

E7311-SDN-22 – Lecture Notes

Lecture 1: Some basic concepts for Networking

Contents

E7311-SDN-22 – Lecture Notes.....	1
Lecture 1: Some basic concepts for Networking.....	1
Connections, forwarding, switching	1
Managing the forwarding entries.....	3
Flow state.....	4
Generalization of state in network.....	5
Types of addressing systems in networks.....	5
Classless Interdomain Routing (CIDR).....	7
Some routing metrics for packet networks	9
IPv6 – solution for future Internet?	9
Software Defined Networks (SDN).....	10
SCION – briefly.....	12
Scope of our discussion on this course	12

Prerequisites for the course

Learning Outcomes

We summarize the goal of the course at the end of this Chapter of the Lecture Notes.

Connections, forwarding, switching

Let us first define a few key terms: *wavelength (or circuit) switching*, *packet switching*, *connection-oriented* and *connectionless*. These terms are fundamental to understanding networking. First, *wavelength switched networks* are always connection oriented. In such networks, the units that are switched in network nodes do not carry address information and a switching fabric state is established in each node between an incoming interface and an outgoing interface such that data units (or the wavelength) will continuously flow from the incoming to the outgoing interface. In these networks only having first established a connection from end to end, users of the network can transfer data. Usually, we establish a bidirectional connection. E.g. routed IP connections can be carried over wavelengths. Each wavelength is a switched connections of constant capacity (e.g. 40 Gbps).

In wavelength or circuit switching, the addressing of the switched units is “implicit” i.e. the placement of the physical connection + the placement of the data within the multiplex that is carried on the physical connection = address.

Packet switched networks may be either connection-oriented or connection less. Packets always carry a header including some address information plus a payload. In a connection-oriented packet network, the packets will carry address info that has only *local* significance. As a result, an end-to-end *virtual connection* needs to be established before two users can communicate. This is the case for example in ATM or Multi-protocol Label Switching (MPLS).

Similarly, in IP networks private IP addresses have only local significance. Therefore, a NAT (network address translator) device is placed on the boundary of the private address realm and the global addressing realm of the Internet and that NAT device will establish a *binding state for a flow*. The binding state has a timeout, i.e. when no data has been transferred for the duration of the timeout, the binding will be removed.

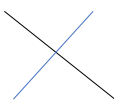
	Connection oriented	Connection-less
Packet	MPLS ATM CGE	IP/NAT IP 802.1
Circuit or Wavelength	GSM 3G Telephony WDM	

Figure 1.1 Types of networks
CGE – Carrier Grade Ethernet, WDM – Wavelength division multiplexing

In a *connectionless packet switched network*, packets carry an address or addresses that have global significance, i.e. are *globally unique* (actually this term “globally unique” is confusing because any ID or address has a finite length and thus can label only things in a certain domain). An example is the traditional Internet based on the IP protocol. For end-to-end connectivity, a sender just needs to know a globally unique address of the receiver and data can start flowing from one user to another. In practice, the current Internet has a lot of limitations for sending packets between any two Internet users and besides globally unique addresses makes use of private IP addresses as well.

The nodes of a classical IP network, called routers, do not keep flow state. There is no need, because given a globally unique destination address and a

pro-actively created routing table; the node always finds the outgoing port for each individual packet.

Figure 1.1 shows a typology of networks in terms of switching and how users reach each other. Connection oriented networks need connection (session or flow) state in the nodes for mapping incoming traffic to outgoing traffic.

There are two important examples of packet switching in modern networks. One is Ethernet switching and the other is multi-protocol label switching (MPLS). In switching, when a packet enters a switch on the incoming interface, the forwarding entry relevant to the frame or packet is looked up in the forwarding information base (FIB), possibly some change is made in the packet header based on the entry and the packet is sent to the outgoing interface again based on what was found in the FIB entry.

Ethernet has many header types starting from the basic header, one with virtual LAN ID (VLAN) to carrier grade Ethernet header types 802.1ad, 802.1ah (found in ISP networks). An Ethernet switch may be able to read all address fields as a bit string and swap any part of the header value to a new value based on the FIB entry.

In a connectionless network, instead of flow state, the *routers must have a routing table (RT)*. The routing table reflects the network state – i.e. which nodes and links exist and are in a working condition and ready to carry traffic. Typical entries in a routing table in the Internet are identified by a destination address prefix, so the entry does not usually relate to a single connected host or device. Instead, it reflects on a whole network, e.g. a stub network where traffic to many hosts will terminate.

Note: the above is usually true, but it is up to the admin to configure any given network and the routing protocols in the way needed for that network.

Managing the forwarding entries

(1) The (flow) state can be configured and monitored by a separate element and *network management system* while the nodes stay relatively dumb. This is the case for example the so-called Carrier Grade Ethernet (CGE) transport networks. A **Software Defined network** is a kind of generalization of this case.

(2) The state can be created dynamically by *signaling* like in PSTN, ISDN and circuit switched GSM. This is possible also in ATM although, most times ATM is used in the first mode. Yet another example is that NAT flow state could be managed by an explicit signaling protocol.

(3) There can be an adaptive algorithm that conspires to create and maintain the state without users knowing about it. This approach is most typically used in IP Network Address Translators (NAT). A NAT function resides on the boundary of a private and the public IP network. One can also state that the 3rd option is a special case of signaling. We call this kind of signaling, that is embedded in the normal message pattern, *implicit signaling*.

The first approach is feasible if the connections do not change too often and the number of communicating parties is not too high.

Signaling is a reasonable approach if the duration of connections is from several seconds to minutes or hours. The shorter the connections are the higher is the ratio of signaling bits to user's data bits. At some point this overhead grows too high. This is the case for example in Web browsing: if an object download would require first setting up a connection (which it does not), this would lead to very high overhead and slow operation. In such a case, the most reasonable approach is to use a connectionless network. Also, the adaptive approach e.g. using Network Address Translators is efficient enough to support short flows such as are typical of web browsing. Signaling scales to any number of destinations provided that a suitable addressing format is available.

Flow state

A packet flow in the Internet is often defined with the 5-tuple:

- (source-IP; dest-IP; source-port; dest-port; protocol), in addition there may be a time limit, ie, the flow state may be removed automatically if there are no packets in the flow for the duration of the timeout.
- If instead of a specific dest-IP, a prefix is allowed, in MPLS context we talk about an equivalence class
- NB. A host is quite free to choose the source port, so network node controls on the flow very rarely can make use of the source port

Flow state entry in a node may be created dynamically (e.g. NAT) or by network management of SDN controller.

If a network node is serving a limited/fixed set of hosts and the node has some means to verify that the flow state is for its own served host, scalability of the flow state management is fine. But if a node is serving any Internet hosts, creating flow state must avoid overuse of the state memory: *a malicious host might be intentionally able to fill any state memory and block the service.*

In particular, if it is possible that the source address is spoofed, creating flow state is a vulnerability. For example, the classical TCP has this flaw: the specification suggests that the receiver creates state upon the first SYN packet

even before it is certain that the source address is genuine. This protocol flaw can be mitigated by a SYNProxy either on the host or in the network.

Generalization of state in network

For the purpose of Software defined networking the network state commonly used in routed IP and flow state in NAT and many other solutions is generalized in generic network state entries that are managed by the SDN approach.

The state: addresses, address masks, ports, protocol(s) and timeout.

Addresses may be included on several protocol levels and also based on placement of the interface or wavelength. As a result, one unified software architecture can manage all network state on several layers of the protocol stack.

Note that when a routing entry is created in connectionless IP, the entry must be updated on regular intervals making sure that the entry is not stale and should not be used any more. Flow state also must have a timeout: if it had not, we could have stale or “hanging state” and address, memory etc reserved for some future event that never comes.

Types of addressing systems in networks

An address in a network is an identity of a place in the network. Addresses may be allocated to network nodes and connected devices or to the interfaces that have. IP addresses are allocated to interfaces.

Figure 1.2 shows a classification of addressing systems in data networks. *Flat addressing* means that any two consecutive addresses can be allocated to nodes or interface residing anywhere in the network. For example, an Ethernet address may be allocated to a Network Interface Card (NIC) at the factory. Obviously, whoever buys them cannot choose the cards by their addresses, also any network keeps changing, devices are always added and removed. So, the Ethernet switch forwarding entries cannot be aggregated, instead one entry is needed for one device. The result is that any switch that is aware of lots of devices in the network would experience lots of forwarding entry changes over time: any change happens with any device, the entries in all switches would need to be upgraded. Maintaining the same data in many places is a hard problem in networks irrespective of higher and higher performance in the equipment.

Type	Examples	Pros	Cons or cost
Flat	Ethernet	Simple to move devices, entities with address Min configuration/OPEX	Does not scale to large/global network
Hierarchical	Internet Old telephony networks	Reduction in routing table size and frequency of changes	Address tied to topology → changes impact remote communication parties
Location independent	Mobile network	Mobility	Need authentication, translate every nr to routing nr
Fixed	Classical IP	simple	Rigid, changes are hard Scalability limited
Dynamic	Dynamic IP addresses in Internet	Only active devices need an address	Address is not an ID: receiver does not know sender → lack of trust

Figure 1.2: Addressing methods

Hierarchical addressing allows aggregating forwarding entries: one entry helps to handle traffic of many devices, potentially of millions of devices. A change in the state or presence of a single device has no impact on the network. The downside is that addressing is tied to network topology making the network rigid in the face of changes. If the communicating parties are not devices but containers or virtual machines that can easily migrate from place to place, the migration would be slowed down by the network that e.g. uses dynamic routing protocols. So hierarchical addressing and virtualization of networks do not fit together nicely.

Classical Internet advocated the idea *that IP address would also act as an ID*. When every network user was a benign friend and the addresses were fixed and globally unique, this idea was simple. Over time this idea has grumbled. Steps have been: allocation of dynamic addresses to host upon-request, network source address spoofing, private addressing, network address translation. One could remedy the lack of stable ID by having the domain name to take the role of the ID. However, a host does not have a requirement to own a domain name to be served by an IP network. Nor is DNS that holds and translates domain names to addresses a trustworthy service. So even host domain names could be added and removed at will using the Dynamic DNS and they also could be faked.

A *dynamic IP address* is allocated to a host typically using the DHCP protocol from a pool of addresses. This implies that two not too far apart sessions initiated by one host, could be using a different IP source address quite legitimately. Or that the same IP address could, after a short period of time be used by another host than before.

Source address spoofing is typically a malicious operation: a host lies about its identity to avoid being penalized for some misdeed. If the serving network

uses source address filtering, an arbitrary source address that does not belong to the subnet, will lead to dropping the packet. Unfortunately, not all networks use such filtering for one reason or the other. Hackers are particularly keen to find and penetrate computers that are being served by such lax policy networks. Note also that spoofing can apply to any UDP packet and the very first TCP packet (TCP SYN packet). However, if the host wishes to get a response from the remote end or uses TCP, the hacker cannot resort to spoofing (after the SYN packet). So, spoofing is a tool for brute force simple denial of service attacks and not really useful for more intricate attack methods.

Spoofing is an annoyingly effective tool for brute force distributed denial of service attack: since the IP network does not have flow state, the victim or the victim's network admin can not trivially find where are the packets coming from nor who is behind the attack. Routed IP networks have been trying to tackle DDoS for the past more than 20 years. The results are not entirely satisfactory. The trouble lies in the fact that such an attack is at all possible in a routed IP network.

Classless Interdomain Routing (CIDR)

In the Internet initial addressing system, 32-bit addresses were classified to 5 classes:

- A: 7 MS-bit after first 0-bit identify the network and 4 LS-bits identify the interface. Only 127 networks could be in this category and each one would be very large (about 16M interfaces)
- B: after first "10" bits identifying the B-class address structure, 14 bits identify the network (about 16 000 networks are possible). The last 16 bits identify an interface.
- C: Class id: 110, 21-bits capable of identifying about 2M networks and each network interface is identified by 8 bits: the network can contain no more than 254 interfaces.
- D: Class for multicasting: Range: 224.0.0.0 to 239.255.255.255
- E: Class: Experimental use: Range: 240.0.0.0 to 255.255.255.255

This led to poor usage of the address space: very few networks really needed 16M addresses; lots of companies did not need a B-class but were too big to fit into a C-class. To make better use of the 32 address bits, it was decided that in addition to the address itself, a mask would indicate now many bits identify the network part and how many are dedicated to the device interface identification.

There are two notations for the mask, using hexadecimal encoding, a mask of 16 "1"s and 16 zeroes would be FF00, where F stands for 1111. Alternatively, the same could be expressed as x.y.z.c/16.

Another example would be a network of up to about 1022 devices combining 4 consecutive C-classes: 101.102.103.0/10.

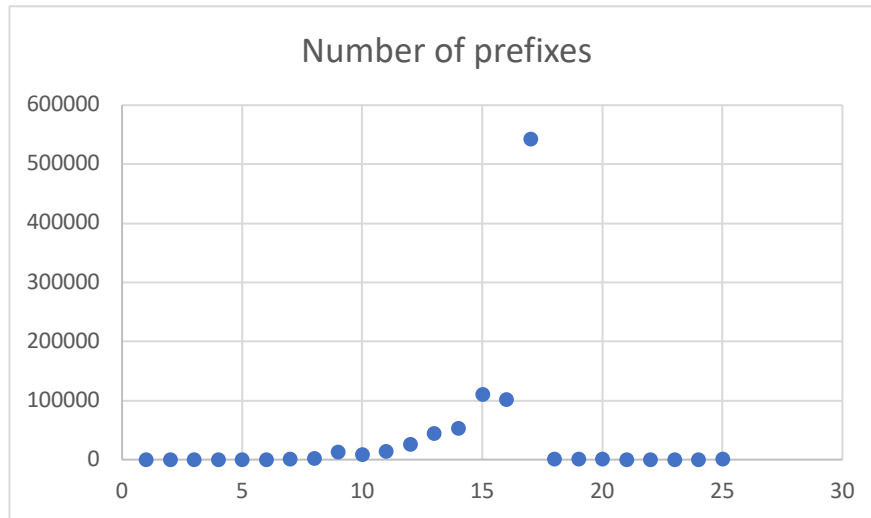
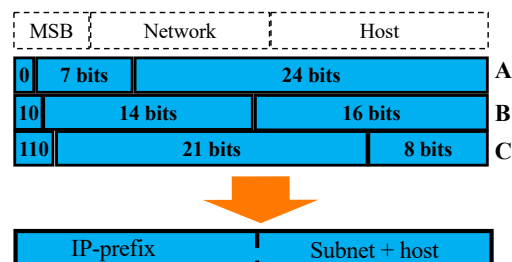


Figure 1.3: How common are different prefix lengths in BGP routing table

Figure 1.3 shows that BGP routing table today has close to 1M entries and that the prefix length varies a lot. The router must execute the *longest match prefix search algorithm* for every packet: to do so, it must compare the destination address in the packet to every entry in the Forwarding table. Over time fast and highly specialized algorithms have been developed for this purpose. The challenge is that the algorithms are power hungry, routers with multi-terabit backplanes consume quite a lot of electrical energy and their growth is limited by the generated heat. (Also, they are noisy due to strong ventilation).



- A sequence of C-class networks can be represented:
 $194.51.120.0 - 194.51.127.255 =$ (8 C-class networks)
network = 194.51.120.0
mask = 255.255.248.0 or /21

Figure 1.4: CIDR address arithmetic.

CIDR like any other hierarchical addressing system ties the addresses to network topology making the network rigid in the face of re-addressing or any address changes. Admin cannot easily take an address with a device and move the device to another subnet with the idea that remote parties would not need

to adapt to the address change. In practice the address change needs to be propagated to the network and also to the remote communicating parties.

Some routing metrics for packet networks

Routing metric is the optimization criterion for choosing the route or path through the network. Optimization may be targeting the optimum either for an individual user or for the network admin. The latter can be seen to represent the common interest of all the users combined. The network can use either a single metric or multiple metrics. In hop-by-hop routing, each node on the path makes its independent routing decision. For avoiding loops, each node on the path must use the metrics in the same way and avoiding looping is the absolute goal of routing.

Table 1 gives some metrics used in packet networks and the pros and cons.

Metric	Pros	Cons	Where used
Hop count	Stable, changes only on link/node failure	Compromise: cannot optimize for delay or BW	Most common metric in IP networks
Best Delay	Can make a diff between SAT links vs. landline	Rough measure	To avoid fall back links under normal conditions
Delay	Can try to optimize for Zoom/Teams sessions	Changes fast, distributed implementation not stable	Rare
Best bandwidth	Can optimize for large files transfer	Competes with hop count; maintaining two parallel metrics increases OPEX	
Residual bandwidth	Best for provisioning	Needs centralized implementation	Provisioning with network mgt or SDN
Min energy	Save OPEX		

IPv6 – solution for future Internet?

The official story by IETF is still that after the current IPv4, we move the Internet to use IPv6 with 16 octet or 128 bit addresses so we can have

sufficient number of addresses. Work on IPv6 draft standard was reached in 1998 and it became Internet standard in 2017. Usually, nothing in networks that takes this long, will finally succeed. Unfortunately, IPv6 shares many weaknesses of IPv4 and moving to it has turned out to be weakly motivated.

A bit more than 10 years ago, the topic of *Future Internet Architecture* became an important direction of research in networking. One of things that has emerged from this work is SDN – Software Defined Networking. Another example architecture that is not so well known but that holds a lot of promise is SCION – Scalability, control, isolation on future networks.

Software Defined Networks (SDN)

The classical Internet was based on the ideas that (a) the network has no flow state and (b) each node is autonomous leading to a distributed design of control functions and capability to recover on network level from link and nodal failures. These ideas can be also summarized that “routing is good – while switching is to be avoided”. It is a part of the picture that user data and network control messages are carried over the same links and the base network is not giving any preference to control traffic over the user traffic. Similarly, there is no systematic separation between “user plane” and “control plane”.

When the classical Internet design almost by accident escaped from the Lab (the University world) to the realm of commercial use, its weaknesses have started to emerge. We can list at the least the following weaknesses in routed IP:

- a) IPv4 addresses are just 32 bits long so there are not enough addresses for all potential users.
- b) The addresses are not trustworthy.
- c) IPv6 has been proposed and is promoted as a replacement but it has turned out to be difficult to move to the new version: it forces existing users and operators (with no growth in number of subs in sight) to invest time and money for the sake of some future potential customers of other operators in other countries.
- d) Core routing tables (RT) in IPv4 Internet have grown to become large. The RTs consume fast power-hungry memory in routers making them more and more expensive. The growth of the RTs depends on what the users want like more reliable connectivity (multi-homing) to the network. IPv6 does not help with this issue, to the contrary, the need for power hungry memory becomes even more pronounced.
- e) The core protocol (IP) does not support mobility directly.
- f) NAT traversal is cumbersome and existing methods are not well suited to wireless devices.
- g) Source address spoofing is still possible although many technically valid solutions have been proposed and some of them are even widely

supported by equipment vendors: once again investments/costs and benefits do not fall into the same hands.

- h) Distributed denial of service attacks (DDoS) are possible and make it impossible to give strong guarantees of services being available in a predictable manner. Both spoofing and DDoS are inherent features of the traditional Internet.
- i) It is difficult or impossible to guarantee quality of service (although solutions have been proposed and even implemented, none of them really work for the interdomain routing case).
- j) Routes can be hijacked in BGP routed networks like the Internet.
- k) All security is heuristic. A security proofs for wide area communications are really not feasible.
- l) Etc.

Contrary to the classical Internet design, SDN assumes that (a') "there shall be some flow state" and (b') centralized design of some network functions is fine and easier than distributed design. Also, the starting point for SDN is that control and data planes are separated. *Data plane carries the packets from a host to another host while the control plane carries the packets that are needed to control the state of the network.*

A motivation for SDN comes from the fact that more and more of the services we consume on the Internet are provided from the cloud. This means that of the two hosts in the basic network mediated interaction, one is the user's machine and the other resides "in the cloud". This implies that the identity of the last physical machine is unimportant – only the content or the service that is provided is important. To make services provisioning efficient and to improve the quality of experience of the user, it makes sense that the network somehow conspires to provide the service from a host as close to the user's host as possible. Since the server is often a container, it can be migrated easily. For this to be true, the cloud part of the network cannot use address aggregation tying containers to a placement in the cloud. So, the cloud part will use flat addressing (although the addresses are IP addresses). From this initial use case for SDN, it is possible to see a path for bringing the SDN also to corporate networks to make them more cloudification friendly.

The concept of SDN is already used in data centers and most modern corporate networks. For wide area networks it is being adopted in 5G.

There are several competing switching technologies under the term SDN. Open Flow was the first to take off in a big way although there is an older protocol called **Forces** from IETF that allows to separate the forwarding plane and the control plane in a router. Another contender is **P4**. If OF is based on the model that we can implement a data plane with a fixed set of operations (actions that the switch must be able to execute), P4 allows free programmability of packet processing for the use case.

SCION – briefly

SCION comes from ETH Zurich. It has a very well working open source implementation and it is already in use by for example the Swiss central bank etc. It however, does not have an official specification yet. A Foundation is being set up to write the specifications.

SCION breaks the wide area network into isolation domains (ISD) that are made of several autonomous systems (AS). A SCION address consists of a triple: ISD:AS:Host and communicating hosts can actually have different types of addresses that need to be just locally significant. Global uniqueness is ensured only for the triple. Roots of trust are maintained by ISD and there are no global roots of trust like on the Internet today. Operation of ISD or routing within ISD cannot be influenced by parties outside the ISD. Route hijacking that is possible on the Internet has been eliminated in SCION.

Routing over SCION is *AS-level source routing* and the path in the packet header is cryptographically protected so that it cannot be faked. Routers do not need routing tables because the path information is in every packet. Packet headers are longer than in IPv4 and even in IPv6 but routers are simpler and less power hungry than IP routers.

The goal of SCION is to offer provable security of connections over the wide area. DDoS mitigation is embedded. Also, wide area Quality of Service is supported, something that has never succeeded in IP networks.

Scope of our discussion on this course

We first cover the state-of-the-art IP Networking briefly starting with addressing the routing principles, then a specific topic that has lot of traction in networking practice, namely NAT, then Interior and Exterior routing. After that the course will focus solely on Software Defined Networking that has been widely adopted recently in data centers, is being adopted in corporate networks and now also in 5G.