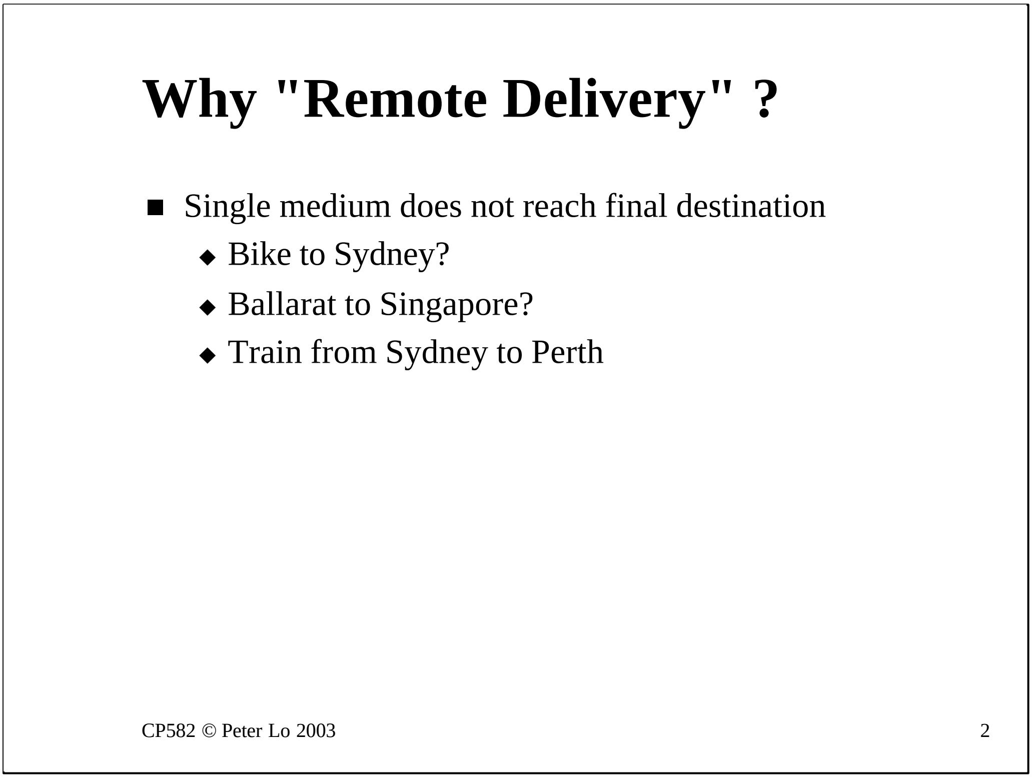
Remote Delivery

Peter Lo



Why "Remote Delivery" ?

- Single medium does not reach final destination
  - Bike to Sydney?
  - Ballarat to Singapore?
  - Train from Sydney to Perth

In the real world, and in networking, a single medium (a single local network) may not get us all the way to where we want to go.

Package Analogy

- Package must travel from Hong Kong to Beijing
- Package point of view
  - Address – Tsinghua University, Beijing, China
- Top down point of view
  - Walk to post Office
  - Van to the airport
  - Fly to Beijing
  - Van to Tsinghua University
  - Etc, etc
- Package does not need to "know"

Here's another analogy:

Suppose we have a package that we want to send to a remote location.

From our point of view we need to put the correct, multi-level address on the package and post it. It arrives at its destination - maybe a phone call can confirm that.

The detailed view - say from a camera taped to the package - is divided into a series of steps:

• Walk to the post office

• Van to the airport

• Fly to Beijing

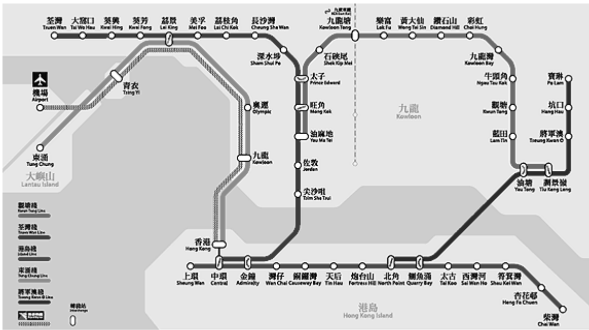
• Van to Tsinghua University

• etc etc

The package remains unmodified throughout all this. Each of these steps along the way is simply something else that is done to the package based on the address that we wrote on the outside! The package does "not need to know" what is going on.

Analogy - MTR Trains

- Assumptions
  - Different gauges in different states
    - No direct end-to-end delivery
  - Various interchanges on borders between states
    - Meeting point for different gauge systems
- Trace a package from Mongkok to Po Lam

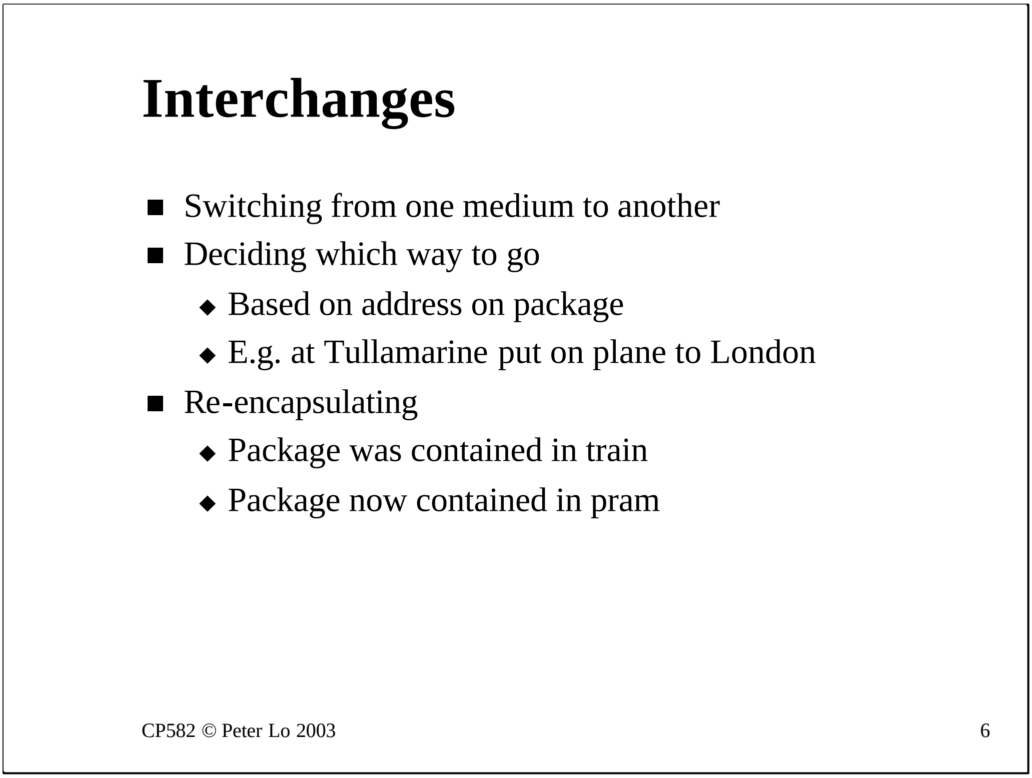


Even within a single country - one in which the rail system has not been properly coordinated - there will be a series of steps along the way.

At each boundary, between different rail gauges there will be an interchange of some sort where the packages are transferred from one system to the next.

A slide like this one looks much better in the lecture!

Looking at the printed slide the thing to note is the different “transports” in each state and the way that the letter (the data, the payload) is simply switched between transports at each boundary - from a “train” to a “pram” to a “ute”



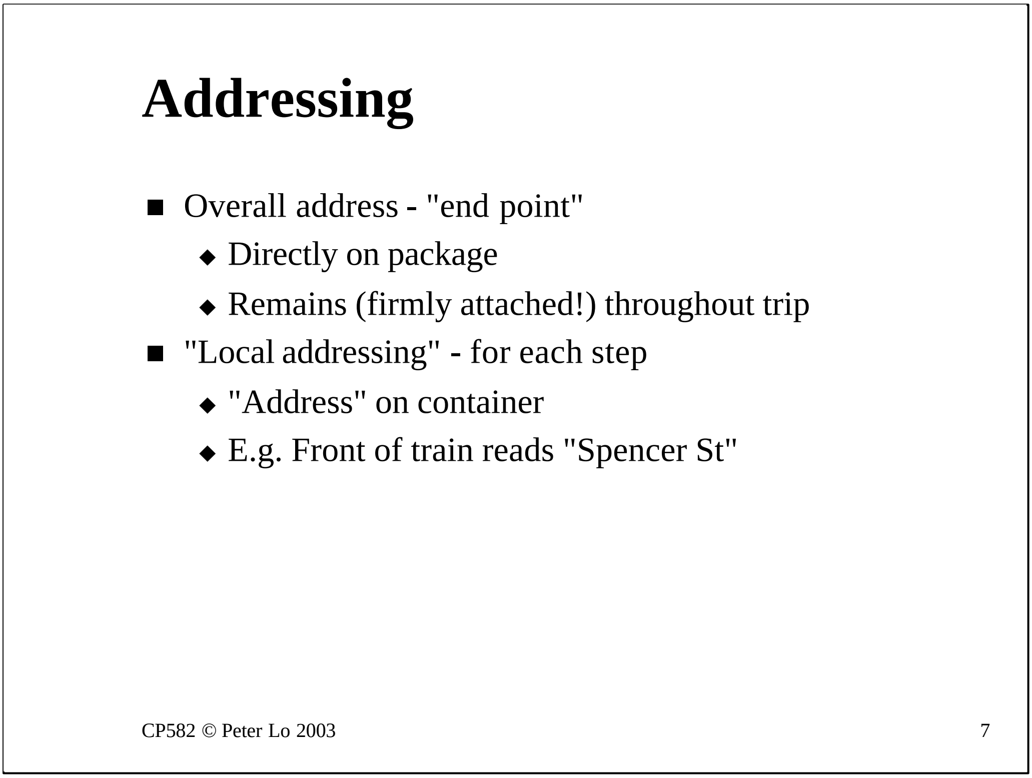
Interchanges

- Switching from one medium to another
- Deciding which way to go
  - Based on address on package
  - E.g. at Tullamarine put on plane to London
- Re-encapsulating
  - Package was contained in train
  - Package now contained in pram

Focusing on what happens at the borders - at the interchanges - we see that there may be a decision made about what direction to send the package. This decision is based on reading the address of the ultimate destination and then having some sense of the geography of the world.

For example, if we were sorting mail at Tullamarine and putting packages onto appropriate planes, we would need to know that the package addressed to Ireland needs to be put on the plane to London and not the plane to LA.

We introduce the term "encapsulation" to describe what happens to the package when it is placed in a particular transport. In Victoria the package was encapsulated in a train, in SA in a pram etc etc



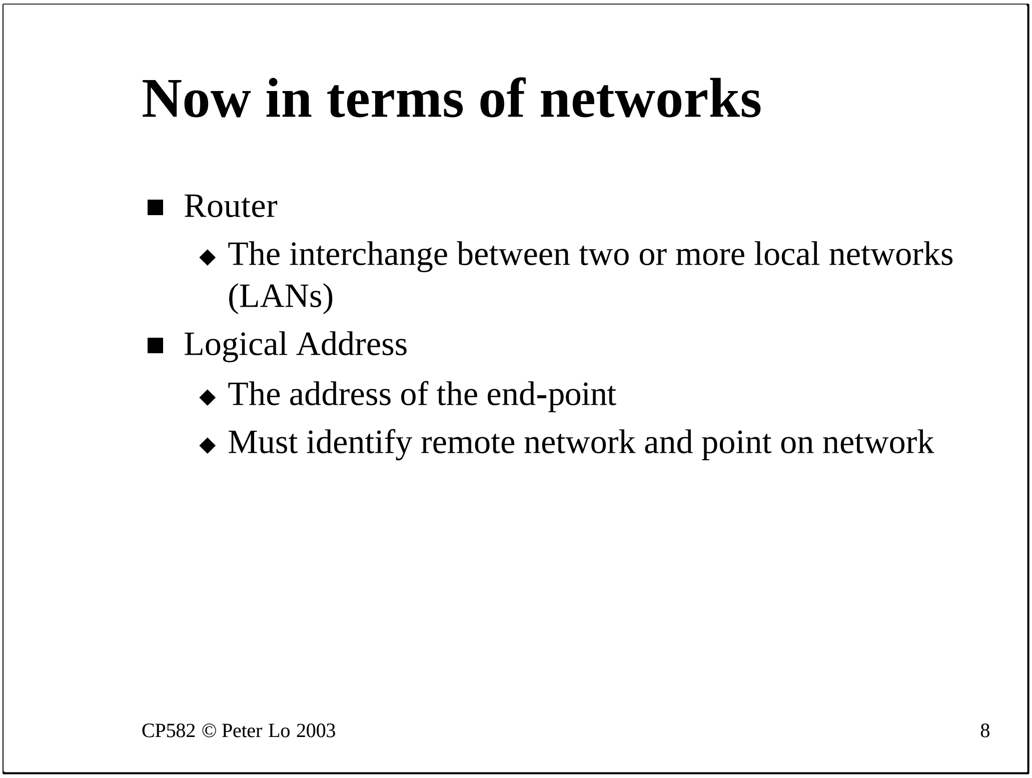
Addressing

- Overall address - "end point"
  - Directly on package
  - Remains (firmly attached!) throughout trip
- "Local addressing" - for each step
  - "Address" on container
  - E.g. Front of train reads "Spencer St"

We can begin to see that there are two types of addressing in operation.

We have already observed that the overall address for the package is written on the outside and is referred to when routing decisions need to be made along the way.

However there is also some form of “local addressing” for each leg of the journey. This local addressing is part of the transport for that stage - for example “Destination of plane = London” or “Destination of train = Spencer St”



Now in terms of networks

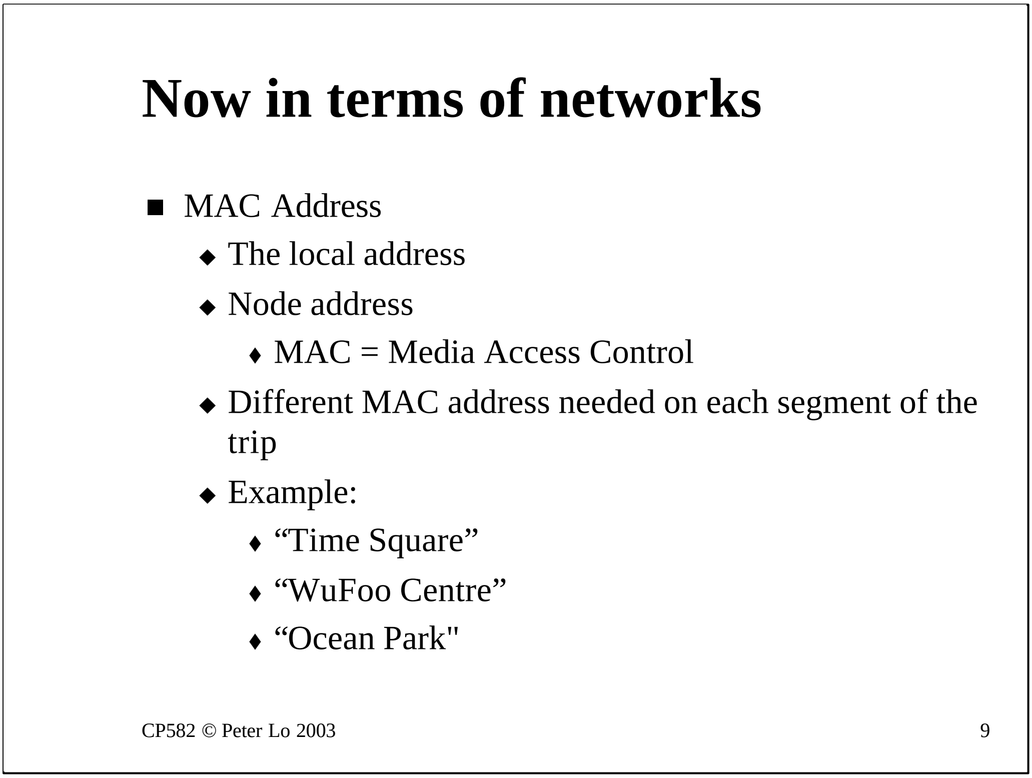
- Router
  - The interchange between two or more local networks (LANs)
- Logical Address
  - The address of the end-point
  - Must identify remote network and point on network

So, leaving analogy land, let’s transfer these ideas to computer networks.

The interchange at the borders is termed a **router.** This makes sense because we already know that decisions about routing need to made at this point.

The overall delivery address is termed a **logical address**. This is harder to make sense of. It needs to be seen in relation to MAC addresses which are “physical” in the sense that they are built in to the NIC. The ultimate destination will have a physical (MAC) address too but when we describe that destination from far away we do **not** describe it in terms of its MAC address - we use some other, multi-level scheme of addressing that can describe locations anywhere in the world. This is termed a “logical address” because it comes from a naming scheme that we have devised and have control over. It could be changed if the *logic* of the naming scheme were altered.

A good example of this was when Telstra changed the STD codes. The code for Ballarat changed from 053 to 0353. This was a *logical **(!!!)*** change - the physical addresses (the phone numbers within Ballarat) did not change.



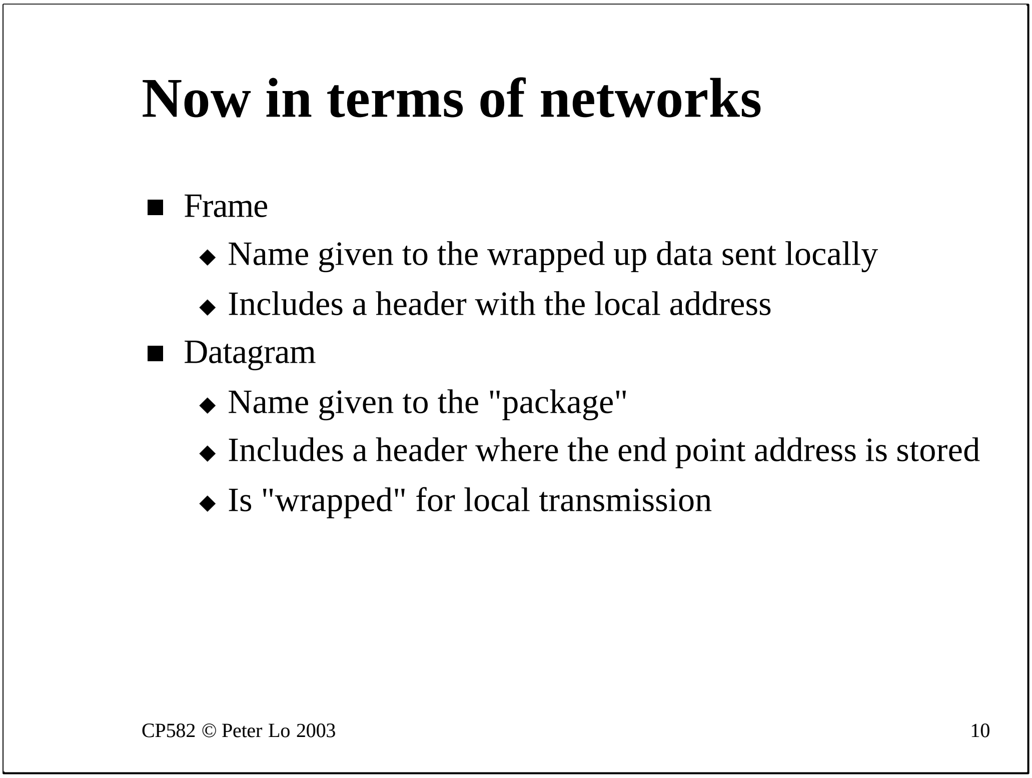
Now in terms of networks

- MAC Address
  - The local address
  - Node address
    - MAC = Media Access Control
  - Different MAC address needed on each segment of the trip
  - Example:
    - “Time Square”
    - “WuFoo Centre”
    - “Ocean Park"

The local address is the MAC address in networking terms.

The interesting realisation is to find that the packet needs a different destination MAC address for each local network that it crosses during its trip

But this should be no surprise because in the package analogy we have already seen that we needed a different local delivery address at each step along the way.



Now in terms of networks

- Frame
  - Name given to the wrapped up data sent locally
  - Includes a header with the local address
- Datagram
  - Name given to the "package"
  - Includes a header where the end point address is stored
  - Is "wrapped" for local transmission

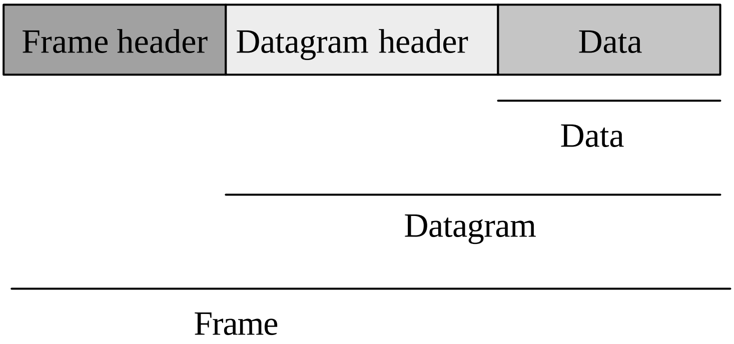
This slide clarifies what was said in an earlier lecture about the different names that are used to refer to “packets” on the network.

The term **frame** is exclusively used to refer to the data packet that is sent on the local network. The frame includes a **header** which takes care of the local delivery addressing.

The term **datagram** will be used in NPS to refer to the part of the frame that is sent to the remote destination. This too includes a **header** but in this header the remote addressing is stored.

As a datagram is sent through the network it is successively wrapped and unwrapped in a series of frame headers - one for each local transmission along the way.

Datagram and Frame


Frame header | Datagram header | Data

Data

Datagram

Frame

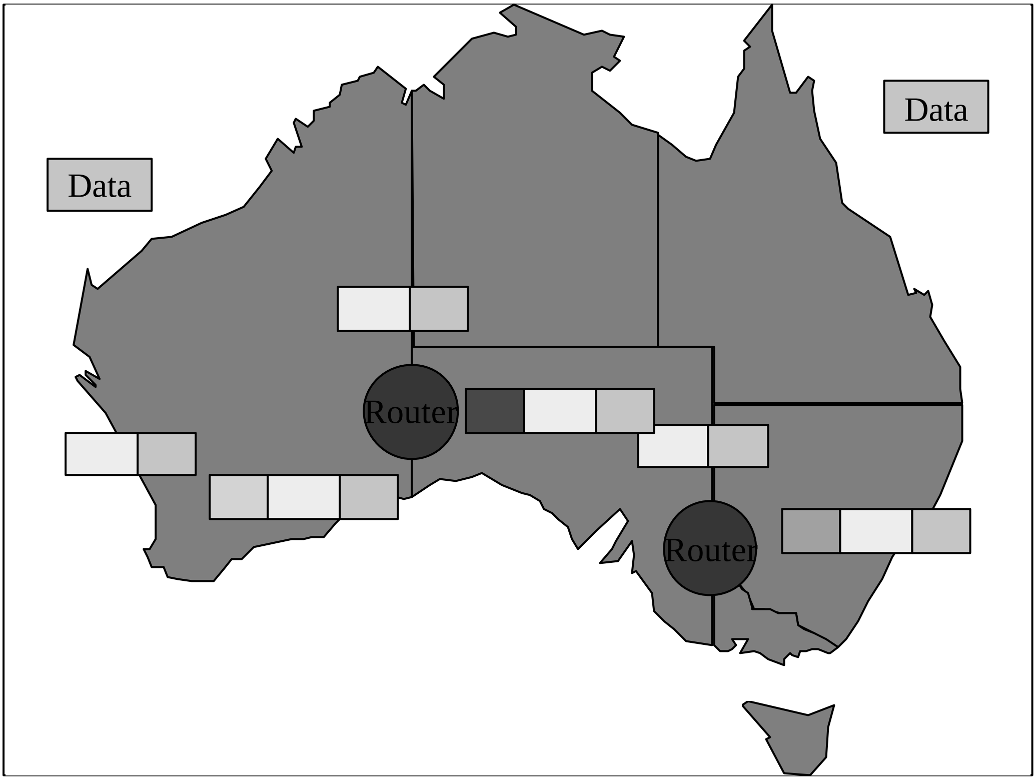
This slide, and the animation, is making a very fundamental point about networks.

The original data is being wrapped by a series of headers that each have a specific role to play in the overall transmission.

In this example the first layer of wrapping is the datagram header. The datagram header has the job of storing the overall, long distance address (e.g. Tsinghua University, Beijing, China). We now have a datagram whereas before we just had data.

The second layer of wrapping is the frame header. The frame header **has a distinct purpose**. It has the job of storing the addressing for the particular local delivery step that we are inspecting (eg Spencer St, Tullamarine etc). It can be discarded when the local delivery is complete.

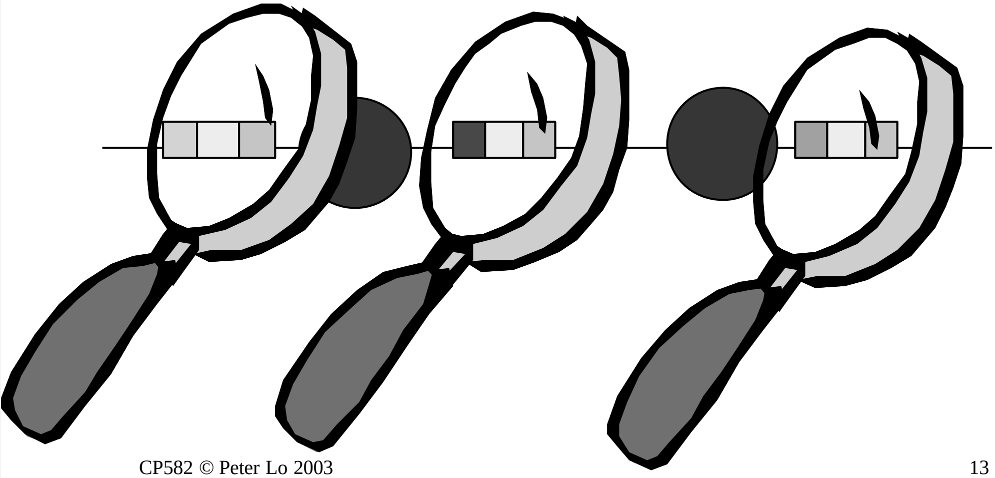
The animation in the next slide will make this clearer.



This animation spells the remote delivery of data out in detail.

The key concept to grasp is the way in which the frame header is discarded and replaced for each leg of the journey.

What does NetXRay see?



NetXRay is necessarily looking at the traffic on one particular network so depending on where we are connected when we run NetXRay we will see a slightly different picture - the same datagram but a distinct frame header.

Logical Address…..

- Wide Area Network
  - A collection of LANs that is interconnected
  - An Internetwork
  - Hence the "Internet"

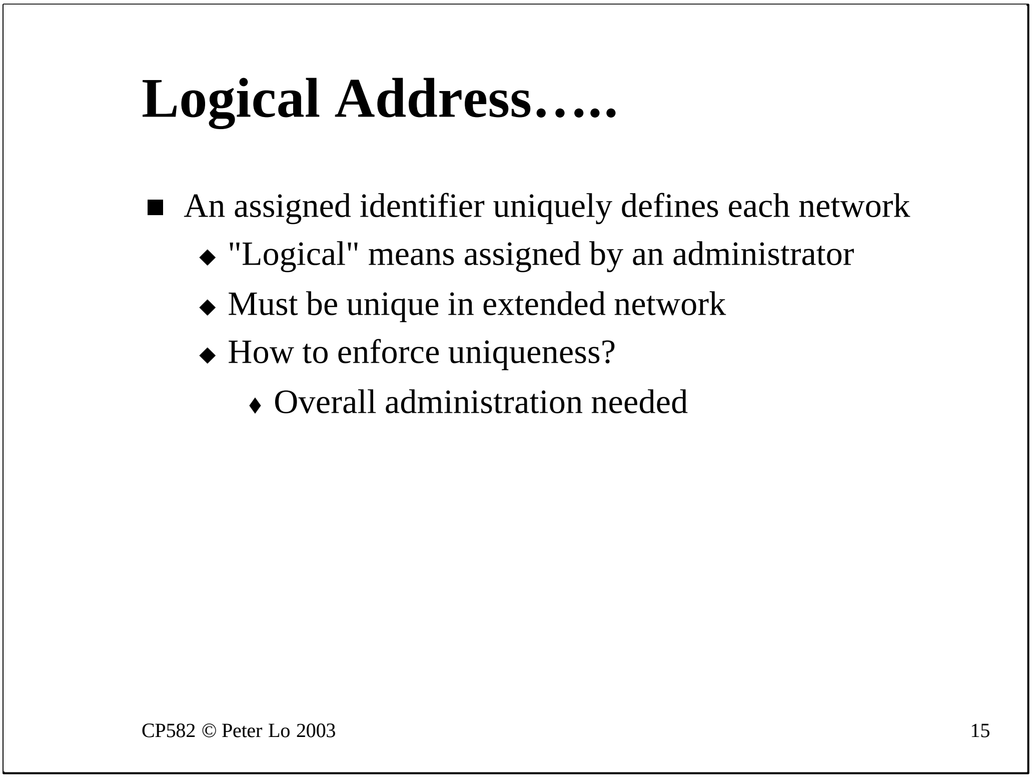
A couple of useful - but possibly confusing acronyms - LAN & WAN

By now you should be comfortable with the definition of a local network as one within which simple, single level addressing can be used. In the University all of ITMS is one local network whereas the rest of the campus is a separate local network. The two are connected together by a router - which in fact is the server ERIS. As a user of these networks you are probably unaware of the details of this structure because the whole thing is running at 10 or often 100 Mbits/sec.

The combined local networks at Mount Helen (there are several others - Residence for example) is termed the University LAN (Local Area Network). Here "local" is used to mean "can be reached quickly". So a LAN may include a number of "Local Networks" interconnected by routers and is technically therefore a small "internetwork"

When we start to connect further afield, with links that are often slower, we are creating a more extended internetwork that is termed a WAN (Wide Area Network). The combined campuses of the whole University (SMB, Wimmerra, Mount Helen) constitute a WAN. The Internet is clearly a WAN.

Just to confuse things further you may also hear of "MAN"s (Metropolitan Area Network) Some modern city centres are being fitted with very high speed cables between all the office buildings. Within one building we clearly have a LAN. If we connect multiple LANs together at LAN-type speeds what do we get? A bigger LAN? In that case we loose sight of the independent nature of each LAN - hence the term MAN.



Logical Address…..

- An assigned identifier uniquely defines each network
  - "Logical" means assigned by an administrator
  - Must be unique in extended network
  - How to enforce uniqueness?
    - Overall administration needed

Navigation in a WAN requires us to be able to identify the different regions and the names used here are not “physical” in the way that MAC addresses are. The names of networks are assigned by administrators and can be changed by administrators.

An address which can be administered in this way is termed a “Logical” address.

Specific Examples

- TCPIP
  - Addressing scheme used in the Internet
- IPX
  - Used mainly by Novell
- Appletalk
  - Macintosh networks

Each of the major protocol families has its own way of defining the logical address of a network.

Hence, if a particular local network is carrying a mixture of these protocols it can have several different names.

For example in the University the ITMS network has the following names:

- IPX - Network 8
- IP - Network 141.132.192.0

Each of these is a different name for exactly the same thing - the cable in our labs. It simply depends which protocol you are going to use to get there.

TCP/IP

- Single address identifier
  - A 32 bit number
  - For human consumption written in 141.132.64.2 format
    - 4 "octets" - each 8 bits
  - A logical address
    - Administratively assigned
      - Network Information Centre

TCP/IP

- Network/Node
  - Left hand part identifies network
  - Right hand part identifies node
- Telephone number analogy
  - Area code = network
    - Identifies location
  - Telephone number = node
    - Identifies phone within area

In a TCPIP address the left-hand part identifies the network - answering the question of **where** in the world to deliver this packet.

The right hand portion identifies the node - hence answering the question of **which computer** to deliver to on the destination network.

This kind of division should be familiar to us because it is exactly what is done with telephone numbers - area codes (on the left) say where in the world and the right hand portion says which phone to ring in the location specified.

The question of where the boundary lies between network and node address in a more advanced topic.

IPX

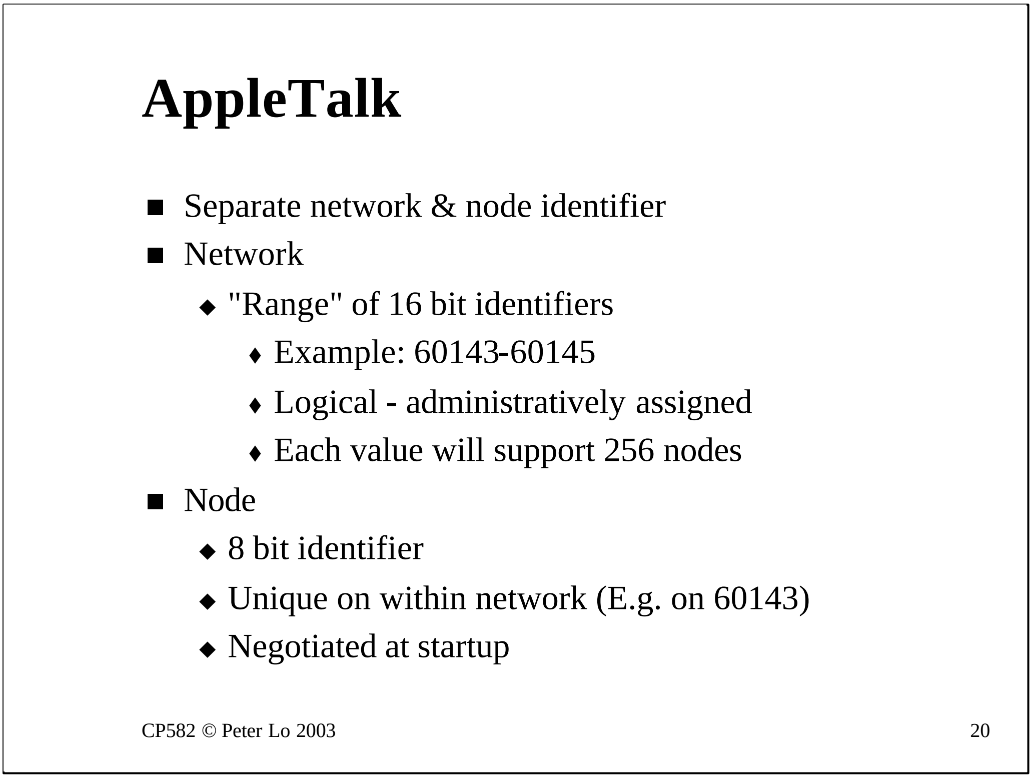
- Separate network & node identifier
- Network
  - 32 bit number
  - Logical - administratively assigned
- Node
  - MAC address of card
  - 48 bit number
  - Cannot be changed - built in to card

Novell's IPX protocol has a different approach.

Networks are identified by an administratively assigned 32bit number. There is no central organisation managing these numbers because in general IPX networks are not interconnected throughout the world.

The node address, once the network has been identified, is simply the 48 bit MAC address of the computer. Hence, in IPX, discovering a computers MAC address is not an issue - it is simply part of the WAN address of that computer.

There are problems, of course, if the NIC is changed!



AppleTalk

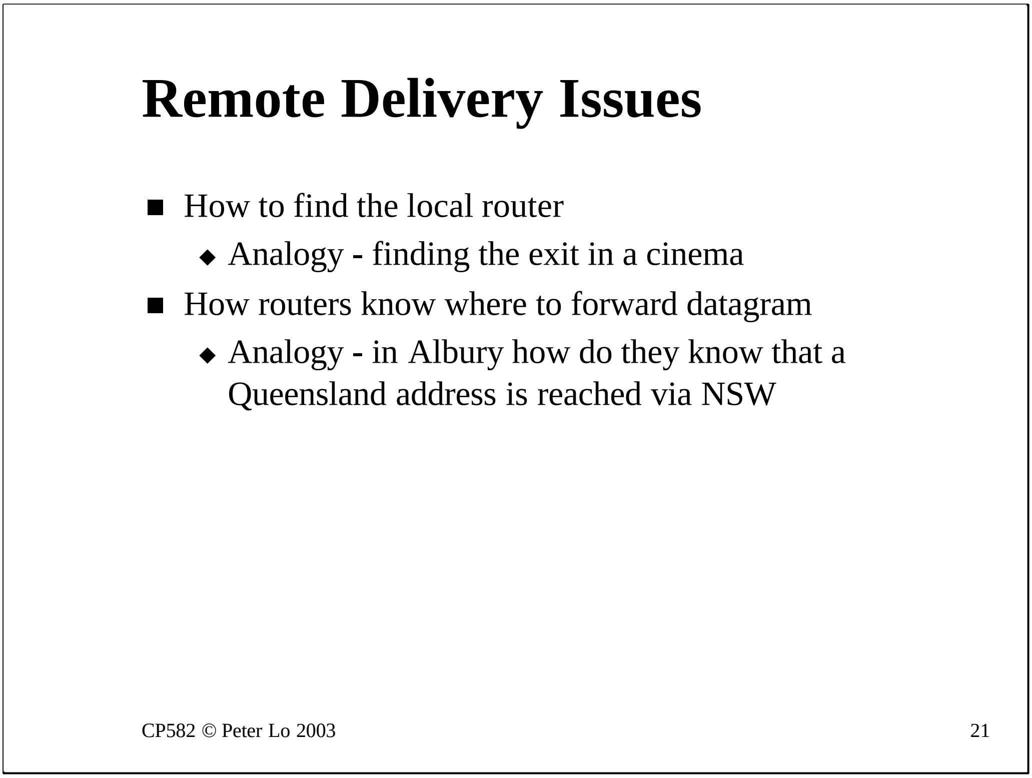
- Separate network & node identifier
- Network
  - "Range" of 16 bit identifiers
    - Example: 60143-60145
    - Logical - administratively assigned
    - Each value will support 256 nodes
- Node
  - 8 bit identifier
  - Unique on within network (E.g. on 60143)
  - Negotiated at startup

Apple Computers have their own unique approach which is complicated by the longer history of networking for the Mac.

The original Mac had networking built in at a time (1980) when networks were quite unusual. It seemed to the developers of the Mac, as if 256 computers on a network would be plenty so Appletalk networking uses an 8 bit identifier for the nodes.

Of course networks grew beyond 256 computers so current Appletalk networking (called Phase 2) uses a range of 16 bit identifiers to describe a single network segment. Each additional number that is used can support an extra 256 Macs!

The 8 bit node address for a Mac is negotiated at startup. Each Mac broadcasts the address it would like to use (usually the one it used yesterday) and the any existing Mac on the network needs to reply is they are already using that address.



Remote Delivery Issues

- How to find the local router
  - Analogy - finding the exit in a cinema
- How routers know where to forward datagram
  - Analogy - in Albury how do they know that a Queensland address is reached via NSW

These are the two major issues that we need to resolve if remote delivery is going to work.

Finding the local router

In order for a node to send a datagram on its way across the world it needs to know where the local "exit" is - the local address of the router that connects this network to the rest of the world.

The analogy is finding the exit in a large building.

What is less easy to see is that this issue of finding the nest router is one that needs to be resolved at every step along the way. For a datagram life is one long search for the next router!

Forwarding the datagram

This issue can only be resolved by the router itself. Assuming that the router has more than 2 connections coming into it each datagram that arrives is going to require a routing decision to be made.

This decision is based upon reading the ultimate destination in the Datagram Header and then deciding which way this particular Datagram should be sent. How to make this decision is the main topic for the rest of this lecture.

Finding the Local Router

- Static definition
  - Administrative setting in each node
  - May be transmitted automatically to node
    - BOOTP
      - Boot Protocol
      - Protocol in TCPIP through which node discover their settings
    - DHCP
      - Dynamic Host Configuration Protocol
      - Many different settings can be discovered
      - "Lease" of address

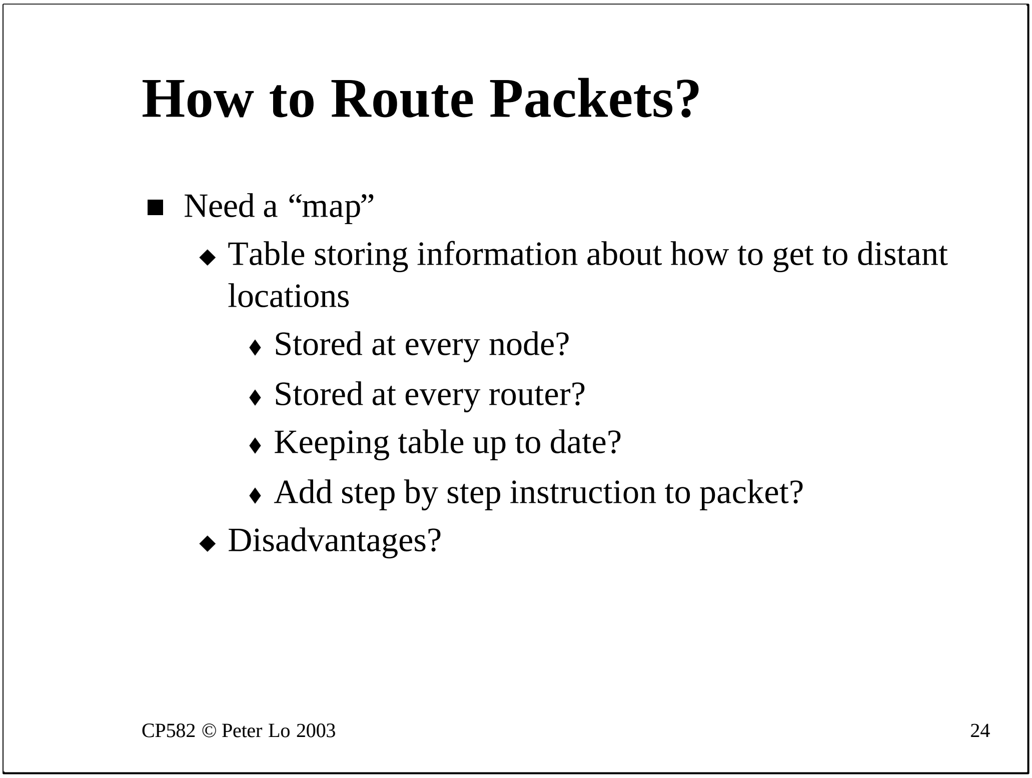
The first question - finding the local router - has a variety of possible solutions.

It is possible for all nodes to be statically configured with the address of their nearest router. By static we mean that the setting has been made by an administrator - either directly at the node or else in a table somewhere that the node is able to read remotely by means of a special protocol.

Finding the Local Router

- Dynamic location
  - Router responds to requests for service
  - Protocol to discover the nearest router
  - IPX ”Give route”

Alternatively there are protocols that can be used to get the router to tell us its address directly. This is “dynamic” in the sense that if one router goes down we can still use the protocol to find another router.



How to Route Packets?

- Need a “map”
  - Table storing information about how to get to distant locations
    - Stored at every node?
    - Stored at every router?
    - Keeping table up to date?
    - Add step by step instruction to packet?
  - Disadvantages?

Making the routing decision is a much more complex issue which raises all kinds of questions.

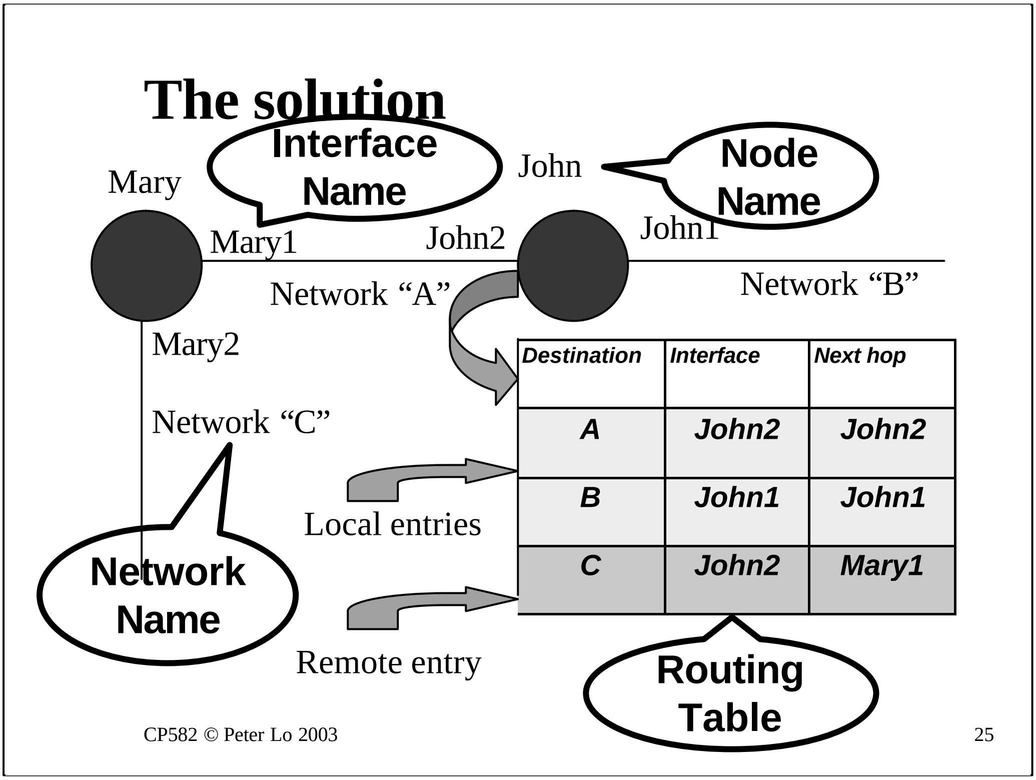
Clearly we need the router to have access to some kind of map but how to accomplish this.

Here are some suggestions:

• Give every node a complete map of the network. The node could use this map to add a complete set of routing instructions to every datagram it sends thereby letting the routers off the hook as far as making any decisions is concerned.

• Give every router a complete map of the network so that it can make the decisions.

Both of these map based solutions beg the question of how the map is going to be kept up to date and also how big the map might get as the network grows. How big would the map be for the whole Internet?



The solution

Destination	*Interface*	*Next hop*
A	***John2***	***John2***
B	***John1***	***John1***
C	***John2***	***Mary1***

The most widely adopted solution to the storage of network maps is the **routing table**

Routing tables are stored at routers and indicate the **next step** to be taken in order to reach each network destination.

There are two aspects to this next step:

- **Which “exit” to go out of**

  In the theatre analogy this might be an instruction to take the left hand exit in the auditorium.

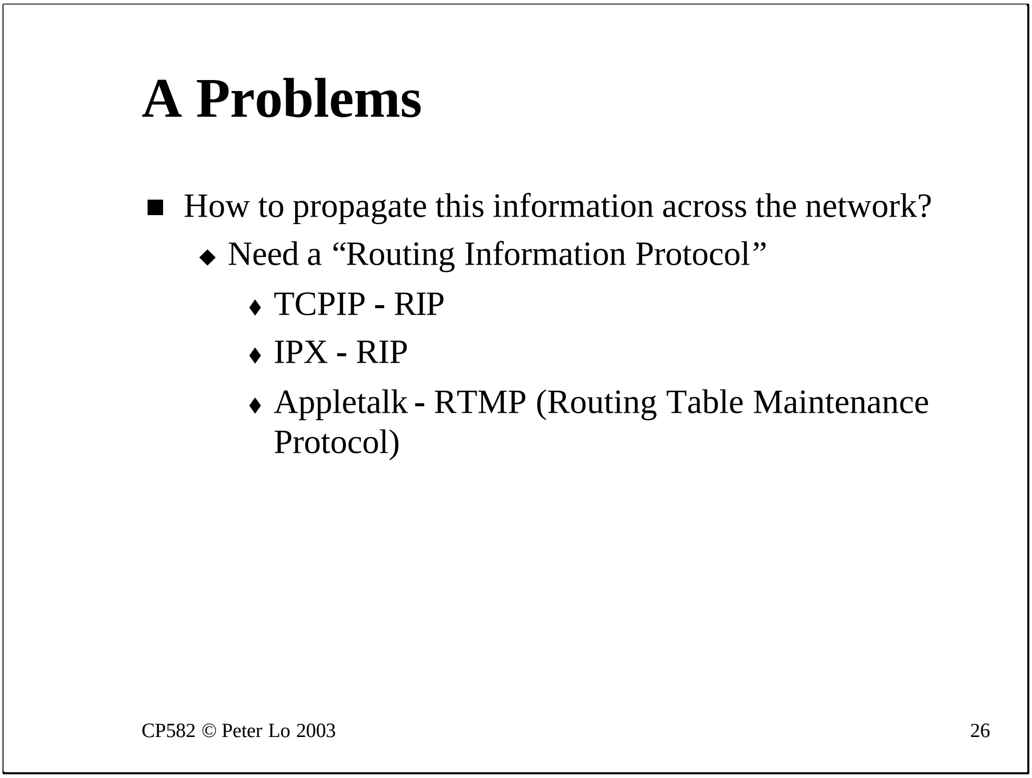
  In a router this is an indication of which “interface” to use. This is simply a name of some kind to distinguish between several NICs that are installed in the router. In NetSim each NIC is named after the computer - with a number added at the end. The slide shows the NetSim approach.

- **Which way to go next**

  In the theatre analogy this might be a instruction such as “turn right after you go through the door” or “go up the stairs”

  In the router this is the name of the next step along the path to the final destination. In the slide there is only one possible way to go to get to network C but if the network were more complex then statement “Go to Mary1 to get to network C” would be crucial.

  Notice that the “next hop” is always **a local destination that can be reached via the interface**. This is important because “local

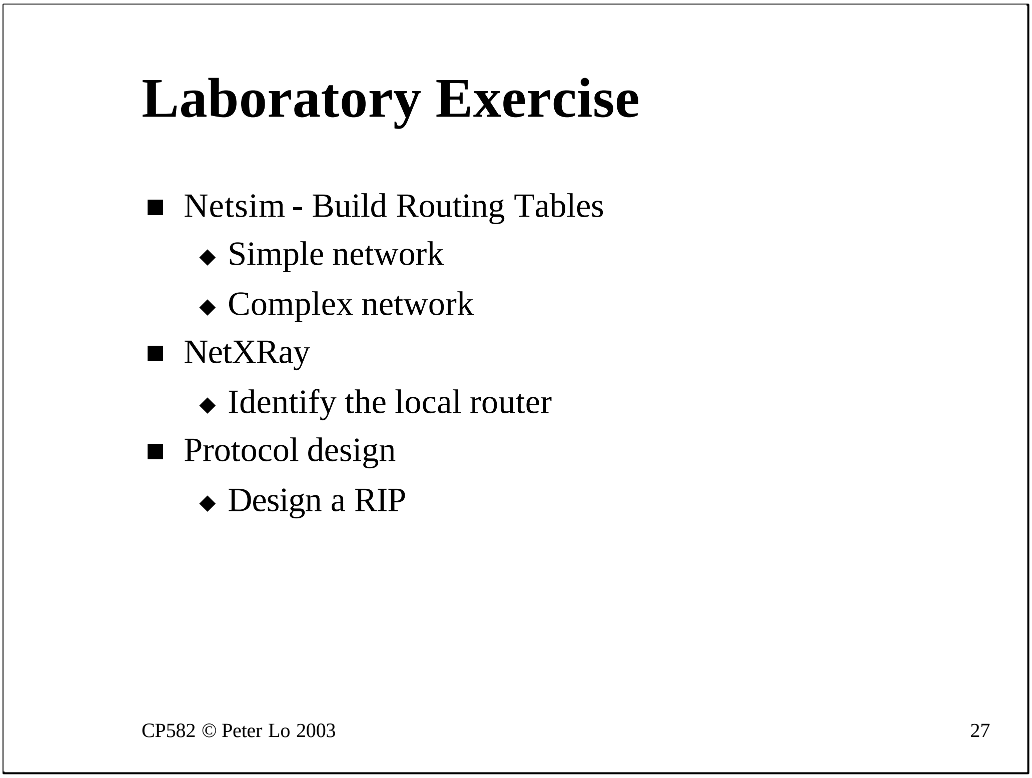


A Problems

- How to propagate this information across the network?
  - Need a “Routing Information Protocol”
    - TCPIP - RIP
    - IPX - RIP
    - Appletalk - RTMP (Routing Table Maintenance Protocol)

To keep their routing tables up-to-date, and to acquire them in the first place, routers need to talk to each other.

Each protocol family has its own protocol for doing this.



Laboratory Exercise

- Netsim - Build Routing Tables
  - Simple network
  - Complex network
- NetXRay
  - Identify the local router
- Protocol design
  - Design a RIP

The lab has three components:

• A NetSim exercise where you will install two cards in your virtual computer. With two card in a computer it is possible to build up a NetSim internetwork. The object of the exercise is to design and build an internetwork and then get the routing tables right in order that communication throughout the internetwork is possible.

• In NetXRay you will look at packets that are heading off the local network and figure out the MAC address of the local router.

• As a protocol design exercise you will be thinking about how to design a routing table protocol.