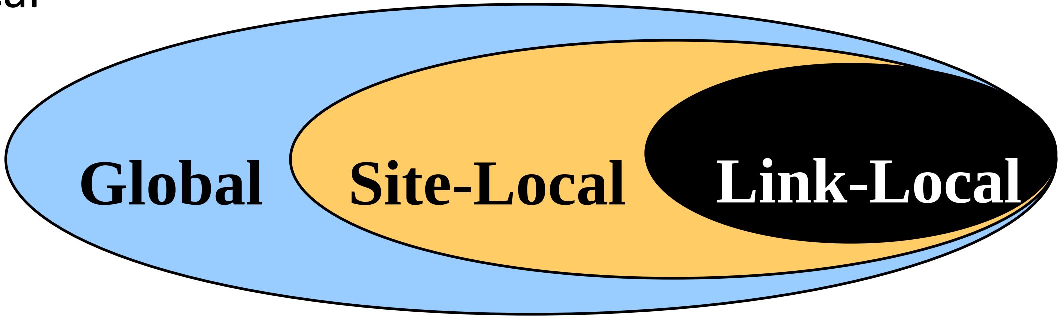
Networking Technologies and Applications

Rolland Vida
BME TMIT

October 20, 2016

IPv6 addressing scheme

- One interface might have several addresses, with different scopes:
  - Link Local
  - Site Local
  - Global



Unicast addesses

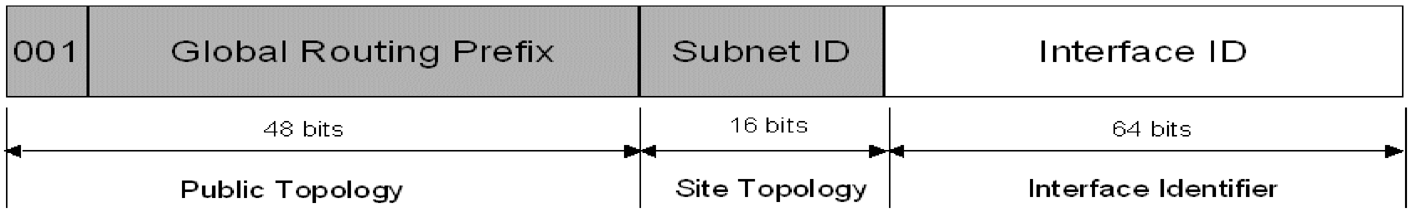
- Valid over a limited scope
  - Scoped address
  - Novelty in IPv6
- Scope = local link
  - For communication among nodes on the same link
  - Unique only on that link, cannot be used for communication outside the link
  - Automatically configured on each interface
  - Each IPv6 node has an initial address with which it can start communicationg
    - Neighbor dicovery, router discovery
  - Format:
    - FE80:0:0:0:<interface identifier>
    - Interface ID – EUI (64) address
      - Extending the previous 48 bit MAC address

Unicast addresses

- Scope = site local
  - Used for communication inside the same site
  - Routers will not forward it outside the site (to the Internet)
  - Similar to IPv4 private addresses
  - Not automatically configured
  - Format:
    - FEC0:0:0:<subnet id>:<interface id>
    - Subnet id = 16 bit = 64K subnet
  - Allows the addessing of an entire organization (company, university)
    - E.g. allocate site-local addesses to the entire network
    - Renumbering when connecting to an IPv6 network
      - We change the first 48 bits (to a site ID)
    - Renumbering when connecting to a new service provider

Unicast addresses

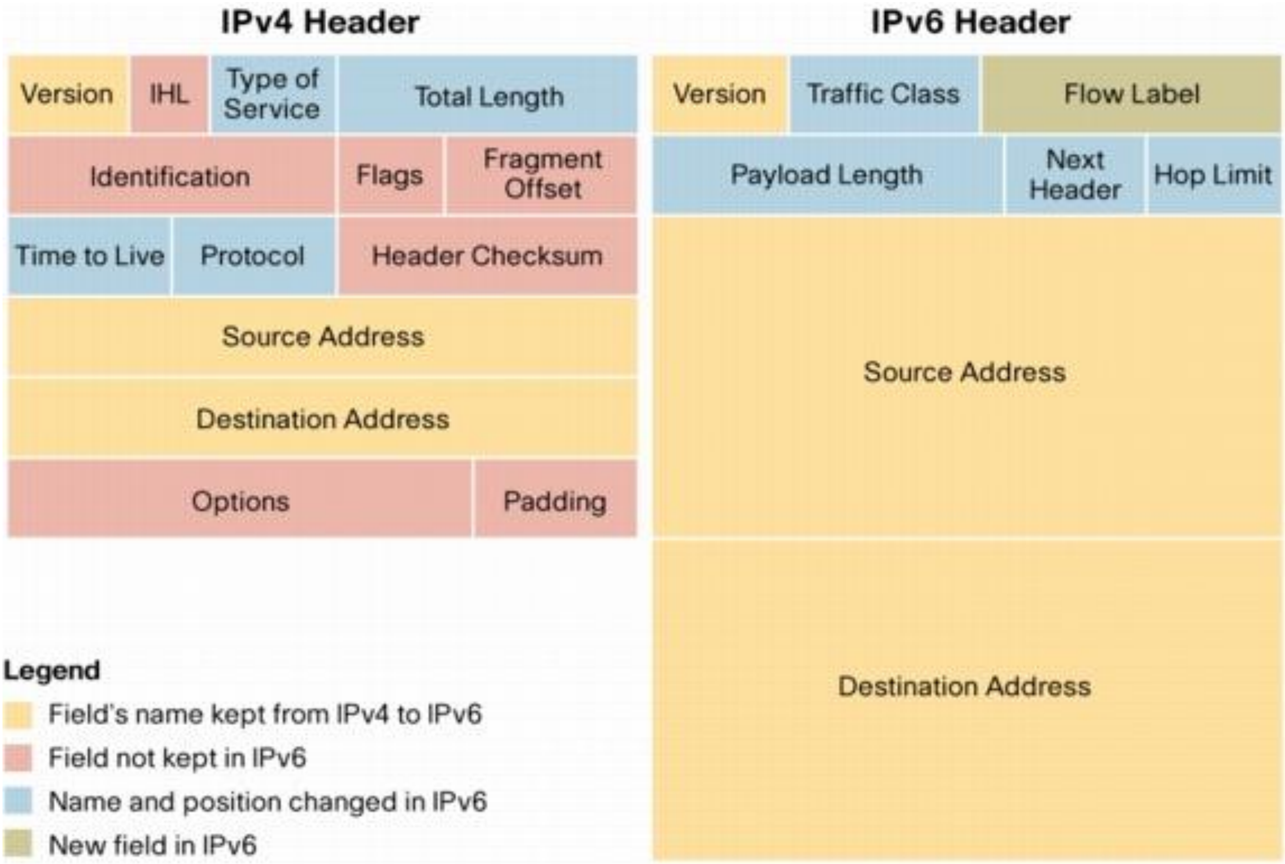
- Global Unicast Address
  - Used for global communication
  - Hierarchical global prefix
    - Structured by RIRs and ISPs
  - Subnet ID
    - Hierarchically structured by the network administrator
  - Interface ID



Multicast addresses

- Instead of broadcast addresses
- Scoped addresses
  - Node, link, site, organisation, global
- Format
  - FF<flags><scope>::<multicast group>
- Flag:
  - 0 – permanent
  - 1 – dynamic
- Scope:
  - 1 – node
  - 2 – link
  - 5 – site
  - 8 – organisation
  - E – global
- E.g.
  - FF02::1 – all nodes on the local network
  - FF02::2 – all routers on the local network

Format of the mandatory fixed IPv6 header



IPv6 mandatory fixed header

- Version (4 bits)
  - IP version
- Class – priority class (8 bits)
  - Defines the priority of the packet
    - ToS (Type of Service) field in IPv4
  - Used for providing Quality of Service (discussed later)

IPv6 mandatory fixed header

- Flow Label
  - Associated to traffic with special QoS requirements
    - 20 bit long
  - Can be used in router caches to decrease the processing time
    - A packet first arrives to the router
    - The fow label is stored in the cache
    - When the next packet with the same flow label arrives…
      - No need to look up the destination address in the routing table
      - Packet immediately routed based on the flow label
- Payload Length (16 bits)
  - Length of the useful data, in bytes

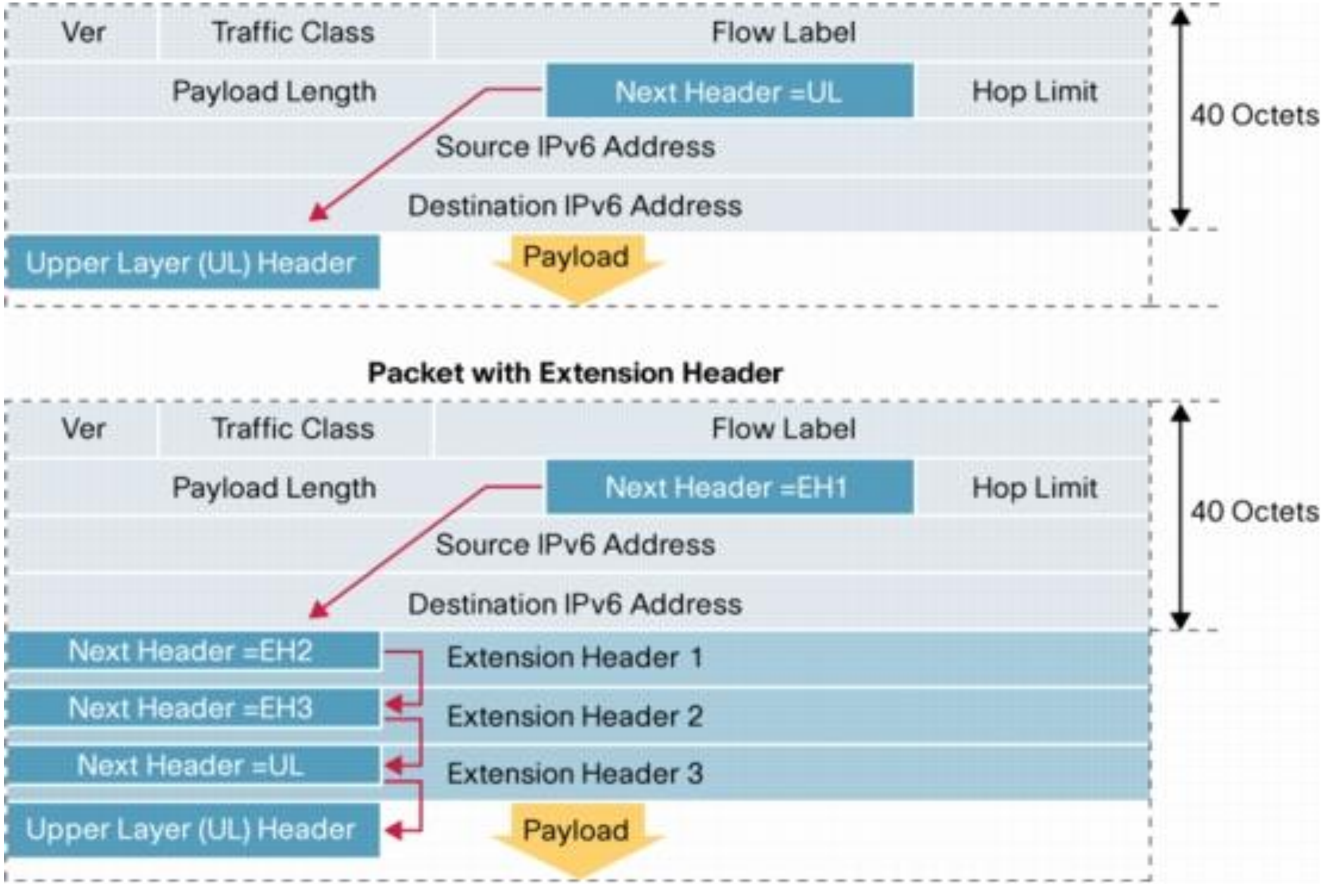
IPv6 mandatory fixed header

- Next-Header (8 bits)
  - Identifies the header directly following the fixed header
    - It might be an extension header, or a higher layer protocol header (e.g., TCP)
- Hop Limit (8 bits)
  - Provides how far a packet can travel
  - Same as the Time To Live (TTL) field in IPv4
- Source Address (128 bits)
  - Address of the original source of the packet
- Destination Address (128 bits)
  - Not necessarily the address of the last receiver, if the packet also contains a Routing Header

IPv6 extension headeres

- IPv6 packet – starts with a 40 byte long mandatory fixed header
- Extra information related to the intermediate network encoded in Extension Headers
- Most of the extension headers are not processed by the intermediate routers, only by the final destination
- Each extension header has a special value for the next header field
  - Many extension headers can be used in parallel
  - The value of the last next header field identifies the upper layer protocol
  - The header can be of any length

IPv6 chained extension headers



IPv6 extension headers

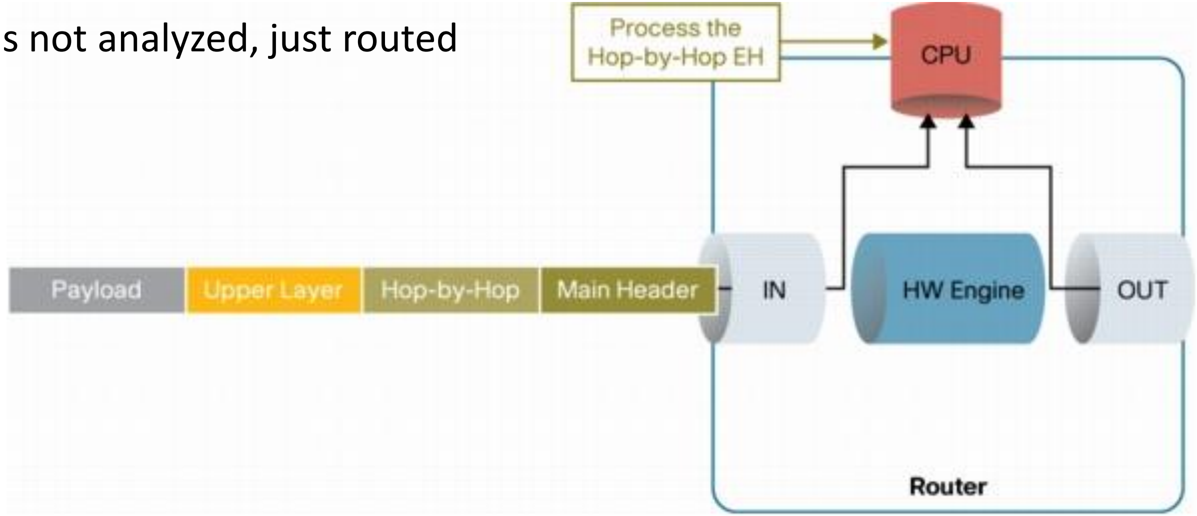
- Header order
  - Important: extension headers should respect the suggested order
    - Easier for routers to process the packet
    - In most cases the routers process only the hop-by-hop options and the routing header
  - Exception: destination option
    - Right before the higher layer header
    - If we want the intermediate routers to process the destination option header, we should put it right before the routing header, and they should be processed together.
    - A packet might contain a destination option headers in both locations

IPv6 extension headers

- The suggested header order:
  - IPv6 Header
  - Hop-by-hop Options Header (type = 0)
  - Destination Options Header (1)
  - Routing Header (type = 43)
  - Fragment Header (type = 44)
  - Authentication Header (type = 51)
  - Encapsulating Security Payload (ESP) (type = 50)
  - Destination Options Header (2) (type = 60)
  - Upper Layer Header (pl. TCP vagy UDP)

IPv6 extension headers

- Hop-by-hop Options Header
  - Contains IP options for the intermediate routers
  - Each intermediate router should analyze and process the Hop-by-hop Header
  - Router Alert option alerts tranzit routers
    - If the packet packet contains information that should be processed by an intermediate router
    - Otherwise the packet is not analyzed, just routed



IPv6 extension headers

- Routing Header
  - In the normal case the source of the IP packet leaves the routing task to the network
  - In case of source routing, the source will specifiy a path with router addresses
    - The entire list in the Routing Header (e.g., A, B, C, D)
    - The destnation address is always the address of the next router specified in this field, except the last router
    - Each router modifies the destination address, before forwarding the packet

IPv6 extension headers

- Fragment Header
  - In IPv4 fragmentation and reassembly automatically, if explicitely not forbidden
    - **Don't Fragment flag**
  - In IPv6 packets are not fragmented by default
    - If the packet is too large for the transmission medium, it is dropped and an ICMP (Internet Control Message Protocol) error message is sent
    - The source discovers the path MTU-t
      - Maximum Transmission Unit
    - Tries to send packets with a lower size than the MTU
    - If we need fragmentation, that can be done using the Fragmentation Extension Header
- Authentication Header
  - Guarantees that …
    - The recieved packet is authentic
    - It was not altered on its path
    - Comes from the specified source
- Destination Option Header
  - Contains options to be processed by the destination node

Transition to IPv6

- Routing services built on IP
  - RIPv6(ng), OSPFv6 (v3), BGPv6
- Network and transport layer protocols built on IP
  - TCPv6, UDPv6, RSVPv6
- Applications
  - Each application that was using directly the IPv4 addresses is not independent from the lower layers, so IPv6 support should be implemented in it
- Gradual transition
  - No „D-day”
- Expectations regarding transition
  - No transition dependencies
    - The transition of a given node can be done independently from the others
    - The most important aspect is backward compatibility
  - It should be as easy as possible for the end user
  - The different transition solutions should be appliable independently of each other
    - At least at the level of the different domains

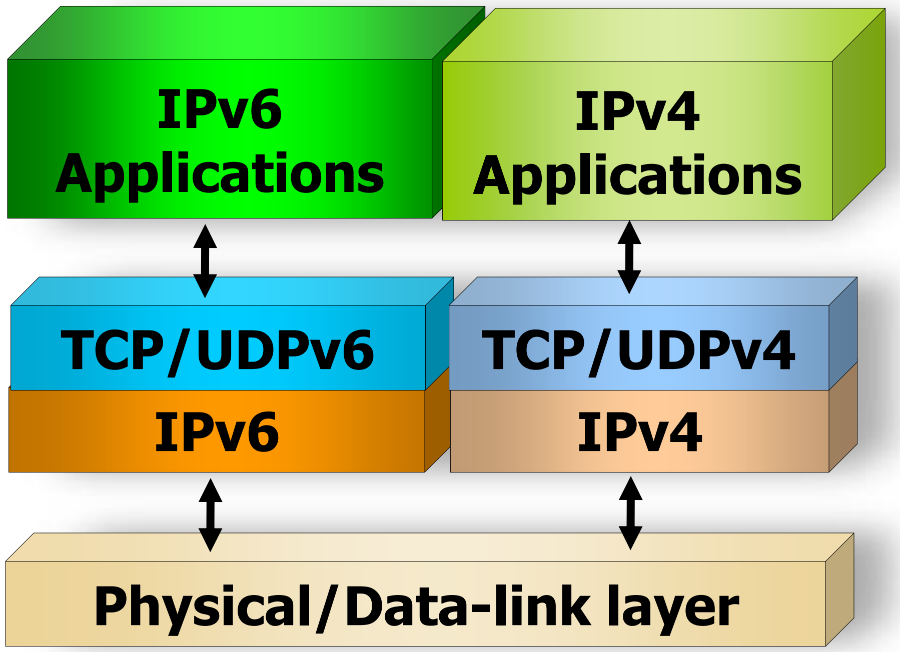
Transition solutions

- Dual Stack
  - Both IPv4 and IPv6 stack on the same device
- Tunnels
  - Initially tunneling IPv6 packets in IPv4 domains
  - Later, tunneling IPv4 packets in IPv6 domains
- Protocol translation
  - Headers containing protocol information should be translated into different protocol headers, based on certain translation rules
  - IPv6 <-> IPv4

Dual protocol stack

- The first step towards deploying IPv6 is deploying some nodes that support IPv6 (as well, next to IPv4)
  - They have a double stack strategy
    - Use IPv6 to communicate with other IPv6 systems
    - Can switch back to IPv4 mode to talk to IPv4 systems

Dual Stack


IPv6 Applications
IPv4 Applications
TCP/UDPv6
TCP/UDPv4
IPv6
IPv4
Physical/Data-link layer

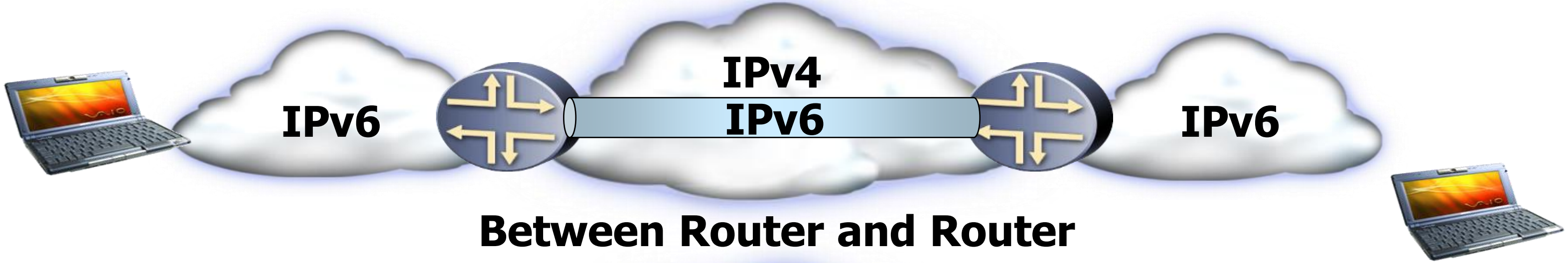
Dual stack

- Advantages
  - Easy to install, configure, maintain
  - The entire functionality of IPv6 can be exploited
  - Any two nodes can communicate exclusively with IPv4 or IPv6 packets
  - Transparent transition for the end users
- Drawbacks
  - Not scalable: each node should have an IPv6 and an IPv4 address, the limitation of the IPv4 address domain obstructs its spreading
  - The size of the routing tables is increased in the routers
  - Not flexible: no communication possibility between nodes speaking just IPv4 or just IPv6

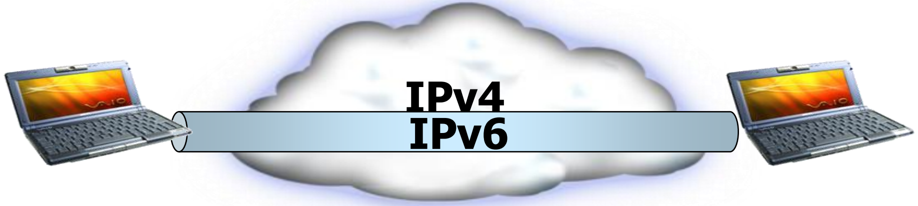
Tunneling

- IPv6 packet encapsulated inside an IPv4 packet
- The tunnel endpoints manage the encapsulation
- The process transparent to the intermediate nodes
- Configured tunnels
  - The tunnel endpoints are explicitely configured
  - They are dual stack nodes
  - Should use gloval IPv4 addresses, no NAT between the endpoints
- Automatic tunnels
  - The tunnel endpoint are automatically discovered by the network
  - Tunnel Brokers (RFC3053)
  - 6to4 (RFC3056)
  - ISATAP (Intra-Site Automatic Tunnel Addressing Protocol)
  - 6over4 (RFC2529)
  - Teredo: support tunnels through IPv4 NAT

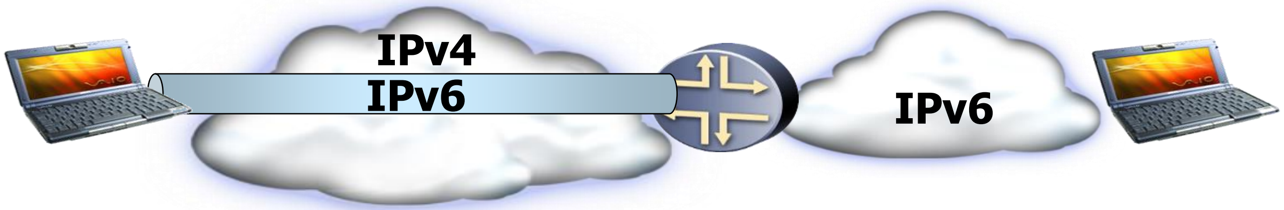
Tunneling



Between Router and Router



Between Host and Host



Between Host and Router / Router and Host

Translators

- Network layer translators
  - SITT (Stateless IP/ICMP Translator Algorithms) (RFC2765)
  - NAT-PT (Network Address Translator-Protocol Translator) (RFC2766)
  - BIS (Bump int the Stack) (RFC2767)
- Transport layer translator
  - TRT (Transport Relay Translator) (RFC3142)
- Application layer translators
  - BIA (Bump in the API) (RFC3338)
  - SOCKS64 (RFC3089)
  - ALG (Application Level Gateway)

Network layer translators

- The IPv4 messages are translated into IPv6 messages, and vice-versa (especially the headers)