IP-in-IP encapsulation: basic tunneling

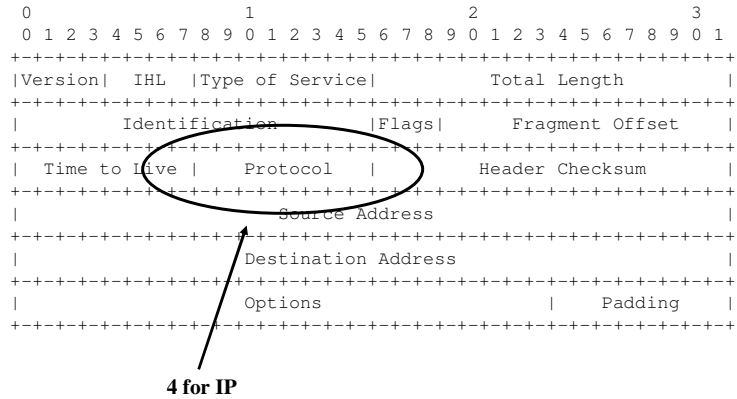
David Morgan

What is it?

- Conveys an IP packet between machines
  - … not as a packet
  - … but as cargo in another packet
- Destination shucks carrier packet, releases cargo as packet into local networking machinery
- "Tunnel" since one packet "passes through" another
- Implemented in linux by module ipip.o

IP itself is an IP subprotocol

IP Header Format

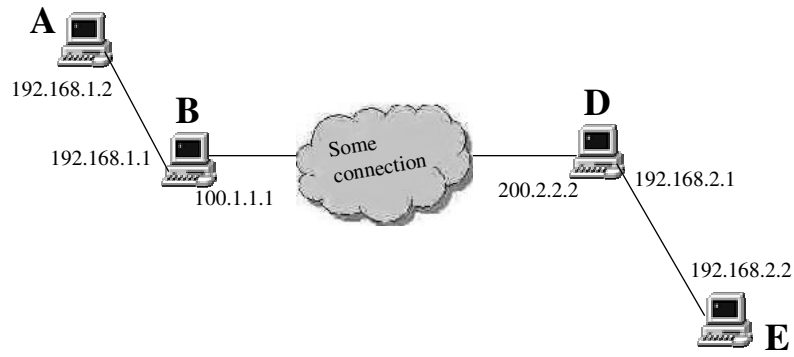


4 for IP
(6 for TCP
17 for UDP
50 for ESP, etc)

Class LAN

Local Network – 192.168.1.0

Remote Network – 192.168.2.0



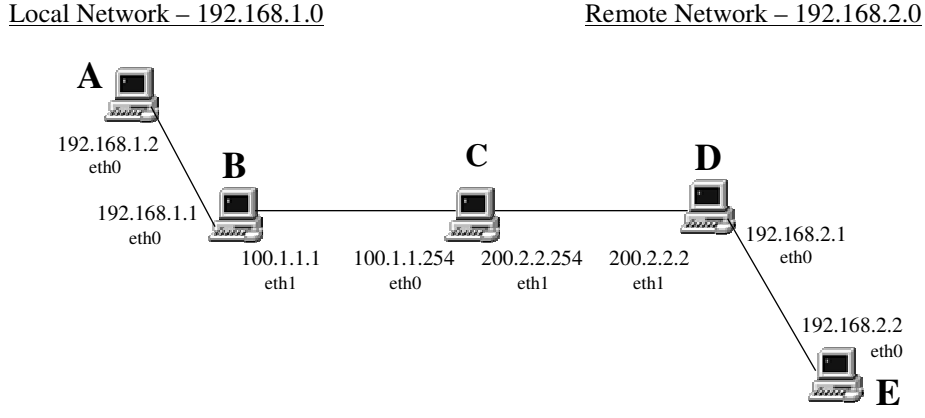
Workstations – A and E

Gateways – B and D

"Some connection"

- Could be the internet
- Could be a single intermediate machine
- Equivalent, for the 2 gateways

Class LAN

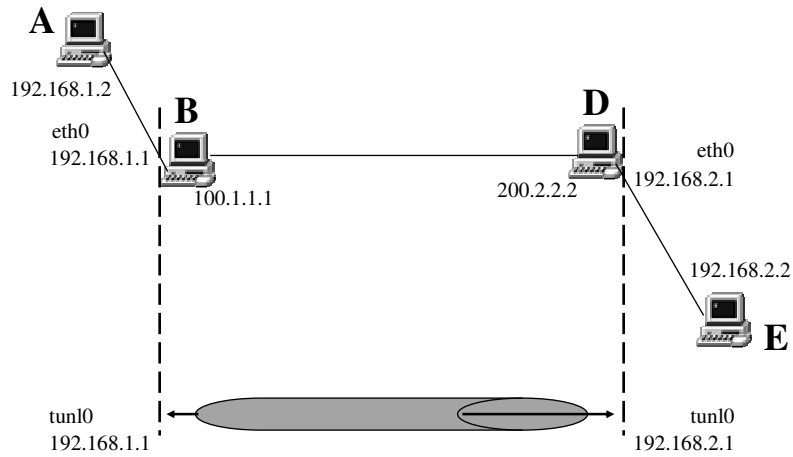


Workstations – A and E
Gateways – B and D
Internet surrogate – C (B's ISP; D's ISP)

Wanted: a 2nd bridge to cross

Local Network – 192.168.1.0

Remote Network – 192.168.2.0



Essential steps (on gateway)

- Load ipip.o module (enabling code)
- Apply internal interface's address to "tunl0" device
- Add gatewayed route to other network
  - specify other gateway by its outside IP
  - specify tunl0 device
- Reciprocate on opposite gateway

(/root/bin/tunl/tunl-vpnb)
(/root/bin/tunl/tunl-vpnd)

Load ipip.o module (if not automatic)

vpnd	modprobe ipip
vpnb	modprobe ipip

Internal IP goes onto tunl0

vpnd	ifconfig tunl0 192.168.2.1
vpnb	ifconfig tunl0 192.168.1.1

Route to opposite network via its gateway's outside IP, tunl0

vpnd	route add –net 192.168.1.0 netmask 255.255.255.0 gw 100.1.1.1 tunl0
vpnb	route add –net 192.168.2.0 netmask 255.255.255.0 gw 200.2.2.2 tunl0

Implementation - vpnb

vpnb

```
[root@vpnb root]# route -n
Kernel IP routing table
Destination     Gateway       Genmask         Iface
192.168.1.0     0.0.0.0       255.255.255.0   eth0
100.1.1.0       0.0.0.0       255.255.255.0   eth1
127.0.0.0       0.0.0.0       255.0.0.0       lo
0.0.0.0         100.1.1.254   0.0.0.0         eth1
[root@vpnb root]# ./tunl-vpnb
[root@vpnb root]# route -n
Kernel IP routing table
Destination     Gateway       Genmask         Iface
192.168.2.0     200.2.2.2     255.255.255.0   tunl0
192.168.1.0     0.0.0.0       255.255.255.0   eth0
100.1.1.0       0.0.0.0       255.255.255.0   eth1
127.0.0.0       0.0.0.0       255.0.0.0       lo
0.0.0.0         100.1.1.254   0.0.0.0         eth1
```

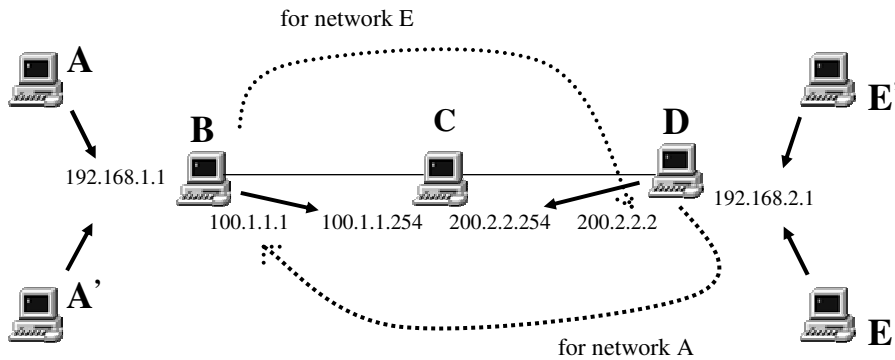
Implementation - vpnd

vpnd

```
[root@vpnd root]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Iface
192.168.2.0     0.0.0.0         255.255.255.0   eth0
200.2.2.0       0.0.0.0         255.255.255.0   eth1
127.0.0.0       0.0.0.0         255.0.0.0       lo
0.0.0.0         200.2.2.254     0.0.0.0         eth1
[root@vpnd root]# ./tunl-vpnd
[root@vpnd root]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Iface
192.168.2.0     0.0.0.0         255.255.255.0   eth0
192.168.1.0     100.1.1.1       255.255.255.0   tunl0
200.2.2.0       0.0.0.0         255.255.255.0   eth1
127.0.0.0       0.0.0.0         255.0.0.0       lo
0.0.0.0         200.2.2.254     0.0.0.0         eth1
```


Routing

Local Network A – 192.168.1.0 Remote Network E – 192.168.2.0



default routes (via eth devices) ⟶

network routes (via tunl devices) ⋯⟶

Outcomes: interfaces and routes

vpnd

```
tunl0     Link encap:IPIP Tunnel  HWaddr
          inet addr:192.168.2.1  Mask:255.255.255.0

Kernel IP routing table
Destination     Gateway         Genmask         Iface
192.168.1.0     100.1.1.1       255.255.255.0   tunl0
```

vpnb

```
tunl0     Link encap:IPIP Tunnel  HWaddr
          inet addr:192.168.1.1  Mask:255.255.255.0

Kernel IP routing table
Destination     Gateway         Genmask         Iface
192.168.2.0     200.2.2.2       255.255.255.0   tunl0
```


Outcomes: new B-D link (1)

vpnc

```
[root@vpnc root]# tcpdump -n -i eth0 icmp
eth0: Setting promiscuous mode.
tcpdump: listening on eth0
18:46:29.290495 100.1.1.1 > 200.2.2.2: icmp: echo request
(DF)
18:46:29.290773 200.2.2.2 > 100.1.1.1: icmp: echo reply

2 packets received by filter
0 packets dropped by kernel
```

vpnb

```
[root@vpnb root]# ping -c1 200.2.2.2
PING 200.2.2.2 (200.2.2.2) from 100.1.1.1 : 56 bytes
64 bytes from 200.2.2.2: icmp_seq=0 ttl=254 time=451 usec
--- 200.2.2.2 ping statistics ---
1 packets transmitted, 1 packets received
[root@vpnb root]# ping -c1 192.168.2.1
PING 192.168.2.1 (192.168.2.1) from 192.168.1.1 : 56 bytes
64 bytes from 192.168.2.1: icmp_seq=0 ttl=255 time=488 usec
--- 192.168.2.1 ping statistics ---
1 packets transmitted, 1 packets received
```

Outcomes: new B-D link (2)

vpnd

```
[root@vpnd root]# tcpdump -n -i eth1
tcpdump: listening on eth1
19:46:49.828659 100.1.1.1 > 200.2.2.2: icmp: echo request (DF)
19:46:49.828743 200.2.2.2 > 100.1.1.1: icmp: echo reply
19:46:52.935084 100.1.1.1 > 200.2.2.2: 192.168.1.1 > 192.168.2.1: icmp: echo
request (DF) (ipip)
19:46:52.935214 200.2.2.2 > 100.1.1.1: 192.168.2.1 > 192.168.1.1: icmp: echo
reply (ipip)

4 packets received by filter
0 packets dropped by kernel
```

vpnb

```
[root@vpnb root]# ping -c1 200.2.2.2
PING 200.2.2.2 (200.2.2.2) from 100.1.1.1 : 56 bytes
64 bytes from 200.2.2.2: icmp_seq=0 ttl=254 time=451 usec
--- 200.2.2.2 ping statistics ---
1 packets transmitted, 1 packets received
[root@vpnb root]# ping -c1 192.168.2.1
PING 192.168.2.1 (192.168.2.1) from 192.168.1.1 : 56 bytes
64 bytes from 192.168.2.1: icmp_seq=0 ttl=255 time=488 usec
--- 192.168.2.1 ping statistics ---
1 packets transmitted, 1 packets received
```


Outcomes: new B-D link (3)

vpnd

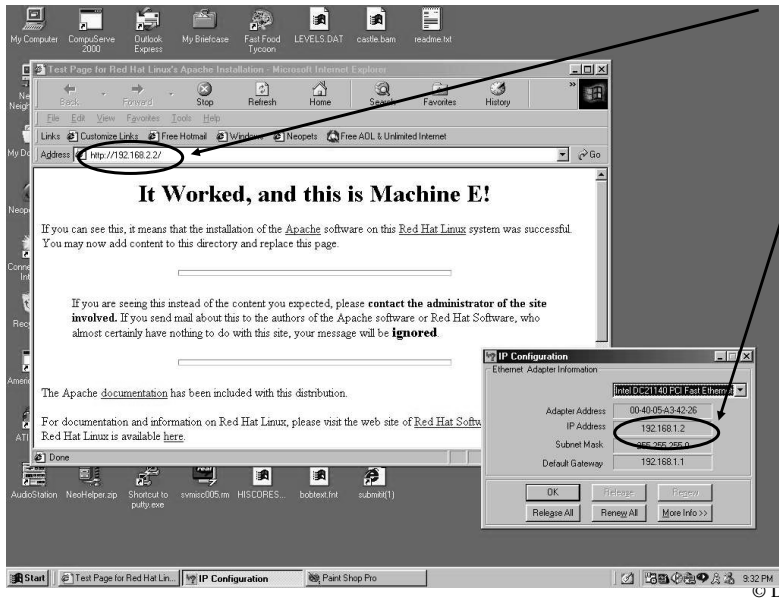
```
[root@vpnd root]# tcpdump -n -i tunl0
tcpdump: listening on tunl0
19:46:52.935084 192.168.1.1 > 192.168.2.1: icmp: echo request (DF)
19:46:52.935174 192.168.2.1 > 192.168.1.1: icmp: echo reply

2 packets received by filter
0 packets dropped by kernel
```

vpnb

```
[root@vpnb root]# ping -c1 200.2.2.2
PING 200.2.2.2 (200.2.2.2) from 100.1.1.1 : 56 bytes
64 bytes from 200.2.2.2: icmp_seq=0 ttl=254 time=451 usec
--- 200.2.2.2 ping statistics ---
1 packets transmitted, 1 packets received
[root@vpnb root]# ping -c1 192.168.2.1
PING 192.168.2.1 (192.168.2.1) from 192.168.1.1 : 56 bytes
64 bytes from 192.168.2.1: icmp_seq=0 ttl=255 time=488 usec
--- 192.168.2.1 ping statistics ---
1 packets transmitted, 1 packets received
```

Outcomes: new A-E link

Browse
To E From A



Outcomes – trace on C

of "capitalizer" between A and E

. . . is unencrypted

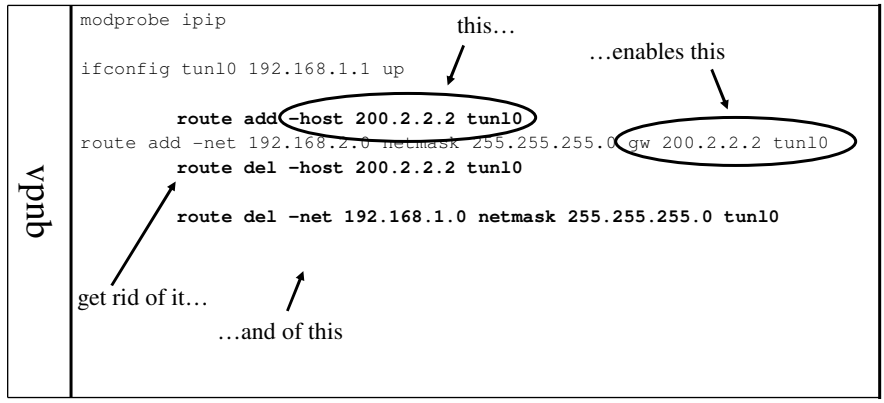
vpnc

```
19:59:36.353508 100.1.1.1 > 200.2.2.2:
        192.168.1.2.1032 > 192.168.2.2.8000: (ipip)
        E..L..@.?...d...
        ....E..8.%@.?..F
        ..........@.r.U
        ...q..}x.......
        .....1.ucla

19:59:36.354162 200.2.2.2 > 100.1.1.1:
        192.168.2.2.8000 > 192.168.1.2.1032: (ipip)
        E..L..@.>.......
        d...E..8'}@.?...
        .........@.....q
        .r.Y..}x.......
        .1...UCLA
```

sent from client (in the clear)

returned from server

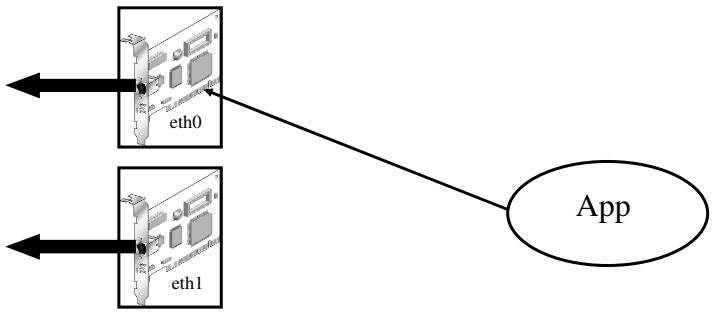
A stupid hack (but necessary)



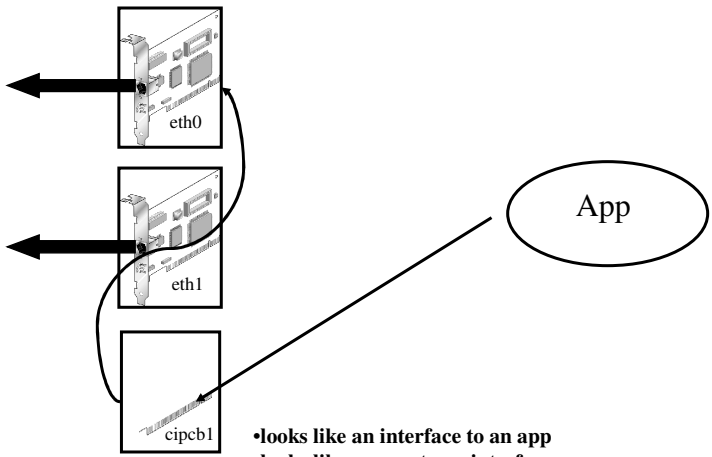
Interfaces

Physical (hardware)	Virtual (software)
●eth0 ●eth1	●tunl0 (ip-ip) ●ipsec0 (IPSec) ●ppp0 (ppp-ssh) ●cipcb1 (CIPE)

Using hardware interfaces



Using software interfaces



- **looks like an interface to an app**
- **looks like an app to an interface**
- **gets to massage traffic passing through**