**CSU33D03 Computer Networks- Project 0**

**By:Ark Nandan Singh Chauhan**

**Id:19323352**

## 1. PREFERRED USE CASE

'A network of AI powered smart refrigerators connected to each other via Internet of things.These smart fridges use supervised machine learning and deep learning to understand usage patterns and uses AI enabled software to identify food items inside to suggest various food recipes and performs actions like grocery order online,dietary control, nutrition monitoring, eating habit analysis and much more. Each of these devices learns from the user and share updates with other refrigerators having the same software version over the network and receive a proper response'. The updates are also shared over other devices like smartphone apps for the refrigerator any other devices connected on the network.

## 2. PROTOCOL OVERVIEW

Application layer protocols are based on TCP or UDP which enables communication of devices on IOT.The TCP protocol enables REST/HTTP communication protocols.

The "**transport layer"** deploys transmission control protocol(TCP) which is a connection-oriented protocol as devices first 'handshake' and then establish a full duplex connection allowing application layer data from process 1 on one host to process 2 on another host and vice-versa at the same time.

At the "**network layer"** we have the Internet Protocol(IP) is responsible for a logical connection between the host devices. One of the primary responsibilities of TCP is to provide IP delivery service between the end systems for communicating updates/data between two processes running on the end devices.

We also use the Border Gateway Protocol version 4 (BGP4) which is an interdomain routing protocol based on path-vector routing.Aside from safety and security, the goal of this protocol is mere reachability thus allowing packets to reach destination more effectively.

## 3. COMMUNICATION MODEL

"For communication between between two refrigerators A(source) and B(destination) as shown in figure1, all the network layers are under consideration for both the devices.A message is created in the application layer and sent down the layer from where its physically transmitted to destination B". The destination host receives the message at the physical layer and then deliver it through the other layers to the application layer. A 'router' is always involved in one network layer, n combinations of link and physical layers where n is the number of links the router is connected to.A 'switch' in a link, however, is involved only in the data-link and the physical layer. Although each switch in the figure below has two different connections, the connections are in the same link, which uses only one set of protocols.

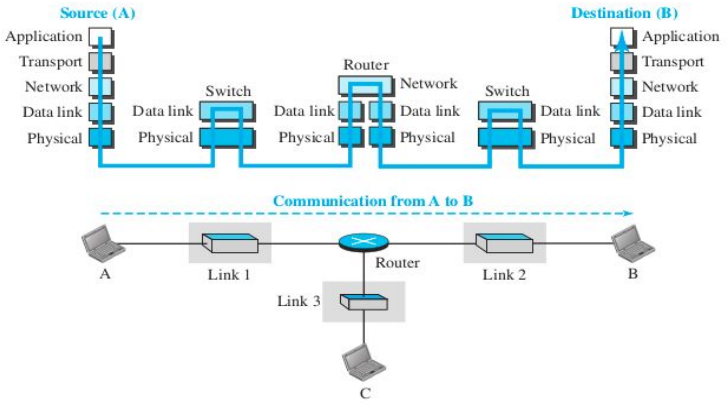


Fig1. Realization of communication via different network layers

We use path vector routing in out network to send the data packets from one refrigerator to another on the network . In contrast to link-state and distance-vector routing which are based on 'least-cost goal', this algorithm focuses more on safety and security of the data packets as the path is selected based on the specific policies of the sender thus preventing refrigerators with a different software version than the sender. The source controls the path, the path from source to all destinations is determined by the best spanning tree which is not a least-cost tree but determined by the policies of the sender, where the policy is defined by selecting the best of multiple paths. Each node creates its own vector when its booted then sends it to its immediate neighbour after they have created their arrows for data transmission direction.

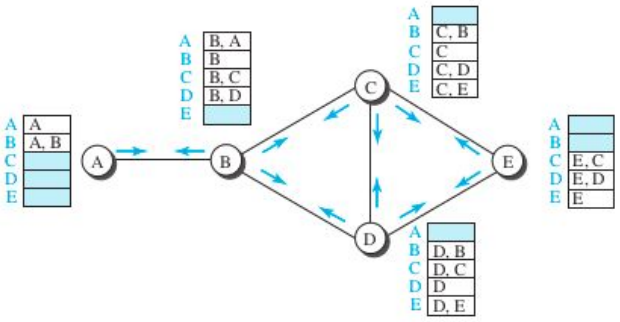


Fig2.Path vectors made by each node at time of their booting

## 4. MODULE DESCRIPTION

Different specific modules of the communication between the devices are specified as follows:

**Transmission Control Protocol's Three Way Handshake**

The TCP protocol uses a three way handshake process which uses a positive acknowledgement and retransmission(PAR) methodology.

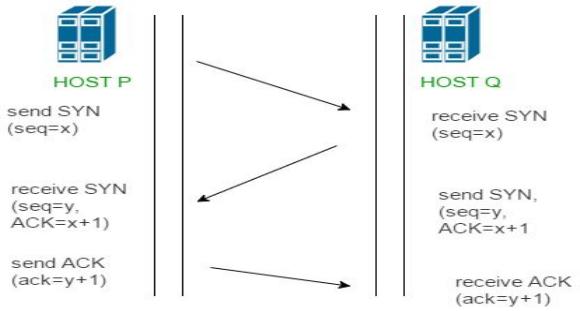


Fig3. Three Way Handshake

**Step 1.** "First host P tries to establish a connection with host Q, so sends segments with SYN(Synchronize Sequence Number) which informs host Q that P wants to start a conversation and the sequence number with which its segments will start".
**Step2.** "Q responds to P with an acknowledgement of the data it received and SYN represents the sequence number with which its segments will start".
**Step3.** "The final step is that the host P acknowledges the response of Q and finally a secure connection is established to start actual sharing of data".

**Latency in the Network**

In our network multiple conversations take place simultaneously where the bandwidth determines the number of conversations while the latency, a measure of delay drives the responsiveness of the network of smart devices.Thus we need to define congestion at the network layer, which is the state when the data traffic is so high that it slows down the network response time thus decreasing network performance.
To prevent congestion TCP uses congestion control policies which uses the concept of congestion window which dictates the size of the sender's window size which makes sure that the rate at which the data is transmitted is same at which it is received.
Congestion control include 3 steps:
**Step1.** The **'Slow Start Phase'** deploys a congestion window whose size is increased exponentially after every round trip delay time is estimated.
**Step2.** After crossing a threshold value the window size increasing additively in order to **avoid congestion**.
**Step3.Congestion Detection Phase:** Congestion is detected when the sender is needed to retransmit the data.The retransmission occurs when the retransmission timer times out(high congestion possibility) or because of receiving 3 duplicate acknowledgements(low congestion possibility).In all of the cases the window size is decreased and the process is started at step1 again.

**Network Security**

Various *cryptographic techniques* can be used to provide data integrity, confidentiality and other point to point authentication. We use an enhanced version of TCP known as *Secure Sockets Layer(SSL)*.SSL provides a simple Application Programmer Interface (API) with sockets, which is similar and analogous to TCP's API. SSL technically resides in the application layer and whenever it is needed to be employed, the application includes the SSL libraries.

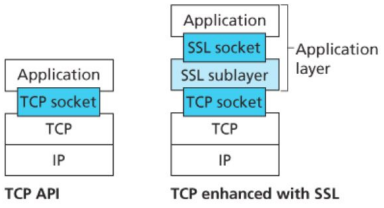


Fig4. SSL resides in the application layer but from a developer's perspective it is a transport layer protocol

"The security is provided by a method called the SSL/TLS(Transport layer security) handshake.During a TLS handshake the two communicating devices message with each other and *acknowledge* and verify one another and then establish an *encryption algorithm* they will use and agree upon the session encryption keys."

**Error Detection and Control(EDC)**

The link layer provides the functionality of error control. Considering the figure below if data D and EDC is sent as a whole message from the sender.Both D and EDC are sent to the receiving node in a link-level frame. At the receiving node, a sequence of bits, D′ and EDC′ is received. Note that D′ and EDC′ may differ from the original D and EDC.In order to verify this we use the Cyclic Redundancy Checks(CRC) method for error detection.

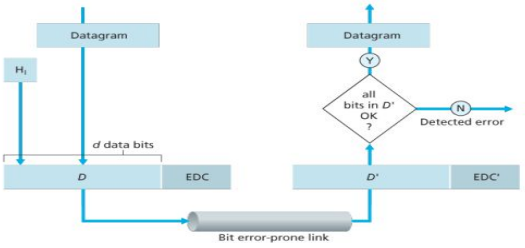


Fig5. Error Detection and correction use case

CRC is used at the link layer. It is one of the most efficient methods since it is possible to view the bit string to be sent as a polynomial whose coefficients are 0 and 1 values in the bit string, with operations on the bit string interpreted as polynomial arithmetic.

## 5. SUMMARY OF ALGORITHMS

Firstly algorithm used for routing of data packets is path vector routing algorithm.

***Pseudo code:***

```
Path_Vector_Routing ( )
{
 // Initialization for the nodes
        for (y = 1 to N) {
                if (y is myself)
                        Path[y] = myself
                else if (y is a neighbor)
                        Path[y] = myself + neighbor node
```

```
                else
                        Path[y] = empty
        }
        Send vector {Path[1], Path[2], ..., Path[y]} to all neighbors
// Update vector after receiving a vector from neighbour
        repeat (forever) {
                wait (for a vector Path w from a neighbor w)
                for (y = 1 to N) {
                        if (Path w includes myself)
                                discard the path
                        else
                                Path[y] = best {Path[y], (myself + Path w [y])}
        }
        If (There is a change in the vector)
                Send vector {Path[1], Path[2], ..., Path[y]} to all neighbors
        // Avoid any loop

  }//End of forever loop

} // End of Path Vector
```

Each node, after the creation of the initial path vector, sends it to all its immediate neighbors. Each node, when it receives a path vector from a neighbor, updates its path vector using an equation similar to the Bellman-Ford, but applying its own policy instead of looking for the least cost. We can define this equation as:

**Path(x, y) = best { Path(x, y), [(x + Path( v , y)] }**

for all v ’s in the network.In the equation, the operator (+) means to add x to the beginning of the path.

## 6. SOFTWARE DEVELOPMENT PRACTICES

In the software development lifecycle we would have to perform requirement analysis which would be put together by inputs from customers, market surveys and thus determine features of smart fridge relevant to public interest with various network parameters.While designing the product architecture we use the ‘***threat modelling approach’*** in which we perform source code analysis in the design stage and find all vulnerabilities in the routing algorithm, before analysing the risk level in applications.

In the building/software development stage of the cycle we make use of the ‘**defense in depth’** secure coding method thus make our applications and network safe from attacks.There should be multiple layers of protection to prevent an attack in case the first layer is breached. We also use the concept of ***‘positive security’*** where our application is trained to understand about new attacks and malware.

During the testing phase of the SDLC we use the concept of ***’fail securely’*** such that if the network fails, it's already prepared with a backup plan and does not reveal any sensitive data.