MANCHESTER 1824
The University of Manchester

# COMP28411 Computer Networks
# Lecture 8

Nick Filer

Multimedia - 1

Some material from:

Kurose & Rose – Chapter 7 + Slides

Halsall – Multimedia Communications

03/11/2016 COMP28411 NPF MM1 1

## Overview

- What is it?
- How do we get it?
  - Telephone networks
  - Data networks
  - Broadcast networks
- Delivery Methods
  - How does streaming work?
  - Buffering
  - Jitter
  - Tight real-time 2 way dialogue

MANCHESTER
1824
The University of Manchester

# What is MultiMedia?

https://www.youtube.com/watch?v=T7P-fwgUYbg

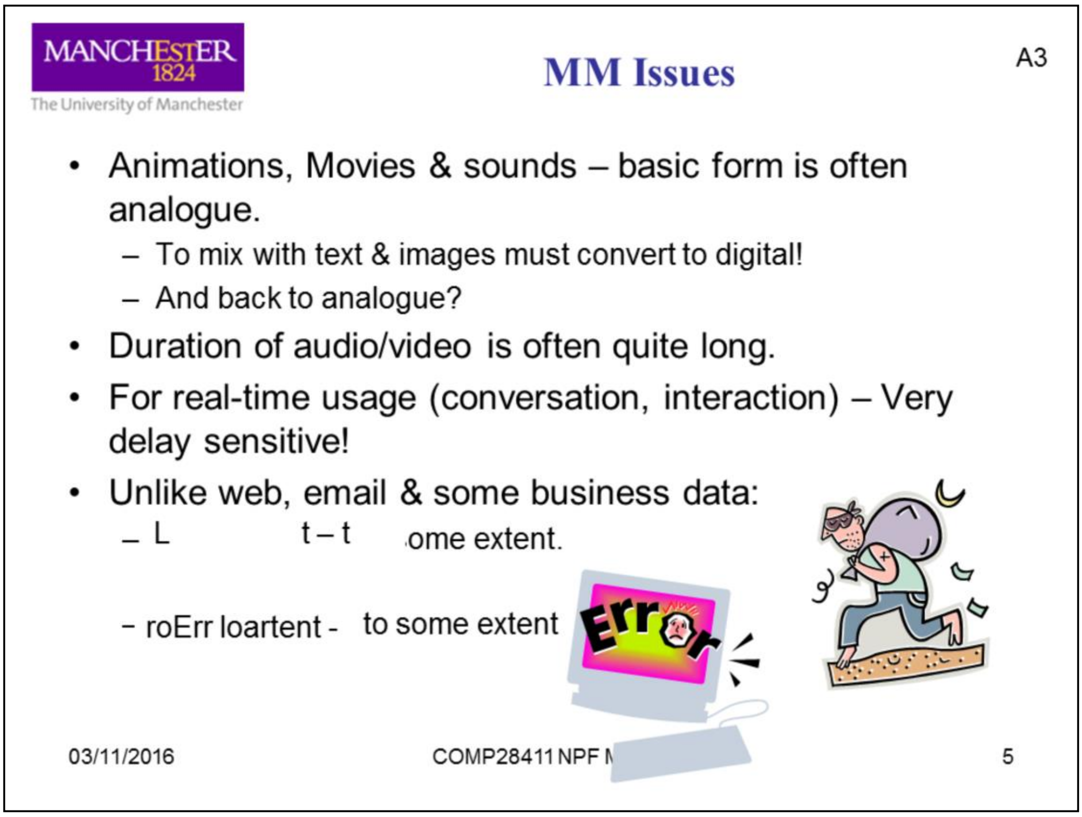



Almost all types of data can be considered to be multi-media whether on their own or mixed with other types. As computer scientists we tend to think of digital media forms first but must not forget that all media is produced by us and experienced by us as analogue signals. So sound is an air wave vibrating at a set of frequencies varying in amplitude with time. Light is electromagnetic radiation again consisting of a set of frequencies varying in amplitude with time. Media does not need to be digital in order to store it and transfer it.

There are some media (?) we sense that so far have proved difficult to convert to and from electrical signals. Two examples I thought of are smell, the ability to detect often minute quantities of specific chemicals and thoughts




In order to experience a piece of text, it can arrive at almost any rate and we still get it with very high accuracy. We notice mistakes in text, so we arrange to usually only look at text that is complete, in the correct order and all of it is present. Luckily, most of the text we use is quite small. Even big documents are not that big. The King James Bible, I'm told is about 3MB, The complete works of Shakespeare about 5MB.

In order to reproduce images and sounds we must play the picture(s) and/or the sound(s) at very close to the same rate at which they were captured. However, if a bit of text is missing, we notice it and it changes the semantics of the document which can become unreadable with relatively small/low losses. Images and sounds however, can be reproduced at much lower accuracy and we can often still work out exactly what it was originally. Errors in many forms of multi-media can be allowed but unlike other forms of media many of these types of media are very delay sensitive.

In Moodle, I have uploaded 3 audio files of the same 16 seconds of a telephone call. The file sizes are 16.3KiB, 646.9KiB and 2.8MiB. I'm not sure I can hear any difference between them!

In contrast, the two messed up lines of text above have missing and wrong characters but in terms of lost information are much closer to complete than the two reduced resolution/quality audio samples above.  The text says "Loss tolerant – to some extent" and "Error tolerant - to some extent".

MANCHESTER 1824
The University of Manchester

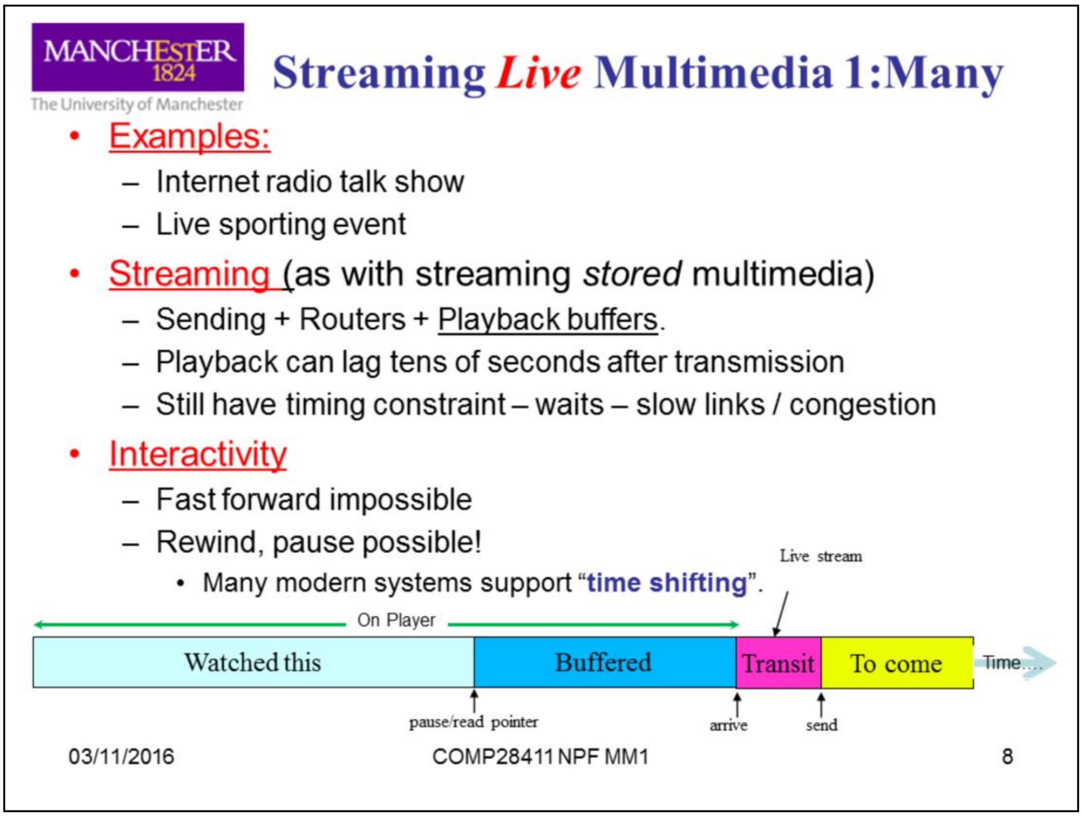
## Problem ????

# HOW DOES STREAMING OF MULTI-MEDIA WORK?

- What protocols are used?
- Do ubiquitous protocols (TCP/UDP) work well?
- What problems are caused by:
  - Large files?
  - Network congestion?
  - User requests to jump backwards (repeat) or forwards during playback?

03/11/2016 COMP28411 NPF MM1 7

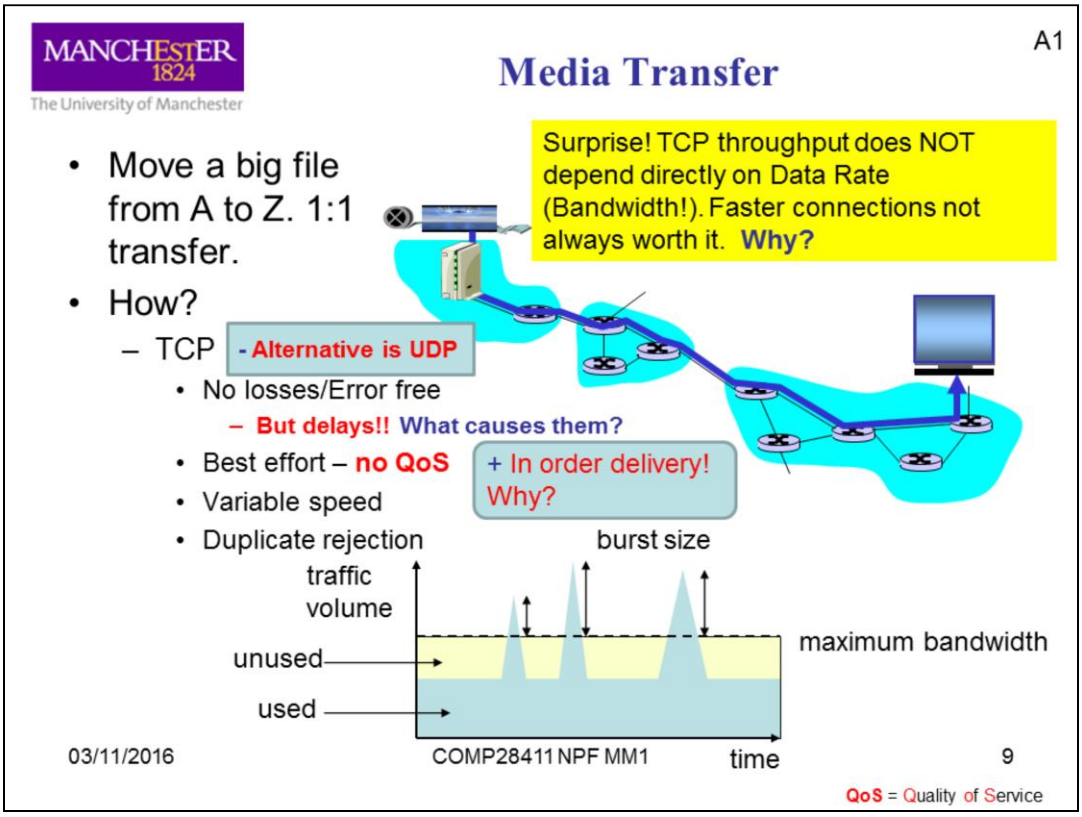



Multi-media creates lots of different issues for the Internet. Live media is often sent by radio and TV stations. At any point in time the number of listeners/watchers can be very large or very small. The numbers can change rapidly. Similar issues occur with streaming sites though here the media itself is not live which means that if high demand is likely or demand is, for example, geographical in nature such as a show in French which is probably be of interest in only/mainly French speaking areas of the world, the data can be pre-distributed to local distributed caches.

Live data cannot be pre-distributed in many cases. By live we do not actually mean "live". We never experience happenings exactly as they happen. There is always a lag. In analogue broadcasting the lag was small and due to capture, processing, transmission, reception and rendering of the data. We normally accept that if the delay is similar to the delay, for example, due to the speed of light then we experience the live event as it happens. Luckily, as sound (approx. 343m/s) travels much slower than light ($3x10^8$m/s) in air, unless we are close to the sound source we do not always perceive it as being live. we see lightning often well before we hear it. Different people will have different opinions of these views!

Unfortunately you cannot jump forwards or play in fast-forward live events. I'd be very rich if this was possible. We can store live media to watch later which is similar to caching a streamed recording. If we jump forwards or backwards in a streamed recording then the lag before the playback resumes will depend on how much data is cached locally and how fast the data can be searched and read. Remotely stored streaming data will always incur much higher lag if we jump around the recording. Hence, downloading before playback is often preferable to playing the streamed data straight away.



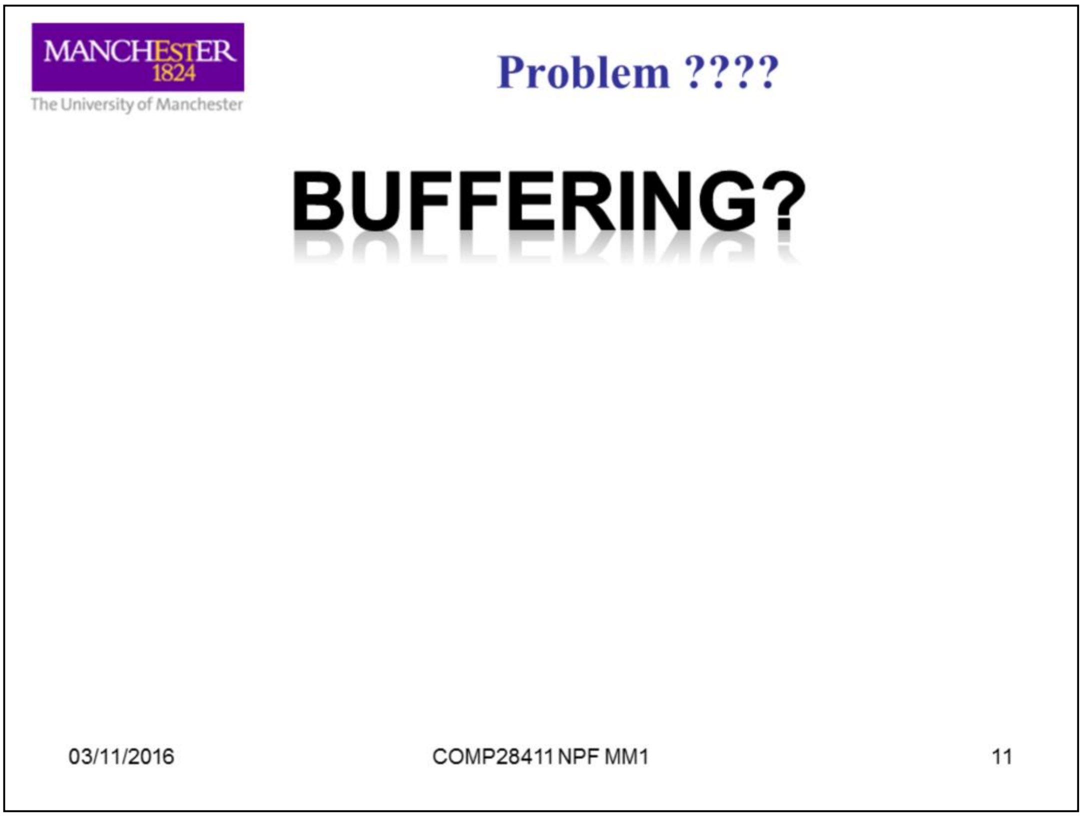

Media transfer is almost always the task of moving one or more big files from source to destination. The transport protocol designed for this task is TCP which has some desirable features for moving data around such as error free delivery in the same order that the data was sent. As you should already know, the Internet offers a “best effort” service to packets. In practice this means the rate of delivery for data can be very close to the available bandwidth between the source and the destination when there are no other competing users for the connections and routers between the source and destination. When there are competing users the available bandwidth must be shared which may reduce the rate users experience. If, the amount of traffic exceeds the capacity of any of the connections or routers for individual data streams then the normal practice is for devices to throw some data away. Normally, if data arrives at a router whose input buffer is already full, the newly arriving data is dropped. This is called “tail-drop”.

Unfortunately, TCP is very sensitive to latency or delay between source and destination and also to any data losses. TCP assumes that data loss is ONLY caused by congestion. This is an OK assumption on wired networks but not on any current form of wireless network including WiFi, mobile phone and satellite.

TCP Throughput = Window Size (WS) / Round Trip Time (RTT).

Older TCP implementations allowed window sizes up to 64K. In modern networks the window size can be scaled to better match the higher bandwidth available.

-------------------- continued on slide 12 ------------------------------------

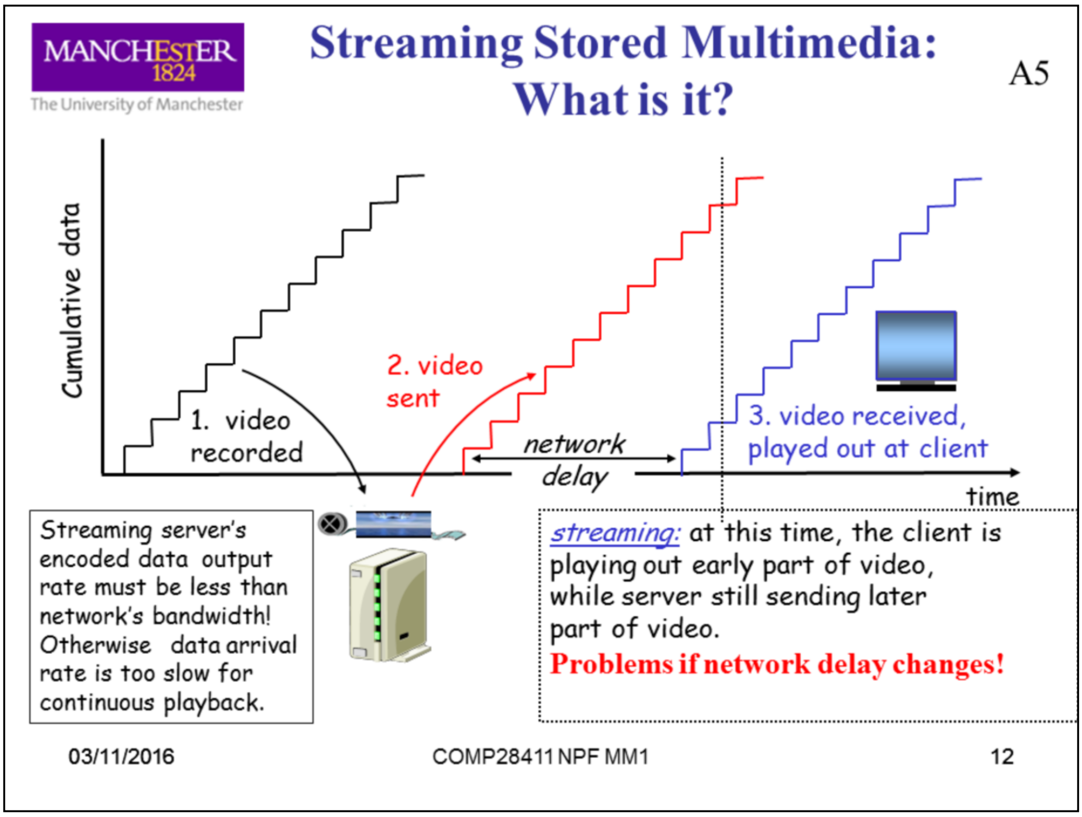



Nothing in our world/universe is totally synchronous. Over networks there are always delays. If no components change, the temperature is fixed …. Then in theory at least the point to point transfer time is fixed by the speed of light for electromagnetic communications. But, be aware, the speed of light though a constant for a given medium varies with the type of medium. The speed in a vacuum or air is almost identical and we tend to estimate it conveniently as $3x10^8ms^{-1}$. In wire or fibre it varies and is almost always considerably slower than in air. The Velocity Factor (VF) is used to specify by how much it varies. Figures between 0.66 and 0.8 are common for wires and fibres. Different batches of wire/fibre from the same manufacturer will often have different but similar VFs'.

In practice, packets over a network take different routes through the many switches, routers and the wires or wireless connections between them. Therefore there are always going to be variations in the delays (or latency) between adjacent packets.

Most media is sent using a near Constant Bit Rate (CBR) from the source. The various differing delays mean then data arrives with varying packet separations at the destinations. In order to play the media back at the same rate as it was sent, it is therefore necessary to introduce some extra delay to allow for delayed packet arrivals. This means that the media data has to be stored or buffered or cached somewhere.

Data can be stored permanently, semi-permanently, or temporarily at various places between the source and destination. It can also be stored at the destination in various buffers within the network stack, the applications, the OS and the media playback hardware.




Streaming is very sensitive to delay and variations in delay (called jitter).

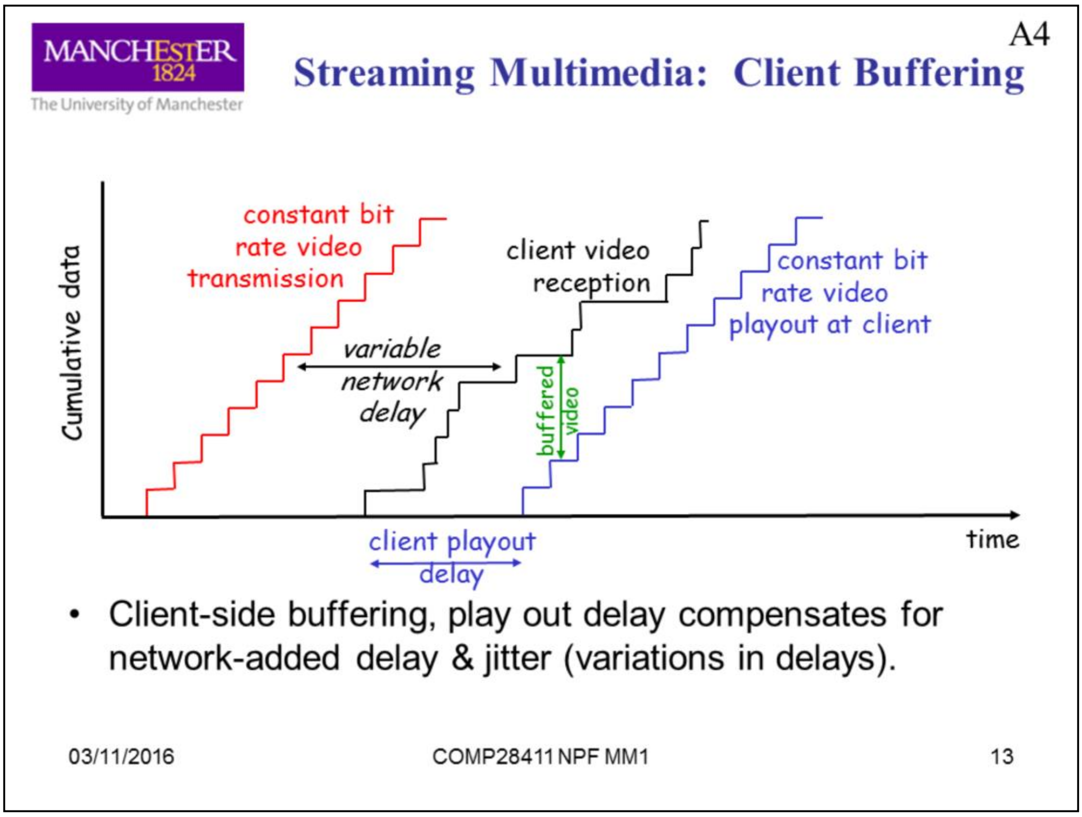
1. Video recorded
2. Video sent
3. Video RX and played
4. Mark NW delay time
5. Have constant NW delay here. Works well.

---------------- continued from slide 8

Due to the normal windowing mechanism this means that most of the time the window size will be 32K. Imagine two data centres connected by a 1G link with a latency/delay of 30ms. In this situation the maximum data rate would be 17.4MB/s, nothing near the 1GB we might have expected. Clearly there is little can be done about the latency, it is due to the speed of light. The window size could be much larger. Using bandwidth * latency = Window Size, we conclude a windows size of 3.75MB will allow all the available 1GB of bandwidth to be used. Large windows need large buffers which might need to be retransmitted if there is an error anywhere in the window.

Maximise Window Size works provided NO errors or dropped packets.

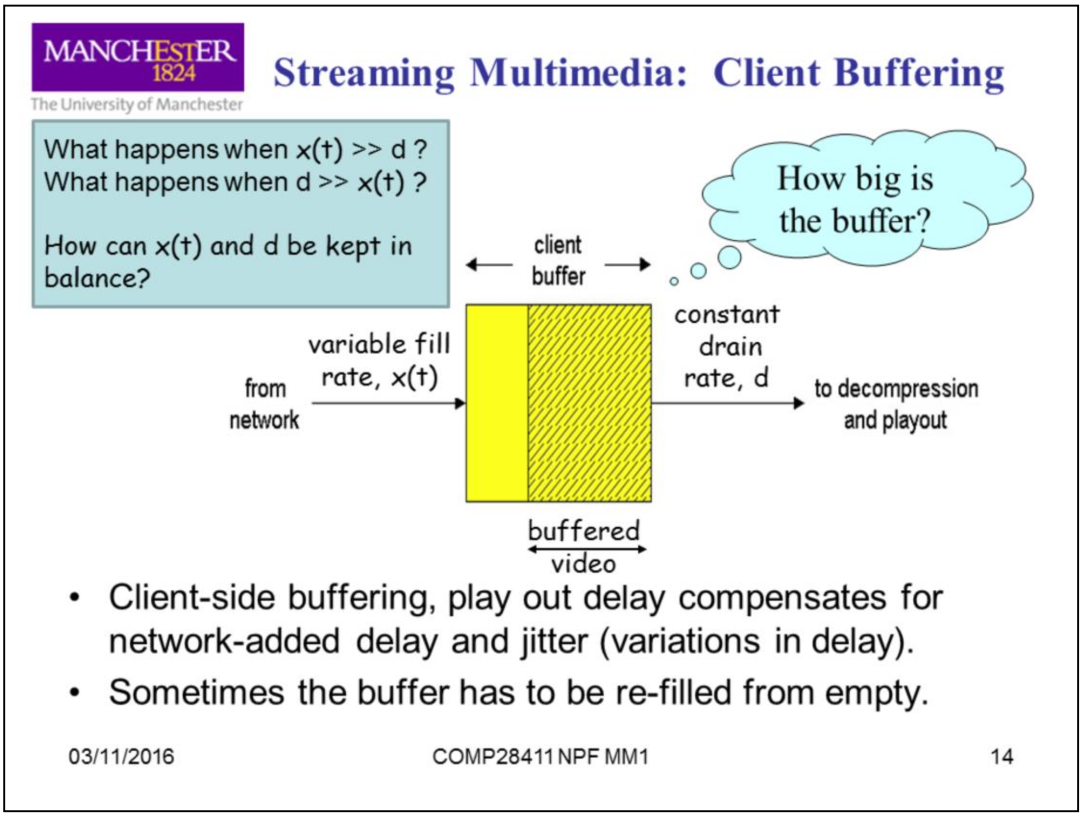
 




The previous slide showed a perfect network where the traffic always took exactly the same amount of time to travel from the source to the destination. In practice, the network delay differs for lots of different reasons such as the loads on the source & destination, the size of queue at intermediate routers and traffic being sent over different routes at different times so as to avoid congestion. The variation in packet to packet delay is called jitter (an imprecise/ambiguous term) or simply variation in packet delay. Sometimes you are given the instantaneous packet delay variation (IPDV) where if we image packets are sent every 20ms at a constant rate but then a packet arrives 30ms after the previous one at the destination this gives an IPDV of -10ms and is sometimes also called "dispersion" because there is a longer delay. Similarly if a packet were to arrive only 10ms after the previous one its IPDV would be +10ms which is also called "clumping" because if it continues you will get a cluster or clump of packets all available within a short time period.

For streamed traffic, packet delay variation can be removed by buffering traffic at the destination before play-back. The buffer is used to absorb the differences in timing and must be larger than the maximum variation (jitter) experienced during the streaming in order for playback to be unaffected. Practical systems such as streamed TV tends to use buffers that are several seconds in size but at my home we certainly see occasional breaks in streamed films and TV programs due to congested networks.

For 2 way live traffic such as VoIP, a small buffer is used to mitigate IPDV effects but for satisfactory long-term results the VoIP stream needs to be given Quality of Service (QoS). Paid telephone networks reserve Internet capacity for voice traffic which is given QoS priority. Many effectively unpaid VoIP services such as Skype rely on normal best-effort service though even here some routers and servers will prioritize small or explicitly voice labelled traffic over other types of traffic when deciding which of many alternative packets to forward next. Of course QoS breaks the first in first out (FiFO) queue with its implicit almost fairness!



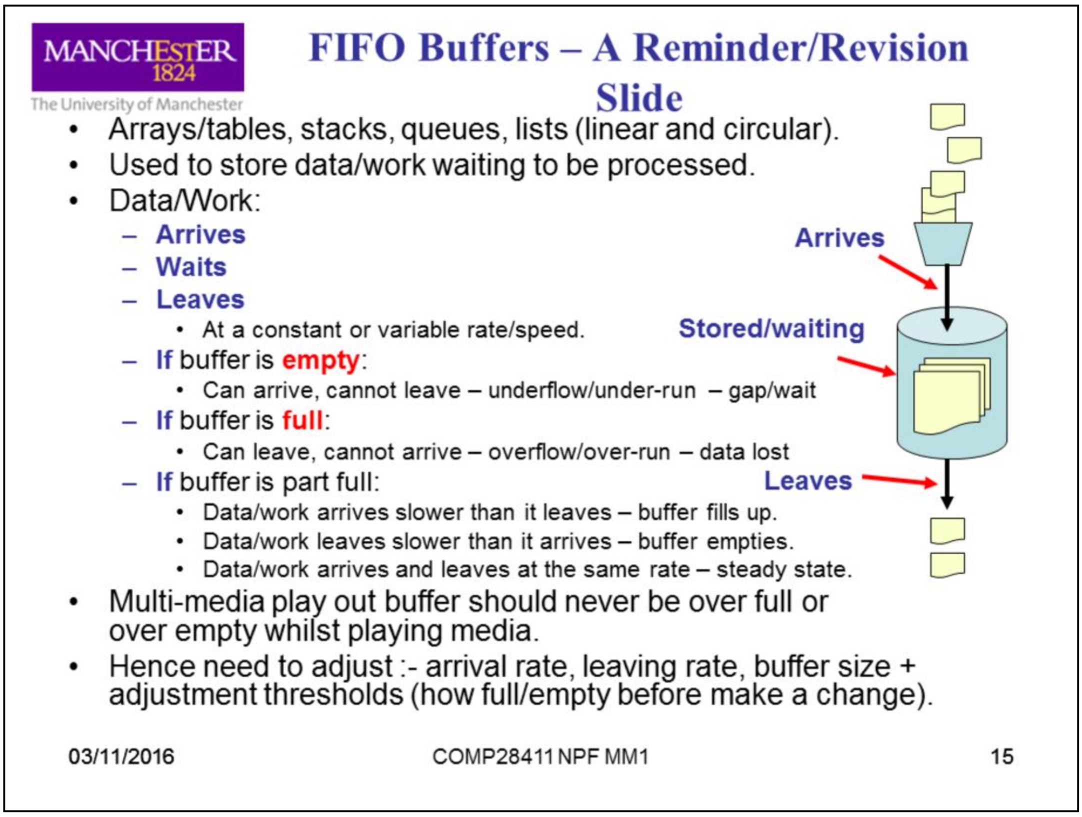

The buffer is normally a simple array in memory organized to be circular so that its implementation is efficient. It models a simple queue. If data arrives at the queue more quickly than data is removed from the queue then the queue fills up. If some data arrives at an already full queue there is nowhere to store it and it is discarded. If data arrives slower than the rate at which data is removed from the queue then the queue shrinks until it is empty.

For constant play-back it is a severe problem if there is nothing in the queue to be played, this leaves a gap in the output. If the buffer is overfull then some data will need to be discarded which may also leave a gap in the playback even though the data to be played after the current gap may be stored in the buffer.

Because packets can arrive either in or out of order but will be used in-order for playback, at the destination (client) we do not keep a strict FIFO buffer like those used in routers, instead we insert packets into their correct position in the queue of packets awaiting playback.

If a gap in the buffer occurs due to a missing packet this is OK until we must remove the packet for playback. At this point we have some choices, we could stop and wait, we could terminate the playback, we could try packet loss concealment (PLC). PLC works by perhaps playing the last played packet again, or playing the packet following afterwards twice if it is ready or by trying to average the packet before and after to produce a synthesised packet or simply leave a gap.

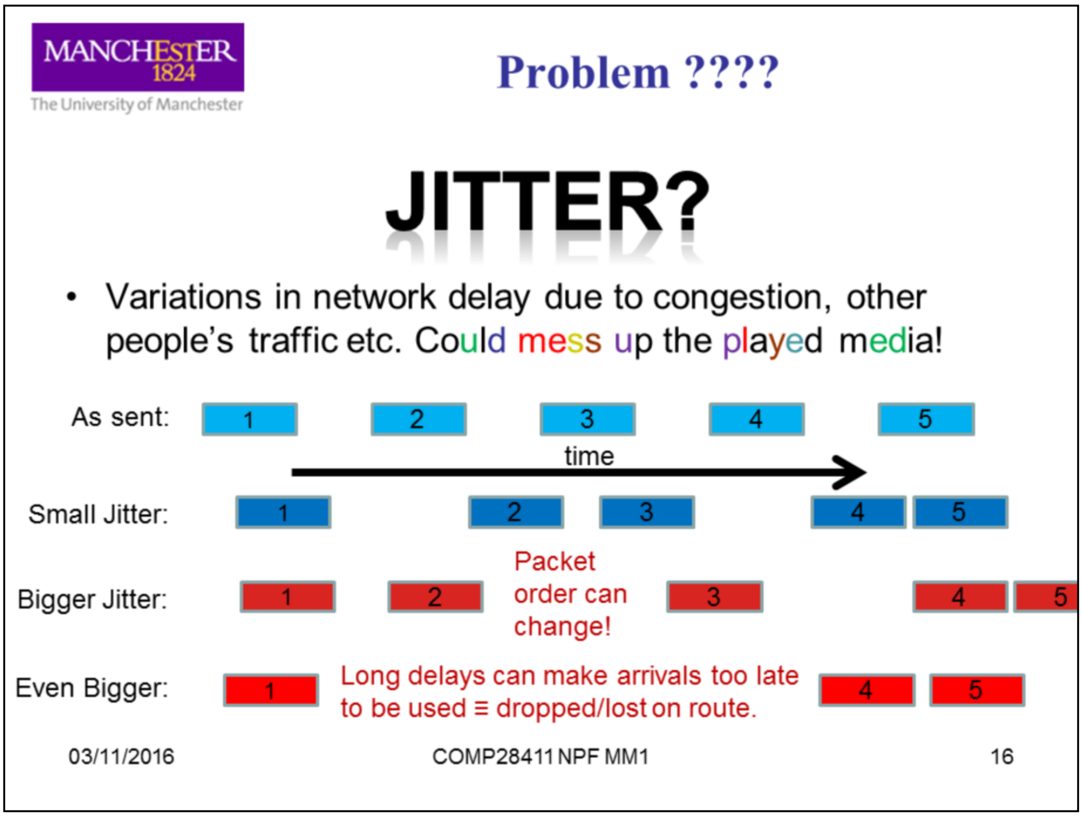
The buffer must be big enough to absorb extra transport delays, bigger than the expected maximum variation in timing. The buffer must also be short enough to provide an OK user experience. For two communication this will be at most a few hundred milliseconds but for one way streaming it may be many seconds or even minutes.




**FIFO** = **F**irst **I**n **F**irst **O**ut = A normal queue.

This should all be obvious, I hope! It is unlikely that the destination can directly change the arrival rate for packets. Some protocols monitor the success/failure rate for packets and try to adjust the source's sending rate. This is similar to the TCP windowing methods except that in order to change the number and rate of sending it is necessary at the application layer to often also change the CODEC which may change the quality of the media, the size and frequency of the packets.

If the network slows down too much then for streaming media the screen goes blank or simply stops + sound stops until enough data has been buffered to restart playback. For live media, instead of waiting for the missing data like for streaming it is often better to leave a gap, throw away late arriving data until the packet stream is back and matched to the real-time + latency of uncongested network or is within real-time constraints for play-back in two way conversation such as the 300-400ms round trip latency humans need in order to converse relatively freely and socially without breaking in unexpectedly to another persons dialogue.




A server sends a packet stream in a fixed order. Each packet may follow a different path through the Internet. Packets may be treated differently by each router along the server to client route. Hence, each packet is subjected to different delays. If the variation in delays (jitter) is large then packets will arrive at the clients out of order.

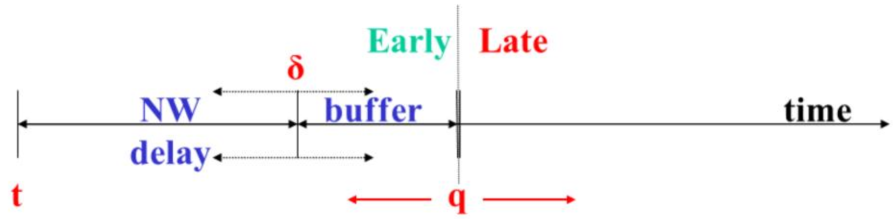
For correct playback, the packets must be rendered in the correct order and with exactly the same per packet time step as that present at the media encoder when the media data stream was captured.

In fact, packets normally contain many different data samples. Often samples are captured and sent at a fixed constant rate resulting in "Constant Bit Rate" (CBR) traffic. However, this can cause problems when networks become, for example, congested. So many media coding methods (CODECS) use "Variable Bit Rate" (VBR) so the amount of information in each packet can be varied. In a congested network sending fewer packets helps to manage and remove the congestion. When the congestion is gone, the bit rate can be increased. Lowering the bit rate means lowering the accuracy of the digital representation of the media. Thus we have lossy and lossless methods of compressing data. Also, different parts of media may require different accuracy. Classical music typically needs much higher bit rates than pop/rock music. Rapidly changing images need higher data rates compared to static ones. Therefore, using VBR can help to mitigate the effects of variations in Internet capacity as well as variations in the reproduction accuracy needed. Humans, often fail to notice momentary or gradual changes in the bit rate. The protocols and technologies take advantage of both our physiology and psychology.

## Fixed Play out Delay - 1

- Receiver attempts to play out each chunk exactly **q** milli-seconds (ms) after chunk was generated.
  - Chunk has time stamp **t**: Play out chunk at **t + q** .

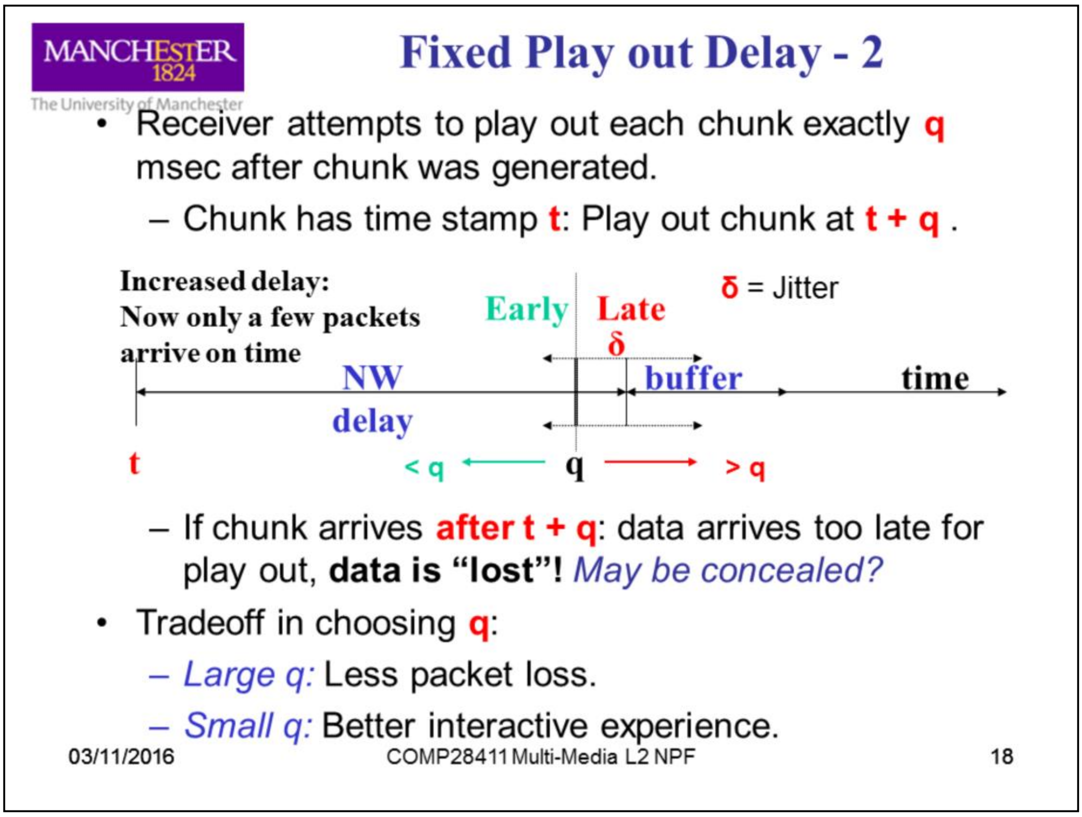



  - If chunk arrives **before t + q**: data arrives early for play out, OK provided buffer is large enough..

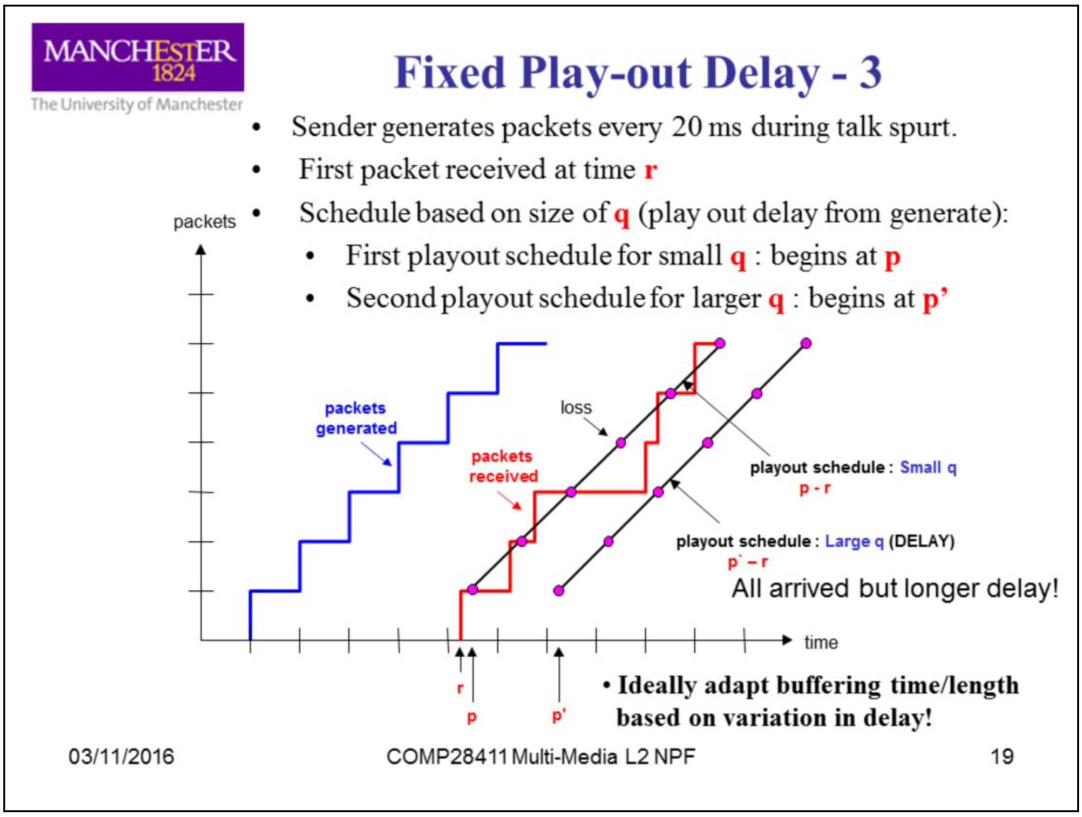
03/11/2016 COMP28411 Multi-Media L2 NPF 17

This illustrates that early arriving data should not be a problem for play back. But, the early data has to be buffered. No machine can buffer infinitely long streams so at some point the buffer is full and then something has to be thrown way. You can choose what to throw away, most Internet applications simply throw away the last arriving item which cannot be inserted into the buffer. Others throw away the item at the front of the buffer or an item somewhere in the middle. Each choice of what to throw away will have different effects on the media? A better solution is to build in a facility whereby the destination can tell the source to stop or delay sending future packets. This can be an explicit command or just statistical information about the state of the buffer and packet arrivals.




Now we have the opposite effect, the data is arriving late. In the example, if the data is just a little late it can probably still be slotted into place and played back such that the user is unaware that there was a problem! If the arrival time exceeds the playback delay then at the point of playback there will be a gap. There are sophisticated Packet Loss Concealment (PLC) techniques which work very well when just a few packets are late. If the network is very congested nothing much may arrive on time and then it is generally better to play silence until the network recovers and the buffer is refilled to a sensible level.




This slide shows what happens when traffic arrives on time (left line of purple dots) to start with and then when it arrives too late (right hand line of purple dots).

For voice a fairly good Codec will try to minimize how much the user notices when one or more packets arrive too late for the normal playback stream. This is called "Packet Loss Concealment" or PLC. There are lots of ways to try and ensure that when not all the packets arrive on time some of the data is still usable. Ideas such as playing nothing for a short time, playing the last played data again, mixing the last and future data (if available) have all been used. Data is quote often interleaved which means the contents of several packets if prepared normally are instead spread across one-another so some of the content is in each packet. Then, if one or more of these packets fails to arrive, the part of the data that did arrive can be used to aid in PLC.

For video, similar ideas can be used for very few losses but beyond a fraction of a second we humans tend to notice something is wrong. One technique that can work very well in fairly static image situations is simply to drop the resolution. Many video systems send separate low and high resolution parts of each image. The low resolution data takes up little space and can be interleaved across other packets.

## Adaptive Play-out Delay-1

A1

- Goal: Minimize playout delay, keeping late loss rate low
- Approach: Adaptive playout delay adjustment:
  - Estimate network delay, adjust playout delay at beginning of each talk spurt.
  - Silent periods compressed and elongated.
  - Chunks still played out every 20 msec during talk spurt.

$t_i$ = timestamp of the $i$th packet

$r_i$ = the time packet $i$ is received by receiver

$p_i$ = the time packet $i$ is played at receiver

$r_i - t_i$ = network delay for $i$th packet

$d_i$ = estimate of average network delay after receiving $i$th packet

Dynamic (moving) estimate of average delay at receiver:

$$\underbrace{d_i = (1-u)d_{i-1}}_{99\%} + \underbrace{u(r_i - t_i)}_{1\%}$$

Where $u$ is a fixed constant (e.g., $u$ = .01).

03/11/2016 COMP28411 Multi-Media L2 NPF 20

Using a fixed size buffer and play back delay would make life simple. However, because the magnitude of delay varies massively during normal Internet usage adjusting the delay and the buffer size is often a better idea as it allows play back to continue in situations where otherwise it may have had to stop or become bitty (staccato). Here we present a simple equation that can be used to dynamically adjust the delay at a destination. This equation uses the average delay summed over several or many packets. As presented it uses a 1% adjustment based on the last arriving packet with 99% of the delay from older packets. Of course by changing the value u or other parts of the scheme other adjustment patterns can be achieved.

Note, however, that this type of scheme will also require some samples to be removed or have their duration reduced if we are trying to shrink the delay or to duplicate some samples or lengthen their duration if we wish to add more delay. Because there is no spare or excess data these operations will always partially distort the play back. A little distortion is no detected by most people but our sensitivity does vary.

## Adaptive play-out delay-2

- Also useful to estimate average deviation of delay, $v_i$:

$$v_i = (1-u)v_{i-1} + u\,|\,r_i - t_i - d_i\,|$$

99% 1% NW delay – est. NW delay

- Estimates $d_i$ , $v_i$ calculated for every received packet (but used only at start of talk spurt
- For first packet in talk spurt, playout time is:

$$p_i = t_i + d_i + Kv_i$$ Timestamp + est. delay + scaled deviation

where K is positive constant

- Remaining packets in talk spurt are played out periodically.
- If a packet is not available ……

03/11/2016 COMP28411 Multi-Media L2 NPF 21

You might like to think about how using other statistics such as standard deviation and its variance can help in delay adjustment.

Also be aware that these schemes all require an intelligent choice of when to start playback.

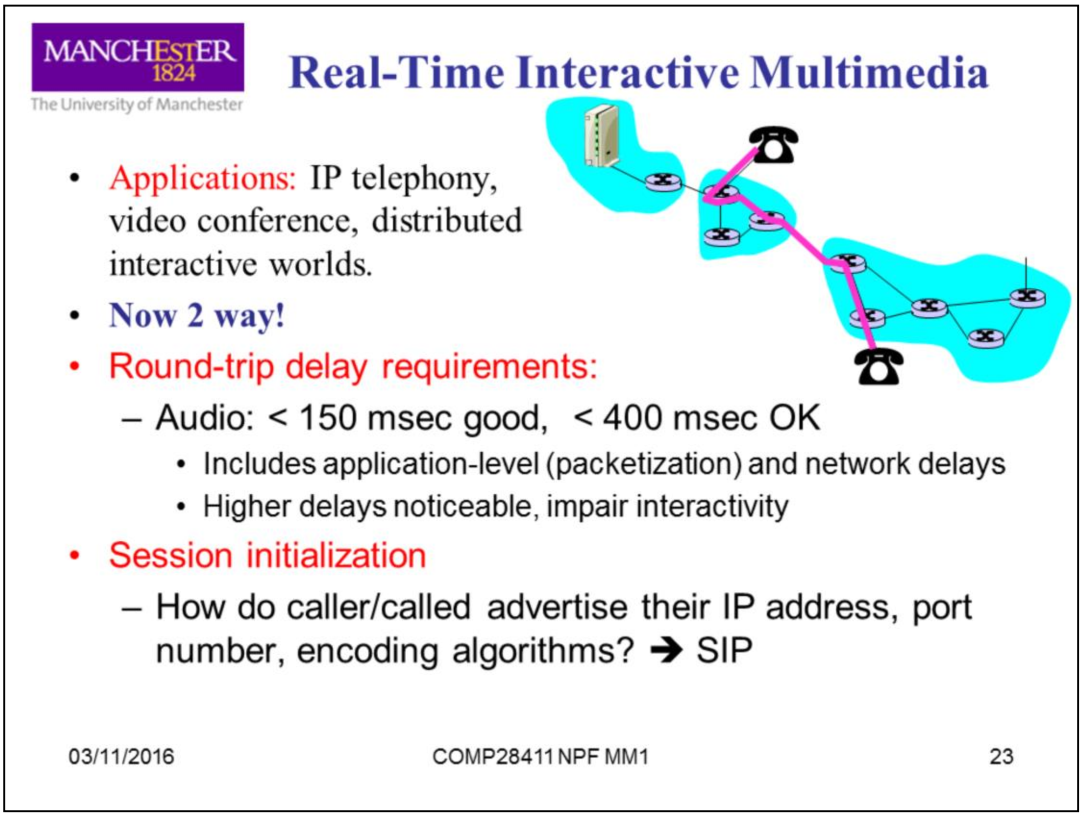
On the Internet, problems tend to happen in clusters or spurts rather than individually. This means that normally, several packets or even groups of streams are effected in similar ways at the same time. Also, the longer the source to destination route is the larger the latency due to the speed of light though whatever medium is being used. Longer delay means there is less scope for adjustment with two way live media. Therefore a VoIP call to someone in the UK or nearby in Europe is much easier to achieve good quality with than a call to the other side of the world. Also, calls routed via terrestrial routes rather than satellite are typically much better. Often the worst case is for data routed via geostationary satellites as these are a long way away 35,786km or 22,236 miles above the equator normally. The data has to go up + down which takes roughly 250ms giving a round trip time of around 500ms or ½ second. We previously said normal conversation breaks down with round trip delays longer than around 300ms so these satellites are not idea. There are satellites in lower orbits used for communications and telephony. These appear to move relative to us so it is necessary to have several satellites not just one and to do handover (similar to mobile phone handover) as satellites come into and then move out of range. Other adjustments must be made to allow for e.g. Doppler shift. Therefore, terrestrial cable or wireless is better. London to Sydney Australia a roughly 13,300km giving a round trip time less than ½ that of a geostationary satellite route which would actually require at least two geostationary satellites with hops between the satellites on different sides of the earth.

MANCHESTER 1824
The University of Manchester

## Problem ????

# TIGHT REAL-TIME TWO-WAY DIALOGUE?

- When talking on the phone we expect a conversation not a monologue!
  - Need to be able to judge when to start talking.
- With video we also want lip and movement synchronization?




The Session Initiation Protocol (SIP) is used to manage the setup and tear-down of multi-media exchange sessions. SIP is part of the IP Multimedia Subsystem (IMS) which is used by Voice Over Long Term Evolution (VOLTE) which is of course part of advanced LTE or the next phase or two of 4G (5G?) mobile telephony. Currently there are lots of SIP services offering VoIP and other media services. The thing to remember about SIP is that it does nothing with actual media, it manages sessions.

SIP users must first register their presence

Start a session using “INVITE from-user to-user Call-id”

Response to INVITE is hopefully “OK from-user to-user call-id”. Call-Id etc. must all match!

Uses DNS to find services, RADIUS etc. for authentication, NAT traversal using STUN etc.,

Clock synchronization using NTP, resolve phone numbers and SIP addresses using ENUM and enhanced DNS, information/configuration exchanged via FTP/HTTP/….

Like telephony and other multi-media  SIP session set-up and the sessions themselves are all real-time. Therefore, driven by clock timers and events. Often use Finite State Machines (FSM) for implementation. But, most network protocols are implemented using FSM.

## *Interactive* Multimedia: Internet Phone Using G711

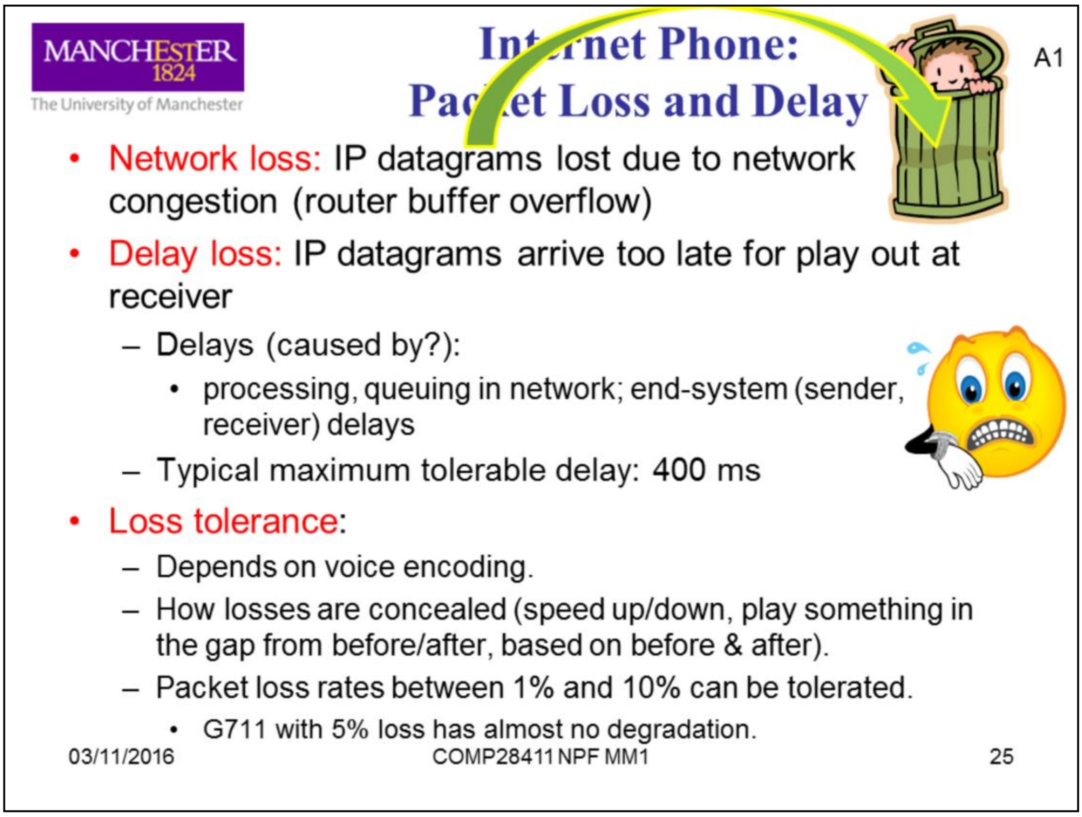
Introduce Internet Phone by way of an example – Already seen this configuration! (last year!)

- Speaker's audio: alternating talk spurts, silent periods.
  - 64 kbps during talk spurt.
  - Packets generated only during talk spurts.
  - Typical, 20 ms chunks at 8 Kbytes/sec: 160 bytes data. Can be longer (e.g. 30ms)/shorter (e.g. 10ms)!
- Application-layer header added to each chunk.
- Chunk + header encapsulated into UDP segment.
- Application sends UDP segment into socket every (e.g.) 20 ms during talk spurt (**VBR**) or continuous (**CBR**). V = **Variable, C = Continuous, BR = Bit Rate**

03/11/2016 COMP28411 NPF MM1 24

G.711 or similar (ILBC Internet Low Bitrate Codec) – seen last year. G711 is an old, fairly simple but still frequently used voice protocol for telephony and other applications. It is used by various Internet, Public Branch Exchange (PBX) as well over the Public Switched Telephone Network (PSTN).

ms = milli second 1000th 0.001




The maximum tolerable delay here is for the round trip (RTT). We'd really like to limit this to 300ms but 400ms is tolerable for a short period. We're dealing with humans here, we are all different in what we will put up with and accept.

For many CODECs it is not so much the data rate that matters but the quality of the PLC. Voice packets are normally very small so making them smaller does not make a lot of difference to network performance as the packet headers will be a large percentage of each packet's size. Our networks are configured for maximum throughput when sending a full MTU (1500 Bytes) at a time. Packets with only a bytes of data can severely effect the overall efficiency of Ethernet and especially WiFi links. A WiFi link may only be able to support 10 or so calls at one time, this number drops if there is competing TCP traffic!

## Summary

- MultiMedia is text, still images, sound and video mixed together.
  - Sound and Video are different – e.g. delay sensitive.
- Send via:
  - Telephone - voice, music, fax….
  - Data network – everything
  - Broadcast TV – Video + Radio + some Data
  - Converged to : Broadband Multi Service networks
- Voice/Music over IP –
  - Buffering,
  - Jitter,
  - Fixed/Adaptive Play-out Delay.

**Next Time**
More on streaming.
Real-Time
Protocols

03/11/2016 COMP28411 NPF MM1 26

## Questions?

- What happens as the end to end delay between server and client continually shrinks?
- What happens as the end to end delay increases?
- Why might it be a bad idea to send live broadcast TV over the Internet?
  - Why do media companies do it anyway?
- What changes in network resources were necessary to allow HD video and sound to be transmitted?
- How is time-slip broadcast implemented e.g. so you can answer the phone and not miss any of your TV show?

03/11/2016 COMP28411 NPF MM1 27

MANCHESTER 1824
The University of Manchester

## Questions? Things to thinks about.

- Is RTP ever used or useful embedded in TCP packets?
- An audio stream samples at 4000Hz. Suggest a suitable timestamp clock increment gap for RTP to use for this audio stream?
- Often RTP SSRC values are chosen randomly. How might a receiver detect and resolve SSRC value collisions? – likely to need a little research!