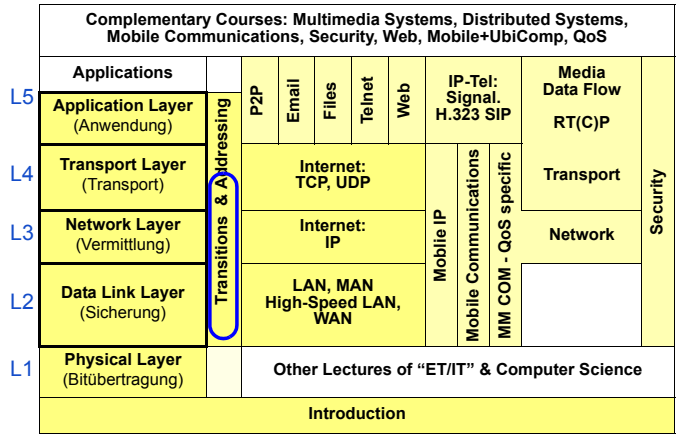
Communication Systems
Internetworking (Bridges & Co)

Prof. Dr.-Ing. **Lars Wolf**

TU Braunschweig
Institut für Betriebssysteme und Rechnerverbund

Mühlenpfordtstraße 23, 38106 Braunschweig, Germany
Email: wolf@ibr.cs.tu-bs.de

Scope



Overview

1. **Motivation**
2. **Repeater (Physical Layer)**
3. **Bridges (Data Link Layer)**

1. Motivation

Many heterogenous networks (in the past, nowadays and in future as well)

Heterogeneous network technologies (data link):

- WAN: telephone networks, ISDN, ATM, ...
  mobile communication: GSM, DECT, satellite networks, ...
- LAN: 802.3, 802.4, 802.5, 802.11, ...
- MAN: FDDI, DQDB, ...

Heterogeneous protocol architectures:

- SNA (> 20 000 networks)
- DECNET (> 2000)
- TCP/IP (world of Unix, also world of PCs)
- Novell NCP/IPX, Appletalk (world of PCs in former times)
- OSI, ...

Heterogeneous application architectures (with same overall purpose):

- Email
- Information access (WWW, WAP)

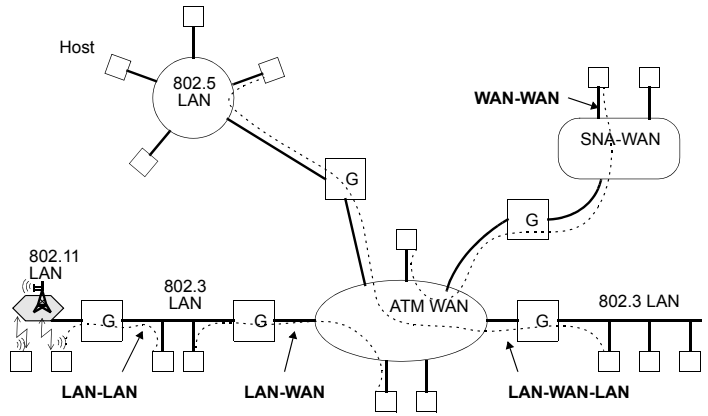
No changes in the near future, because

- **high investments, migration difficult**
- **decentralized investment decisions (departments install different networks)**
- **constantly new technologies**

Networks can Differ

Item	Some Possibilities
Service offered	Connection oriented vs. connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) vs hierarchical (IP)
Multicasting	Present or absent (same for broadcasting)
Packet size	Maximum different among nearly any two networks
Quality of service	Present or absent; many different flavors
Error handling	Reliable, ordered, unreliable, or unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

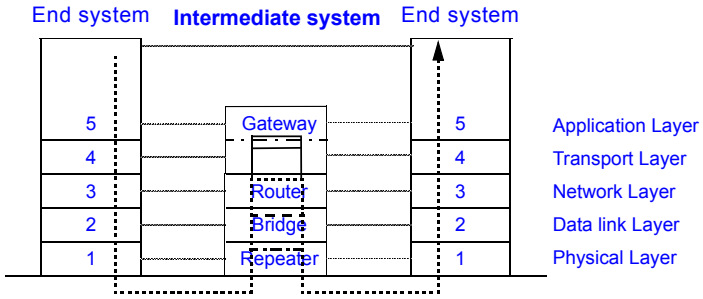
Interconnecting Different Networks



Why is it desirable to connect (heterogeneous) networks?

- **resource sharing (CPU, data bases, programs, mailboxes, ...)**
- **increased availability**
- **...**

Connecting Networks by "Relays"



Layer 1: Repeater / Hub

- copies bits between cable segments
- works solely as a repeater (does not modify the information)
- does not influence the traffic between networks
- example: connecting 802.3 cable segments (larger range)

Layer 2: Bridge / Switch

- relays frames between LANs (MAC level)
- minor frame modifications, increases the number of stations
- example: 802.x to 802.y

Connecting Networks by "Relays" (2)

Layer 3: Router (or Layer 3 Gateway)

- relays packets between different networks
- (modifies the packets)
- (converts different addressing concepts)
- (example: X.25 to SNA)

Layer 4 - 5: Gateway (or Protocol Converter)

- converts one protocol into another one (usually no 1-to-1 mapping of functions)
- example: TCP in ISO Transport Protocol,
  OSI Mail (MOTIS) in ARPA Internet Mail (RFC 822)
  change of media encoding (transcoding)

Note:

- **names (in products) are often intermixed**
- **e.g. bridge and switch**

Basic components

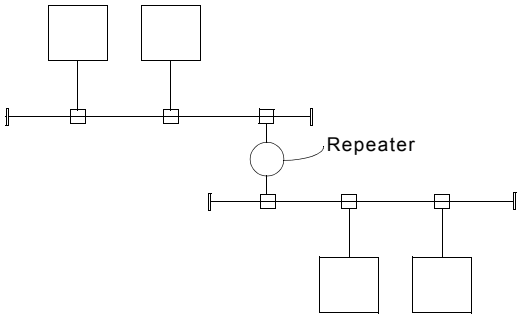
- **2 or more network connections**
- **connection entity**
- **control entity**

2 Paths

- **control path and data path**

2. Repeater (Physical Layer)

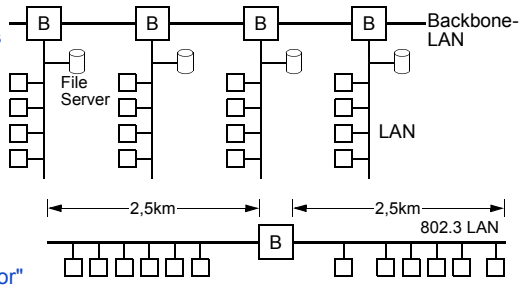
example: IEEE 802.3 configuration



Function
- **to amplify the electrical signals**
- **to increase the range**

3. Bridges (Data Link Layer)

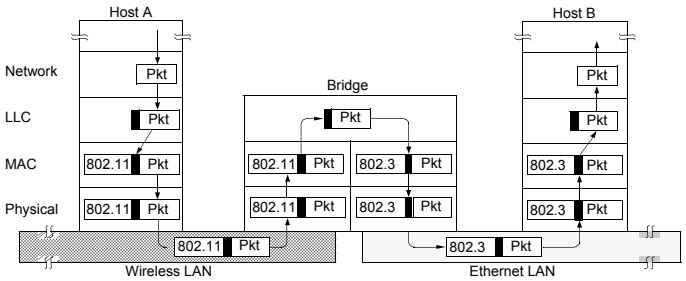
Tasks:
- **coupling of different LANs**
- **scalability of networks**
- **to increase capacity**
- **to cover larger distances**
- **to increase reliability**
  - bridge serves as "fire door"
- **to improve security**
  - stations can work in a promiscuous mode, i.e., read all frames on the network
  - bridge placement limits the spreading of information
- **to offer independence from protocols (IP, OSI, ...)**
  - in opposite to routers



Important goal: to achieve TRANSPARENCY
- **change attachment point without changes to HW, SW, configuration tables**
- **machines on any two segments should be able to communicate without regard to types of LANs used (directly or indirectly)**

3.1 Connecting 2 Different Networks: IEEE 802.x - Bridges

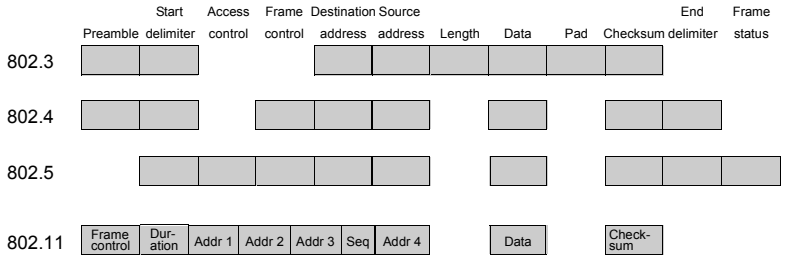
Example: 802.11 (Wireless LAN) and 802.3 (Ethernet)



Approach
- **LLC as common layer**
- **frames are routed to the respective MAC**
- **bridge contains**
  - its own implementation for each different MAC
  - for each physical layer the corresponding implementation

802.x ↔ 802.y: Tasks

Some different 802.x frame formats:
- **there are even more different frame formats ...**
- **some fields are technically necessary in one case but useless in another**
  - `duration` of 802.11



802.x ↔ 802.y: Tasks (2)

Different transmission rates (4/10/11/16/100/1000/... Mbps)
- bridge between fast LAN and slow LAN (or several LANs to one)
  - link can be overloaded
- buffering frames which cannot be transmitted immediately
- potentially many frames must be buffered within bridge
- (end-to-end) retransmission timer (at higher level) tries n*retransmissions
  - but then reports that end system is not available

Different frame lengths
- 802.3: 1518 bytes, 802.4: 8191 bytes, 802.5: unlimited, 802.11: 2346 bytes
- 802 does not support segmentation
  - not the task of this layer (at least typically seen this way)

⇒ **frames that are too long are dropped**
  - loss of transparency

802.x ↔ 802.y: Tasks (3)

Security
- 802.11 provides some data link layer encryption
- 802.3 does not

Quality of Service / Priorities
- supported (in various forms) by both 802.4 and 802.5
- NOT supported by 802.3
- 'kind of' in 802.11 (PCF / DCF)

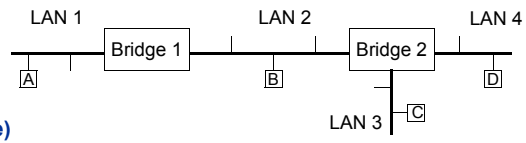
Acknowledgements
- supported by 802.4 (temporary token handoff)
- supported by 802.5 (C+A bits)
- not supported by 802.3

3.2 Connecting Several Networks: Transparent Bridges

Transparency:
- bridges not visible as such for the other components of the network

  ⇒ **simplifies other components**

Principle: transparent bridge
- bridge works in promiscuous mode (receives every frame of each connected LAN)
- bridge manages table: station → LAN (output line)



Bridge1: A → LAN 1 B → LAN 2 C → LAN 2 D → LAN 2

Decision procedure
1. source and destination LANs identical

⇒ frame dropped
2. source and destination LANs differ

⇒ frame rerouted to destination LAN
3. destination unknown

⇒ flooding

Transparent Bridges

Bridge table initially empty
- use flooding for unknown destination

Learning process: backward learning
- bridge works in promiscuous mode: receives any frame on any of its LANs
- bridge receives frames with source address Q on LAN L

  ⇒ Q can be reached over L

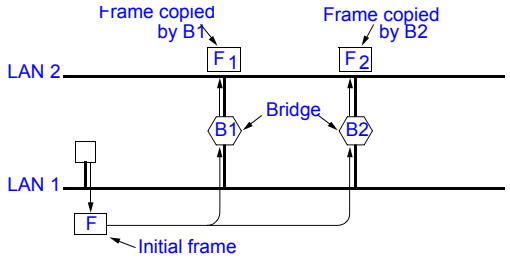
  ⇒ create table entry accordingly

Adaptation to changes in topology
- entry associated with timestamp (frame arrival time)
- timestamp of an entry (Z, LAN, TS) is updated when frame received from Z
- table scanned periodically and old entries purged (if no update for some time, usually several minutes)
  - e.g., because system moved and reinserted at different position
  - flooding is used if machine was quiet for some minutes

3.3 Transparent Bridges: Spanning Tree

Increase reliability:
- **connect LANs via various bridges in parallel**

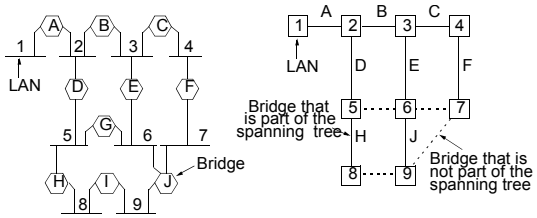
Problem
- **this creates a loop in the topology**
- **frames with unknown destination are flooded**
  - frame is copied again and again



Solution:
- **Communication among bridges**
- **Overlay actual topology by spanning tree reaching every LAN**
  - exactly one path from any LAN to every other LAN

Transparent Bridges: Spanning Tree (2)

Example



Bridges between LANs — A Spanning Tree

Algorithm
- **root of tree selection**
  - all bridges broadcast their serial number, lowest wins
- **generation of spanning tree (from the root to every bridge and LAN)**
  - configured with bridges representing the nodes within the tree
  - thereby avoiding loops
- **adaptation if configuration is changed (bridge or LAN)**

Drawback:
- **ignores some potential connections between LANs i.e., not all bridges are necessarily present in the tree**

3.4 Source Routing Bridges

Has been proposed (and used) as alternative to transparent bridges

Principle
- **the frame's sender defines path**
- **bridge routes the frame**

Prerequisite
- **LAN has a unique address (12 bit)**
- **bridge at the respective LAN is also unique (4 bit)**

then
- **sender flags the frame (top bit of its own address = 1), if destination address is not reachable in LAN**
- **bridge routes only frames that have been flagged in such a way**

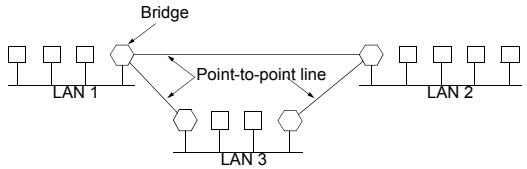
Determining Path
- **sender sends discovery frames as broadcast**
- **each bridge reroutes these (reaches every LAN)**
- **during return (route) the complete path is copied and transmitted to sender**
- **problem: high traffic**

Conclusion: usually transparent bridges are used

3.5 Connecting Two Equal Networks: Encapsulation

Remote bridge

Example:
- **Interconnect different sites of one organization**



Principle
1. **incoming data unit is packaged as payload,**
2. **transmitted and**
3. **then fed into the destination network**

Properties
- **certain protocol on point-to-point lines, e.g. PPP, i.e. MAC frame in PPP**
- **only station at the destination network can be reached (but for example not the network being bridged)**
- **simple**

3.6 Repeaters, Hubs, Bridges, Switches

Repeaters & Hubs (L1):

- **one collision domain**

Bridges (L2):

- **connects two or more LANs (potentially different types)**
- **each line is its own collision domain**
- **typically store-and-forward and (traditionally) CPU-based**

Switches (L2)

- **typically connects two or more computers**
- **each port / line is its own collision domain (no collisions)**
- **typically cut-through switching devices**
  - begin forwarding as soon as possible
  - when destination header has been detected, before rest of frame arrived
- **hardware-based**

Bridges vs. Switches

- **sometimes difference seems to be more a marketing issue than technical one**