Cybersécurité des systèmes maritimes et portuaires
The network layer principles

Isabel Amigo

2021

Notes:

Today's (and tomorrow) objectives

First pass

- understand the main objectives and challenges of network layer
- understand and distinguish key network functions: routing and forwarding
- get the feel of routing algorithms
- get the main characteristics of datagrams and virtual circuits

Second pass

- understand addressing in the Internet: IPv4 and IPv6
- in depth look to routing and forwarding:
  - in-class example
  - lab interdomain routing: OSPF

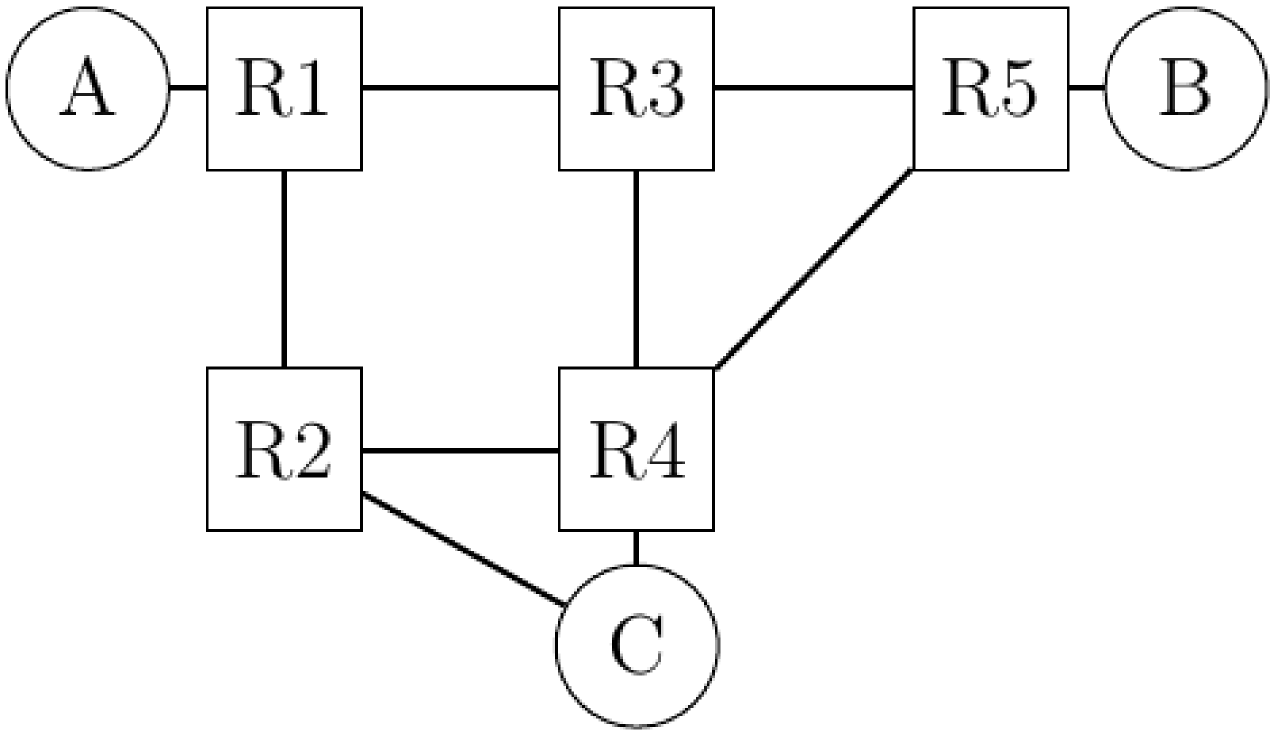
Notes:

1. Main objectives of Network layer

Notes:

Network layer enables transmission of information between hosts not directly connected

Example



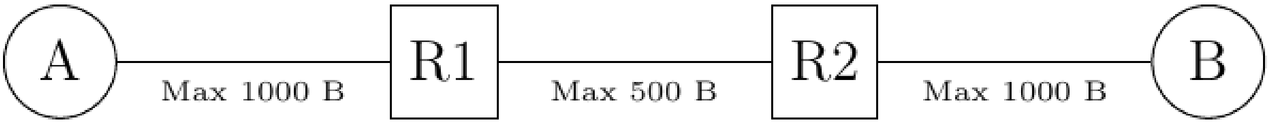
Notes: Main objective of the NL: allow hosts, connected to different networks, to exchange information through intermediate systems (routers).
Packet: unit of information in the network layer
Example:

- We consider a simple network, and imagine we want to transfer a byte from A to B (so we do not have MTU problems)
- We also consider single-homed hosts in order to simplify
- A sends the packet to R1, but R1 needs to know that the packet is not for it but for B!
- Each interface has to be uniquely identified ⇒ address usually fixed-length addresse represented by a sequence of bits
- To send one byte of information to host B, host A needs to place this information inside a packet.
- Packet has data + control information: either a) the addresses of the source and the destination nodes or b) information of path towards destination

Network layer is also responsible for dealing with heterogeneous datalink layers

Example

A wants to send a 900 bytes packet (870 bytes of payload and 30 bytes of header) to host B via router R1. Host A encapsulates this packet inside a single frame. The frame is received by router R1 which extracts the packet. What happens next?



Notes: Hosts connected to different DL exchange packets via routers that are using other types of DL. Thanks to the NL, this exchange of packets is possible provided that each packet can be placed inside a DL frame. This is simple if all DLs support the same frame size. But more complex if this is not the case, since the packet can not be encapsulated in a frame. Consider the example. R1 has three possible options to process this packet.

1. rejects the packet and sends a control packet back to the source (host A) to indicate that it cannot forward packets longer than 500 bytes (minus the packet header) (plus action possible by source)

- fragment the packet into two parts. possibilities with different pros and cons.
    2. R1 fragments the packet before transmitting them to R2. R2 reassembles the two packet fragments in a larger packet before transmitting them on the link towards B.
    3. Each of the packet fragments is a valid packet that contains a header with the source (host A) and destination (host B) addresses. When R2 receives a packet fragment, it treats this packet as a regular packet and forwards it to its final destination (host B). B reassembles the received fragments.

Two Possible organizations of the Network Layer

- Datagrams (packet-switched networks)
  - No 'call' set-up, each packet is independently forwarded
  - No resource reservation
- Virtual circuits
  - 'Call' set-up, a circuit is established before data transfer
  - Typically allows resource reservation

! Concern in previous slide (fragmentation) typically in datagram mode. Why?

Notes: Datagrams, inspired by the organization of the postal service. Each host is identified by a network layer address. To send information to a remote host, a host creates a packet that contains:

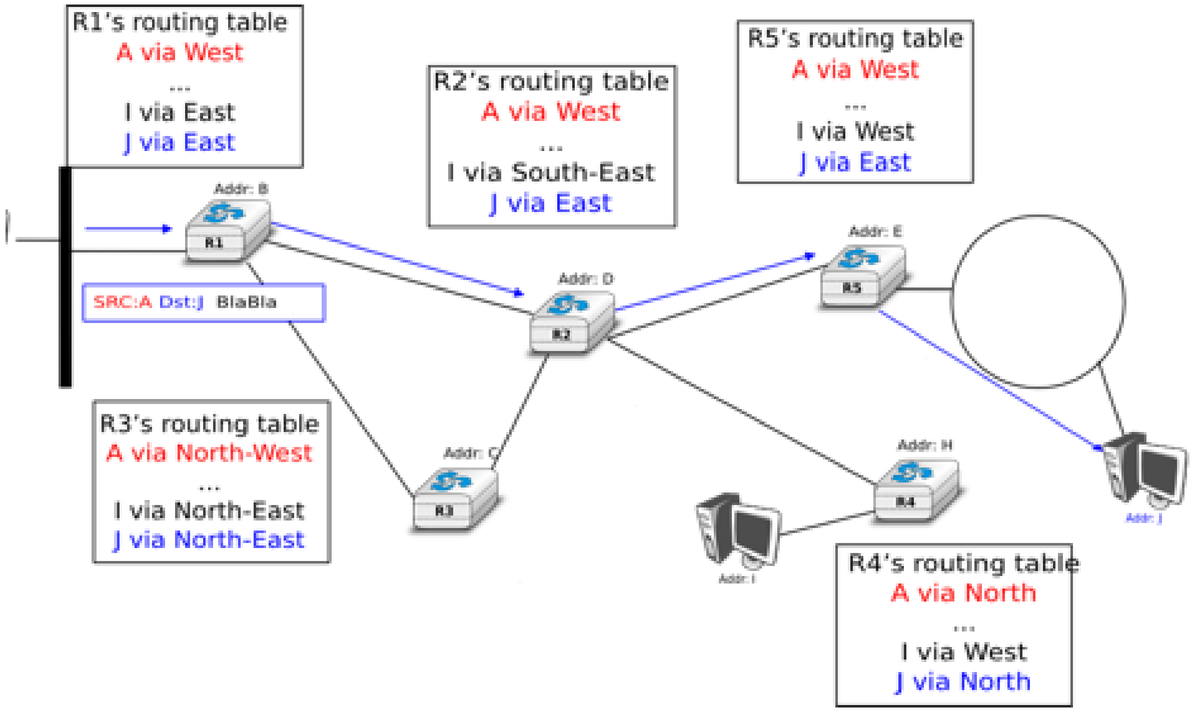
- the network layer address of the destination host
- its own network layer address
- the information to be sent

A datagram is the unit of information of packet-switched networks. Term defined in RFC1594 as"“A self-contained, independent entity of data carrying sufficient information to be routed from the source to the destination computer without reliance on earlier exchanges between this source and destination computer and the transporting network.” "

2. Datagrams

Notes:

Hop-by-hop Forwarding: *using* forwarding tables to send packets



Notes:

- Routers use hop-by-hop forwarding.

  When a router receives a packet that is not destined to itself, it looks up the destination address of the packet in its forwarding table.

- Forwarding table: data structure that maps each destination address (or set of destination addresses) to the outgoing interface over which a packet destined to this address must be forwarded to reach its final destination.

- Router consults its forwarding table to forward each packet that it handles.

- Another similar concept is the *routing table*, some protocols like OSPF make a difference on routing table (all routes known and related information like metrics) and forwarding table (the actual table being used for forwarding). We'll see this with routing protocols.

Routing: *computing* forwarding tables

Different possible techniques, we shall focus on

- Manually
- Topology information exchange + algorithm

Computing correctly the forwarding tables is key aspect.

Q What could happen if forwarding tables accross routers are not consistent?

Notes:

- Key element for network operation: computation of forwarding tables
- Its done using either distributed or centralized algorithms
- Each algo. provides different performance, may lead to different types of paths
- In a network that has valid forwarding tables, all the paths between all source/destination pairs contain a finite number of intermediate routers.
- If forwarding tables have not been correctly computed, two types of invalid paths can occur :
  - black hole : a router that receives packets for at least one given source/destination pair but does not have an entry inside its forwarding table for this destination. Packet is discarded.
  - loops : as before the destination is not reachable from all sources, but the problem is a forwarding loop. More annoying than the black hole because bandwidth is consumed

Data plane and control plane

Network functions are typically separated in what we call the *control plane* and the *data plane*.

Control plane

- e.g. all the protocols and algorithms that compute the forwarding tables that run on routers
- simplest control plane for a network: to manually compute the forwarding tables

Data plane

- e.g. forwarding tables and the precise format of the packets that are exchanged

---

Notes:

- The forwarding tables and the precise format of the packets that are exchanged inside the network are part of the data plane of the network. This data plane contains all the protocols and algorithms that are used by hosts and routers to create and process the packets that contain user data.
- The control plane includes all the protocols and algorithms (often distributed) that compute the forwarding tables
- While there is only one possible data plane for a given networking technology, different networks using the same technology may use different control planes.

Flat vs hierarchical addresses

Flat addressing

- ☺ approach: unique address pre-configured in network interface card
- ☺ easy lookup operation in the forwarding table (exact match)
- ☹ forwarding tables grow linearly with the number of hosts and nodes

Hierarchical addressing (analogy mail system)

- ☺ allows to significantly reduce the size of the forwarding tables
- ☹ lookup in the forwarding table is more complex
- ☹ not possible to use a permanent, pre-configured address
  Q how to obtain self address when node comes up?
- 😐 the allocation of the addresses must follow the network topology ⇒ blocks

Q Which scheme do you think is used in the Internet?

Notes: Further comments:

- Flat addressing: upon packet arrival, the node only needs to check whether this address is included in the forwarding table or not (computationally easy)
- Hierarchical addressing scheme : drawback when a host connects for the first time to a network, it must contact one network node to determine its own address ⇒ packet exchanges between the host and some network nodes. Furthermore, if a host moves and is attached to another network node, its network address will change. This can be an issue with some mobile hosts.

Routing algorithms allow to compute forwarding tables

Different flavors exist:

Distance vector algorithms: relay on protocols to exchange information for running a distributed algorithm

Link state algorithms: relay on protocols to learn network topology.

Notes:

Distance Vector routing algorithms

- use a distributed algorithm to discover shortest routes towards all destinations
- main idea: regularly each router sends *routing table to their neighbours* (distance towards each known destination)
- some extra rules to reduce problems such as 'count to infinity'
- upon convergence each router has a routing table containing for each destination the next hop and a cost
- example in the Internet: BGP

Notes:

- simple distributed routing protocol. allows routers to automatically discover reachable destinations and shortest path to them
- The shortest path is computed based on metrics or costs that are associated to each link.
- Each router maintains a routing table. The routing table R can be modeled as a data structure that stores, for each known destination address d, the following attributes : the outgoing link that the router uses to forward packets towards destination d, the sum of the metrics of the links that compose the shortest path to reach destination d, a timestamp
- router regularly sends its distance vector over all its interfaces (summary of the router's routing table that indicates the distance towards each known destination.)
- Split-horizon: only contains the routes that have not been learned via this neighbor.
- split-horizon with poison: routers advertise a cost of ∞ for the dest. reached via the router to which they send the distance vector.

Link State routing algorithms

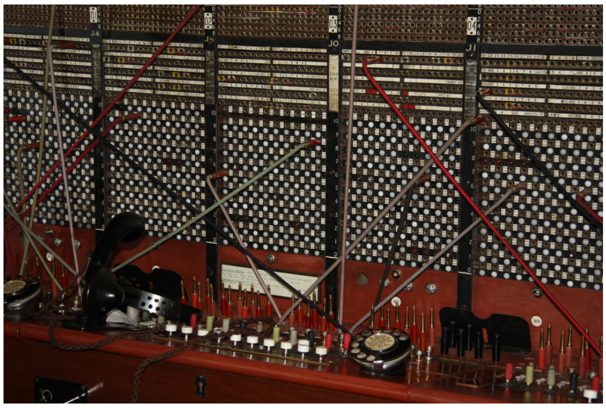
- routers exchange topology information
- regularly each router sends *the information of directly connected networks to everyone (flooding)*
- upon convergence, each router has a representation of the whole topology, and runs a shortest path algorithm (e.g. Dijkstra) to compute the routing table
- example in the Internet: OSPF
- more on Lab OSPF!

Notes:

- link-state routers exchange messages to allow each router to learn the entire network topology.
- Based on this learned topology, each router is able to compute its routing table by using a shortest path computation such as Dijkstra's
- network is modeled as a directed weighted graph. Each router is a node, and the links between routers are the edges in the graph. A positive weight is associated to each directed edge and routers use the shortest path to reach each destination.

3. Virtual Circuits

Notes:



Notes:

Telephonists (from Milton Keynes Telephone Museum) and switchboard from the manual telephone exchange at Enfield (north London). Taken out of service and replaced by automatic equipment in 1960. Now on display in the Science Museum in London.

Virtual circuits

- call setup, teardown for each call before data can flow
- each packet carries VC identifier (not destination host address)
- every router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be allocated to VC (dedicated resources = predictable service)

Notes:

What about forwarding and routing?

- hosts identified with an address
- packet forwarding based on a label on packet's header (and not on global address) and on label switching tables present at each intermediate node
- need of a signaling protocol to set-up path

Notes:

- In a network using virtual circuits, all hosts are also identified with a network layer address. However, packet forwarding is not performed by looking at the destination address of each packet.
- Each data packet contains one label: an integer which is part of the packet header.
- Network nodes implement label switching to forward labelled data packet.
- Upon reception of a packet, a network nodes consults its label forwarding table to find the outgoing interface for this packet. In contrast with the datagram mode, this lookup is very simple

Is packet-switching *better* than circuit-switching?

☺Packet-switching is great for bursty data

- resource sharing
- simpler, no call setup

☹No performance guarantees, excessive congestion possible: packet delay and loss

- protocols needed for reliable data transfer and congestion control

Q Examples of applications generating bursty and non-bursty data?

Q How to provide bandwidth guarantees, needed for some applications, on packet-switching networks?

- several “patches” exist, but till a research problem!

Notes:

Summary

Principal objectives of the network layer

- Transfer information between distant hosts, connected through routers
- Deal with heterogeneous datalink layers

For achieving such purposes:

- Addresses
- Packets

Types of network layers

- Datagrams
- Virtual circuits

Principal network functions

- Forwarding
- Routing

Notes:

Announcements

Prepare your lab!

- Introduction to Mininet $\Rightarrow$ to do at home before coming to lab
- Lab OSPF $\Rightarrow$ to prepare at home before coming to lab

Notes:

Acknowledgements

The contents of these slides is mostly based on the e-book
Computer networking: principles and protocols
`http://beta.computer-networking.info/syllabus/default/principles/network.html`

Notes: