Introduction to Real-Time Communications

Colin Perkins

http://csperkins.org/teaching/2003-2004/rtes4/lecture14.pdf

Reading for this week: Chapter 11

UNIVERSITY of GLASGOW

Lecture Outline

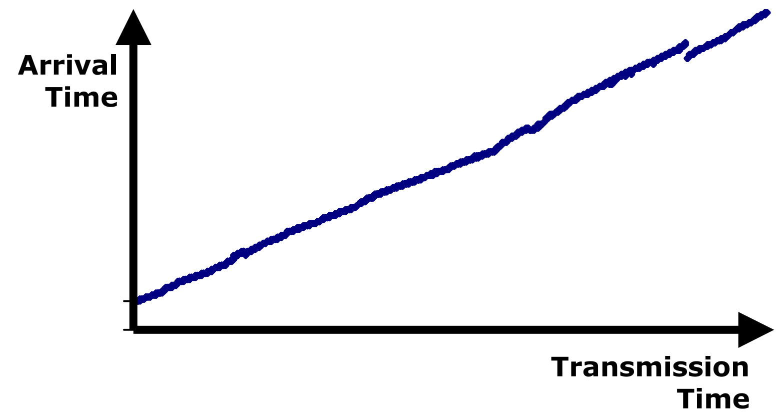
- Administrivia and programming assignment
- Modelling real-time communications
- Timing properties of networks
- Predictability and the need for resource reservation
- Examples
  - Controller area networks
  - IP and the Internet

Administrivia

- The strike: lectures will **go ahead as normal** this week
  - Wednesday: Quality of Service for Packet Networks
  - Thursday: Real-Time Communication on IP Networks

The Programming Assignment

- Due at 5pm on Friday 12th March
- Hints:
  - Consider using **`nanosleep()`** and **`select()`** with a timeout
  - Consider using **`send()`** and **`recv()`** with **`SOCK_DRAM`** sockets
  - Concentrate on the network, threading and timing code
  - Store data in simple files, use something like gnuplot or Excel to plot timing graphs
  - Think about, and discuss, the timing graphs you plot

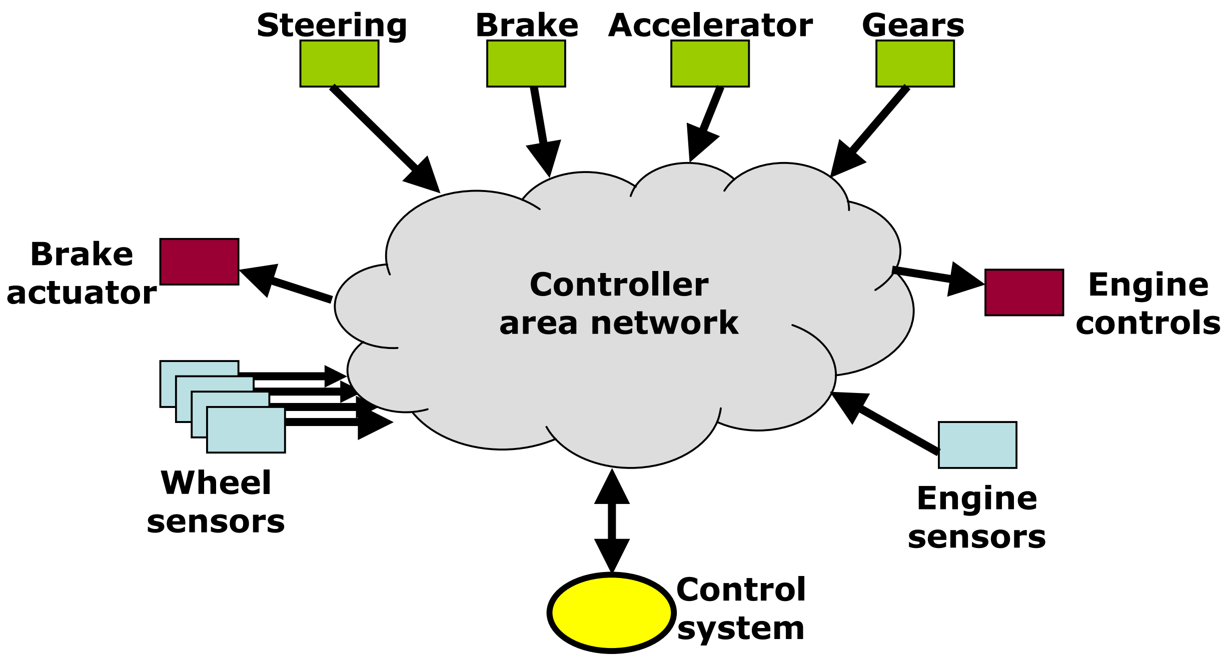


- Don’t:
  - Waste time giving your programs a nice user interface
  - Waste time writing your own graph plotting routines

Real Time Communications

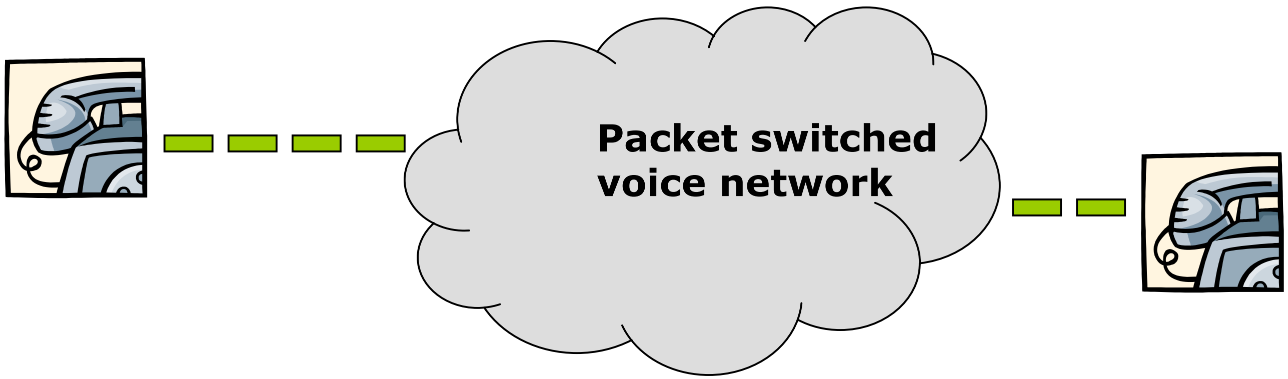
- In most data communications, it is important that the data arrives reliably
  - Would like it to be fast, but prefer reliable
  - E.g. web, email, etc.
  - Often characterised as **elastic** applications
- In real time communication it is important that the message arrives in a timely manner
  - Timeliness may be more important than reliability
  - Messages may have priority
- Examples:
  - A “drive by wire” system in a car
  - Packet voice and telephony applications

Example: Drive by Wire



- All data must be delivered reliably
  - Is bad if you turn the steering wheel, and nothing happens
- Commands from control system have highest priority, then sensors and actuators, then control inputs
  - Anti-lock brakes have a faster response time than the driver, so prioritise to ensure the car doesn't skid
- Network must schedule and prioritise communications

Example: Packet Voice



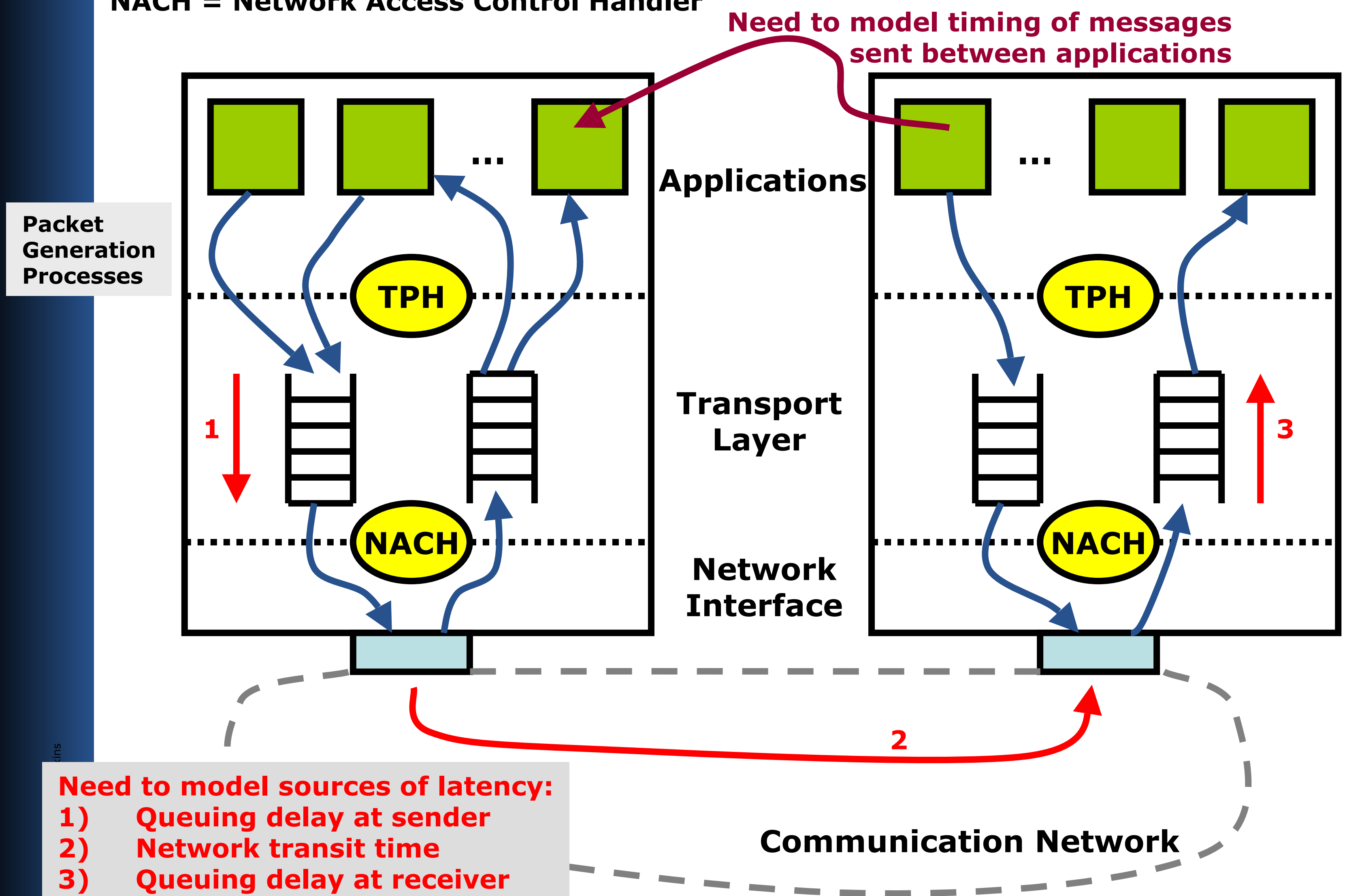
- Voice is digitised and sent as a sequence of packets
  - Constant spacing, every 10-30ms depending on codec
- Strict timeliness requirement
  - Mouth to ear delay needs to be less than approximately 150ms
  - Packets must be played out with equal spacing
- Relaxed reliability requirement
  - Some small fraction of packets can be lost, and just sound like crackles on the wire; most need to arrive
- Emergency calls may have priority

Modelling Application Traffic

- Assume that messages are split into “packets” for transport through the communications network
- Isochronous (synchronous) flows
  - Produced and consumed in a continuous basis
    - Messages are generated and consumed by tasks according to some schedule
  - Can be generated by periodic tasks
    - Fixed rate flows (e.g. sensor data, speech)
    - Characterise by tuple (P_i, e_i, D_i): period (inter-packet spacing), message length, reception deadline
  - Can be generated by sporadic tasks
    - Variable rate flows (e.g. MPEG-2 video, control traffic)
    - How to characterize? Leaky bucket, etc
  - Generally require some performance guarantee
- Aperiodic (asynchronous) messages
  - No deadline, best effort delivery, but want to keep delays small
  - Characterise by average delivery time

General Model of Hosts and Network

TPH = Transport Protocol Handler
NACH = Network Access Control Handler



Modelling Sources of Timing Variation

- Ideally the network will deliver messages to the receiver with no delay, preserving the timing
- In reality there is:
  - Queuing delay at sender
    - Network is not always ready to accept a packet when it becomes available; data may be queued if it's produced faster than the network can deliver it
  - Queuing delay at receiver
    - Application is not always ready to accept a packet when it arrives from the network
    - Network may deliver data in bursts
  - Network transit time
    - Fixed propagation delay

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 and Delay

- 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** is the time taken to deliver a packet across the network
  - 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

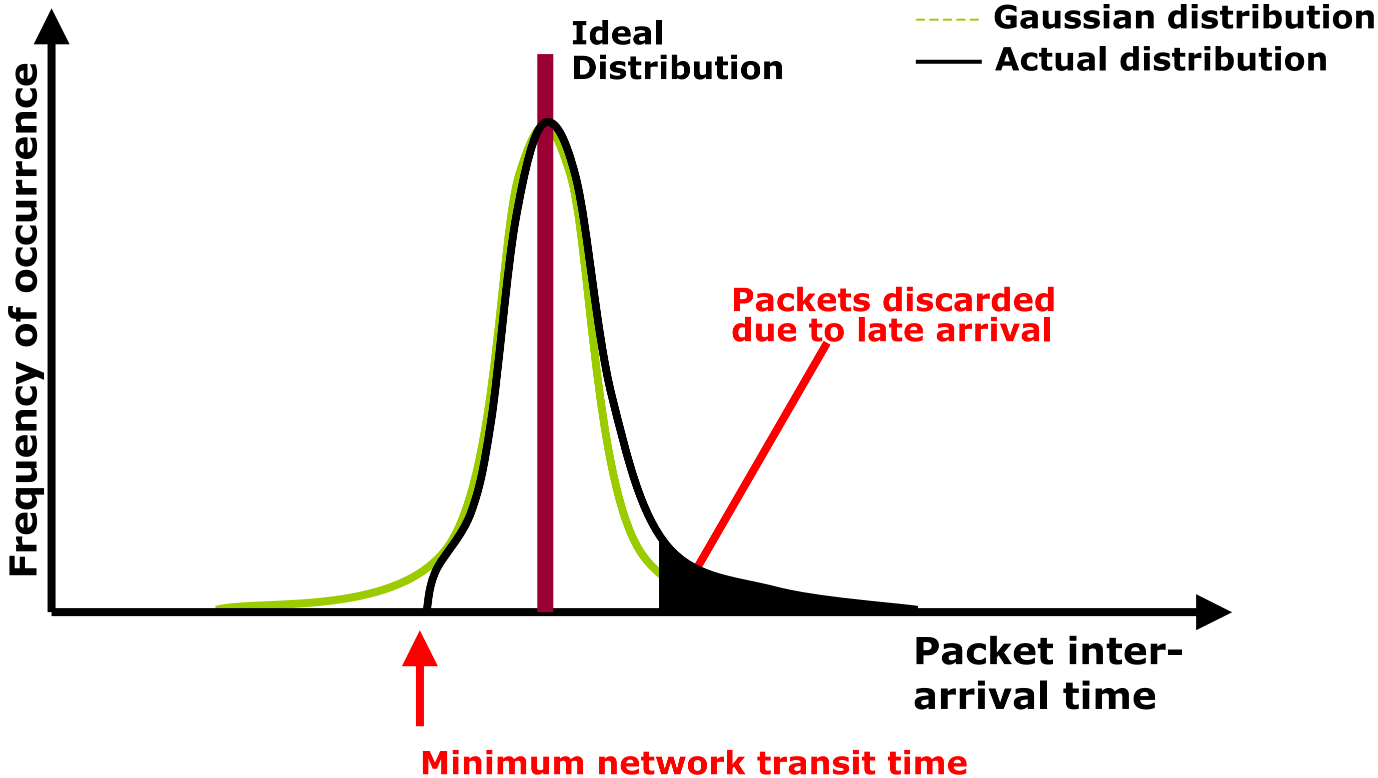
Throughput and Delay

- 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 $\Rightarrow$ controller $\Rightarrow$ 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
  - Talked earlier in the module that 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 are designed to keep not only latency, but 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!

Loss

- Throughout, we have assumed that no job 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

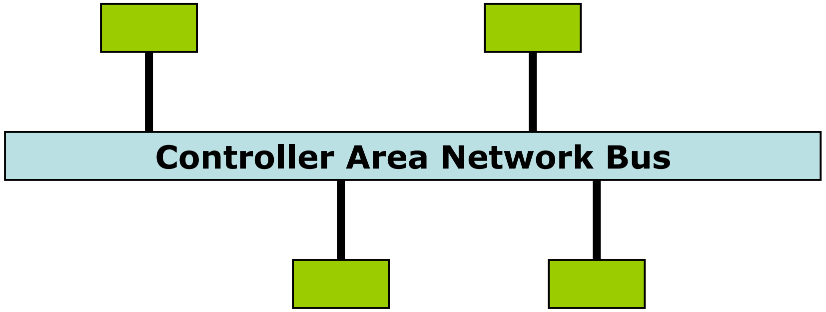
- Packets may also be dropped due to corruption or other errors
  - Not discussed further, since not affected by scheduling

Characterisation of Networks

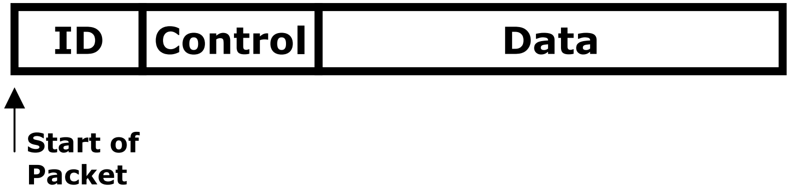
- Implication: for real time communication, it is necessary to characterise the timing behaviour of a network
  - Prove/demonstrate that throughput, latency, and jitter are within appropriate bounds for the application
- Some network technologies allow this, others do not
  - Examples: CAN, Ethernet

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
- EDF scheduling has been suggested as an alternative, with better utilization

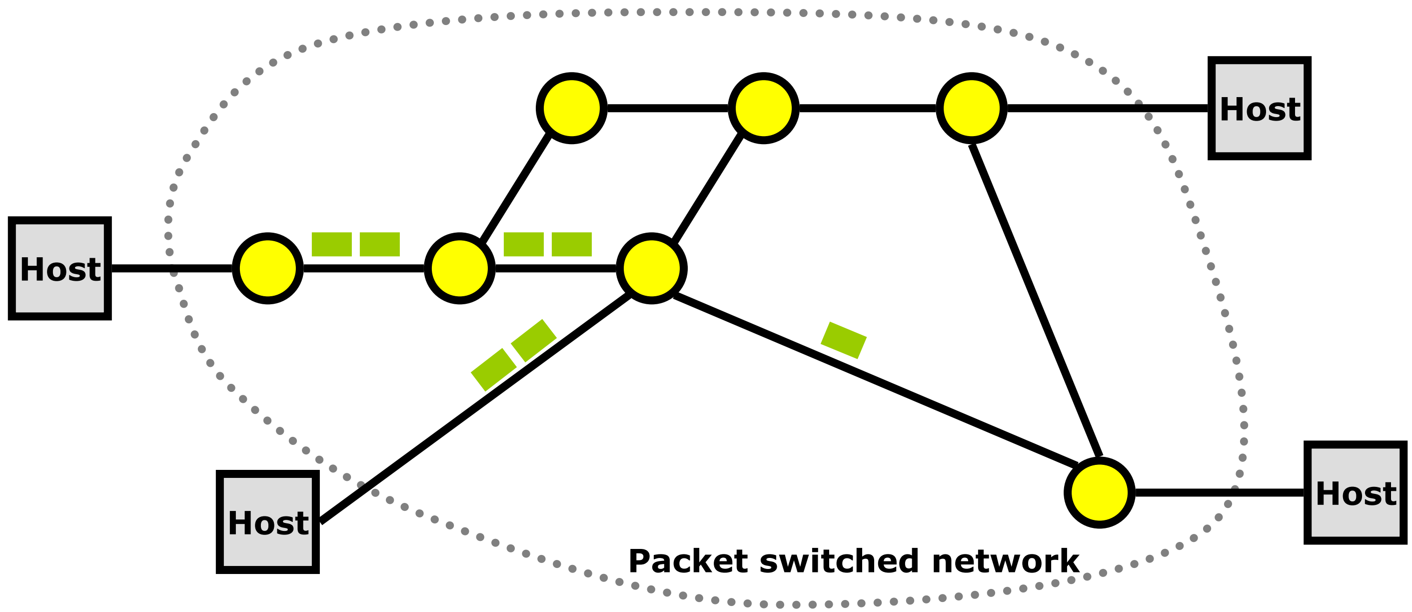
Example: Ethernet

- Recall that Ethernet uses CSMA/CD with exponential backoff
  - 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

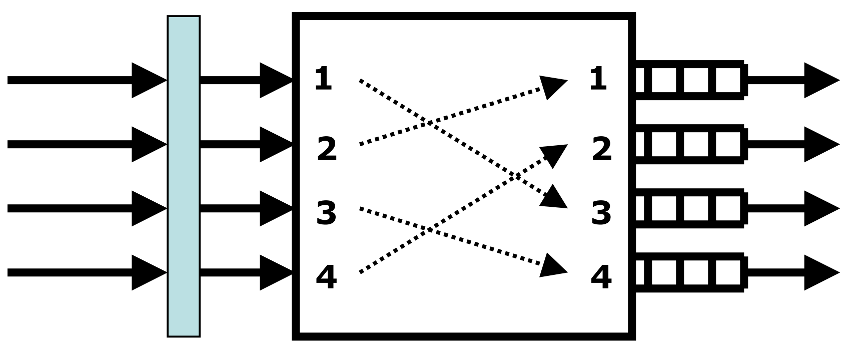
Extension to Packet Switched Networks



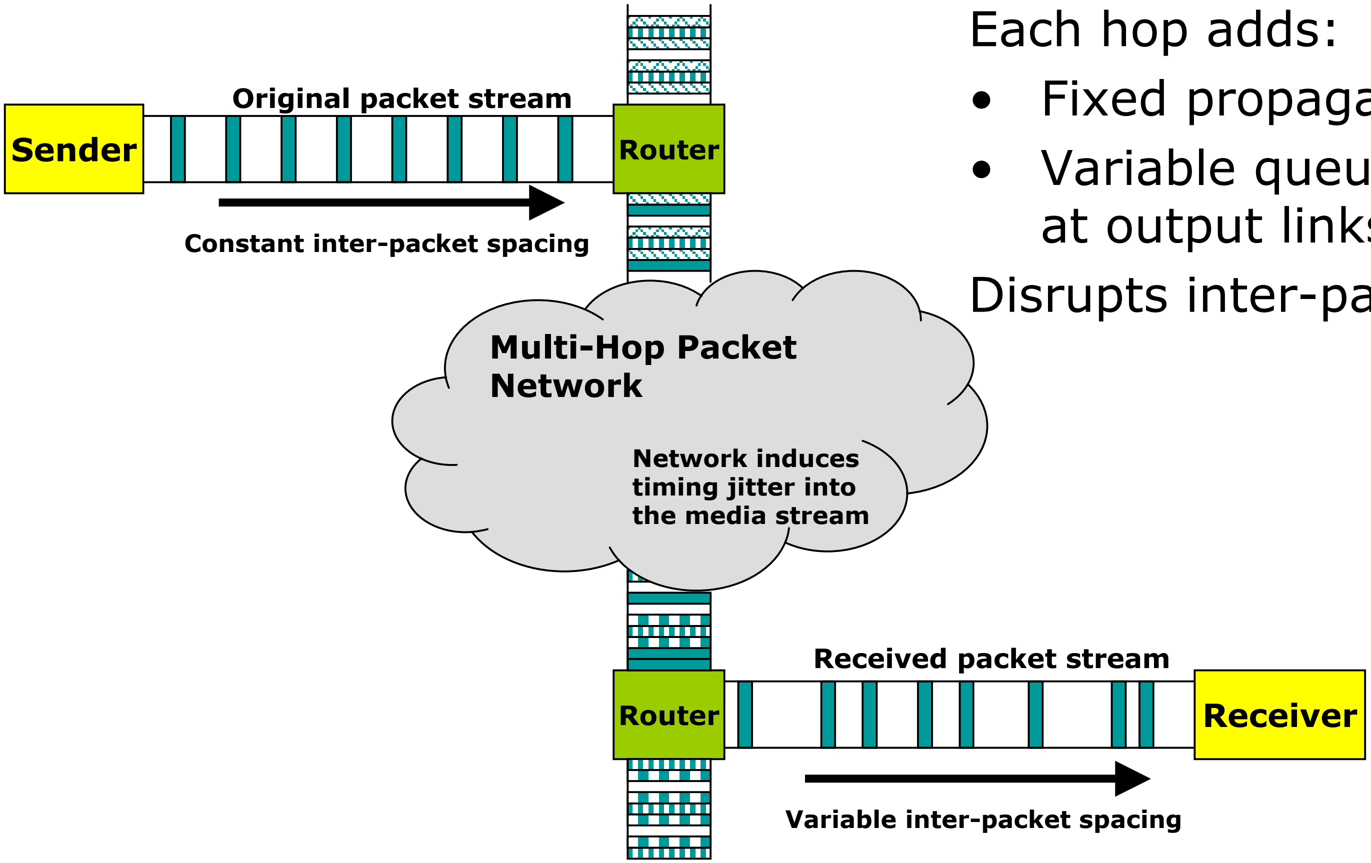
Links have constant **propagation delay**

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

Choice of job scheduling algorithm on the output link is critical for real time traffic – tomorrow!



Timing Recovery



Each hop adds:

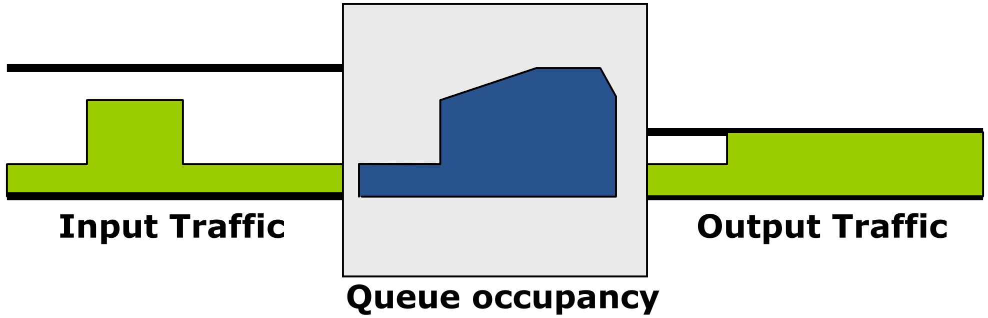
- Fixed propagation delay
- Variable queuing delay at output links

Disrupts inter-packet timing

- Consider the **throughput**, **latency** and **jitter** of the network with different scheduling algorithms in the routers
- Want to minimize the jitter and latency, while keeping sufficient throughput

Throughput, Latency and Jitter

- Should be clear that throughput and latency 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 of the real-time flow 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 there is capacity, unless requirements are signalled in advance

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

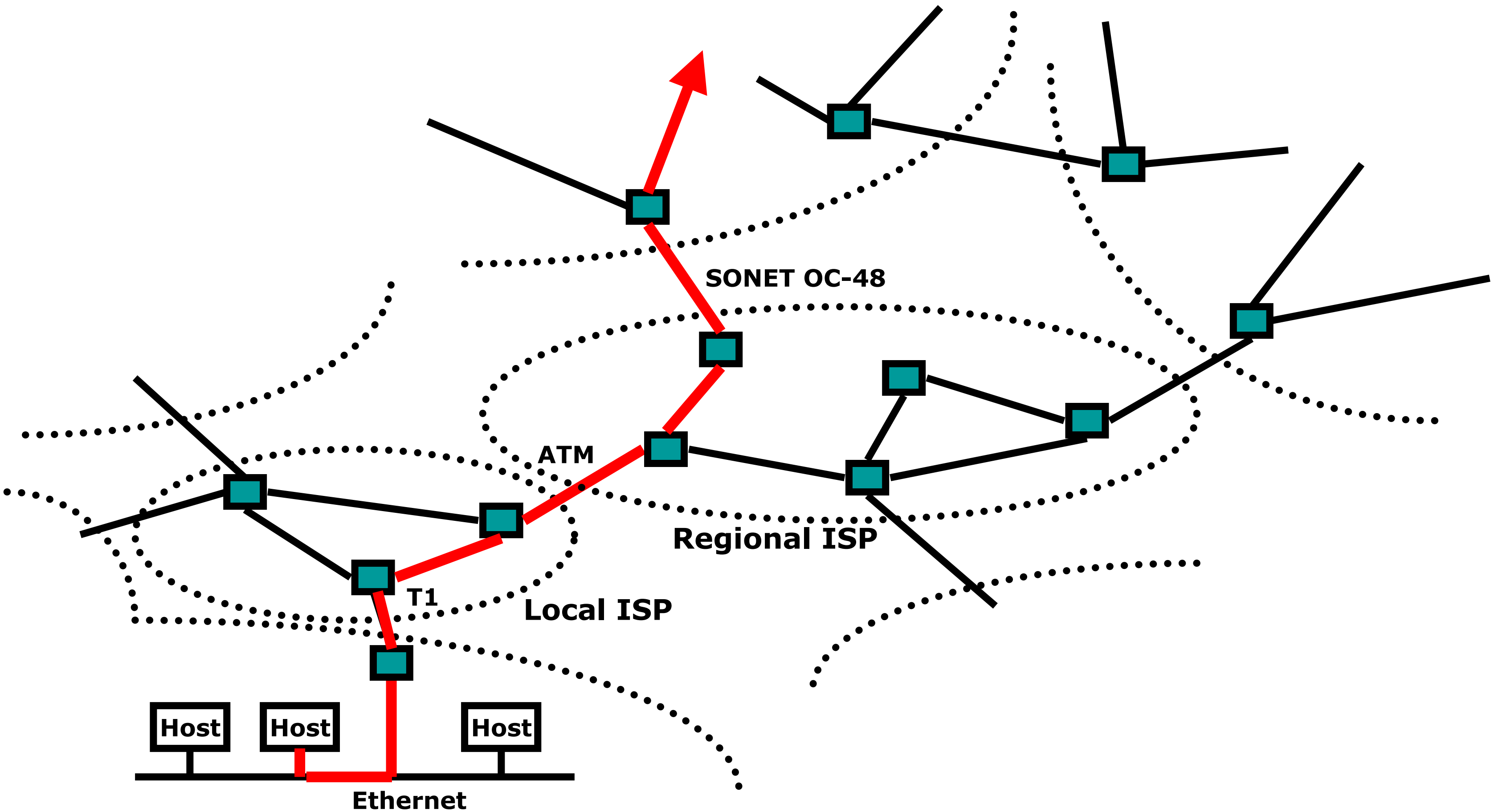
Clock Skew and Synchronisation

- In a packet switch 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

Example: IP Networks and the Internet

- The Internet is an example of a packet switched network with uncontrolled queuing behaviour
  - Different hops use link layers with different timing behaviour
    - Ethernet
    - SONET
    - ATM
    - DSL
  - Different queuing algorithms in routers
    - FIFO with drop-tail
    - FIFO with RED
    - Weighted fair queuing
    - Weighted round robin
- Observed performance depends on the underlying network, the route taken, and the characteristics of the cross traffic
  - Difficult to predict in advance, unless you control the network

Structure of the Internet



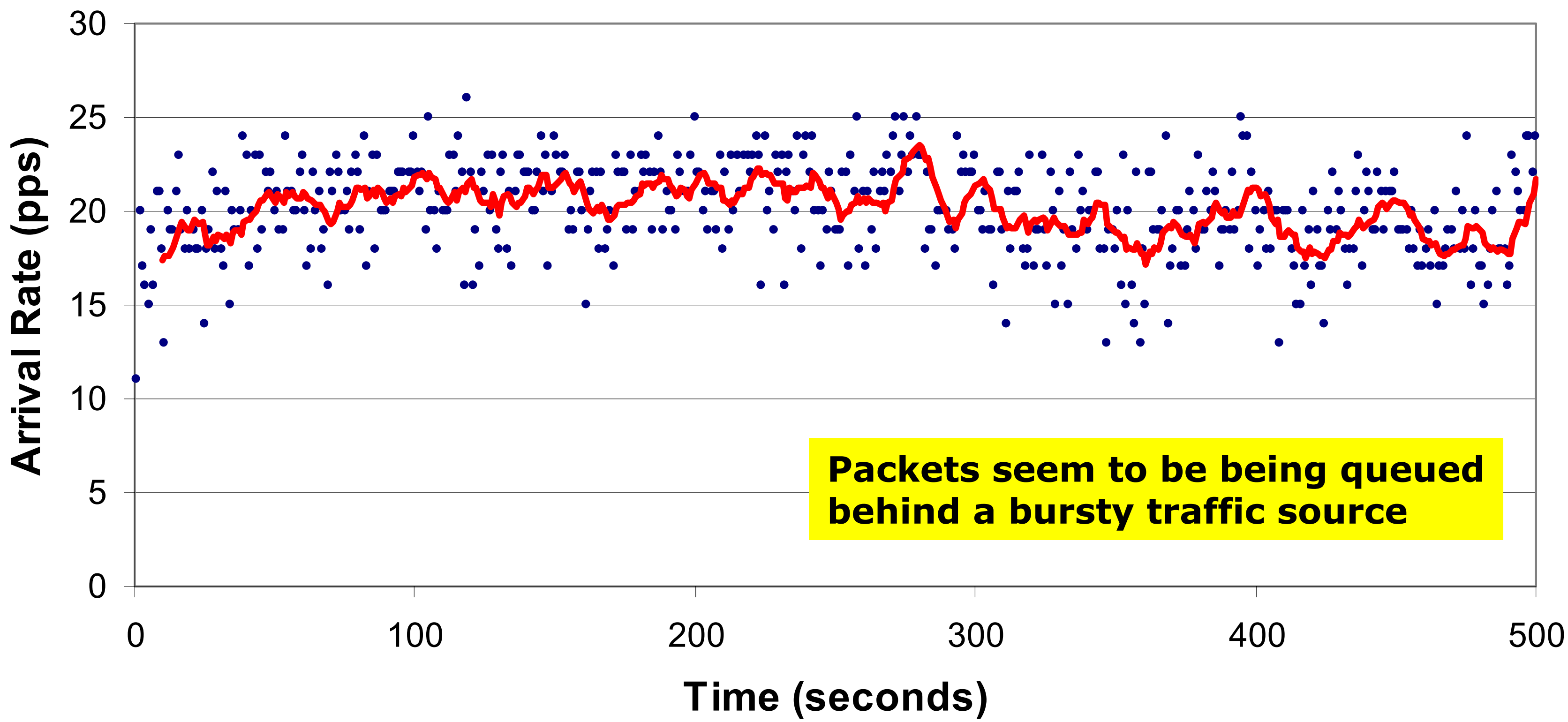
- Traffic passes through many hops, which can be maintained by different ISPs. How is the packet timing affected?
- Do you have an SLA with each?

Sample Internet Measurements

- Tests using a multimedia application running between the University of Oregon and University College London
  - An Internet TV station, re-broadcasting NASA TV
  - Measurements taken in 1999 and 2000
    - Recent measurements to less well connected sites look like these
    - Recent measurements in Europe, US and Japan often show less variation due to improvements in capacity
      - Although many DSL and cable modem providers have poor quality networks, and measurements look a similar to these

- Observed the audio traffic at the IP layer
  - Constant rate (isochronous) traffic source
  - Packets generated by a periodic task every 20ms
  - Desired behaviour is constant arrival rate, no jitter and no clock skew

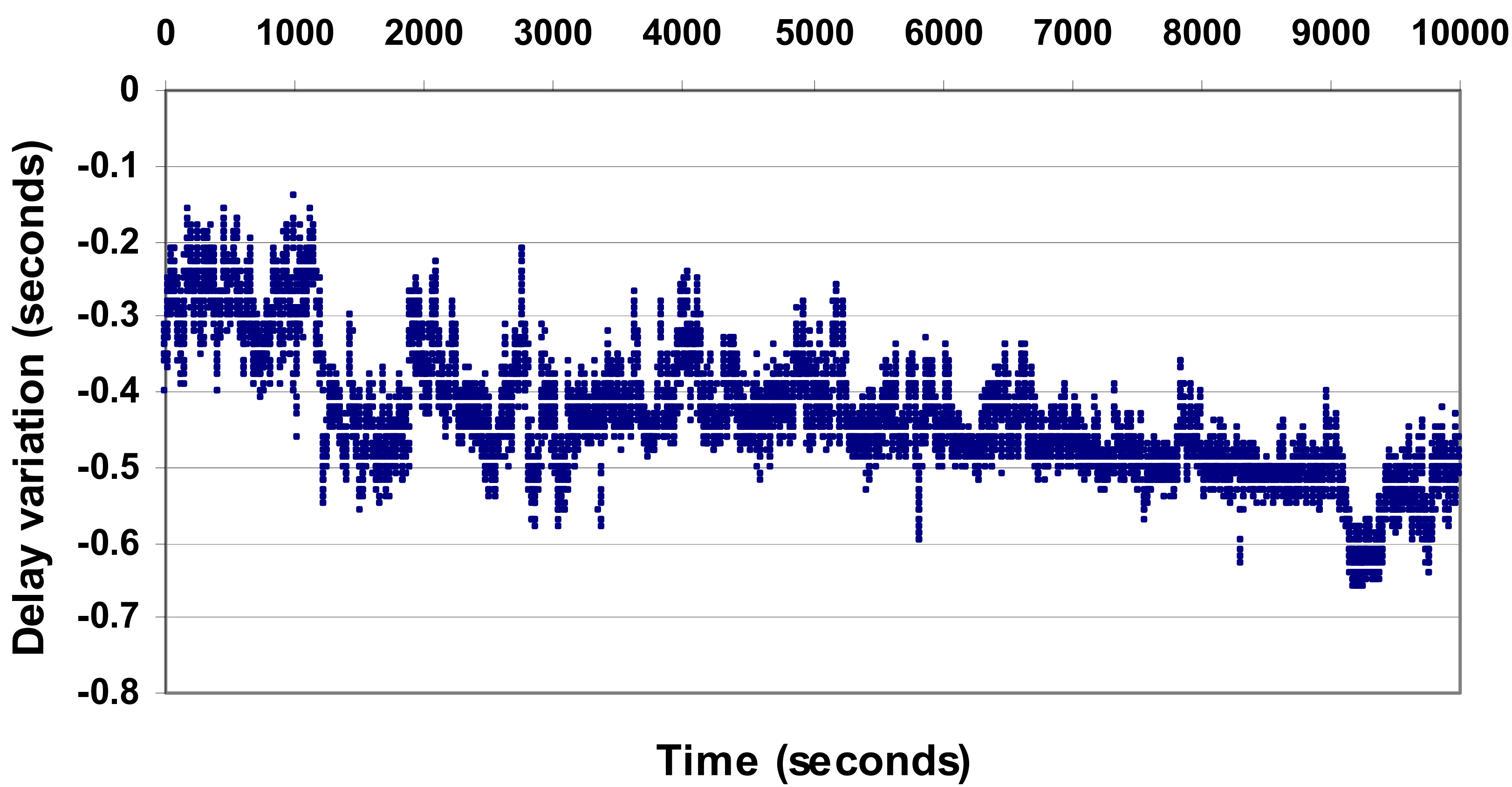
Throughput Variation in an IP Network



Blue points show a 1 second average

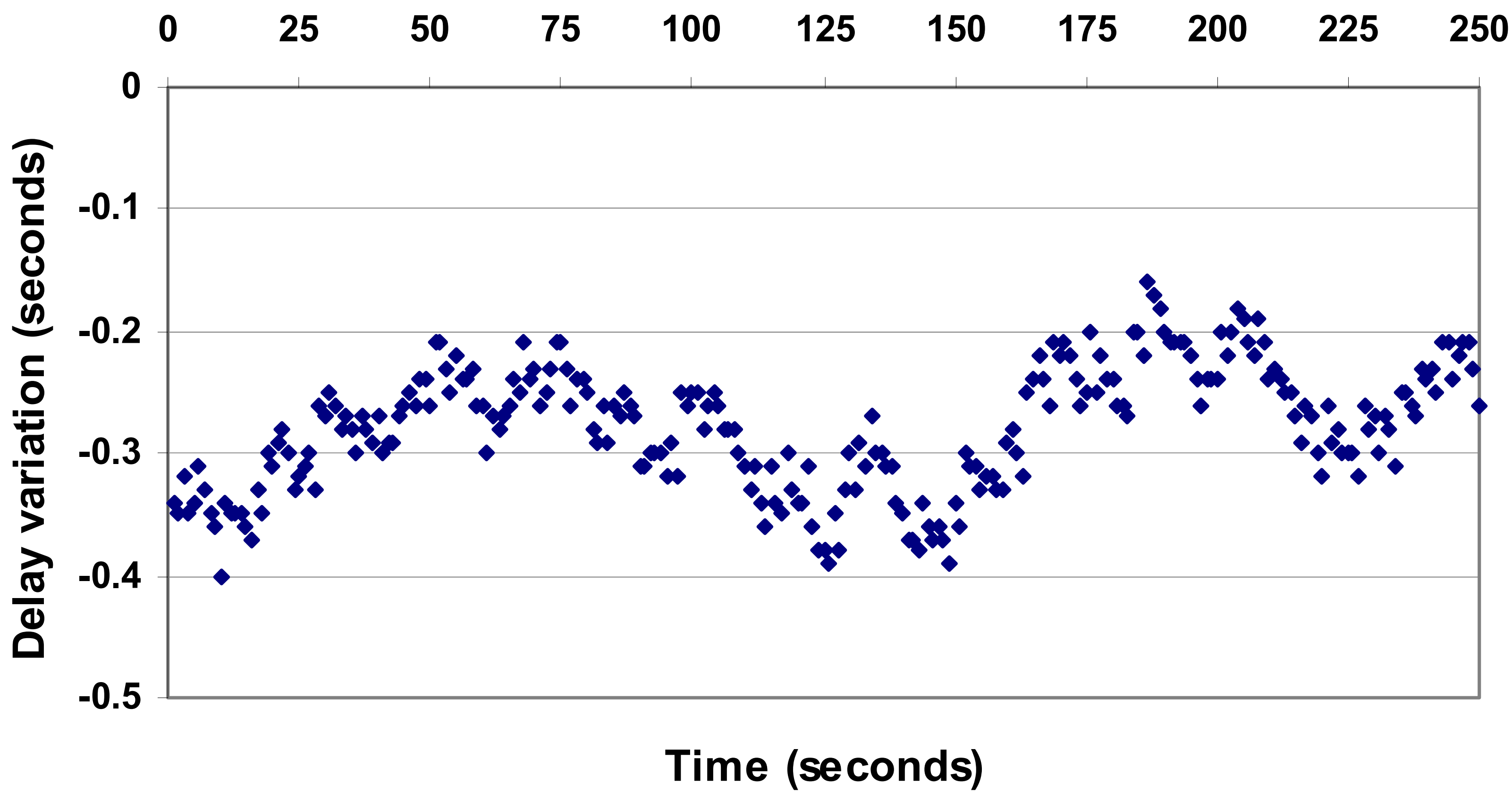
Red line shows a 10 second moving average

Jitter in an IP network



Track jitter from queuing delays over almost 3 hours
Downward trend due to clock skew

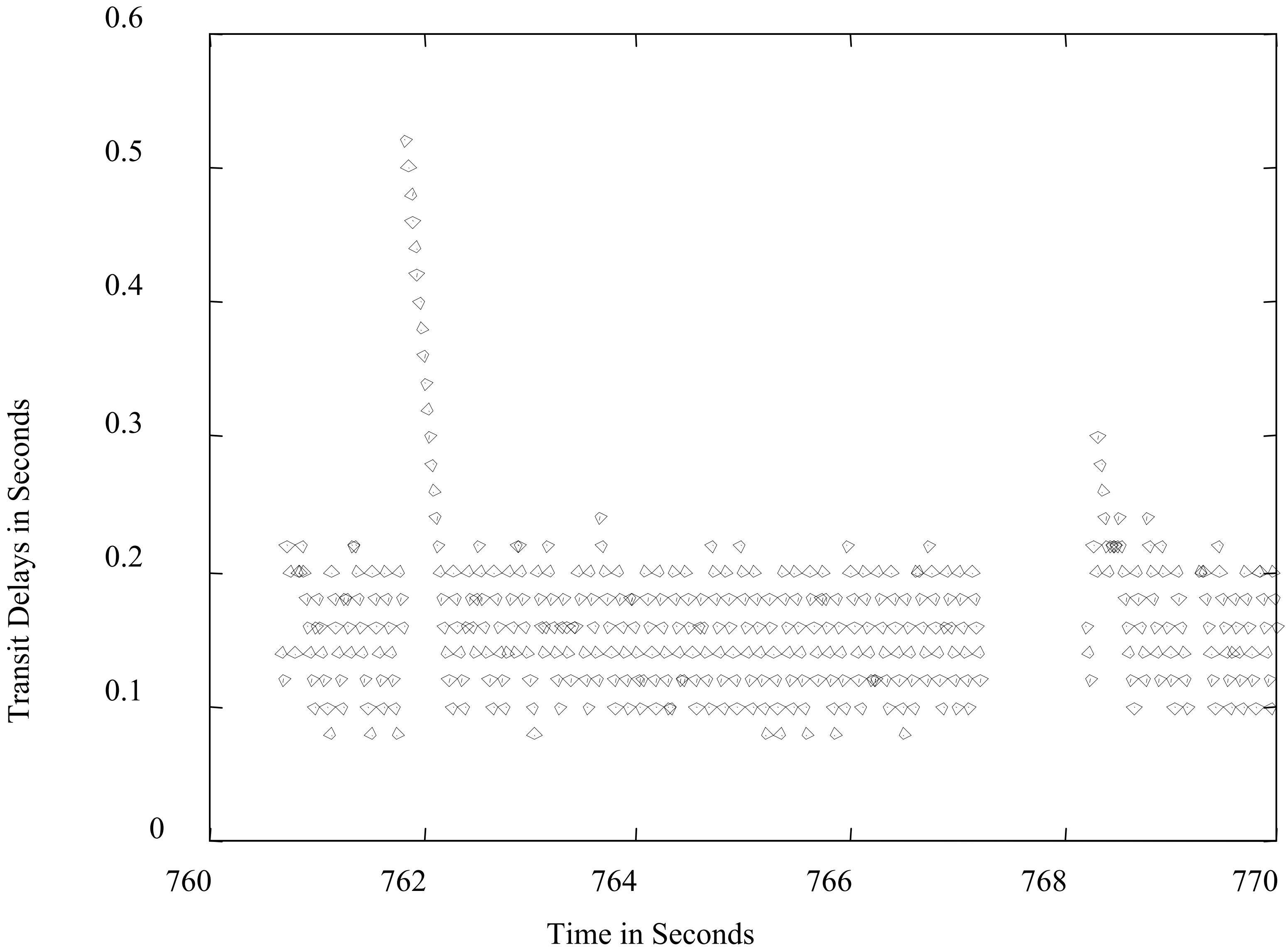
Jitter in an IP network



Magnification of first 4 minutes of the previous graph

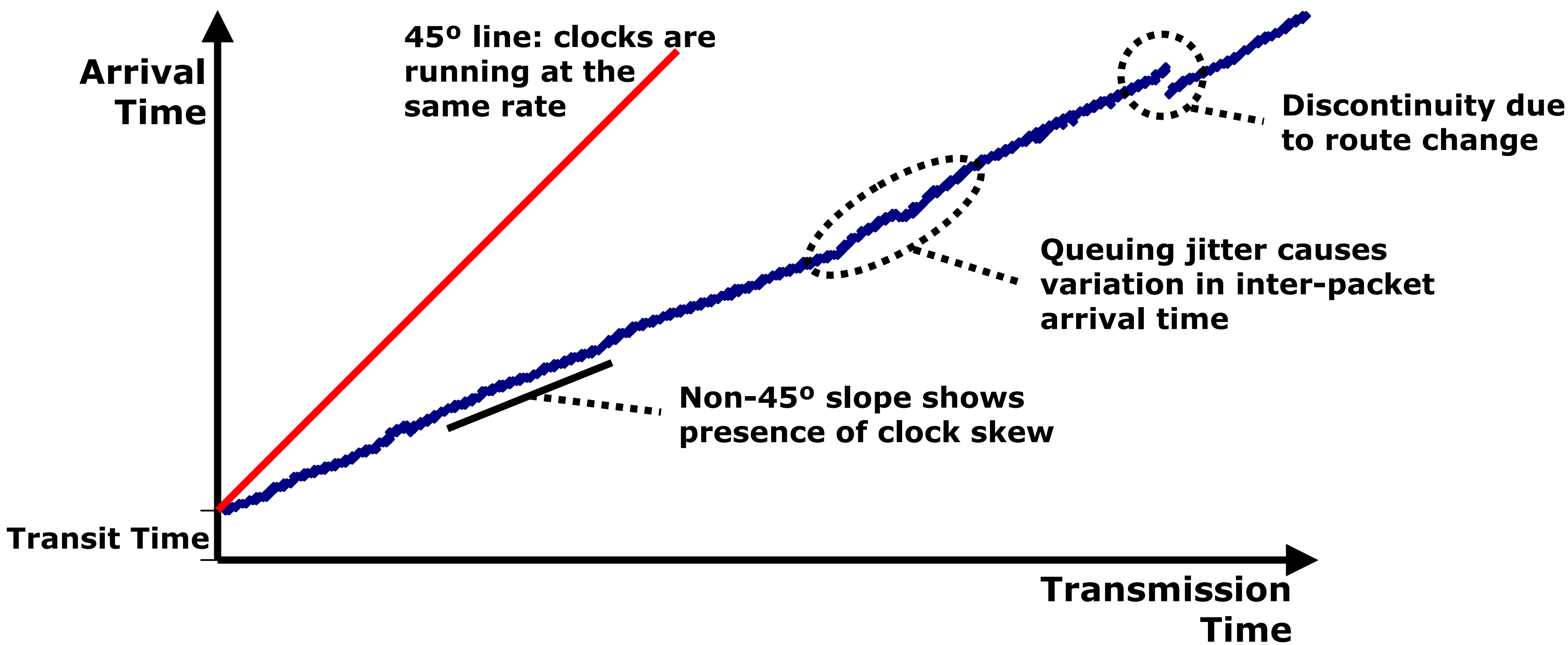
Influenced by overlapping periodic processes

Jitter in an IP network



Packet Timing Graph

- Can visualise all these effects using a **packet timing graph**
- Plot transmission time versus arrival time



Predictability of the Network

- Should be clear that timing of an IP network may be roughly characterised, but is not sufficiently predictable for hard real time applications
- Is this a property of all packet networks? No, it occurs because:
  - Uncontrolled service discipline; no jitter or rate control
  - No admission control/connection establishment
- Other packet networks may be different:
  - ATM and MPLS have a connection establishment phase to allow resources to be reserved for a new flow
  - RSVP allows you to reserve resources on an IP network
- This is the difference between best effort networks, and networks that provide enhanced quality of service (QoS)
- When using packet networks for real time communication, important to either characterise the best effort network or use appropriate QoS

Summary

By now, you should know:

- What is real time communication
- Factors that affect real time communication
- Examples of networks and their timing properties

Tomorrow, we'll talk about resource reservation and QoS