

NSWI184 – Řízení počítačových sítí

Přednáška devátá

Ondřej Zajíček, Kateřina Kubecová

2025-12-03

IP Tunneling

- ▶ Encapsulating IP packets within another network or upper layer protocol
- ▶ Creates virtual point-to-point L3 links over internet
- ▶ Site-to-site connectivity
- ▶ Virtual private networks
- ▶ Network virtualization / overlay networks

How IP Tunneling Works

Original Packet: [IP Header | Payload]

Tunneled Packet: [Outer IP Header | Tunnel Header | Inner IP Header | Payload]

- ▶ Route original packet through tunnel iface
- ▶ Wrap original packet in new headers
- ▶ Send through intermediate network
- ▶ Remove outer headers at destination
- ▶ Route original packet again

IP-in-IP Tunneling

- ▶ No tunnel header, just outer and inner IP header
- ▶ IP protocol number to 'Protocol' / 'Next Header' field
- ▶ IPv4-in-IPv4: RFC 2003 (Protocol 4)
- ▶ IPv6-in-IPv6: RFC 2473 (Protocol 41)
- ▶ 6in4 / 4in6 as IPv6 transition mechanisms

IP-in-IP Tunneling

Pros:

- ▶ The simplest protocol
- ▶ Stateless

Cons:

- ▶ NAT traversal issues
- ▶ No way to distinguish parallel tunnels
- ▶ No authentication / encryption
- ▶ Limited to IP payload

GRE (Generic Routing Encapsulation)

- ▶ RFC 2784
- ▶ Simple GRE header
- ▶ Versioned and somewhat extensible
- ▶ Optional checksum, seqnum, 32-bit key
- ▶ Ethertypes as protocol types

GRE (Generic Routing Encapsulation)

```
+-----+
|C| |K|S| Reserved0          | Ver |           Protocol Type           |
+-----+
|      Checksum (optional)    |      Reserved1 (Optional)      |
+-----+
|                               Key (optional)                              |
+-----+
|               Sequence Number (Optional)                                |
+-----+
```

GRE (Generic Routing Encapsulation)

Pros:

- ▶ Stateless
- ▶ Parallel tunnels distinguishable
- ▶ Any network protocol as payload

Cons:

- ▶ NAT traversal issues
- ▶ No authentication / encryption

MTU (Maximum Transmission Unit) Problems

- ▶ Outer IP and tunnel headers add overhead (20-50+ bytes)
- ▶ Original packet + overhead may exceed link MTU
- ▶ ICMP 'Fragmentation Needed' messages sometimes blocked by firewalls
- ▶ Results in 'black hole' scenarios where packets are silently dropped
 - ▶ Large packets fail while small ones succeed
 - ▶ Applications connect but hang during data transfer

MTU Problem Mitigations

- ▶ Set tunnel MTU to such value that encapsulated packets fit link MTU
($\text{Tunnel MTU} = \text{link MTU} - \text{tunnel overhead}$)
- ▶ Fragmentation: Allow tunnel endpoint to fragment encapsulated packets
($\text{Tunnel MTU} = \text{link MTU}$, performance impact)
- ▶ MSS Clamping: Reduce TCP Maximum Segment Size at tunnel endpoints

Hop Limit Processing

- ▶ Inherit outer TTL/HL from inner packet
- ▶ Use fixed value and treat tunnel as single hop
- ▶ Effects on traceroute

NAT / Firewall traversal

- ▶ RFC 8086: GRE-in-UDP encapsulation
- ▶ Foo-Over-UDP, Generic UDP encapsulation
- ▶ No keepalives
- ▶ No dynamic endpoints

Multipoint tunnels

- ▶ No multicast → no neighbor discovery
- ▶ But it could behave as PtMP interface
- ▶ Tunnel with no explicit remote IP
- ▶ How to resolve route next hops to remote IPs?
- ▶ IPIP hack: route next hop *is* remote IP
- ▶ Manually fill neighbor table with remote IP as link address
- ▶ Lightweight tunnels – remote IP controlled by route attributes
- ▶ Incoming traffic is accepted and decapsulated from any IP (!)

IPSec

- ▶ IP-in-IP / GRE tunnels have no authentication / encryption
- ▶ Can be used together with IPSec to provide security / VPN
- ▶ IPSec itself is not IP tunneling mechanism (even in 'tunnel' mode)

WireGuard

- ▶ Modern non-standard VPN protocol
- ▶ Strong cryptography: ChaCha20, Poly1305, Curve25519
- ▶ Endpoints have public and private keys
- ▶ Integrated keepalives
- ▶ Handles dynamic endpoints
- ▶ UDP for NAT traversal

WireGuard

- ▶ Each wireguard interface has a list of peer
- ▶ Each peer consist of public key, remote IP, and a list of allowed IP ranges
- ▶ Allowed IP ranges used both for egress routing and ingress filtering
- ▶ Routes should direct just to the interface, with no next hop
- ▶ Incompatible with regular dynamic routing protocols
- ▶ Just use multiple PtP interfaces

Virtual Local Area Networks (VLANs)

- ▶ Link layer / L2 / Ethernet technology
- ▶ Separate L2 networks within the same physical network infrastructure
- ▶ Devices in different VLANs are isolated from each other
- ▶ IEEE 802.1Q standard

VLAN Tagging

- ▶ VLAN ID: 12-bit field (4094 possible VLANs)
- ▶ VLAN tag: 4-byte header in Ethernet frame
- ▶ Tagged frames carry VLAN info, untagged do not

Switch Port Configuration

- ▶ Each port has a list of VLANs it is member of
- ▶ Each VLAN on port set as tagged or untagged → used for egress frames
- ▶ One VLAN on port can be set as PVID → used for ingress untagged frames
- ▶ Access Ports: Belong to single VLAN set as untagged and PVID
- ▶ Trunk Ports: Carry multiple VLANs, set as tagged

Layer 2 Tunneling

- ▶ Transporting Ethernet frames over IP networks
- ▶ Creates virtual (point-to-point) L2 links over internet
- ▶ Allows to bridge LANs over internet
- ▶ Virtual LANs over IP infrastructure

NVGRE (Network Virtualization using GRE)

- ▶ RFC 7637
- ▶ Ethernet in GRE, Protocol 0x6558
- ▶ VSID (Virtual Subnet ID): 24-bit identifier, stored in GRE key
- ▶ Frame Format:

[IP Header | GRE Header | Ethernet Frame]

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0| |1|0|   Reserved0       | Ver |   Protocol Type 0x6558           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Virtual Subnet ID (VSID)              |   FlowID   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

VXLAN (Virtual Extensible LAN)

- ▶ RFC 7348
- ▶ Ethernet in UDP-based protocol (port 4789)
- ▶ VNI (VXLAN Network Identifier): 24-bit identifier
- ▶ Frame Format:

[IP Header | UDP Header | VXLAN Header | Ethernet Frame]

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|R|R|R|R|1|R|R|R|                Reserved                |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                VXLAN Network Identifier (VNI) |   Reserved   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Geneve (Generic Network Virtualization)

- ▶ RFC 8926
- ▶ Ethernet in UDP-based protocol (port 6081)
- ▶ VNI (Virtual Network Identifier): 24-bit identifier
- ▶ Flexible header, variable-length options
- ▶ Options delegated to third-parties

Geneve (Generic Network Virtualization)

```
[IP Header | UDP Header | VXLAN Header | Ethernet Frame]
```

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|Ver|  Opt Len  |O|C|      Rsvd.  |          Protocol Type          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Virtual Network Identifier (VNI)          |      Reserved      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
~          Variable-Length Options          ~
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

L2TPv3 (Layer 2 Tunneling Protocol v3)

- ▶ RFC 3931
- ▶ Multiple L2 types (Ethernet, ATM, Frame Relay, ...)
- ▶ Control messages: Session establishment, keepalives
- ▶ Data messages: Encapsulated data link frames
- ▶ Both UDP and direct IP encapsulation
- ▶ Multiple pseudowires per tunnel

L2 tunnels and VLANs

- ▶ Encapsulate tagged frames
- ▶ Translate VLANs to VSIDs / VNIs

Multipoint L2 tunnels

- ▶ Popular case – network virtualization
- ▶ Neighbor discovery
- ▶ BUM traffic delivery
 - ▶ Use IP multicast
 - ▶ Ingress replication
- ▶ Out-of-band control protocols (EVPN)
- ▶ IP neighbor discovery replacement