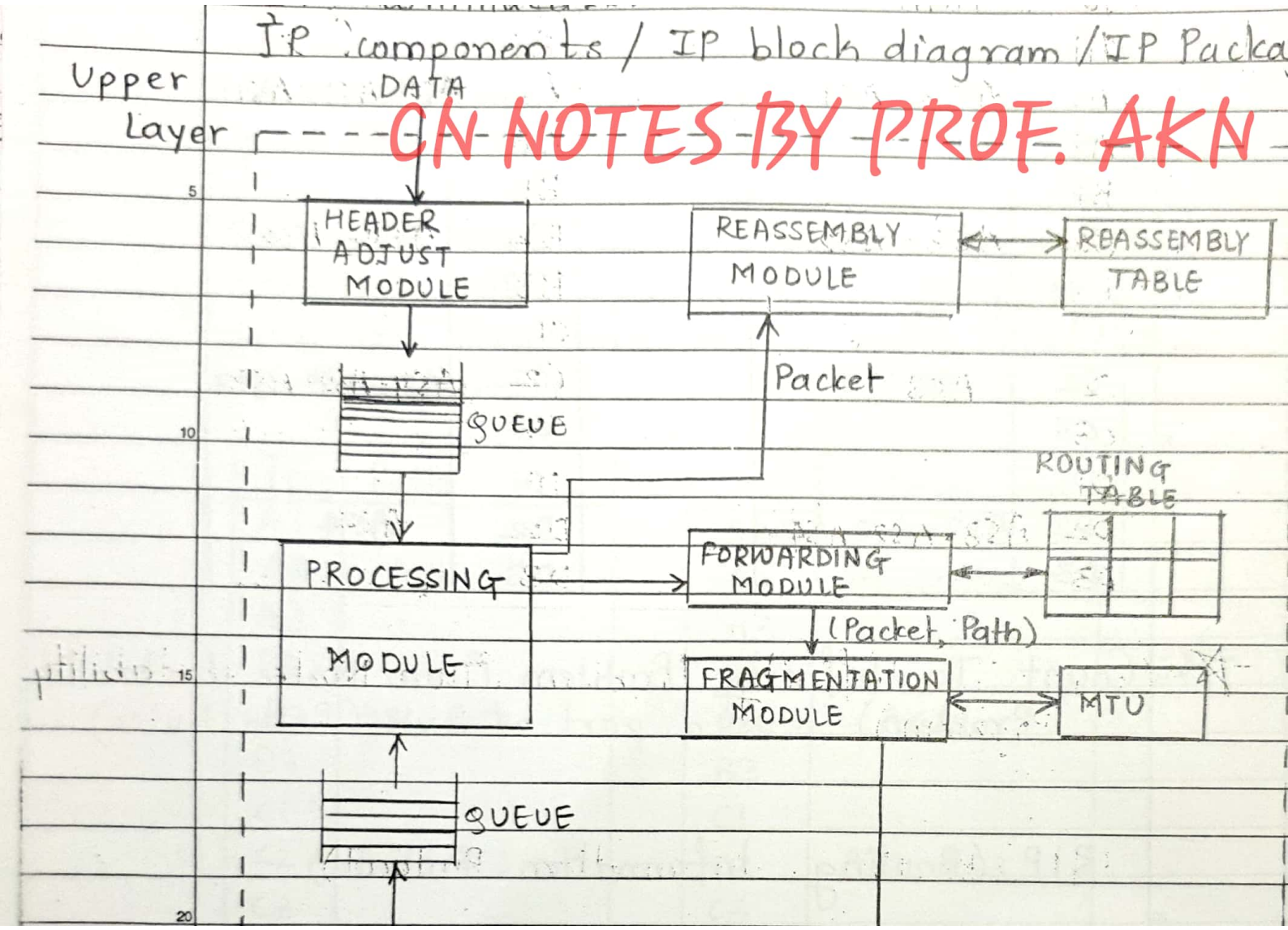

# CN NOTES BY PROF. AKN

1) Header adding module -

```
Header_Add_Mod (Data)
{
  1. Apply Header (Encapsulated)
  2. Return
}
```

2) Forwarding module -

```
Forward_Mod (IP Packet)
{
   1. Check the destination IP address where
      the packet needs to be forwarded.
   2. Refer routing table and select the
      path.
   3. Return
}
```

3) Fragmentation module -

```
Frag_Mod (IP Packet, Path)
{
   1. Check the network capacity of the
      path selected
   2. Refer MTU
   3. If (network capacity > packet size)
      {
         a. Send the IP Packet to lower level
         b. Return
      }
```

4. else
{
a. if ( DF == 1)
{
i. Discard the packet and inform the sender using ICMP
ii. Return
}
b. else
{
i. Fragment the packet
ii. Perform necessary calculation
iii. Apply header on every fragment
iv. Send IP packet to lower layer
v. Return
}
}
}

4) Processing module -

```
Process_Mod (IP Packet)
{
  1. if (sender)
     {
       a. Forward the IP packet to forwarding module
       b. return
     }
```

```
2. if (receiver machine) // destination
   {
      a. Forward the IP packet to
         reassembly module
      b. Return
   }
3. if (router)
   {
     a. if (TTL == 0)
        {
           i. Drop the packet and inform the
              sender using ICMP.
           ii) Return
        }
     b. else
        {
           i. TTL--
           ii. Forward the IP Packet to
               forwarding module
           iii. return
        }
   }
}
```

# CN NOTES BY PROF. AKN

5) Reassembly module-

```
Reassemble_mod(IP Packet)
{
  1. if (Packet) // MF = offset = 0
    {
      a. Remove the header (decapsulate)
      b. Send data to upper layer
      c. Return
    }

  2. else // Fragment
    {
      a. Start the timer
      b. Collect all the fragments in the
         Re-assembly table.
      c. if (timer == 0) // expires
        {
          i. Perform re-assembly of all fragment
          ii. if all fragments received go to
              step 1
          iii. else discard all the fragments
               and inform sender using ICMP.
          iv. return
        }
    }
}
```