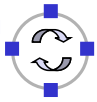
# Telematics I

## Chapter 6
## Internetworking

(Acknowledement: These slides have been compiled from H. Karl's set of slides)

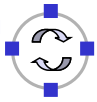




## Goals of This Chapter

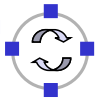
- So far: we can communicate between nodes all connected directly to the same medium

- How to grow beyond a single medium?
- What options exist to interconnect local networks into larger configurations?
  - Repeaters, hubs, bridges, switches, routers, gateways
- What are their limitations?
- How does it relate to the networking layer in the ISO/OSI stack?



# Overview

- ***LAN interconnection***
- Physical-layer interconnects
- Data-link-layer interconnects
- Higher-layer interconnects

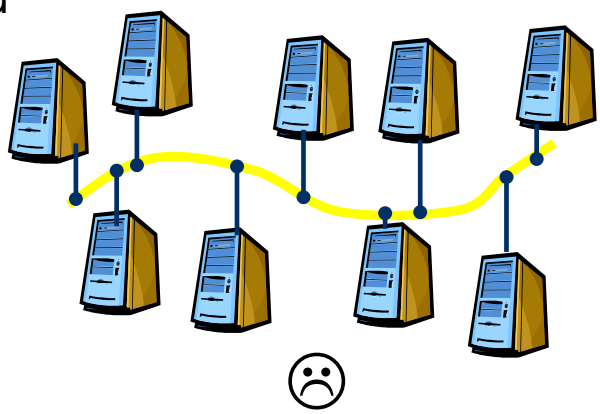




# The Problem

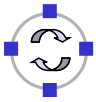
- Let's start from classic Ethernet
  - Single wire, single collision domain
- Works fine for a limited number of stations
- Collapses when number of nodes becomes too large
  - CSMA/CD will not keep up, limited bandwidth

! Multiple LANs are necessary
  - Not an inherent Ethernet problem
  - Will happen on any medium, with any protocol



# Several Reasons for Multiple LANs

- Limited number of users/throughput in a single LAN
- Historical reasons
  - Different groups started out individually setting up networks
  - Usually heterogeneous
- Geographic distribution of different groups over different buildings, campus, …
  - Impractical/impossible to use a single LAN because of distance
  - Long round-trip delay will negatively influence performance
- Reliability
  - Don't put all your eggs into one basket
  - "Babbling idiot" problem
- Security
  - Promiscuous operation – contain possible damage

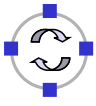




# Several Options to Overcome Some of These Limitations

- Can be classified according to the layer in which they work

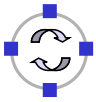
| | |
|---|---|
| Application layer | Application gateway |
| Transport layer | Transport gateway |
| Network layer | Router |
| Data link layer | Bridge, switch |
| Physical layer | Repeater, hub |



## Overview

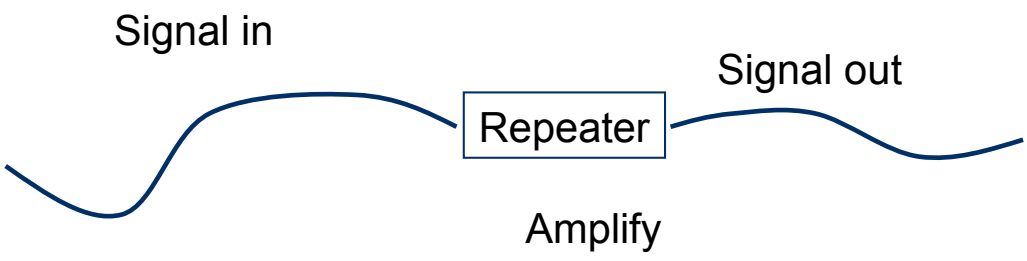
- LAN interconnection
- ***Physical-layer interconnects***
- Data-link-layer interconnects
- Higher-layer interconnects



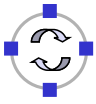


## Repeaters

- Simplest option: Repeater
  - Physical layer device
  - Connected to two cables
  - Amplifies signal arriving on either one, puts on the other cable
  - Essentially an analog amplifier to extend physical reach of a cable
  - Combats attenuation
  - Neither understands nor cares about *content (bits)* of packets

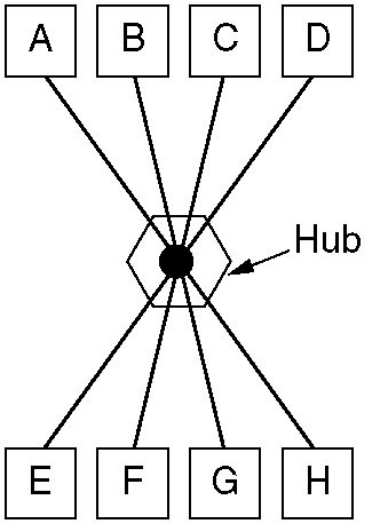




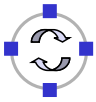



# Hub

- Similar to repeaters: Hubs
  - Connects multiple cables electrically, not just two
  - Usually, does not amplify the signal
  - Also physical layer device
  - Also does not understand or process *content* of packets
  - All connected cables form a single collision domain





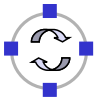



# Physical Layer Solutions Not Satisfactory

- Physical layer devices – repeater, hub – do not solve the more interesting problems
  - E.g., how to handle load
- Some knowledge of the data link layer structure is necessary
  - To be able to inspect the content of the packets/frames and *do* something with that knowledge

! Link-layer solutions
  - Bridge & switch
  - Switch: Interconnect several *terminals*
  - Bridge: Interconnect several *networks*
  - But terms sometimes used interchangeably

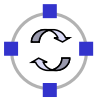




# Overview

- LAN interconnection
- Physical-layer interconnects
- ***Data-link-layer interconnects***
- Higher-layer interconnects

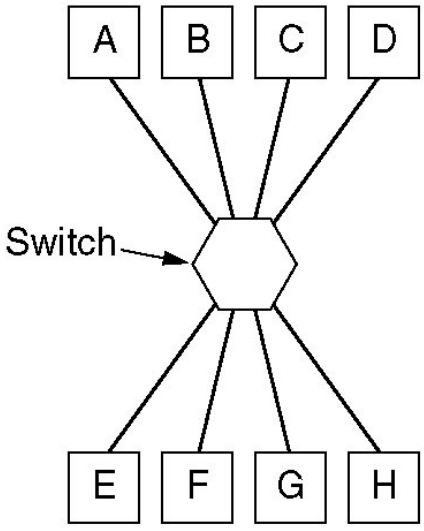




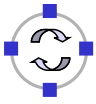


# Switch

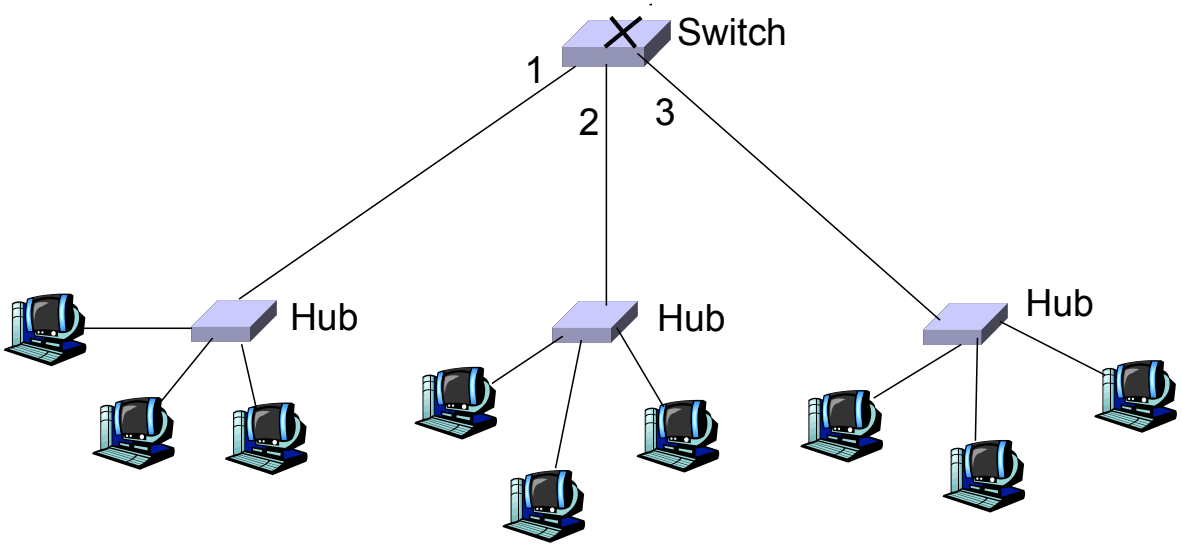
- Use a switch to connect several terminals *without* forming a single collision domain
- A switch:
  - Stores and forwards link layer frames (e.g. Ethernet)
  - When frame is to be forwarded on segment, uses CSMA/CD to access segment
  - Inspects an arriving packet's addresse and forwards its *only* on the right cable
    - Does not bother the other terminals
    - Needs: buffer, knowledge *where* which terminal is connected



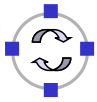


# Forwarding




- How do determine onto which LAN segment to forward frame?
- Looks like a routing problem...

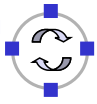






# Determining Directions: Self Learning

- A switch has a switch table
- Entry in switch table:
  - (MAC Address, Interface, Time Stamp)
  - Stale entries in table dropped (TTL can be 60 min)
- Switch *learns* which hosts can be reached through which interfaces
  - When frame received, switch “learns” location of sender: incoming LAN segment *(“backward learning”)*
  - Records sender/location pair in switch table



# Filtering/Forwarding

When switch receives a frame:

index switch table using MAC dest address

**if** entry found for destination

**then{**

**if** dest on segment from which frame arrived

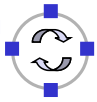
**then** drop the frame

**else** forward the frame on interface indicated

**}**

**else** flood

*forward on all but the interface on which the frame arrived*

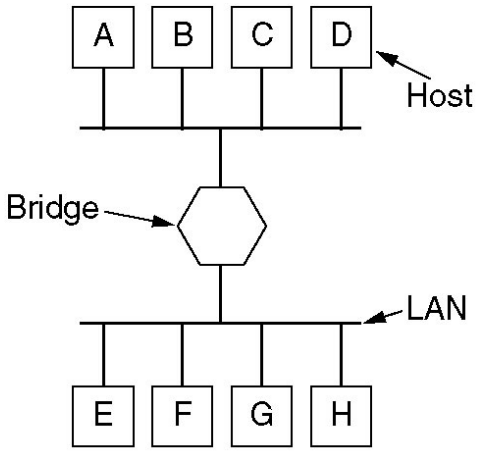


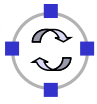




# Bridges

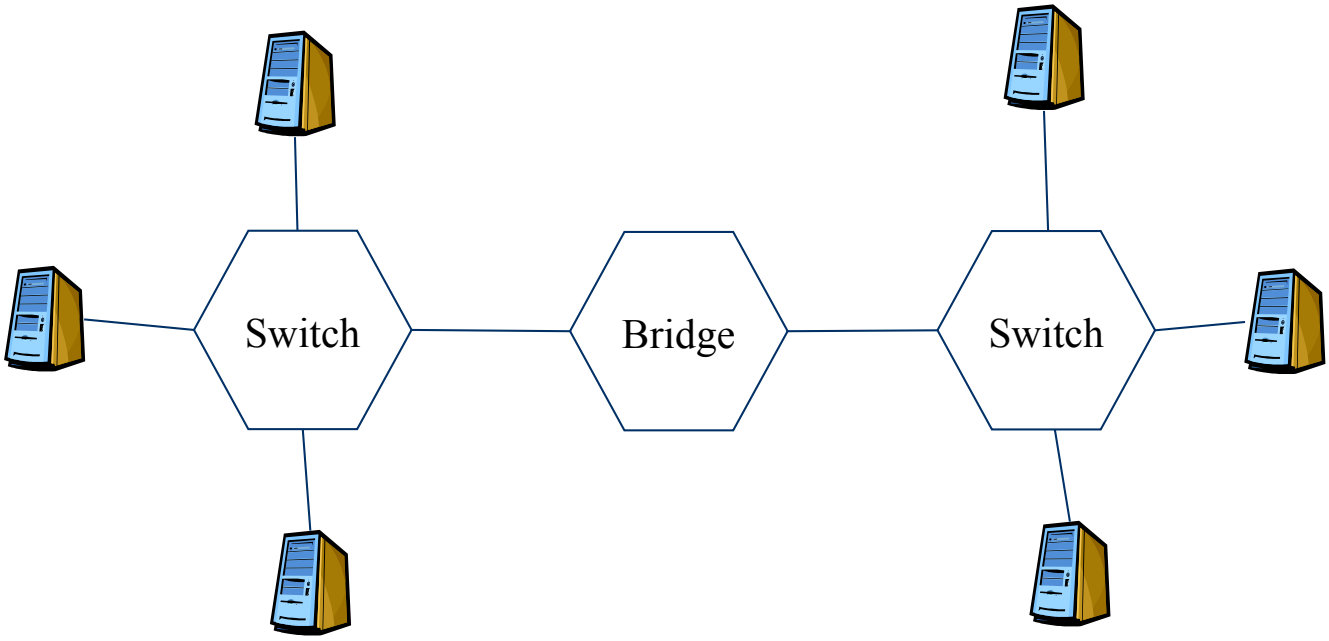
- Switches are limited in that they connect simple terminals
- Sometimes, entire networks have to be connected: Bridges
- Bridge also inspects incoming packet and forwards only towards destination
  - How to learn here where destination is? Does simple “backward” learning suffice?
- Each network connected to a bridge is a separate collision domain
- Bridges can also interconnect different LAN types
  - Not possible on physical layer only

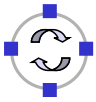


# Switches & Bridges

- Typical combination: Bridge as “just another terminal” for a switch



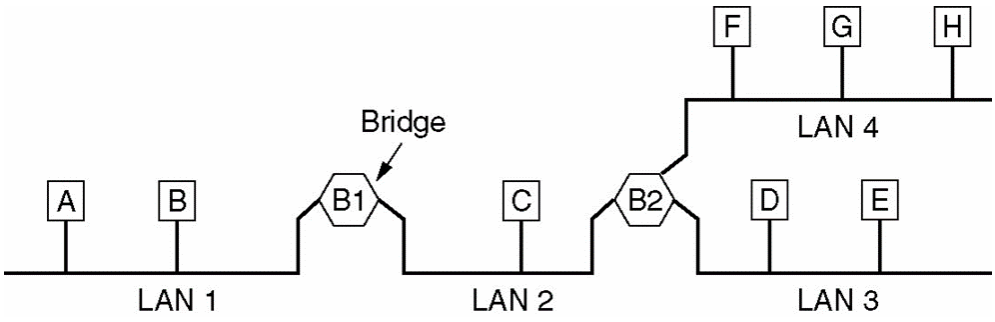



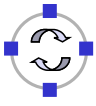




# Backward Learning in Bridges

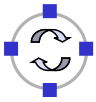
- Backward learning is trivial in a switch – how about a bridge?
- Example: A sends packet to E
  - Suppose bridges B1 and B2 know where E is
  - B2 will see A’s packet coming from LAN2
  - Since B2 does not know about LAN1, B2 will assume A to be on LAN2
  - Which is fine! B1 will forward any packet destined to A arriving at LAN2 to LAN1, so that works out nicely



## Backward Learning in Bridges – Bootstrapping

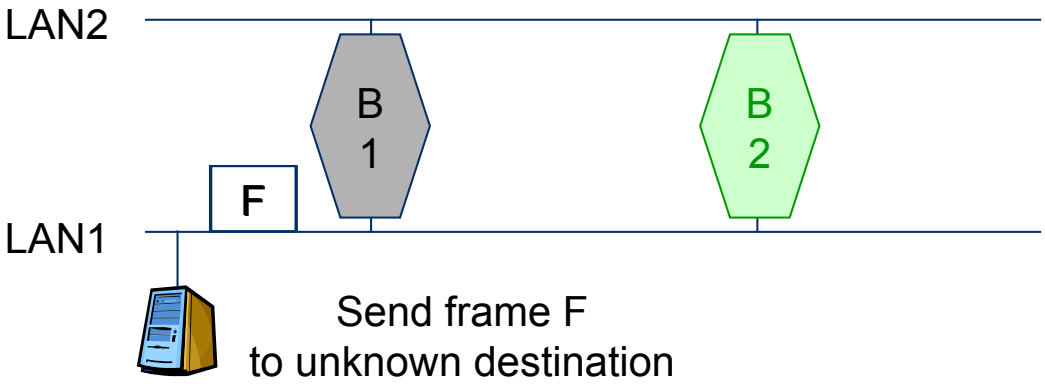
- In previous example:
  - How does bridge B2 know initially where node E is?
- Answer: It does NOT know
  - Option 1: Manual configuration – not nice!
  - Option 2: Do not care – simply forward the data everywhere for an unknown address
    - Except to the network where it came from
- Algorithm is thus:
  - flood if not known, or
  - discard if known to be not necessary, or
  - forward specifically if destination is known



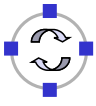


## Flooding by Bridges – Problems

- Previous “backward learning by flooding” is simple, but problematic
- Consider example topology:
  - Second bridge for reliability




- When B2 hears packets flooded from B1 it will flood them as well...
  ... and vice versa!
- How to avoid such packet loops?



## Solution 1: Somehow Restrict Flooding

- Unrestricted, brute-force flooding evidently fails
- Avoid packet looping indefinitely by ***remembering*** which packets have already been forwarded
  - If already seen and forwarded a packet, simply drop it
- Requires: State & uniqueness
  - Bridges have to remember which packets have passed through
  - Packets must be uniquely identifiable – at least source, destination, and sequence number are necessary to distinguish packets
- Big overhead!
  - State is a problem, as is time to search this amount of state
  - Usually not used

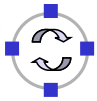
Note: Restricted flooding is still important – for control packets, in wireless networks, …



## Solution 2: Spanning Trees

- Packet loops are caused by cycles in the graph defined by the bridges
  - Think of bridges as edges, LANs as nodes in this graph
  - Redundant bridges form loops in this graph
- Idea: Turn this into a loop-free, acyclic graph
- Simplest approach: Compute a spanning tree on this LAN-bridge graph
  - Simple, self-configured, no manual intervention
  - But not optimal: actual capacity of installed bridges might not be fully exploited

Definition spanning tree: Given a graph G=(V,E), a ***spanning tree*** T=(V, $E_T$) is a subgraph of V, $E_T \subseteq E$, which is a tree (in particular, connected and acyclic)

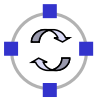


## Convergence: Switch and Bridge

- Traditionally, distinction between switch and bridge made sense
  - Bridges need more memory for storing addresses
  - Bridges need to implement spanning tree algorithm
- Today: most devices contain both types of functionality

- Often more a marketing distinction than a technical one

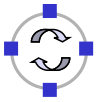




## Overview

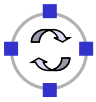
- LAN interconnection
- Physical-layer interconnects
- Data-link-layer interconnects
- ***Higher-layer interconnects***





## Routers

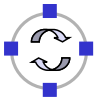
- All devices so far either ignored addresses (repeaters, hubs) or worked on MAC-layer addresses (switches, bridges)
- For interconnection *outside* a single LAN/connection of LAN, these simple addresses are insufficient
  - Main issue: “flat”, unstructured addresses do not scale
  - In spanning tree, there is an entry for *every* device’s designated output port!
- Need more sophisticated addressing structure and devices that operate on it
- Routers and routing!
  - Treated in the next chapter





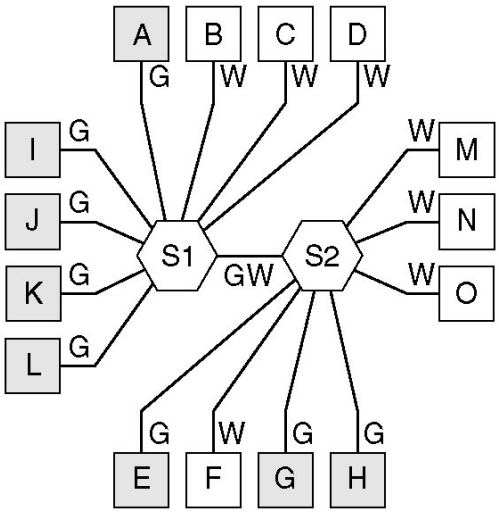
## Gateways

- If even routers will not do, higher-layer interconnection is necessary: ***Gateways***
  - Work at transport level and upwards
  - E.g., application gateways transforming between HTML and WML/HTTP and WAP
  - E.g., transcoding gateways for media content



# Further Topic in LAN/LAN Interconnection: VLAN

- Problem: LANs/switches are geared towards physical proximity of devices
- But: LANs should respect *logical* proximity
  - Connect devices of working groups together, irrespective where they happen to be located
- Idea: put a ***virtual LAN*** on top of an existing physical LAN
- Switches (or bridges) need configuration tables which port belongs to which VLAN
  - Only forward packets to ports of correct VLAN
- Membership of *incoming* packets determined by port, MAC address! VLAN mapping, or IP address ! VLAN mapping
  - Buzzword: IEEE 802.1Q






# Conclusions

- Single LANs are insufficient to provide communication for all but the simplest installations
- Interconnection of LANs necessary
  - Interconnect on purely physical layer: Repeater, hub
  - Interconnect on data link layer: Bridges, switches
  - Interconnect on network layer: Router
  - Interconnect on higher layer: Gateway
- Problems
  - E.g., redundant bridges can cause traffic floods; need spanning tree algorithm
  - Simple addresses do not scale; need routers