UE PRIP
Principes des réseaux informatiques par la pratique
The network layer principles

Isabel Amigo

Automne 2023

Today's 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

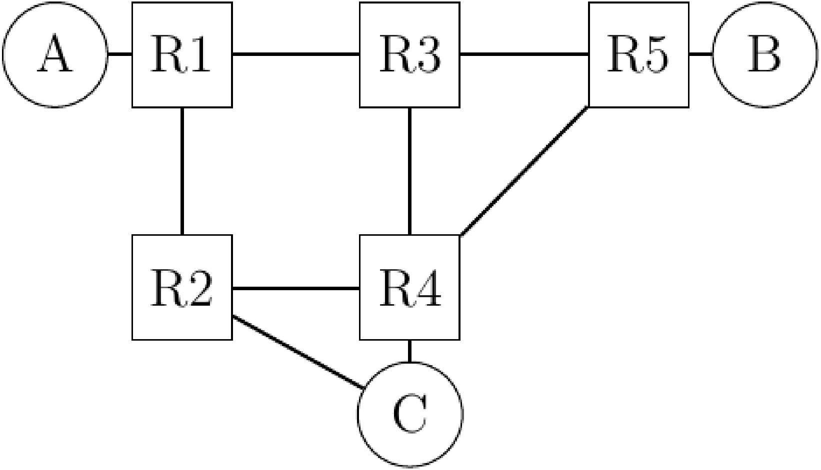
Second pass

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

1. Main objectives of Network layer

Network layer enables transmission of information between hosts not directly connected

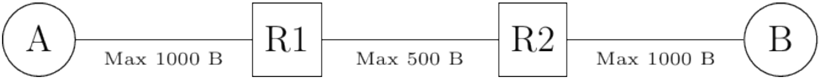
Example



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?



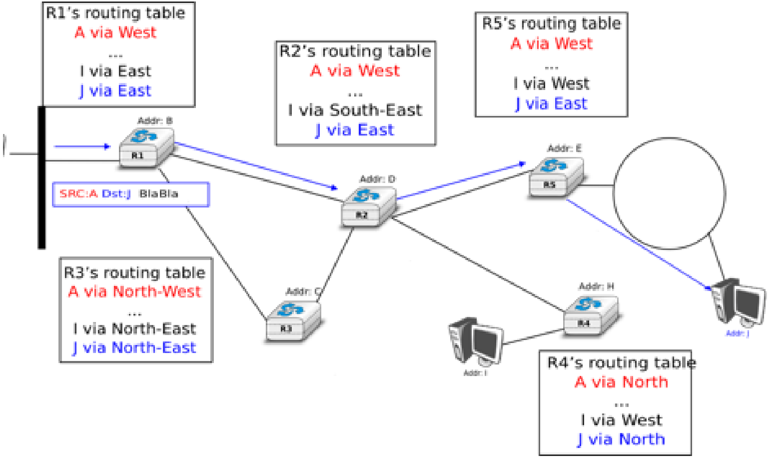
Two Possible organizations of the Network Layer

- 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?

2. Datagrams

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



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?

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

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?

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.

Distance Vector routing algorithms

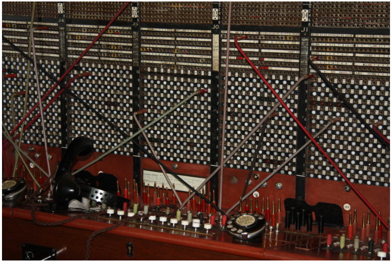
- use a distributed algorithm to discover shortest routes towards all destinations
- main idea: regularly each router sends *routing table to their neighbours* (distance towards each known destination)
- some extra rules to reduce problems such as 'count to infinity'
- upon convergence each router has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

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!

3. Virtual Circuits


15



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)

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

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!

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

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

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