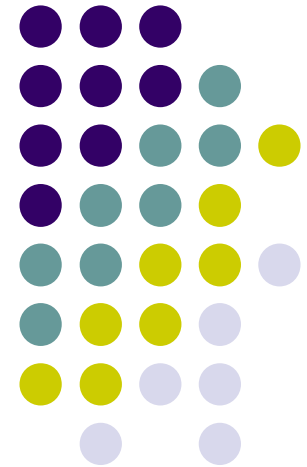


# IP Multicast

---

*Stig Venaas*  
*sv@ecs.soton.ac.uk*



# Overview



- What is multicast and why is it useful?
- Service model and routing challenges
- Multicast routing
  - Flood and prune
  - More sophisticated ways
- Source-Specific Multicast (SSM)
- Interdomain multicast routing
- Why isn't multicast widely used today?

# What is IP multicast?



- Usually an IP packet is sent to one specific host
  - The IP destination address specifies which host
- With IP multicast an IP packet is sent to a group of hosts
  - The IP destination address is a group address and not a host address
    - IPv4 multicast addresses, class D. 224.0.0.0 – 239.255.255.255
    - IPv6 multicast addresses, FF00::/16 (all addresses starting with FF)
  - The group can contain any number of hosts (0 to infinity)
  - The group members can be anywhere
  - A bit like IP subnet broadcast, where a single packet is received by all on the subnet. Multicast is not restricted to the subnet though, and is not sent to all the hosts
- The multicast packets will be replicated by routers where needed
  - The routers keep track of which interfaces they should forward the packet on
  - The same multicast packet is ***never sent twice on the same link***, hence the bandwidth used on a specific link is ***independent of the number of receivers***

# Why is it useful?



- Imagine BBC streaming TV on the Internet to every UK home
  - Using multicast, they only need a relatively basic machine and their Internet connectivity need not be better than a UK home
    - Remember, to send you don't need more bandwidth than a single receiver
  - BBC did tests sending the last Olympic Games over multicast
- An ADSL user could easily send video to thousands of other users, again the number of receivers is not an issue
- Also useful for multi-party applications like conferencing or gaming where typically each participant wants to send the same data to all the others
- Multicast also useful for doing discovery
  - Imagine all printers on your network joining a specific multicast group
  - You might then be able to send a query to all the printers (and not any other hosts) asking them to identify themselves

# Service model and routing challenges



- The basic multicast service model is as follows
  - Anyone can send to the multicast group
    - Senders don't need to know where the receivers are or how many (if any)
  - Hosts interested in the group join it
    - They don't need to know who is sending, where they are, or what other receivers there are
    - They just receive anything sent by anyone to the group while they are members of the group
- The big challenge is routing
  - If anyone can be anywhere and only telling the routers which group they are sending to or joining, how can routers learn from where and to where they should forward the data
- In the beginning there was multicast only on ethernet links, no routing
  - This was trivial, especially when ethernet was just a single coax cable or a hub. With switches it's more complicated
- Then one wanted to do routing across larger networks...

# Multicast routing

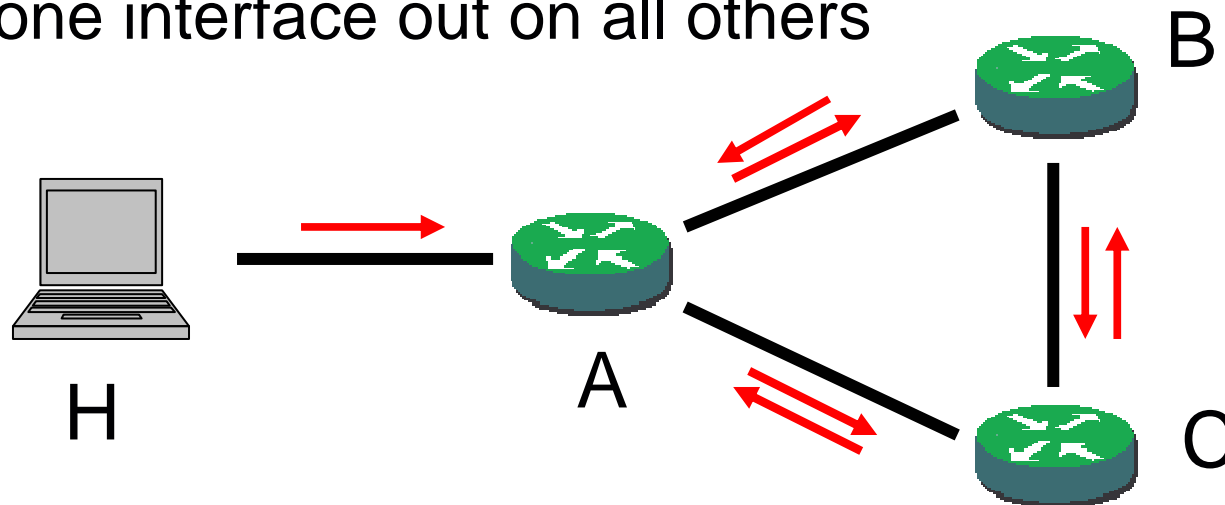


- The simplest possible solution would be to flood multicast over the entire network
  - Ensures that everyone interested gets it. But also all the others
    - This is what simple ethernet switches do
  - Many multicast routing protocols are so-called “flood-and-prune” protocols
    - Pruning is a way of restricting the flooding to where it’s wanted
- To do something more intelligent than flooding, routers need to know what groups the connected hosts want
  - This is done using the protocols IGMP and MLD for IPv4 and IPv6 resp. These are used only on the local network between the hosts and the routers
- We will now first discuss flooding, and next take a look at the pruning mechanisms

# Flooding



- Trivial, isn't it? Every router just forwards what it gets on one interface out on all others

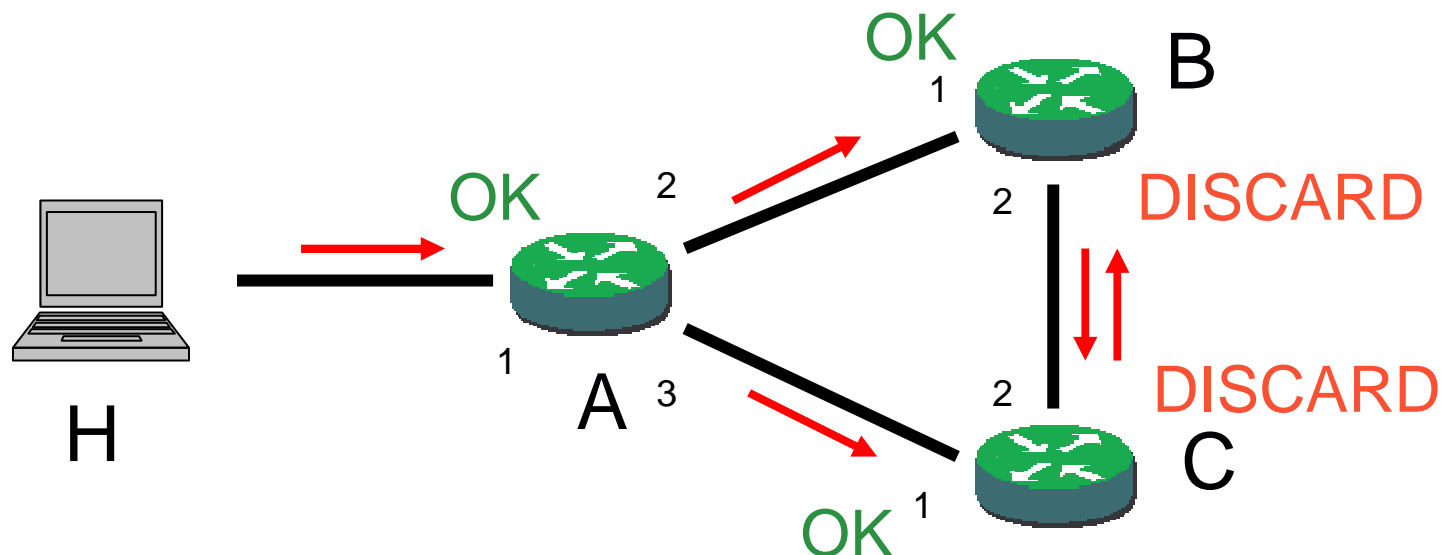


- But what about loops?
  - Same packet on same link many times
  - One could imagine packets going in a loop forever
    - However, there is ttl/hop limit in IP packets, so might be restricted to say 40 router hops
- So how do we stop this from happening?

# Reverse Path Forwarding (RPF)



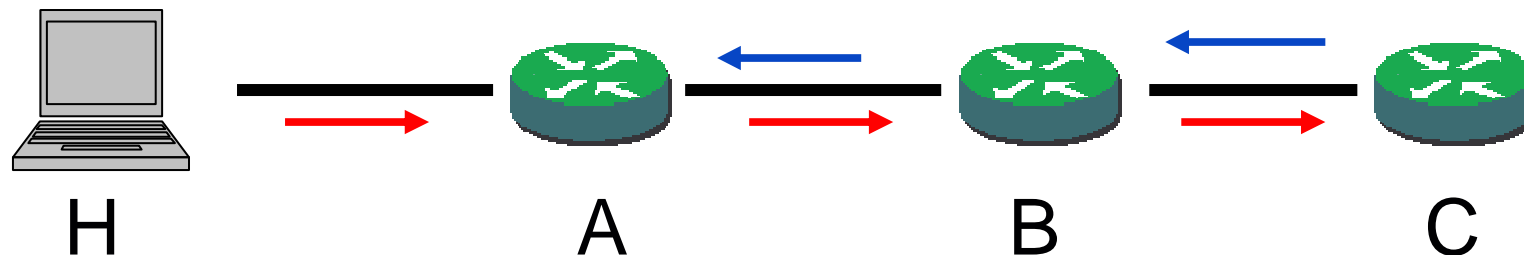
- RPF solves most of the problem
- The idea is that a router discards a multicast packet if it isn't coming in on the interface which it would use for sending packets out to sender
  - In this example, H sends a multicast packet. A accepts it on interface 1, because it would use interface 1 for sending packets to H
  - B and C also receive packet from H on their interface 1 and accepts it since they would use interface 1 for sending to H
  - Next B and C receives packet from H on their interface 2. Both discard it because they would not use interface 2 for sending to H
- How to avoid duplication on the link B-C is also solved by some protocol



# Pruning



- For each group we want to build multicast distribution trees that are rooted at the sources and only branching out to where there are receivers
  - Hence we want to prune unwanted branches
- An edge router (with directly connected hosts) needs to know which groups the hosts are interested in
  - As we said earlier, this is done using the IGMP and MLD protocols
- If the edge router receives a packet on one interface and it knows that no one on the other interfaces is interested, it can send a prune message saying, “Hey! Don’t send this stuff this way, we don’t want it over here”.
- Similarly the router in front of the edge router may itself send a prune if it receives a multicast packet on one interface, and has received prunes for the group on all the others
- Remember that this pruning is done per group



# Flood-and-Prune

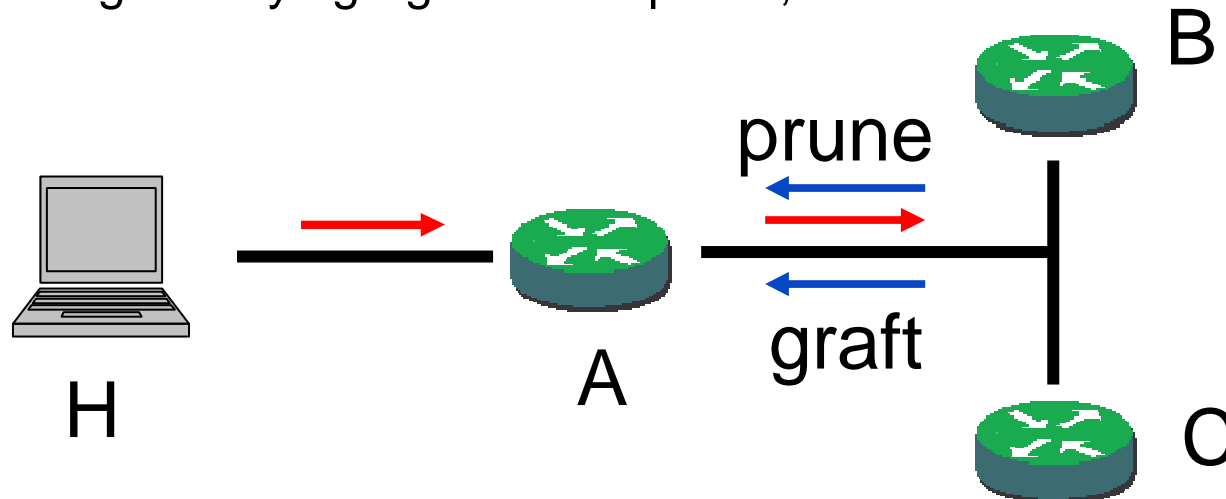


- Initially packets for a group are flooded everywhere
- Routers at the edge start sending prunes when not interested
- Pruning propagates towards the sources so that end up with trees only branching out where needed, hence packets flow only where needed
- The flooding is repeated periodically
  - Network topology might change, links that were broken are back up etc.
- No state is permanent
  - Prunes triggered by data packets
  - Routers remember they have received prunes for some period of time. When they forget, the flooding will be repeated

# Grafting



- What if a router didn't have anyone interested and sent a prune message. Then right after a host joins the group. Rather than wait for the prune to expire, it sends a graft message saying "Sorry, forget what I told you, I want packets anyway".
- It's also useful in another case
  - Imagine A, B, C being connected to the same ethernet link. Then A receives from H and forwards on the link.
  - If B is not interested, it will send prune to A. But what if C is interested?
  - C will then see the prune (it's multicasted on the link) and send A a graft saying "Ignore that prune, there is still someone that wants it"



# Multicast routing protocols



- Flood-and-Prune is okay if receivers are densely populated. Not so good for a sparse population which would typically be the case for the Internet
- One sometimes talk of dense and sparse mode protocols
- The most commonly used routing protocol today is PIM (Protocol Independent Multicast)
  - It's said to be independent because it typically uses the unicast routing tables to do RPF. Which protocols are used for maintaining the routing tables does not matter
- PIM has two flavours, PIM-SM (Sparse Mode) and PIM-DM (Dense Mode)
- PIM-DM is pretty much flood-and-prune as we have seen. There are some minor improvements
- We will now look at PIM-SM which is commonly used, and is the protocol most suited to Internet use

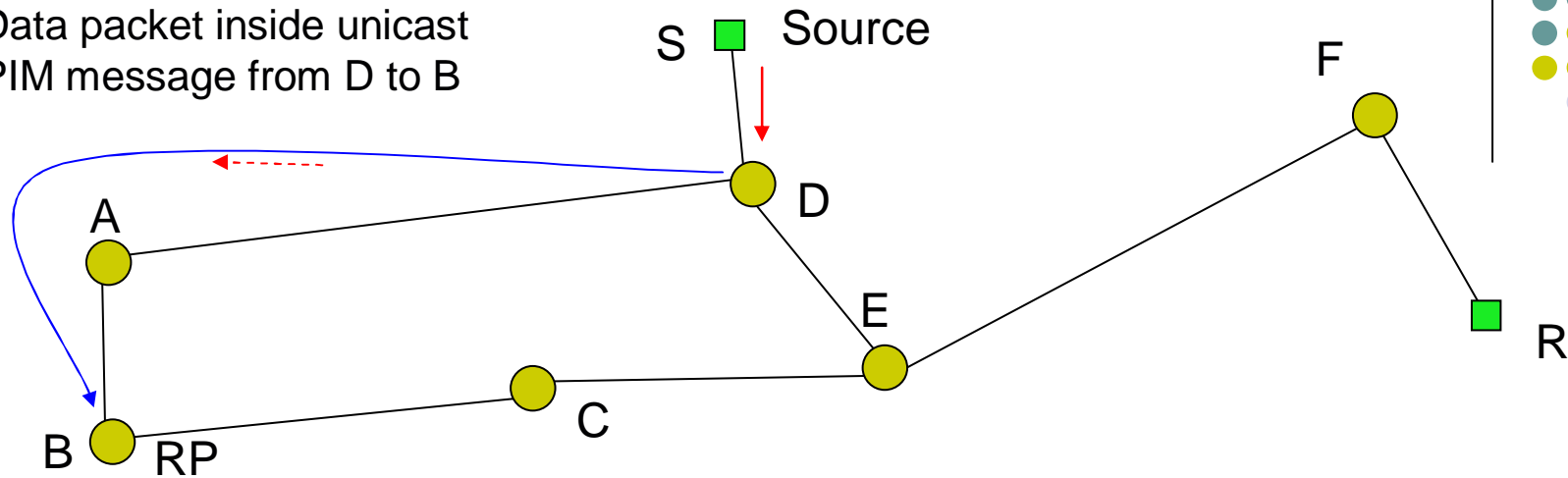
# PIM-SM (PIM Sparse Mode)



- Edge routers may learn that hosts behind them want to receive some group, but who should they tell that they are interested? They don't know where the senders (called sources in multicast terminology) are
- The sources might send, but when the edge router next to the source receives it, what should it do with it? We don't want to pass it on to other routers unless there is someone there that wants to receive it
- PIM-SM solves this by introducing something called a Rendezvous Point (RP)
- An RP is like a meeting place between sources and receivers. The source somehow sends their stuff to the RP, and the RP somehow learns where the receivers are

# How the RP learns of sources

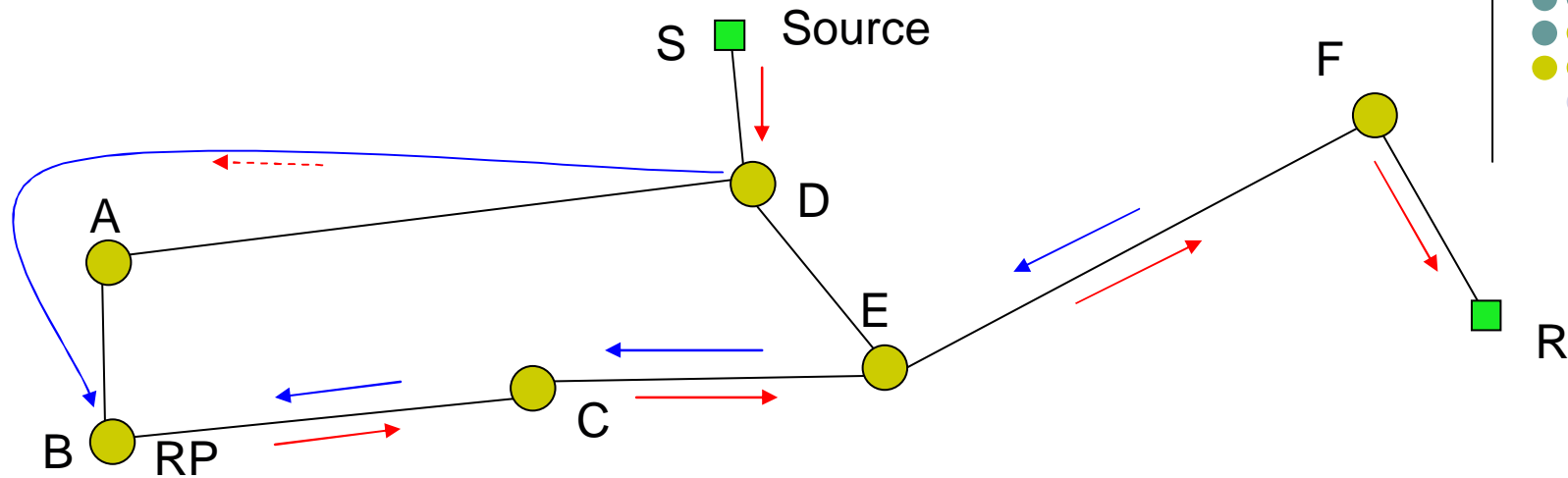
Data packet inside unicast  
PIM message from D to B



The blue arrows are PIM messages and the red are data packets

- All the routers are preconfigured with the same RP address B
- We're looking at what happens initially when a source starts sending
- On the link where the source is, one router is elected as Designated Router. It will encapsulate multicast packets into unicast, addressed to the RP. Here router D is the DR
  - Imagine you have a letter with a multicast address on the envelope. You then put that letter (envelope and all) into a new envelope and put the unicast address of the RP on that one
- So in this way, the RP B will learn about the source, as well as receiving the actual multicast data

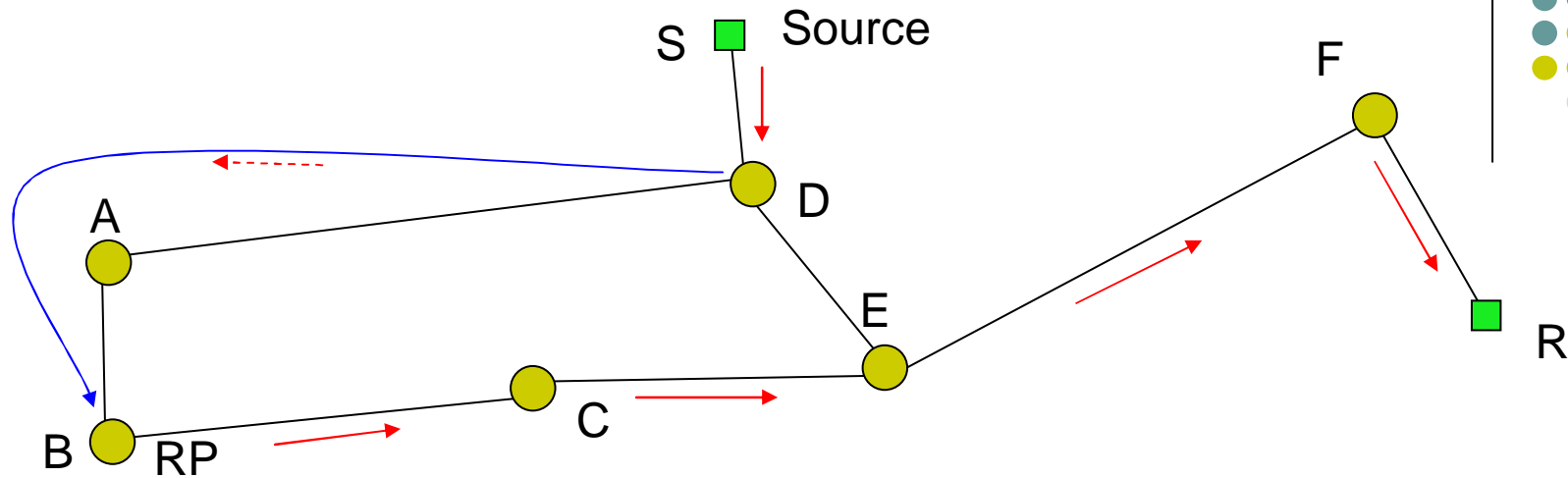
# How the RP learns of receivers



The blue arrows are PIM messages and the red are data packets

- All the routers are preconfigured with the same RP address B
- We're looking at what happens when a host R behind router F joins the group
- F should send a PIM join message towards the RP. It checks the routing table, and finds that E is the next hop for reaching B, so it sends the join to E
  - E is said to be F's RPF neighbour for B
- When E receives the join from F, it will find that C is next hop towards the RP, and sends join to C.
- Finally, C will send join to B which is the RP
- If RP is receiving any data (in this case it is), it will as soon as it receives the join (from C here) forward data. Data goes back to receiver following the joins backwards

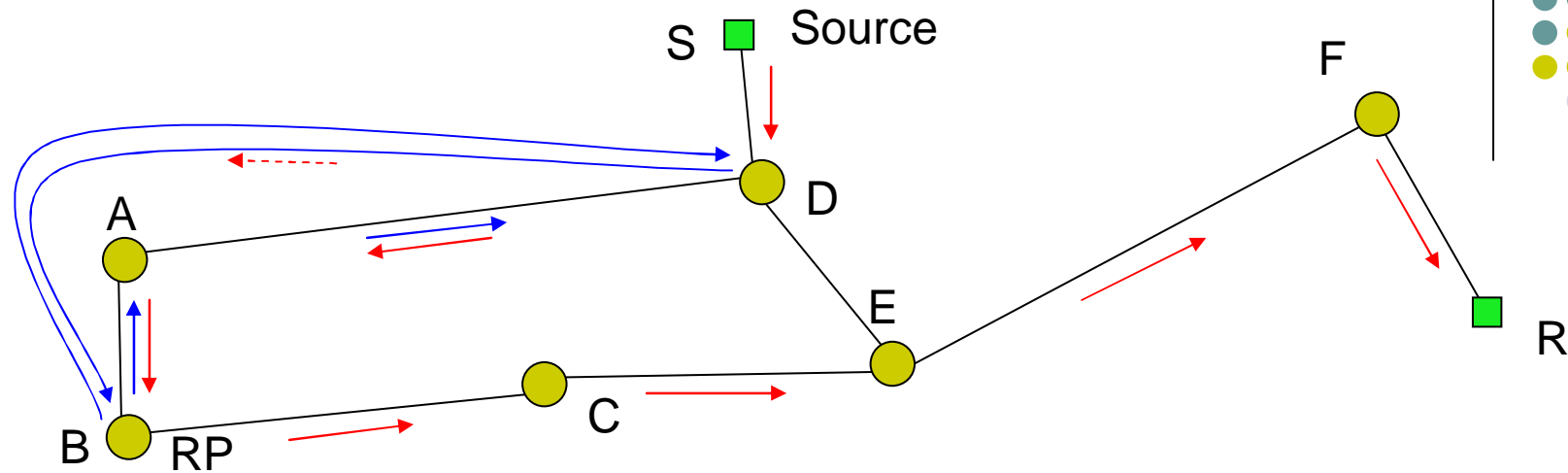
# Avoiding unicast encapsulation 1/2



The blue arrows are PIM messages and the red are data packets

- Packets are now flowing from source to interested receivers, but typically wants to avoid encapsulating all multicast into unicast. Requires lots of resources to encapsulate packets at the DR (D) and then decapsulate them at the RP (B)
- PIM-SM allows joining towards a specific source, and this is what the RP will do when it receives encapsulated packets and there is someone wanting to receive them
  - If no one wants to receive, there is also a way for the RP to tell the DR to not send anything for a while

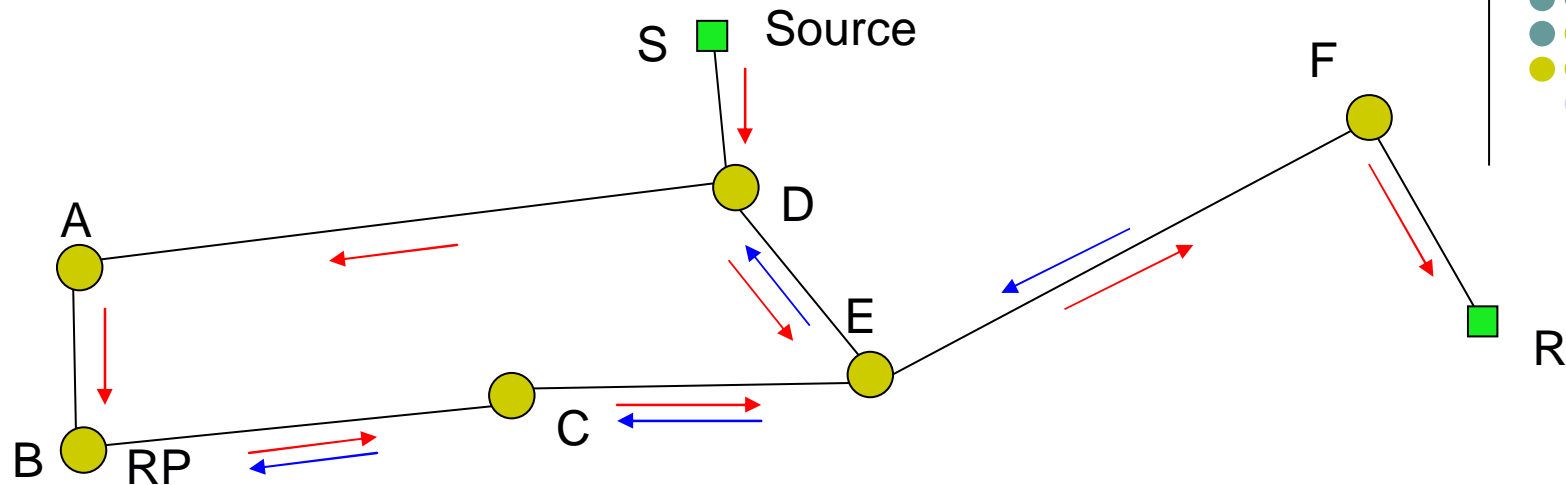
# Avoiding unicast encapsulation 2/2



The blue arrows are PIM messages and the red are data packets

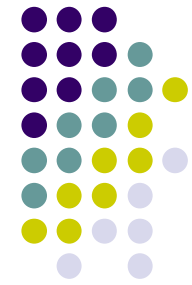
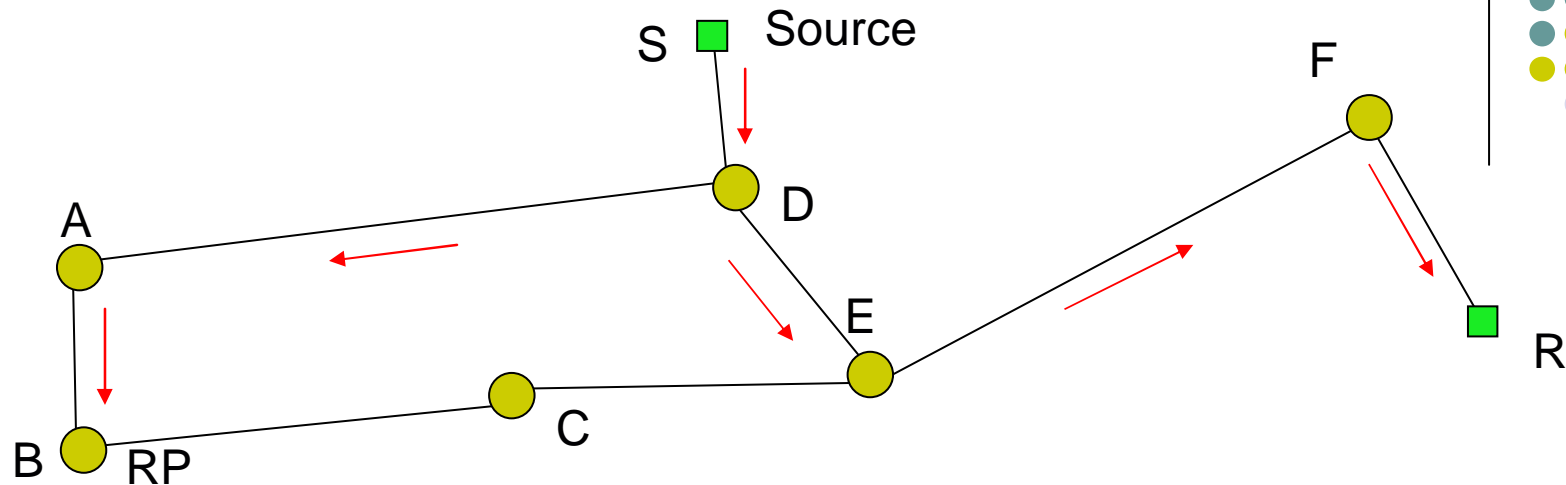
- First B sends source-specific joins towards source S
- B will after joins reach D, start receiving packets natively (not encapsulated)
- B will then send a unicast message to D, asking it to stop sending encapsulated packets from S

# Optimising forwarding path 1/2



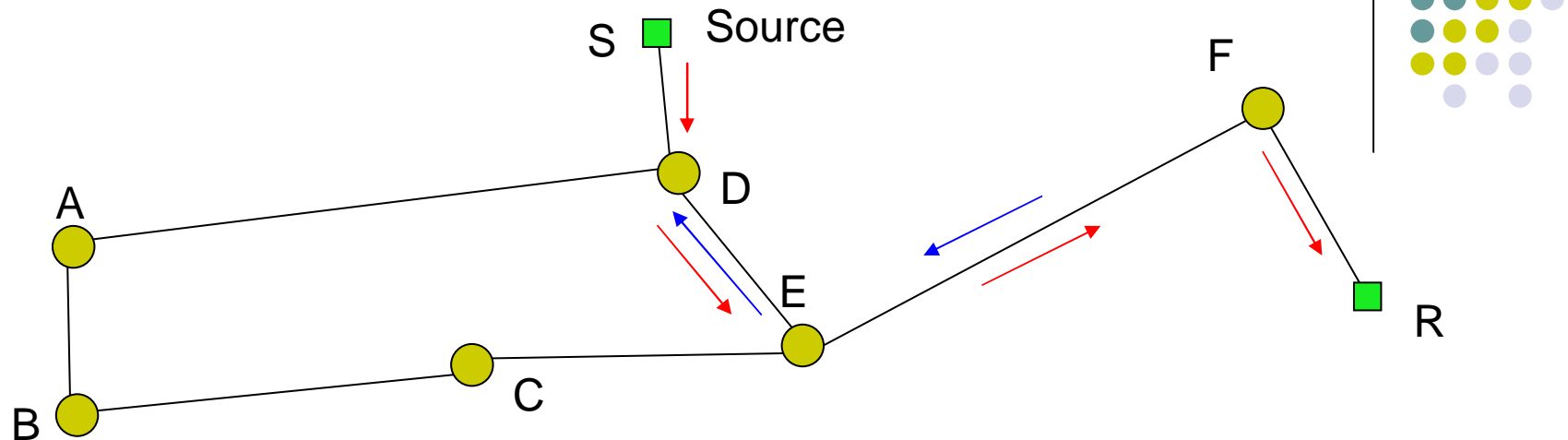
- Now packets are flowing natively, but why send through RP when could go the shorter path from D to E?
- F is often configured to immediately join towards sources
  - It can do this after first packet from S. It sees S's address in the IP header
- So F sends source-specific join towards S
- When these joins reach D, D will also forward multicast packets to E
- When E receives from D, it will send prune messages towards RP saying, don't send me packets from S
  - It might still receive from other sources via the RP

# Optimising forwarding path 2/2



- Now almost everything is perfect
- But why should RP receive data if the receivers get it directly?
- So, if no one wants to receive S from RP, it will stop joining S
- Joins are in general sent periodically. If no joins are sent, state will expire and no more multicast will be sent
- Finally, everything is optimal

# Source-Specific Multicast



- With SSM, the receiver somehow knows the source address S
- So host R tells F it wishes to receive a particular group from S
- F then sends source-specific join towards S
- So we immediately get the optimal path and no RP is needed
- Also this is a bit more secure. Other sources cannot disturb by sending to the group. R only receives from the sources it has asked for
- There is a problem though. How can the receiver R, or the application at the receiver, learn the source addresses?

# PIM-SM Summary



- PIM-SM requires an RP for source discovery
- All routers must use the same RP and somehow know the address for it
- Initially packets from a source will be sent to RP
  - Even if no one wants to receive
- Except for this, packets are only sent out on an interface if a join has been received on it
- Initially packets flow from source to receivers via the RP
- Optimal path (not via RP) usually established quickly
- With SSM one can immediately construct the optimal path, but receivers need to know the source addresses

# Interdomain multicast routing 1/3



- What we've seen so far works fine inside a single organization, or say for a single provider and its customers
- But impossible to have everyone in the world agreeing on using the same RP
- Part of the problem is that two parties A and B, don't want to rely on a third party C to run an RP properly
- So, people want to have their own RPs
  - Different organizations with different RPs still want to talk to each other though
  - But an RP doesn't know about sources and receivers at other RPs
  - So for IPv4 there is a protocol called MSDP that lets RPs exchange this information
- MSDP is generally believed not to scale with widespread multicast use. Mainly because every RP tells every other RP in all other organizations what is going on
- For IPv6 another simpler solution exists

# Interdomain multicast routing 2/3



- Idea is that for almost every group you can have a unique RP. Someone hosting a multicast session, doing streaming or whatever, have their own RP, and that RP will be used by everyone using this group address
- If the party providing the service also is responsible for the RP, there is no 3<sup>rd</sup> party dependencies as mentioned on previous slide
- The problem here is that all routers on the internet need to know which RP to use for each group, and all routers must use the same RP for each group
- The IPv6 solution is to pick particular group addresses that have the address of the RP encoded into them. This technique is called Embedded-RP
  - This is possible by putting some restrictions on which addresses are used for RPs. The idea is mainly to have RP addresses with many 0-bits in them so that they can be encoded into fewer bits
  - An IPv6 embedded-RP multicast address is identified by beginning with ff70::/12
  - A router seeing such a group address knows it is an embedded-RP address and knows how to compute the RP address

# Interdomain multicast routing 3/3



- There is simply too few bits in an IPv4 address to do the same
- To do something similar for IPv4, one would need some kind of new protocol allowing the routers to learn which RPs to use
- There is also a problem agreeing on who should use which addresses for their sessions
  - For IPv6 with Embedded-RP this is based on the RP address. This is a unicast address and organizations already have unique unicast addresses. As a consequence, the embedded-RP multicast addresses are unique per organization

# Why isn't multicast widely used today?



- It's used a bit in academic networks and some other specialised uses. BBC did tests sending last Olympic Games over multicast though
- Multicast as you see is a bit complicated and requires network administrators, ISPs etc to have people that understand it. So this takes extra effort and must somehow be paid for
  - Can multicast be made simpler to understand and manage?
    - SSM might be a key here, also embedded-RP for IPv6 is simpler than MSDP
  - How can providers gain from this?
  - They do gain some by having less traffic in their network
  - Perhaps those providing content gains the most? Could they somehow pay the ISPs?
  - Would end-users pay extra to get multicast?