Introduction to Real-Time Communications

Real-Time and Embedded Systems (M)

Lecture 15

UNIVERSITY
of
GLASGOW

Lecture Outline

- Modelling real-time communications
  - Traffic and network models
- Properties of networks
  - Throughput, delay and jitter
  - Clock skew
  - Congestion and loss
- Examples
  - Controller area networks
  - Ethernet

Material corresponds to chapter 11 of Liu's book

Real Time Communications

- In most data communications, important that data arrives reliably
  - Would like it to be fast, but prefer reliable
    - E.g. web, email, p2p, etc.
  - Often characterised as *elastic* applications

- In real time communication it is important that the message arrives in a timely manner
  - Timeliness may be *more* important than reliability
  - Messages may have priority
  - Examples:
    - A "drive by wire" system in a car
    - Packet voice and telephony applications

Modelling Real Time Traffic

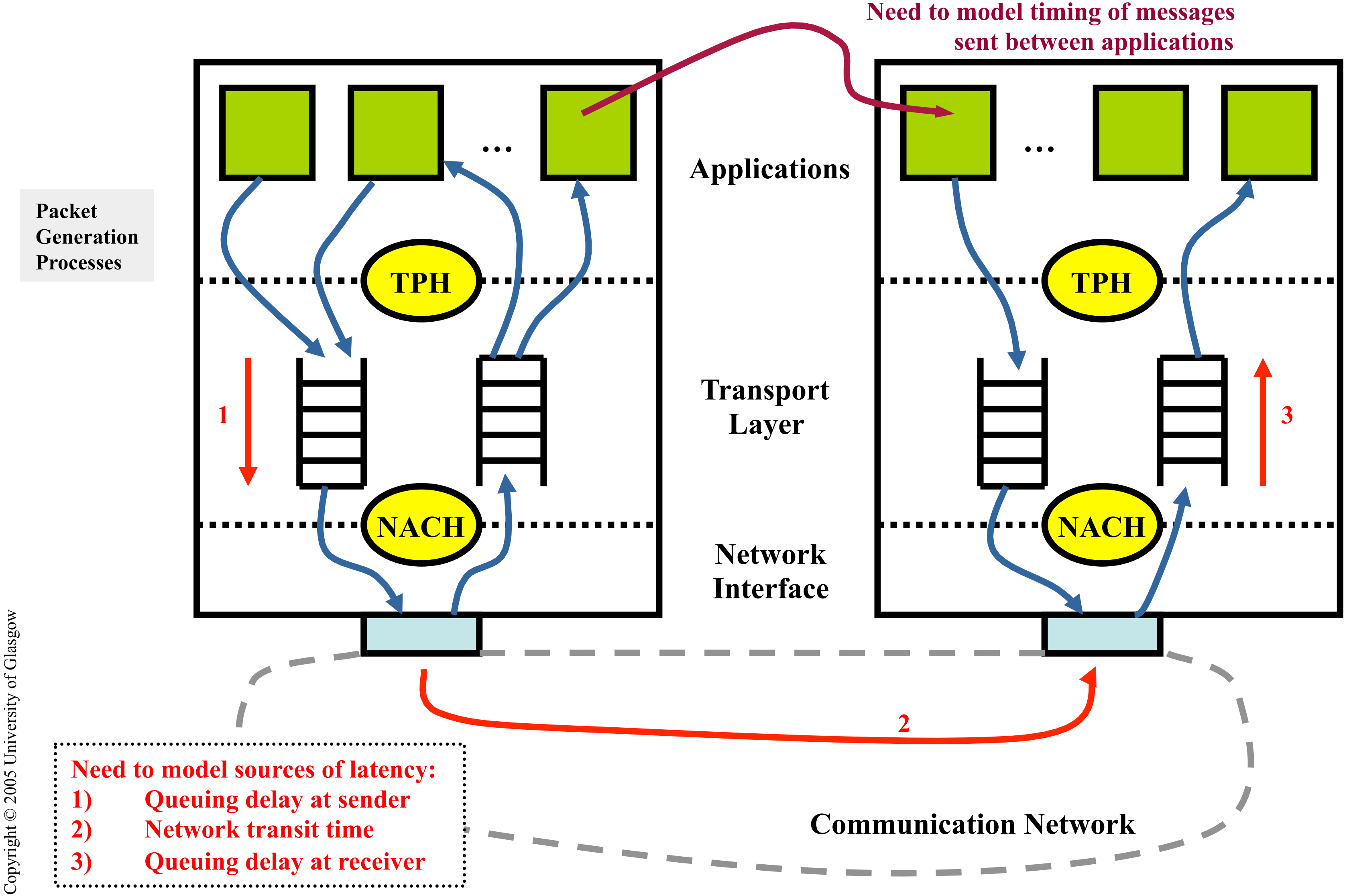
- Assume a packet-based network
  - Real-time traffic on circuit switched network trivial after connection setup
- Traffic falls into two categories:
  - Synchronous periodic flows
    - Produced and consumed in a continuous basis, according to some schedule
    - Can be generated by periodic tasks
      - Fixed rate flows (e.g. sensor data, speech)
      - Characterise by inter-packet spacing, message length, reception deadline
    - Can be generated by sporadic tasks
      - Variable rate flows (e.g. MPEG-2 video, control traffic)
      - Characterise by average throughput + maximum burst size
    - Generally require some performance guarantee
  - Aperiodic (asynchronous) messages
    - No deadline, best effort delivery, but want to keep delays small
    - Characterise by average delivery time

Modelling Sources of Timing Variation

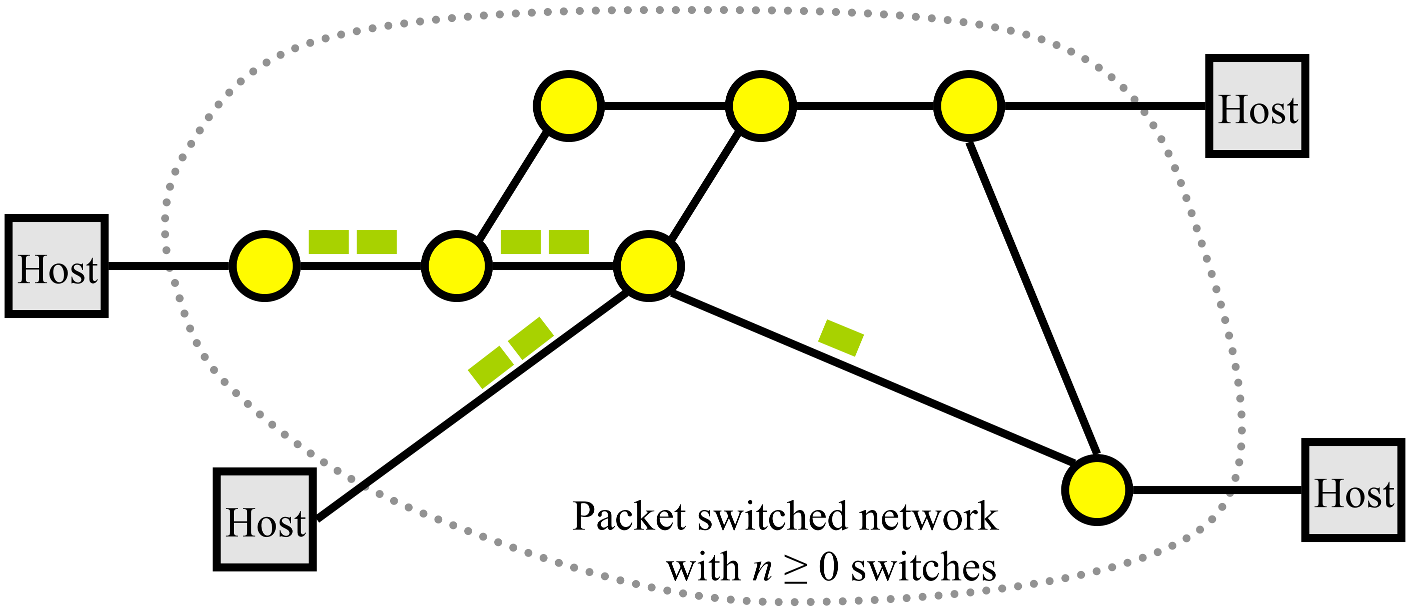
- Ideally the network delivers messages to receiver with no delay, preserving timing
- In reality there is:
  - Queuing delay at sender
    - Network not always ready to accept a packet when it becomes available; data may be queued if produced faster than the network can deliver it
  - Queuing delay at receiver
    - Application not always ready to accept packets arriving from network
    - Network may deliver data in bursts
  - Queuing delay in the network
    - Due to cross-traffic or bottleneck links
  - Network transit time
    - Fixed propagation delay

General Model of Hosts and Network

TPH = Transport Protocol Handler
NACH = Network Access Control Handler



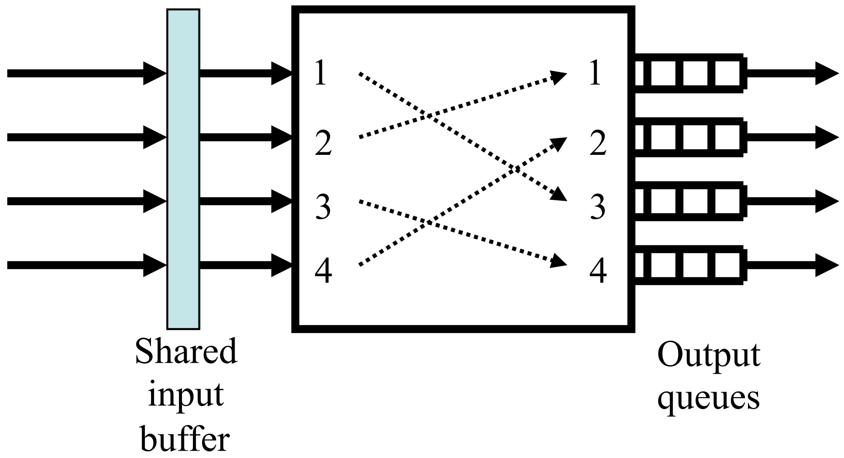
Model of Packet Switched Networks



Links have constant *propagation delay*

Switches are *output buffered* with packets queued for transmission if output link busy

Choice of *job scheduling algorithm* on the output link is critical for real time traffic



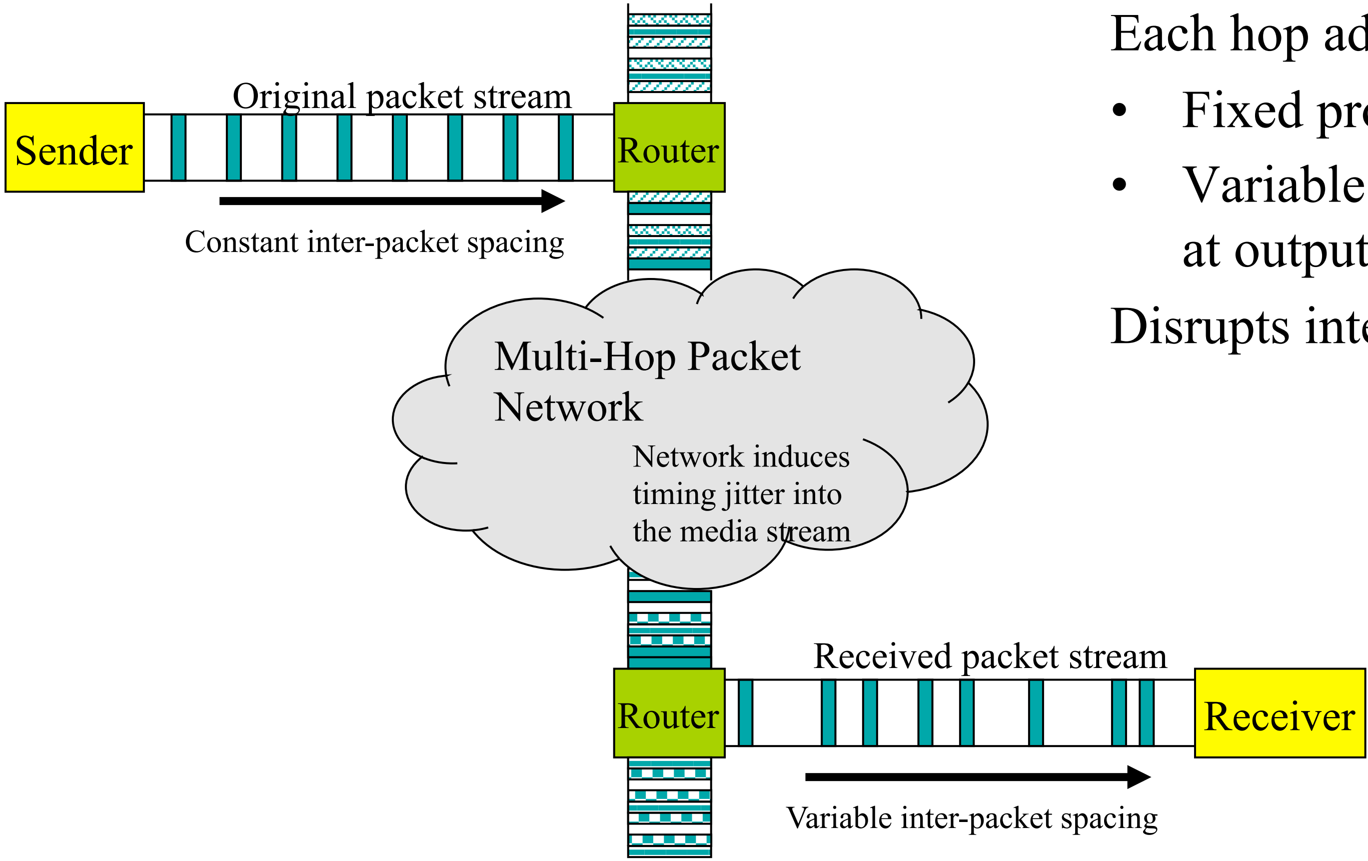
Performance Objectives and Constraints

- Have seen the types of traffic produced, and the sources of timing variation in the network
- What are the effects of these? Interactions?
  - Throughput and delay
  - Jitter and buffer requirements
  - Miss rates, when jitter causes a deadline to be missed
  - Packet loss and invalid rates
- If we can characterise, we can schedule communications

Throughput, Delay and Jitter

- The *throughput* (or rate) is a measure of the number of packets that the network can deliver per unit time
  - Throughput may be average or instantaneous
- The *delay* (or latency) is time taken to deliver a packet
  - There will be a fixed minimum delay, due to propagation time of the signal across the network medium
    - Ultimately limited by the speed of light
    - Significant for long distance communications
  - There will be variation due to elements along the network path
    - Queuing delay at sender
    - Queuing delay at receiver
    - Affect end-to-end delay as seen by the application
- The delay *jitter* is the variation in the delay

Throughput, Delay and Jitter



Each hop adds:

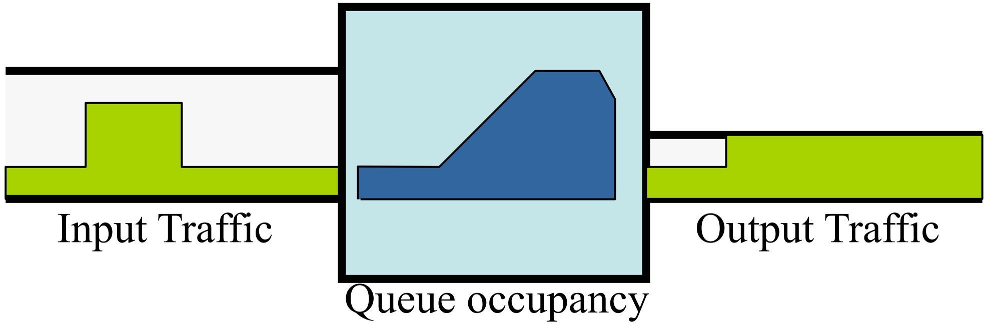
- Fixed propagation delay
- Variable queuing delay at output links

Disrupts inter-packet timing

- Consider the *throughput*, *delay* and *jitter* of the network with different scheduling algorithms in the routers
- Want to minimize the jitter and latency; retain throughput

Throughput and Delay

- Clear that throughput and delay depend on the capacity of each link, and on the queuing delay at each hop
- Queuing delay will vary based on the traffic
  - Variation in the throughput may cause queues to build up at bottleneck links



  - Cross traffic will also affect queue occupancy
- Throughput may be limited by an intermediate link, which cannot be directly observed by sender and receiver
  - How to tell if the throughput is limited by the network, or by other traffic using the network?
  - Cannot know if capacity available, unless requirements signalled in advance

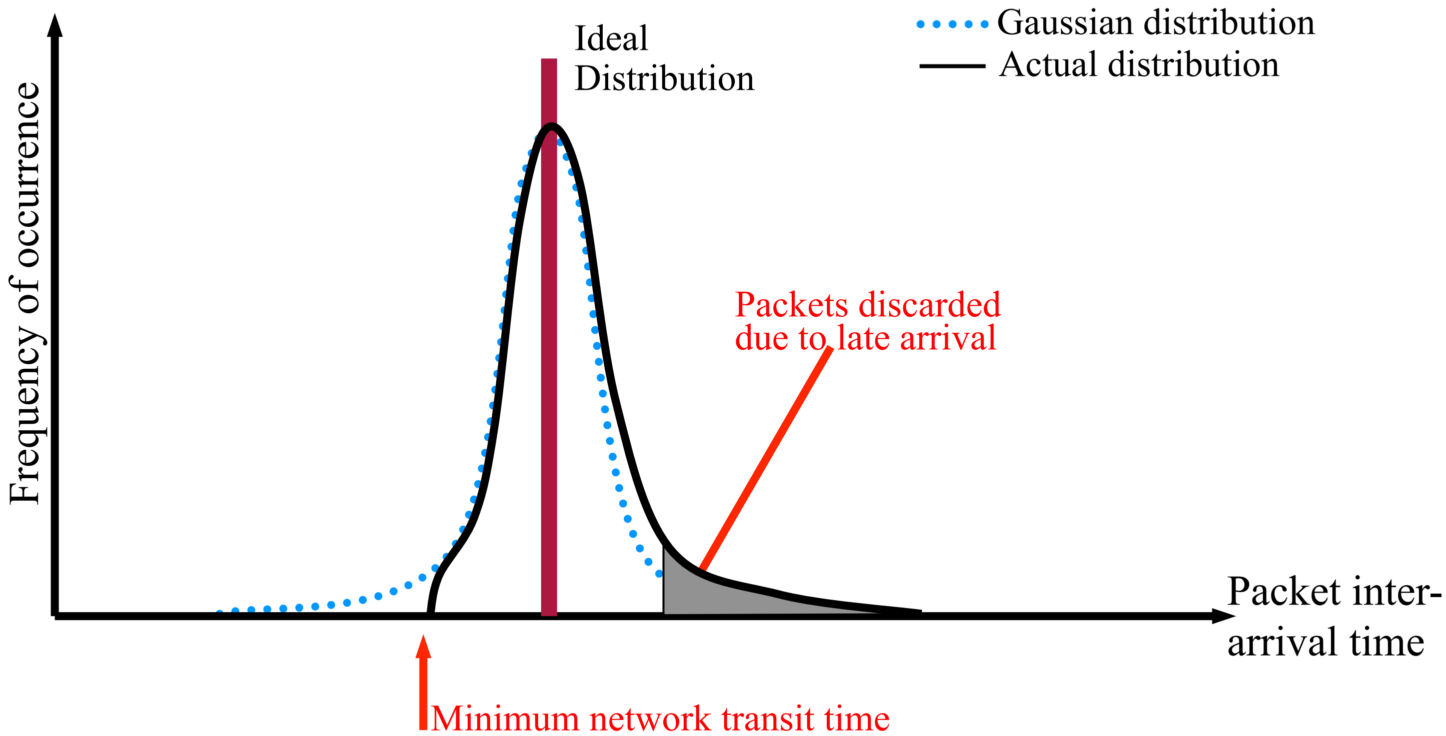
Throughput and Delay

- Delay matters for some applications, but not others
  - Interactive applications need low delay
    - Telephony, video conferencing and games
    - Control applications often need low delay in the sensor ⇒ controller ⇒ actuator loop
  - Non-interactive applications are less delay sensitive
    - Video on demand, TV and radio distribution
- Throughput typically very important
  - Need to sustain a certain rate, to support the application
  - May wish to use scheduling algorithms to prioritise which packets are to be sent, and guarantee throughput

Jitter and Buffering Requirements

- Delay *jitter* is the variation in delay across a network path
  - For isochronous traffic, often talk about absolute value and standard deviation of packet inter-arrival time
  - Assumes we can characterise the jitter – see examples later
- Jitter will affect the buffer requirement
  - We gain no advantage in completing a job early
  - Indeed, a periodic or sporadic message that is delivered early may be problematic, since it must be buffered until the destination is ready to process it
  - Larger jitter implies more buffering is needed
    - Used to be a big problem for TV set-top-box manufacturers, since memory was expensive
  - Many systems designed to keep jitter as small as possible

Jitter and Miss Rate



- System may have limited buffering, due to lack of memory or application timing requirements
- Packets may arrive late due to jitter
- Fraction of packets lost is the *miss rate*
  - For soft real-time systems only!

Clock Skew and Synchronisation

- In a packet switched communication system, it is common that sender are receiver are widely distributed
- As a result, the sender clock may not be synchronised with the receiver clock and clock skew may occur
  - Results in a steady increase or decrease in the inter-packet spacing observed at the receiver

- This is problematic for isochronous applications:
  - Queues can build up in the receiver or in intermediate systems
    - Eventually buffer space will be exceeded
    - Some data will be dropped
  - Queues can empty in the receiver
    - Initial queue created, to buffer for jitter
    - Sender is slightly slower than receiver
    - Queue slowly empties, eventually there is no data to process

Congestion and Loss

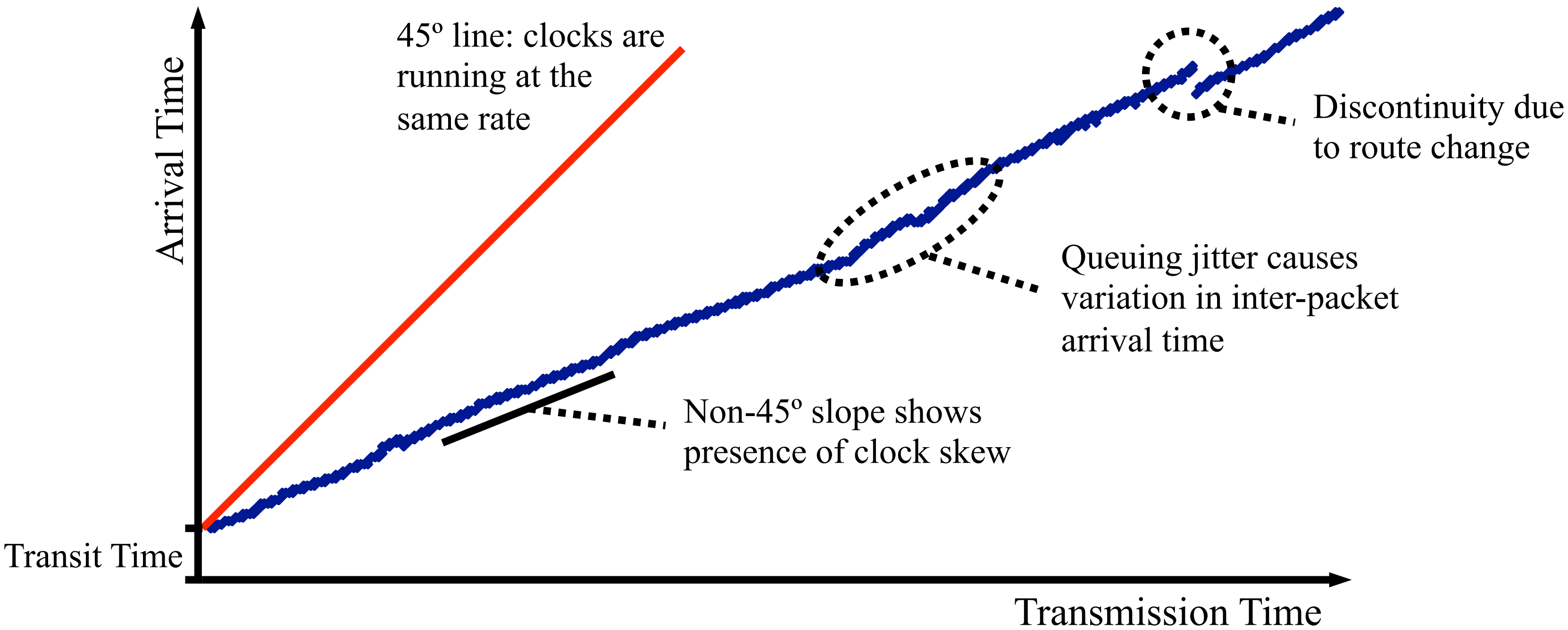
- Assumed that no traffic is ever blocked or lost because there is no space in the ready queue when it becomes available for execution
- Usually valid for operating systems and LAN communication
- Not valid for many wide-area communication systems
  - Too expensive to provision buffering in all routers
  - Provision for typical load plus a safety factor, not worst case
- Queues may overflow, hence packets are dropped
  - The *loss rate* gives the fraction of packets that are dropped
  - Patterns of loss may also be important
    - Packet scheduling algorithms affect loss pattern
- Packets may also be dropped due to corruption or other errors
  - Not discussed further, since not affected by scheduling

Congestion and Loss

- Implication: not only may we cause overloads and congestion, so might the cross traffic
  - Temporary congestion will cause queuing delays
  - Persistent congestion will result in queues that stay full, hence packets may be lost
- How to avoid this?
  - Control the amount of traffic at a bottleneck link
    - Applications need to signal their requirements
    - Network needs to perform admission control
  - Or prioritise traffic, to give preference to important flows
    - What scheduling algorithm to use?
    - May allow real-time traffic, but discard best effort data traffic when the network is overloaded

Packet Timing Graph

- A network can significantly disrupt timing of a flow
- Visualise these effects using a packet timing graph



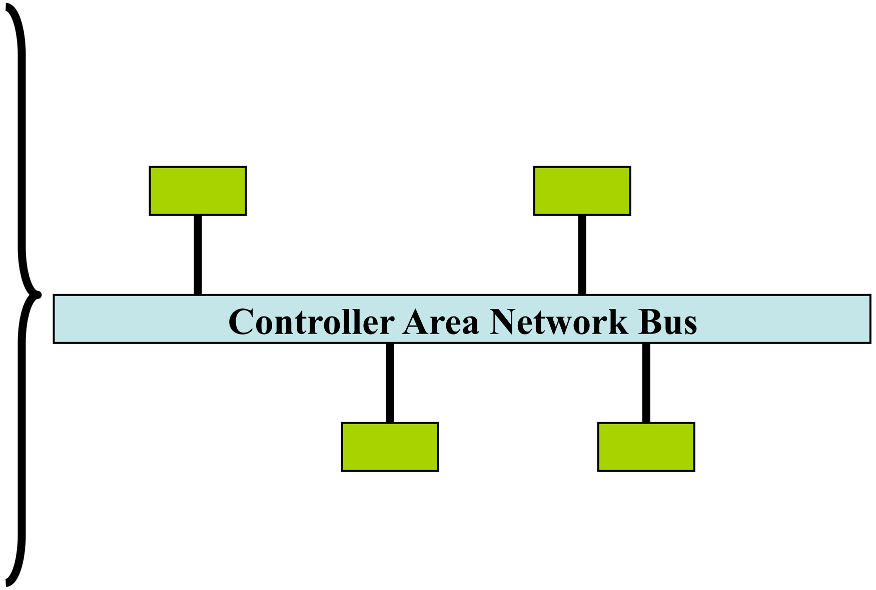
Characterisation of Networks

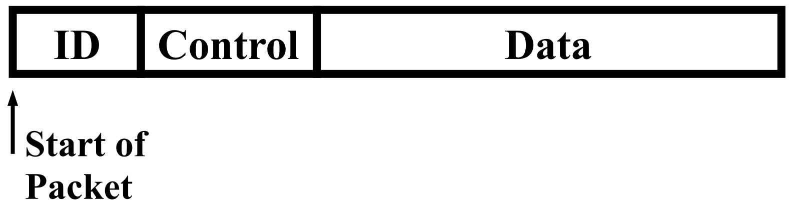
- Real-world performance constraints force us to characterise the timing behaviour of a network
  - Prove/demonstrate that throughput, latency, and jitter are within appropriate bounds for the application

- Some network technologies allow this, others do not
  - Examples: CAN, Ethernet

Example: Controller Area Networks

- Shared serial bus, send at 1Mbps, maximum bus length is 50 metres
- All stations hear transmissions within a fraction of a bit time
- Connections wired together as a logical AND function
  - Stations only see a 1 bit on the bus if all transmitters are sending a 1 bit
- Packets start with an ID, then control and data
- Slotted CSMA/CD: wait until start of slot, then begin to send with the ID field, but:
  - Stop if you hear a 0 on the bus when you are sending a 1
  - Packet with smallest ID is sent first; priority network protocol





Example: Controller Area Networks

- Widely used in automotive systems, for example
- Allows communications to be scheduled using the fixed priority scheduling algorithms we have discussed
  - Look at the communications patterns, assign deadlines to each message exchange
  - Use deadline monotonic scheduling to assign priorities
    - 11 bit ID field, implies 2048 priority levels
    - Treat sporadic messages as periodic messages, according to worse case assumptions
      - Waste capacity, but ensures schedulability
  - The CAN will not pre-empt a message once it has started
  - Low utilisation, but can prove that all messages will be delivered before their deadlines and calculate jitter
    - Standard schedulability analysis, as for any set of jobs

Example: Ethernet

- Recall that Ethernet uses CSMA/CD with exponential back off
  - Try to transmit, listening for collision
  - If a collision occurs, stop sending, wait before retry
  - Random binary exponential back-off
    - After *i* collisions back-off by up to 2*i* slots, randomly chosen
- Potentially unbounded delay on busy network
  - Cannot schedule transmissions to avoid collision
- No prioritisation of messages

- Implications:
  - Cannot easily reason about timing properties
  - Difficult to schedule messages to ensure timely delivery

Summary

- What is real time communication
- Factors that affect real time communication
  - Throughput, delay and jitter
  - Clock skew
  - Congestion and loss
- Examples of networks and their timing properties
  - Some networks provide timing guarantees, others do not

Tomorrow: real-time communication on IP networks