Coursera : Nick Feamster.
Network protocol layers :
Application layer, Transport layer : end to end and reliable, Network layer : routing , congestion control, addressing and Quality of service Link layer : link by link basis, Physical layer.

Components of a router :

- Line cards.
  - Each line card is for one particular file layer technology. Based in transmitter/Recv).
  - Line cards these days have CPU and a memory dedicated to it.
  - It also takes care of forwarding. A packet based on the input header and route it to the next router on the route to the destination router. From the input port which output port it goes out in this router. Based on forwarding table it does all these. A simple forwarding table will have an input port mapped to the output port based on the header information. Address has two parts : network address and host address. Number of possible networks is the bound for the number of entries in the forwarding table.
- Control Card : Also has CPU and memory. Handles all control packets. Computes and loads the forwarding table.
- Management Card : Also has CPU and memory. Handles all monitoring packets.
- Switch fabric connects the different cards.

There are some packets which could be control packet like non data packet which is for routing protocols like OSPF. These are called control packets. RIP, REP and ISIS are other routing protocols. An autonomous system is the set of systems under one name (like reliance's systems under reliance name). BGP is the border gateway protocol is like sendin packt from one cluster to the next cluster. RIP is the protocol for internal transfer.

We have various configurations possible which can be configured using SMP.
Some are simply forwarded like data packets. The other are used to make some changes to the memory of the router.

Data plane or the forwarding plane, control plane and forwarding plane.
Routers do look at TCP header.

Innovations inside the network were growing at a slow rate.
Physical layers has been growing over the years. Wifi has grown from 2 Mbps to about 1 Gbps. Cellular receives about 5Mbps. Like from 2g to 3g to 4g the speeds increase. The base state for optic networks was 155 Mbps. It has gone upto 100 Gbps per channel. Hence

it has gone to about 10 Tbps on a link.

Application also has proceeded nicely. Changing things easily is possible.
FIND : Future Intermediate Directions.
NGI : Next Generation of Internet.
Clean State Internet.
Goals :

- Enable faster innovation in a production/live network. People who want the above are vendors or manufactureres of network equipment like cisco, juniper, tejas, huawei. The others who want these are Internet Service Providers. They take the equipment from the vendors and give internet service. The ISP has to give some service agreement. If they want to add a new service they have to go to the equipment vendor and ask for it. Based on something already existing, if an innovation is happening, then it must be done quickly.
- Network element programmability : An administrator should be able to write high level programs to modify the network element behaviour.
- Given a large scale network, formally verifying connectivity between two other nodes and such.
- How to eliminate proliferation of middle boxes.
  Middle box is a hardware sitting in the network between routers, basically anywhere between soure and destination.
  They enable :
  - firewall.
  - proxies / caching.
  - NAT(Network Address Translation) / Gateway.
  - Load balancing.
  - Intrusion detection or prevention.
  - WAN optimizers/Application accelerators.
  - This might have the power to send acks, etc to the nodes in order to change the parameters like recevier window size of the nodes.

The players involved are :
Internet service providers.
Enterprise nets are private networks like IIT, University of texas, etc.
Data center networks basically have racks of servers. They have almost 2 million IP addresses.

Content delivery network.

Network virtualization :
Jon turner made effort to incorporate network virtualization. Host OS over hardware was the initial stage. VMWare added a virtual box over Host OS. Now we can have linux running over windows and such. For cloud computing VMWare is necessary. (Using different versions over the underlying model of old machines)
We can start assigning cores to different versions. Slices of the computer infrastructure is the one discussed above.

The slices of network Infrastructure is network virtualization. Each slice can run its own protocol stack. PlanetLAB was the first testbed which can be used for wide area testing. Now there are GENI testbeds.

Network function Virtualization (NFV) will be discussed later.

Read about : Active networking, FORCES(IETF), Routing Control Protocol(RCP), Ethane project in stanford and write a one page report on each of them.

Notions of SDN :
Separation of control plane and data plane : Router's internals include :

- Communications hardware
- CPU, Memory, Buffers, etc
- Switching fabric
- Router operating systems : Cisco's (IOS), JunOS. Open Networking Linux is another OS.
- Features :
  - OSPF
  - RIP
  - daemon
  - BGP

The above are Vetically Integrated Closed Proprietory System.
The control plane is tightly interconnected with the data plane in the above.

Example of network virtualization:
An user wants a different set of protocols for the system. The routers have hardwares for

these. Now a user must be able to run the protocols which they want to run on. Different operators working on the same underlying interface.

SDN wants better. The integrated part was split into two parts as control and data plane.
The idea was to use the switch/router only implements the data plane (ie) the packet forwarding aspects only.

Forward based on :

- Destination address only are split into class based IP addressing(Fixed length prefix and has an exact matching to the incoming address) and the other is the classless IP addressing(longest matching prefix). Class based is more like a cluster while classless is not so.
- Based on several fields in the IP. This is actually breaking the IP stack rule. Based on <Src IP, Dest IP, Src Port, Dest Port, Protocol>. This has 5 fields. Therefore the routing becomes extremely complicated due to all these.
- The following is called the policy based routing which has about 40 fields :
  Forwarding data plane has a forwarding table. There may be various fields in the packet header. The packet has MAC, IP, TCP headers which in themselves have several fields ending up in about 40 fields. Based on the fields, we have action and auxillary actions. Because of these the forwarding table might become exponential. Hence we have some kind of aggregation for every field. There will always be a default rule. This exists now. A given packet can match multiple entries in the forwarding table.
  100 Gbps. TCP : 40 byte minimum packet size (TCP + IP). 40 bytes and 1500 bytes are where the spikes of statistics of data sent.
  Time available to process a packet in the forwarding table = (40*8)/(100 Gbps) = 3.2 ns. Therefore the processing time should be less than this for the packets not to queue up.

  The same packet might get different treatment at different routers.

Telephone networks :
The paths have been done via circuit switching. The controller decides these paths. The controller has the global view. This one is centralized routing.

OSPF is distributed routing as there is no one centralized node. Hence the distributed one is where each router has the ability to route on its own without the help of the centralized node.

Current challenge is to have some controllers but not one like centralized controller.

SDN model : Switch or router only implements the data plane functionality.
The control plane line card is to be removed from the router.
A well defined interface to the control plane. The controller is now being implemented as a software. Therefore this is moving towards programmability. The interface between control and data plane must be open and standardized. Only if that happens the communication between different companies would be easier. The controller is central for some routers and it is not specific to a router as before. Example for open interface is open flow. This is called south bound interface.

East west is between servers traffic and North south is between the server and the internet in data center terms.

The applications that talk to the controller software is called the north bound interface. Different south bound interfaces are NOX, POX, Beacon, Floodlight, OpenDayLight, ONOS.

The switch and router are to be designed to be dumb devices. The only functionalities are looking up forwarding table. Implementing packet classification in hardware :
First do a good algorithm in software and translate part of it as hardware. Do an ASIC/FPGA implementation of an algorithm.
FPGA : Field Programmable Gate Array.
ASIC : Application Specific Integrated Circuit.
FPGA is slightly better. We can modify easily.
Hardware lookup tables are also better. These are called ternary content address memory.
It is called ternary because 0, 1 or dont care.
TCAM is also programmable. TCAM is very fast, but costly and consumes more power. We must also try to ensure lesser size in TCAMs. Now the data plane has this hardware dedicated for it. This is a very simple router and hence it would become a commodity and the USP of CISCO, etc breaks. The vendors dont want to make dumb routers.

Active networking : Every packet has embedded code in it which tells the router how to progress. Specialized instruction could be given to the router for it to process the router. Read about IETF.

FORCES : separation of control and data within or outside of a router.
Ethane giving rise to OpenFlow.
Look at the slides. (History of programming networks). B - Broadcast, U - Unicast, M - Multicast which is specialized way of combining local networks. Like a match being streamed to one machine and several people near the machine gets the copies. So long range transmissions are enabled better. MBone was designed for multicast over the underlying network. They had specialized routers which did muliticast routing. They were called application level routers. This was considered because the underlying stuff need not be changed.
6Bone built IPV6 over IPV4. IPV6 is fixed length 40 byte header. IPV4 is variable length header. Similar to MBone they built IPV6 application level routers. Most of these were application level overlay.

Application to SDN controller(Control plane) to routers(Data plane). Standards are made by ONF. The application layer needs a consistent interface to the controller. The management and administration planes has interfaces to all the three. Management supplies software updates to the routers, it configures policy and monitors performance with the control layer.

Applications of SDN:
Data center networks, Enterprise networks, Wireless Access Networks, Optical networks, home and small business networks.

(Look at picture in slide)
Green arrows : When the entry is present router goes through these.
Blue arrows : When it doesnt know how to handle.
Scalability in terms of number of new flows is a huge bottle neck.
Data center networks :
Elephant flow : The number of flows is small and the bandwidth is huge. Backups, distributed computation (east west traffic). Most of the traffic is inside the data center. VM migration also takes a lot of time.
Mice flows : There are lot of such flows but very small bandwidths.

OpenFlow protocol (Read from slides).
A set of rules define flows like a tuple matching instead of just src IP and destination IP.

- Switch has two parts, Datapath and the SDN client.
- The communication between the client and the controller is generally TLS over TCP.
- TLS is transport layer security. We might TCP alone if we want faster performance. Forwarding element implements CPU like instructions.
- These were for Ethernet switches initially. Now they have extended to other switches and routers. Current version is 1.4. Openflow version 1.0 came in 2009.

Openflow Specs :

- In version 1.0 there were 12 fields.
- It had only one flow table. As we went on more flow tables were added for easier access or specialized processing.
- Initially the table is empty we learn the mac address of every port. If a new mac address comes in the controller is notified and a broadcast of the mac is made.

- Src Port, Destination Port, IP Protocol, Type of Service and 8 more fields are there in the flow table.

- There are many different actions which could be there for processing of a packet. Statistics are also collected based on what is done and would be given to the management plane. Actions are forward or drop in the first version. Special actions are Enqueue, modify. A flow table can specify multiply actions for a given packet. A port can have multiple queues. Modify packet headers can also be done.

- Packets are of two types:
  From a host or from the controller. If it is from the controller, we might need to update the router.

- If there are more than one flow table, the packet will go through all the flow tables and collect all the actions and do it finally.

- We might need to modify the src IP even while forwarding listed as an action (ie) modify packet. Update action set, update metadata are also different instructions to be applied. Controller is a piece of software.

- Now there are hardware switches implementing OpenFlow.

- Interoperability is majorly maintained.

- There are special virtual ports like LOCAL, ALL, CONTROLLER specifying where it must be sent. IN_PORT meaning send packet back on input port. TABLE packets sent by controller forwarded either on specified output in PACKET_OUT message or the Flow Table.FLOOD : Similar to ALL except that it send on links of the STP tree. NORMAL is the normal processing.

Hubs simply forward whatever it gets to all the nodes. Group table is a set of actions specified. OXM Header followed by TLV packets. The OXM header contains 4 bytes which also has the length of the overall packet which is from 5 to 259 bytes (1 to 255 otherwise).
There are master and slave controllers. When master fails, the slave takes over. There is a concept of equal controller where two controllers can access the same router.
Data path id is the connection number of the router to the controller.
Wildcard matching can be used to bypass some tables.
There are ways to look at the meter reading of how many bytes are received and such statistics.
Firefox : downthemall option for faster download due to parallelization.
Similarly there are multiple connections between the controller and the switch.
Controller caches data of PACKET_IN and uses it later. There is a cookie along with that packet.

Further versions of Open Flow also had mechanisms to notify the controller and ask which rules to evict. An atomic bundle mechanism was also developed where we either execute all the actions within or none of them.

Networks supporting tenants.
One tenant using 10.* and another also wants to use 10.* . How to implement something like this.

Look at all possible header space. In flow visor, you can split your header space into multiple slices. We take some slices and the remaining is considered the master slice.

- Policy language to map flows to slices.
- Resource can be sliced in terms of BW, Topology, Device CPU, Flow table entries and so on.
- Can operate on deplayed on networks. The production network can be a testbed.
- Slices work on different types of services.
- We can have multiple controllers and have each controller correspond to one slice.

The controller thinks the flow visor is the router and router thinks of the flow visor as the controller. The individual controllers are aware that they are part of the slice network though.

Flow visor controls the controllers and ensure that they are mutually exclusive. A new controller can be added while the other controllers are notified. There might be multiple controllers implementing the same slice which the flow visor will manage.
You can specify a virtual topology which would be embedded on a physical topology. This can be different. The mapping would end up being complicated. We can also specify the parameters for the topology.

Virtual links can be made up of multiple physical links. There is a possibility of same address space being shared across different virtual topologies. Different topologies may be implemented/controlled by different controllers. The OVX is the interface between the controllers and the switches.

For mice flows we require less latency and bandwidth can be compromised, while it is the reverse for elephant flows. Fat tree means more connections to the higher level switches. The hierarchy is Edge(or Top of the rack) switch to aggregation switch to core switch.

Over subscription could be an issue if all children keep producing to their best which might end up creating a problem. These are between the racks in the servers.