Master Course
Computer Networks
IN2097

Prof. Dr.-Ing. Georg Carle

Chair for Network Architectures and Services
Department of Computer Science
Technische Universität München
http://www.net.in.tum.de

- Acknowledgements:
  - Jim Kurose, University of Massachusetts, Amherst
  - Keith Ross, Polytechnic Institute of NYC
  - Olivier Bonaventure, University of Liege
  - Srinivasan Keshav, University of Waterloo

Chapter roadmap

- Addresses and Address Mapping
- Link Layer Switches

Addresses & Naming

- Addresses are defined across three layers

- Physical / link level
  - Medium Access Control (MAC)
- Network/IP level
  - IP addresses

    ↔ mapping to domain names
- Transport/application level
  - Ports

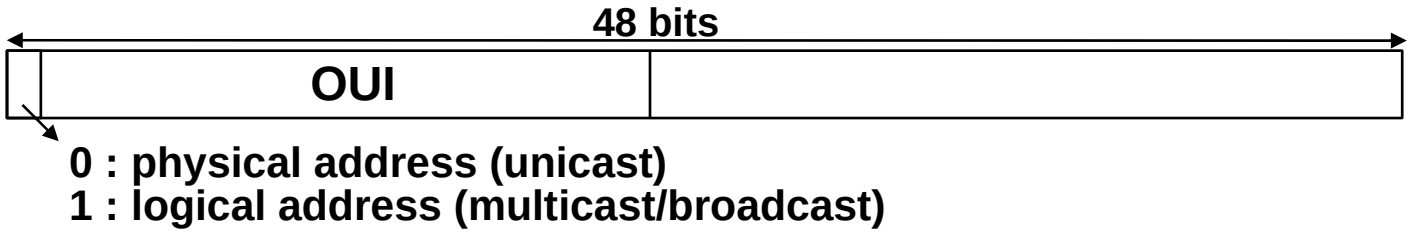
    ↔ mapping to services
    - – Standardized,well-known ports
    - – Dynamic mapping

MAC Addresses and IP Addresses

- MAC (or LAN or physical or Ethernet) address
  - L2 service: *transmit frame from one interface to another physically-connected interface (same network) with specified destination address*
  - address length: 48 bit (for most LANs)
    - burned in network adapter ROM, or software settable
    - assumption: two hosts on the same LAN will not use the same Ethernet address
- IP address: *network-layer* address
  - L3 service: get datagram to destination IP subnet
  - L3 address: has role of locator & identifier *(c.f. HIP, LISP)*
  - IP address
    - address length: 32 bit (IPv4) or 128 bit (IPv6)
    - separated into
      - network part (i.e. *network identifier & locator*)
      - host part (i.e. *host identifier*)

MAC Addressing Modes

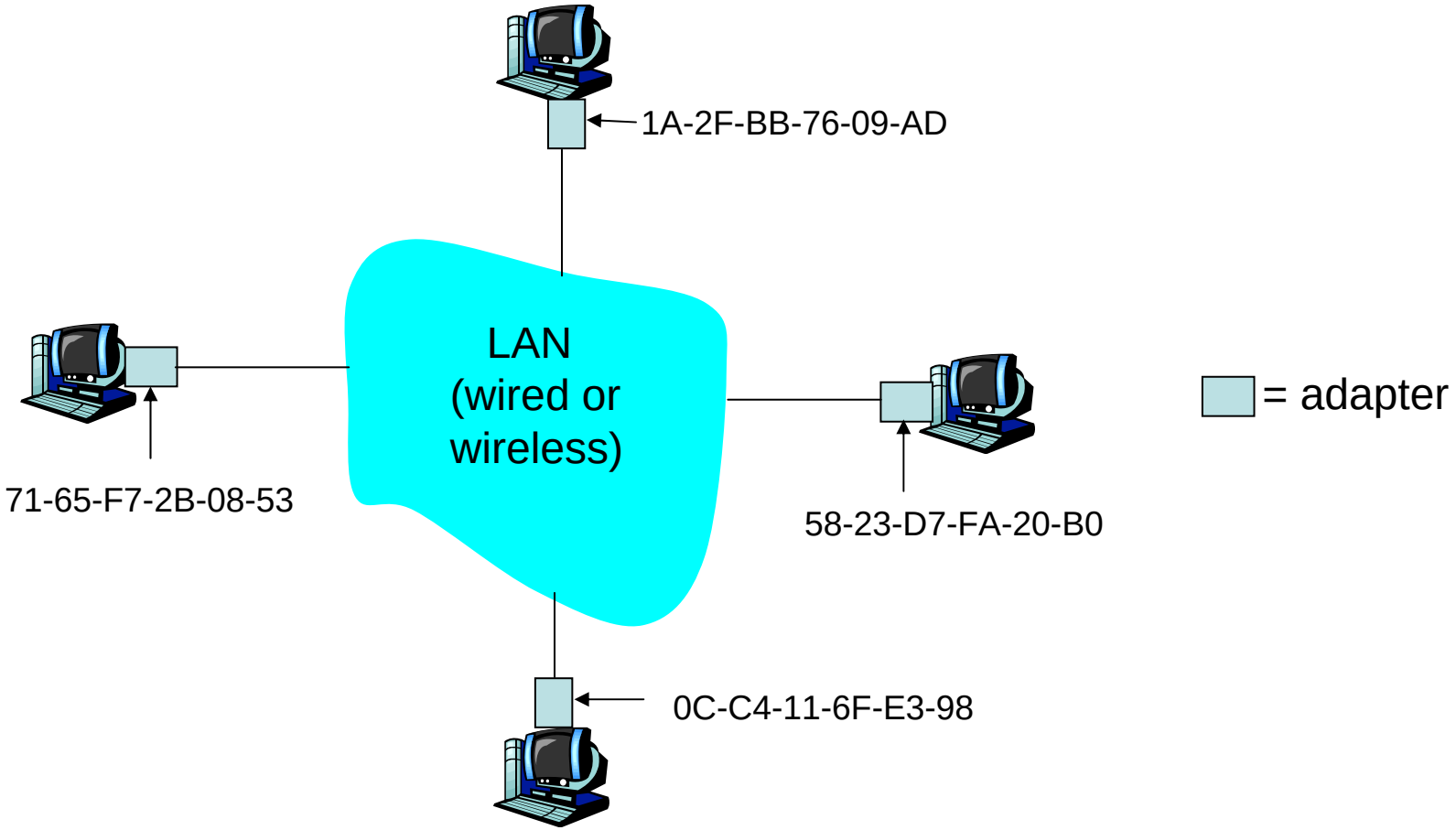
- General address types (L2 and L3):
  Unicast, Multicast, Broadcast, Anycast
- Distinguishing destination MAC addresses
  - *Physical* addresses: identify specific MAC adapters
  - *Logical* addresses: identify logical group of MAC destinations



- Transmission of multicast frames
  - sender transmits frame with multicast destination address
- Reception of multicast frames
  - Ethernet adapters can be configured to capture frames whose destination address is
    - Their unicast address
    - One of a set of multicast addresses

MAC Addresses and ARP

- Each adapter on LAN has unique MAC address



Address Resolution

- Mapping between addresses of different layers
  Examples:
  - IPv4 → MAC
  - MAC → IPv4

- Mapping from L3 host address to MAC address
  - Needed to identify correct L2 adapter of L3

  ⇨ Address Resolution Protocol (ARP)

- Mapping from MAD address to L3 address

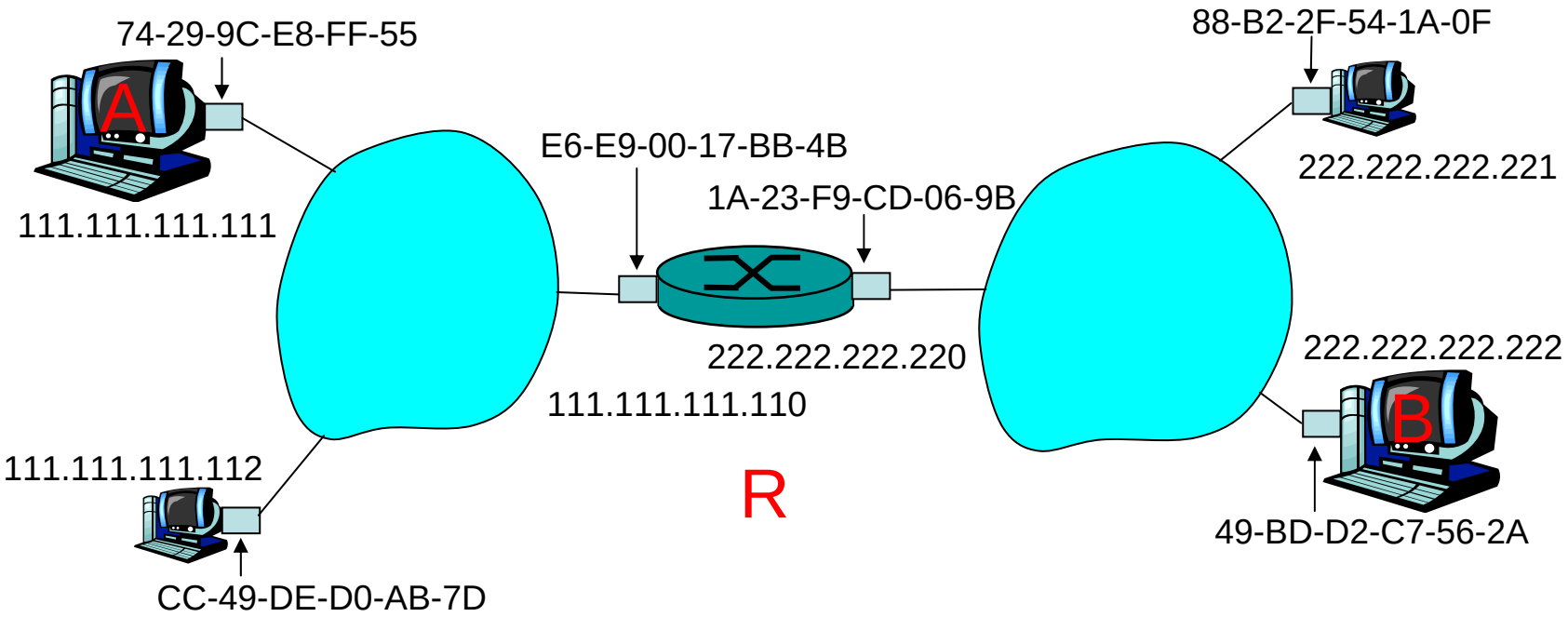
  ⇨ Reverse Address Resolution Protocol (RARP)

Addresses and Names

	Example	Organization
MAC address	8:0:20:72:93:18	flat, permanent
IP address	132.151.1.35	topological (mostly)
Host name	www.ietf.org	hierarchical

Addressing: Routing to Another LAN

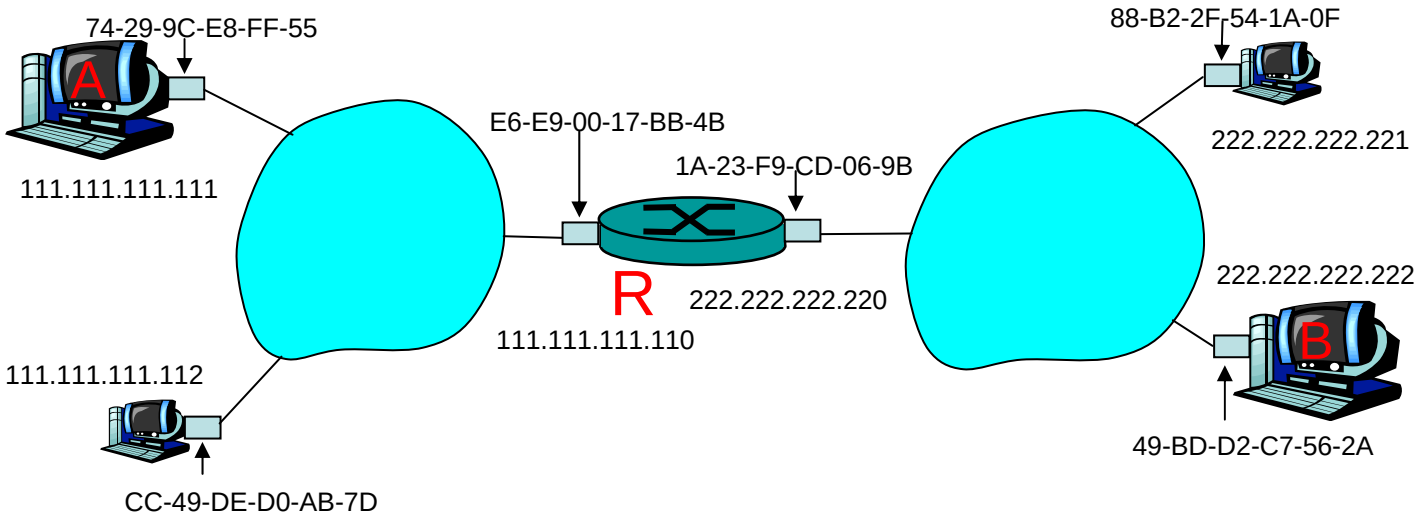
- Example: send datagram from A to B via R
  (assumption: A knows B's IP address)



- two ARP tables in router R, one for each IP network (LAN)

Addressing: Routing to Another LAN (2)

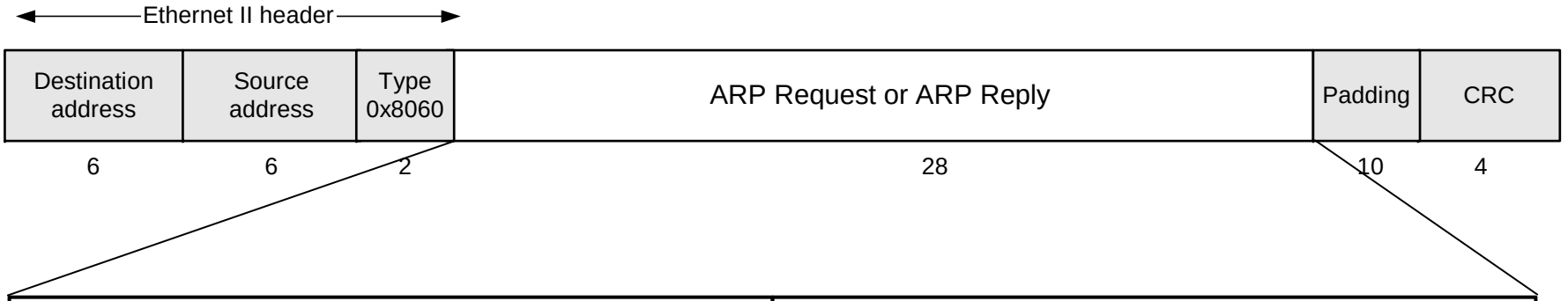
- A creates IP datagram with source A, destination B
- A uses ARP to get R's MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram
- A's NIC sends frame
- R's NIC receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's MAC address
- R creates frame containing A-to-B IP datagram sends to B



ARP Protocol: Same LAN (Network)

- A wants to send datagram to B, and B's MAC address not in A's ARP table.
- A ***broadcasts*** ARP query packet, containing B's IP address
  - destination MAC address = FF-FF-FF-FF-FF-FF
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)
- A caches IP-to-MAC address pair in its ARP table until information times out
  - ***soft state***: information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables without intervention from network administrator

ARP Packet Format



Hardware type (2 bytes)		Protocol type (2 bytes)
Hardware address length (1 byte)	Protocol address length (1 byte)	Operation code (2 bytes)
Sender hardware address*		
Sender protocol address*		
Target hardware address*		
Target protocol address*		

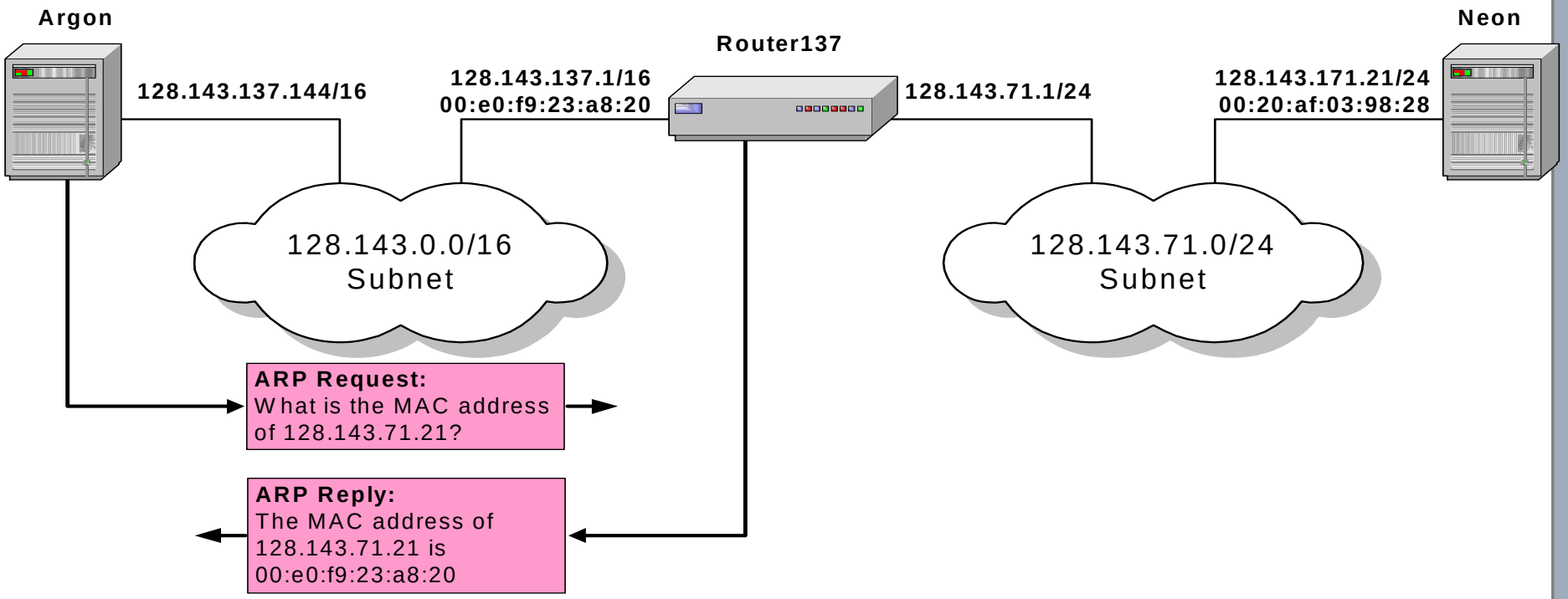
* Note: The length of the address fields is determined by the corresponding address length fields

ARP Details

- ARP supports different protocols at L2 and L3
  - any network protocol over any LAN/MAC protocol
  - type and address length fields specified in ARP PDUs
- Reverse ARP (RARP) c.f. RFC 903
- L2 MAC fields (hardware)
  - hardware type: 6 = IEEE802 (with LLC/SNAP)
  - address length: 6 for a 48 byte MAC address
  - source hardware address (SHA)
  - target hardware address (THA)
- L3 network fields (protocol)
  - protocol type: IP = 0800
  - address length: 4 for IPv4 addr.
  - source protocol address (SPA)
  - target protocol address (TPA)
- Operation Code
  - 01: req; 02: reply; 03: rev. req.; 04: rev. reply
    c.f. `http://www.iana.org/assignments/arp-parameters`

Proxy ARP

- **Proxy ARP:** Host or router responds to ARP Request that arrives from one of its connected networks for a host that is on another of its connected networks.



ARP Optimisations

- When should a host send ARP requests ?
  - Before sending each IP packet ?
    - No, each host/router maintains an ARP table that contains the mapping between IP addresses and Ethernet addresses. An ARP request is only sent when the ARP table is empty
- How to deal with hosts that change their addresses ?
  - Expiration timer is associated to each entry in the ARP table
    - ARP table entry is removed upon timer expiration
    - Some implementations send ARP request to revalidate before removing table entry
    - Some implementations remember when ARP table entries were used to avoid removing important entries

Things to know about ARP

- What happens if an ARP Request is made for a non-existing host?

  Several ARP requests are made with increasing time intervals between requests. Eventually, ARP gives up.

- Gratuitous ARP Requests: A host sends an ARP request for its own IP address:
  - Useful for detecting if an IP address has already been assigned.

Vulnerabilities of ARP

1. Since ARP does not authenticate requests or replies, ARP Requests and Replies can be forged
2. ARP is stateless: ARP Replies can be sent without a corresponding ARP Request
3. According to the ARP protocol specification, a node receiving an ARP packet (Request or Reply) must update its local ARP cache with the information in the source fields, if the receiving node already has an entry for the IP address of the source in its ARP cache. (This applies for ARP Request packets and for ARP Reply packets)

Typical exploitation of these vulnerabilities:

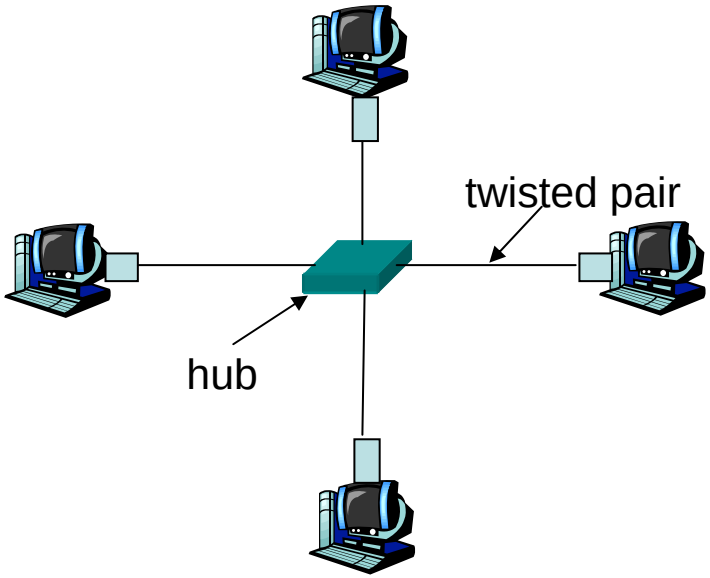
- A forged ARP Request or Reply can be used to update the ARP cache of a remote system with a forged entry (***ARP Poisoning***)
- This can be used to redirect IP traffic to other hosts

Outline

- Addresses and Address Mapping
- **Link Layer Switches**

Hubs

- … physical-layer (“dumb”) repeaters:
  - bits coming in one link go out all other links at same rate
  - all nodes connected to hub can collide with one another
  - no frame buffering
  - no CSMA/CD at hub: host NICs detect collisions

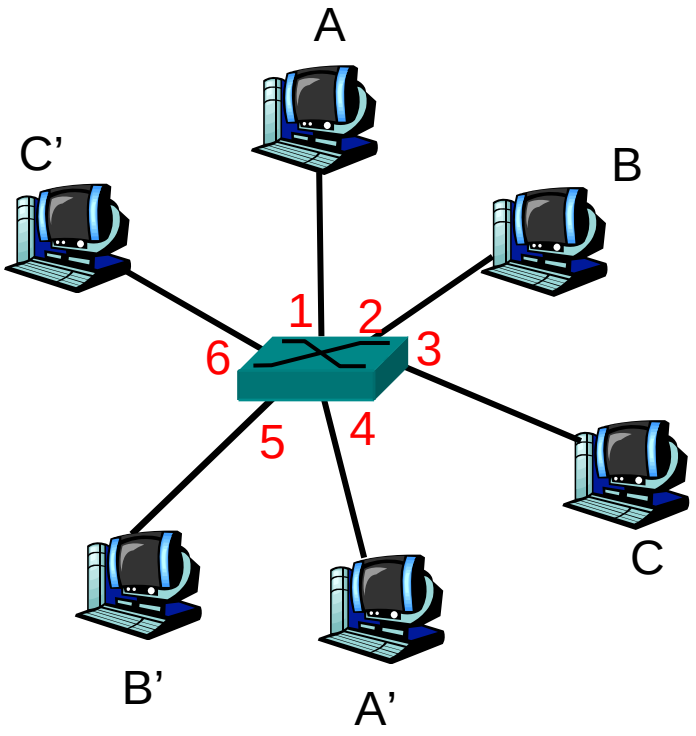


Switch

- Link-layer device: smarter than hubs, take *active* role
  - Store & forward of Ethernet frames
    or: ***cut-through-switching***
  - examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- Transparent
  - hosts are unaware of presence of switches
- Plug-and-play, self-learning
  - switches do not need to be configured

Switch: Allows Multiple Simultaneous Transmissions

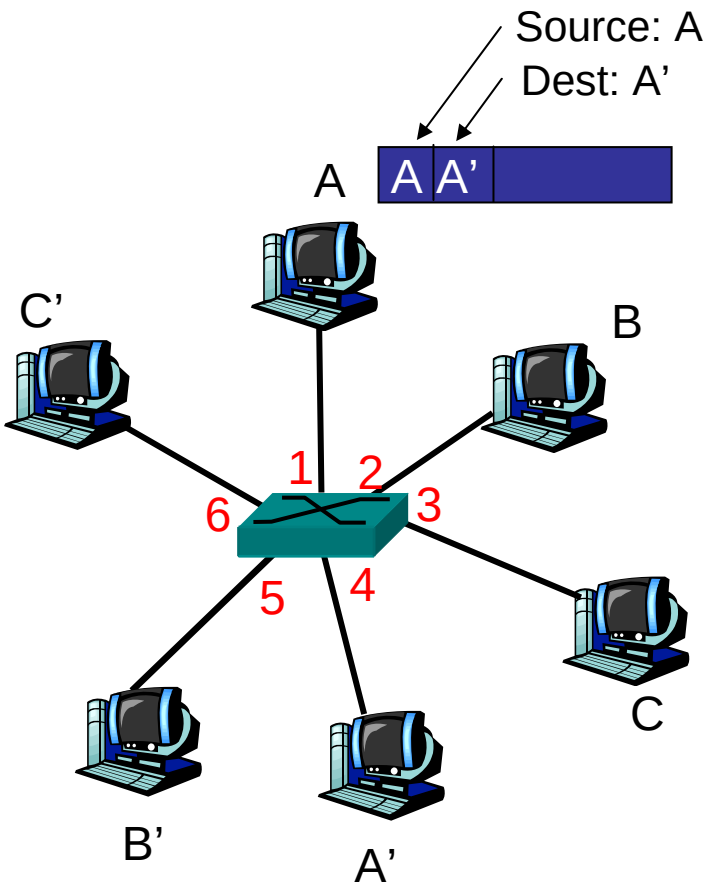
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
  - each link is its own collision domain
- *switching:* A-to-A' and B-to-B' simultaneously, without collisions
  - not possible with dumb hub



switch with six interfaces (1,2,3,4,5,6)

Switch: Self-Learning

- switch *learns* which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table



MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

Switch: Frame Filtering/Forwarding

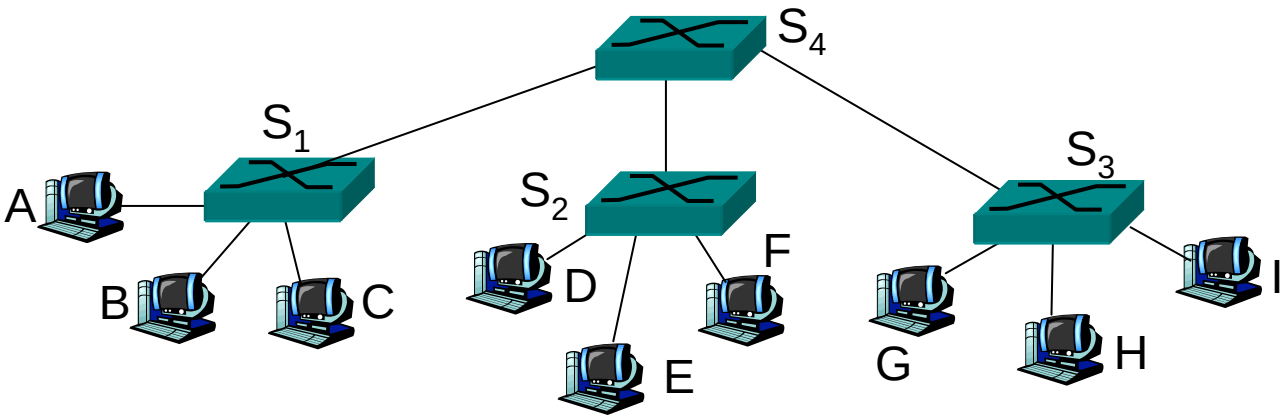
When frame received:

1. record link associated with sending host
2. index switch table using MAC dest address
3. **if** entry found for destination
   then {
   if dest on segment from which frame arrived
   then drop the frame
   else forward the frame on interface indicated
   }
   else flood

forward on all but the interface on which the frame arrived

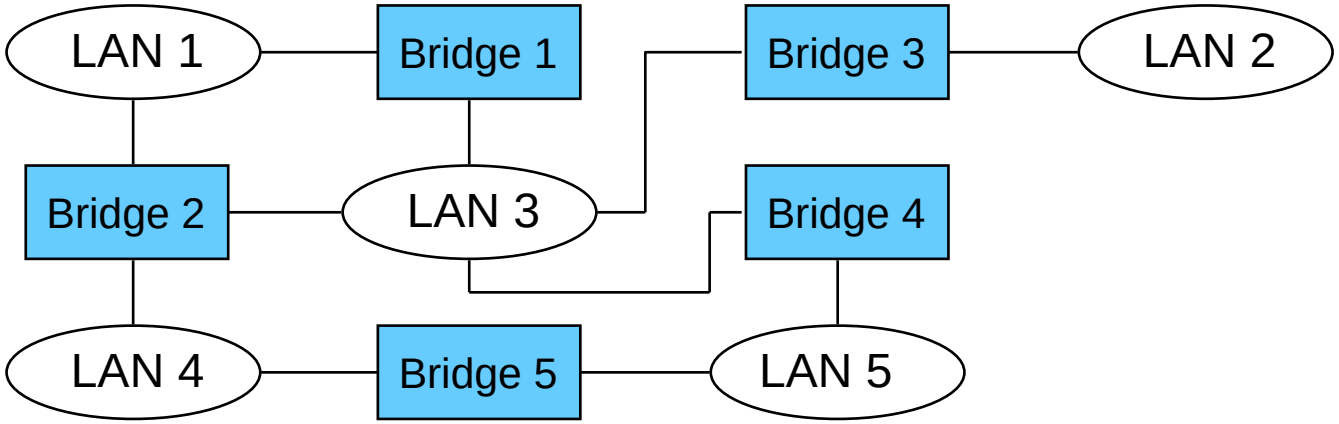
Interconnecting Switches

- switches can be connected together



- *Q:* sending from A to G - how does S_1 know to forward frame destined to F via S_4 and S_3?
- *A:* self learning! (works exactly the same as in single-switch case!)

How to Prevent Loops



- Spanning tree algorithm
  - Bridges gossip amongst themselves
  - Compute loop-free subset
  - Forward data on the spanning tree
  - Other links are backups

Spanning Tree Protocol

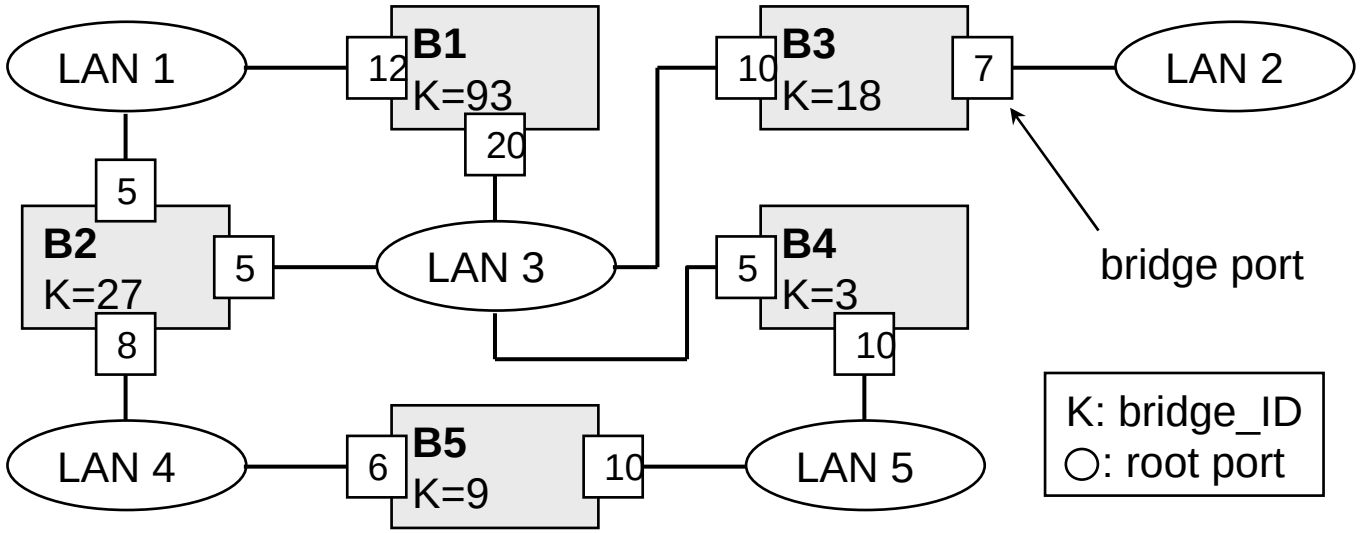
- Spanning Tree Protocol (STP): standardized as IEEE 802.1D
- Algorithm by Radia Perlman
- Algorithm:
  - Uses bridge_ID (bridge_priority + MAC_addr)
  - Step 1: select root bridge, i.e. bridge with lowest bridge_ID
  - Step 2: determine least cost paths to root bridge,
    - each bridge determines cost of each possible path to root
    - each bridge picks least-cost path
    - port connecting to that path becomes *root port* (RP)
    - bridges on network segment determine bridge port with least-cost-path to root, i.e. the designated port (DP)
  - Step 3: disable all other root paths
- Bridge Protocol Data Units (BPDUs) are sent to STP multicast address, with bridge_IDs and root path costs

Spanning Tree Protocol

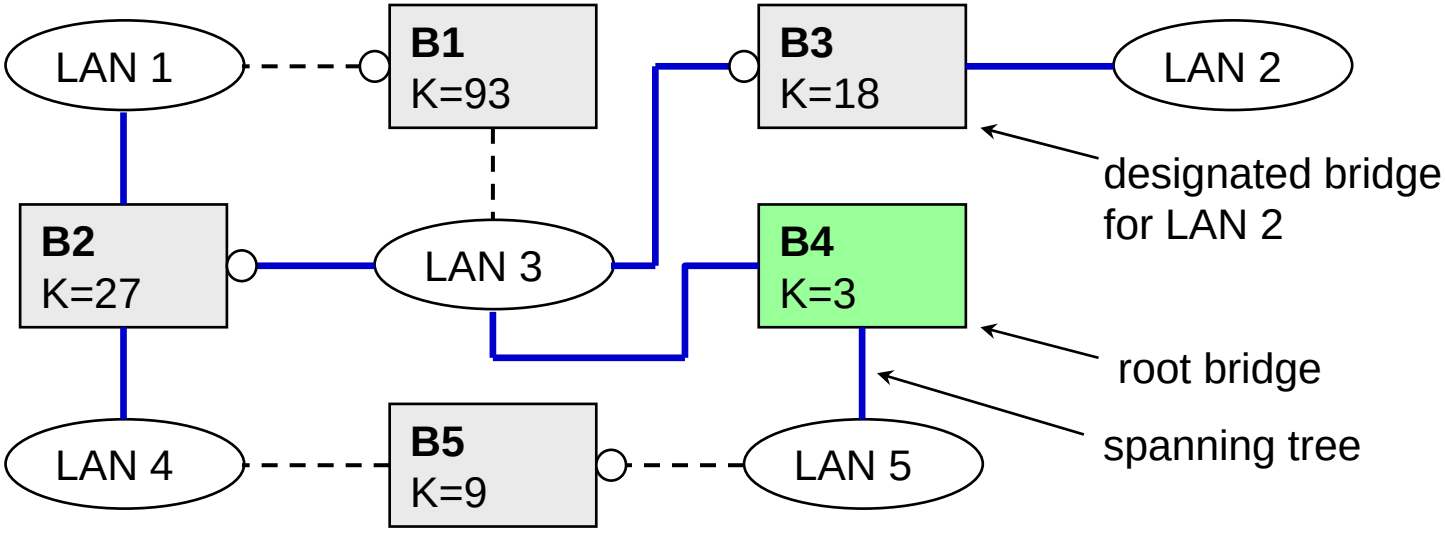
- Bridge Protocol Data Units (BPDUs)
  - are sent regularly (default: 2 s) to STP multicast address
  - Configuration BPDUs transmit bridge_IDs and root path costs
  - Topology Change Notification (TCN) BPDU announce changes in network topology
  - Topology Change Notification Acknowledgment (TCA)
- STP switch port states
  - Blocking
  - Listening
  - Learning
  - Forwarding
  - Disabled

Spanning-Tree Algorithm Example

Before:

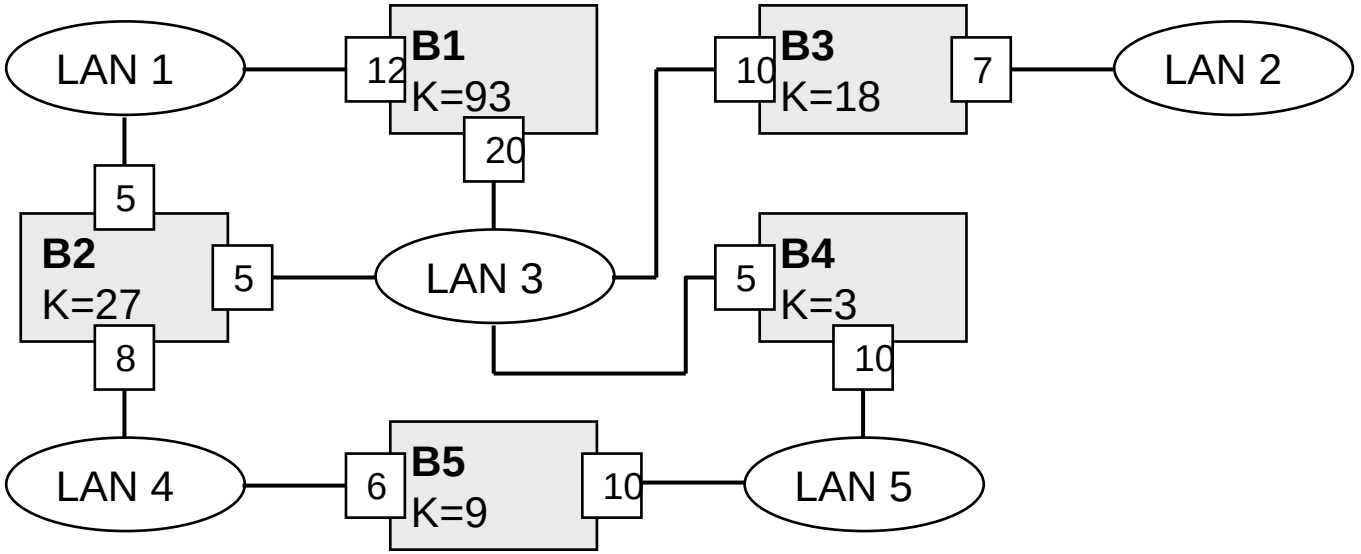


After:



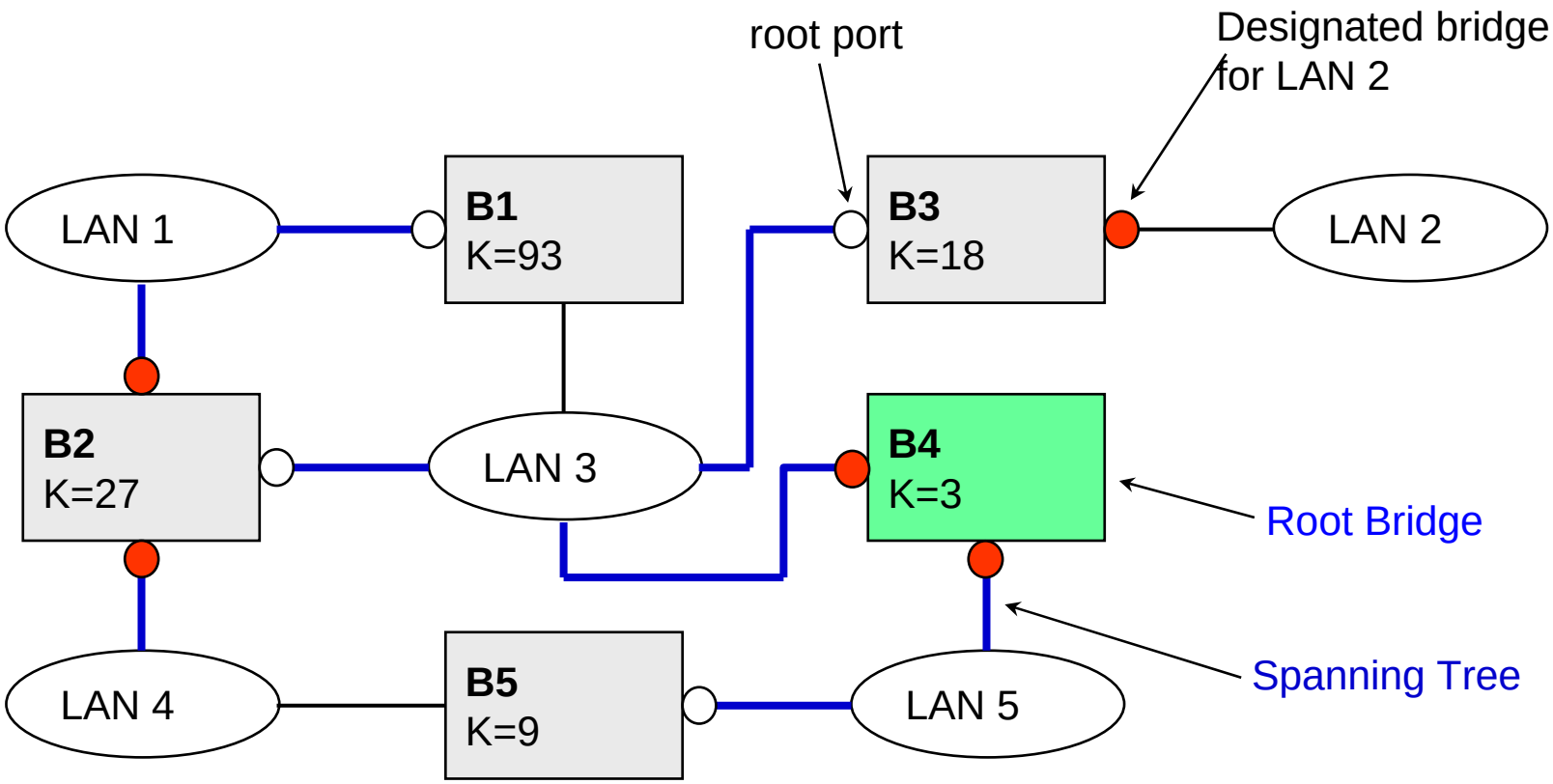
Spanning-Tree Algorithm: Root Path Costs

Before:

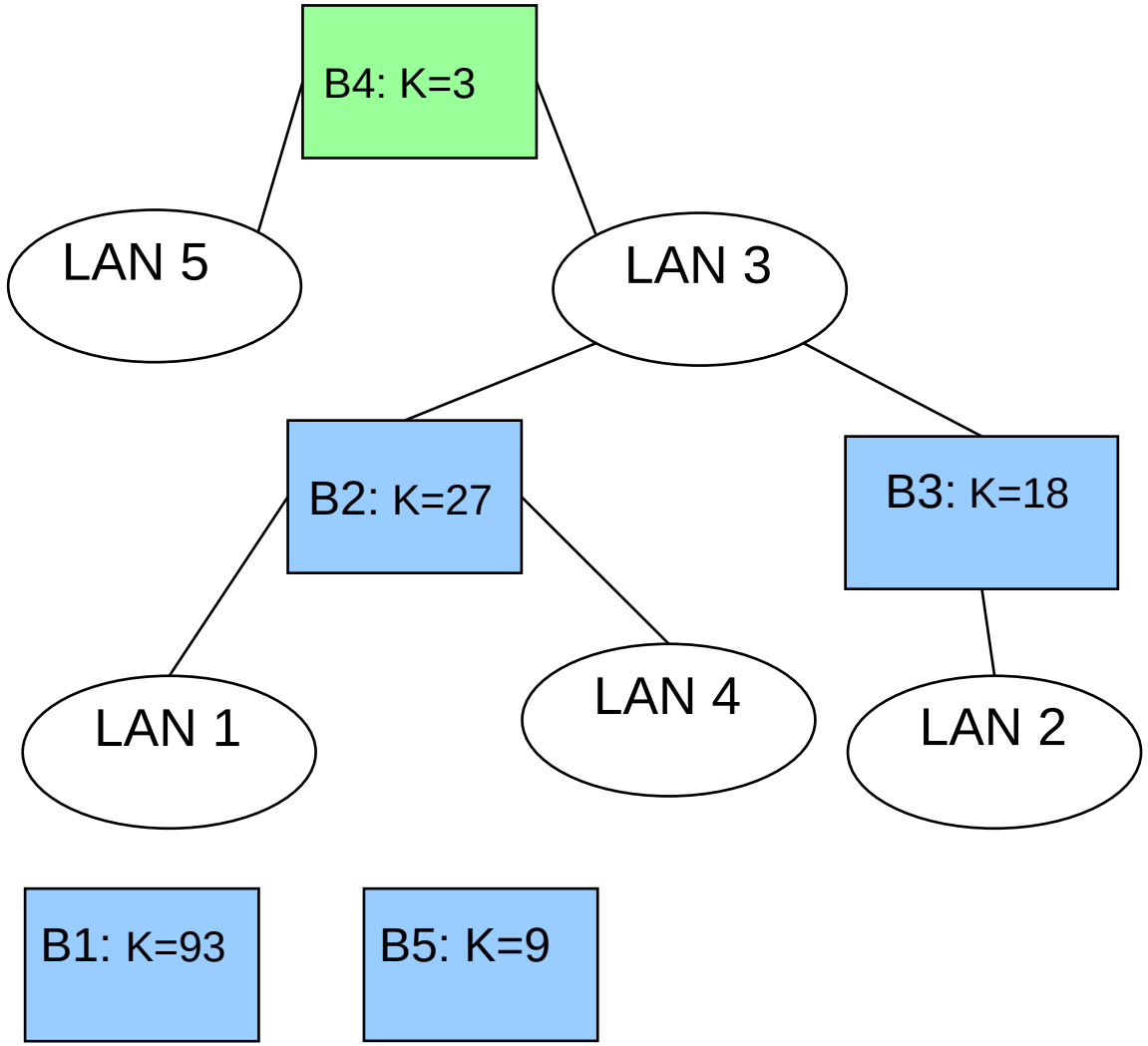


Bridge	Cost of path to root bridge
B3	**10** (via LAN 3)
B1	20 (via LAN 3)
	17 = 12 + 5 (via LAN 1 & LAN 3)
B2	**5** (via LAN 3)
	18 = 8 + 10 (via LAN 4 & LAN 5)
	25 = 5 + 20 (via LAN 1 & LAN 3)
B5	**10** (via LAN 5)
	11 = 6 + 5 (via LAN 4 und LAN 3)

Spanning-Tree Algorithm: Designated Bridges



Spanning-Tree Algorithm (Result)



LAN	Designated Bridge
LAN 1	B2
LAN4	B2
LAN2	B3
LAN3	B4 (Root Bridge)
LAN5	B4 (Root Bridge)
---	B5
---	B1

Switches vs. Routers

- both store-and-forward devices
  - routers: network layer devices (examine network layer headers)
  - switches are link layer devices
- routers maintain routing tables, implement routing algorithms
- switches maintain switch tables, implement filtering, learning algorithms