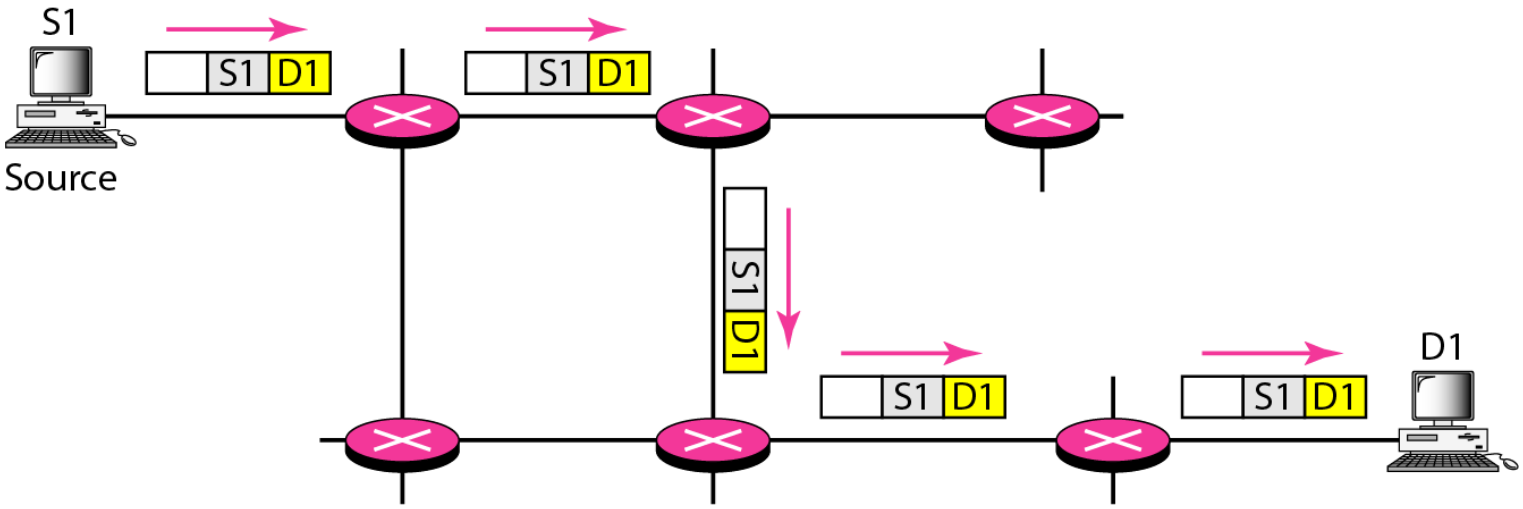
CS2031
Telecommunications II

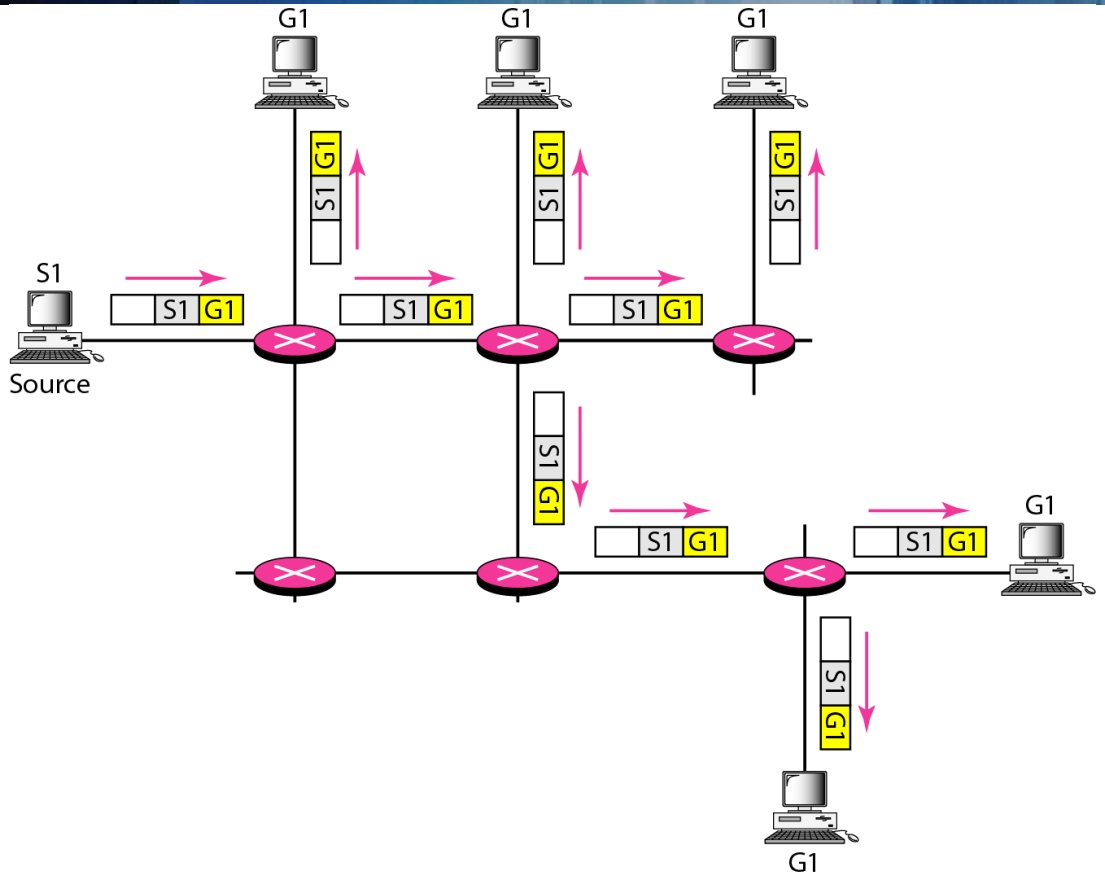
Multicast Routing

Routing & Unicast



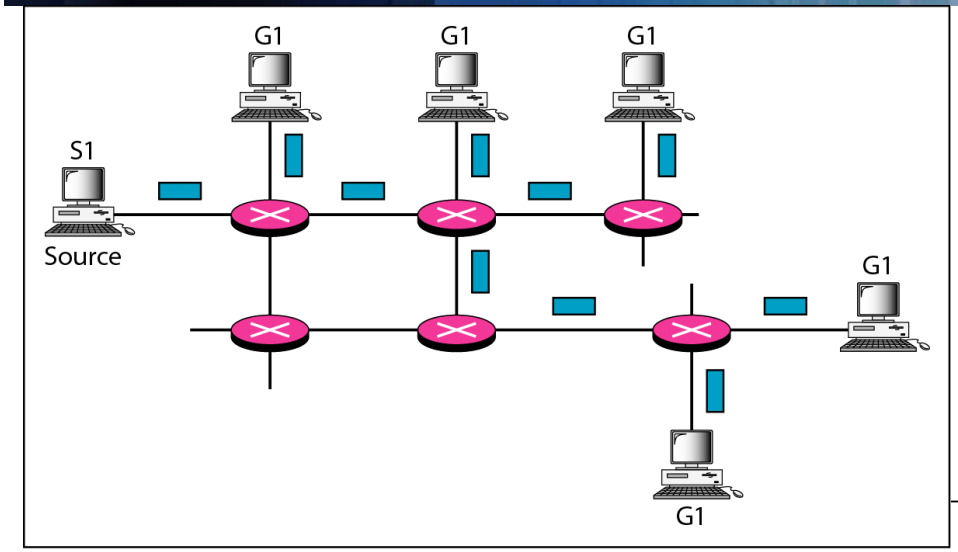
- Routers guide traffic towards destionation

Multicast

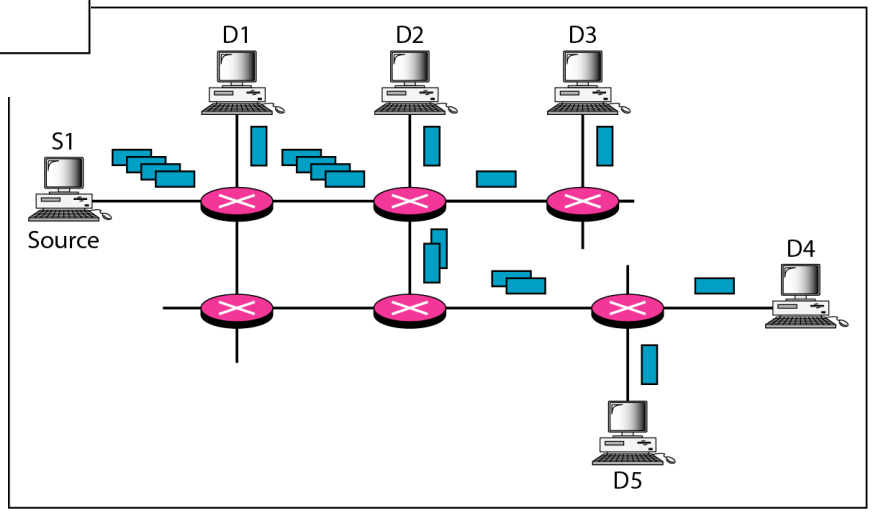


- G1= multicast address e.g. 230.0.0.1

Multicast vs Multiple Unicasts



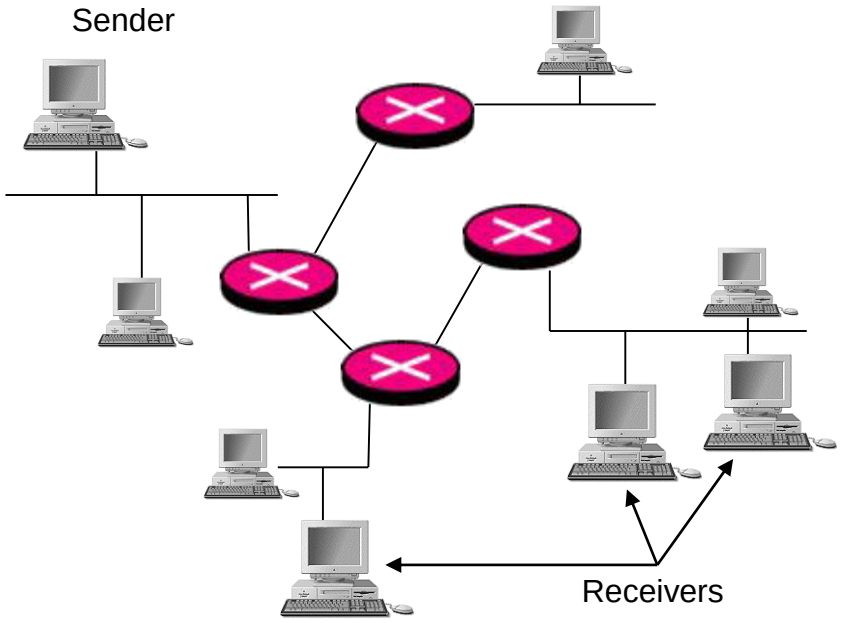
a. Multicasting



b. Multiple unicasting

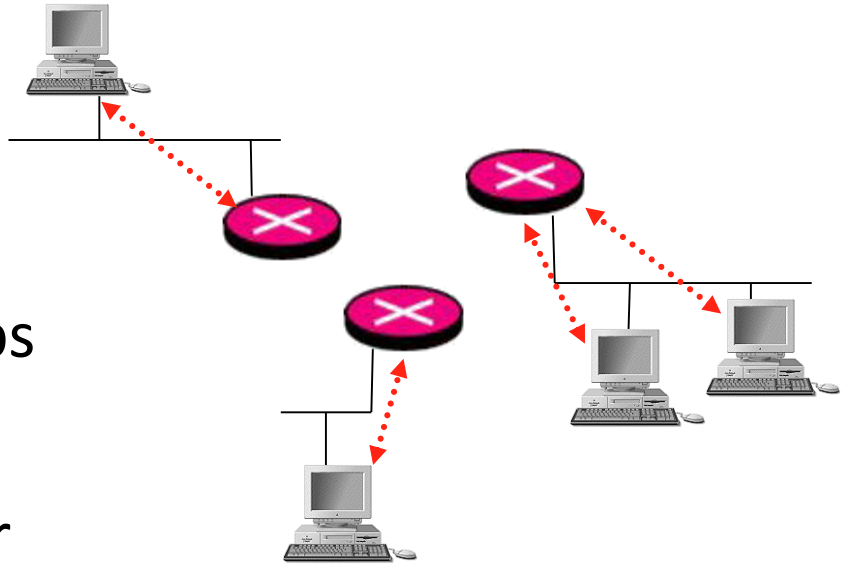
Multicast Overview

- Multicast requires group management
- Receivers join&leave multicast groups
- Multicast Addresses: 224.0.0.0 – 239.255.255.255 or 224.0.0.0/4

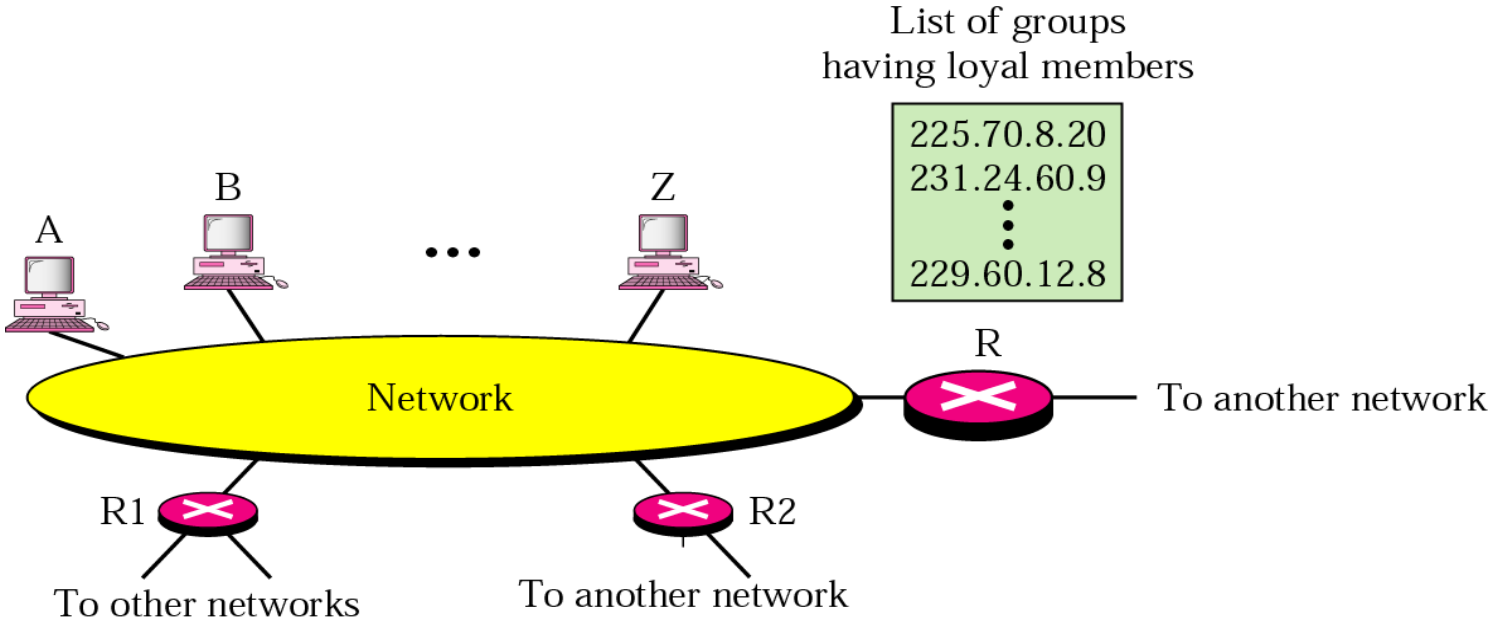


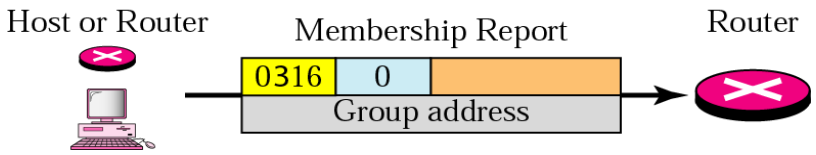
Internet Group Management Protocol (IGMP)

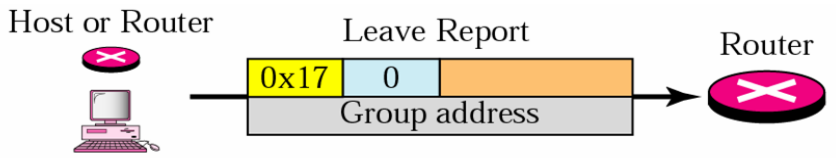
- Defines communication between hosts and router
- Specifies messages for hosts for joining and leaving groups
- Specifies query messages for routers

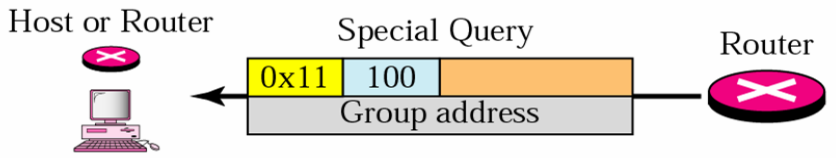


IGMP Operation



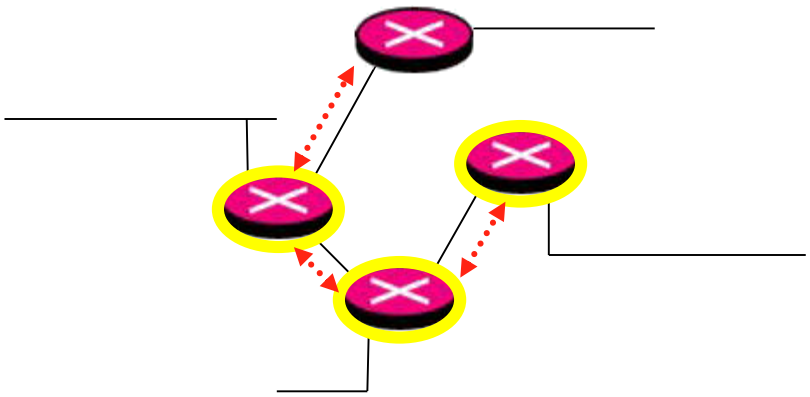




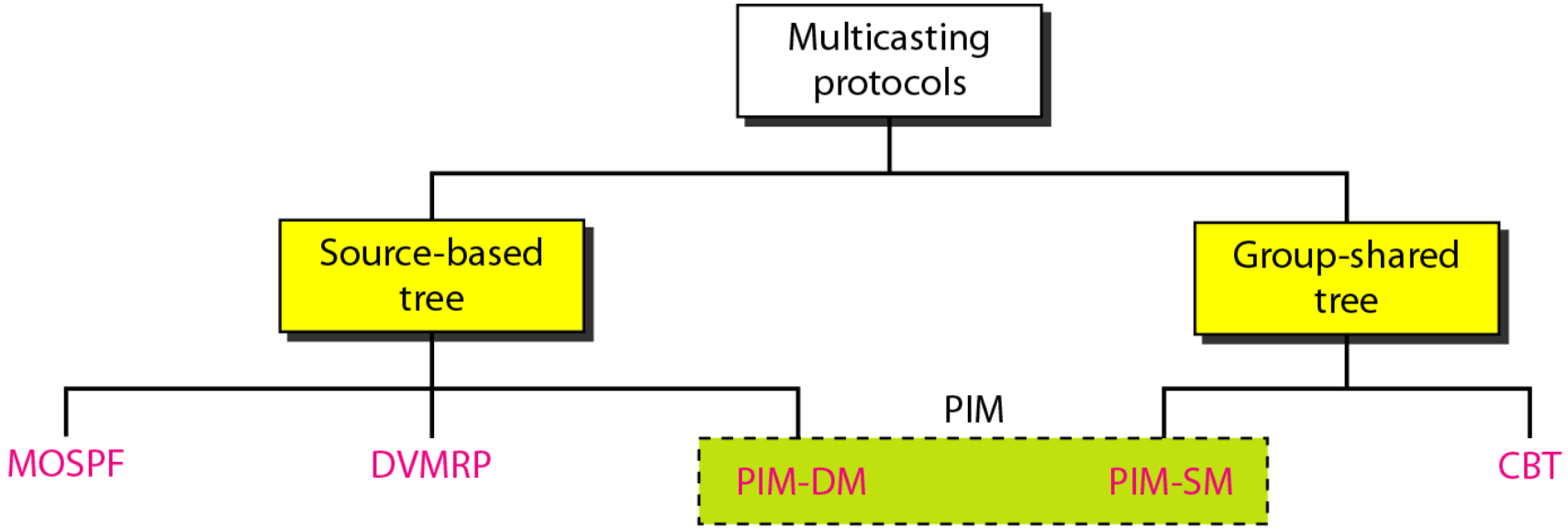


Network-Layer Multicast Protocols

- Distance Vector Multicast Routing Protocol (DVMRP)
- Multicast Open Shortest Path First protocol (MOSPF)
- Protocol Independent Multicast (PIM)



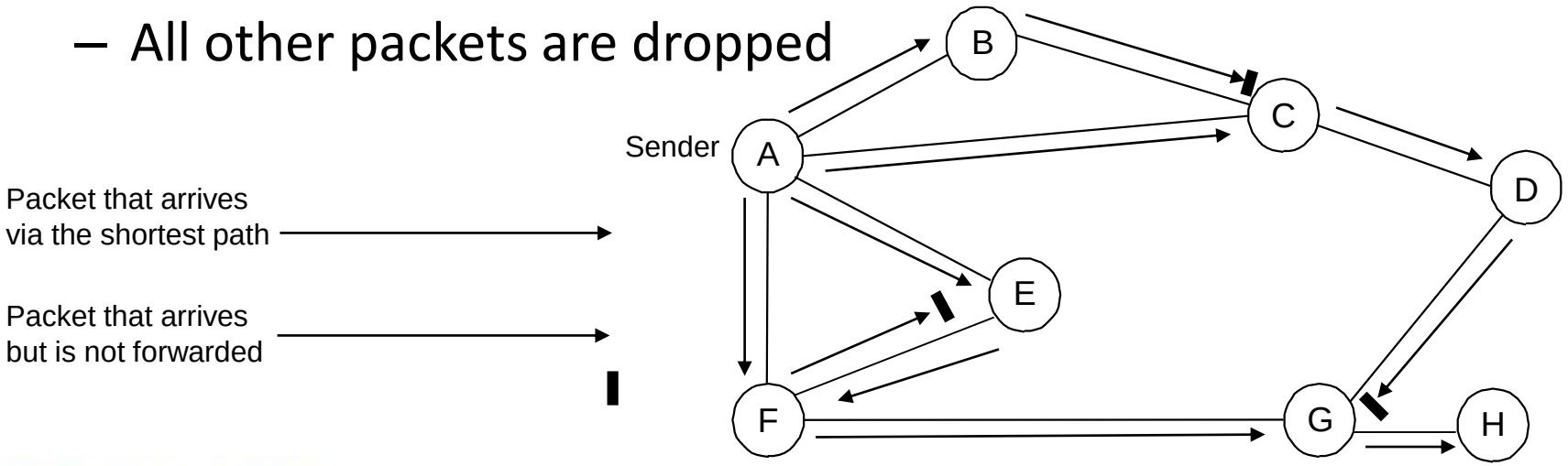
Multicast Routing Protocols



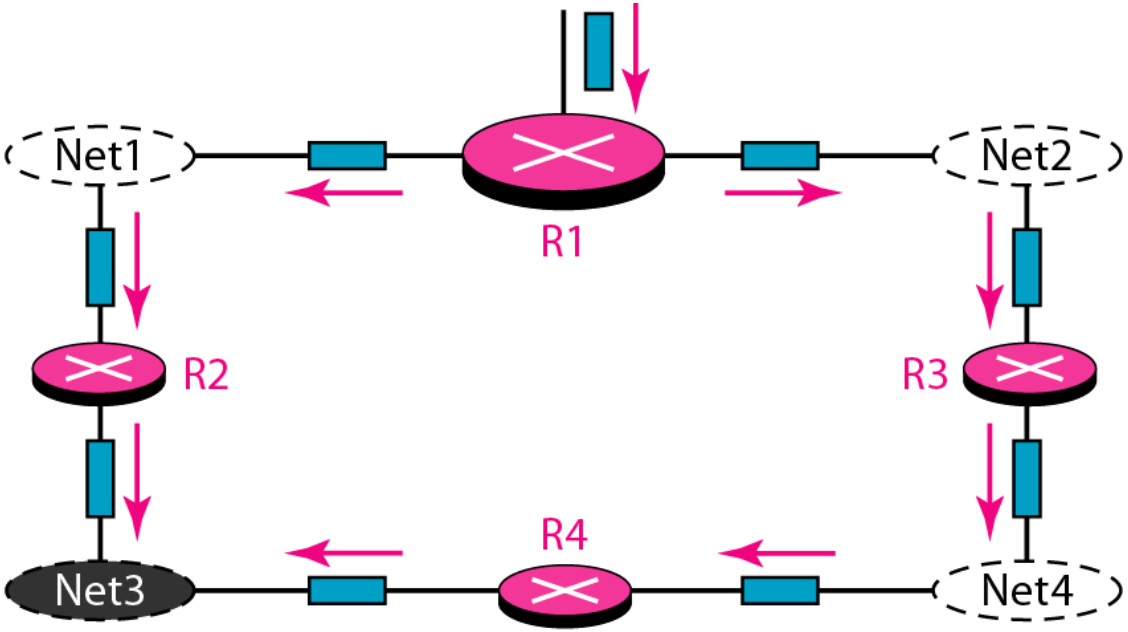
- Intra-AS
  - MOSPF
  - DVMRP
  - PIM
    - Sparse mode
    - Dense mode
- Inter-AS
  - MBGP + MSDP
  - BGMP + MASC

Reverse-Path Forwarding (RPF)

- Reverse-path forwarding simulates spanning tree routing without keeping state in the router
  - Each router knows shortest path to destination
  - Packets from A arriving on next hop to A are presumed to have followed shortest route from A, so they are forwarded on all other links
  - All other packets are dropped



Problem with RPF

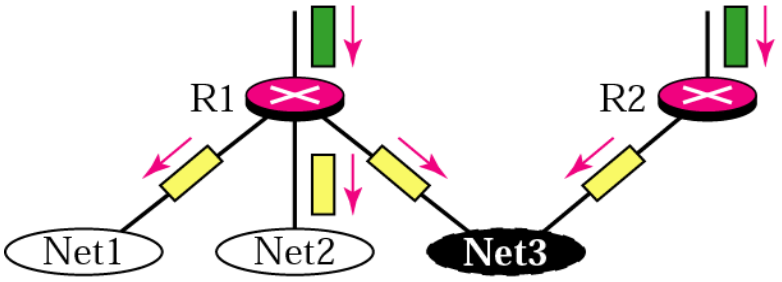


Net3 receives two copies of the packet

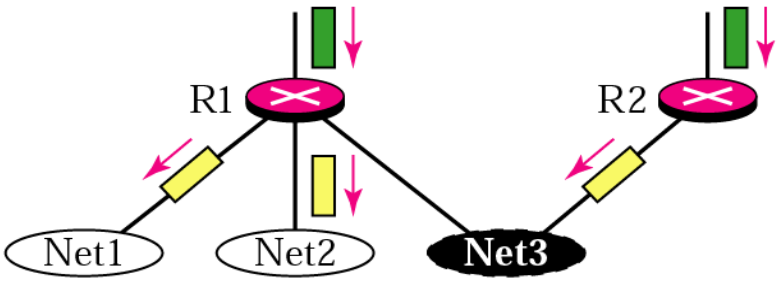
From R.P. Broadcast to Multicast

- Reverse Path Broadcast

R1 is the parent of Net1 and Net2.
R2 is the parent of Net3.

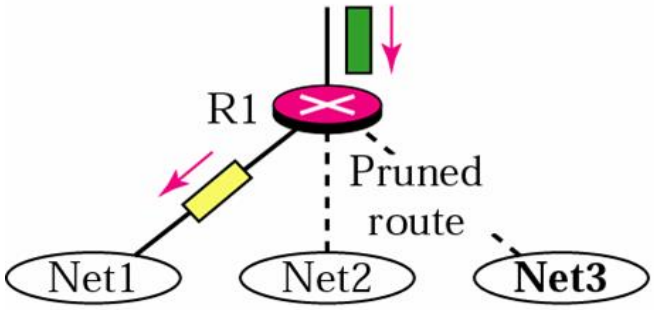


a. RPF

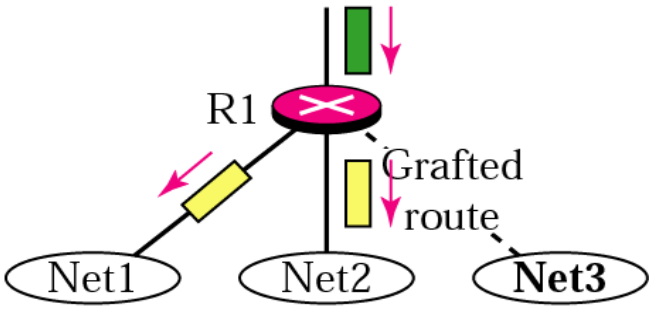


b. RPB

- Reverse Path Multicast



c. RPM (after pruning)



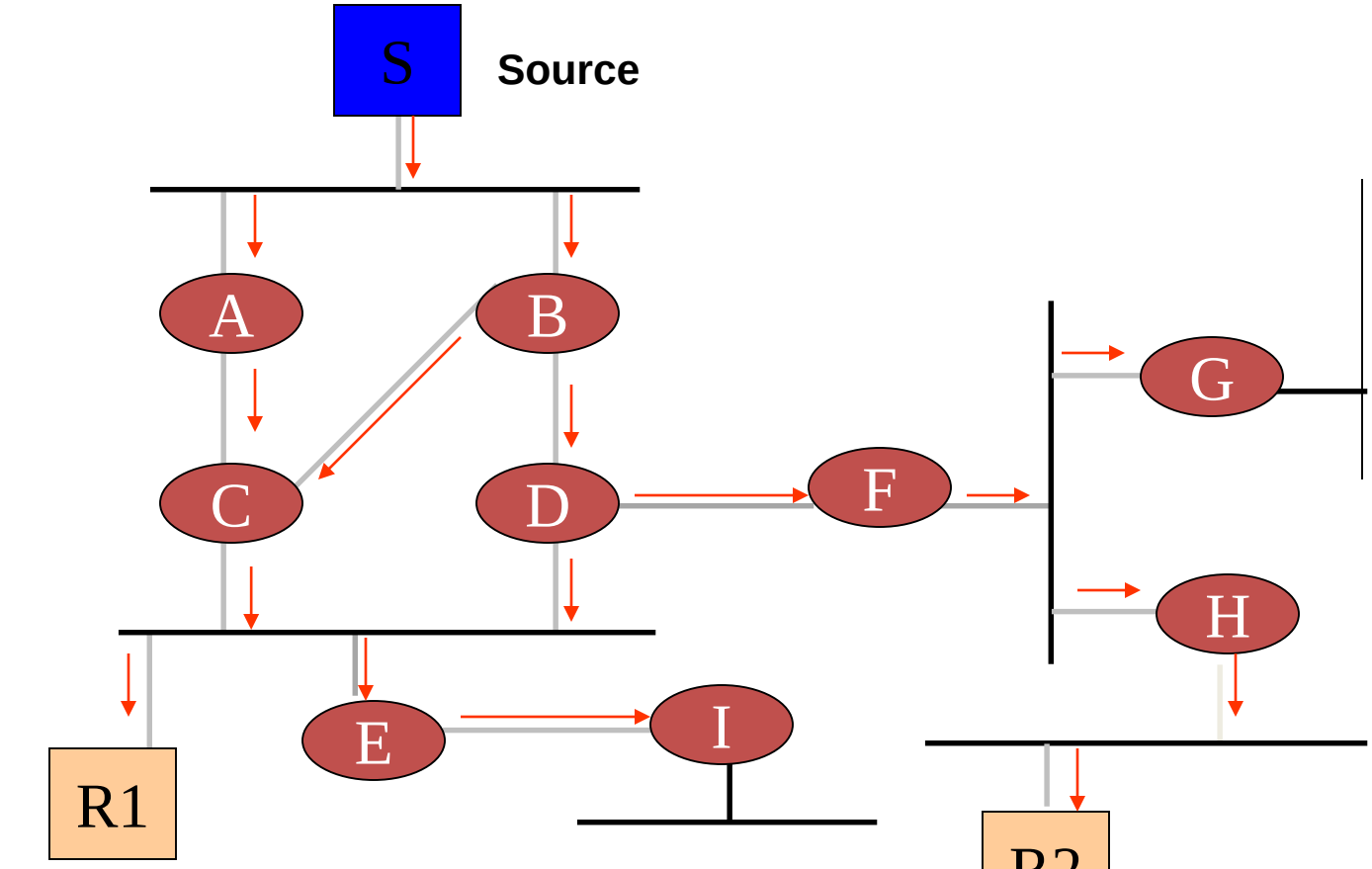
d. RPM (after grafting)

PIM – Dense Mode (DM)

- When it is likely that many routers are involved in multicast routing

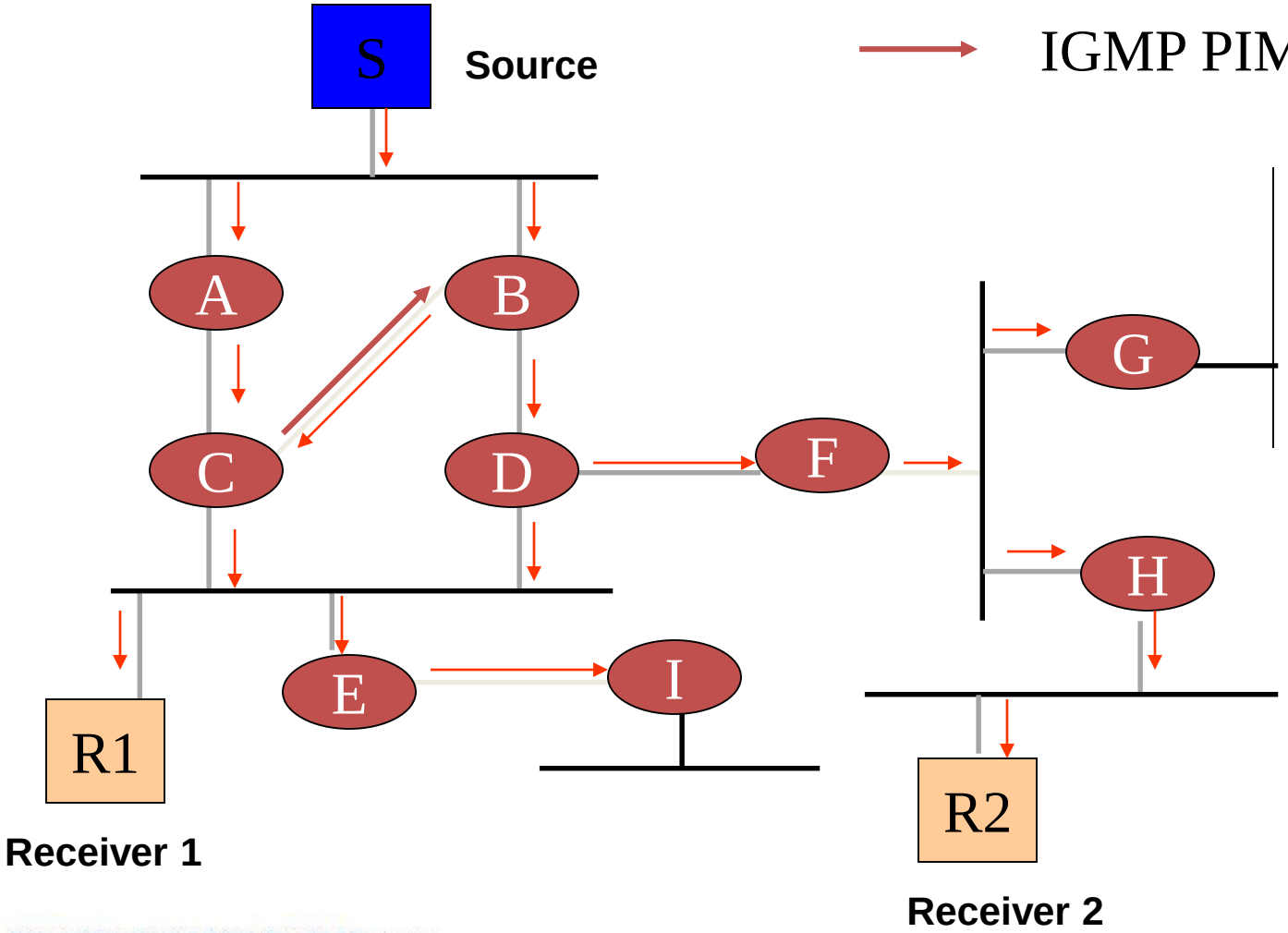
- Source tree created on demand based on RPF rule
- If the source goes inactive, the tree is torn down
- Branches that don't want data are pruned
- Grafts are used to join existing source tree

PIM-DM - Initial flood of data



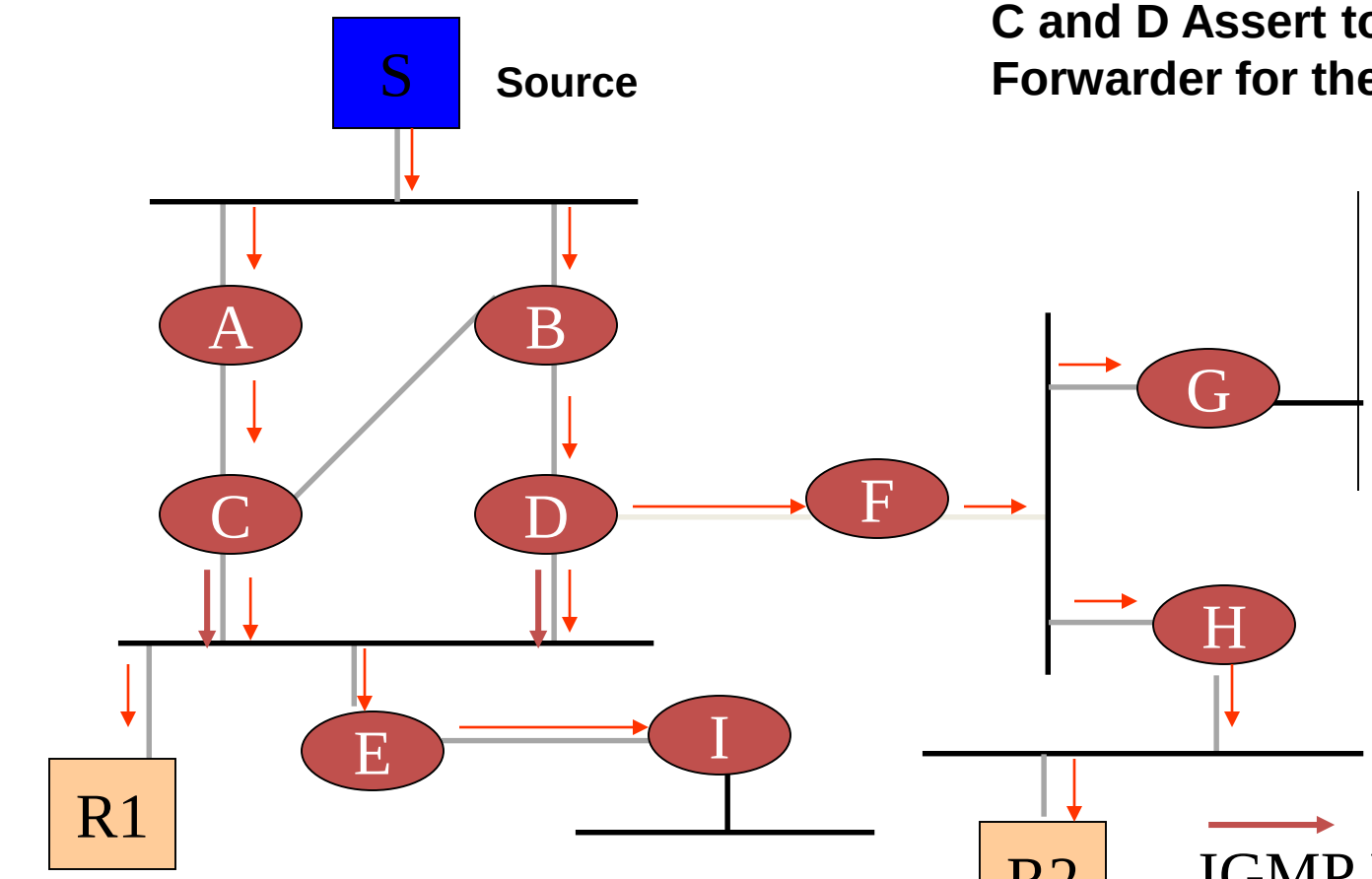
* Taken from J. Murai, Wisconsin

PIM-DM - Prune non-RPF P2P link



* Taken from J. Murai, Wisconsin

PIM-DM

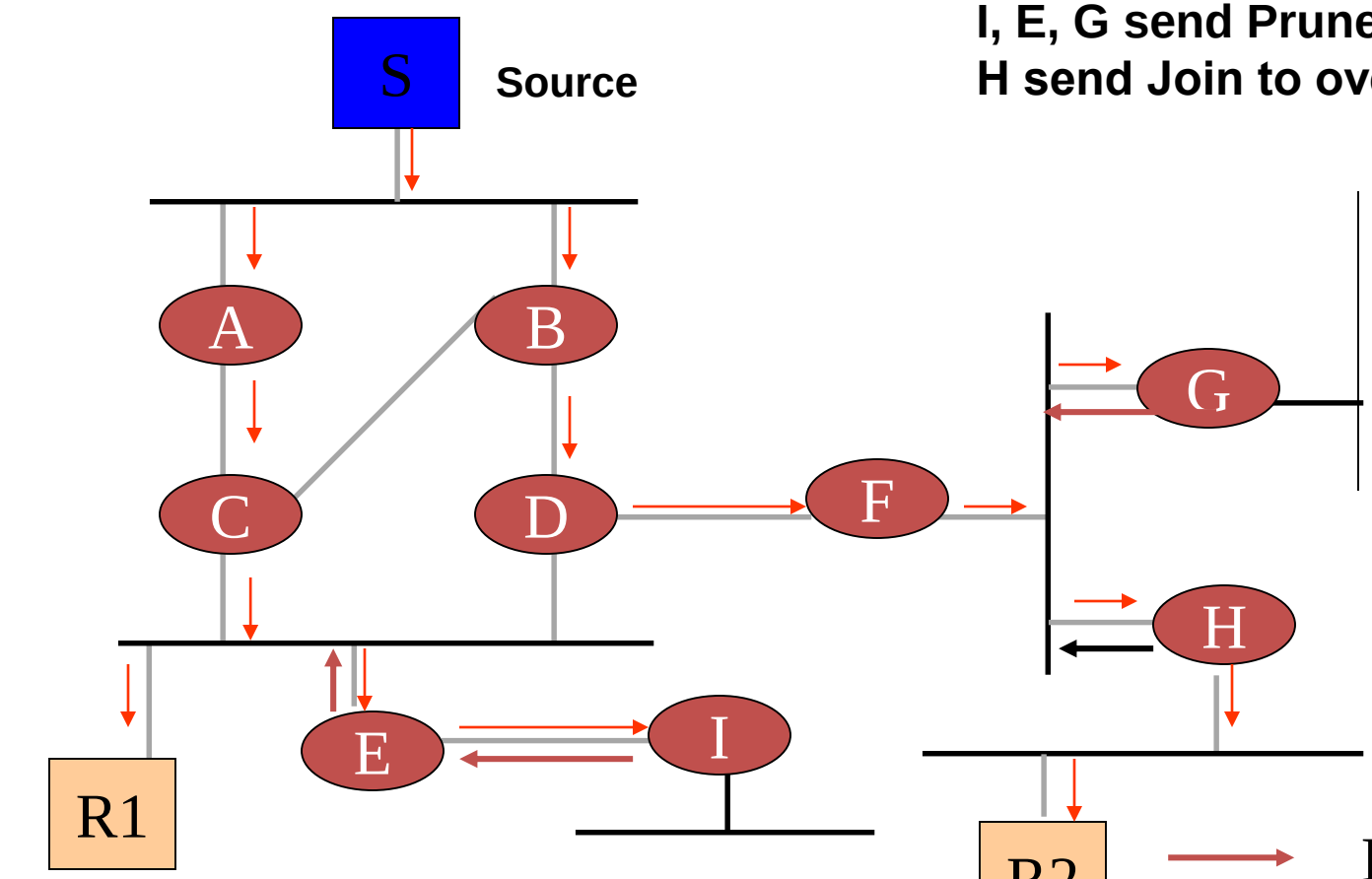


C and D Assert to Determine Forwarder for the LAN, C Wins

IGMP PIM-Assert with its own IP address

* Taken from J. Murai, Wisconsin

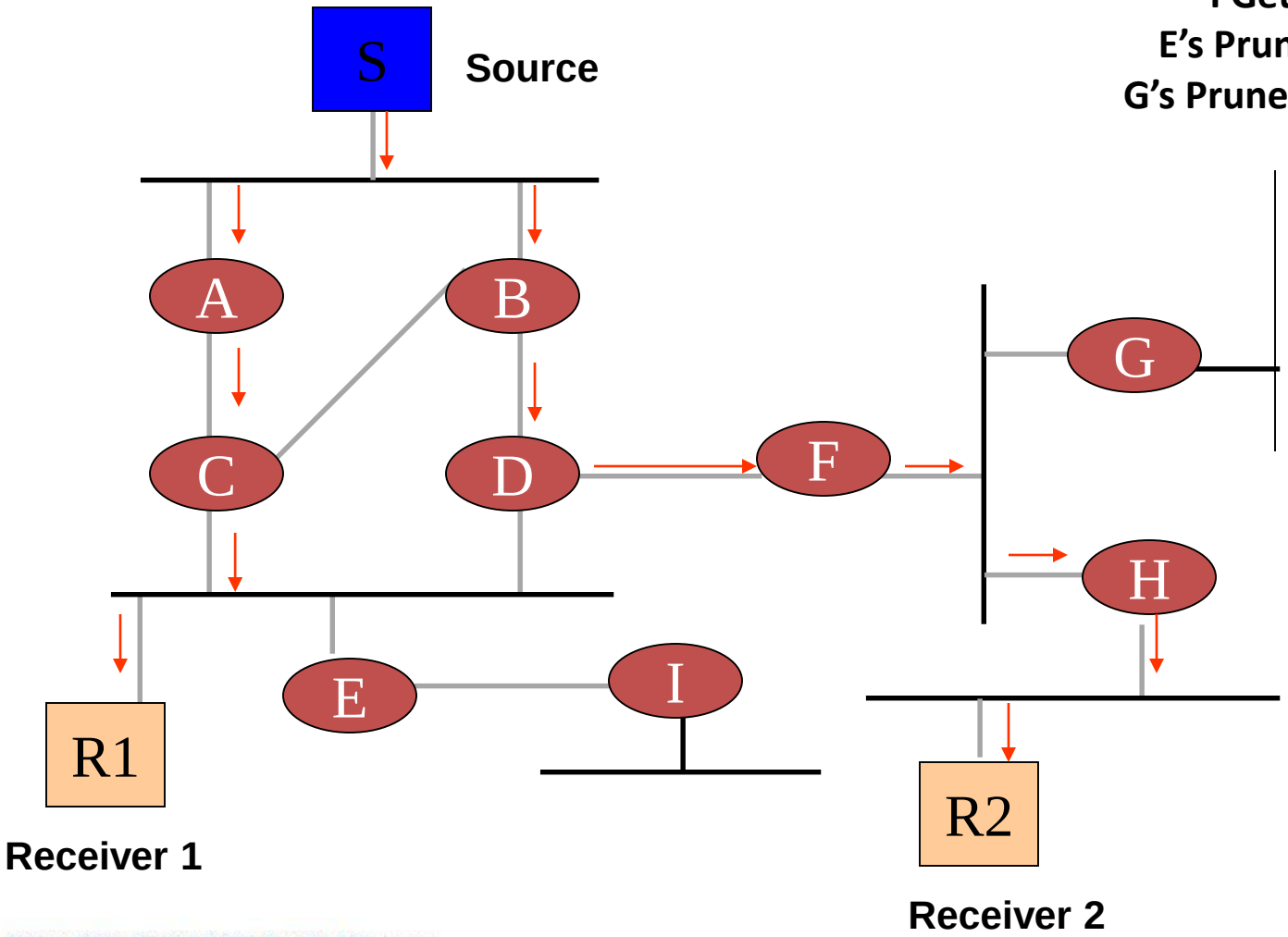
PIM-DM



I, E, G send Prune
H send Join to override G's Prune

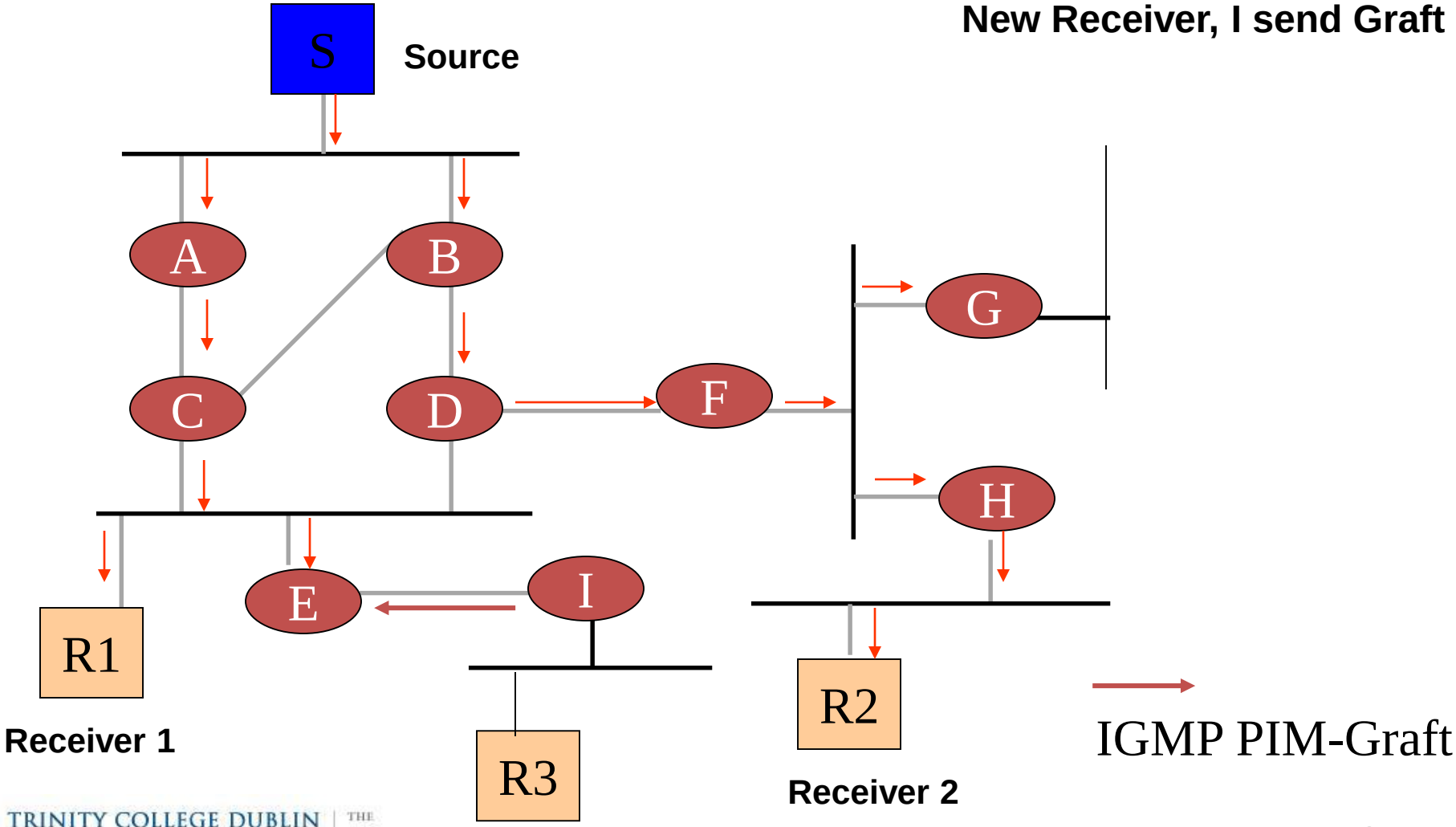
* Taken from J. Murai, Wisconsin

PIM-DM



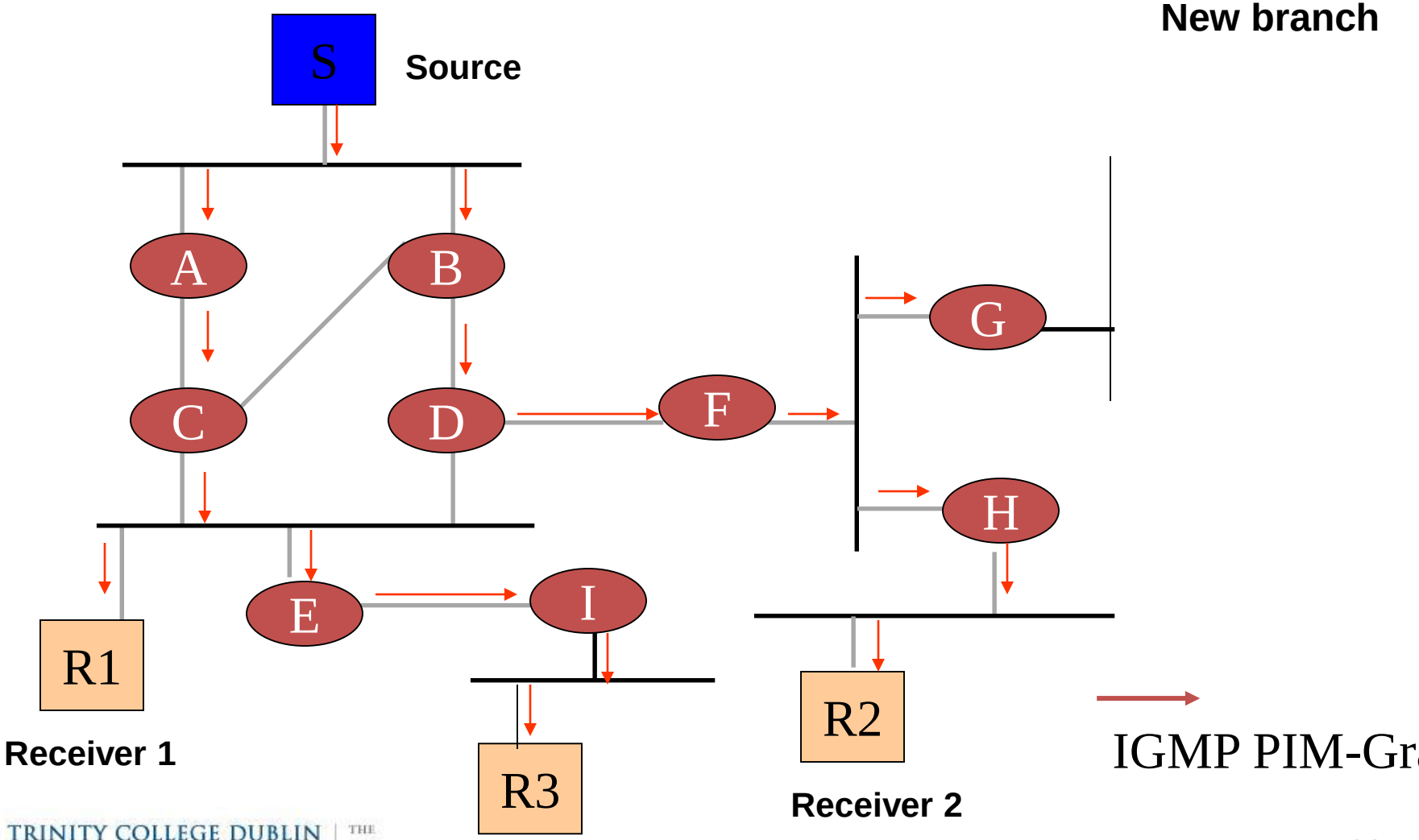
* Taken from J. Murai, Wisconsin

PIM-DM



* Taken from J. Murai, Wisconsin

PIM-DM



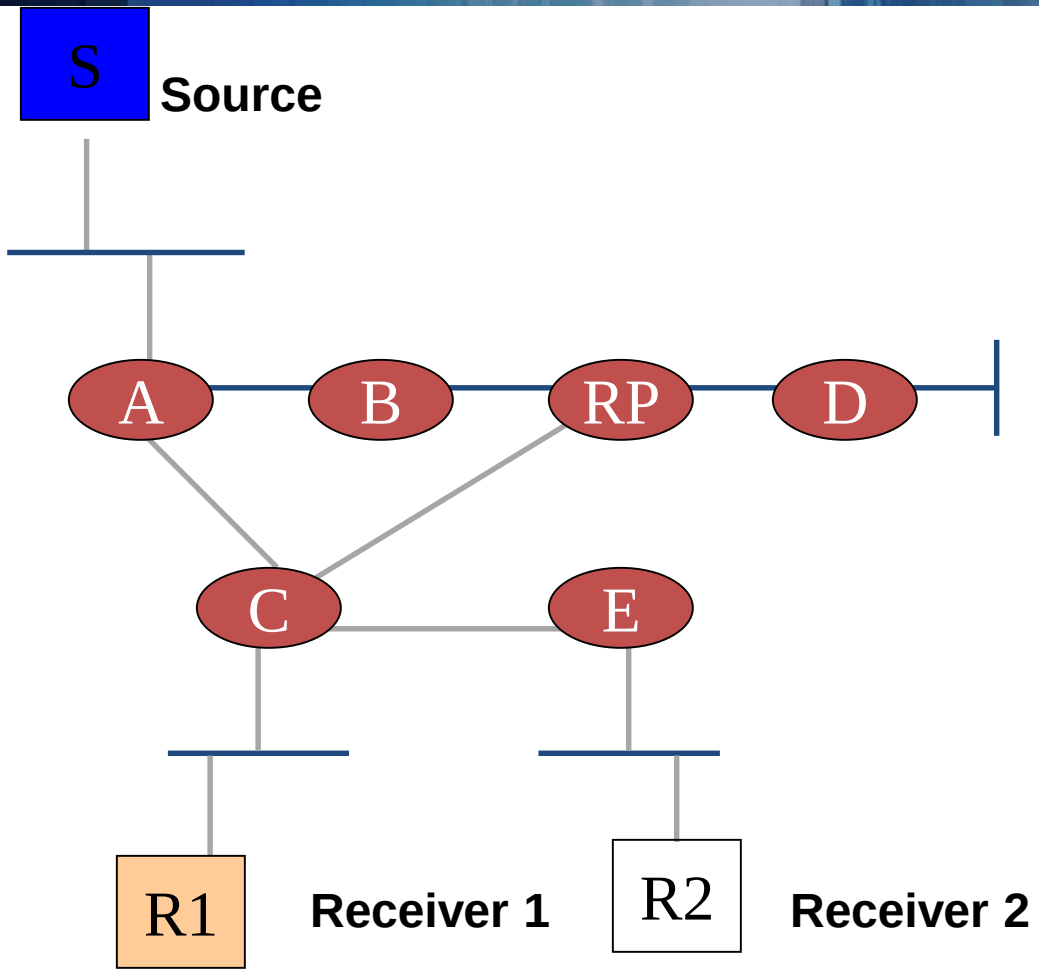
* Taken from J. Murai, Wisconsin

PIM – Sparse Mode (SM)

- When it is likely that many routers are involved in multicast routing

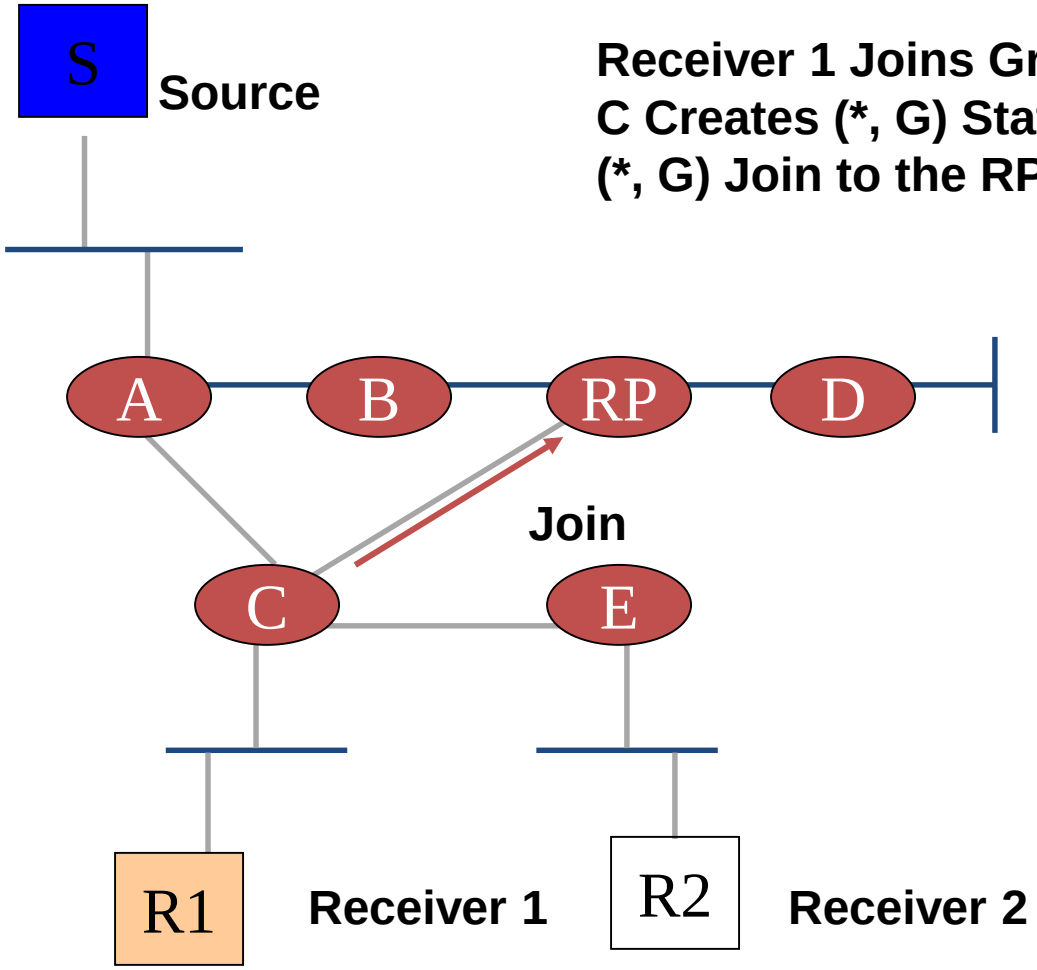
- One Rendez-Vous Point (RP) per group
- Explicit Join Model
  - Receivers send Join towards the RP
  - Sender Register with RP
  - Last hop routers can join source tree if the data rate warrants by sending joins to the source
- Dedicated “All-PIM-Routers” (224.0.0.13, ff02::d) multicast group

PIM-SM



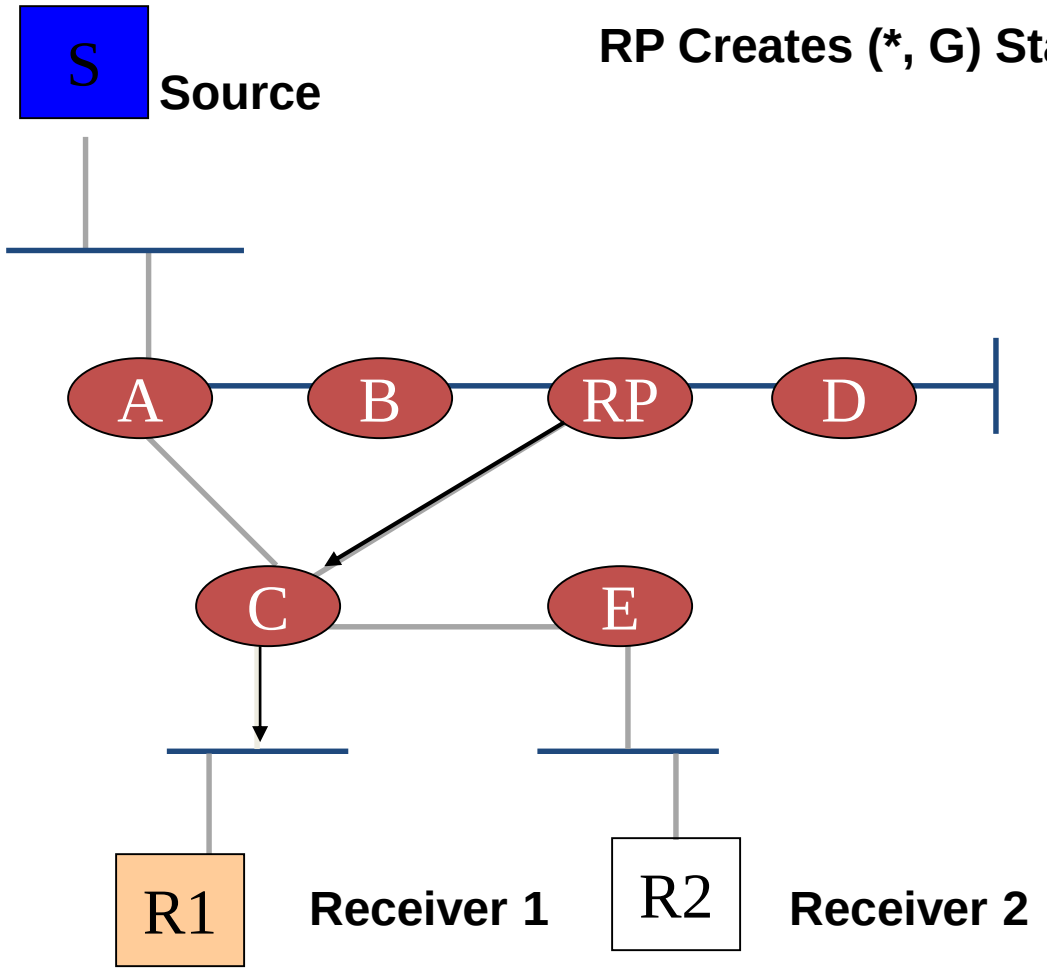
* Taken from J. Murai, Wisconsin

PIM-SM



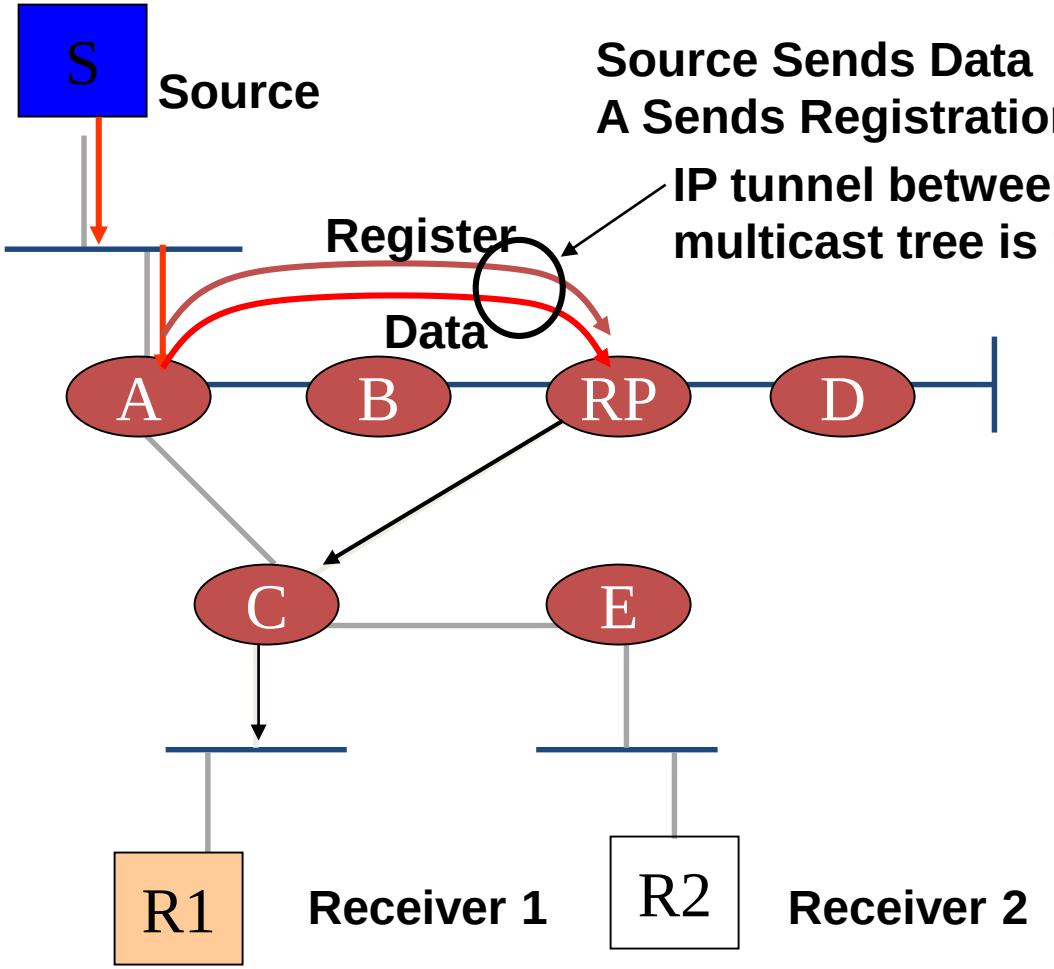
* Taken from J. Murai, Wisconsin

PIM-SM



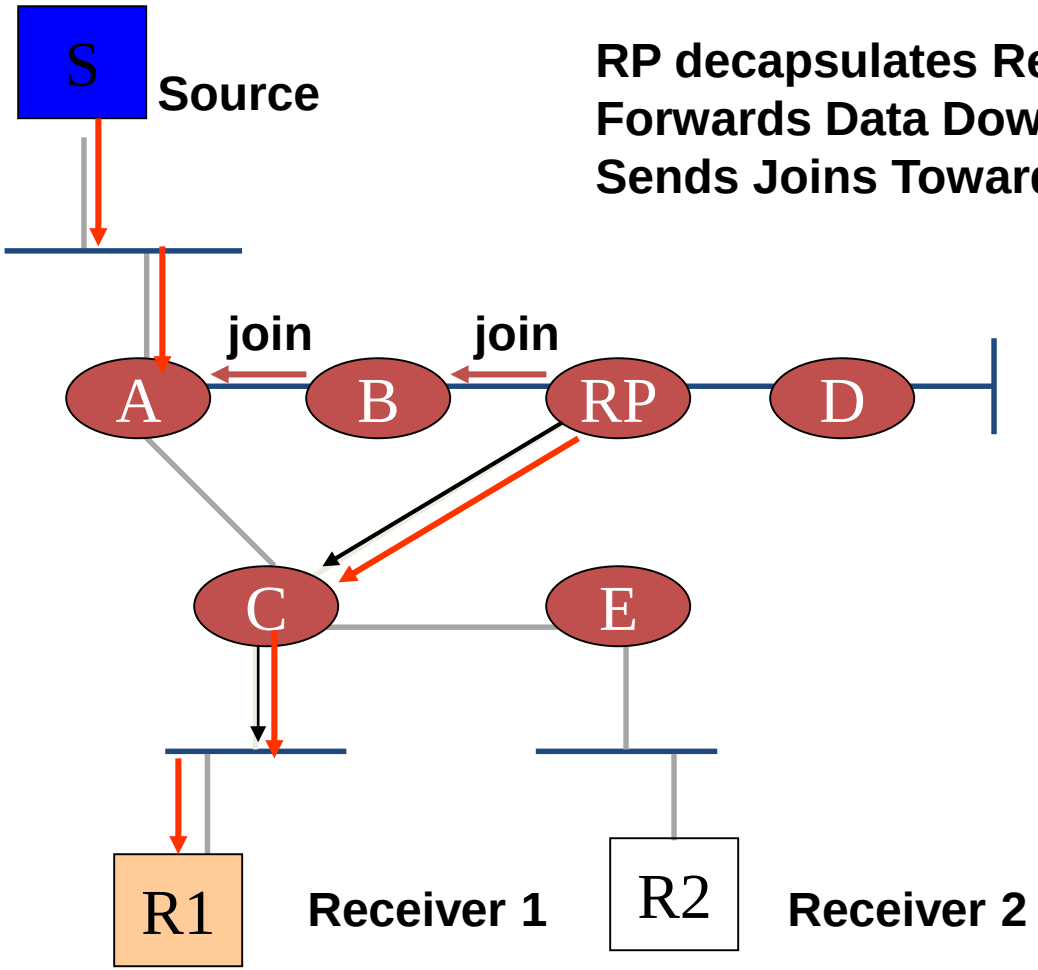
* Taken from J. Murai, Wisconsin

PIM-SM



* Taken from J. Murai, Wisconsin

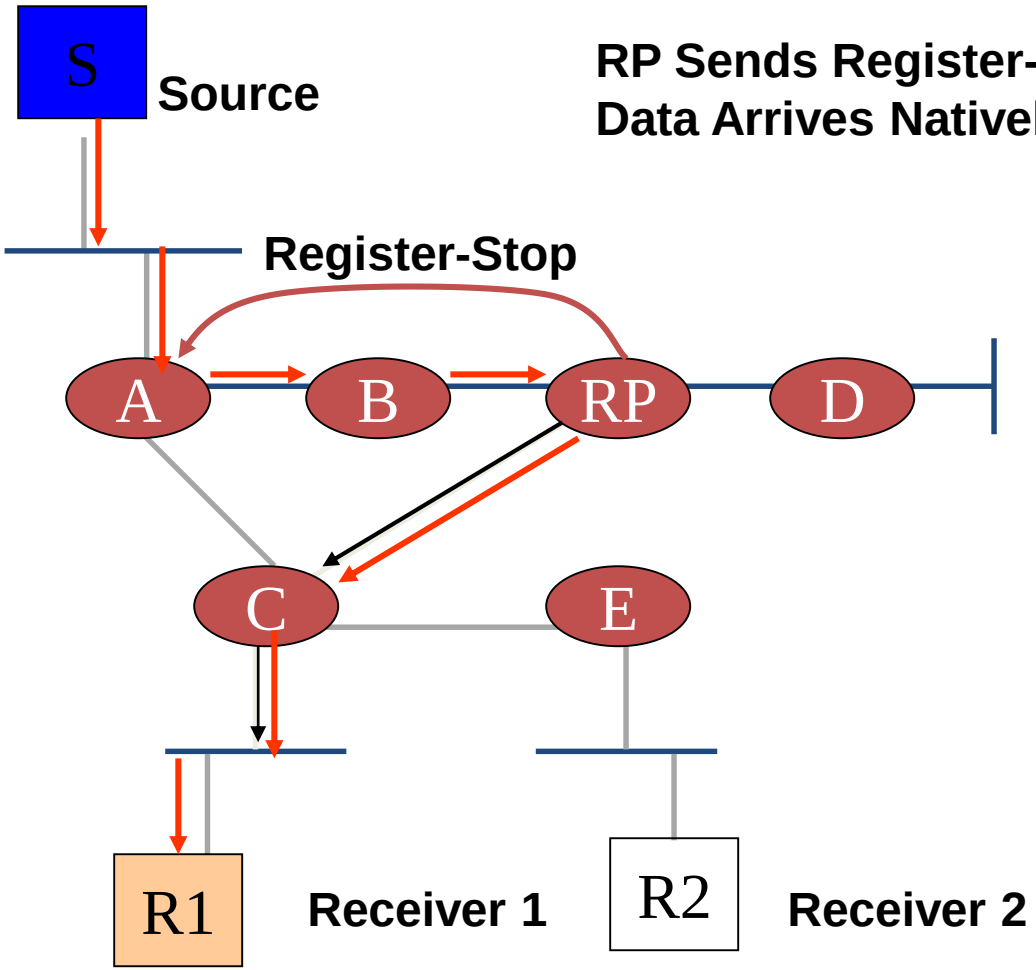
PIM-SM



RP decapsulates Registration
Forwards Data Down the Shared Tree
Sends Joins Towards the Source

* Taken from J. Murai, Wisconsin

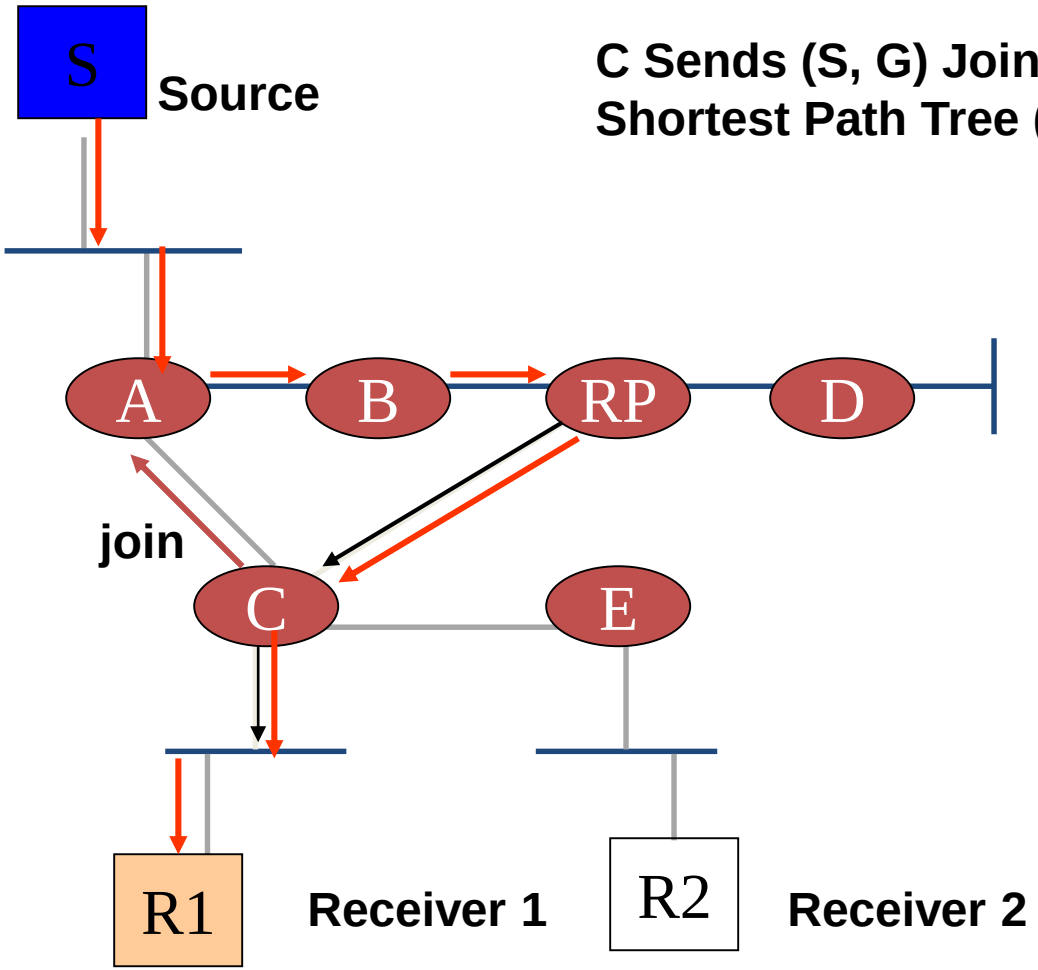
PIM-SM



* Taken from J. Murai, Wisconsin

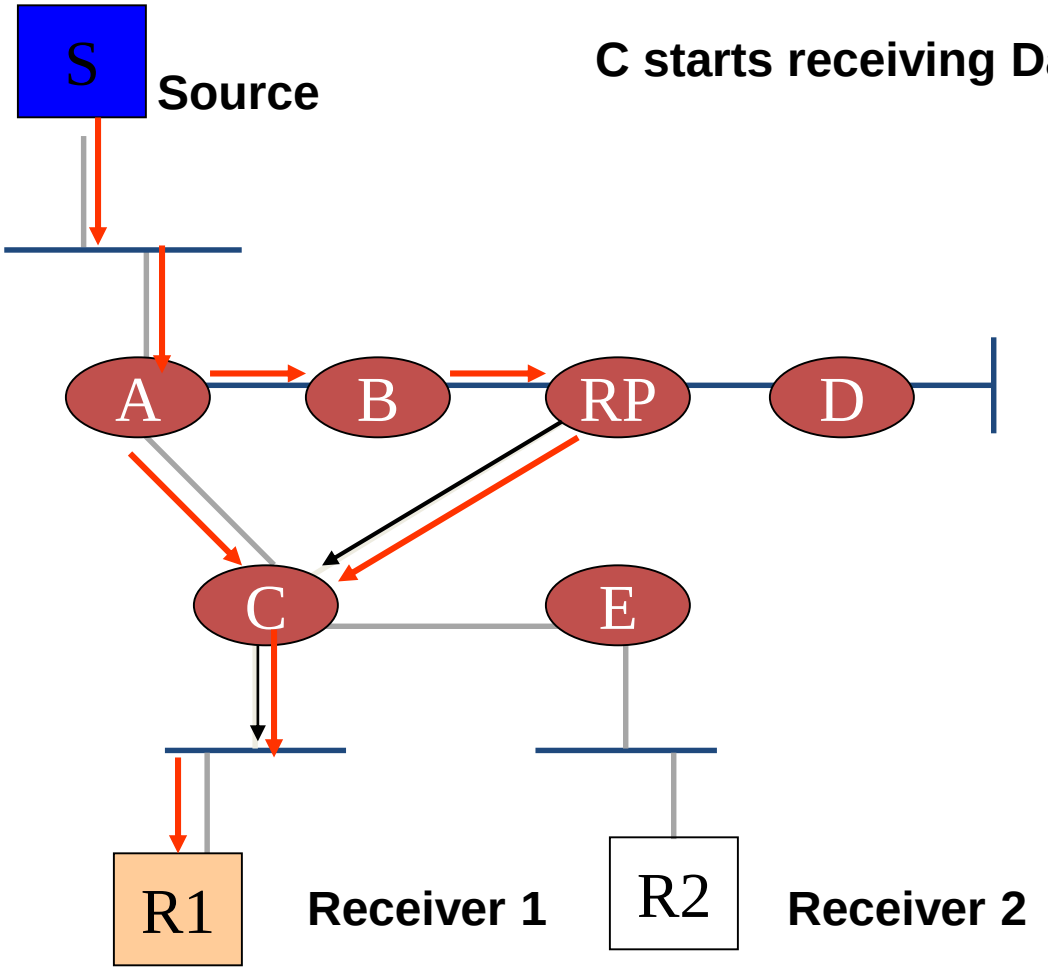
SPT Switchover

PIM-SM



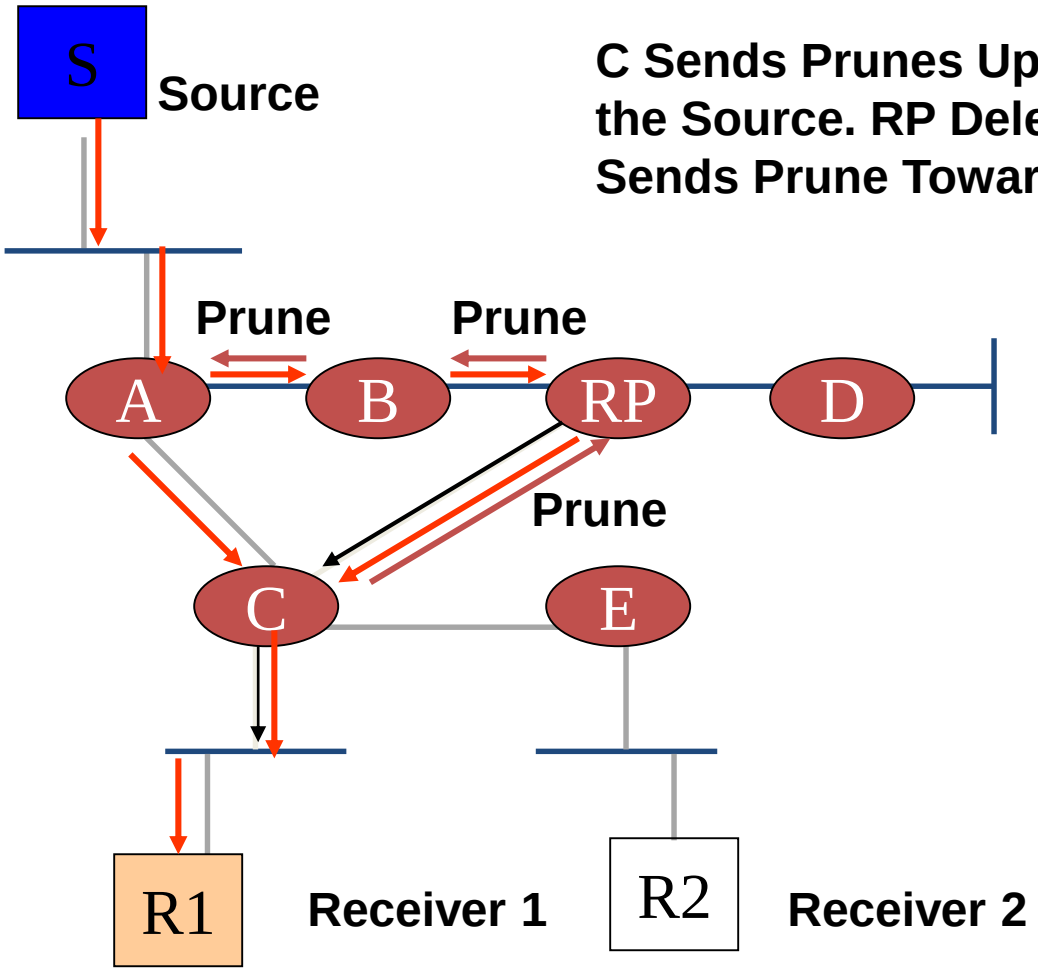
* Taken from J. Murai, Wisconsin

PIM-SM



* Taken from J. Murai, Wisconsin

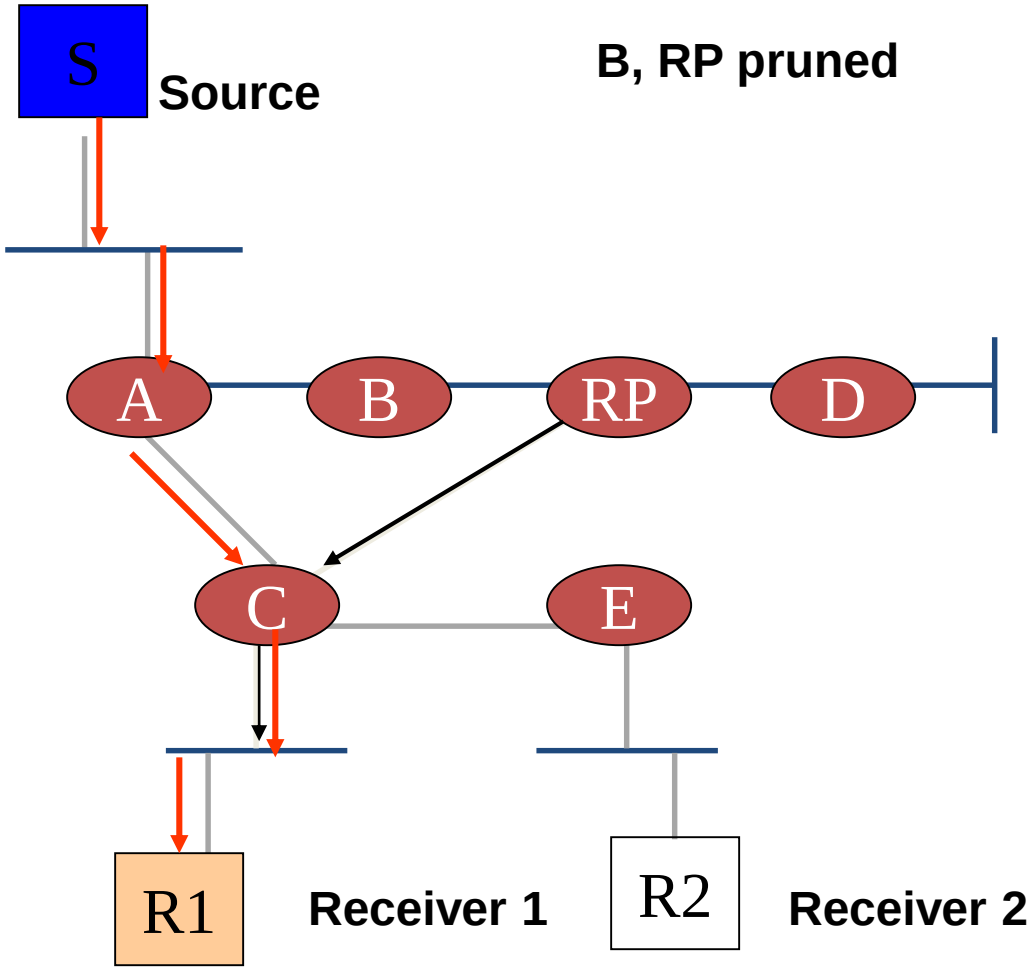
PIM-SM



C Sends Prunes Up the RP tree for the Source. RP Deletes (S, G) OIF and Sends Prune Towards the Source

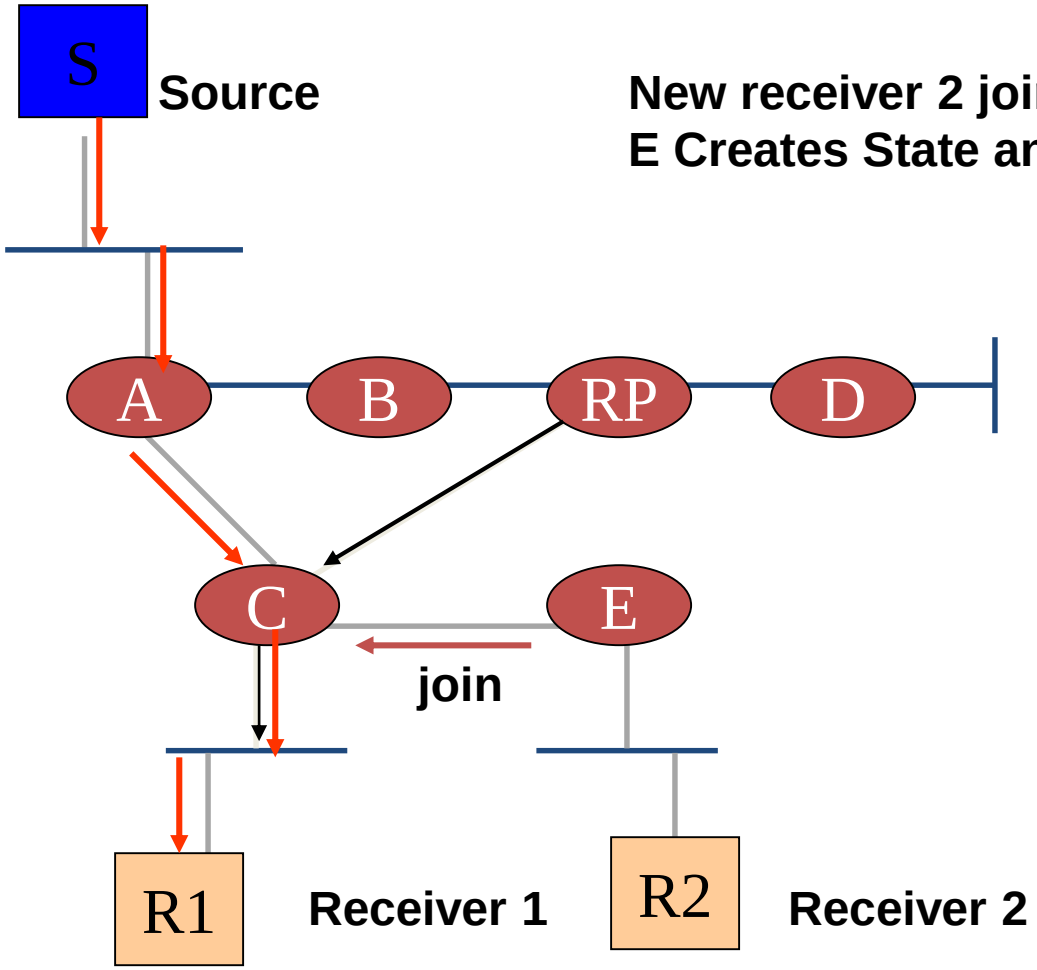
* Taken from J. Murai, Wisconsin

PIM-SM



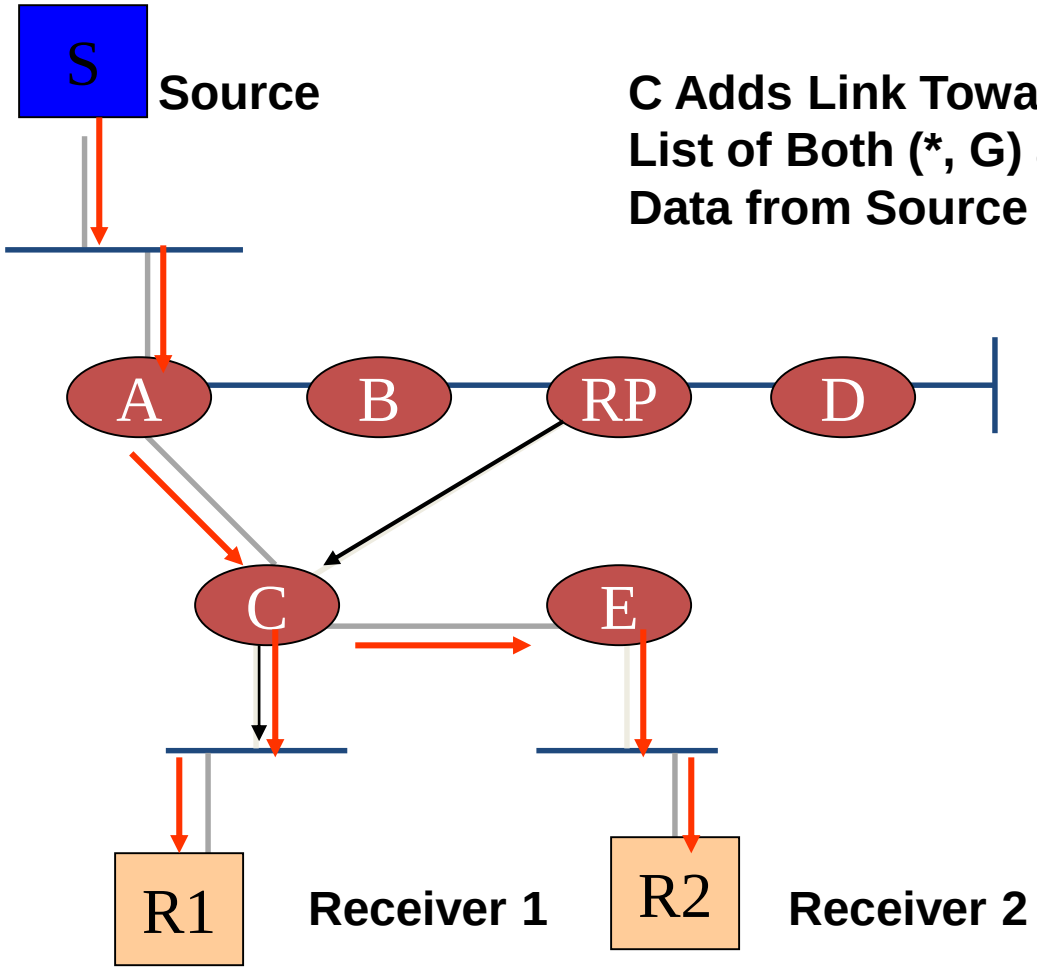
* Taken from J. Murai, Wisconsin

PIM-SM



* Taken from J. Murai, Wisconsin

PIM-SM



C Adds Link Towards E to the OIF List of Both (*, G) and (S, G)
Data from Source Arrives at E

* Taken from J. Murai, Wisconsin

Summary: Multicast Routing

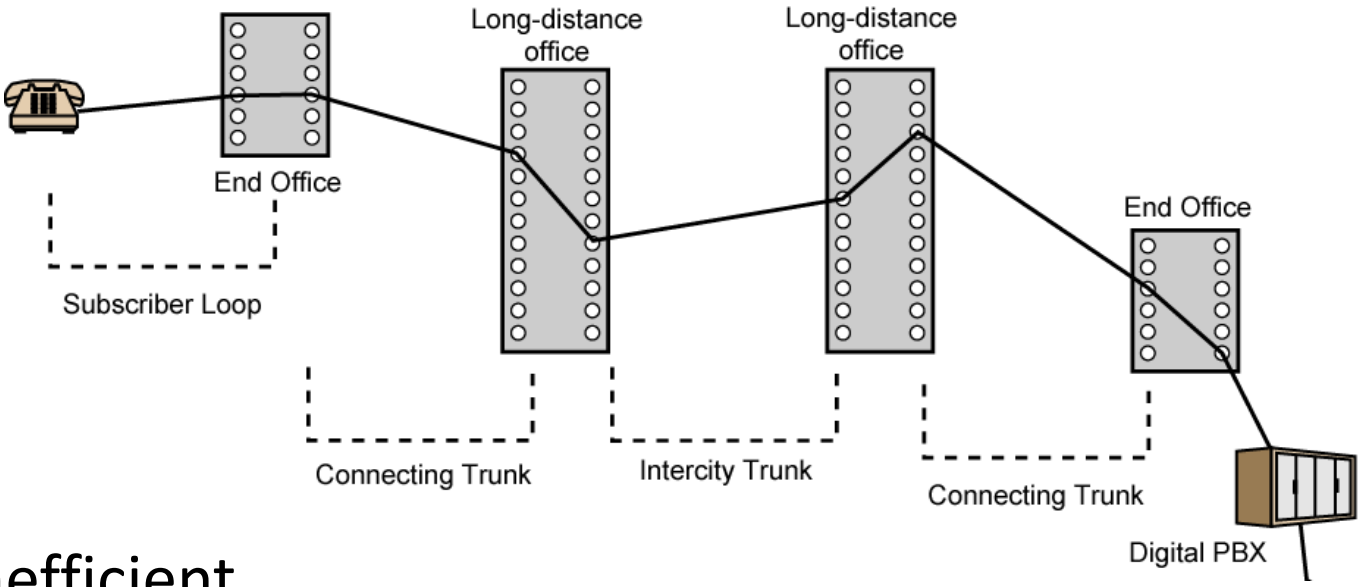
- Internet Group Management Protocol (IGMP)
  - Join&leave messages from hosts to routers
- Most protocols based on source trees
  - Reverse-Path Forwarding/Broadcast
  - Prune – remove subtree from tree
  - Graft – join subtree to tree
- Protocol Independent Multicast (PIM)
  - Dense Mode (DM)
  - Sparse Mode (SM)


That's all folks
Telecomms II

CS2031
Telecommunications II

Circuit Switching

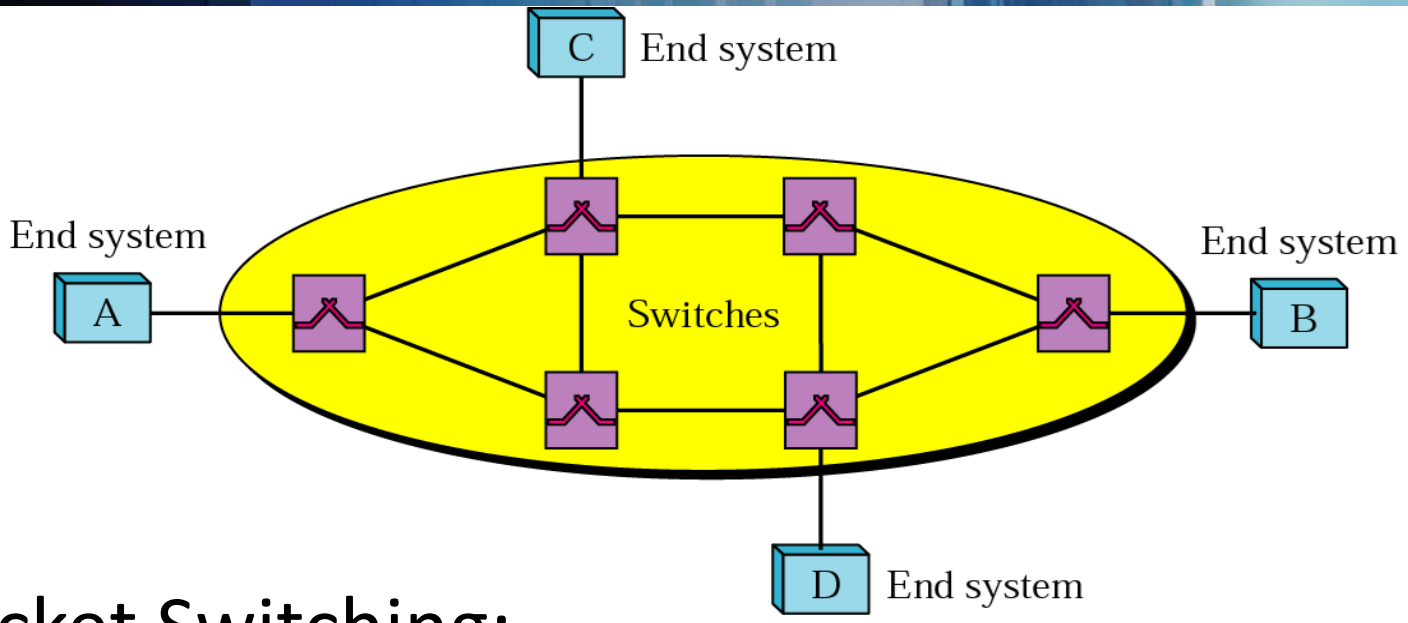
Public Circuit Switched Network



- Inefficient
  - Channel capacity dedicated for duration of connection
  - If no data, capacity wasted
- Set up of connection takes time
- Once connected, transfer is transparent

* Figure is courtesy of A. Tanenbaum

Switched Networks

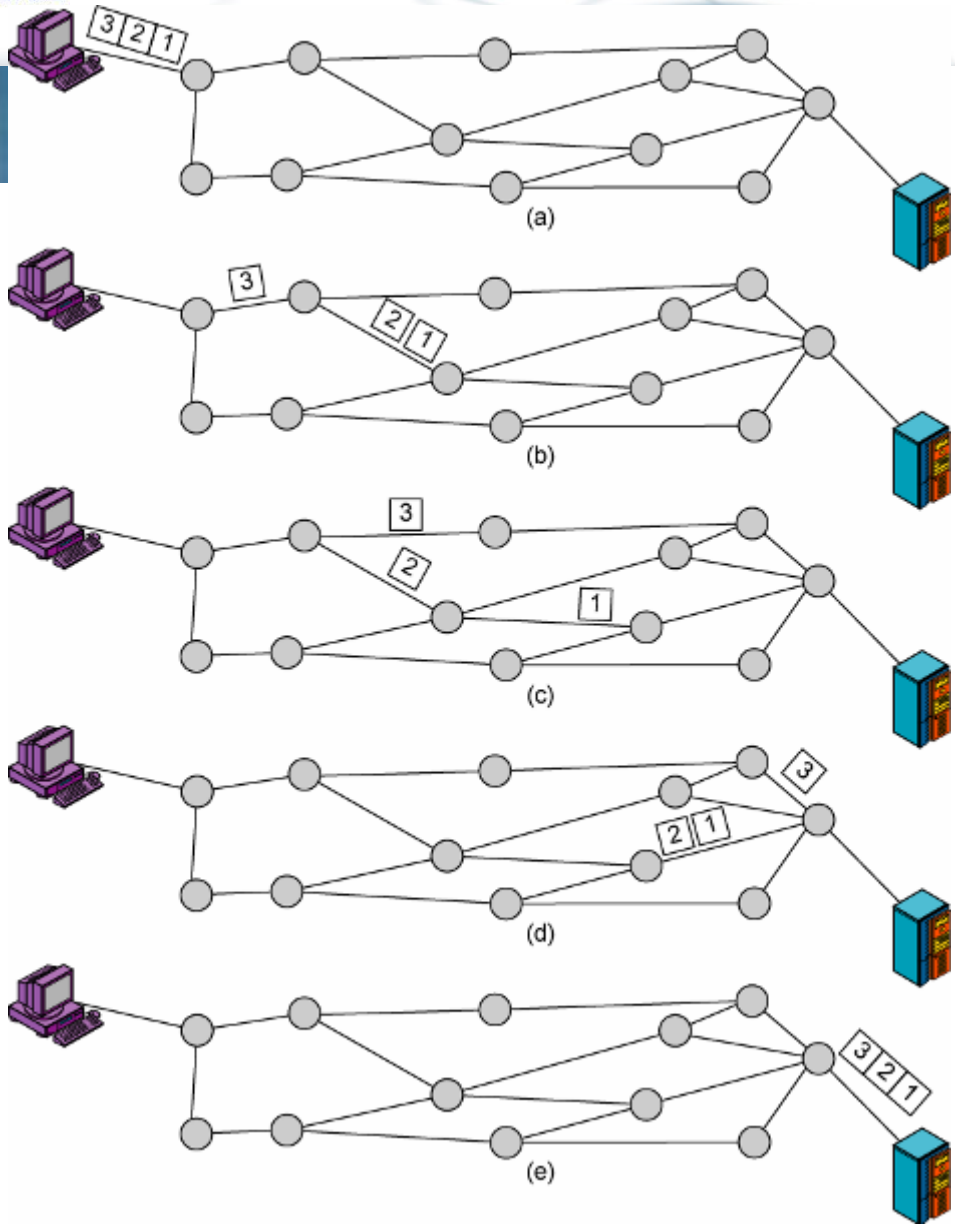


- Packet Switching:
  - Switching decisions are made on individual packets
- Virtual Circuit Switching:
  - A circuit is setup explicitly for individual connections

* Figure is courtesy of B. Forouzan

Packet switching

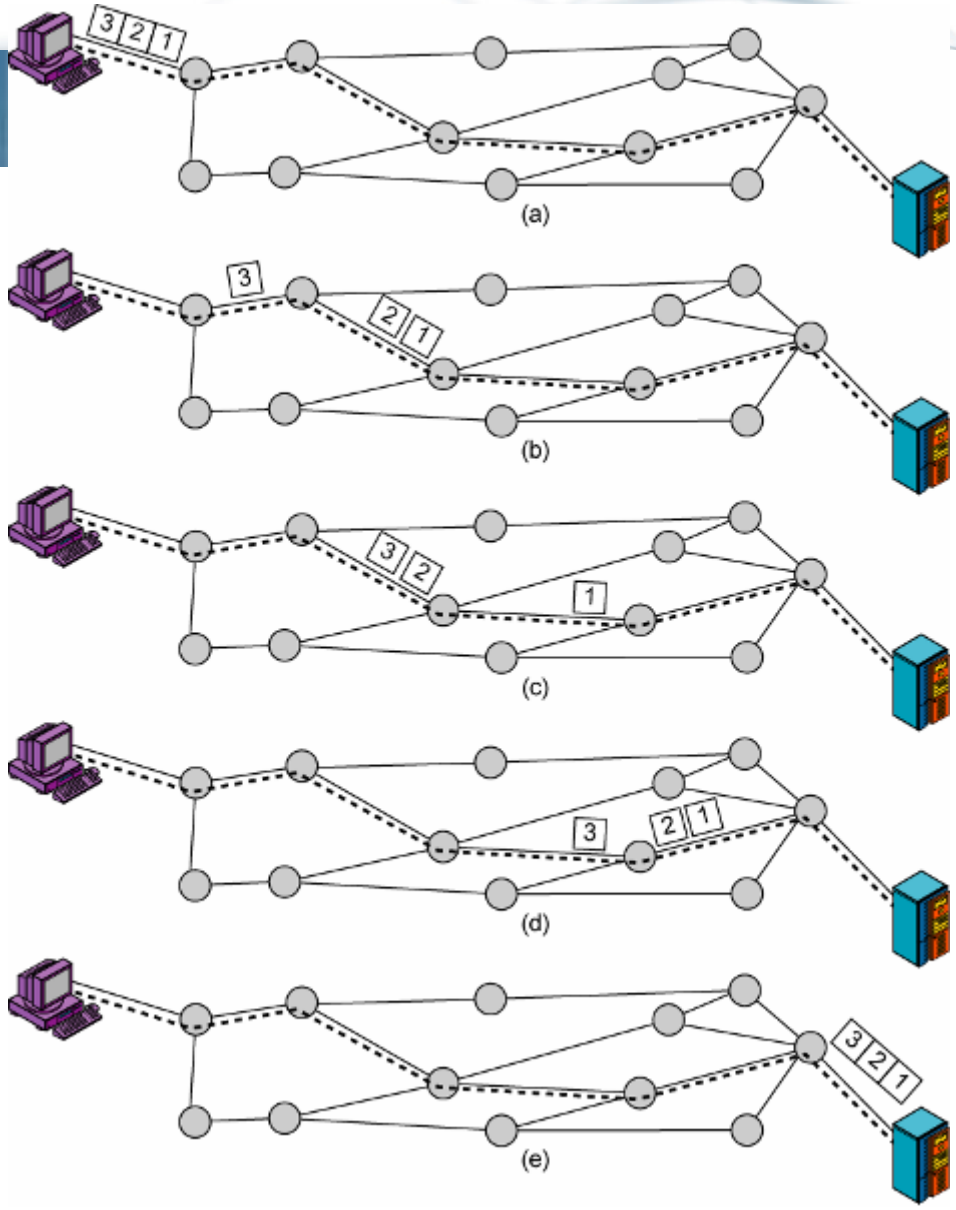
- Frames can be transferred over different paths in the network
- Reliability is generally delegated to higher layers
- Order is not necessarily maintained



* Figure is courtesy of B. Forouzan

Virtual Circuits

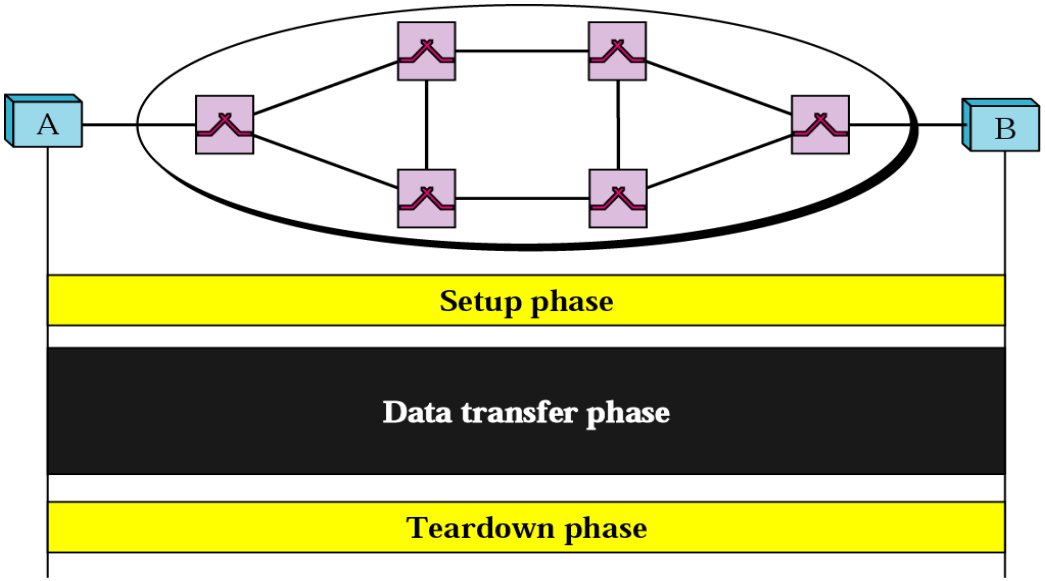
- Connection-oriented communication
- Connection is established before communication
- The network maintains order



* Figure is courtesy of B. Forouzan

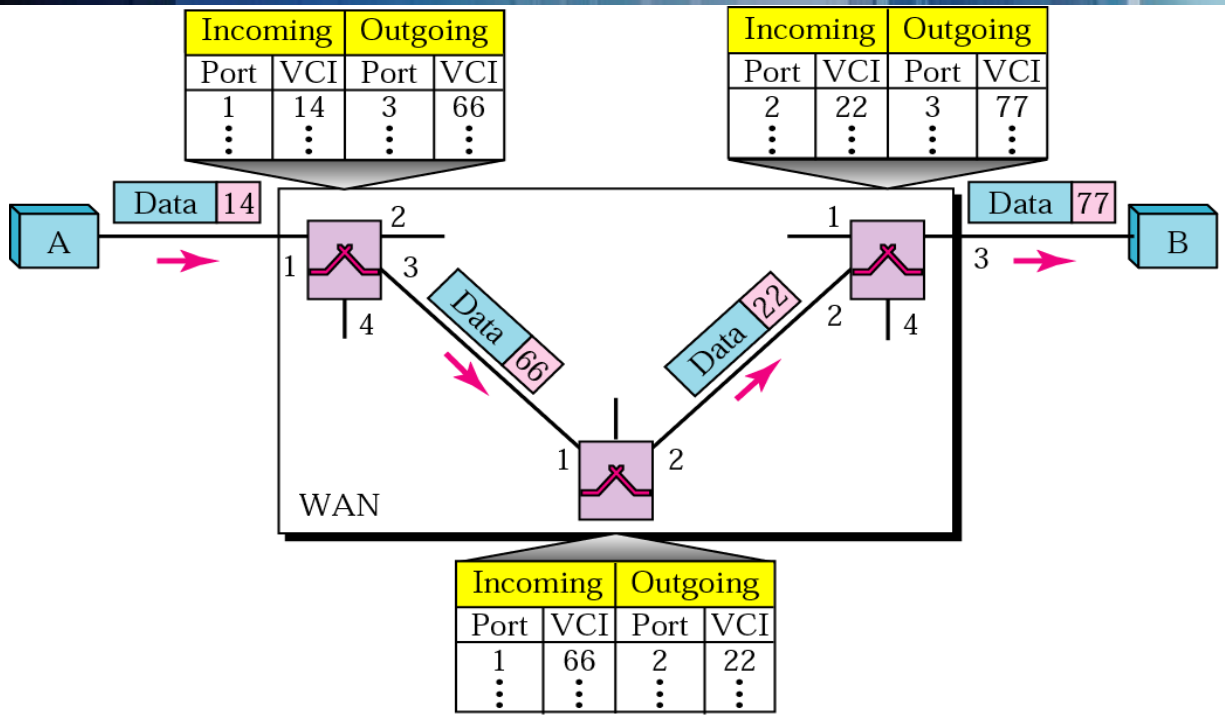
Phases for Virtual Circuit Communication

- Three phases
  - Connection setup
  - Data Transfer
  - Connection termination



* Figure is courtesy of B. Forouzan

Virtual Circuit Switching

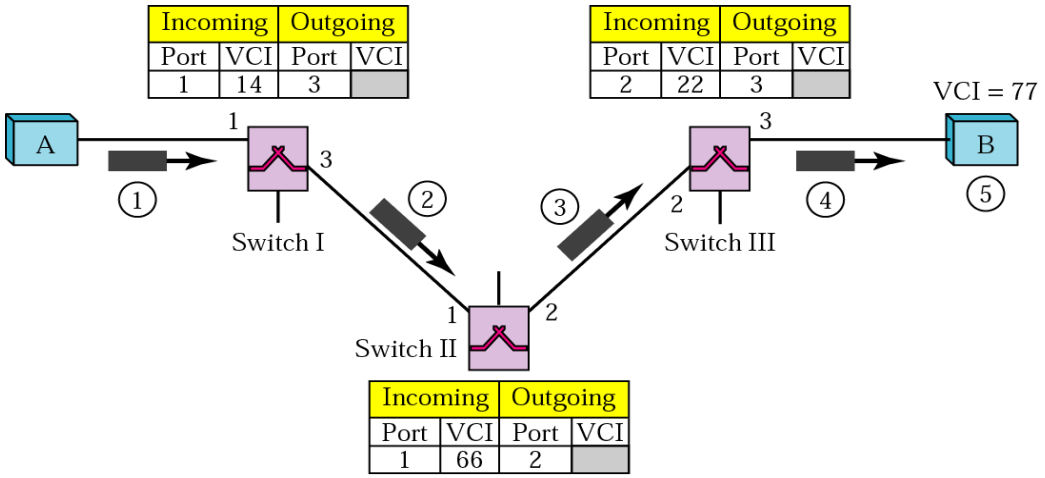


- Every switch maintains a table
  - For duration of communication one entry for incoming and outgoing line
  - Incoming and outgoing line are identified by port number and virtual circuit identifier

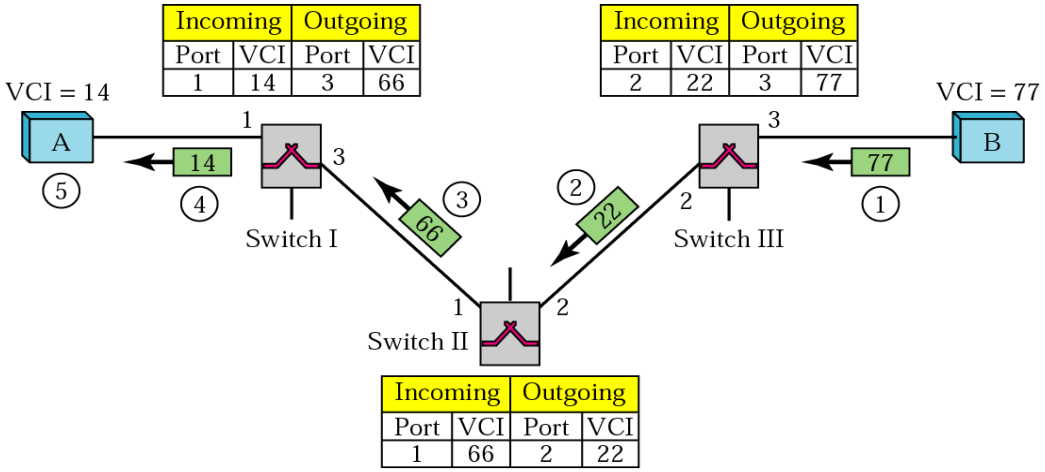
* Figure is courtesy of B. Forouzan

Setup Phase

- Setup request

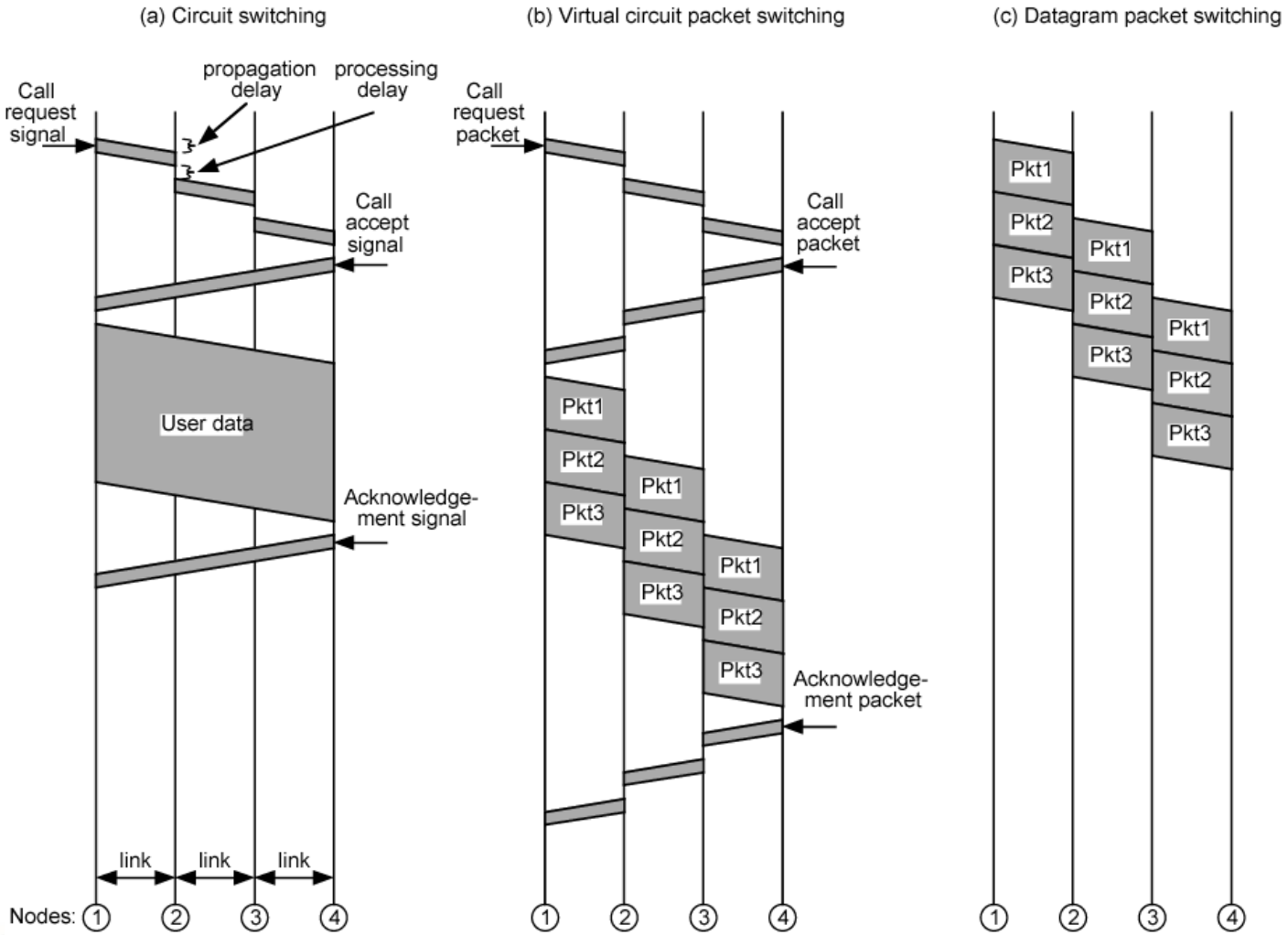


- Setup acknowl.



* Figure is courtesy of B. Forouzan

Event Timing

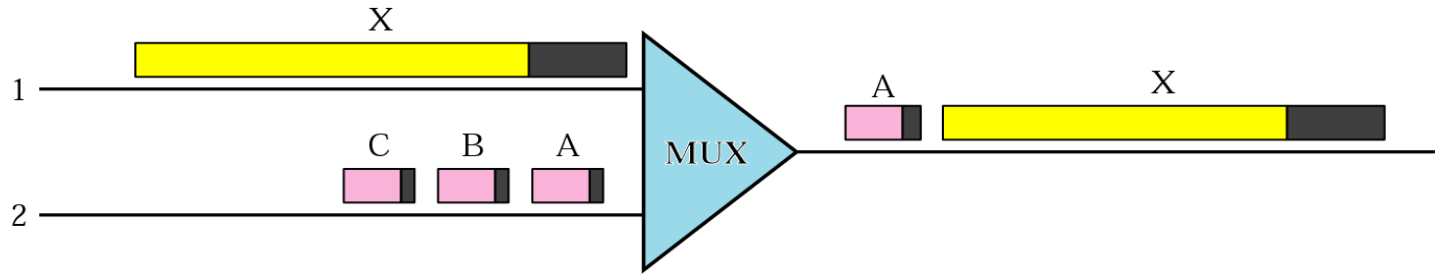


* Figure is courtesy of W. Stallings

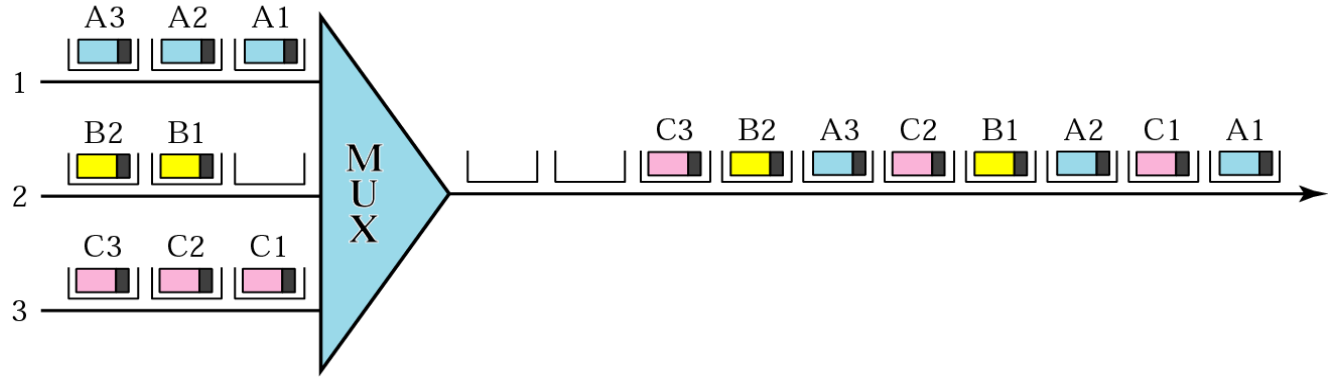
Asynchronous Transfer Mode (ATM)

- Example of virtual circuit switching
  - Cell-Switching
- Similarities between ATM and packet switching
  - Transfer of data in discrete chunks
  - Multiple logical connections over single physical interface
- In ATM flow on each logical connection is in fixed sized packets called cells
- Minimal error and flow control
  - Reduced overhead

Motivation for ATM



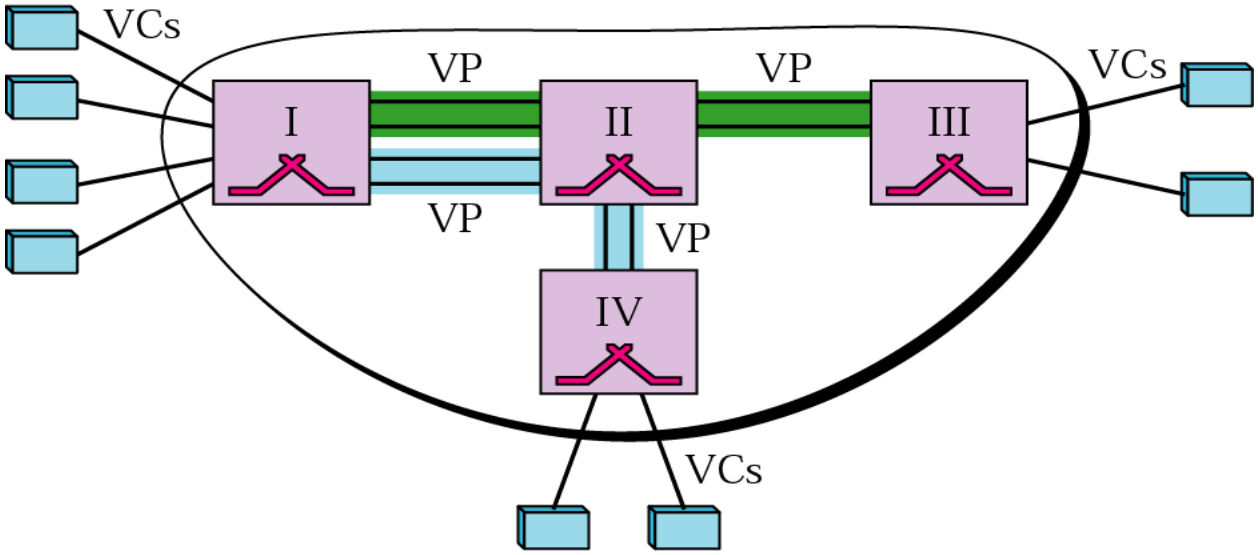
- Frames at a switch may be handled in any order and occupy switch for underspecified time



- Small, fixed-size frames allow simple, fast switches

* Figure is courtesy of B. Forouzan

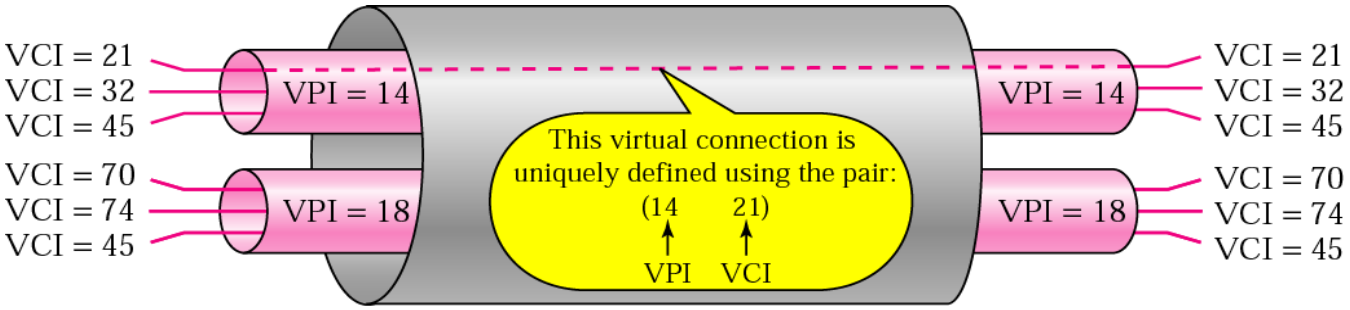
Virtual Circuits / Virtual Paths



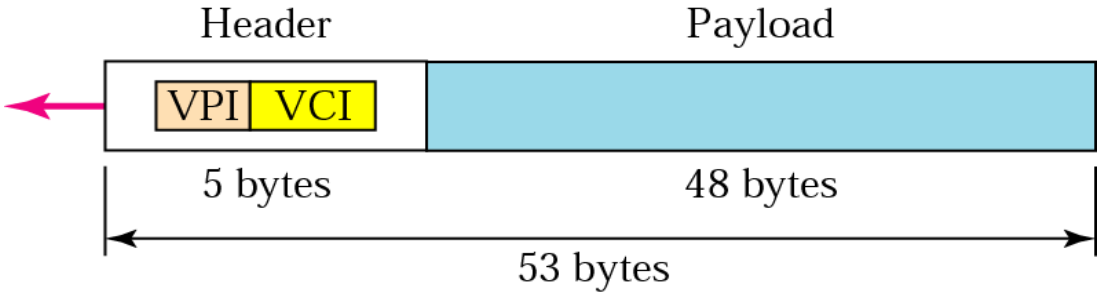
- Virtual circuits are collected into virtual paths

* Figure is courtesy of B. Forouzan

ATM Packet



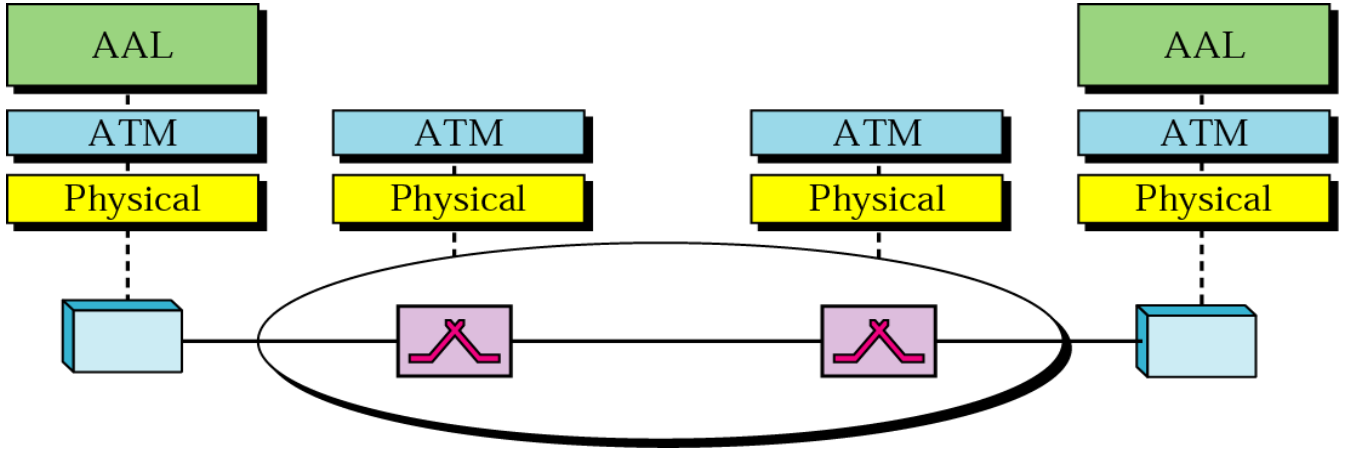
- Connection is specified by combination of Virtual Path ID and Virtual Circuit ID



- Every frame is exactly 53 bytes

* Figure is courtesy of B. Forouzan

Application Adaptation Layer (AAL)



- ATM defined a number of AALs for various purposes (each has its own header format):
  - AAL1: Constant bit rate e.g. multimedia
  - AAL2: Variable-data-rate
  - AAL3/4: Connection-oriented data services
    - Sequencing and Error Control
  - AAL5: Simple and efficient adaptation layer (SEAL)

* Figure is courtesy of B. Forouzan

ATM – It Didn't Happen

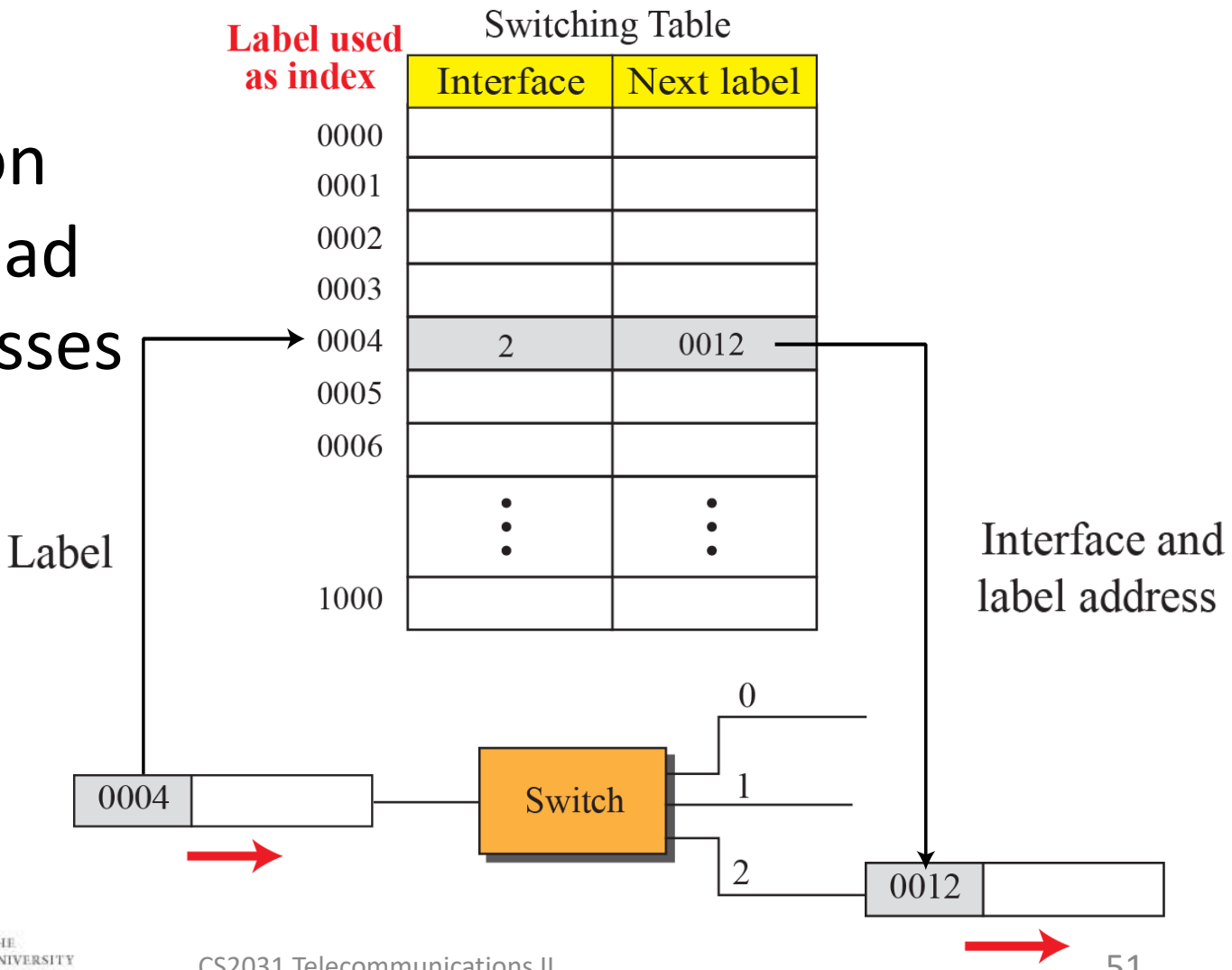
- From Tanenbaum:

  "ATM was going to solve all the world's networking and telecommunications problems by merging voice, data, cable television, telex, telegraph, carrier pigeons, …"

- It didn't happen:
  - Bad Timing
  - Technology
  - Implementation
  - Politics

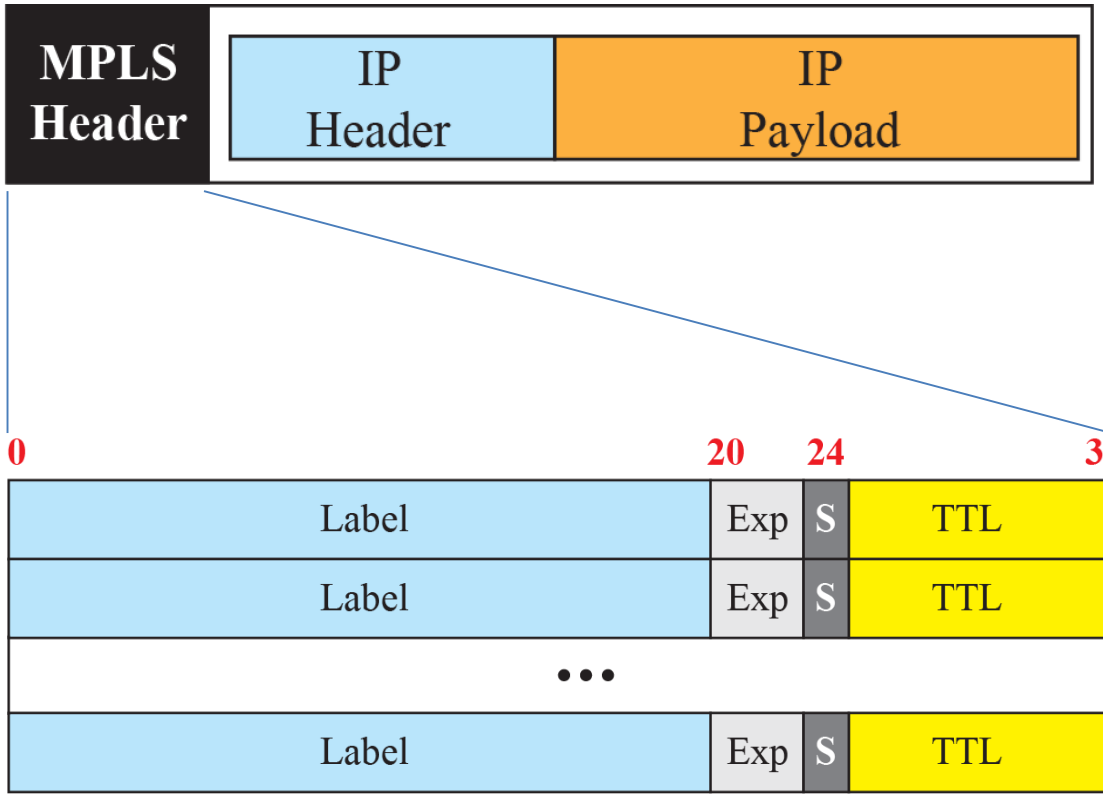
Multiprotocol Label Switching (MPLS)

- Enables switching on labels instead of IP addresses

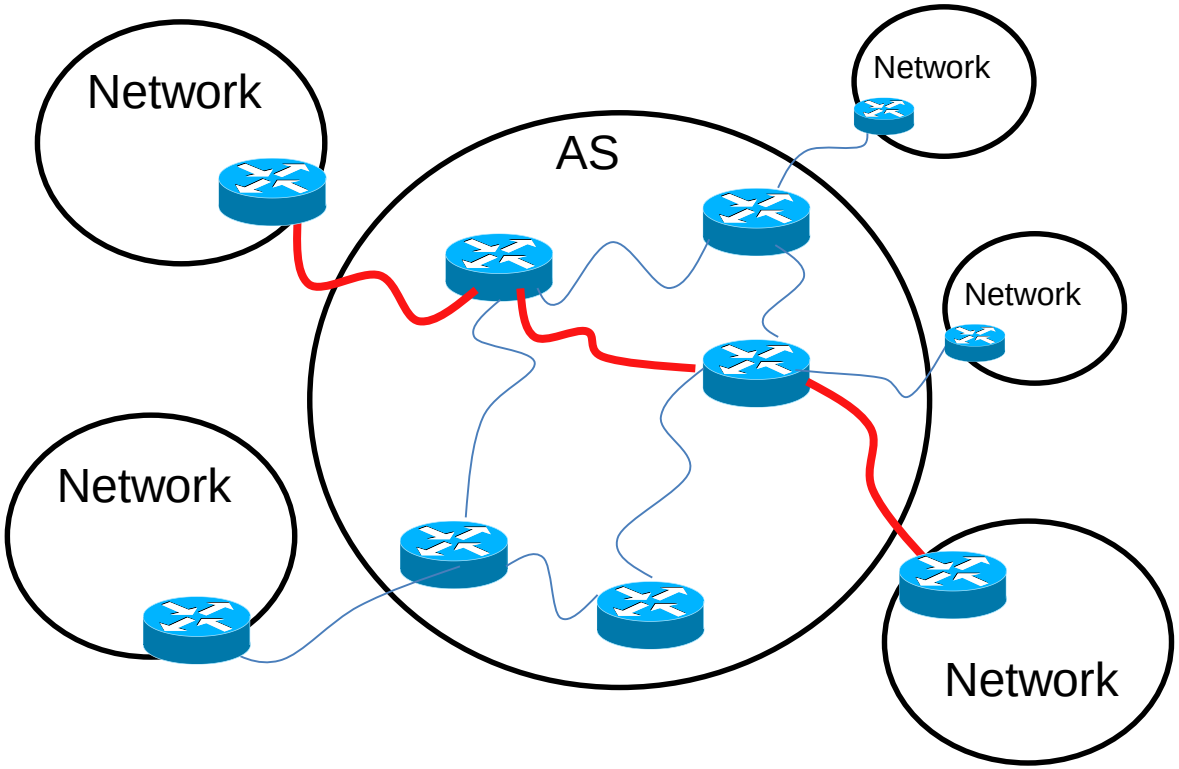


MPLS Header

- MPLS header as stack of labels

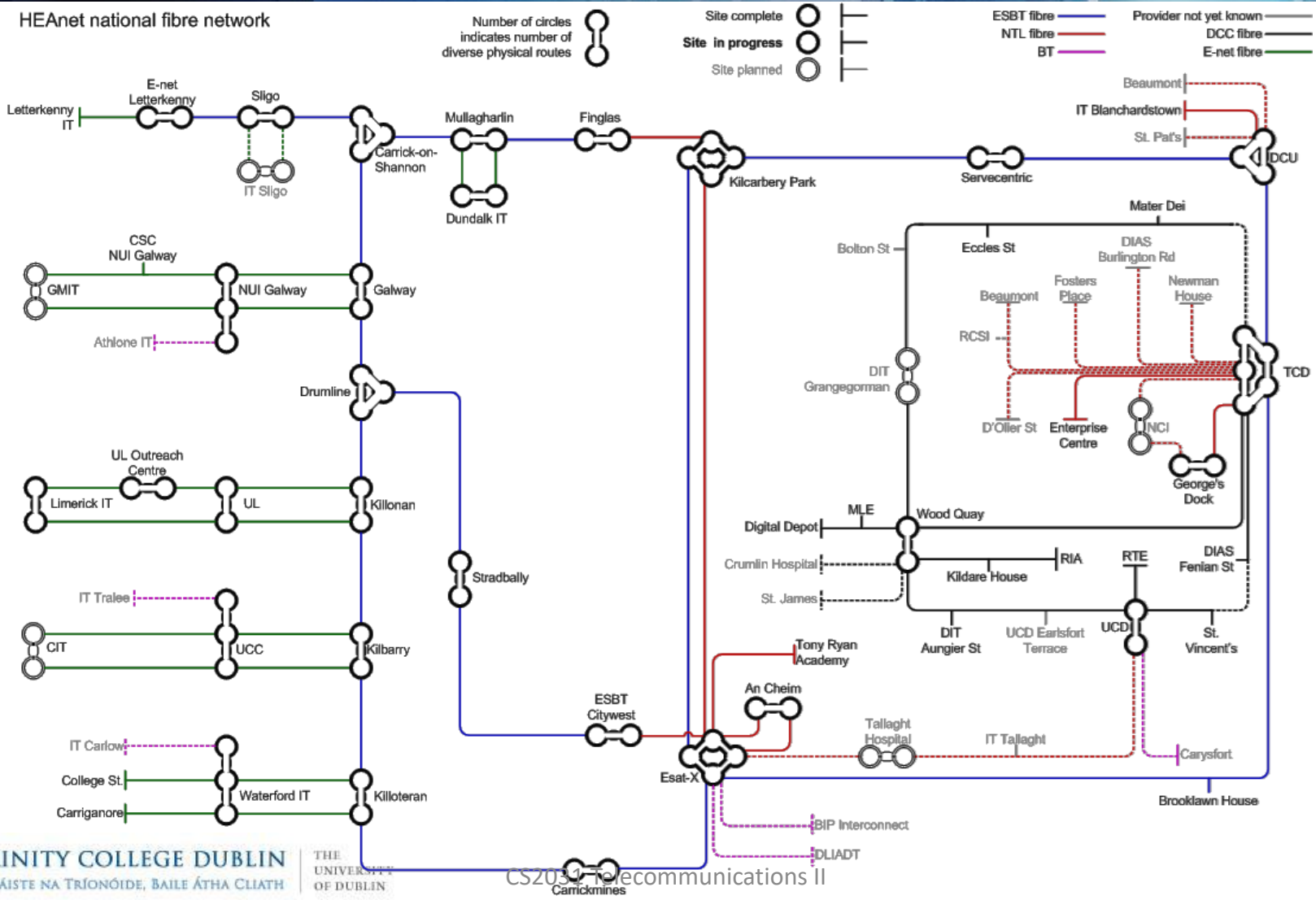


MPLS Use Case



- Creating a virtual network

HEAnet Fibre Network



Summary:Virtual Circuit Switching – ATM

- Virtual Circuit Switching
  - Preplanned route established before any frames sent
  - Call request and call accept frames establish connection (handshake)
  - Each frame contains a virtual circuit identifier instead of destination address
  - No routing decisions required for each frame
  - Clear request to drop circuit
  - Not a dedicated path
- Asynchronous Transfer Mode (ATM)
  - Example for virtual circuit switching
  - Cells consist of 5-byte header and 48-byte payload
  - Circuits identified by virtual circuit ID and virtual path ID
  - Application adaptation layer (AAL) for specific application areas


That's all folks
Telecomms II

Assignment Deadlines

- Extended until December 31st 2018
  - Applied to both assignments
  - Knowledge of assignment 1 helps with assignment 2
  - Thank your SU Convenor for that ☺

CS2031
Telecommunications II

Assignment 2

Openflow – Quick Intro

OpenFlow: Enabling Innovation in Campus Networks

Nick McKeown
Stanford University

Tom Anderson
University of Washington

Hari Balakrishnan
MIT

Guru Parulkar
Stanford University

Larry Peterson
Princeton University

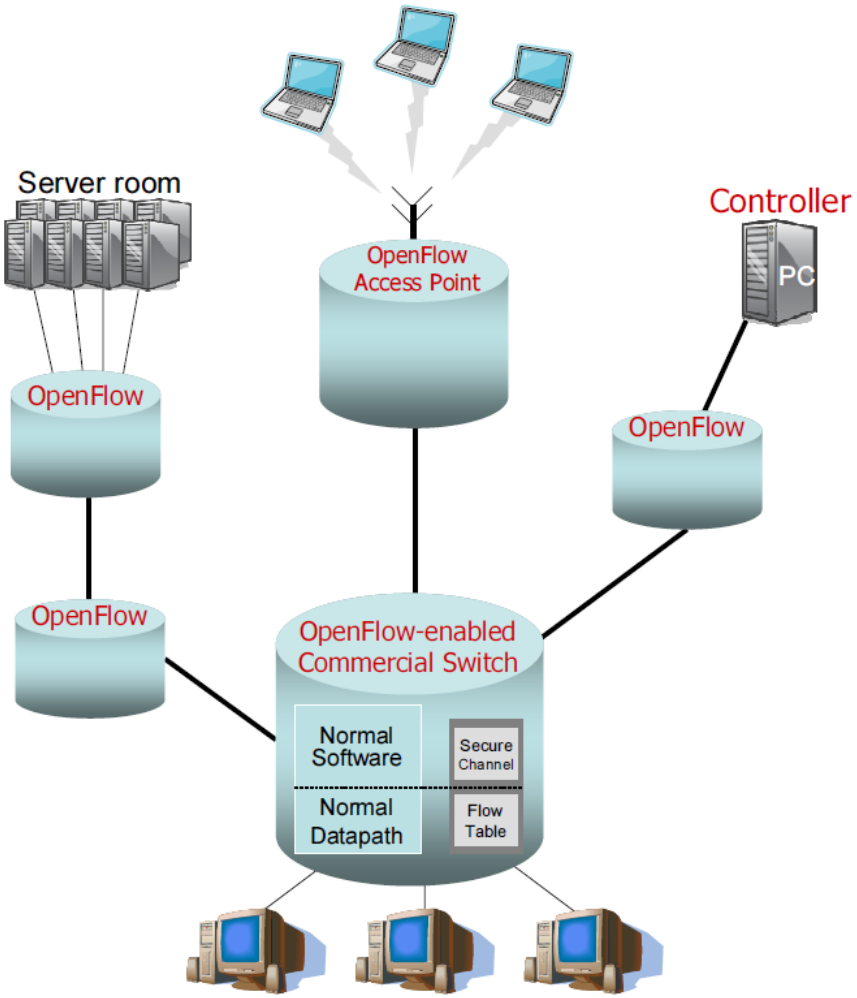
Jennifer Rexford
Princeton University

Scott Shenker
University of California, Berkeley

Jonathan Turner
Washington University in St. Louis

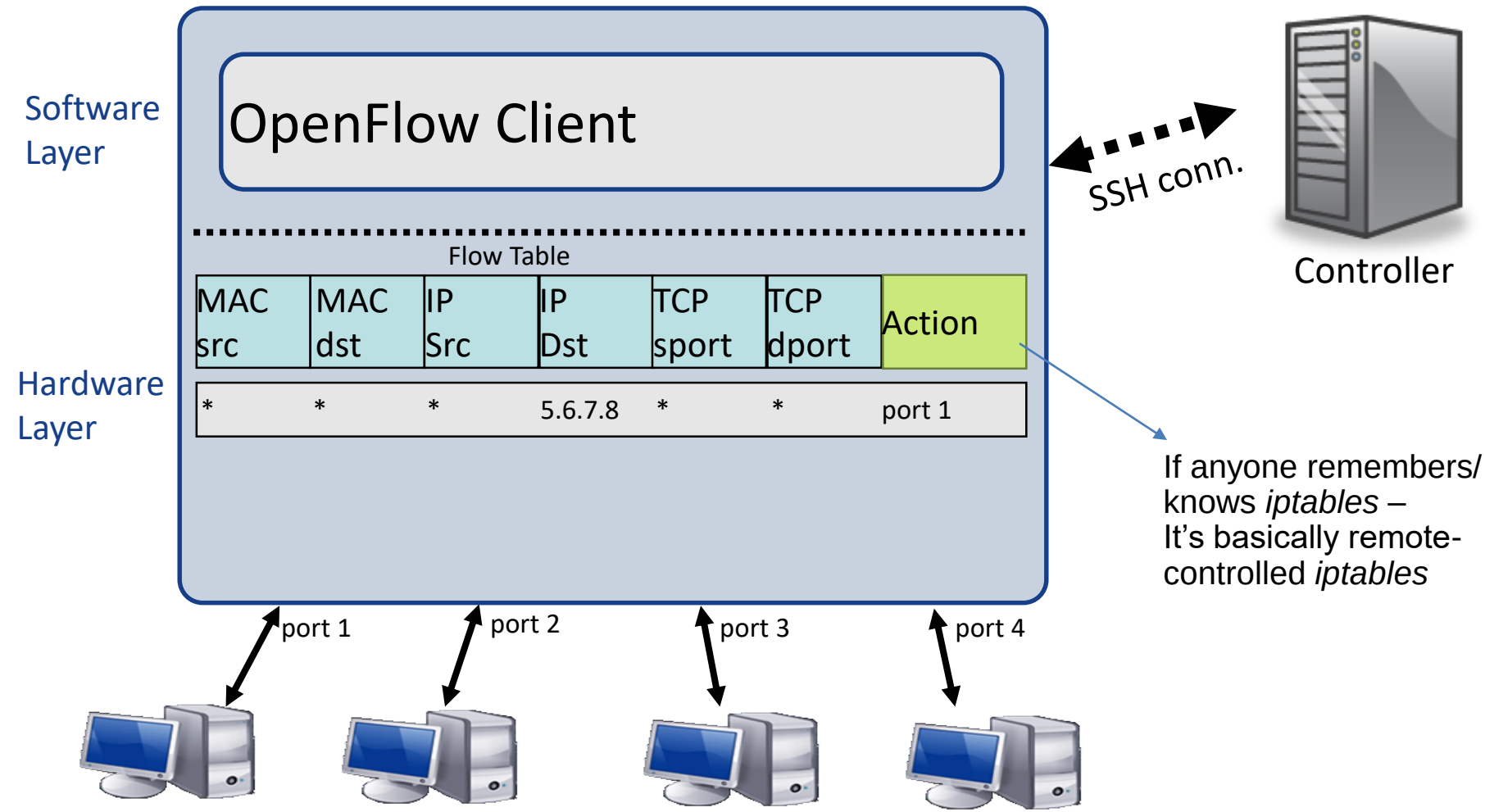
Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner, OpenFlow: Enabling Innovation in Campus Networks. *SIGCOMM Computer Communications Review,* vol 38, issue 2, March 2008, pp 69-74.

From the Original Openflow Paper



•Nick McKeown et al, SIGCOMM 2008

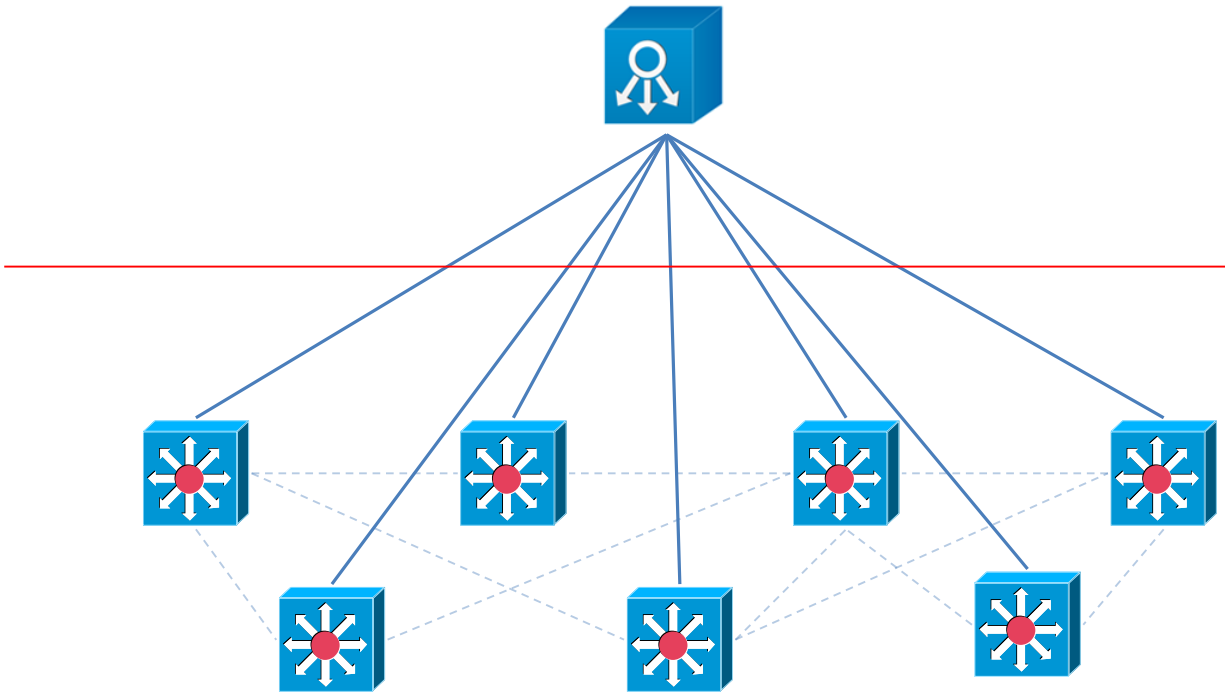
Openflow Switch



If anyone remembers/ knows *iptables* –
It's basically remote-controlled *iptables*

•Yiannis Yiakoumis, et al, HOT-Interconnects 2010

Control Plane vs Data Plane



Control Plane

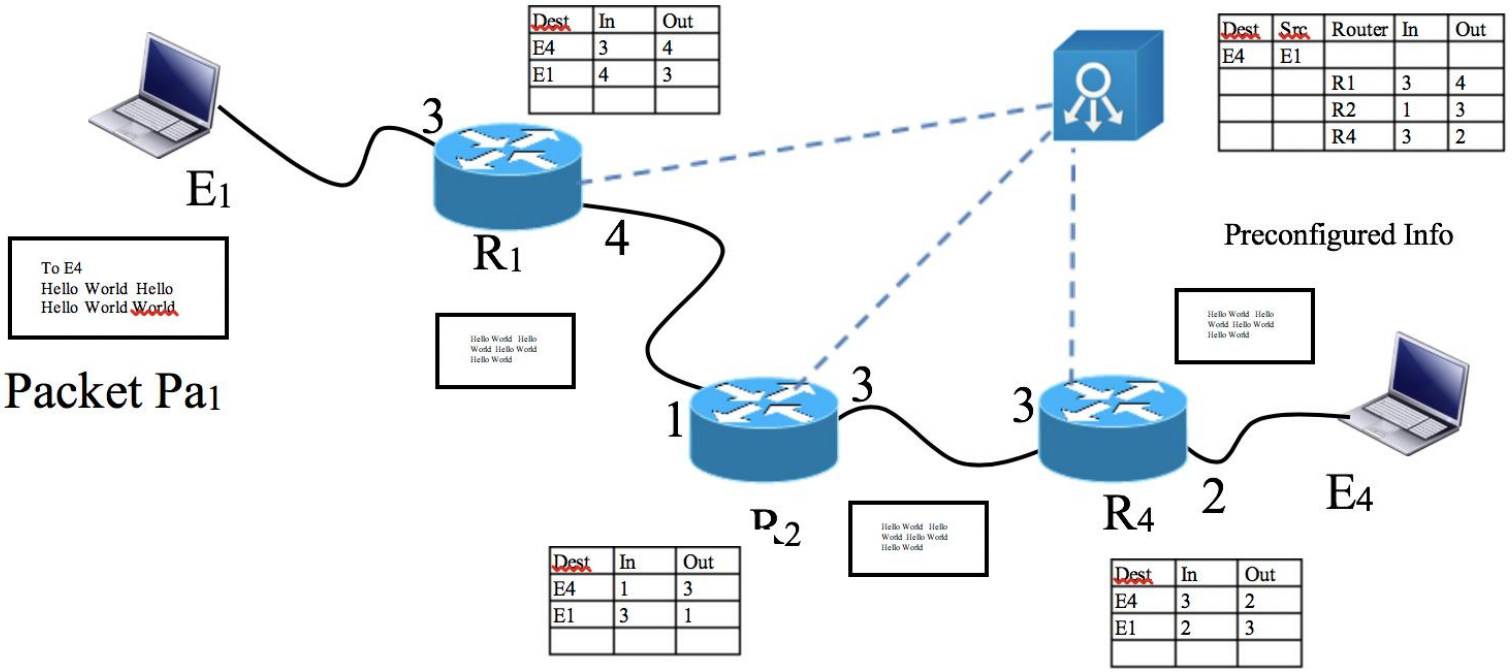
making routing decisions

Data Plane

forwarding data

Assignment 2: OpenFlow

- Focus: Semi-realistic implementation of OpenFlow i.e. Flow tables & Packet types



OpenFlow Packet Type

enum **ofp_type** {

/* Immutable messages. */

OFPT_HELLO = 0,

OFPT_ERROR = 1,

OFPT_ECHO_REQUEST = 2,

OFPT_ECHO_REPLY = 3,

OFPT_EXPERIMENTER = 4,

/* Switch configuration messages. */

OFPT_FEATURES_REQUEST = 5,

OFPT_FEATURES_REPLY = 6,

OFPT_GET_CONFIG_REQUEST = 7,

OFPT_GET_CONFIG_REPLY = 8,

OFPT_SET_CONFIG = 9,

/* Asynchronous messages. */

OFPT_PACKET_IN = 10,

OFPT_FLOW_REMOVED = 11,

OFPT_PORT_STATUS = 12,

OFPT_PACKET_OUT = 13,

OFPT_FLOW_MOD = 14,

OFPT_GROUP_MOD = 15,

OFPT_PORT_MOD = 16,

OFPT_TABLE_MOD = 17,

/* Multipart messages. */

OFPT_MULTIPART_REQUEST = 18,

OFPT_MULTIPART_REPLY = 19,

/* Barrier messages. */

OFPT_BARRIER_REQUEST = 20,

OFPT_BARRIER_REPLY = 21,

/* Controller role change request messages. */

OFPT_ROLE_REQUEST = 24,

OFPT_ROLE_REPLY = 25,

/* Asynchronous message configuration. */

OFPT_GET_ASYNC_REQUEST = 26,

OFPT_GET_ASYNC_REPLY = 27,

OFPT_SET_ASYNC = 28,

/* Meters and rate limiters configuration messages. */

OFPT_METER_MOD = 29,

/* Controller role change event messages. */

OFPT_ROLE_STATUS = 30,

/* Asynchronous messages. */

OFPT_TABLE_STATUS = 31,

/* Request forwarding by the switch. */

OFPT_REQUESTFORWARD = 32,

/* Bundle operations. */

OFPT_BUNDLE_CONTROL = 33,

OFPT_BUNDLE_ADD_MESSAGE = 34,

/* Controller Status async message. */

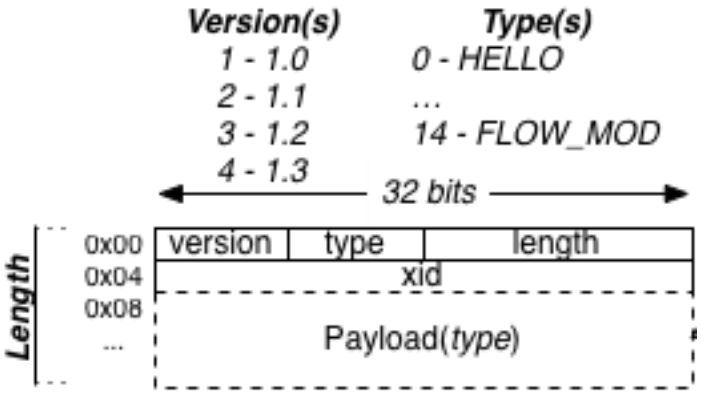
OFPT_CONTROLLER_STATUS = 35,

}

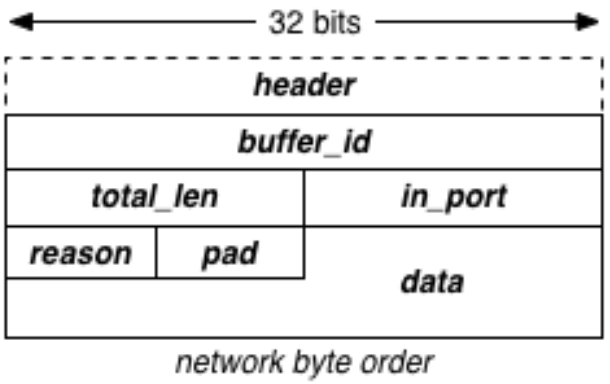
OpenFlow Switch Specification,v1.5.1, Open Net. Found.

OpenFlow Messages

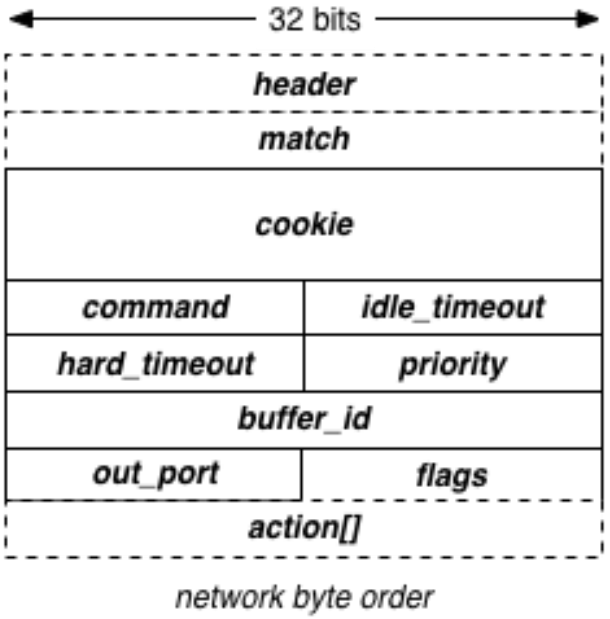
General Message Layout:



PacketIn:

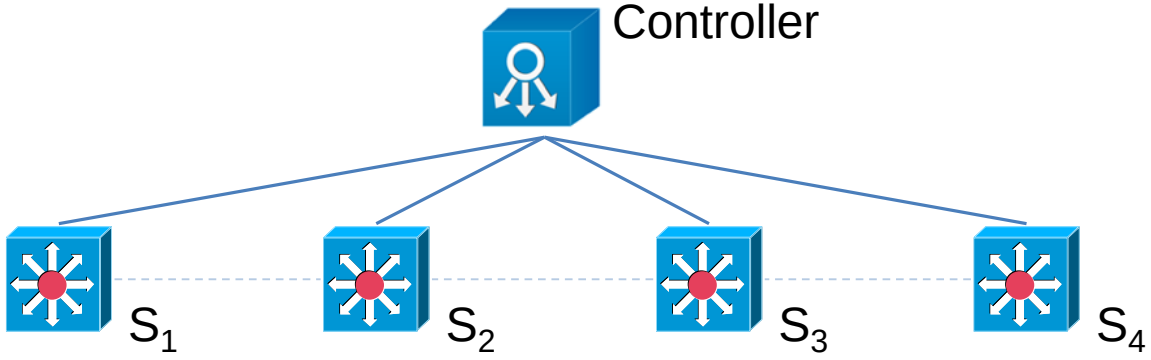


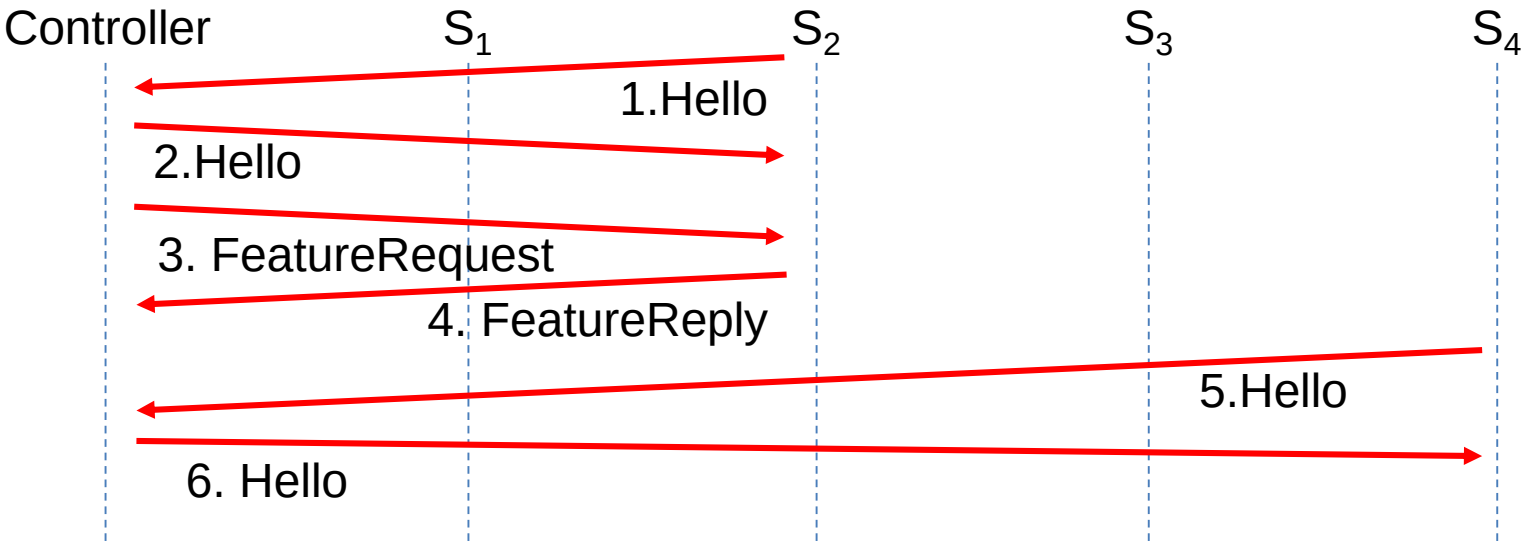
FlowMod:



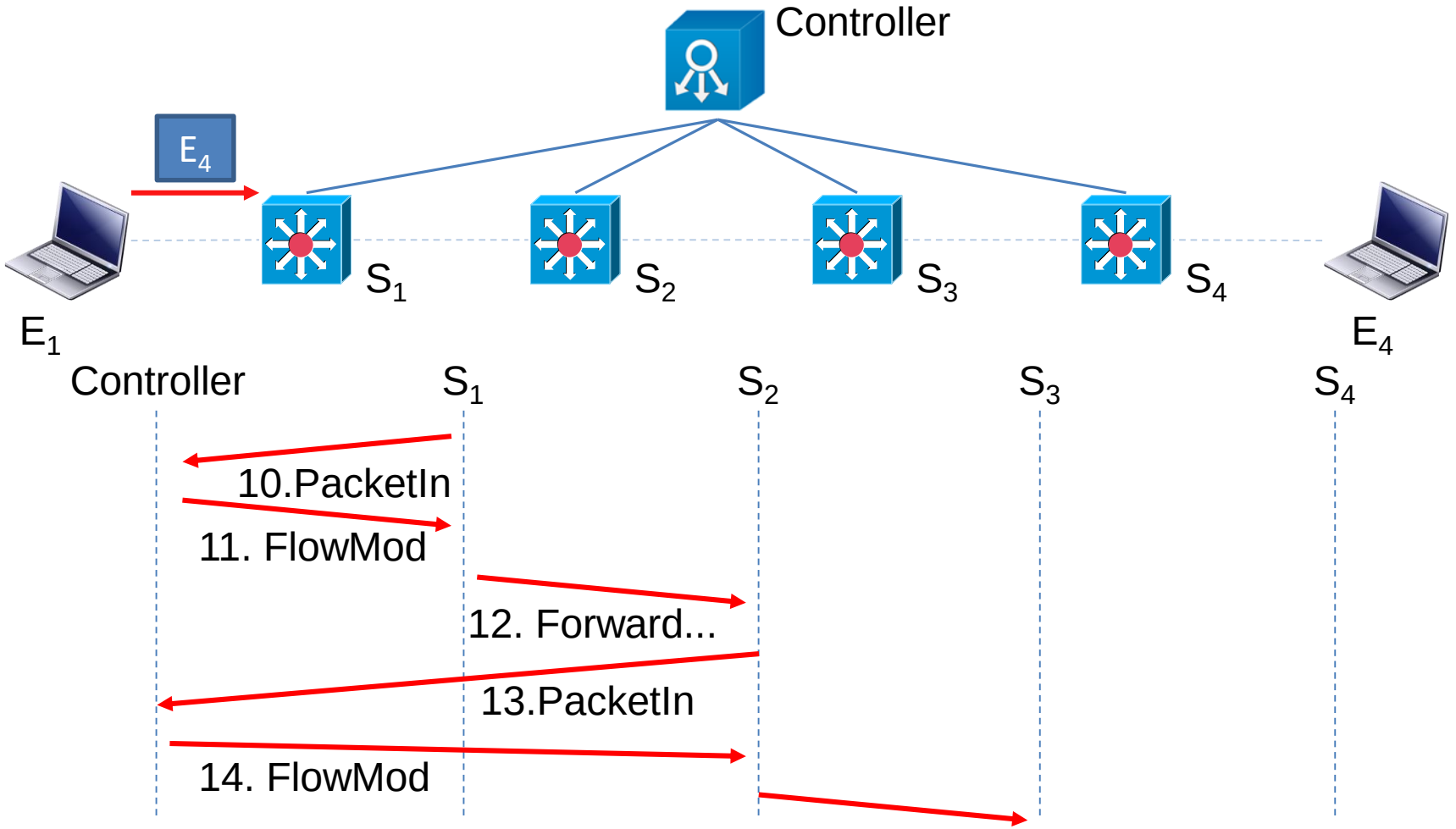
* http://flowgrammable.org/sdn/openflow/message-layer

OpenFlow Messages - Start

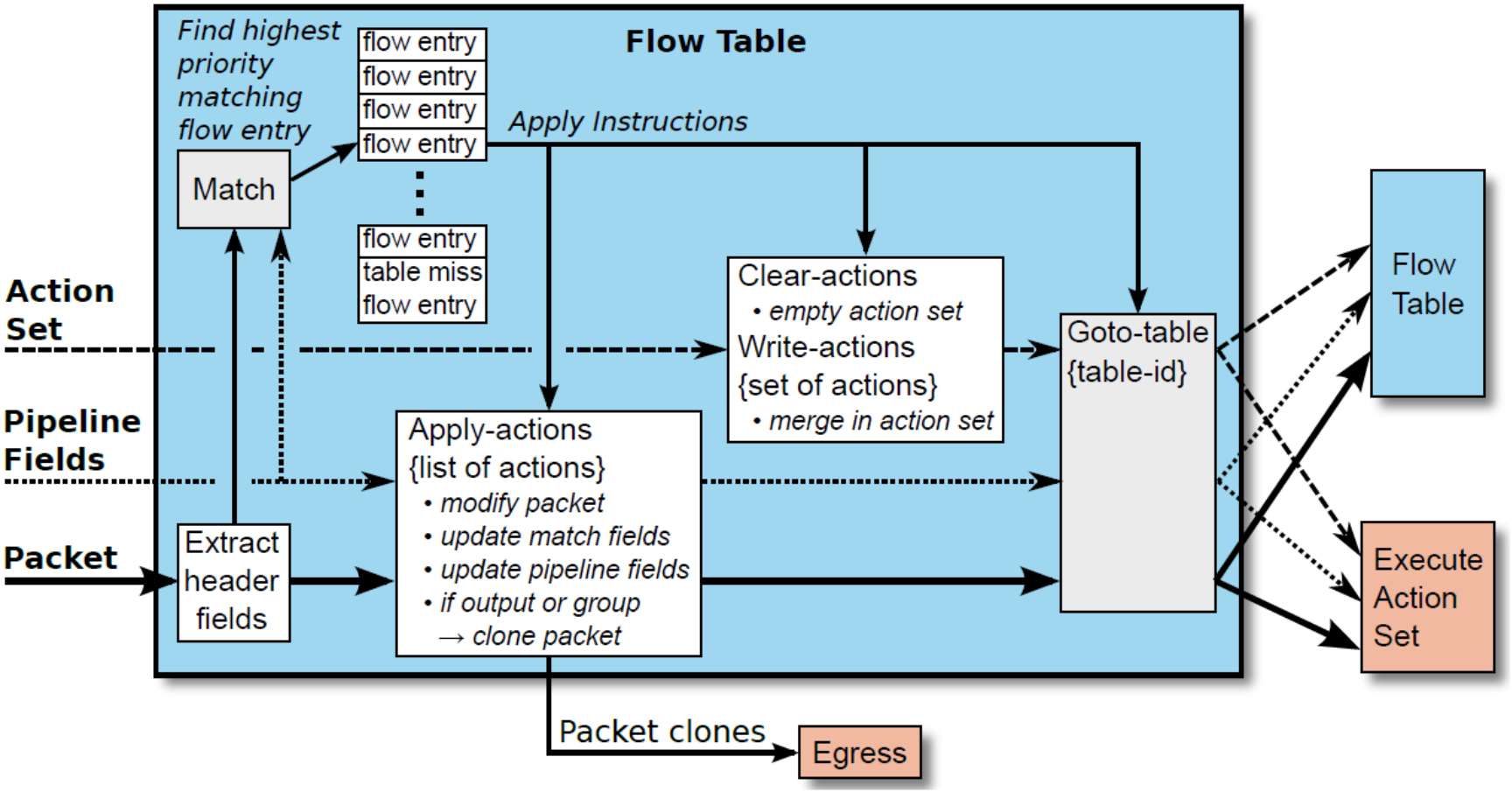




OpenFlow Messages - Message Del



Flow Table Processing



OpenFlow Switch Specification,v1.5.1, Open Net. Found.

Assignment 2: Feature List, pt 1

- Controller
  - Accept contact from switches and issue feature request
  - Accept PacketIn messages
  - Send out FlowMod messages
- Switch
  - Make contact with controller and reply to feature request
  - Send out PacketIn messages
  - Receive and incorporate instructions from FlowMod messages
  - Receive messages from end-nodes
  - Forward messages depending on flow table

Assignment 2: Feature List, pt 2

- End-node
  - Send messages addressed to another end-node
  - Receive messages

A.2: Indicative Marking Scheme*

Implementation

- Feature 1, Controller: 25 points
- Feature 2, Switch: 30 points
- Feature 3, Endnode: 15 points
- Feature 4, Packets: 15 points
- Feature 5: 10 points
- Feature 6: 5 points

Documentation

- Controller: 25 points
- Switch: 30 points
- Endnode: 15 points
- Packet: 15 points
- Discussion: 15 points

*subject to change

Assignment Timeline

- ~~Preliminary~~ Deadlines:

  31st December: Publish/Subscribe

  31st December: OpenFlow

- Submission through Blackboard mymodule.tcd.ie
- **Deadlines on Blackboard count**