IP has only a few basic principles

- Gateway / subnet architecture
  - Implemented as encapsulation of IP header within subnet protocol header
- Fragmentation to conform to subnet MTU size
- Best effort at IP layer
  - upper layers responsible for additional “services”
  - nothing expected from lower layers
- E2E IP address distinct from subnet addresses
  - Hourglass: one IP, many different subnets and transports

Gateway / subnet layering, more than anything else, led to IP’s success

(in my humble opinion)

- This layering (encapsulation) allowed the Internet to easily absorb Ethernet
  - X.25 couldn’t do this as easily, for instance

Main benefits of encapsulation

- Modularity
  - Develop subnet technologies without thinking about IP
- Scalability
  - Subnet is not impacted by the tremendous scale of IP
- These are important benefits, and as it so happens:
  - They apply to “mutual encapsulation” as well as to IP-on-subnet encapsulation!
What is “mutual encapsulation”?  
- The situation where “peer network protocols” may each encapsulate over the other  
  - First encountered with PUP and IP around 1980
    - Bob Metcalfe originated the term
  - Sometimes each might view the other as a “subnet”
- The more general term “tunnel” evolved to mean an instance of this type of encapsulation  
  - Subnet encap is of course also a “tunnel” of sorts
- By the early 90’s, it was clear that IP-in-IP was a useful form of tunnel

Why IP-in-IP tunneling???
- Originally (late ’80s) for routing tricks  
  - From RFC 1241 (1991):
    - A tunnel . . . circumvents conventional routing mechanisms
    - . . . bypass routing failures, avoid broken gateways and routing domains, or establish deterministic paths for experimentation
  - To do policy routing over administrative domains (RFC1479)
Was Postel stupid?

- Didn’t he foresee a need to tunnel IP around routing failures etc.?? (RFC791, 1981)
- Of course he did: Loose Source Routing (LSR)
  - IP LSR option carries a series of router addresses
  - Each router is visited in turn
  - By swapping router address into the destination address field
- But LSR was never widely implemented
- And we figured out how to solve routing without tunnels
  - Dynamic routing protocols (OSPF, ISIS, RIP, . . .)
  - BGP and next hop resolution

Even so, IP-IP tunnels have proliferated!

- L2TP
  - R-R, prot 115
  - XX-L2TP-UDP-IP
- PPTP
  - R-R, later H-R
  - XX-PPP-GRE-IP
- MIP
  - H-R, prot 55 (135 for v6)
  - IP-IP, or IP-GRE-IP
- GRE
  - R-R, H-R (PPTP), prot 47
  - XX-GRE-IP
- IP-IP
  - R-R, H-R (MIP), prot 4
- IPsec
  - R-R or H-R or H-H, prot 50,51
  - IP-IPsec-IP, or
  - IP-IPsec-UDP-IP
- IPv6-IP(v4)
  - R-R, H-R, or H-H, prot 41
  - IPv6-IP, or IPv6-UDP-IP
- IP multicast-IP (mbone)
  - Uses IP-IP
- link-IP!
  - Eth-IP, prot 97
  - MPLS-IP, prot TBA
* Yes, this is meant to be confusing**
** Assume errors here…

Why so many tunnels???

- Four primary reasons:
  - Virtualization
  - Security
  - Preserve an interface
  - Protocol evolution (incremental deployment)
- (Note that solving routing problems per se is not one of the reasons!)

Some tunnel terminology . . .

(This is my terminology)

- Symmetric versus Cone
  - Symmetric: Tunnel Endpoint (TE) and Tunnel Startpoint (TS) bound together and explicitly configured
    - Tunnel may or may not be authenticated
    - Packets may or may not be authenticated
  - Cone: TE and TS not explicitly bound--any TS can send to any TE (this is rare)
- Unidirectional versus Bidirectional
  - Cone is by definition unidirectional
  - Symmetric is typically bidirectional
Other tunnel characterizations

- How is the tunnel endpoint (TE) discovered?
- How is the tunnel established?
- What types of systems (host, router, etc.) can be tunnel endpoints?
- Are the tunnel endpoints authenticated, and how?
- Are packets in the tunnel authenticated, and how?
- How are fragmentation and TTL handled?

GRE (Generic Routing Encapsulation)

- The only tunnel standardized outside of a specific context
- Meant to satisfy several “generic” tunnel requirements:
  - Allows anything Some tunnels should mimic link characteristics in terms of packet ordering and loss
  - Some tunnels have a certain virtual context (i.e. VPN)

GRE Header (RFC 1701)

<table>
<thead>
<tr>
<th>C</th>
<th>R</th>
<th>K</th>
<th>S</th>
<th>s</th>
<th>Recur</th>
<th>Flags</th>
<th>Ver</th>
<th>Protocol Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For Routing field processing
- Same as Ethernet

More about GRE tunnels

- GRE spec says nothing about how the tunnel is configured
  - Which is appropriate
- GRE provides no authentication
  - Of the tunnel or of the packets in the tunnel
  - The tunnel can run over IPsec though, thus allowing multiprotocol and mcast over IPsec
- GRE tunnel does not have to be symmetric
  - But typically it is (i.e. for VPN)
GRE VPN usage from cisco

GRE (and other) tunnel issues

• All router-to-router tunnels must deal with two basic issues:
  – Fragmentation
  – IP TTL (hop count) field
    (Actually this not a big deal—just copy TTL over for both encap and decap)
• Problem with fragmentation is that packet may fragment in tunnel, but ICMP error message doesn’t identify sending host
  – Router may need to know tunnel MTU, and generate its own ICMP message to host
  – I don’t know if this is still a real issues or not . . .

L2TP and PPTP

• Purpose is to extend a PPP link across the Internet
  – Mainly for client VPN functionality
  – They are essentially “competing” protocols
• PPP is a link-layer protocol originally designed for authentication and framing for dial-up links
  – Between a host and network access controller box
  – Now also used for high-speed router links

Draw a PPTP/L2TP example (with Radius tunnel parameter)
L2TP and PPTP

- L2TP and PPTP tunnels are always bidirectional (and symmetric)
- The tunnel itself may be authenticated
- The tunnel may be dynamically configured via a RADIUS attribute
- User sessions running over the tunnel are also authenticated (using PPP authentication methods)
- These days PPTP often runs directly from the client host
  - As a client VPN solution

Mobile IP (MIP)

- Allows a host to maintain the same IP address as it changes access points
- Operates by establishing a tunnel from the mobile host to a fixed router (the Home Agent)
  - This tunnel is IP-IP or GRE
- Mutual authentication of Home Agent and mobile host
  - Originally used a MIP-specific authentication, later evolved to use same authentication as PPP
    - CHAP with Network Access Identifiers (NAI)

IPv6 – IPv4

- IPv6 – IPv4 needed to transition to IPv6
  - Run IPv6 over existing IPv4 infrastructure
  - Can be GRE, but often not
- IPv6 folks have been quite creative about how to auto-configure these tunnels
  - 6to4: embed IPv4 address in IPv6 address to cross global IPv4 backbone
  - ISATAP: embed IPv4 address in IPv6 address to cross enterprise network
  - Teredo: embed NAT address in port in IPv6 address to cross NAT (IPv6-UDP-IPv4)
  - Plus protocols for negotiating and establishing v6-v4 tunnels

MIP and VPN

- Some commercial products combine benefits of VPN and MIP (mobile host access to VPN)
  - Runs IPsec over MIP (over UDP, in order to deal with NAT boxes!)
- MIP tunnels have evolved to have much in common with L2TP/PPTP tunnels
  - Bidirectional, authenticated
  - RADIUS can now be used to assign the tunnel endpoint (HA)
  - Indeed some folks derive mobility from L2TP by maintaining abstraction of a stable PPP session during mobility

IPv6 – IPv4

- IPv6 – IPv4 needed to transition to IPv6
  - Run IPv6 over existing IPv4 infrastructure
  - Can be GRE, but often not
- IPv6 folks have been quite creative about how to auto-configure these tunnels
  - 6to4: embed IPv4 address in IPv6 address to cross global IPv4 backbone
  - ISATAP: embed IPv4 address in IPv6 address to cross enterprise network
  - Teredo: embed NAT address in port in IPv6 address to cross NAT (IPv6-UDP-IPv4)
  - Plus protocols for negotiating and establishing v6-v4 tunnels
IPv6 tunnel broker

1. AAA Authorization
2. Configuration request
3. TB chooses:
   - TS
   - IPv6 addresses
   - Tunnel lifetime
4. TB registers tunnel IPv6 addresses
5. Config info sent to TS
6. Config info sent to client:
   - Tunnel parameters
   - DNS name
7. Tunnel enabled

(Figure stolen from Juniper slides)

IPv6 Tunnel
IPv4 Network
IPv4 Network

IPv6 tunnel broker

6to4

- Designed for site-to-site and site to existing IPv6 network connectivity
- Site border router must have at least one globally-unique IPv4 address
- Uses IPv4 embedded address

**Example:**

<table>
<thead>
<tr>
<th>Reserved 6to4 TLA-ID:</th>
<th>2002::/16</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 address:</td>
<td>138.14.85.210 = 8a0e:55d2</td>
</tr>
<tr>
<td>Resulting 6to4 prefix:</td>
<td>2002:8a0e:55d2::/48</td>
</tr>
</tbody>
</table>

- Router advertises 6to4 prefix to hosts via RAs
- Embedded IPv4 address allows discovery of tunnel endpoints

6to4 is not bidirectional

- Mostly so far we’ve seen bidirectional (symmetric) tunnels
- 6to4 is the first cone tunnel we’ve seen
  - Because any 6to4 router may send packets to any other 6to4 router
### mbone
- The mbone is perhaps the earliest example of an IP-IP overlay network
  - Used to run IP multicast over an IP unicast infrastructure
- Used IP-IP encapsulation
- Note:
  - Most global multicast done as application overlays (i.e. Akamai, Real Networks)
  - Native IP multicast usage growing in enterprises

### Link over IP
- Ethernet over IP
  - Used to preserve an Ethernet interface abstraction
- MPLS over IP
  - Naturally

### MPLS “tunnels”
- MPLS is a “subnet” (below IP) technology
- But it is often seen as an IP tunneling technology because it is closely coupled with IP
  - BGP carries information about MPLS tunnel endpoints for running provider VPNs
  - MPLS labels can be “stacked”, so it is a powerful primitive for tunneling
    - Convey tunnel context, for instance

### Do we have enough tunnels???
- Well, yes and no . . .
- We have enough tunnel formats (more than enough!), but we are still nowhere near getting all we can from tunneling!
  - My opinion anyway
- What’s missing?
  - General purpose lightweight cone tunnels at routers
  - Ability to establish per-socket tunnels at hosts
    - Not just per-interface as we have today
Per-socket host tunnels

- Needed because of “middleboxes”
  - Firewalls, NATs, web proxies, virus filters, protocol boosters, etc.
- Today hosts can establish “per-interface” tunnels (i.e. to VPN server), but not per-socket
- Per-socket tunnel definition allows packets to be routed through middleboxes as appropriate
- A signaling protocol like SIP could be used to specify the middleboxes

NAT Example

Need for lightweight router cone tunnels

- Traffic engineering within an ISP
  - This courtesy Jennifer Rexford
- Traffic engineering across ISPs
- Better BGP scaling
  - These last two from Joy Zhang’s TBGP research
  - TBGP = Tunneled BGP!

TBGP

- Problem:
  - BGP overloaded: slow response times, hard to understand and debug
  - BGP does not provide adequate traffic engineering (especially site multihoming)
- TBGP solution:
  - Pull as much out of BGP as possible, making it more responsive and simpler to understand
    - Use BGP only to route to POPs, not all destinations
    - Use tunnels and flat tunnel mapping tables to select appropriate POP
  - Intuition: Flat mapping tables much easier to deal with than BGP distributed route computation
Intra-ISP traffic engineering

- Problem:
  - Traffic engineering through OSPF metric manipulation is very hard
    - One metric change ripples through the system in hard to predict ways
  - MPLS is too heavyweight (label setup protocols etc.)
- Solution:
  - Use IP-IP tunneling from ingress POP to egress POP for simple, fine-grained traffic engineering
  - Perhaps managed from a replicated central controller

Can’t we do intra-ISP tunneling today???

- Why not configure N\(^2\) symmetric tunnels?
  - After all, N is probably only a few hundred
- Two problems:
  - Today routers can establish only a limited number of tunnels
  - Detunneling is slow (double the packet processing time)
- These problems exist, in essence, because routers treat tunnels as symmetric
- What we need is fast detunneling!

Example “services” router packet handling
Packet handling for detunneling (two loops through the process)

Packet In
Apply Ingress ACL
Apply Pre-forwarding Services
Forwarding Table Lookup
Apply Post-forwarding Services
Apply Egress ACL
Packet Out

Here is where router "discovers" that packet is for "self" and detunnels (e.g. forwarding table has an entry for "self")

Inner header is then processed in the "normal way"

Faster detunneling

- Note that detunneling is nothing more than glorified decapsulation
- Routers can decapsulate the link layer fast, so why not the network layer?
  - Because link layers are local...we trust the encapsulator and understand its limited context
  - It is architecturally convenient to discover packet is for "self" in the forwarding table
- Technically, a router could detunnel the link layer fast
  - simple pattern match on a few header fields, move a pointer
  - But is it safe to do so???

Possible tunnel dangers?

- Subvert ACLs?
  - I distrust packets from A, and trust packets from B
  - Source at A tunnels packet via B!
  - (Not clear that this is a serious problem)
- Hide source of DDoS attack?
  - Attack appears to come from tunnel endpoint
- Others?

Trusted intra-ISP lightweight tunneling

- Seems straightforward to trust an intra-ISP tunnel
  - ISP doesn't advertise tunnel endpoint prefixes outside of ISP
  - ISP puts explicit blackhole routes for tunnel endpoints at tunnel startpoints (ISP edge routers)
Trusted intra-ISP lightweight tunneling

- Packet enters ISP at this edge router
- Blackhole routes to TEs here
- Add tunnel headers here
- Core Routers
- Core Routers
- Core Routers
- Strip tunnel headers here
- Blackhole routes to TEs here
- Packet leaves ISP at this edge router

Trusted inter-ISP lightweight tunnels?

- This is more difficult
- Perhaps a similar model (among participating ISPs) would be adequate?
- Other ideas?