone, or from in-car navigation systems. Personal parking slot management device is used to automatically detect the available vacant slots for parking vehicles. Here, sensors are placed in each parking slot, to detect whether the parking slot is empty or occupied. This received information is aggregated by a local micro controller and then sent over the internet to a server. Each parking slot has ultrasonic sensors fixed, which is used to detect the presence of a vehicle in a parking slot. Each sensor is read at regular intervals and the condition of the parking slot (occupied or empty) is updated in a database. The domain model updates a physical entity for the parking slot and the corresponding virtual entity. The device implemented in this example is a Raspberry-pi embedded system which has ultrasonic sensor and weather monitoring units attached to it. The domain model has the open source services and web services required for information transmission. The smart parking information model defines the state (attribute) of the parking slot virtual entity with only two possible values (empty or occupied). The weather information model reports to a twitter account and no need to use additional services.

### 4.4.2 Weather reporting system

Weather information reporting device is very useful in crisis such as a landslide. The main purpose of the weather reporting system is to collect information data on environmental conditions such as temperature, pressure, light and humidity in an area using multiple types of end nodes. The weather information of the specific place is sent through tweets on Twitter. The every end node is comprised of a Raspberry Pi minicomputer, temperature, pressure, light and humidity sensors. The system consists of multiple nodes placed in various locations for monitoring temperature, pressure and humidity in an area. The end nodes are equipped with different sensors (such as temperature, pressure, light and humidity). The end nodes send the sensed data to the cloud and the data is stored in a cloud database. The analysis of sensed data is done in the cloud to aggregate the data and make possible predictions. Here, a Twitter app is used for visualizing the received data. The centralized controller can send control commands to the end nodes, for example, to configure the monitoring interval on the end nodes. The components and devices used in this example are Raspberry Pi minicomputer, temperature and humidity sensor (DHT22), pressure and temperature sensor (BMP085) and LDR sensor. An analog-to-digital (A/D) converter (MCP3008) is used for converting the analog input from LDR to digital.

### 4.4.3 Smart irrigation

Smart irrigation system use IoT devices and soil moisture sensors to determine the amount of moisture in the soil and release the flow of water through the irrigation pipes only when the moisture levels go above a predefined threshold. Data on the moisture levels is also collected in the cloud where it is analyzed to plan watering schedules. The system consists of multiple nodes placed in different locations for monitoring soil moisture in a field. The end nodes send the data to the cloud and the data is stored in a cloud database. A cloud-based application is used for visualizing the data.  The end node includes a Raspberry Pi mini-computer and soil moisture sensor. A solenoid valve is used to control the flow of water through the irrigation pipe. When the moisture level goes above a threshold, the valve is opened to release water.

### 4.4.4 Air pollution monitoring

IoT based air pollution monitoring system can monitor emission of harmful gases by factories and automobiles using gaseous sensors. The system consists of multiple nodes placed in different locations for monitoring air pollution in an area. The end nodes are equipped with CO Sensors. The end nodes send the data to the cloud and the data is stored in a cloud database. A cloud-based application is used for visualizing the data. The end node includes a Raspberry Pi minicomputer   and Gas sensor (CO). An A/D converter (MCP3008) is used for converting the analog service for air pollution monitoring system. The controller service Runs as a native service on the device and obtains the store these measurements in the cloud.

## 4.5 Chapter Conclusion

Congestion control and fragmentation in IoT are studied in this chapter. Tunneling concepts are generally used on top of an existing IPv4 protocol and it uses IPv4 protocol to route the IPv6 packets between IPv6 based networks by transporting encapsulated packets in IPv4. SNMP and POP3 protocols are studied in this chapter. In this research work, the internet checksum is protected without altering the version and HLen fields and thus retaining IPV4 packets. By doing fragmentation based on the nature of the link, link layer fragmentation can be avoided in IPv4 & IPv6. It is assumed that all the updates are done concurrently and split horizon technique is observed by all the nodes. Link reliability issues, Fair Queuing (FQ) and Weighted Fair queuing are studied. Random Early Detection (RED) and Slow start are the two important concepts are implemented in this research work. General features of IoT and Cloud computing are discussed and tabulated in this chapter.