

# Link layer, LANs: outline

6.1 introduction, services

6.2 error detection,  
correction

6.3 multiple access  
protocols

6.4 Switched LANs

- addressing, ARP
- Ethernet
- switches

6.7 a day in the life of a  
web request *(Self Study)*

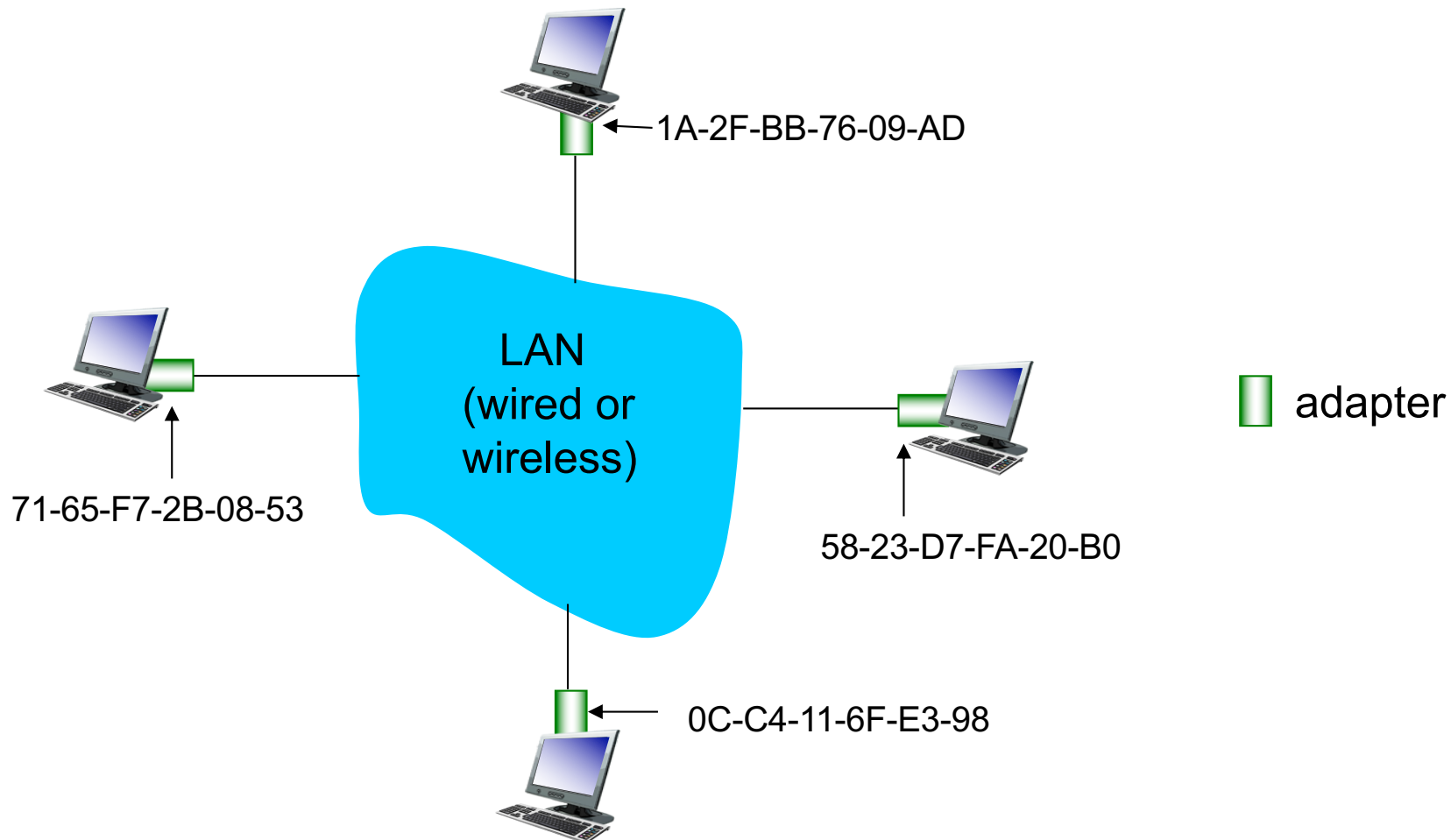
# MAC addresses and ARP

- ❖ 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
- ❖ MAC (or LAN or physical or Ethernet) address:
  - function: *used ‘locally’ to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)*
  - 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation  
(each “number” represents 4 bits)

# LAN addresses and ARP

each adapter on LAN has unique **LAN** address





# LAN addresses (more)

- ❖ MAC address allocation administered by IEEE
- ❖ manufacturer buys portion of MAC address space (to assure uniqueness) 24-24 split
- ❖ MAC flat address → portability
  - can move LAN card from one LAN to another
- ❖ IP hierarchical address *not* portable
  - address depends on IP subnet to which node is attached

# MAC Address vs. IP Address

- ❖ MAC addresses (used in link-layer)
  - **Hard-coded** in read-only memory when adapter is built
  - **Flat** name space of 48 bits (e.g., 00-0E-9B-6E-49-76)
  - Portable, and can stay the same as the host moves
  - Is used to get a packet between interfaces within the same IP subnet
  
- ❖ IP addresses
  - **Configured**, or learned dynamically
  - **Hierarchical** name space of 32 bits (e.g., 12.178.66.9)
  - Not portable, and depends on where the host is attached
  - Is used to get a packet to destination IP subnet

# Taking Stock: Naming

Layer	Examples	Structure	Configuration	Resolution Service
App. Layer	www.cse.unsw.edu.au	organizational hierarchy	~ manual	 DNS
Network Layer	129.94.242.51	topological hierarchy	DHCP	
Link layer	45-CC-4E-12-F0-97	vendor (flat)	hard-coded	 ARP

# ARP: address resolution protocol

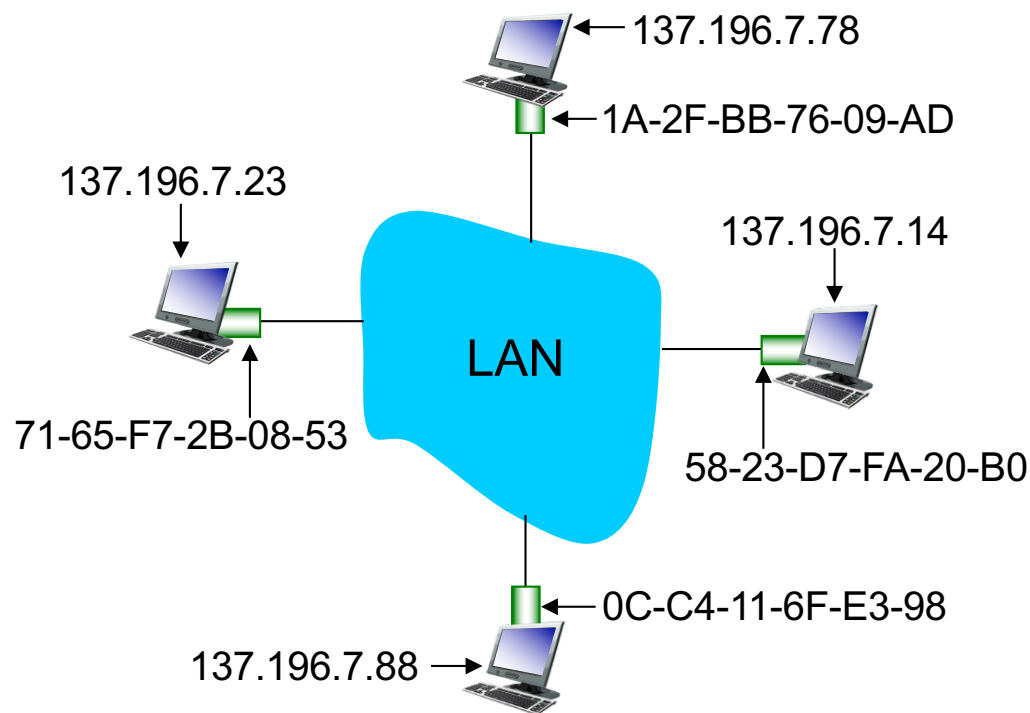
**Question:** how to determine interface's MAC address, knowing its IP address?

**ARP table:** each IP node (host, router) on LAN has table

- IP/MAC address mappings for some LAN nodes:

< IP address; MAC address; TTL >

- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)



# ARP protocol: same LAN

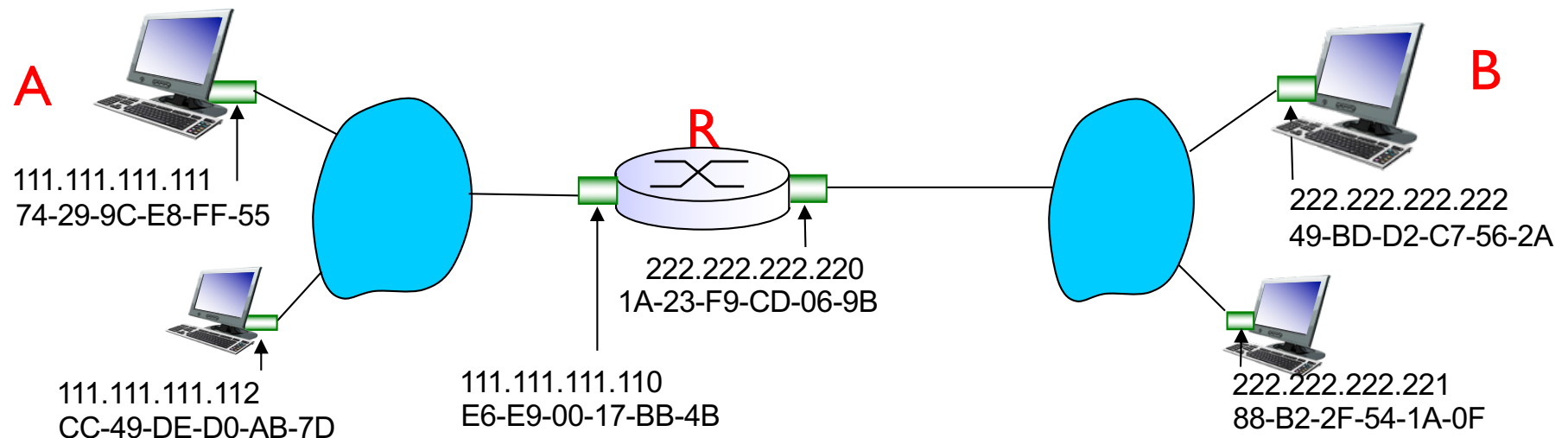
- ❖ A wants to send datagram to B
  - B's MAC address not in A's ARP table.
- ❖ A **broadcasts** ARP query packet, containing B's IP address
  - dest MAC address = FF-FF-FF-FF-FF-FF
  - all nodes on LAN receive ARP query
- ❖ B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)
- ❖ A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ❖ ARP is “plug-and-play”:
  - nodes create their ARP tables *without intervention from net administrator*



# Addressing: routing to another LAN

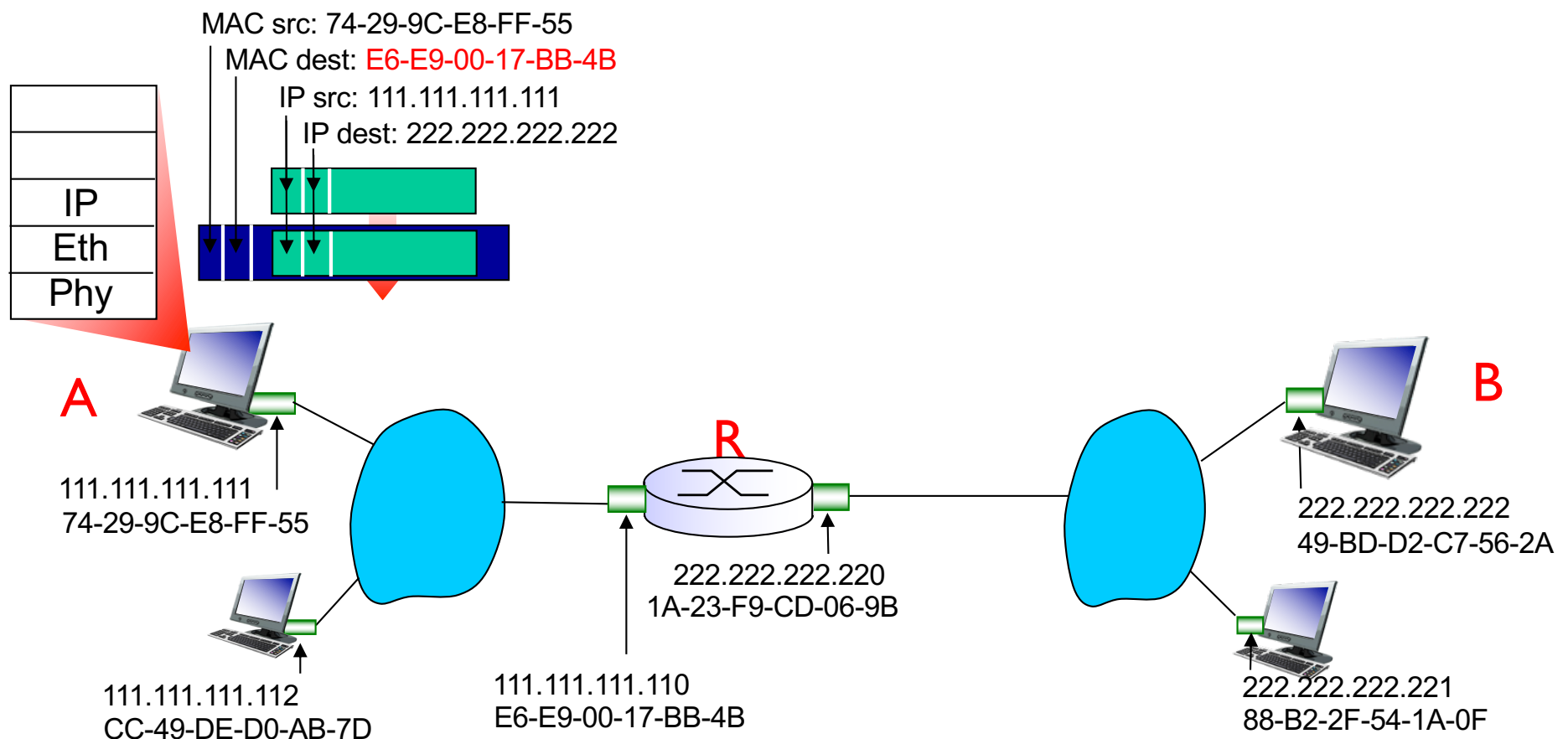
walkthrough: **send datagram from A to B via R**

- focus on addressing – at IP (datagram) and MAC layer (frame)
- assume A knows B's IP address (how?)
  - How does A know B is not local (i.e. connected to the same LAN as A) ?
    - Subnet Mask (discovered via DHCP)
- assume A knows IP address of first hop router, R (how?)
  - Default router (discovered via DHCP)
- assume A knows R's MAC address (how?)
  - ARP



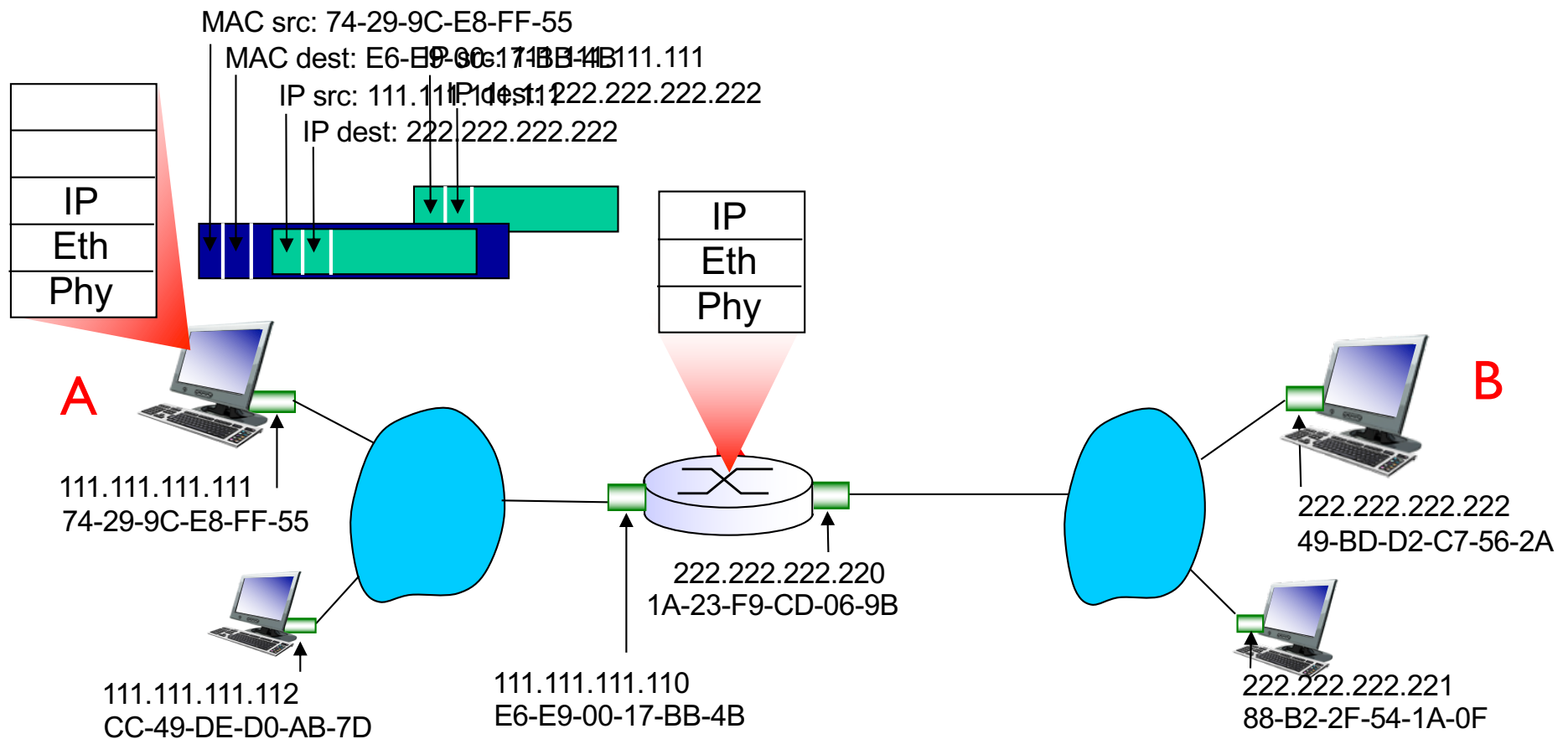
# Addressing: routing to another LAN

- ❖ A creates IP datagram with IP source A, destination B
- ❖ A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram



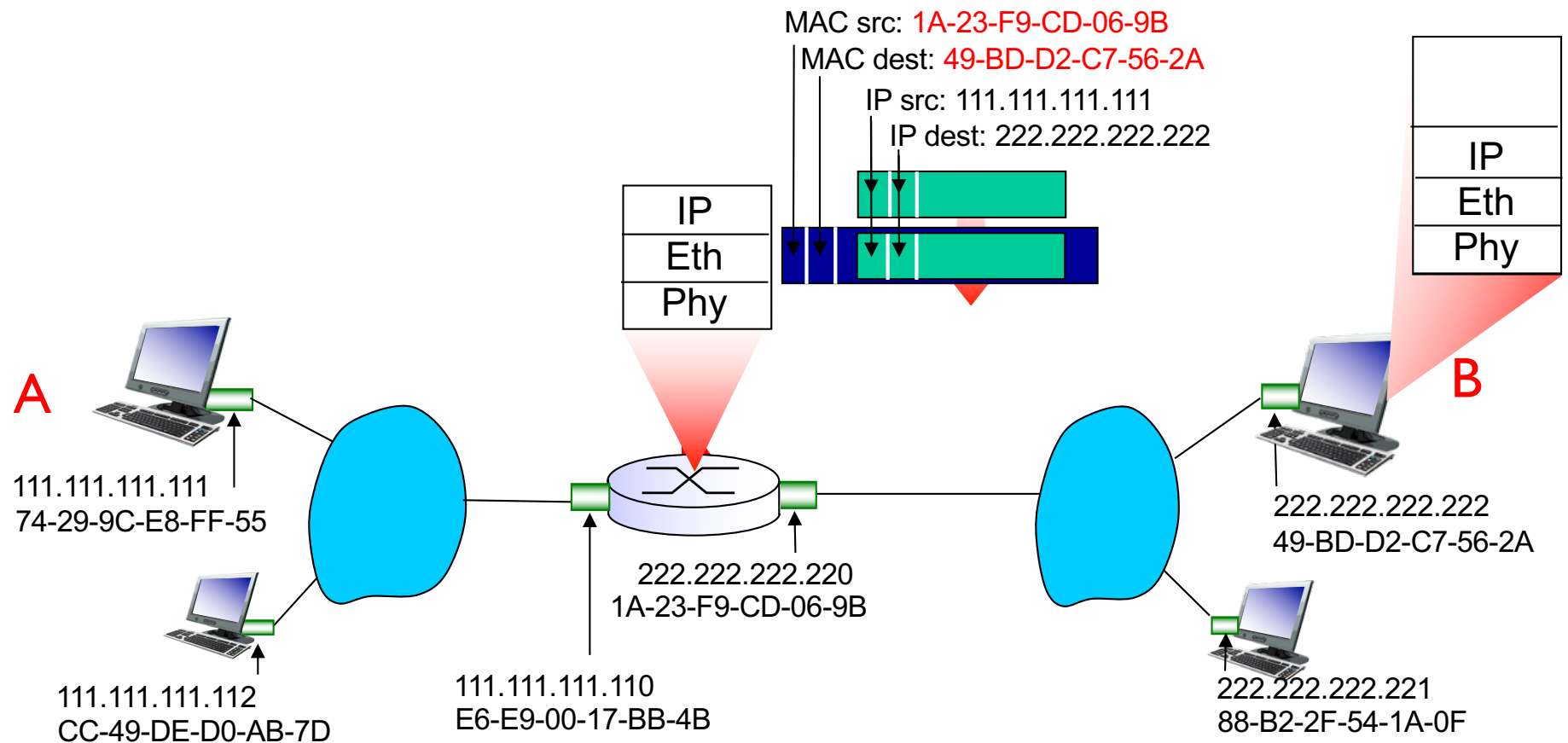
# Addressing: routing to another LAN

- ❖ frame sent from A to R
- ❖ frame received at R, datagram removed, passed up to IP



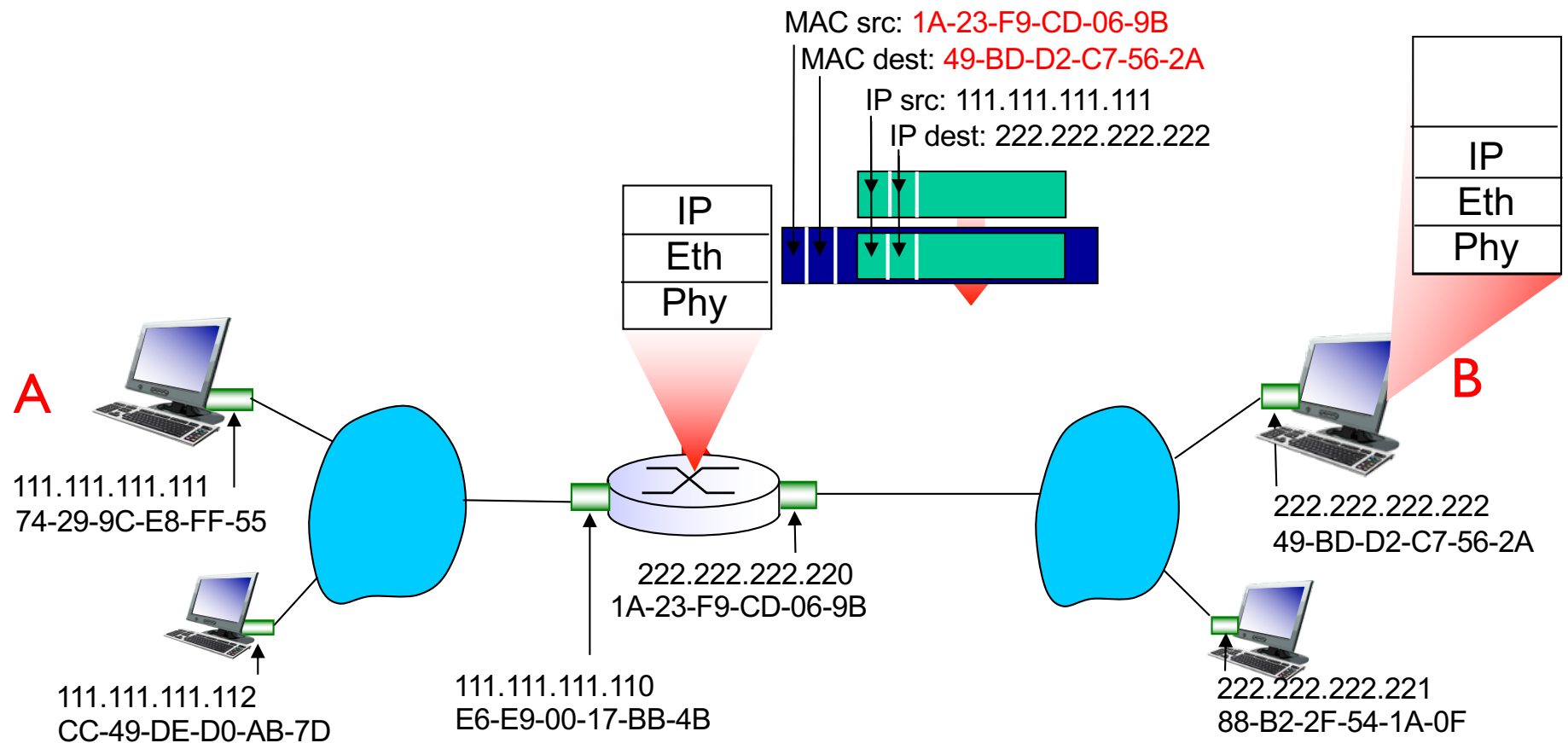
# Addressing: routing to another LAN

- ❖ R forwards datagram with IP source A, destination B (forwarding table)
- ❖ R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



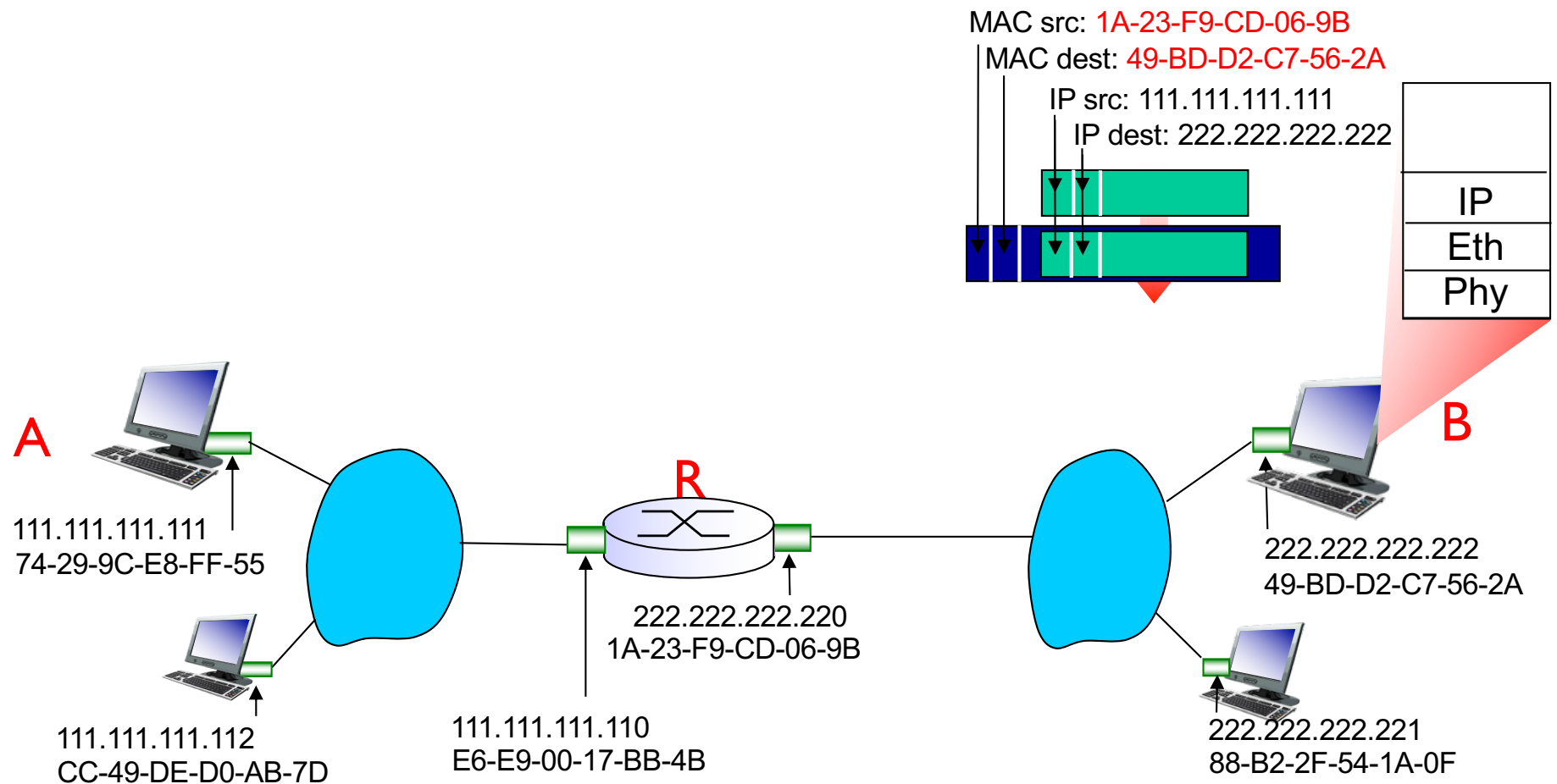
# Addressing: routing to another LAN

- ❖ R forwards datagram with IP source A, destination B
- ❖ R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



# Addressing: routing to another LAN

- ❖ R forwards datagram with IP source A, destination B
- ❖ R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



# Example ARP Table

```
C:\Windows\system32\cmd.exe

C:\Users\admin>arp -a

Interface: 192.168.150.155 --- 0xb
Internet Address      Physical Address      Type
192.168.150.2         00-10-db-82-4d-52     dynamic
192.168.150.10        00-0e-7f-af-6d-b8     dynamic
192.168.150.24        00-0f-fe-25-74-40     dynamic
192.168.150.32        00-0b-cd-6e-b8-2c     dynamic
192.168.150.36        00-0f-fe-3a-aa-3f     dynamic
192.168.150.42        00-0f-fe-87-1e-98     dynamic
192.168.150.48        00-0e-7f-63-8d-d1     dynamic
192.168.150.54        00-16-35-ae-3b-a9     dynamic
192.168.150.58        00-16-35-ae-39-53     dynamic
192.168.150.60        00-21-63-68-e9-29     dynamic
192.168.150.62        00-0f-fe-9b-e8-38     dynamic
192.168.150.78        00-0f-fe-3a-a7-d7     dynamic
192.168.150.90        00-0e-7f-f2-f8-e8     dynamic
192.168.150.92        00-0f-fe-3a-a7-96     dynamic
192.168.150.98        00-0f-fe-85-8d-6b     dynamic
192.168.150.114       00-0e-7f-6c-81-25     dynamic
192.168.150.144       00-22-5f-12-67-a2     dynamic
192.168.150.156       00-0f-fe-d1-7e-1e     dynamic
192.168.150.157       00-0f-fe-d1-7e-1e     dynamic
192.168.150.159       00-06-1b-c2-e1-f3     dynamic
192.168.150.208       00-19-66-32-53-25     dynamic
192.168.150.219       00-00-aa-8c-be-07     dynamic
192.168.150.221       00-0e-7f-64-5f-d0     dynamic
192.168.150.255       ff-ff-ff-ff-ff-ff     static
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.251           01-00-5e-00-00-fb     static
224.0.0.252           01-00-5e-00-00-fc     static
224.0.1.134           01-00-5e-00-01-86     static
239.255.255.250       01-00-5e-7f-ff-fa     static
```

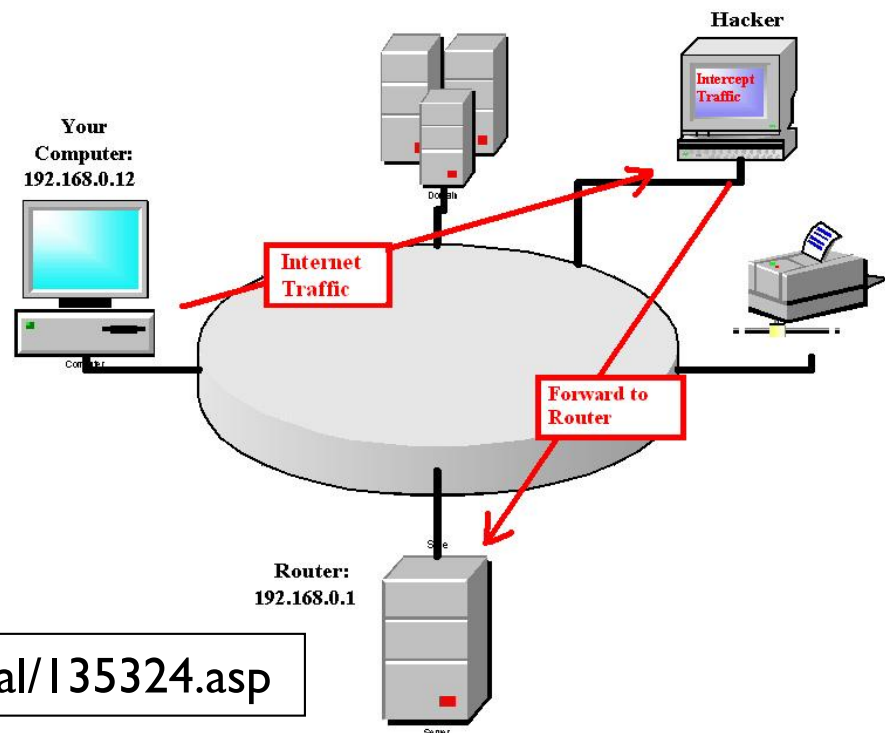
# Security Issues: ARP Cache Poisoning



- ❖ Denial of Service - Hacker replies back to an ARP query for a router NIC with a fake MAC address
- ❖ Man-in-the-middle attack - Hacker can insert his/her machine along the path between victim machine and gateway router
- ❖ Such attacks are generally hard to launch as hacker needs physical access to the network

## Solutions -

- Use Switched Ethernet with port security enabled (i.e. one host MAC address per switch port)
- Adopt static ARP configuration for small size networks
- Use ARP monitoring tools such as ARPWatch



<http://www.watchguard.com/infocenter/editorial/135324.asp>



# Link layer, LANs: outline

6.1 introduction, services

6.2 error detection,  
correction

6.3 multiple access  
protocols

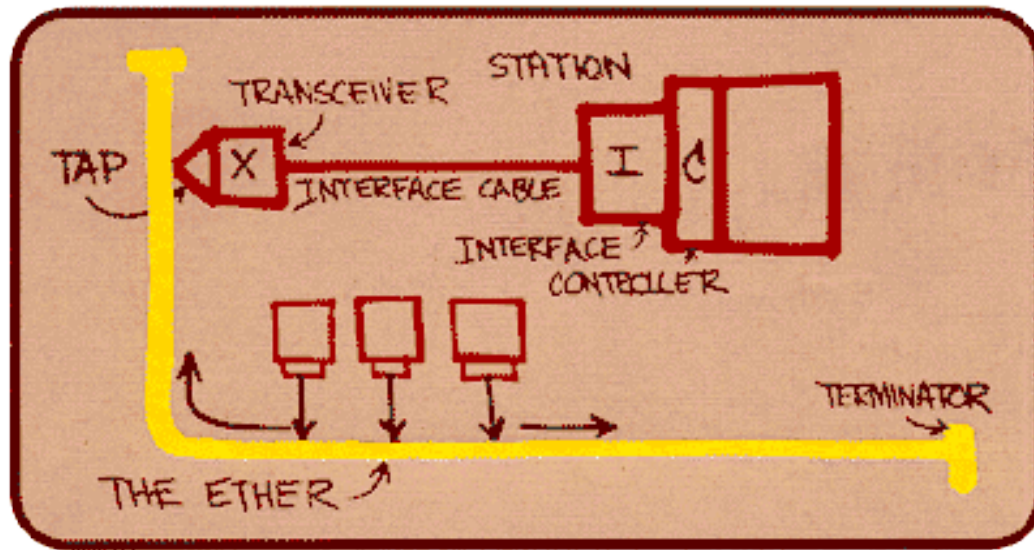
6.4 LANs

- addressing, ARP
- Ethernet
- switches

6.7 a day in the life of a  
web request

# Ethernet

Bob Metcalfe, Xerox PARC, visits Hawaii and gets an idea!



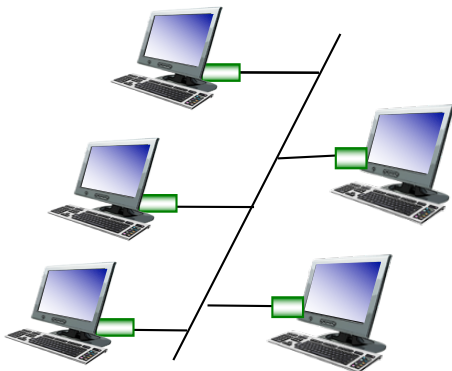
*Metcalfe's Ethernet sketch*

“dominant” wired LAN technology:

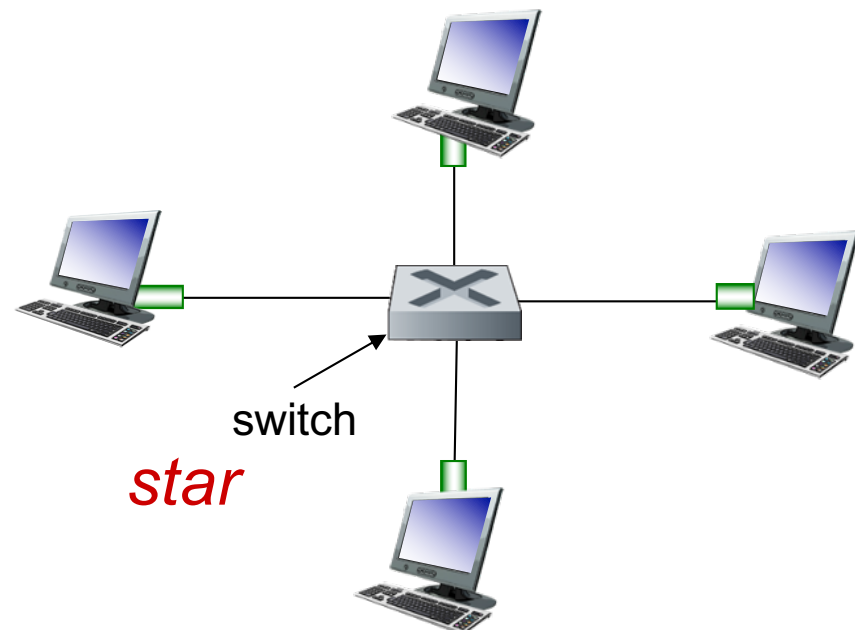
- ❖ first widely used LAN technology
- ❖ simpler, cheaper than token LANs and ATM
- ❖ kept up with speed race: 10 Mbps – 10 Gbps

# Ethernet: physical topology

- ❖ *bus*: popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
  - CSMA/CD for media access control
- ❖ *star*: prevails today
  - active *switch* in center
  - each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other)
  - No sharing, no CSMA/CD



*bus*: coaxial cable



# Ethernet frame structure

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in **Ethernet frame**

<b>Preamble</b> <b>7 Bytes</b>	<b>SFD</b> <b>1 Byte</b>	<b>Dest</b> <b>MAC</b> <b>6 Bytes</b>	<b>Source</b> <b>MAC</b> <b>6 Bytes</b>	<b>Type/Le</b> <b>ngth</b> <b>2 Bytes</b>	<b>Payload</b> <b>46-1500</b> <b>Bytes</b>	<b>FCS/C</b> <b>RC</b> <b>4</b> <b>Bytes</b>	<b>Inter</b> <b>Frame</b> <b>Gap</b>
-----------------------------------	-----------------------------	---	---	---	--	---	--

## *preamble:*

- ❖ Start of frame is recognized by
  - Preamble : Seven bytes with pattern 10101010
  - Start of Frame Delimiter (SFD) : 10101011
- ❖ used to synchronize receiver, sender clock rates
- Inter Frame Gap is 12 Bytes (96 bits) of idle state
  - 0.96 microsec for 100 Mbit/s Ethernet
  - 0.096 microsec for Gigabit/s Ethernet

# Ethernet frame structure (more)

- ❖ **addresses:** 6 byte source, destination MAC addresses
  - if adapter receives frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
- ❖ **type:** indicates higher layer protocol (mostly IP but others possible, e.g., ARP, Novell IPX, AppleTalk)
- ❖ **CRC:** cyclic redundancy check at receiver
  - error detected: frame is dropped



# Ethernet: unreliable, connectionless

- ❖ *connectionless*: no handshaking between sending and receiving NICs
- ❖ *unreliable*: receiving NIC does not send acks or nacks to sending NIC
  - data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost
- ❖ Ethernet's MAC protocol: unslotted *CSMA/CD with binary backoff*

# Link layer, LANs: outline

6.1 introduction, services

6.2 error detection,  
correction

6.3 multiple access  
protocols

6.4 LANs

- addressing, ARP
- Ethernet
- switches

6.7 a day in the life of a  
web request

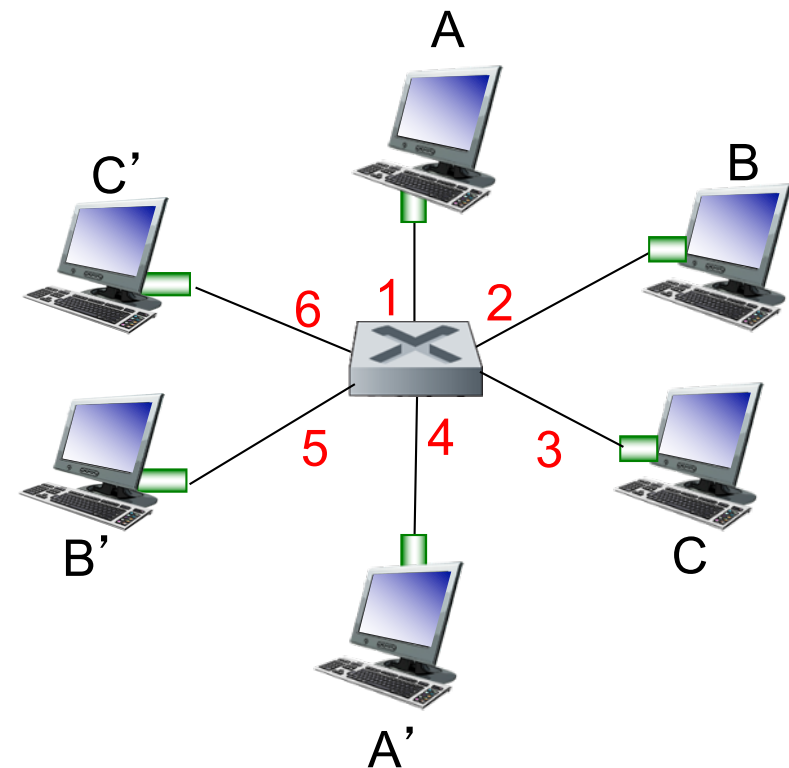
# Ethernet switch

- ❖ *link-layer device: takes an active role*
  - store, forward Ethernet frames
  - examine incoming frame's MAC address, *selectively* forward frame to one-or-more outgoing links when frame is to be forwarded on segment
- ❖ *transparent*
  - hosts are unaware of presence of switches
- ❖ *plug-and-play, self-learning*
  - switches do not need to be configured



# Switch: *multiple* simultaneous transmissions

- ❖ hosts have dedicated, direct connection to switch
- ❖ switches buffer packets
- ❖ Ethernet protocol used on *each* incoming link, but no collisions; full duplex
  - each link is its own collision domain
- ❖ *switching*: A-to-A' and B-to-B' can transmit simultaneously, without collisions



switch with six interfaces  
(1,2,3,4,5,6)

# Switch forwarding table

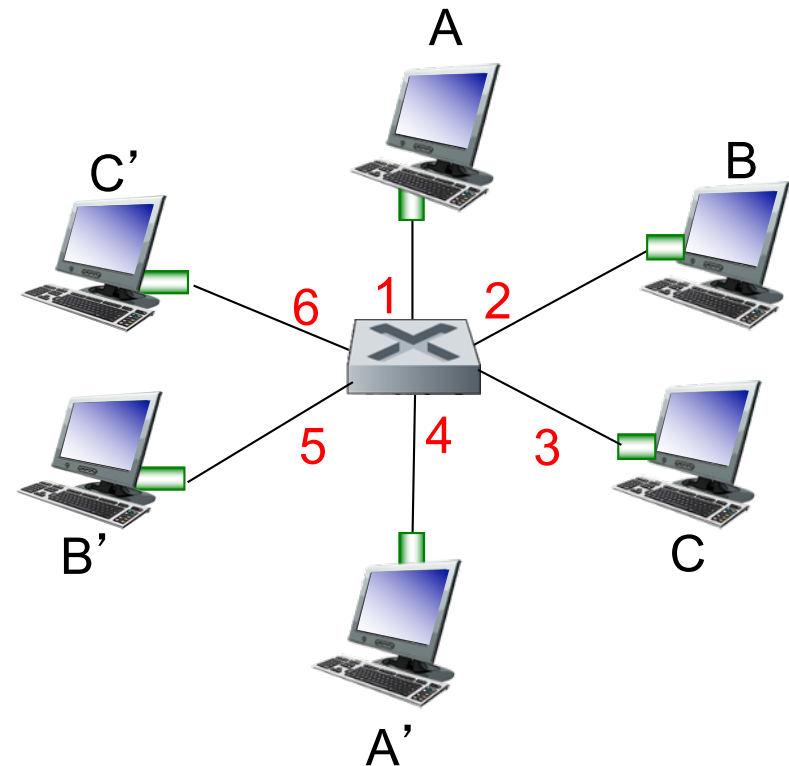
Q: how does switch know A' reachable via interface 4, B' reachable via interface 5?

❖ A: each switch has a **switch table**, each entry:

- (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!

Q: how are entries created, maintained in switch table?

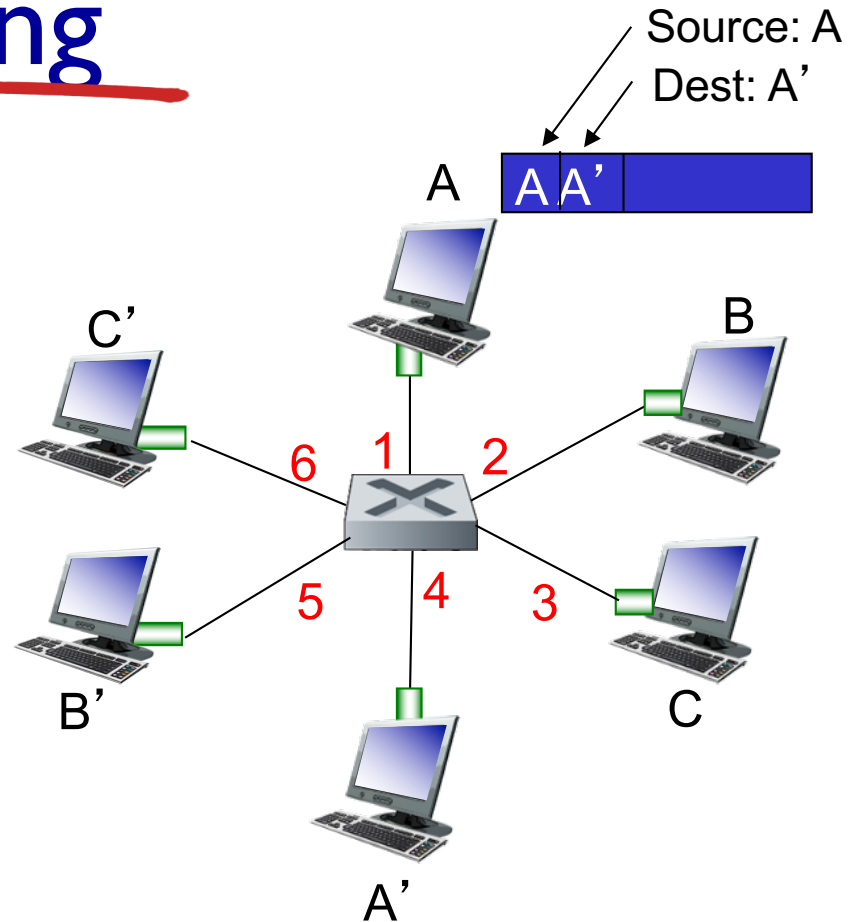
- something like a routing protocol?



switch with six interfaces  
(1,2,3,4,5,6)

# Switch: self-learning

- ❖ switch *learns* which hosts can be reached through which interfaces
  - when frame received, switch “learns” location of sender: incoming LAN segment
  - records sender/location pair in switch table



MAC addr	interface	TTL
A	1	60

*Switch table  
(initially empty)*

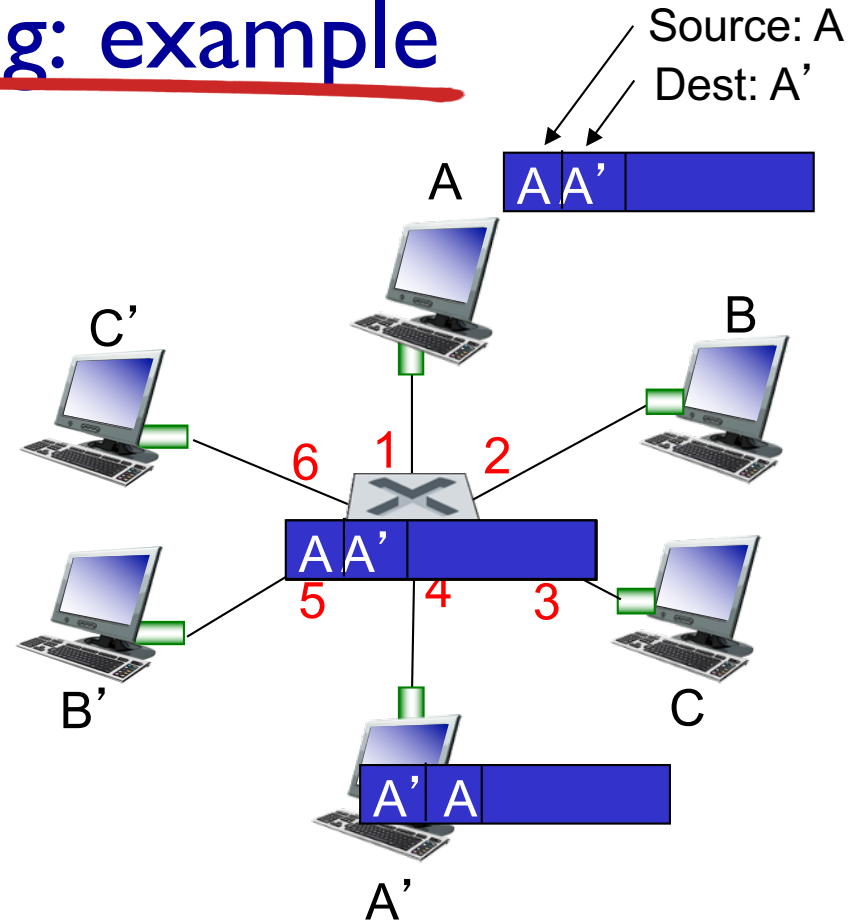
# Switch: frame filtering/forwarding

when frame received at switch:

1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. if entry found for destination  
    then {  
        if destination on segment from which frame arrived  
            then drop frame  
            else forward frame on interface indicated by entry  
        }  
    else flood /\* forward on all interfaces except arriving  
                    interface \*/

# Self-learning, forwarding: example

- ❖ frame destination, A', location unknown: *flood*
- ❖ destination A location known: *selectively send on just one link*

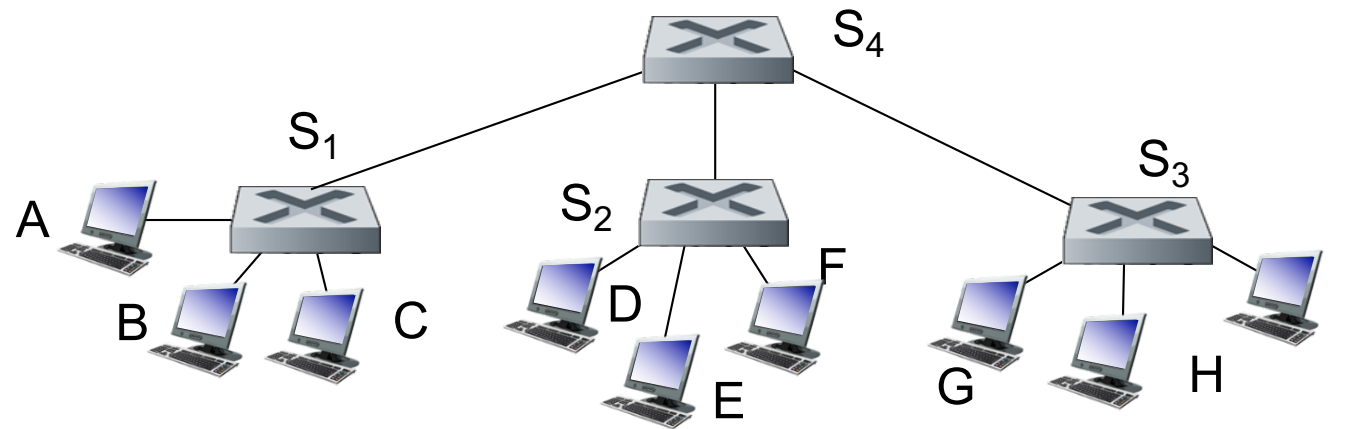


MAC addr	interface	TTL
A	1	60
A'	4	60

*switch table  
(initially empty)*

# Interconnecting switches

- ❖ switches can be connected together



Q: sending from A to G - how does  $S_1$  know to forward frame destined to G via  $S_4$  and  $S_3$ ?

- ❖ A: self learning! (works *exactly* the same as in single-switch case!)

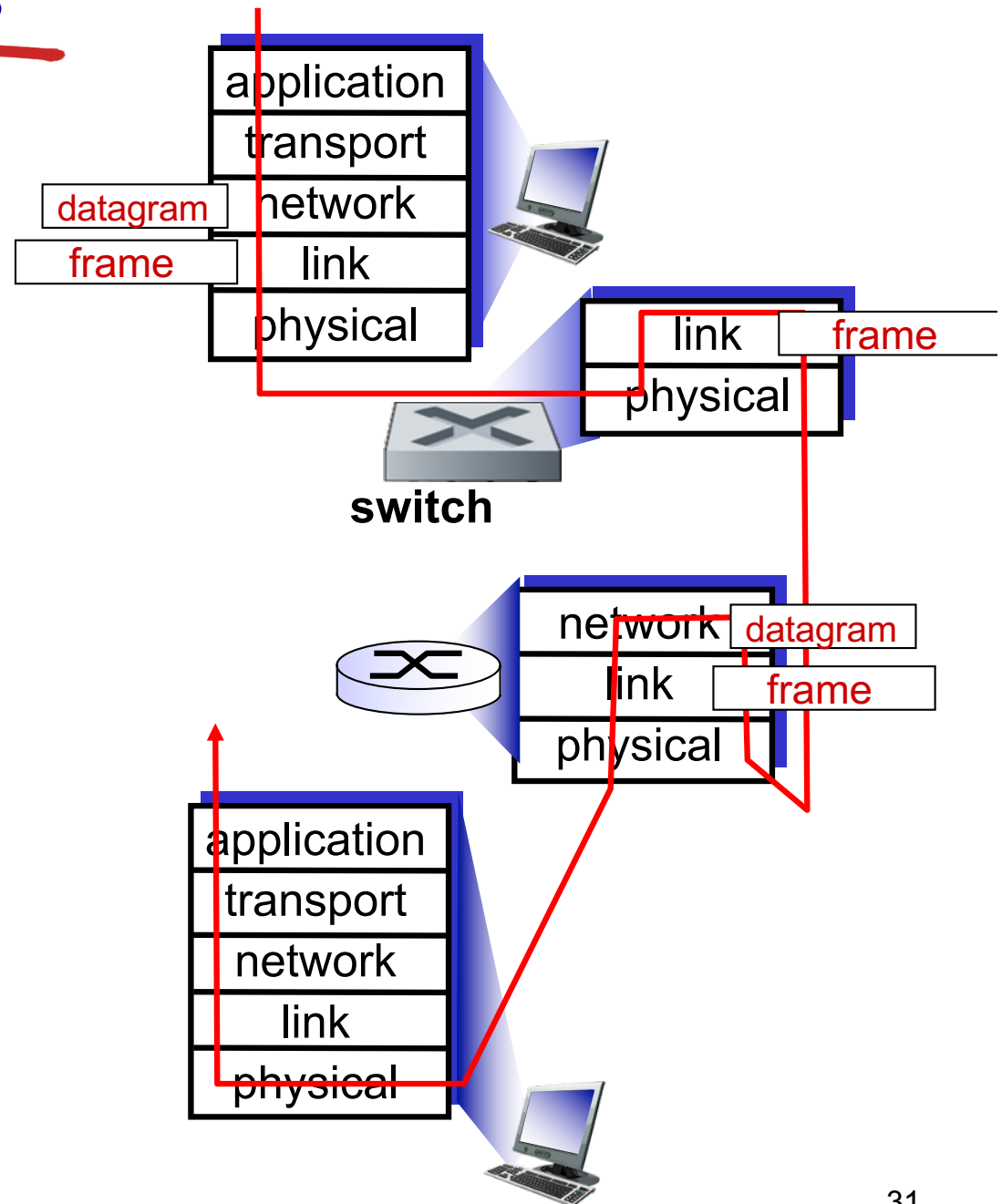
# Switches vs. routers

both are store-and-forward:

- **routers:** network-layer devices (examine network-layer headers)
- **switches:** link-layer devices (examine link-layer headers)

both have forwarding tables:

- **routers:** compute tables using routing algorithms, IP addresses
- **switches:** learn forwarding table using flooding, learning, MAC addresses



# Security Issues

- ❖ In a switched LAN once the switch table entries are established frames are not broadcast
  - Sniffing frames is harder than pure broadcast LANs
  - Note: attacker can still sniff broadcast frames and frames for which there are no entries (as they are broadcast)
- ❖ Switch Poisoning: Attacker fills up switch table with bogus entries by sending large # of frames with bogus source MAC addresses
- ❖ Since switch table is full, genuine packets frequently need to be broadcast as previous entries have been wiped out



# Quiz

---

❖ A switch can

- A. Filter a frame
- B. Forward a frame
- C. Extend a LAN
- D. All of the above

B

# Link layer, LANs: outline

6.1 introduction, services

6.2 error detection,  
correction

6.3 multiple access  
protocols

6.4 LANs

- addressing, ARP
- Ethernet
- switches

6.7 a day in the life of a  
web request

# Link Layer: Summary

- ❖ principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
- ❖ instantiation and implementation of various link layer technologies
  - Ethernet
  - switched LANS

# Link Layer: let's take a breath

- ❖ journey down protocol stack *complete* (except PHY)
- ❖ solid understanding of networking principles, practice
- ❖ ..... could stop here .... but *we have not covered wireless yet!*
  - *Next week: Wireless in the link layer*



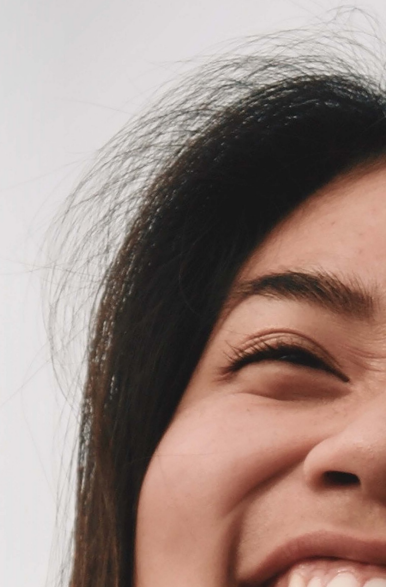
**“Having the lecture recordings up – absolute lifesaver when it comes to revision”**

**Shape  
our  
Future**

Tell us about your experience and shape the future of education at UNSW.



Click the link in Moodle now



# Shape our Future

*Complete your myExperience and shape  
the future of education at UNSW.*

Click the  Experience link in Moodle

**or login to [myExperience.unsw.edu.au](https://myExperience.unsw.edu.au)**

(use z1234567@**ad**.unsw.edu.au to login)

*The survey is confidential, your identity will never be released*

*Survey results are not released to teaching staff until after your results are published*