

Bridging and (versus) routing

David Morgan

© David Morgan 2007

Given this setup...

LEFT

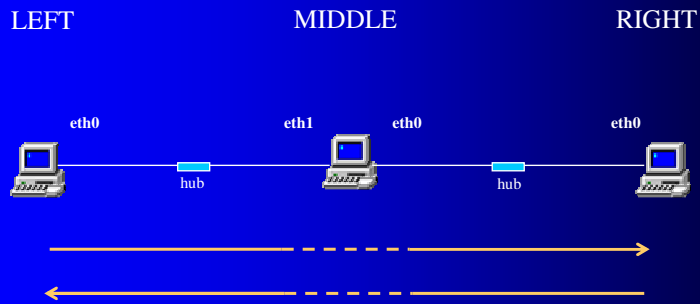
MIDDLE

RIGHT



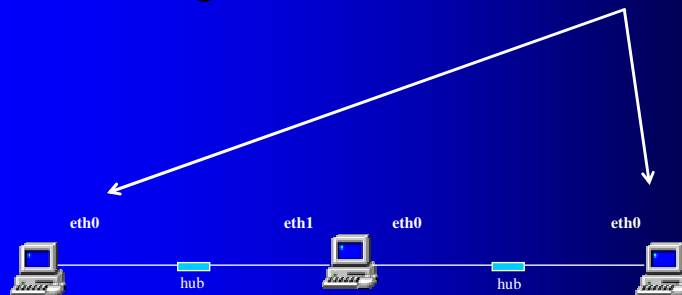
© David Morgan 2007

... we want LEFT to ping RIGHT
and get a reply



© David Morgan 2007

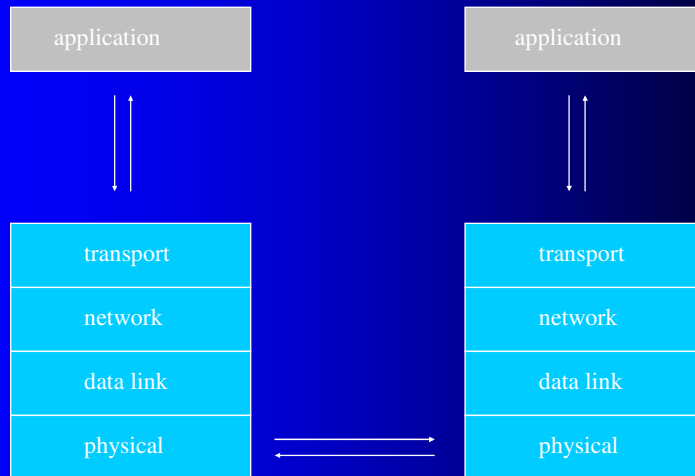
... 2 configs could make 'em ping



1. **Routing**
make 2 LANs out of it (2 broadcast domains)
end-to-end connection achieved by *routing* the *IP packets*
2. **Bridging**
make 1 consolidated LAN out of it (single broadcast domain)
end-to-end by *bridging* the *ethernet frames*

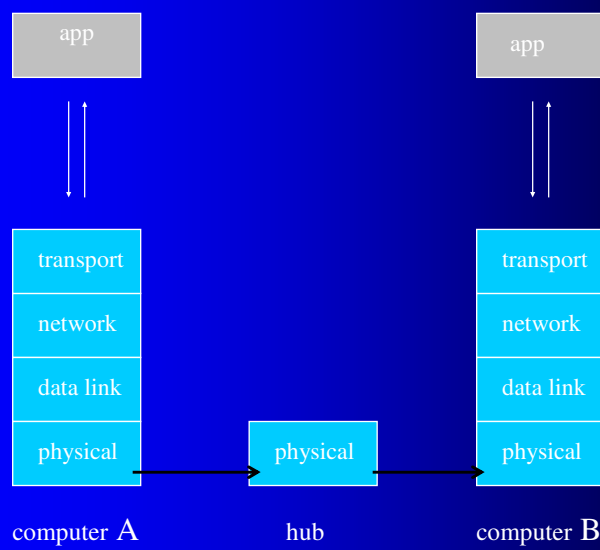
© David Morgan 2007

Info's usual trans-layer itinerary

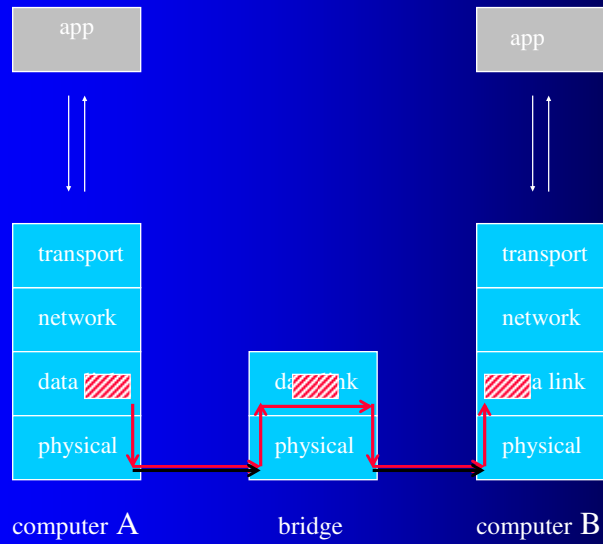


© David Morgan 2007

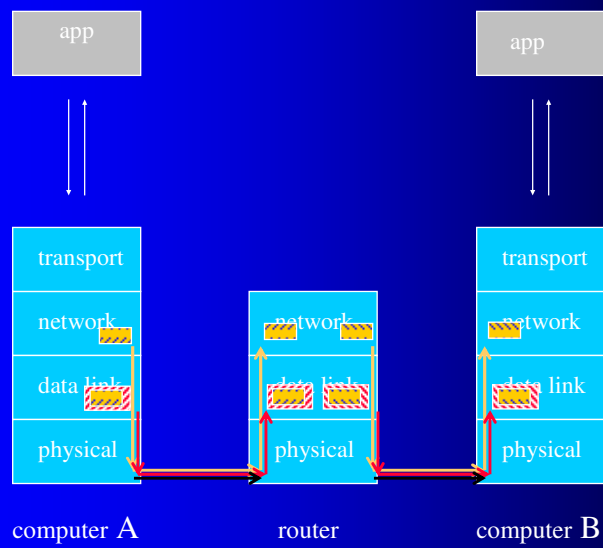
Signals via hub ("layer 1 device")



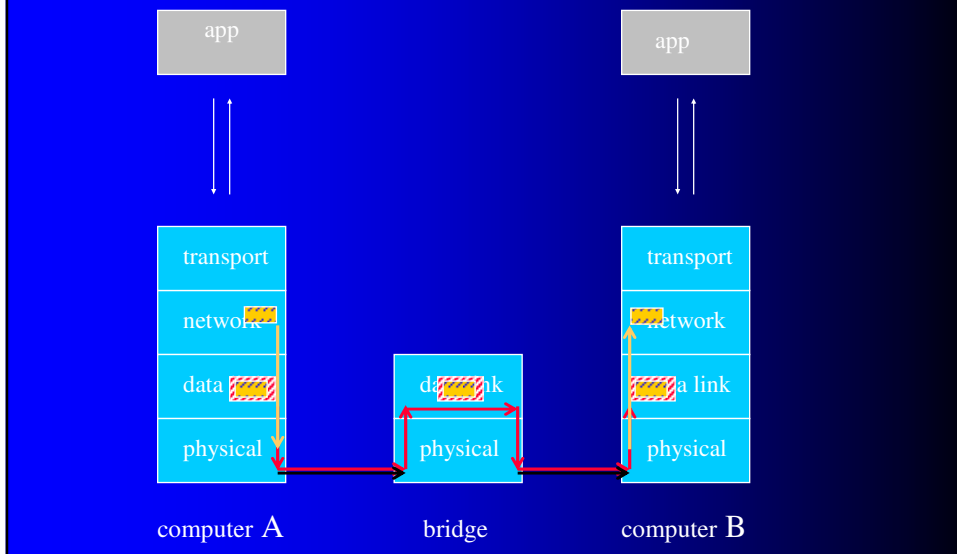
Frames via bridge ("layer 2 device")



Packets via router ("layer 3 device")



Note, bridge scenario: frame's contained packet untouched



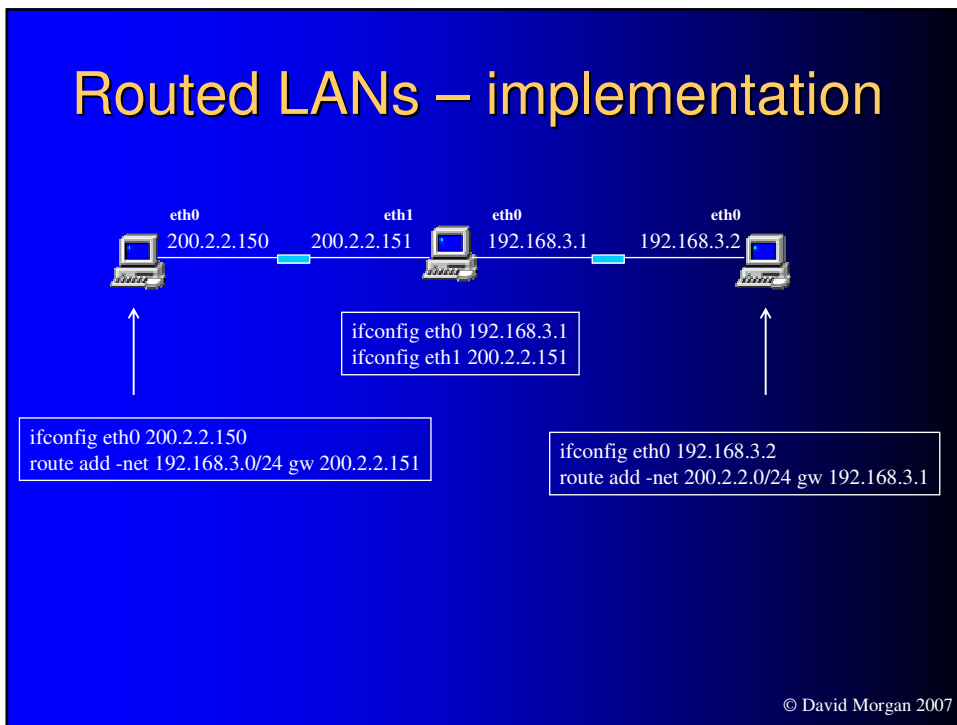
1. Two LANs – routed connection end-to-end

Left Network – 200.2.2.150/24

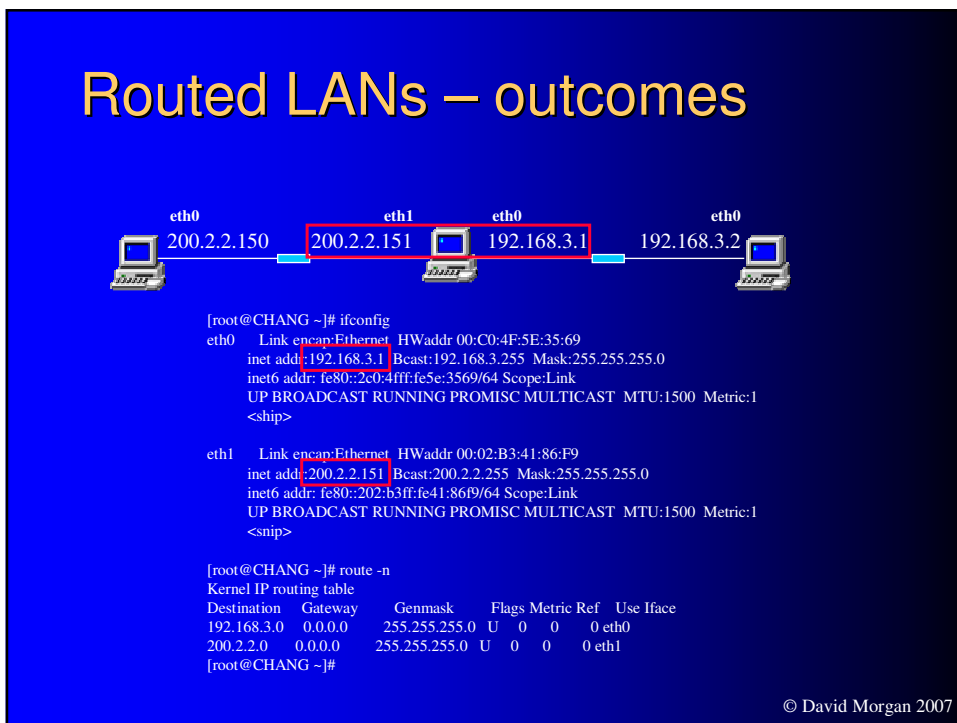


Right Network – 192.168.3.0/24

Routed LANs – implementation



Routed LANs – outcomes



Routed LANs – outcomes



```
[root@emach4 ~]# ifconfig
eth0  Link encap:Ethernet  HWaddr 00:40:CA:B4:E3:FC
       inet addr:200.2.2.150  Bcast:200.2.2.255  Mask:255.255.255.0
       inet6 addr: fe80::240:caff:feb4:e3fc/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:220 errors:0 dropped:0 overruns:0 frame:0
       TX packets:282 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:25015 (24.4 KiB)  TX bytes:32463 (31.7 KiB)
       Interrupt:18 Base address:0x6000

[emach4 ~]# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.3.0 200.2.2.151 255.255.255.0 UG 0 0 0 eth0
200.2.2.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

[emach4 ~]#

[emach4 ~]# ifconfig
eth0  Link encap:Ethernet  HWaddr 00:60:08:96:AB:B2
       inet addr:192.168.3.2  Bcast:192.168.3.255  Mask:255.255.255.0
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:202 errors:0 dropped:0 overruns:0 frame:0
       TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:100
       RX bytes:20514 (20.0 Kb)  TX bytes:3678 (3.5 Kb)
       Interrupt:5 Base address:0xd400

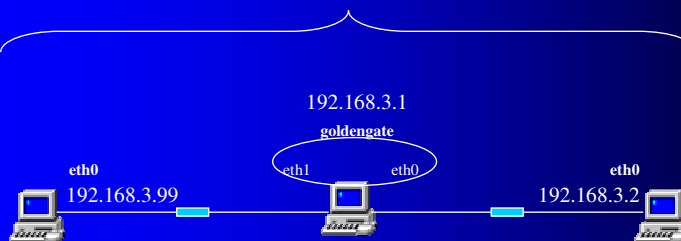
[emach4 ~]# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.3.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
200.2.2.0 192.168.3.1 255.255.255.0 UG 0 0 0 eth0

[emach4 ~]#
```

© David Morgan 2007

2. Single LAN – bridged connection end-to-end

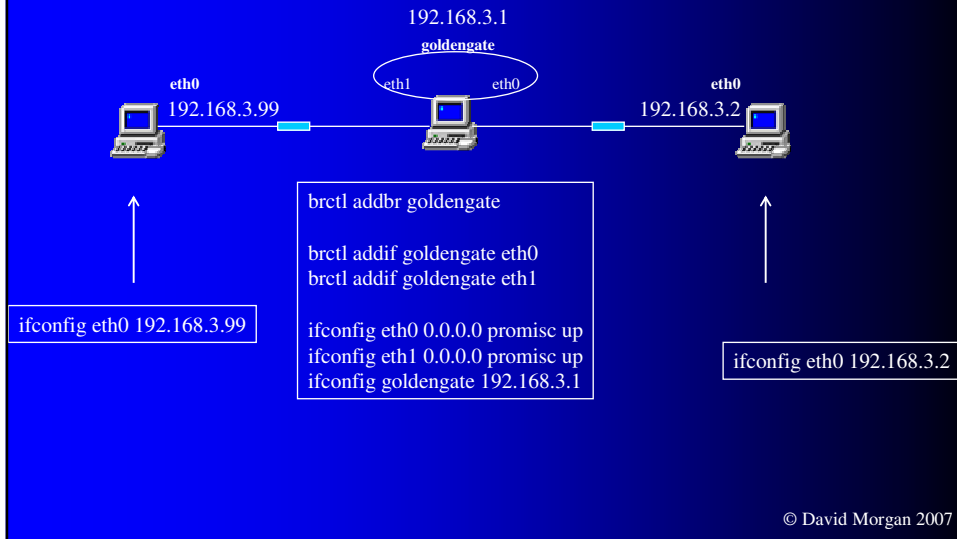
Sole Network – 192.168.3.0/24



NOTE: *physical NICs eth0/eth1 get no IP addresses
virtual NIC goldengate gets one (not two)*

© David Morgan 2007

Bridged LAN – implementation



Bridged LAN – outcomes



```

[root@CHANG ~]# ifconfig
eth0  Link encap:Ethernet HWaddr 00:C0:4F:5E:35:69
      inet6 addr: fe80::2e0:4fff:fe5e:3569/64 Scope:Link
      UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
      <snip>

eth1  Link encap:Ethernet HWaddr 00:02:B3:41:86:F9
      inet6 addr: fe80::202:b3ff:fe41:86f9/64 Scope:Link
      UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
      <snip>

goldengate Link encap:Ethernet HWaddr 00:02:B3:41:86:F9
      inet addr: 192.168.3.1  Bcast:192.168.3.255  Mask:255.255.255.0
      inet6 addr: fe80::202:b3ff:fe41:86f9/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      <snip>

[root@CHANG ~]# route -n
Kernel IP routing table
Destination  Gateway      Genmask     Flags Metric Ref  Use Iface
192.168.3.0  0.0.0.0     255.255.255.0  U    0    0    0 goldengate
[root@CHANG ~]#
    
```

© David Morgan 2007

Bridged LAN – outcomes



```
[root@emach4 ~]# ifconfig
eth0  Link encap:Ethernet  HWaddr 00:40:CA:B4:E3:FC
       inet addr:192.168.3.99  Bcast:192.168.3.255  Mask:255.255.255.0
       inet6 addr: fe80::240:caff:feb4:e3fc/64  Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:185 errors:0 dropped:0 overruns:0 frame:0
       TX packets:81 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:20772 (20.2 KiB)  TX bytes:8983 (8.7 KiB)
       Interrupt:18 Base address:0x6000
```

```
[root@emach4 ~]# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.3.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
[root@emach4 ~]#
```

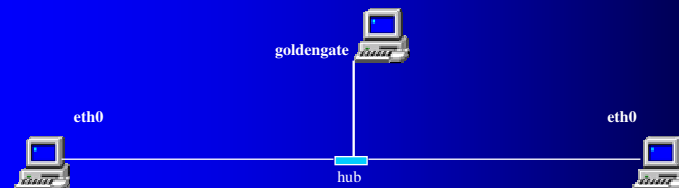
```
[root@rh root]# ifconfig
eth0  Link encap:Ethernet  HWaddr 00:60:08:96:AB:B2
       inet addr:192.168.3.2  Bcast:192.168.3.255  Mask:255.255.255.0
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:170 errors:0 dropped:0 overruns:0 frame:0
       TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:100
       RX bytes:16533 (16.1 Kb)  TX bytes:2436 (2.3 Kb)
       Interrupt:5 Base address:0xd400
```

```
[root@rh root]# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.3.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
[root@rh root]#
```

Bridged LAN – conceptual physical equivalent



becomes:



1. eth0/eth1 interfaces pulled
2. goldengate interface inserted
3. hub fashioned from old eth0/eth1 hardware

© David Morgan 2007

Who's sending to whom, and what

- the fields that give a clue -

- ethernet sources and destinations
- IP sources and destinations
- IP's header checksum

"A checksum on the header only. Since some header fields change (e.g., time to live), this is recomputed and verified at each point that the internet header is processed." rfc 791
- IP's time-to-live

"This field is modified in internet header processing. ...every module that processes a datagram must decrease the TTL by at least one..." rfc 791
- ICMP/ping's message checksum

© David Morgan 2007

ping's itinerary by router*

| | LEFT eth0 | MIDDLE eth1 | MIDDLE eth0 | RIGHT |
|----------------------|-------------------|-------------------|-------------------|-------------------|
| Request | | | | |
| ethernet source | 00:40:ca:b4:e3:fc | 00:40:ca:b4:e3:fc | 00:c0:4f:5e:35:69 | 00:c0:4f:5e:35:69 |
| ethernet destination | 00:02:b3:41:86:f9 | 00:02:b3:41:86:f9 | 00:60:08:96:ab:b2 | 00:60:08:96:ab:b2 |
| IP source | 200.2.2.150 | 200.2.2.150 | 200.2.2.150 | 200.2.2.150 |
| IP destination | 192.168.3.2 | 192.168.3.2 | 192.168.3.2 | 192.168.3.2 |
| IP checksum | ac66 | ac66 | ad66 | ad66 |
| IP time-to-live | 64 | 64 | 63 | 63 |
| ICMP checksum | 9cba | 9cba | 9cba | 9cba |
| Reply | | | | |
| ethernet source | 00:02:b3:41:86:f9 | 00:02:b3:41:86:f9 | 00:60:08:96:ab:b2 | 00:60:08:96:ab:b2 |
| ethernet destination | 00:40:ca:b4:e3:fc | 00:40:ca:b4:e3:fc | 00:c0:4f:5e:35:69 | 00:c0:4f:5e:35:69 |
| IP source | 192.168.3.2 | 192.168.3.2 | 192.168.3.2 | 192.168.3.2 |
| IP destination | 200.2.2.150 | 200.2.2.150 | 200.2.2.150 | 200.2.2.150 |
| IP checksum | 4c41 | 4c41 | 4b41 | 4b41 |
| IP time-to-live | 63 | 63 | 64 | 64 |
| ICMP checksum | a4ba | a4ba | a4ba | a4ba |

*data from Wireshark capture © David Morgan 2007

ping's itinerary by bridge*

| | LEFT eth0 | eth1 | MIDDLE goldengate | eth0 | RIGHT eth0 |
|----------------------|-------------------|-------------------|----------------------|-------------------|-------------------|
| identical | | | | | |
| Request | | | | | |
| ethernet source | 00:40:ca:b4:e3:fc | 00:40:ca:b4:e3:fc | 00:40:ca:b4:e3:fc | 00:40:ca:b4:e3:fc | 00:40:ca:b4:e3:fc |
| ethernet destination | 00:60:08:96:ab:b2 | 00:60:08:96:ab:b2 | 00:60:08:96:ab:b2 | 00:60:08:96:ab:b2 | 00:60:08:96:ab:b2 |
| IP source | 192.168.3.99 | 192.168.3.99 | 192.168.3.99 | 192.168.3.99 | 192.168.3.99 |
| IP destination | 192.168.3.2 | 192.168.3.2 | 192.168.3.2 | 192.168.3.2 | 192.168.3.2 |
| IP checksum | b2f3 | b2f3 | b2f3 | b2f3 | b2f3 |
| IP time-to-live | 64 | 64 | 64 | 64 | 64 |
| ICMP checksum | 91d0 | 91d0 | 91d0 | 91d0 | 91d0 |
| Reply | | | | | |
| ethernet source | 00:60:08:96:ab:b2 | 00:60:08:96:ab:b2 | 00:60:08:96:ab:b2 | 00:60:08:96:ab:b2 | 00:60:08:96:ab:b2 |
| ethernet destination | 00:40:ca:b4:e3:fc | 00:40:ca:b4:e3:fc | 00:40:ca:b4:e3:fc | 00:40:ca:b4:e3:fc | 00:40:ca:b4:e3:fc |
| IP source | 192.168.3.2 | 192.168.3.2 | 192.168.3.2 | 192.168.3.2 | 192.168.3.2 |
| IP destination | 192.168.3.99 | 192.168.3.99 | 192.168.3.99 | 192.168.3.99 | 192.168.3.99 |
| IP checksum | fc14 | fc14 | fc14 | fc14 | fc14 |
| IP time-to-live | 64 | 64 | 64 | 64 | 64 |
| ICMP checksum | 99d0 | 99d0 | 99d0 | 99d0 | 99d0 |

*data from Wireshark capture © David Morgan 2007

A doubtful baseball analogy— infield cutoff man

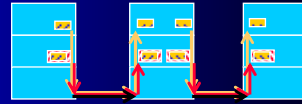
- Routing is like...
 - shortstop catches the ball from the outfielder
 - turns and makes a second, relay throw to the catcher
 - the throw caught by the catcher is *the shortstop's*
- Bridging is like...
 - shortstop sees the outfielder's throw strong and true
 - stands aside to let it go through on the bounce
 - the throw caught by the catcher is *the outfielder's*

© David Morgan 2007

You don't like my analogies? OK.

- Routing is like...

- bridge machine catches the incoming packet
- makes a new (slightly) modified one
- puts it in a new frame that he makes and sends
- the frame received by the destination endpoint is *the bridge machine's*



- Bridging is like...

- bridge machine is indifferent to the incoming frame
- copies frame between interfaces without touching it
- the frame received by the destination endpoint is *the source endpoint's*



© David Morgan 2007

Don't just take it from me...

“An ethernet bridge is a device commonly used to connect different networks of ethernets together, so that these ethernets will appear as one ethernet to the participants.

“Each of the ethernets being connected corresponds to one physical interface in the bridge. These individual ethernets are bundled into one bigger ('logical') ethernet, this bigger ethernet corresponds to the bridge network interface....

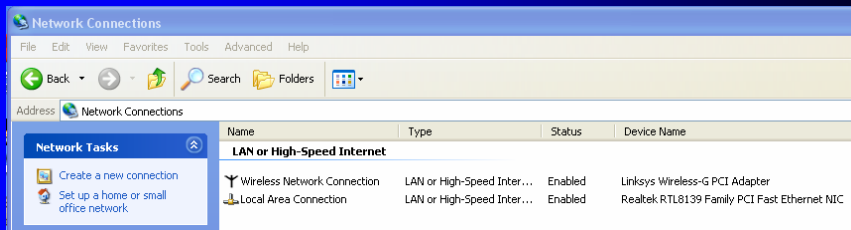
“Each bridge has a number of ports attached to it. Network traffic coming in on any of these ports will be forwarded to the other ports transparently, so that the bridge is invisible to the rest of the network...”

-- manpage for linux brctl command

© David Morgan 2007

Windows similarly - prebridge

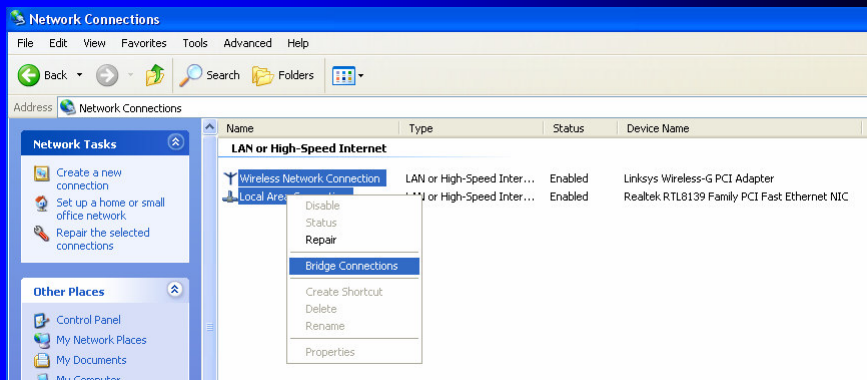
2 interfaces (aka "adapters") to start with:



© David Morgan 2007

Windows – creating the bridge

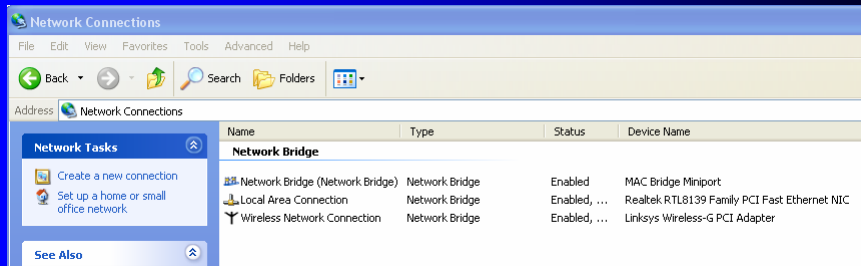
combine them into a bridge:



© David Morgan 2007

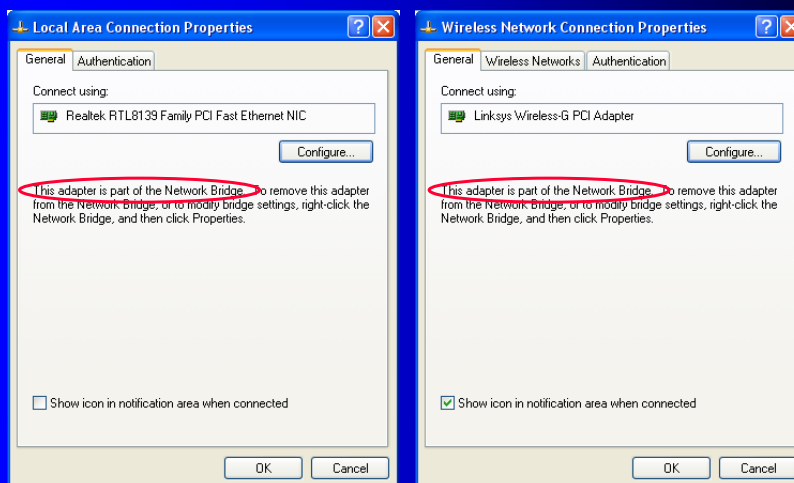
Windows - postbridge

new “Network Bridge” interface subsumes the 2 interfaces



© David Morgan 2007

Bridge “absorbs” interfaces



© David Morgan 2007