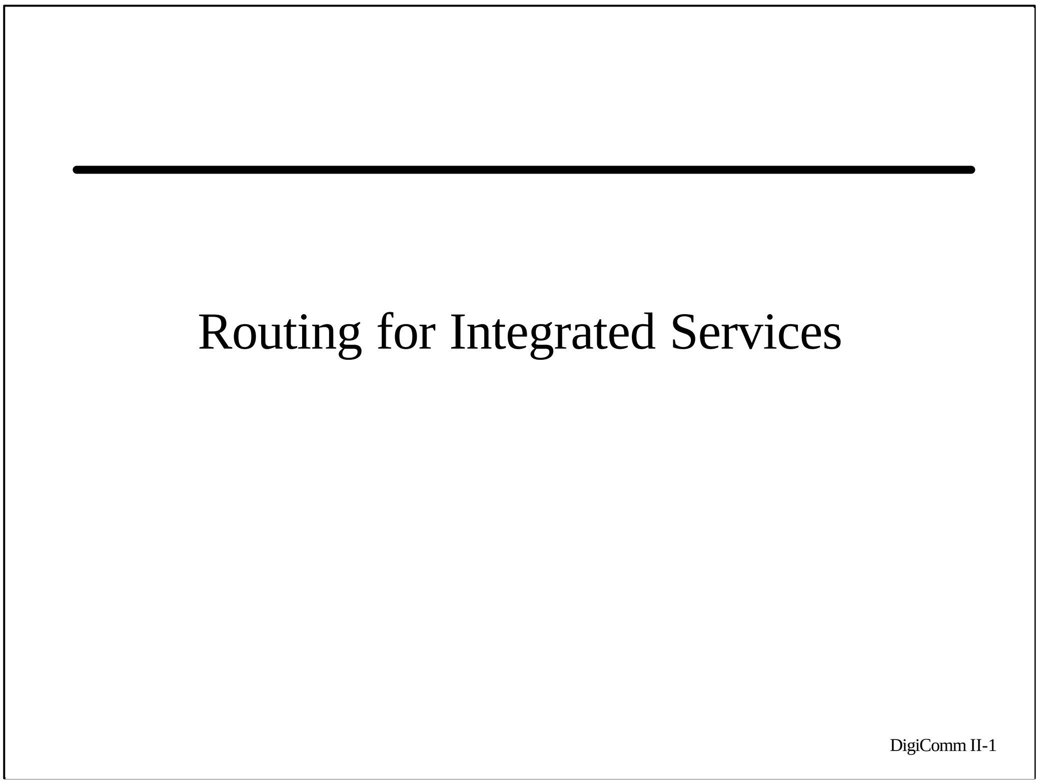
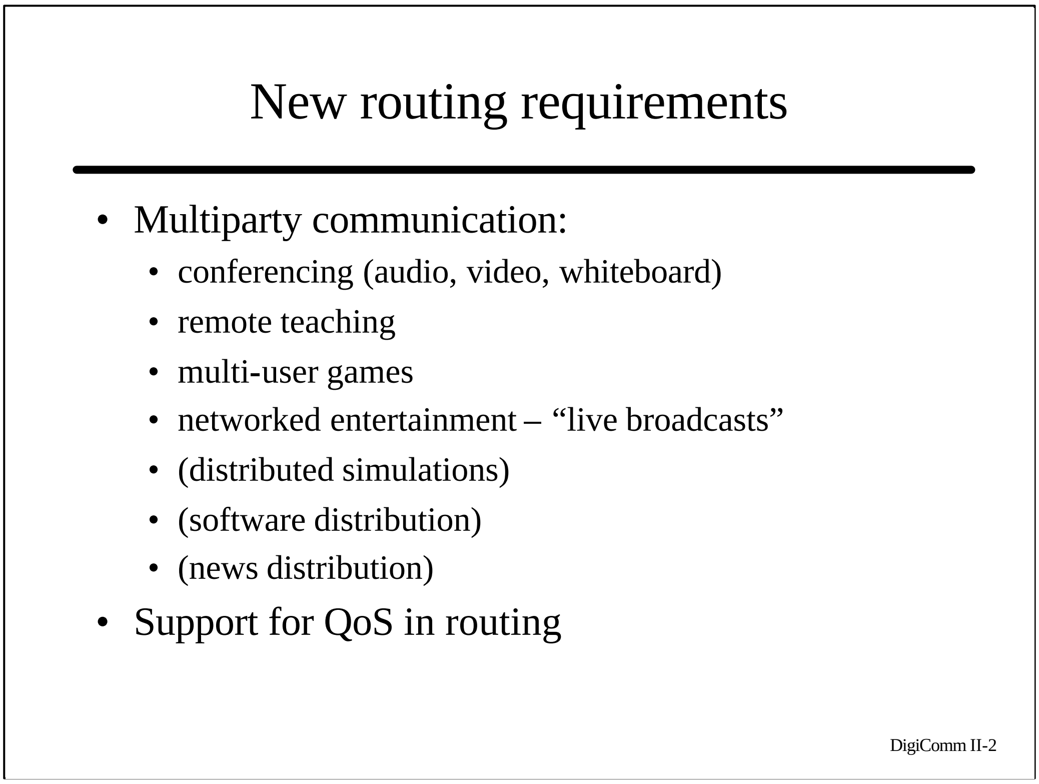


Routing for Integrated Services

DigiComm II-1



New routing requirements

- Multiparty communication:
  - conferencing (audio, video, whiteboard)
  - remote teaching
  - multi-user games
  - networked entertainment – “live broadcasts”
  - (distributed simulations)
  - (software distribution)
  - (news distribution)
- Support for QoS in routing

DigiComm II-2

As we have already discussed, there are a whole new range of applications that will support **Integrated Services** – one network all services. However, in order for Integrated Services to be possible on an IP-based network we need additional support – things that were not specified in the original IPv4 specification.

One aspect of communication that is increasing rapidly is that of multiparty communication. This is the ability to have a communication session that is not just one-to-one, but perhaps one-to-many or many-to-many. Such application including multimedia conferencing, remote teaching and multi-user games. These may demanding have QoS requirements as well as the requirement for many-to-many communication. (Other multi-party communication applications distributed simulation, software distribution and news distribution whose main requirement may be reliable multiparty communication.)

Let us also consider the current mechanisms for routing and forwarding. These are built around the use of destination addresses for building routing tables, and not other constraints are applied. Traditionally, there is only one route between a source and destination. However, what if we would like to perform routing specifying QoS criteria, allowing alternative route selection based on, for example, the requirement for low-end-to-end delay and loss? Traditionally, the use of such QoS constraints are not used generally in constructing routing information.

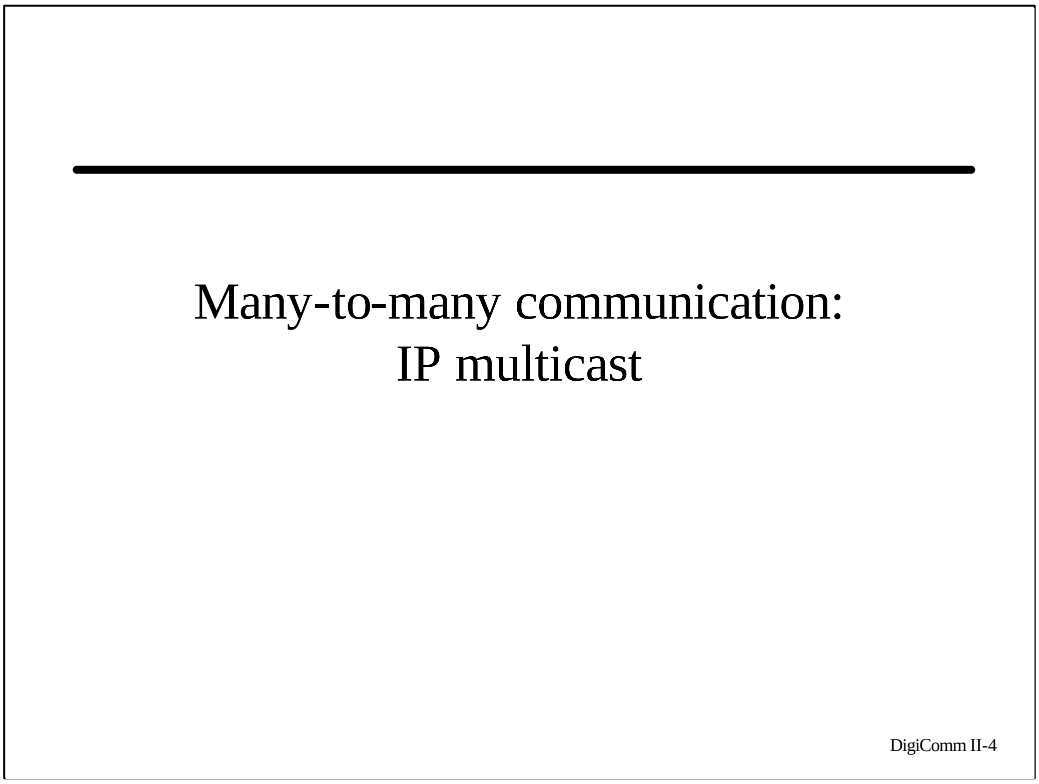
Questions

- How can we support multiparty communication?
- How can we provide QoS support in routing?

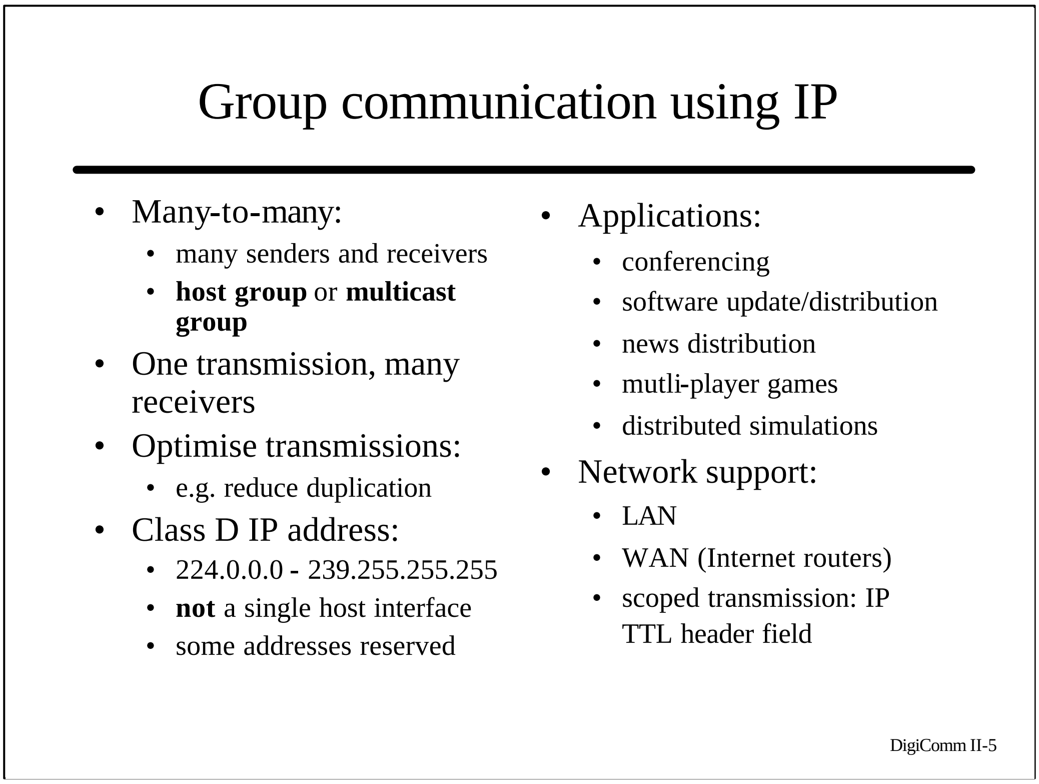
So we would like to answer two questions in this section:

How can we support many-to-many communication? This is not a simple case of having $O(N^2)$ point-to-point unicast connections for our N end-points. Such a nave solution is not practical – it will not scale.

Also, how can we provide QoS-based decision making for constructing and selecting routes? Again, this is not a simple case of adding extra information about QoS parameters to routing updates as we must consider carefully the implications for the operation of the routing algorithms and protocols, especially the intra -domain and inter-domain interactions.



Many-to-many communication: IP multicast



Group communication using IP

- Many-to-many:
  - many senders and receivers
  - **host group** or **multicast group**
- One transmission, many receivers
- Optimise transmissions:
  - e.g. reduce duplication
- Class D IP address:
  - 224.0.0.0 - 239.255.255.255
  - **not** a single host interface
  - some addresses reserved
- Applications:
  - conferencing
  - software update/distribution
  - news distribution
  - mutli-player games
  - distributed simulations
- Network support:
  - LAN
  - WAN (Internet routers)
  - scoped transmission: IP TTL header field

DigiComm II-5

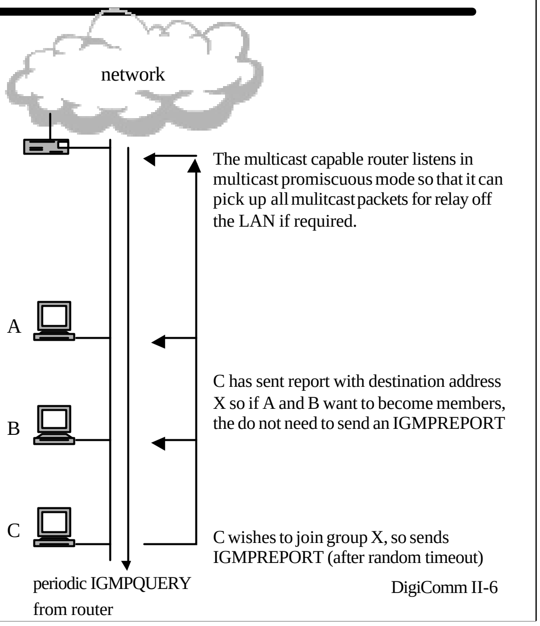
Multicast can be defined, loosely, as the ability to logically connect a group of hosts in a network in order that they perform many-to-many communication. This group of hosts is called a **multicast group** or a **host group**. In a an IP network, multicast is the process whereby *a source host or protocol entity sends a packet to multiple destinations simultaneously using **a single 'transit' operation*** which implies that the packet transit only takes place once from sender to all destinations in the group rather than once for each destination. The connectionless nature of packet switched network means that the packet sender is not necessarily in the multicast group. A packet switched network is said to provide a multicast service if it can deliver a packet to a set of destinations (a multicast group), rather than to just a single destination. Basically, a multicast service can offer many benefits to network applications in terms of reducing the transmission overhead on the sender, reducing the overhead on the network and time taken for all destinations to receive all the information when an application must send the same information to more than one destinations. The key to efficient multicast is to optimise the duplication of the transmitted data in some sense. Normally, this means keeping the duplication of the transmitted information to a minimum.

IP multicast uses Class D IP addresses in the range 224.0.0.0 – l 239.255.255.255. These addresses do not identify a single host interface as unicast IP addresses do, but a group of hosts that may be widely, geographically dispersed. This means that special routing procedures are required in the wide-area to enable multicast connectivity. Some of these are reserved, e.g. 224.0.0.1 is the "all systems" address which all hosts must listen to. To contain the scope of IP multicast packets, the TTL field in the IP header is used to limit the maximum number router hops that a multicast packet can traverse before it should be silently discarded.

Multicast has many benefits over unicast communication in certain areas, e.g. conferencing, software distribution/updates and news distribution. To enable multicast communication, support is needed in the end-systems (hosts and LANs) as well as in the wide-area Internet.

IP multicast and IGMP

- Features of IP multicast:
  - group of hosts
  - Class D address
  - leaf nodes (hosts) and intermediate nodes (routers)
  - dynamic membership, leaf-initiated join
  - non-group member can send to group
  - multicast capable routers
  - local delivery mechanism
- IGMP: group membership control



The multicast capable router listens in multicast promiscuous mode so that it can pick up all mulitcast packets for relay off the LAN if required.

C has sent report with destination address X so if A and B want to become members, the do not need to send an IGMPREPORT

C wishes to join group X, so sends IGMPREPORT (after random timeout)

periodic IGMPQUERY from router

DigiComm II-6

Here we briefly introduce the fundamentals of IP multicast:

• IP multicast allows efficient simultaneous communication between hosts in a logical group called the **host group** or **multicast group**. A host/multicast group which includes a set of zero or more hosts, is identified by a single IP destination address from a specially designated address space.

• The group communication path is modelled as a tree network with the hosts (senders and receivers) within the group located at the **leaf nodes** of the tree, and the intermediate nodes representing distribution/replication points of the communication path.

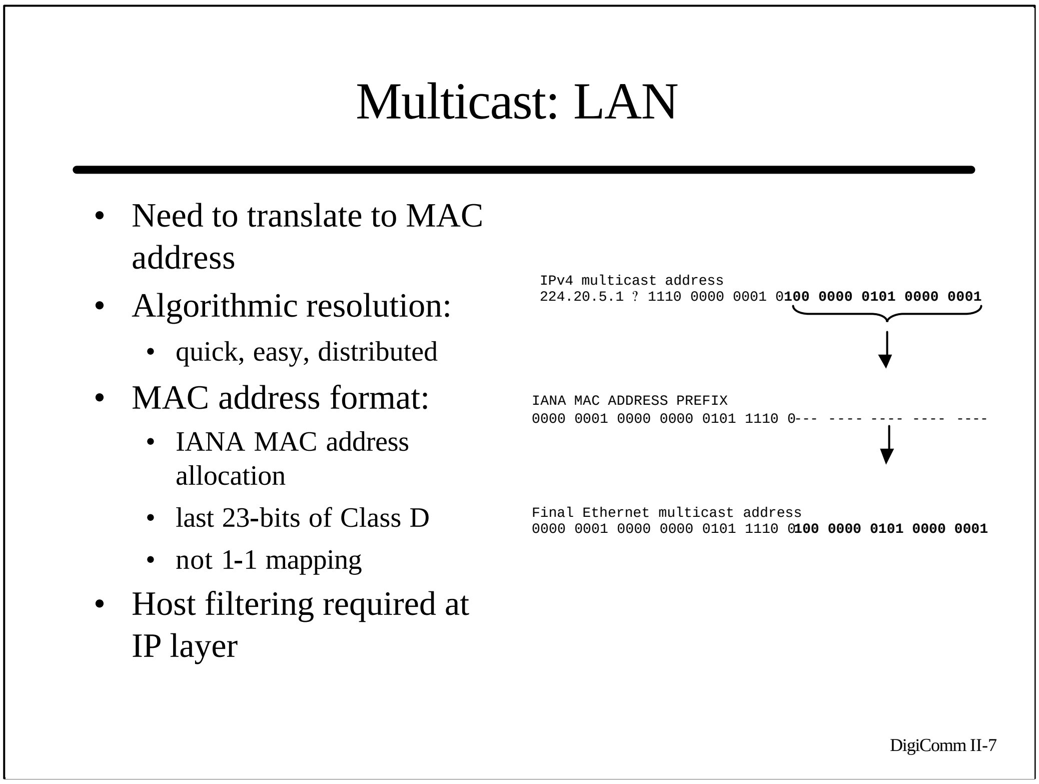
• The membership of a host group is dynamic; i.e., hosts may join and leave groups at any time (leaf initiated join). This is achieved using the Internet Group Management Protocol (IGMP). There are no restrictions on the physical location or the number of members in a multicast group. A host may be a member of more than one multicast group concurrently.

• A host need not be a member of a group to send packets to the multicast group.

• Inter-network IP multicast is supported by multicast routing mechanisms. This means that inter-network forwarding of IP multicast packets is handled by multicast routing mechanisms residing in “multicast capable routers”. The intermediate nodes of the communication path should be multicast capable routers.

• IP multicast relies on the existence of an underlying multicast delivery system to forward data from a sender to all the intended receivers within a sub-network.

IGMP is a very simple protocol with only to messages, IGMPQUERY (sent by a router to see if there are any members of a particular group) and IGMPREPORT (sent by a node to indicate it is leaving or joining a group). Each message refers to a single multicast group, i.e. a single IP multicast address. For Internet-wide connectivity every LAN must have at least one **multicast router** that can listen out for hosts that send group membership reports. If at least one group member exists, then the router should forward multicast packets for that group. To minimise traffic, hosts set random timers and do not send a IGMPREPORT for joining groups until a random timer has expired. IGMP messages are only used in the local area.



Multicast: LAN

- Need to translate to MAC address
- Algorithmic resolution:
  - quick, easy, distributed
- MAC address format:
  - IANA MAC address allocation
  - last 23-bits of Class D
  - not 1-1 mapping
- Host filtering required at IP layer

```
IPv4 multicast address
224.20.5.1 ? 1110 0000 0001 0100 0000 0101 0000 0001

IANA MAC ADDRESS PREFIX
0000 0001 0000 0000 0101 1110 0--- ---- ---- ---- ----

Final Ethernet multicast address
0000 0001 0000 0000 0101 1110 0100 0000 0101 0000 0001
```

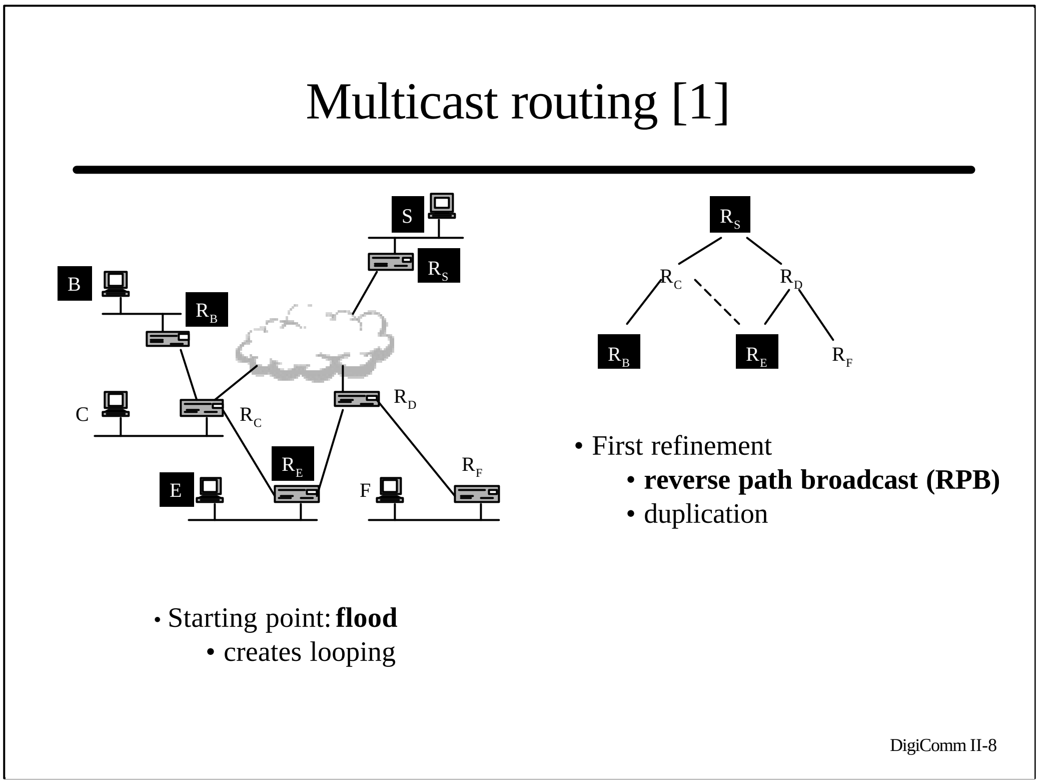
DigiComm II-7

Single LAN multicast is possible without the need for a multicast router. However, LANs do not understand IP addresses they understand MAC addresses. We need address resolution.

MAC multicast addresses cannot be hardwired into LAN adaptor cards in the same way as ordinary MAC addresses. They need to be configured at run-time, i.e. the host must tell its LAN adaptor which multicast MAC addresses to listen for. This must be done the first time a process on the host expresses interest in joining a particular IP multicast group. At this point, the host needs to map the IP multicast group address to a MAC multicast address which it can pass to the adaptor. The mapping must be identical in all hosts and in the router since all participants in the group must end up listening to the same MAC multicast address. This could be done through consultation with a server or, perhaps, a broadcast address resolution protocol could be devised. In fact, the decision made was that the mapping should be algorithmic.

IANA owns a block of Ethernet addresses in the range 00:00:5e:00:00:00 to 00:00:5e:ff:ff:ff and allocates the lower half of these for multicast. The Ethernet convention is that the first byte must be set to 01 to indicate a multicast address. Therefore the range we can use for multicast is 01:00:5e:00:00:00 to 01:00:5e:7f:ff:ff . This means we have 23 bits to play with. These bits are set to the low-order 23 bits of the IP multicast group address to generate the MAC address. So, the address 224.20.5.1, which is e0.14.05.01 in hex, will map to the MAC address 01:00:5e:14:05:01. This is shown in binary below. (We have shown the bit ordering in the conventional way so that 0x01 appears as 00000001. In fact the bits are inserted into the Ethernet frame fields with each byte reversed - so, for example, that the first byte goes out on the wire as 10000000.)

Now, this is obviously not a 1-1 mapping and it is possible that we end up with two IP multicast groups on a LAN mapped to the same MAC multicast address. This is unfortunate, but not disastrous. It means that a host which has joined the group with address 224.20.5.1 will also receive datagrams intended for (say) 224.148.5.1 and will have to filter these out in software. However, many LAN interface cards do not filter multicast traffic efficiently, so this software filtering will need to be present in any case.



Multicast routing [1]

- First refinement
  - **reverse path broadcast (RPB)**
  - duplication

- Starting point: **flood**
  - creates looping

DigiComm II-8

IGMP allows routers to determine which multicast group addresses are of interest in the LAN. We now need a routing mechanism which ensures that all transmissions to a multicast address reaches the correct set of routers and hence the correct set of LANs. Therefore, we need an efficient dynamic multicast routing protocol. This turns out to be a hard problem to crack and is still the subject of much research. In this section we look at the problem and examine some of the protocols which have been developed to date.

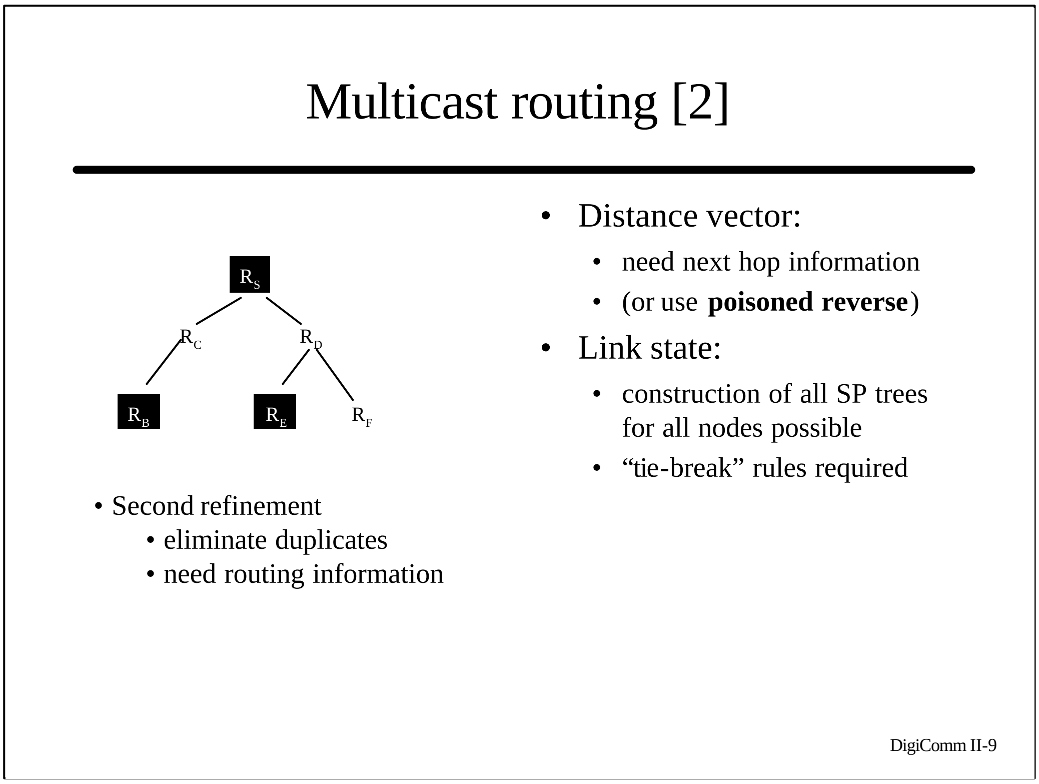
The host S is transmitting to a multicast group address. Hosts B and E have joined the group and have announced the fact to R_B and R_E via IGMP. We need to calculate a spanning tree which interconnects the relevant routers. We can approach a solution through a series of refinements:

Starting point: Flood a multicast datagram to all neighbours except the one which sent it.

The problem with this is that we will get loops; R_C will forward to R_D, R_D to R_E and R_E to $R_{C.}$ One way of solving this problem would be for each router to keep a list of the datagrams it has seen, check this each time it receives a datagram, and delete it if it is in the list. This is clearly not feasible for a multicast which might last several hours and involve millions of datagrams.

First refinement: Reverse Path Broadcasting

It turns out that routers already have quite a lot of the information they need in order to calculate a spanning tree simply from the operation of normal unicast routing protocols. In particular, each node will have a notion of the shortest path from itself to R_S - at the very least, they will know the length of this path and the identity of the first hop on it. This is true irrespective of which unicast routing protocol they are using. We can adopt the following rule - “flood a datagram that comes from the first-hop (on the path back to the source),but delete all others”. Now, when R_C forwards to R_D, R_D will delete the datagram because it did not arrive from its “first-hop to source” (which, for R_D, is R_S itself). This technique is called **reverse path broadcasting (RPB)**.



Multicast routing [2]

- Second refinement
  - eliminate duplicates
  - need routing information

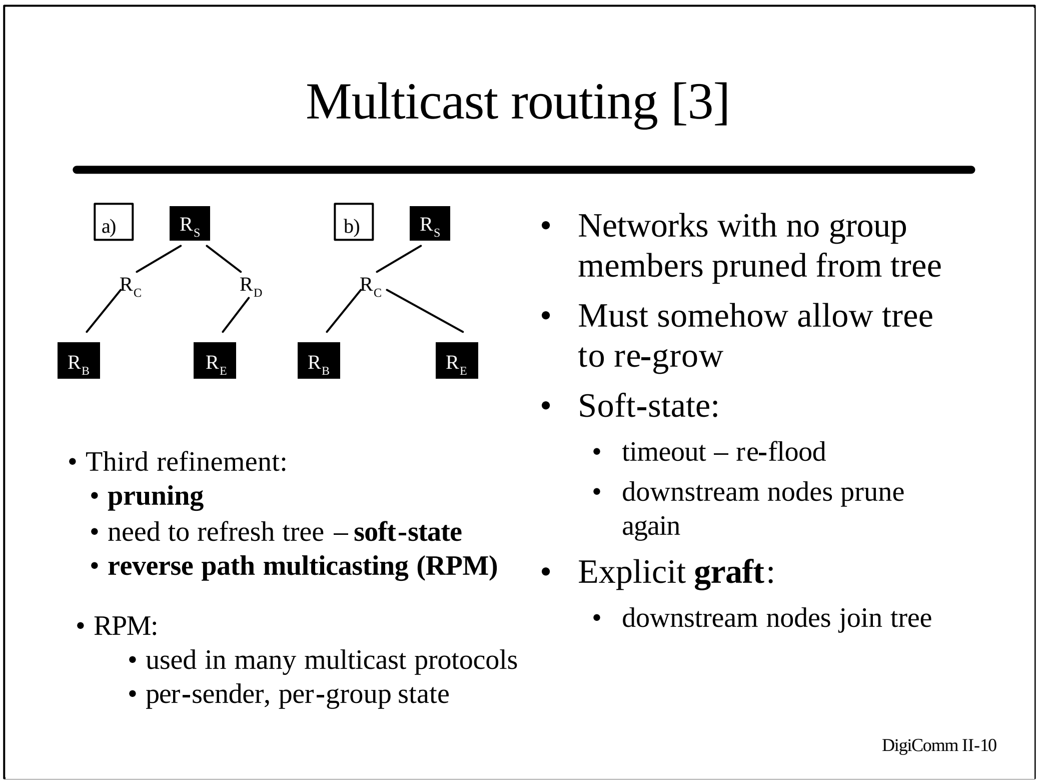
- Distance vector:
  - need next hop information
  - (or use **poisoned reverse**)
- Link state:
  - construction of all SP trees for all nodes possible
  - “tie-break” rules required

Second refinement: Duplicate elimination

As things stand, even with RPB, both R_C and R_D will forward a multicast datagram to R_E. R_E will delete one of these on the basis of the RPB rule. However, we have still wasted effort with a useless transmission to R_E. If R_C and R_D knew that R_E’s path to R_S was via R_D (say) then R_C need not forward to R_E. How can R_C and R_D learn about R_E’s paths? There are two cases to consider:

1) **distance-vector routing:** the distance-vectors R_E sends will contain distances but no indication of first-hop. One possibility is to modify the protocol to include this information. A second possibility is to make use of the **poisoned reverse** rule – send a hop count of “infinity” (i.e. value 16) back to the first hop on the route.

2) **link state routing:** link-state algorithms flood link-state information to all other nodes in the network. By this means, each node ends up with a complete picture of the state of every link in the network. In a unicast link-state algorithm, a node now proceeds to calculate a shortest path tree from itself to every other node in the network. In fact, each node has enough information to calculate shortest path trees for *every* node in the network. All the routers shown can calculate shortest-path trees with R_S as source. If we ensure that they all perform **precisely** the same calculation, they will all end up with the same result. This means that the calculation algorithm has to be formally part of the protocol and needs to specify unambiguous “tie-breaking” rules to select between equal length routes. For example, there are clearly two equal-length routes from R_E back to R_S – we must ensure that all routers make the same choice between them. This can be done, for example, by choosing the router with the numerically higher IP address.



Multicast routing [3]

- Third refinement:
  - **pruning**
  - need to refresh tree – **soft-state**
  - **reverse path multicasting (RPM)**
- RPM:
  - used in many multicast protocols
  - per-sender, per-group state

- Networks with no group members pruned from tree
- Must somehow allow tree to re-grow
- Soft-state:
  - timeout – re-flood
  - downstream nodes prune again
- Explicit **graft**:
  - downstream nodes join tree

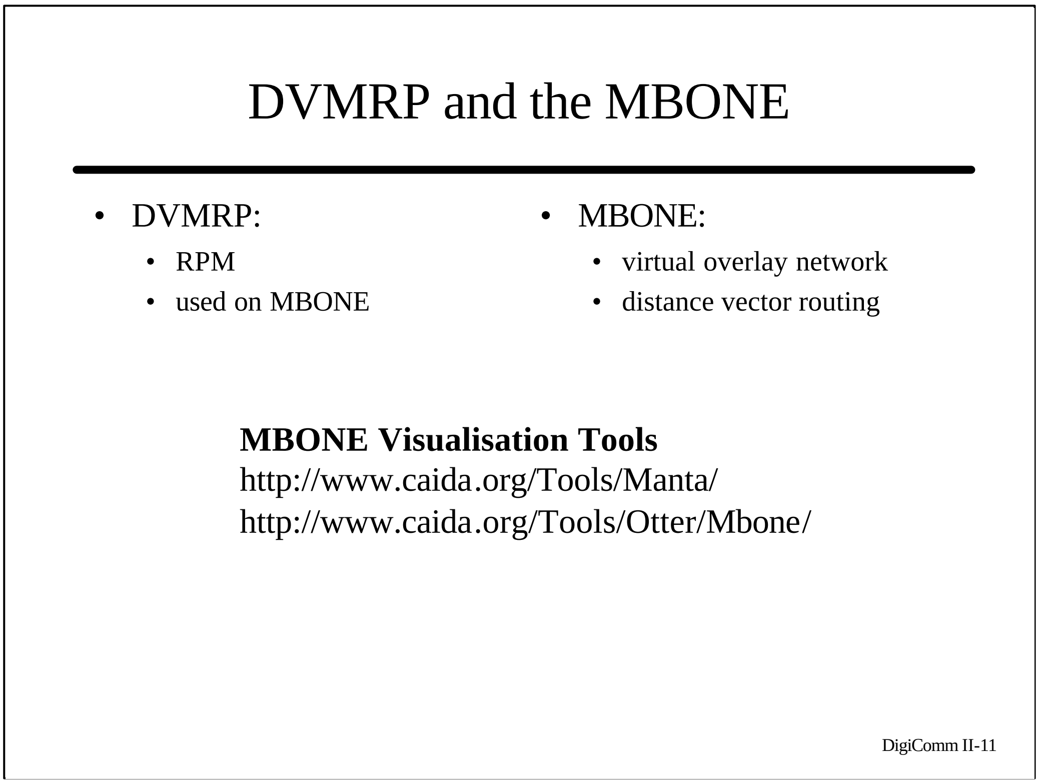
DigiComm II-10

Third refinement: Pruning

By careful application of rules such as those above, it is possible for the routers to agree on a spanning-tree for the whole network. However, we are still wasting effort in forwarding datagrams to R_F when it has no group members. The solution is to introduce special **prune** messages.

When a router such as R_F receives a datagram for a multicast group which has no members on its attached LAN, it sends a prune message back to the router which forwarded the datagram. This router (R_D in this case) now adjusts its routing database to remove R_F from the tree. If we are in the situation of b), R_D will now know it has no-one to forward to, in which case it can, itself, send a prune message to R_S. With the addition of pruning, RPB becomes **reverse path multicasting (RPM)**. We need to have a method of restoring pruned links in case a host the other side of the link joins the group. We can either let prunes time-out (at which point the flow is restored and then, maybe, pruned again) or we can add explicit **graft** messages to the protocol. The former mechanism is a use of **soft-state** which is applied extensively in Internet protocols. Anticipating that state information is perishable in this way and building in mechanisms to restore it is fundamental to the operation of the Internet. It is key concept in making the Internet robust.

By using all these refinements, we can arrive at a reasonably efficient spanning tree. The two possibilities are shown. Both of these use shortest path routes from the source router (R_S) to R_B and R_E. On the face of it, the tree in diagram b) is more efficient since it involves one fewer transmission hop. However, this is not necessarily so since the network cloud might might be a LAN. If it is, then R_S can reach R_B and R_C with one transmission. We may then prefer diagram b) since it shares the forwarding load between the two routers.



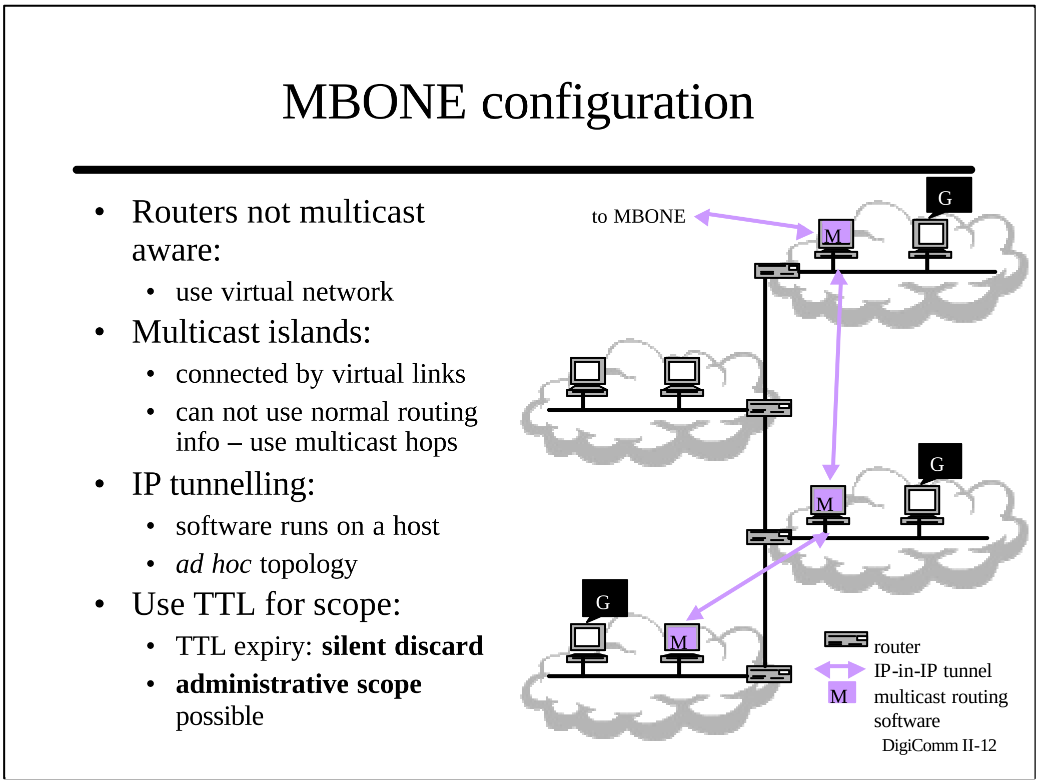
DVMRP and the MBONE

- DVMRP:
  - RPM
  - used on MBONE
- MBONE:
  - virtual overlay network
  - distance vector routing

MBONE Visualisation Tools
http://www.caida.org/Tools/Manta/
http://www.caida.org/Tools/Otter/Mbone/

DigiComm II-11

The Internet's first multicast routing protocol - **Distance Vector Multicast Routing Protocol (DVMRP)** [RFC1075] – is a RPM protocol. It is based on RIP includes all the refinements outlined above, including the poisoned reverse trick. However, it suffers all the well-known problems of distance-vector algorithms and is regarded very much as a simple, interim solution intended to get Internet multicasting off the ground (in which it succeeded mightily). DVMRP has been used extensively in the **MBONE (multicast backbone)**.



MBONE configuration

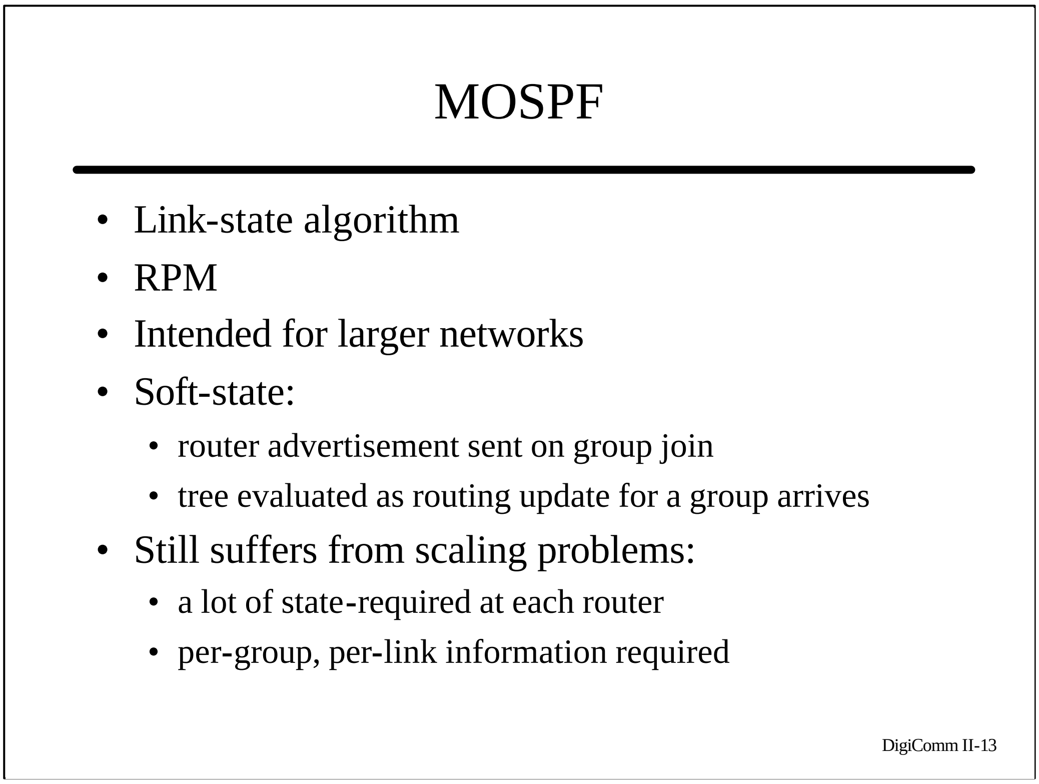
- Routers not multicast aware:
  - use virtual network
- Multicast islands:
  - connected by virtual links
  - can not use normal routing info – use multicast hops
- IP tunnelling:
  - software runs on a host
  - *ad hoc* topology
- Use TTL for scope:
  - TTL expiry: **silent discard**
  - **administrative scope** possible

DigiComm II-12

The MBONE is a multicast network that spans the Internet, but consists of multicast islands connected together. It is a virtual network that is overlaid on the existing Internet unicast infrastructure. This approach was adopted in order to get experience of multicasting at a time when very few Internet routers actually supported it. The links between the multicast routers are **virtual links**. In order to send multicast datagrams along these links, they must be encapsulated within an ordinary (non-multicast) IP datagram with the destination address being the IP address of the multicast router at the end of the virtual link. This is called **IP-in-IP encapsulation** or **IP tunneling** [RFC1853]. This datagram is then forwarded by the normal routers in the ordinary way. On arrival, the multicast router extracts the multicast datagram and routes it according to the multicast group address it contains – it will have to re-encapsulate it in order to send it along the next virtual link. This arrangement is necessary because most "normal" routers do not yet understand multicast group addresses. In practice, the multicast routers are usually instances of the freely available **mrouted** program which runs on Sun workstations. The topology of the MBONE is *ad hoc*. To become part of the MBONE you simply negotiate the establishment of an IP tunnel between your site and a site that is already connected to the MBONE.

Unfortunately, when operating in an overlay network like the MBONE, we cannot use normal RIP distance-vectors directly. Normal RIP distance vectors will refer to the real nodes and links and not to the multicast nodes and virtual links. Therefore, DVMRP has to send its own distance-vectors containing information related to the MBONE itself. The poisoned reverse rule (which is optional in RIP) is used. In typical Internet fashion, DVMRP uses soft-state (explicit prunes) to maintain the tree.

To control the scope of transmission (how far they are transmitted on the network), the **time-to-live (TTL)** in the IPv4 header is used. The TTL is set by the transmitter to indicate how many MBONE router hops this packet should “live” for. When the TTL becomes zero, the packet is subject to **silent discard** – no ICMP TIME EXCEEDED message is generated to avoid packet implosion to the sender. The use of administrative scope by controlling the use of multicast addresses and controlling forwarding policy at multicast routers is also possible.

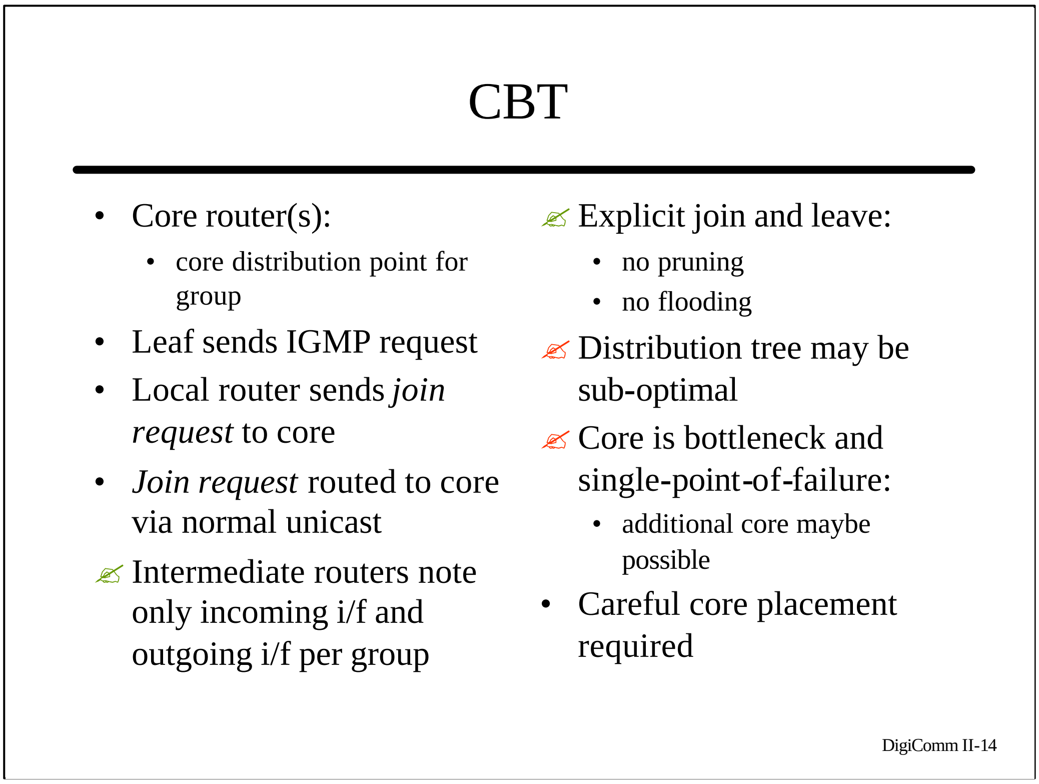


MOSPF

- Link-state algorithm
- RPM
- Intended for larger networks
- Soft-state:
  - router advertisement sent on group join
  - tree evaluated as routing update for a group arrives
- Still suffers from scaling problems:
  - a lot of state-required at each router
  - per-group, per-link information required

DigiComm II-13

A link-state based algorithm called **Multicast Extensions to Open Shortest Path First (MOSPF)** [RFC1584] is also available. MOSPF ends up being quite complex since it has to deal with OSPF's concepts of Areas and Autonomous Systems. It is designed to cope with large networks, however it still has some scaling problems. In larger networks, there could be hundreds of multicast groups in existence at any time. Only a few of these will pass through any particular node. Therefore it makes no sense for each node to pre-calculate trees for every possible source and every possible group. Instead, trees are calculated on the fly when a multicast datagram is received. Like DVMRP, MOSPF uses a soft-state approach, but does not need to use flood-and-prune (as DVMRP does). This is because when a router detects a group join from a leaf node, it send a routing update to the network to let other MOSPF routers know of the new group member. However, this is also MOSPF's short-coming: it needs to send many routing updates and holding routing information on a per-group, per-link basis, resulting in a large database of information. Also, it needs to evaluate the shortest-path algorithm for every source in the group, which is computationally expensive if there are many senders.



CBT

- Core router(s):
  - core distribution point for group
- Leaf sends IGMP request
- Local router sends *join request* to core
- *Join request* routed to core via normal unicast

✍ Intermediate routers note only incoming i/f and outgoing i/f per group

✍ Explicit join and leave:
  - no pruning
  - no flooding

✍ Distribution tree may be sub-optimal

✍ Core is bottleneck and single-point-of-failure:
  - additional core maybe possible

- Careful core placement required

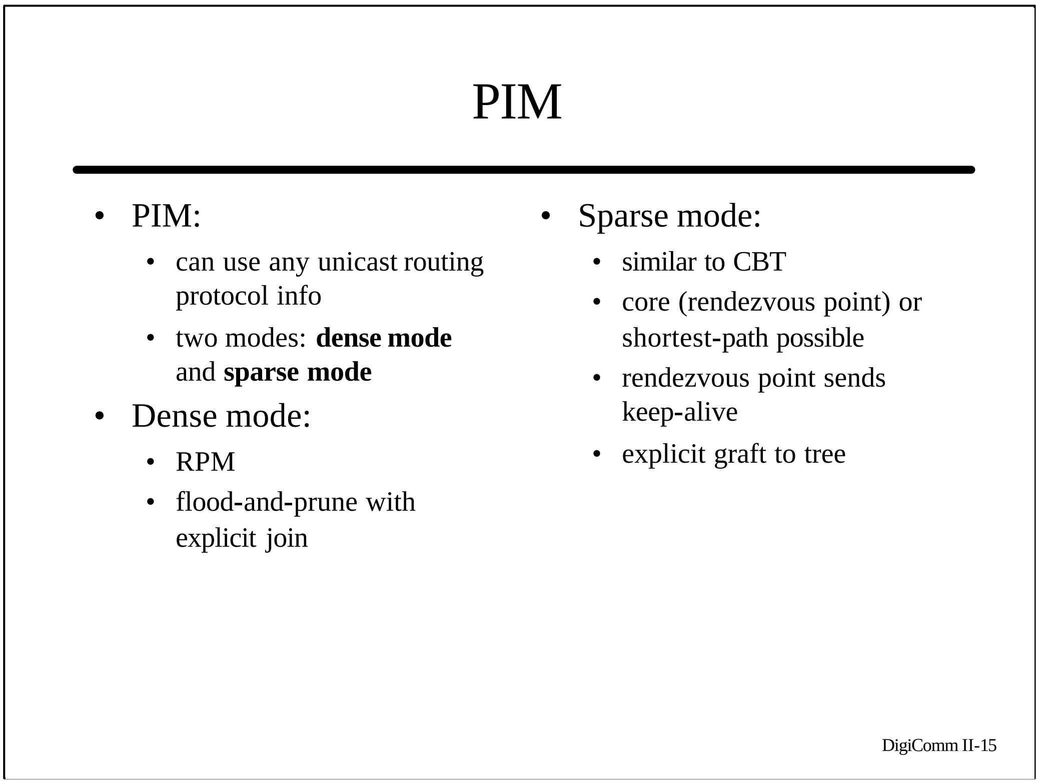
DigiComm II-14

In **Core Based Trees (CBT)** [RFC2201] routers are explicitly designated as **core routers** for the group – in the simplest case, there will be a single core router. When a host wishes to join the group, it informs its local multicast router via IGMP. This router then forwards an explicit join message towards a core router. This is contained in a perfectly ordinary unicast IP datagram and so follows a route which has been established by unicast routing protocols in the normal way. Eventually a single shared tree results; we no longer require routers to be able to calculate different trees for each source as they had to for DVMRP and MOSPF. In fact, the state information retained by the on-tree routers is little more than the identity of the parent and child routers in the tree. Intermediate routers need only to maintain information about which interface a packet came in on, and which interface it was forwarded on. This information need is per group only, so the amount of information is *O*(G) for multicast, as opposed to *O*(G.S) for DVMRP and OSPF (where G is the number of groups and S is the number of senders). Also, join and leave request in CBT are explicit, and so CBT is quite well suited to sparsely populated groups.

The disadvantages with CBT are:

•that a tree may be sub-optimal and is heavily influenced by the location of the core; careful core location may be required

•the core router becomes a single point of failure, though a recovery mechanism is being added

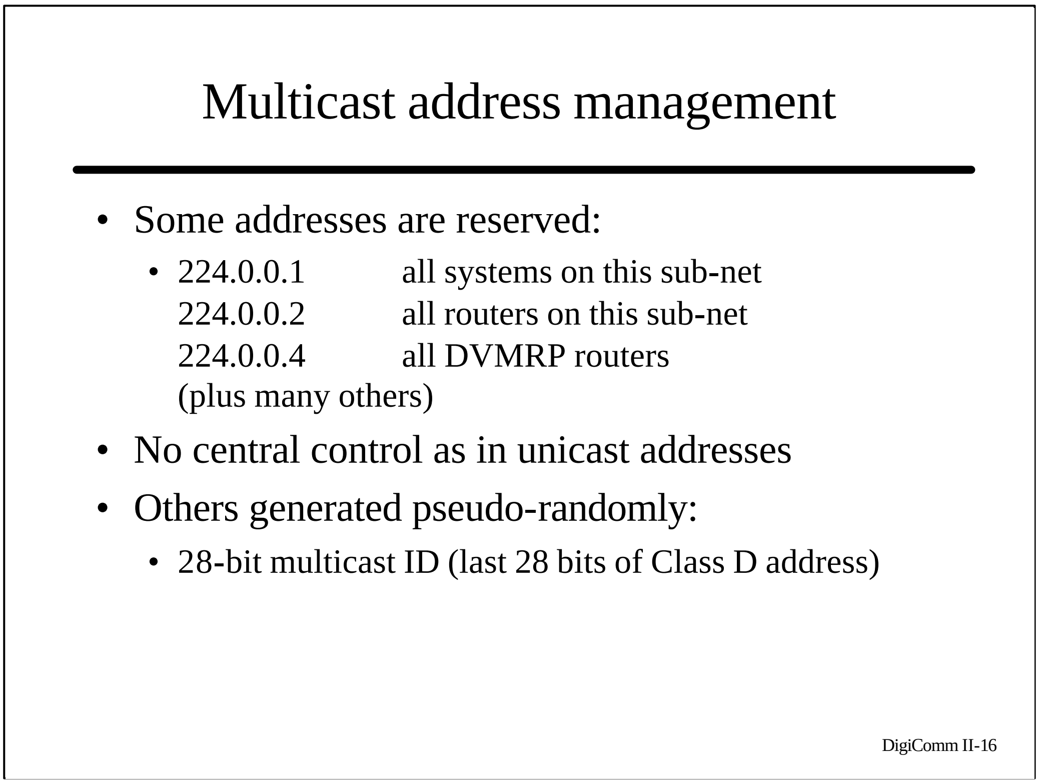


PIM

- PIM:
  - can use any unicast routing protocol info
  - two modes: **dense mode** and **sparse mode**
- Dense mode:
  - RPM
  - flood-and-prune with explicit join
- Sparse mode:
  - similar to CBT
  - core (rendezvous point) or shortest-path possible
  - rendezvous point sends keep-alive
  - explicit graft to tree

DigiComm II-15

An important observation is that some groups are quite **dense** - heavily populated and in a relatively small geographical area. Other groups are **sparse** - lightly populated and spread right around the globe. For dense trees there is a lot of scope for link-sharing and it is worth exchanging state information frequently and expending computational effort to achieve this. For sparse trees there is unlikely to be much link-sharing. This has serious implications for a global Internet in which thousands of multicast groups might exist concurrently. The **Protocol Independent Multicast (PIM)** protocol incorporates these concepts having both dense and sparse modes - in fact it is really two protocols. PIM dense mode is a RPM algorithm. PIM sparse mode [RFC2362] uses an explicit graft mechanism to allow addition to a tree, similar to CBT.



Multicast address management

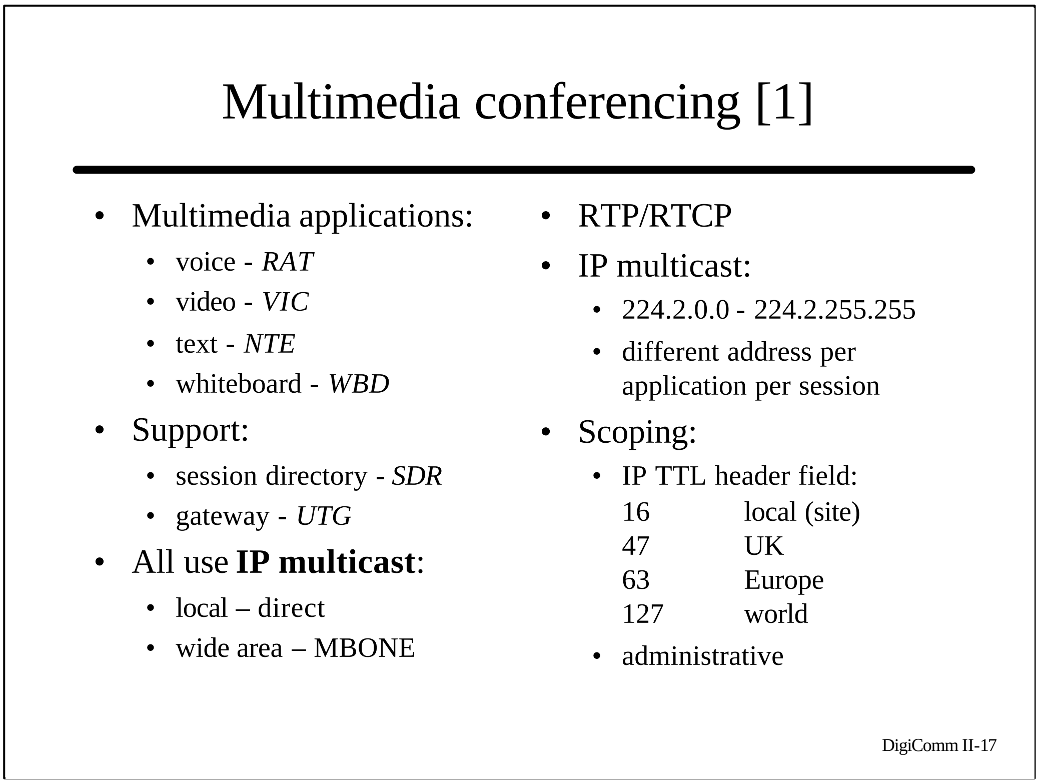
- Some addresses are reserved:

224.0.0.1	all systems on this sub-net
224.0.0.2	all routers on this sub-net
224.0.0.4	all DVMRP routers

  (plus many others)
- No central control as in unicast addresses
- Others generated pseudo-randomly:
  - 28-bit multicast ID (last 28 bits of Class D address)

DigiComm II-16

Unlike the unicast address space in which address allocation is controlled, the multicast address space is (almost) a free-for-all. Some addresses have been reserved and there are certain allocations of ranges of addresses for particular use. However, within these constraints, if a multicast addresses are chosen on an *ad hoc* basis. To help avoid clashes of different addresses, suggestion have been made as to how readily available information (such as time of day, IP address of the host initiating the group, etc.) might be used to produce the last 28 bits – the multicast ID – of a Class D address in a pseudo-random fashion.



Multimedia conferencing [1]

- Multimedia applications:
  - voice - *RAT*
  - video - *VIC*
  - text - *NTE*
  - whiteboard - *WBD*
- Support:
  - session directory - *SDR*
  - gateway - *UTG*
- All use **IP multicast**:
  - local – direct
  - wide area – MBONE
- RTP/RTCP
- IP multicast:
  - 224.2.0.0 - 224.2.255.255
  - different address per application per session
- Scoping:
  - IP TTL header field:

16	local (site)
47	UK
63	Europe
127	world

  - administrative

UCL have been heavily involved with networked multimedia, especially multimedia conferencing. The standards for such applications are still developing. Example applications can be found at:

http://www-mice.cs.ucl.ac.uk/multimedia/

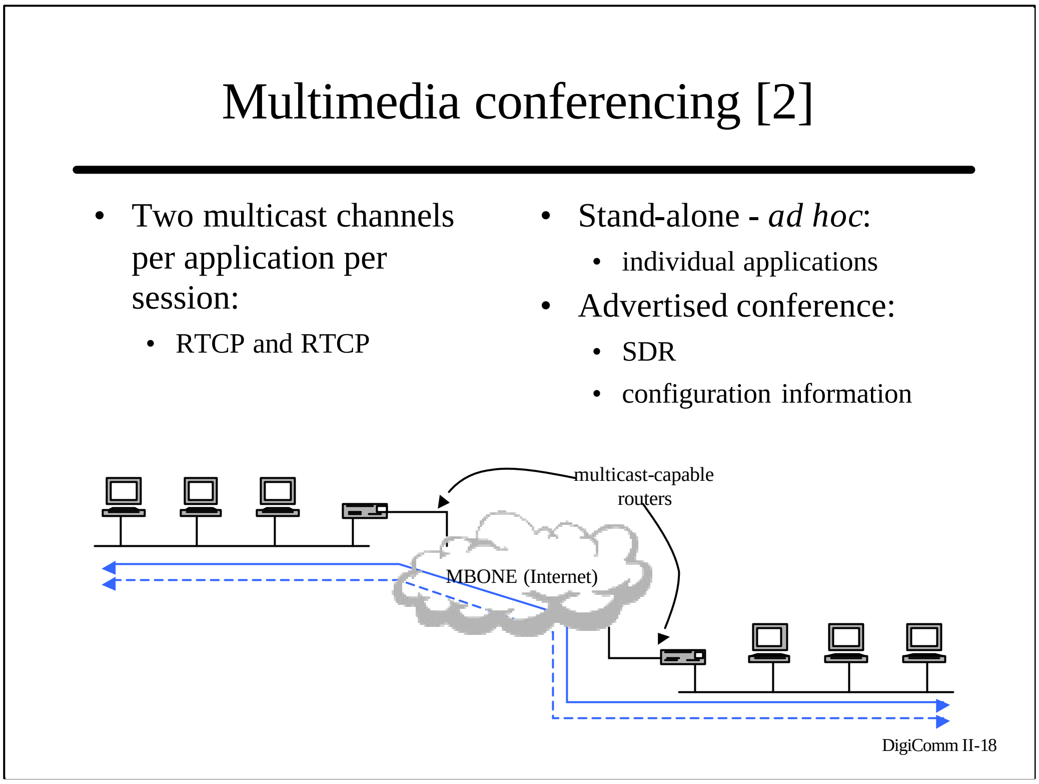
which include an audio tool (*RAT*), a video tool (*VIC*), a text editor (NTE) and a whiteboard (*WBD*). All these applications can run as standalone applications or can be run together within an integrated user interface. All are designed to operate over IP multicast for group communication (on a single LAN or across the MBONE), but unicast (one-to-one) communication is possible. Two additional support applications are a session directory (*SDR*) for a allowing advertisements multicast sessions on the MBONE and a transcoding gateway (*UTG*) for supporting dial-up users and allowing receiver heterogeneity.

All the applications use RTP and RTCP.

When used on the MBONE, the IP multicast addresses used are in the range 224.2.0.0 - 224.2.255.255. These have been designated by IANA for MBONE use by conferencing applications. Each application uses a different multicast address for each multicast **session**.

To restrict the extent of the transmission of the multicast traffic - its scope - the TTL field of the IPv4 header is used. This currently the most common mechanis m used as it is simple to implement but there is a move to adopt a more administratively controlled approach, based on the actual values of multicast addresses being used.

A multicast conference may consist of the use of one or more of the user applications. The support applications may be required for configuration (*SDR*) and supporting LAN users (*UTG*).



Multimedia conferencing [2]

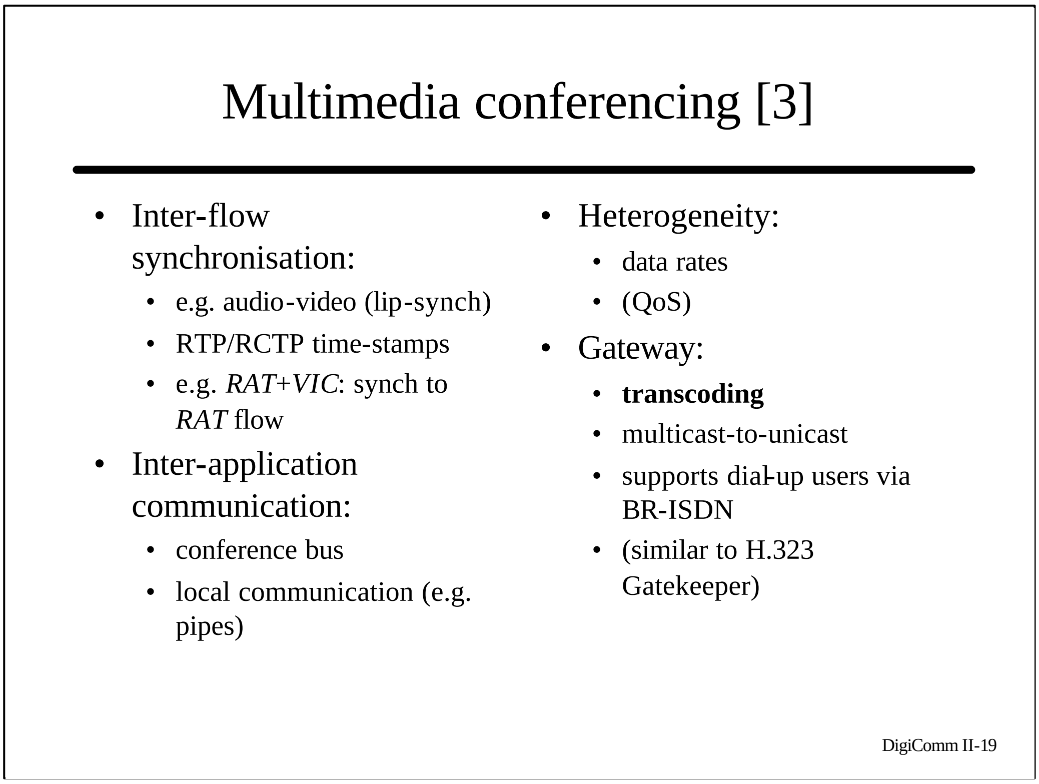
- Two multicast channels per application per session:
  - RTCP and RTCP
- Stand-alone - *ad hoc*:
  - individual applications
- Advertised conference:
  - SDR
  - configuration information

DigiComm II-18

Each application establishes a multicast session. This consists of two logical channels for multicast traffic, one for RTP traffic (the application data) and one for RTCP traffic (signalling and control). These two channels share the same multicast address but have different port numbers. The convention is that a multicast address, *D*, and an even port number greater than 5000, *K*, is chosen by the application user. The session then consists of two channels at *D*/*K* for the RTP traffic and *D*/(*K*+1) for the RTCP traffic.

This configuration is true whether or not the multicast session is to be local or to be sent across the MBONE. If the MBONE is to be used, the LAN requires a multicast capable router to distribute the local traffic and to act as a relay for any traffic from remote group members. The applications default to use local scope but this can be overridden through a command line option or via a configuration menu to change the TTL field as required (unless administrative scoping is being used).

Applications can be started individually as required. However, if the session is to be used on the MBONE, the **Session Directory Rendezvous (*SDR*)**, can be used to advertise the session beforehand, along with configuration parameters. *SDR* listens on some well-known multicast addresses and ports designated for *SDR* to pick up advertisements for other multicast sessions. *SDR* can be seen as the equivalent of a TV guide for the MBONE. When a session is advertised, it may include timing information (when the session is to be executed) as well as information about the media flows to be used. *SDR* can be configured to launch particular applications in order to process certain media types, e.g. *RAT* for audio.



Multimedia conferencing [3]

- Inter-flow synchronisation:
  - e.g. audio-video (lip-synch)
  - RTP/RCTP time-stamps
  - e.g. *RAT+VIC*: synch to *RAT* flow
- Inter-application communication:
  - conference bus
  - local communication (e.g. pipes)
- Heterogeneity:
  - data rates
  - (QoS)
- Gateway:
  - **transcoding**
  - multicast-to-unicast
  - supports dial-up users via BR-ISDN
  - (similar to H.323 Gatekeeper)

DigiComm II-19

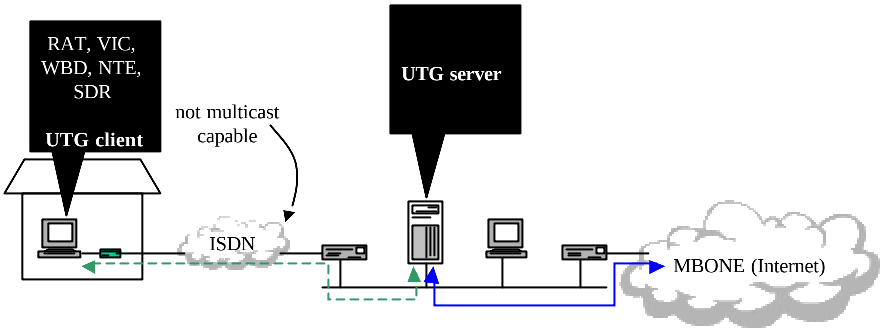
When several applications are used together to process different media flows, there my be a requirement to have inter-flow synchronisation, e.g. to achieve lip-synchronisation between audio and video in a virtual meeting. On the MBONE, as there is no timing signals from the network itself (unlike say, ISDN), the timing information for synchronisation must be built into higher layers. In fact, the timing information is carried in RTP packets and RTCP packets. NTP timestamps give the absolute time, and media-specific timestamps give the intra-flow synchronisation. By comparing the flow-specific timestamp with the NTP timetsamp, it is possible to achieve inter-flow synchronisation. Inter-process communication is required between the application instances on a particular host. This is typically achieved by the use of pipes (for example) and the use of a a well-defined set of message on a **conference bus**. The bus is a mechanism for allowing the transfer of control and configuration information between application instances. It can be seen as a signalling channel.

When many different users exist in a large multicast group, there is likely to be some heterogeneity in the capability of the end-systems and their connectivity. We have also seen that the MBONE leaf-nodes are assumed to be on a LAN. What if the end-user is a dial-up user, with lower data rates than a LAN and no multicast relay? To support such users, **transcoding gateways** can be used to transform the data in multicast flows and redistribute as required. Transcoding the is process of converting a media flow encoding into a different format, e.g. reducing the audio data rate by converting from PCM (64Kb/s) to ADPCM (32Kb/s). A transcoding gateway may perform such flow transformations, as well as act as a relay between a multicast-capable network and users not connected to multicast network, for example users connecting to an office network using BR-ISDN.

(Transcoding and providing relay services between connection-oriented and connectionless networks are two of the functions that are performed by the Gatekeeper function that is described in H.323.)

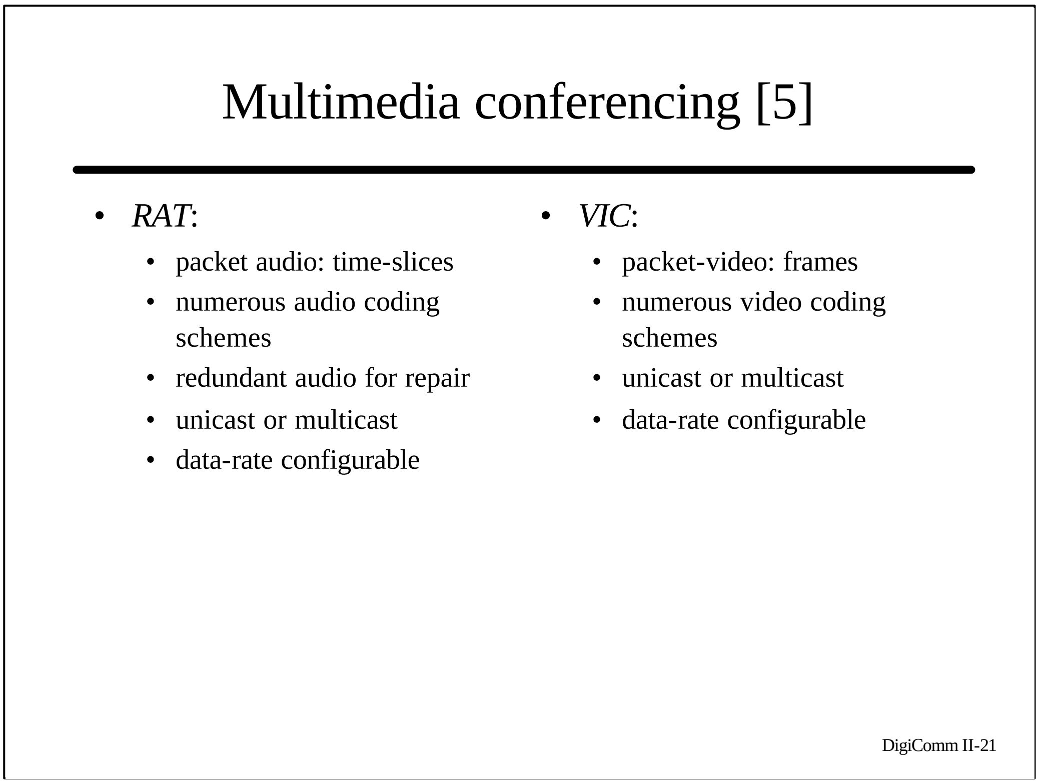
Multimedia conferencing [4]

- *UTG* server:
  - performs transcoding and relay
  - *UTG* clients register with server
- Dial-up users:
  - unicast to UTG client
  - local multicast at remote (client) host



In the UCL toolkit, the transcoding functionality is provided by the **UCL Transcoding Gateway (*UTG*)**. The UTG consists of a client and a server. The server is a central point of contact for users wishing to have a transcoding and relaying service. The user executes the normal MBONE applications locally on their workstation. The workstation must be multicast capable. The user also executes a UTG client process that liases with the UTG server. The client registers with the server and provides information about its capability, e.g. data rate of the link, whether it requires a relay service, which audio and video formats it can support. It can also register which multicast groups the user wishes to join or it can use *SDR* via the *UTG* to dynamically join groups. The *UTG* server then provides the services requested.

For example, consider a dial-up user connecting using BR-ISDN (128Kb/s). This user would like to connect to a conference that will audio and video flows but knows that it will not be able see the full video rate as well as receive good quality audio. The *UTG* client at the remote site registers with the UG server at the main site (which could be, for example, a main office site for a teleworker, or an ISP PoP site). The *UTG* client asks that the *UTG* server provide a 32Kb/s audio flow and a 96Kb/s video flow. (Video flow data-rate reduction can be achieved by reducing the number of colours used, the frame refresh rate, the size of the picture, etc.) The actual multicast conference may be using 64Kb/s audio and 384Kb/s video. The *UTG* server joins the relevant multicast groups, transcodes the data audio flow and video flow, and the sends them to the *UTG* client using IP-in-IP tunnelling. The *UTG* client, on receiving the tunnelled packets, removes the inner multicast packet and redistributes locally.



Multimedia conferencing [5]

- *RAT*:
  - packet audio: time-slices
  - numerous audio coding schemes
  - redundant audio for repair
  - unicast or multicast
  - data-rate configurable
- *VIC*:
  - packet-video: frames
  - numerous video coding schemes
  - unicast or multicast
  - data-rate configurable

DigiComm II-21

RAT and *VIC* are both multicast tools that use RTP to transport audio and video (respectively) across IP networks.

RAT sends time-slices of audio in 20ms, 40ms, 80ms or 160ms chunks (configurable). Larger time-slices are preferable, but packet loss then leaves larger gaps in the audio flow at the receiver. Numerous audio encoding techniques allow use of lower data-rate channels:

linear:	16-bit linear, 128Kb/s
PCM:	?-law companded Pulse Code Modulation, 64Kb/s
DVI:	Digital Video Interactive (Intel), 32Kb/s
GSM:	Global System for Mobile communication, 13.2Kb/s
LPC:	Linear Predictive Coding, 5.8Kb/s

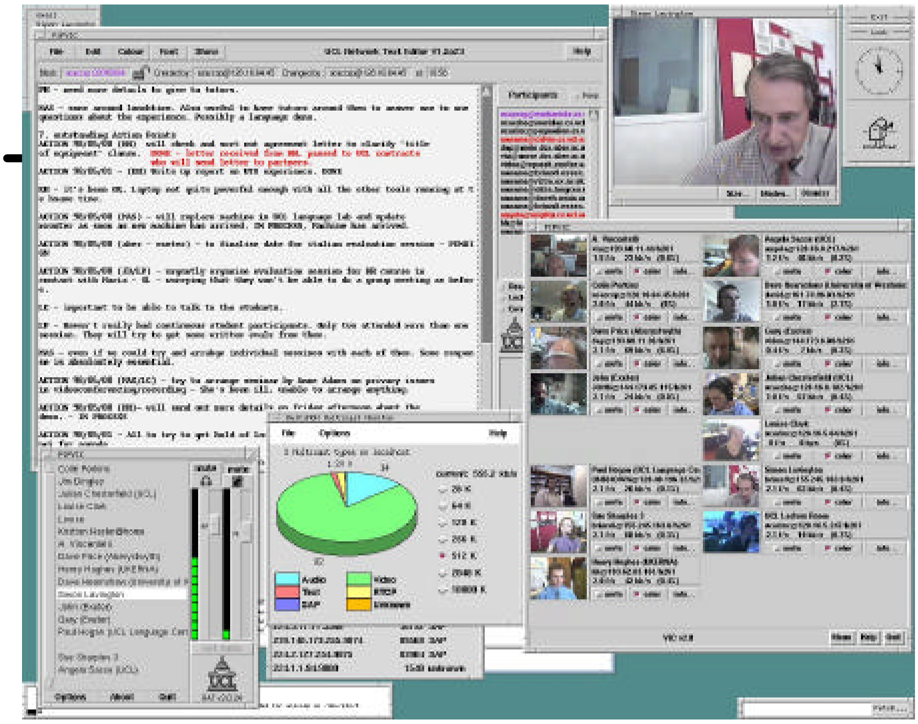
as we go down this list, quality decreases, as does required data-rate, and computational cost increases. All use 8KHz sampling. Typically, linear or PCM is used on the LAN, PCM or DVI over the Internet and GSM or LPC over a modem.

RAT also uses redundant encoding to allow repair of the audio stream to counter packet loss.

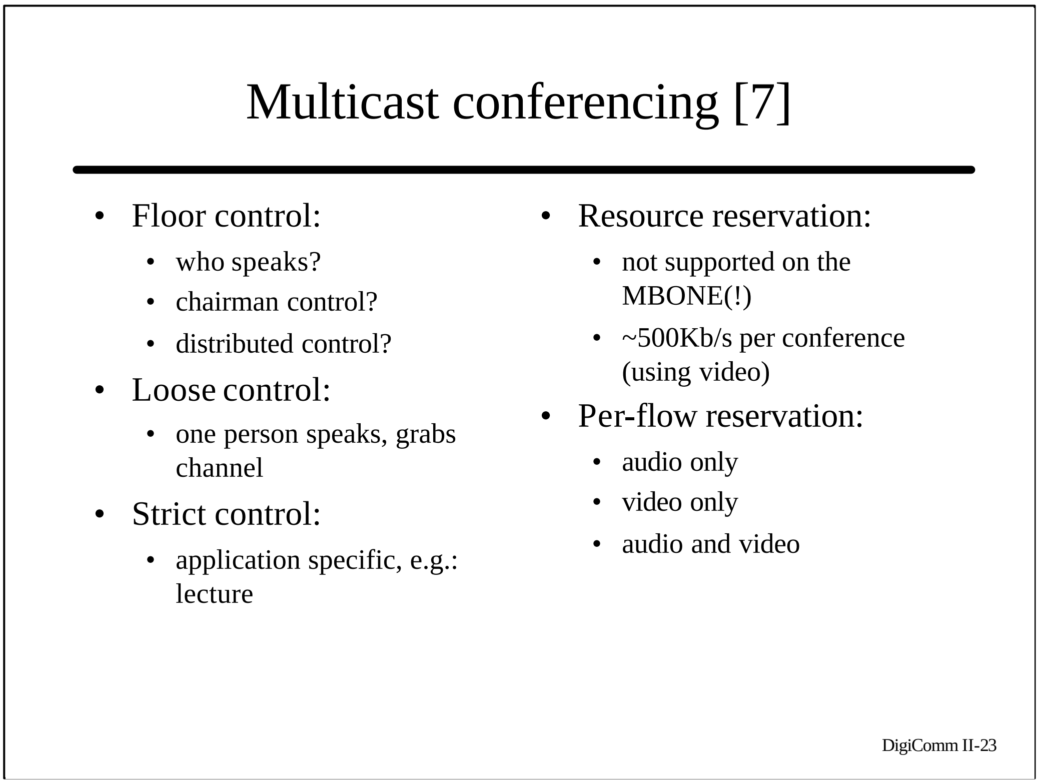
VIC sends single time-slices - single frames (not to be confused with link-level frames) - of video at anywhere between 1 frame per second (fps) and 30 fps (which is suitable for full motion video). It supports the following video encodings at various image sizes:

raw:	24-bit frame-by-frame dumps
JPEG:	motion JPEG
MPEG:	MPEG1
H.261:	intra-frame H.261
H.263:	intra-frame H.263
CellB:	Sun Microsystems proprietary encoding
NV:	Xerox PARC Network Video encoding

The frame rate and overall data rate can be adjusted independently for fine-grained control of the video transmission rate.



DigiComm II-22



Multicast conferencing [7]

- Floor control:
  - who speaks?
  - chairman control?
  - distributed control?
- Loose control:
  - one person speaks, grabs channel
- Strict control:
  - application specific, e.g.: lecture
- Resource reservation:
  - not supported on the MBONE(!)
  - ~500Kb/s per conference (using video)
- Per-flow reservation:
  - audio only
  - video only
  - audio and video

DigiComm II-23

In a conference, discussion or seminar, there is normally an orderly way that humans conduct themselves. This has to be available in multimedia conferencing tools and is called **floor control**. Floor controls requires communication between the humans using the applications as well as some automatic communication between the applications themselves. This latter communication is sometimes also referred to as **application-level signalling**. The floor control models are currently an area of research but two basic concepts exist:

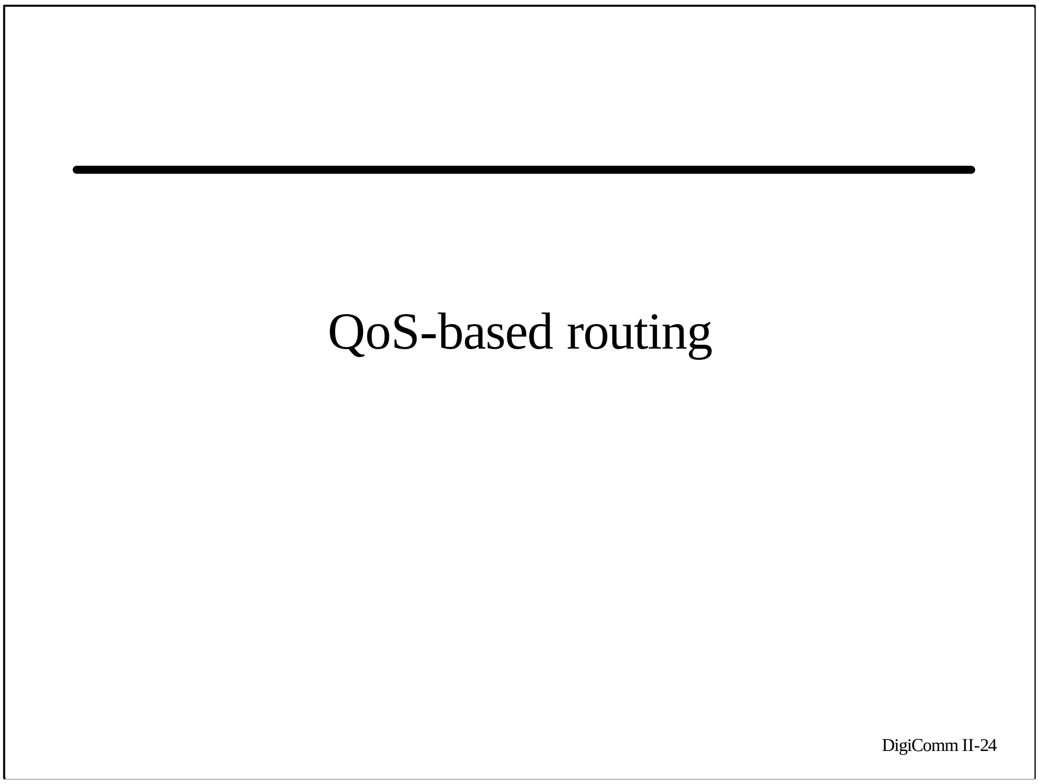
• **loose floor control:** when anyone who speaks grabs the floor. This model is suitable for discussions or *ad hoc* meetings

• **strict floor control:** a chairman has explicit control controls which participants speak. This model is suitable for conferences or lectures.

To enable such control, the applications in the multicast groups must be able communicate. This is enabled through signalling between the applications based on the chose floor control model.

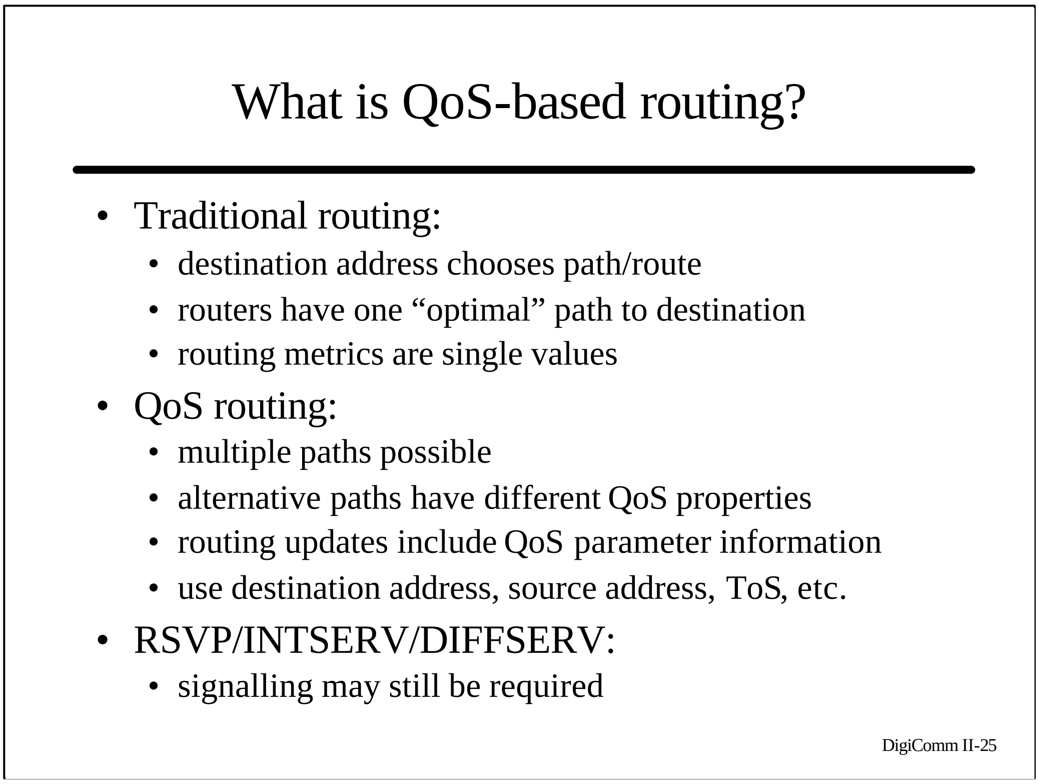
Resource reservation may also be required in a conference in order to allow adequate capacity for audio and video flows. A typical conference with several several tens of participants using audio and head-and-shoulders 8fps video may require around 500Kb/s for operation. The MBONE does not currently support resource reservation, so it may not be possible to have an audio and video conference across the MBONE (remember that the MBINE is an overlay network across the Internet so sees the same QoS as other Internet applications.) Typically, it may be required that some sites in the conference remove the video stream in order to allow continued participation in the conference. Within a LAN environment, if there is a light load on the network then a single-LAN conference is possible without requiring resource reservation (as loss, delay and jitter are likely to be low). Indeed it is often not possible to make resource reservations in a LAN environment based on certain network technology (e.g. Ethernet).

If reservation is used, it could be applied independently to each of the audio video and data flows. For example, human users are fare more intolerant to loss in video flows than audio flows so a reservation could be made for the video flow and the audio (with its relatively modest data rate requirements) could continue operation at “best-effort” service.



QoS-based routing

DigiComm II-24



What is QoS-based routing?

- Traditional routing:
  - destination address chooses path/route
  - routers have one “optimal” path to destination
  - routing metrics are single values
- QoS routing:
  - multiple paths possible
  - alternative paths have different QoS properties
  - routing updates include QoS parameter information
  - use destination address, source address, ToS, etc.
- RSVP/INTSERV/DIFFSERV:
  - signalling may still be required

DigiComm II-25

Traditionally, routing involves routers exchanging information about connectivity/reachability with a single metric to indicate some kind of “cost” that makes sense to the routing table algorithm. This metric may be hop count (e.g. RIP/DVMRP) or link cost (e.g. OSPF/MOSPF). The router uses this single metric to create a single “optimal” path to a destination. The path is optimal with respect to the single metric being used. Other, sub-optimal paths may exist, but they are not used.

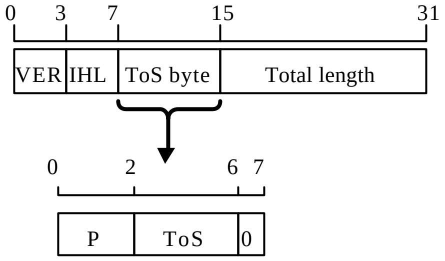
With **QoS-based routing** (also called **constraint-based routing**), multiple paths are possible between sender and destination, and the choice of which path is followed is based on policy criteria selected by looking at packet header information such as source address, the ToS/DIFFSERV byte, etc. This requires that the router hold information about multiple paths per destination, running its routing algorithm multiple times to set up this information, and to include various QoS-related metrics in its routing updates. This is a non-trivial change to the operation of the router and the network as a whole.

A good overview of the issues in QoS-based routing is presented in [RFC2386].

Note that the aim of QoS routing is to indicate that paths with suitable QoS characteristics are available, but other mechanisms (such as RSVP and/or INTSERV and/or DIFFSERV) may still be required in order to ensure that resources along that path remain for the duration of the flow.

IPv4 ToS byte

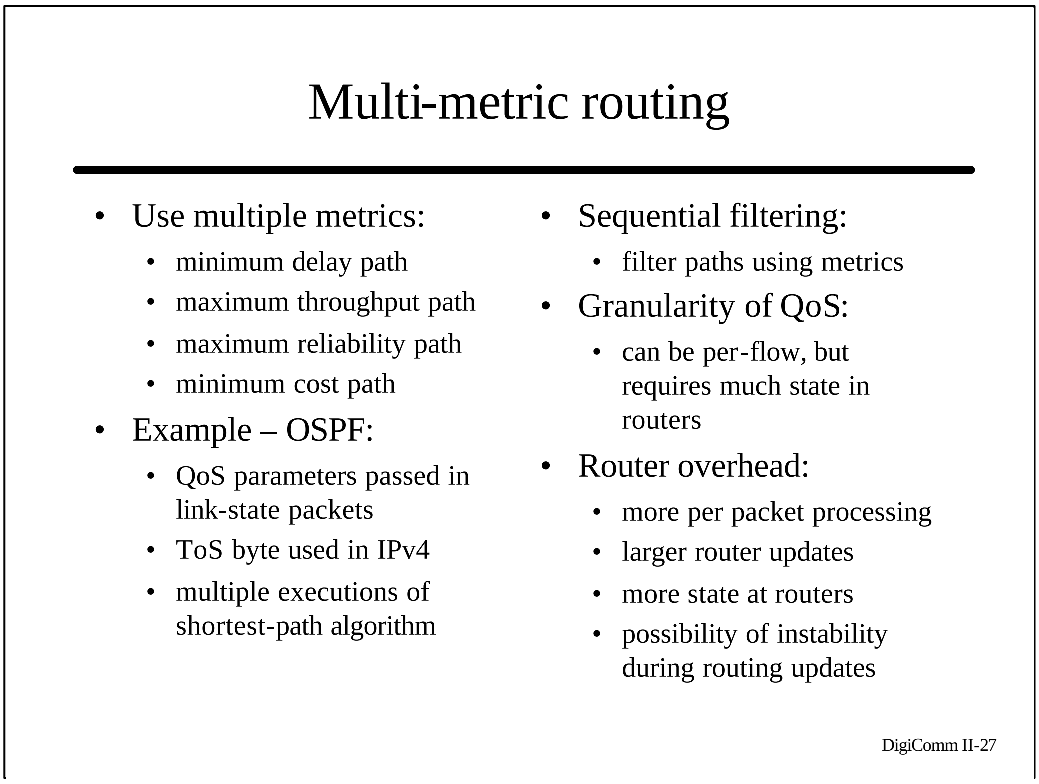
- IPv4 header – ToS byte:
  - 3-bit precedence, P
  - 4-bit ToS
- Precedence:
  - 000: lowest
  - 111: highest
- ToS – flags:
  - 1xxx: minimise delay
  - x1xx: maximise throughput
  - xx1x: maximise reliability
  - xxx1: minimise cost (£)
  - 0000: “normal” service



- Not widely used:
  - no global agreement
  - (some use in Intranets)
- RFC1349 – now historic:
  - superseded by DIFFSERV
  - not compatible with ECN

In [RFC1349] is documented as way of using the 8-bit **Type of Service (ToS)** byte in the IPv4 header to provide a class of service indicator. The byte is plit into two fields, a precedence indicator, P, and a set of flags indicating the type of service (ToS) required for the packet. P takes values from 0 – 7, with 0 being the lowest precedence and 7 being the highest. The ToS flags indicate whether the packet requires minimum delay, maximum throughput, maximum reliability (low loss) or minimum (monetary) cost. The terms “maximum” and “minimum” are not that well defined.

This system was not widely use across the Internet, but found its way into use in some intra-domain (intra-AS) routing mechanisms. Although [RFC1349] is now historic (superseded by the DIFFSERV work), it serves to illustrate how we might perform QoS routing by indicating, in a packet, some simple handling requirements.



Multi-metric routing

- Use multiple metrics:
  - minimum delay path
  - maximum throughput path
  - maximum reliability path
  - minimum cost path
- Example – OSPF:
  - QoS parameters passed in link-state packets
  - ToS byte used in IPv4
  - multiple executions of shortest-path algorithm
- Sequential filtering:
  - filter paths using metrics
- Granularity of QoS:
  - can be per-flow, but requires much state in routers
- Router overhead:
  - more per packet processing
  - larger router updates
  - more state at routers
  - possibility of instability during routing updates

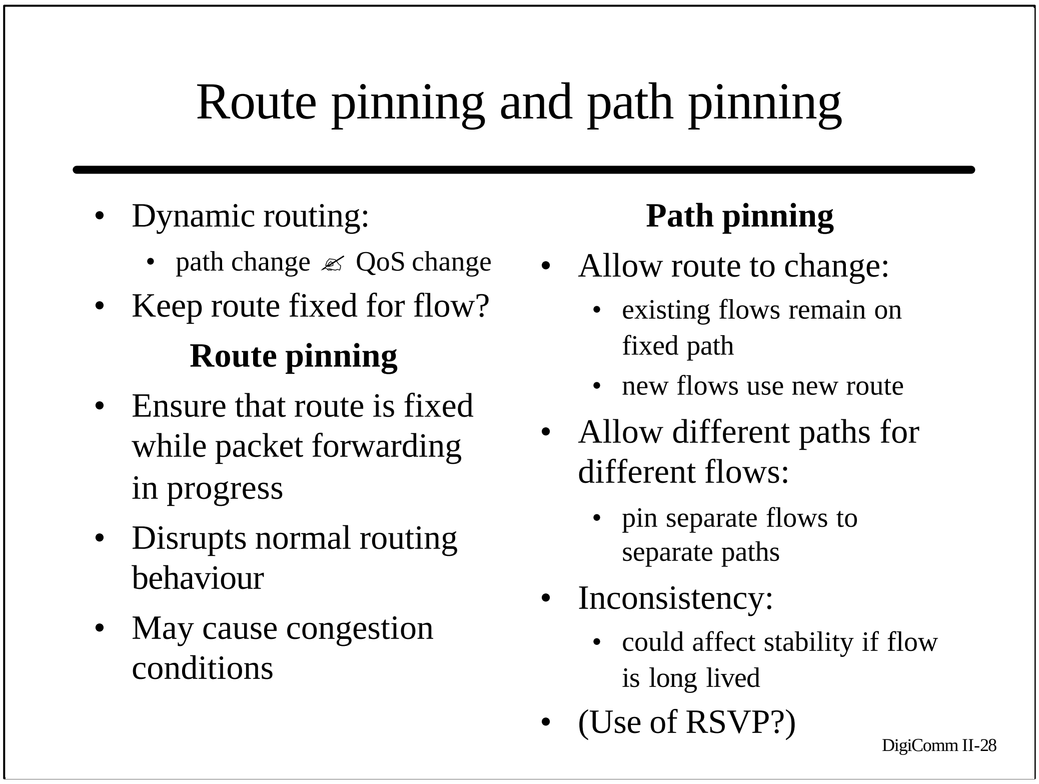
DigiComm II-27

Multiple metrics can be used to establish multiple paths based on QoS parameter criteria. For example, OSPF [RFC2328] allows the use of delay, throughput loss and cost information to establish routes. Information about these parameters is included in link-state packets emitted by OSPF routers. When routing tables are evaluated, the the SP algorithm is run multiple times, once for each metric and the resulting routes are stored. When a packet arrives with a ToS marking, say, for “maximum reliability” in its ToS markings, the router makes a path selection based on the routing table evaluated using the loss/reliability information.

In general, where multiple selection criteria are specified, sequential filtering can be used to select a path. For example, if “high throughput” and “low delay” are selected, initially some candidate paths are selected by applying the “high throughput” criteria only. Then, these candidate paths are filtered based on the “low delay” criteria so selecting the path(s) with both “high throughput” and “low delay”. This allows flexibility but requires extra processing, compared to using a single metric to describe/summarise both “high throughput” and “low delay”. The added processing could increase the latency of transmission, at least for the first packet in a flow, before the selected path is cached to the routers forwarding table.

The granularity of such an approach is generally kept quite coarse in order to keep processing overhead low. It could be possible to define polices that select packets based on header information down to a per-flow level, but this would introduce a large amount of extra processing and storage of state at the routers.

General disadvantages of multi-metric routing are that there is an increased overhead on the router, in terms of per-packet processing, generating and processing router updates, holding state for paths. There is also the possibility on instability and routing loops during updates, or if inconsistent implementation of routing policy causing conflicts in routing behaviour, e.g. routers in the same domain find they have different routing tables even though they have seen the same routing updates.



Route pinning and path pinning

- Dynamic routing:
  - path change ✍ QoS change
- Keep route fixed for flow?

Route pinning

- Ensure that route is fixed while packet forwarding in progress
- Disrupts normal routing behaviour
- May cause congestion conditions

Path pinning

- Allow route to change:
  - existing flows remain on fixed path
  - new flows use new route
- Allow different paths for different flows:
  - pin separate flows to separate paths
- Inconsistency:
  - could affect stability if flow is long lived
- (Use of RSVP?)

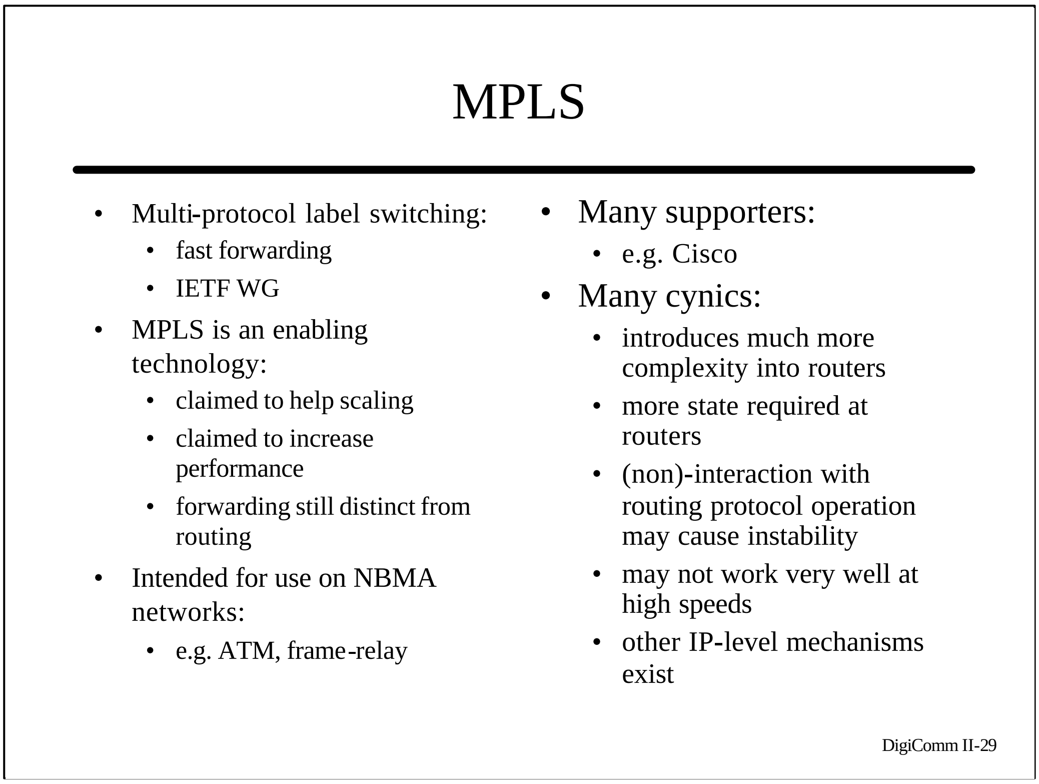
DigiComm II-28

We have already noted that changes in QoS for a flow can occur due to the changes in the network path being followed by the packets in the flow. This is a natural consequence of the dynamic routing changes that give IP its robustness. However, routing changes can often occur when a the existing route is still serviceable, but just not "optimal". Remember that, traditionally, routers only compute one optimal route based on the routing metric. This itself could cause instability as much traffic may be re-routed, but also this will normally result in an observable QoS change. Holding a routes constant – **pinning routes** – for the duration of a flow (e.g. based on some caching/time-out criteria) might help to alleviate this. However, this could disrupt the network stability, as routers with active flows may not change their routing tables, whilst other routers in the domain do, and routing loops and congestion effects could result.

An alternative is to use **path pinning**, allowing the routing table to be updated as normal but keeping knowledge of the current path for exiting flows. So, any existing flows continue to use the same path but new flows would use a different path. This could still lead to instability and consistency in the network if there are many long-lived flows that hold paths pinned for a long time.

Another proposal is to use RSVP to signal path pinning for some flows, and where paths really do have to change, to try and use RSVP to establish provide some likeness of QoS one the new path as was present on the old path.

These are still research areas.



MPLS

- Multi-protocol label switching:
  - fast forwarding
  - IETF WG
- MPLS is an enabling technology:
  - claimed to help scaling
  - claimed to increase performance
  - forwarding still distinct from routing
- Intended for use on NBMA networks:
  - e.g. ATM, frame-relay
- Many supporters:
  - e.g. Cisco
- Many cynics:
  - introduces much more complexity into routers
  - more state required at routers
  - (non)-interaction with routing protocol operation may cause instability
  - may not work very well at high speeds
  - other IP-level mechanisms exist

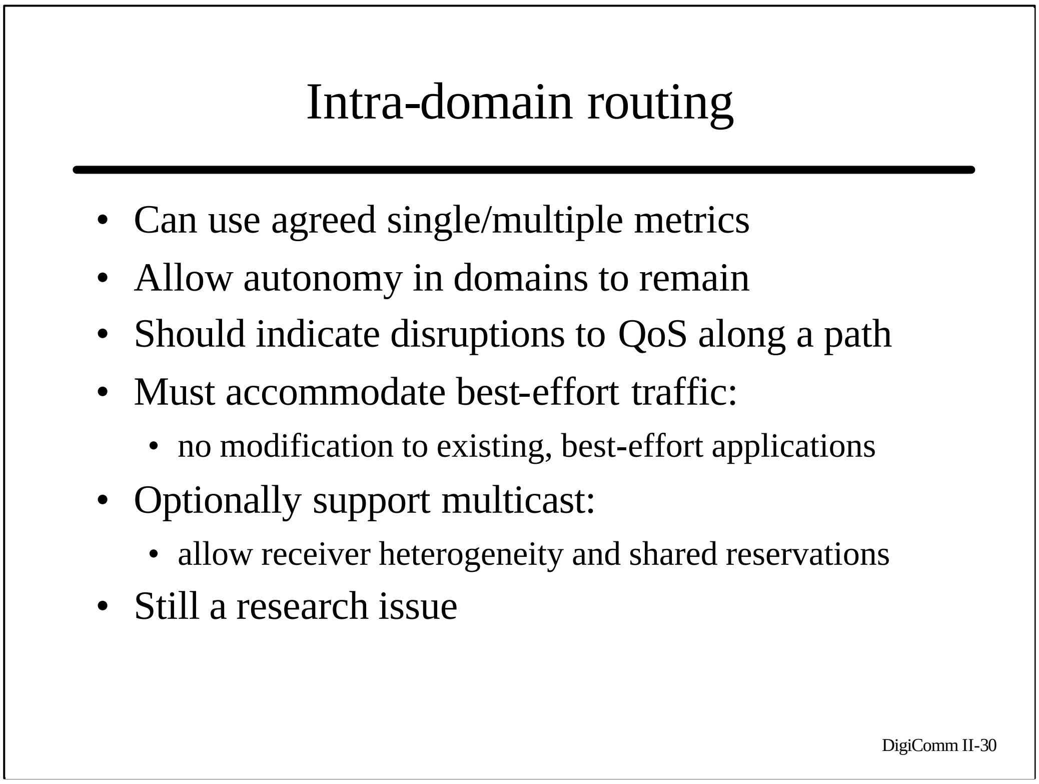
DigiComm II-29

The Multi-Protocol label Switching (MPLS) WG of the IETF is seeking to define a standard that will support fast-forwarding mechanisms.

It is intended that the use of MPLS in place of traditional IP forwarding will allow better performance and scaling in certain IP network scenarios. Its is intended that such mechanisms will help scaling an and performance of IP networks in certain environments, i.e. where it is likely that the layer-2 technology will offer a faster forwarding mechanism than the layer-3 forwarding of IP.

MPLS is designed to be complementary to existing routing mechanisms. Indeed, routing information is used to establish the forwarding entries used by MPLS.

Although independent of any particular bearer technology and any particular layer-3 technology, there is particular interest in finding MPLS solutions tailored to provide IP-over-ATM and IP-over-FR (Frame Relay) – Non-Brodcast Multiple Access (NBMA) network technologies.



Intra-domain routing

- Can use agreed single/multiple metrics
- Allow autonomy in domains to remain
- Should indicate disruptions to QoS along a path
- Must accommodate best-effort traffic:
  - no modification to existing, best-effort applications
- Optionally support multicast:
  - allow receiver heterogeneity and shared reservations
- Still a research issue

DigiComm II-30

Intra-domain QoS routing may be achievable by using mechanisms such as OSPF with ToS or DIFFSERV or traffic engineering in the underlying network. Multi-metric routing is possible with OSPF as we have already said.

The requirements listed in [RFC2386] for intra-domain QoS routing include:

• allow autonomy of operation within domains, as exist at the current time

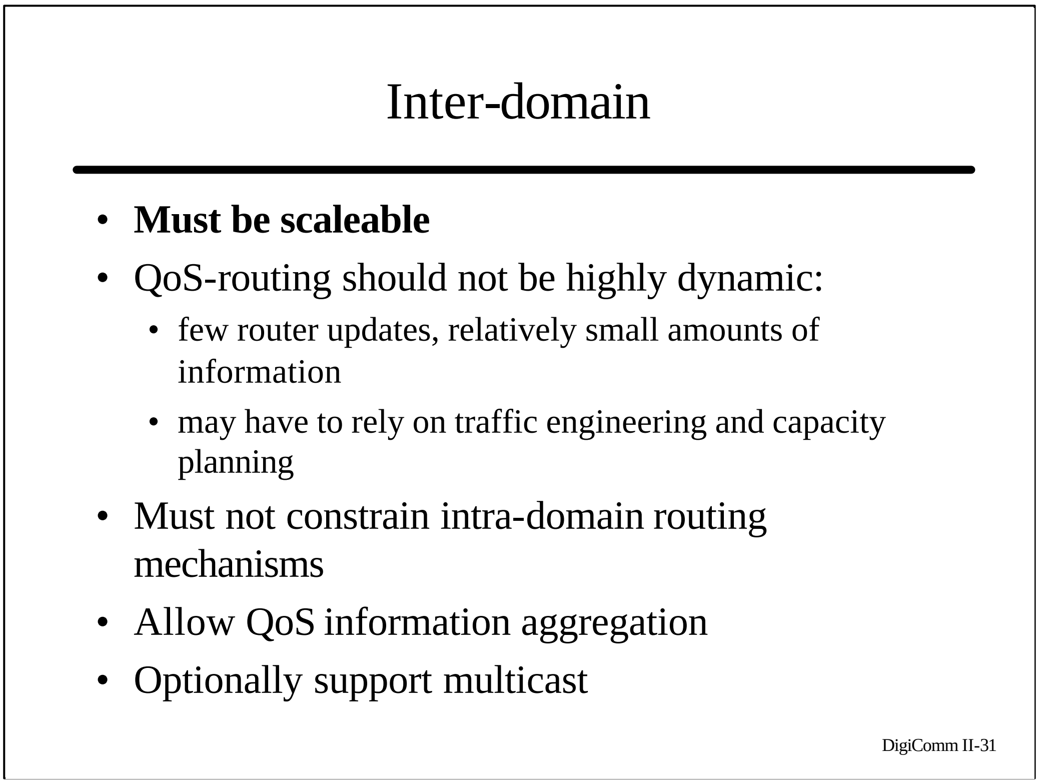
• flow must be routed along a path with QoS requested or requested/indicated or a notification must be generated to say that such QoS capability can not provided at this time

• indications of QoS disruption should be signalled during the lifetime of a flow if disruption is due topological changes

• must accommodate best-effort flows without requiring changes to the applications that generate them

• optionally support multicast and allow receiver heterogeneity and shared reservations

A QoS routing protocol that fulfils alls these criteria does not exist … yet.



Inter-domain

- **Must be scaleable**
- QoS-routing should not be highly dynamic:
  - few router updates, relatively small amounts of information
  - may have to rely on traffic engineering and capacity planning
- Must not constrain intra-domain routing mechanisms
- Allow QoS information aggregation
- Optionally support multicast

For inter-domain routing the key property that any QoS-based routing mechanism must possess is scalability. As there are large amounts of traffic between AS boundaries and the stability of the boundary routers is key to connectivity, we must ensure that such nodes are not subject to excessive load/processing due to the QoS-based routing mechanisms. To ensure this, [RFC2386] lists the following requirements:

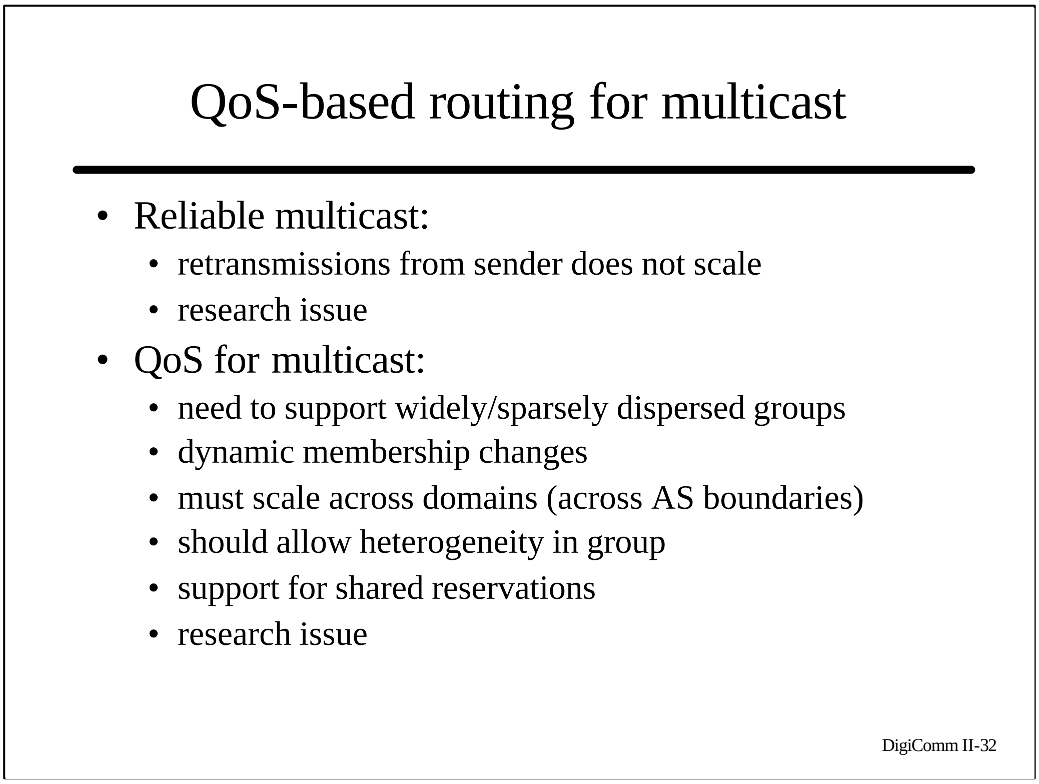
• QoS routing mechanisms must not be highly dynamic, there must be relatively few routing updates with small amounts of information. So, there may be a need to rely on more traditional forms of engineering, such as capacity planning, in order to ensure that border routers are kept lightly loaded

• metrics should be agreed and consistent. Internal AS/domain specific metrics may need to be mapped to metrics that have global semantics

• path computation should not be constrained, and be allowed to use QoS request for flows, path metrics, local policy, heuristics as well as other reachability information available from normal operation

• flow aggregation should be supported as it will not be practical to maintain state for thousands of individual flows. Mechanisms must be defined to ensure that aggregate flow descriptions for QoS are consistent with the combined requirements of the individual flows so composition and comparison rules for QoS metrics must be established

• optionally support multicast



QoS-based routing for multicast

- Reliable multicast:
  - retransmissions from sender does not scale
  - research issue
- QoS for multicast:
  - need to support widely/sparsely dispersed groups
  - dynamic membership changes
  - must scale across domains (across AS boundaries)
  - should allow heterogeneity in group
  - support for shared reservations
  - research issue

DigiComm II-32

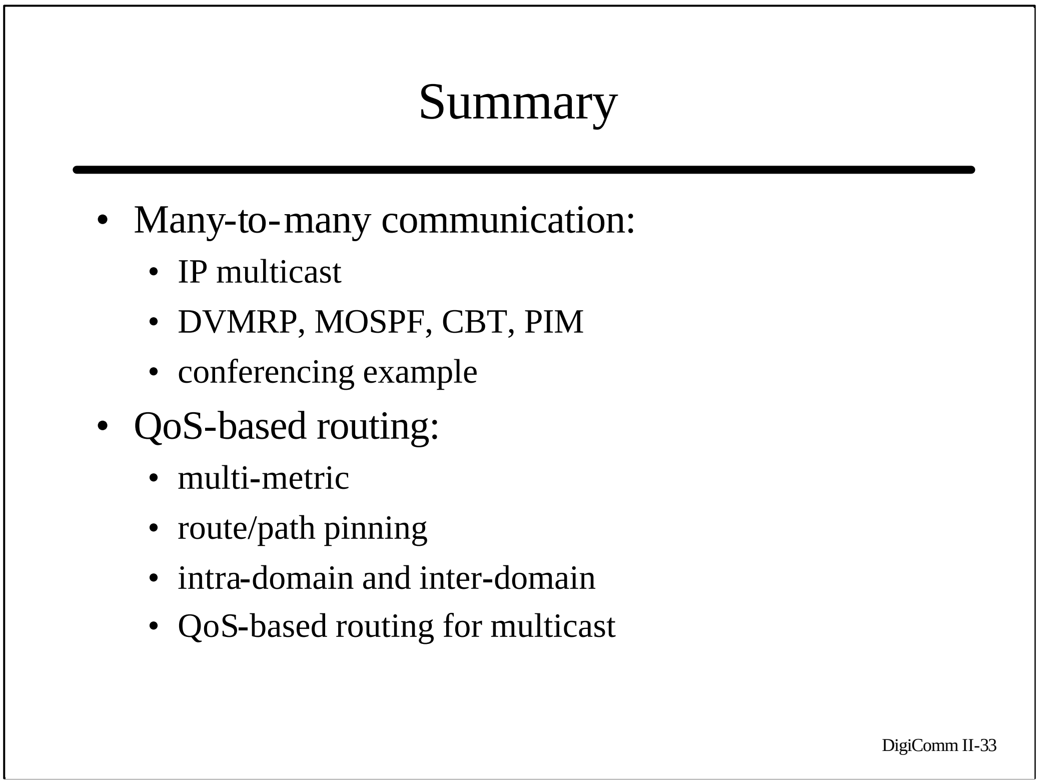
QoS for multicast is still a research issue.

For the moment, there is work in progress to develop reliable mu lticast, for example the Reliable Multicast Transport (RMT) WG of the IETF. Normal, sender-based based retransmissions coupled with acknowledgements form the receiver does not scale to the multicast environment.

RSVP/INTSERV was designed with multicast very much in mind but we have already seen it has scaling problems and does not support receiver heterogeneity very well. Also, reservation merging is inflexible. So, [RFC2386] lists these key requirements for QoS-based multicast routing:

- support widely and sparsely dispersed groups
- allow dynamic membership changes for groups
- scale across domains
- allow heterogeneity within groups
- support shared reservation styles

Needless to say, this is still a research issue.



Summary

- Many-to-many communication:
  - IP multicast
  - DVMRP, MOSPF, CBT, PIM
  - conferencing example
- QoS-based routing:
  - multi-metric
  - route/path pinning
  - intra-domain and inter-domain
  - QoS-based routing for multicast