Networks

- Network-based design.
  - Communication analysis.
  - System performance analysis.
- Internet.
- Internet-enabled systems.
- Vehicles as networks.
- Sensor networks

Communication analysis

- First, understand delay for single message.
- Delay for multiple messages depends on:
  - network protocol;
  - devices on network.

Message delay

- Assume:
  - single message;
  - no contention.
- Delay:
  - $t_m = t_x + t_n + t_r$
  - $=$ xmtr overhead + network xmit time + rcvr overhead

Example: I²C message delay

- Network transmission time dominates.
- Assume 100 kbits/sec, one 8-bit byte.
- Number of bits in packet:
  - n_{packet} = start + address + data + stop
  - = 1 + 8 + 8 + 1 = 18 bits
- Time required to transmit: 1.8 x 10^{-4} sec.
- 20 instructions on 8 MHz controller adds 2.5 x 10^{-6} delay on xmtr, rcvr.

Multiple messages

- If messages can interfere with each other, analysis is more complex.
- Model total message delay:
  - $t_y = t_d + t_m$
  - $= $ wait time for network + message delay

Arbitration and delay

- Fixed-priority arbitration introduces unbounded delay for all but highest-priority device.
  - Unless higher-priority devices are known to have limited rates that allow lower devices to transmit.
- Round-robin arbitration introduces bounded delay proportional to N.

Further complications

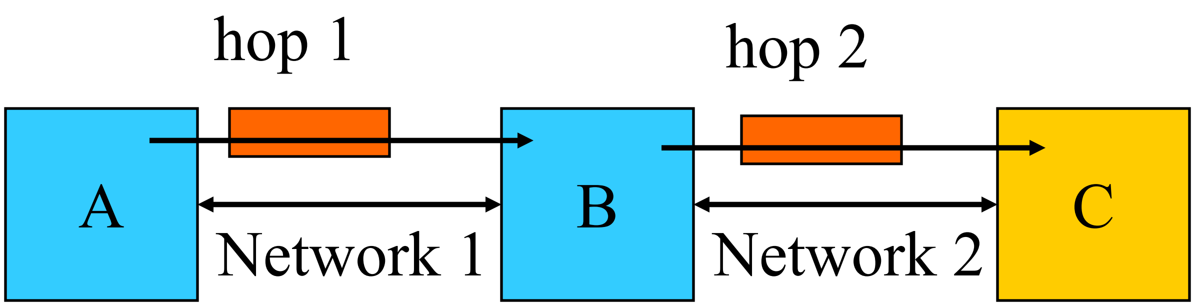
- Acknowledgment time.
- Transmission errors.

Priority inversion in networks

- In many networks, a packet cannot be interrupted.
- Result is priority inversion:
  - low-priority message holds up higher-priority message.
- Doesn't cause deadlock, but can slow down important communications.

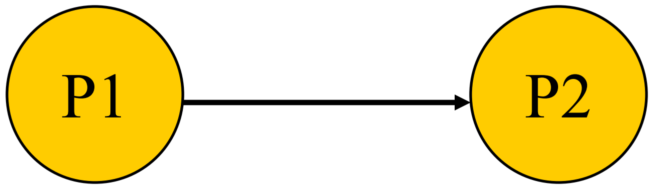
Multihop networks

- In multihop networks, one node receives message, then retransmits to destination (or intermediate).



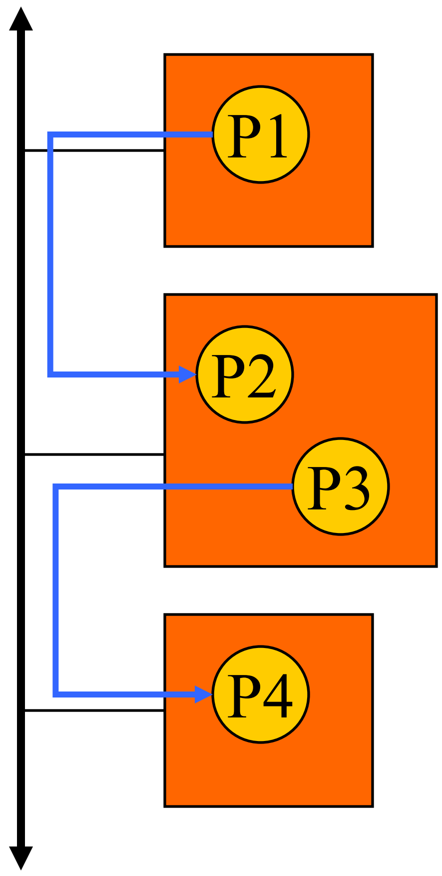
System performance analysis

- System analysis is difficult in general.
  - multiprocessor performance analysis is hard;
  - communication performance analysis is hard.
- Simple example: uncertainty in P1 finish time -> uncertainty in P2 start time.



Analysis challenges

- P2 and P3 can delay each other, even though they are in separate tasks.
- Delays in P1 propagate to P2, then P3, then to P4.



Lower bounds on system

- Computational requirements:
  - sum up process requirements over least-common multiple of periods, average over one period.
- Communication requirements:
  - Count all transmissions in one period.

Hardware platform design

- Need to choose:
  - number and types of PEs;
  - number and types of networks.
- Evaluate a platform by allocating processes, scheduling processes and communication.

I/O-intensive systems

- Start with I/O devices, then consider computation:
  - inventory required devices;
  - identify critical deadlines;
  - chooses devices that can share PEs;
  - analyze communication times;
  - choose PEs to go with devices.

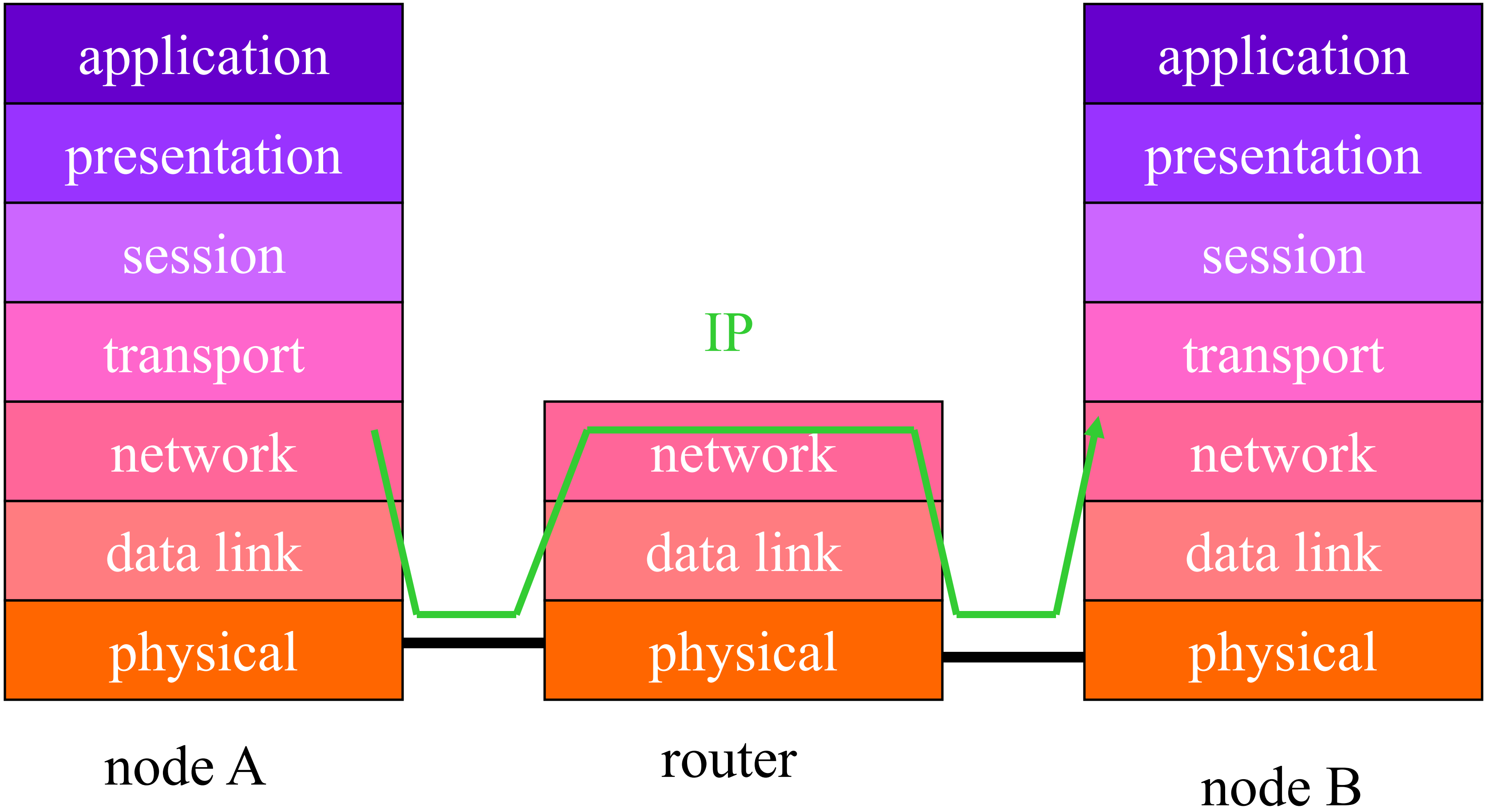
Computation-intensive systems

- Start with shortest-deadline tasks:
  - Put shortest-deadline tasks on separate PEs.
  - Check for interference on critical communications.
  - Allocate low-priority tasks to common PEs wherever possible.
- Balance loads wherever possible.

Internet Protocol

- Internet Protocol (IP) is basis for Internet.
- Provides an internetworking standard: between two Ethernets, Ethernet and token ring, etc.
- Higher-level services are built on top of IP.

IP in communication



IP packet

- Includes:
  - version, service type, length
  - time to live, protocol
  - source and destination address
  - data payload
- Maximum data payload is 65,535 bytes.

IP addresses

- 32 bits in early IP, 128 bits in IPv6.
- Typically written in form `xxx.xx.xx.xx`.
- Names (foo.baz.com) translated to IP address by domain name server (DNS).

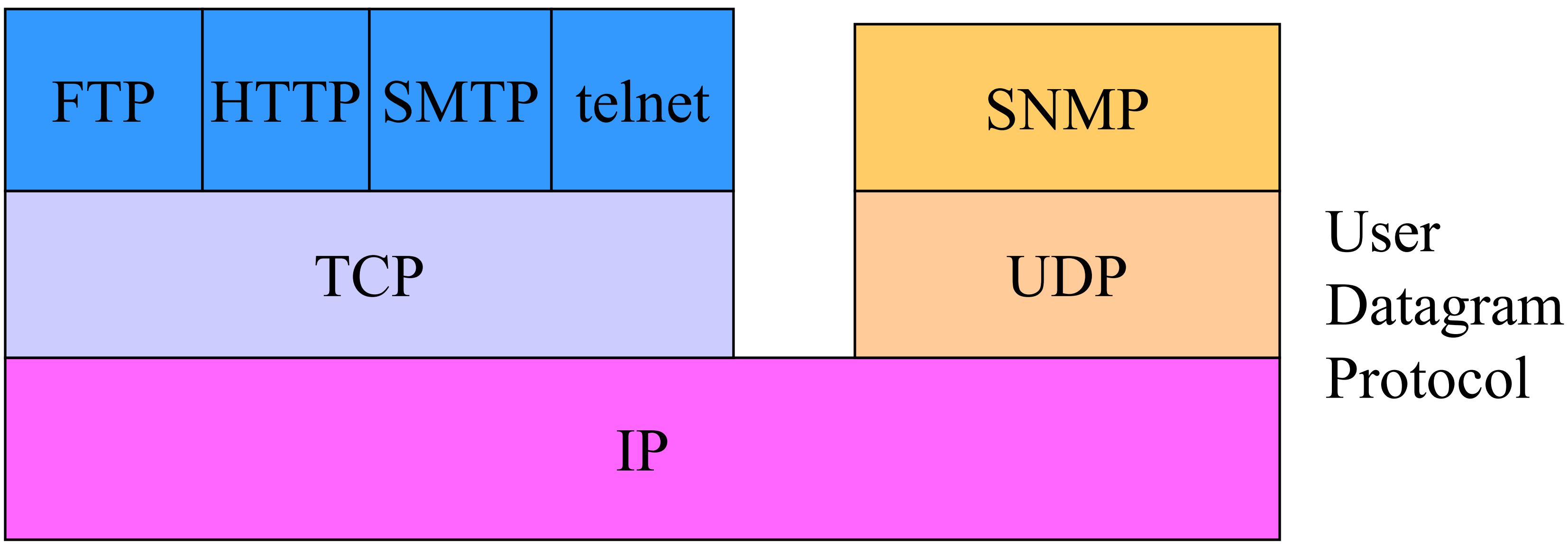
Internet routing

- Best effort routing:
  - doesn’t guarantee data delivery at IP layer.
- Routing can vary:
  - session to session;
  - packet to packet.

Higher-level Internet services

- Transmission Control Protocol (TCP) provides connection-oriented service.
- Quality-of-service (QoS) guaranteed services are under development.

The Internet service stack



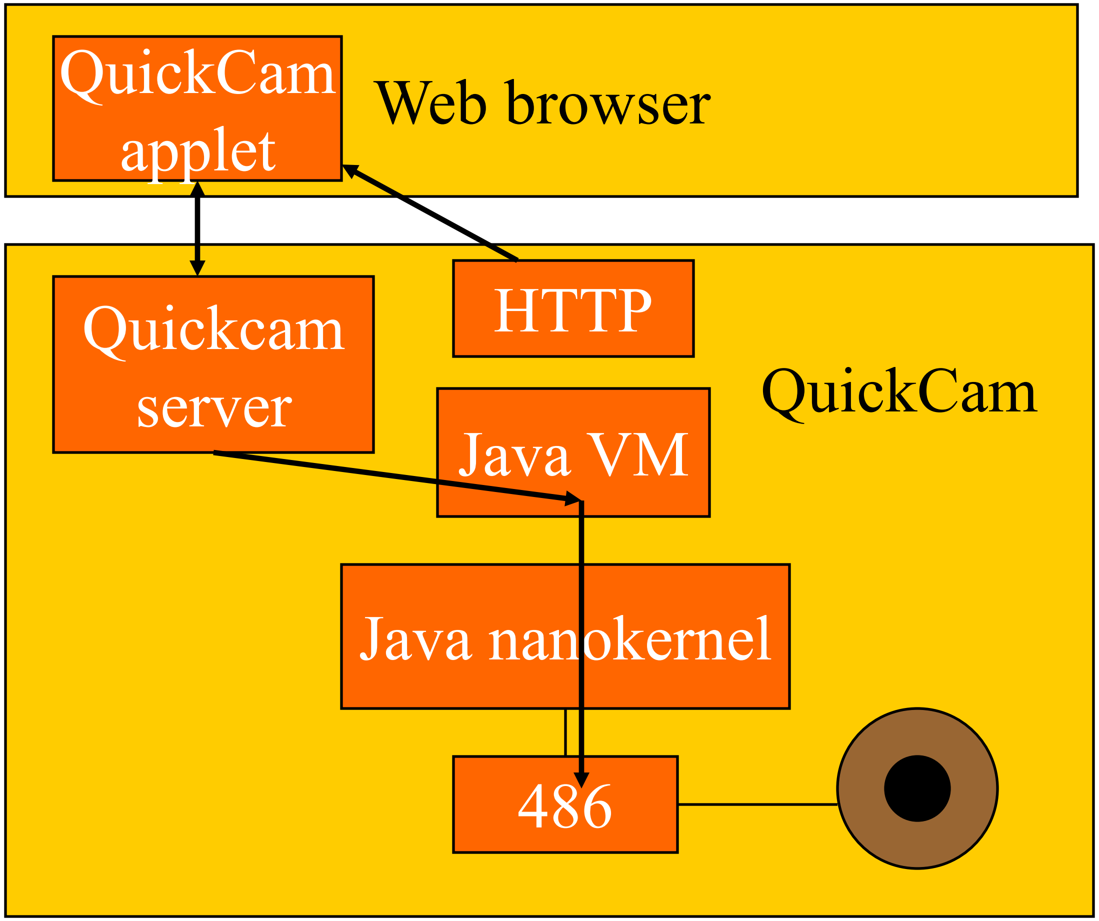
Internet-enabled embedded system

- Internet-enabled embedded system: any embedded system that includes an Internet interface (e.g., refrigerator).
- Internet appliance: embedded system designed for a particular Internet task (e.g. email).
- Examples:
  - Cell phone.
  - Laser printer.

Example: Javacam

- Hardware platform:
  - parallel-port camera;
  - National Semi NS486SXF;
  - 1.5 Mbytes memory.
- Uses memory-efficient Java Nanokernel.

Javacam architecture

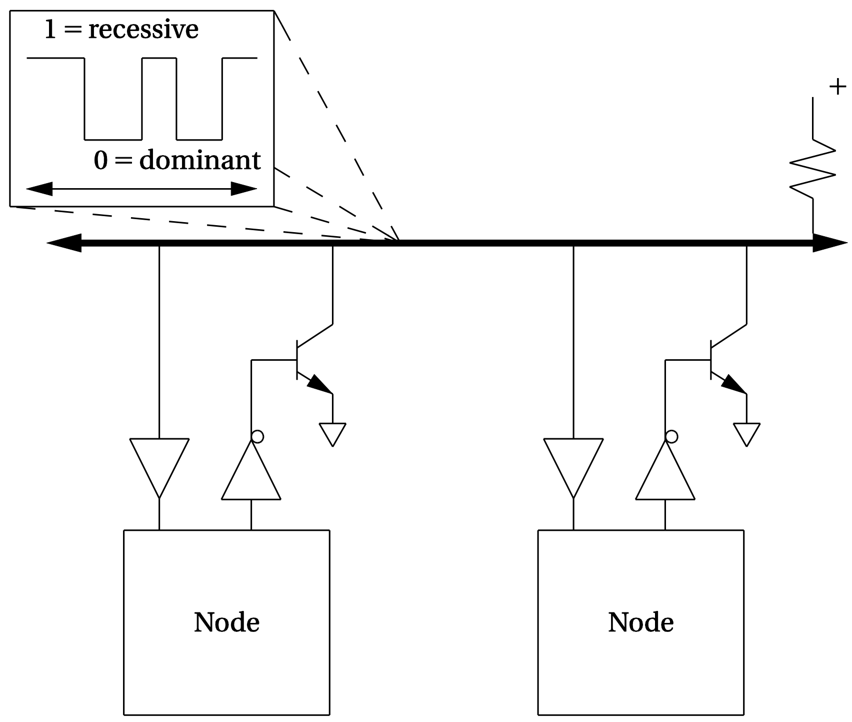


Vehicles as networks

- 1/3 of cost of car/airplane is electronics/avionics.
- Dozens of microprocessors are used throughout the vehicle.
- Network applications:
  - Vehicle control.
  - Instrumentation.
  - Communication.
  - Passenger entertainment systems.

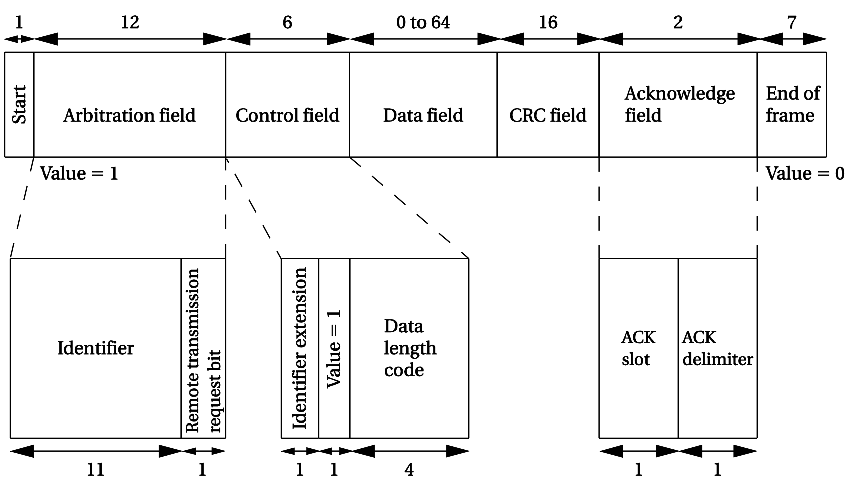
CAN bus

- First used in 1991.
- Serial bus, 1 Mb/sec up to 40 m.
- Synchronous bus.
- Logic 0 dominates logic 1 on bus.
- Arbitrated with CSMA/AMP:
  - Arbitration on message priority.



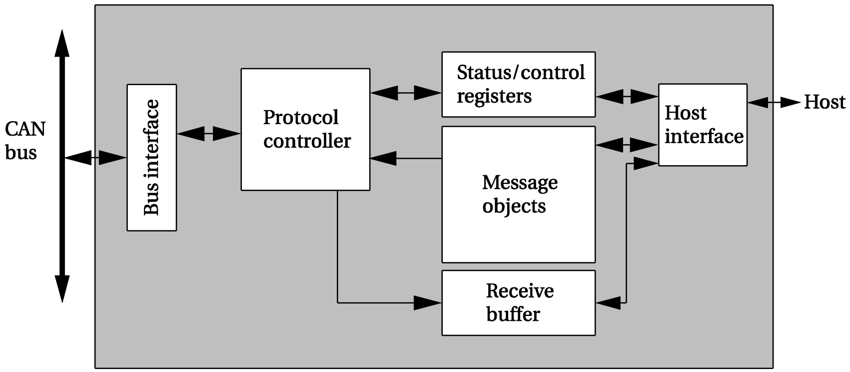
CAN data frame

- 11 bit destination address.
- RTR bit determines read/write from/to destination.
- Any node can detect bus error, interrupt packet for retransmission.



CAN controller

- Controller implements physical and data link layers.
- No network layer needed---bus provides end-to-end connections.



Other vehicle busses

- FlexRay is next generation:
  - Time triggered protocol.
  - 10 Mb/s.
- Local Interconnect Network (LIN) connects devices in a small area (e.g., door).
- Passenger entertainment networks:
  - Bluetooth.
  - Media Oriented Systems Transport (MOST).

Avionics

- Anything permanently attached to the aircraft must be certified by FAA/national agency.
- Traditional architecture uses separate electronics for each instrument/device.
  - Line replaceable unit (LRU) can be physically removed and replaced.
- Federated architecture shares processors across a subsystem (nav/comm, etc.)

Sensor networks

- Wireless networks, small nodes.
- Ad hoc networks---organizes itself without system administrator:
  - Must be able to declare membership in network, find other networks.
  - Must be able to determine routes for data.
  - Must update configuration as nodes enter/leave.

Node capabilities

- Must be able to turn radio on/off quickly with low power overhead.
  - Communication/computation power = 100x.
- Radios should operate at several different power levels to avoid interference with other nodes.
- Must buffer, route network traffic.