Automatic Tunneling Setup for/with IPv6 (ATS6)

Softwires Solution Proposal

Softwires Interim Meeting, HK - 23/02/2006 (v1.5)

miguelangel.diaz@consulintel.es
jordi.palet@consulintel.es

draft-palet-softwires-ats6-01

Requirements

- To setup (and activate) IPvX-IPvY tunnels
- Typically to get either:
  - IPv6 address in an IPv4-only or IPv6-only network
  - IPv4 address in an IPv6-only network
  - IPv6 prefix in an IPv4-only or IPv6-only network
- Low overhead on communications
- Low overload on user device (PC, mobile phone, etc.)
- Lightweight deployment
- NAT/PAT and Firewall traversing
- To authenticate the user, just in case
- Compatibility with existing protocols to obtain the IP address (DHCP, DHCPv6, DHCPv6-PD)
- In general the ones specified on:
  - draft-suryanarayanan-v6ops-zeroconf-reqs-01
  - draft-nielsen-v6ops-3GPP-zeroconf-goals-00
  - draft-ietf-v6ops-assisted-tunneling-requirements-01
  - draft-palet-v6tc-goals-tunneling-00.txt
  - IPv6 address in an IPv4-only network

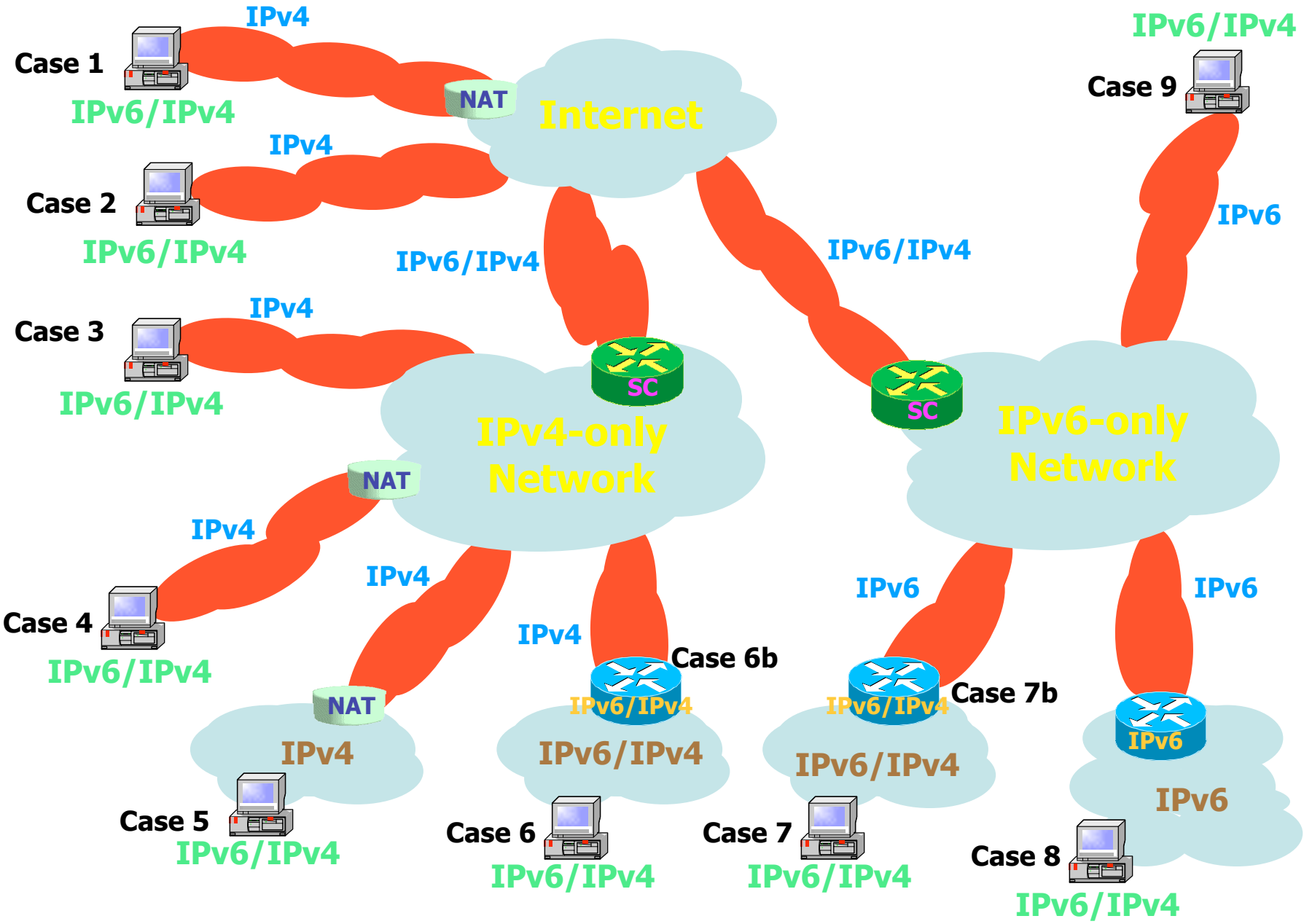
Assumptions

- The SC (Softwires Concentrator) is discovered by other means before starting the tunnel setup
- The SI (Softwires Initiator) is pre-registered within the domain.
  - A registered user is not the same that an authenticated user
- Registered user meaning that user has an IDENTIFIER (which is assigned during the registration process), and other profile information (name, authentication method, etc…). It could be anonymous
- Intermediate boxes (NAT or Firewalls) support or not proto-41 forwarding, either within the user's network or within the operator's network

Scenarios (I)

- Realms where the user is already authenticated (Pre-Auth)
  - 3GPP users using cellular devices
  - Users already connected to their ISP (xDSL, Cable, Modem, …)
- Realms where the user is registered but not authenticated yet (Non-Auth)
  - Users willing to use a SC from where they're registered but located in another domain

Scenarios (II)



Scenarios (III)

Case	Host	LAN	CPE	Access	Core	Encapsulation
Case 1	IPv6/IPv4	-	-	IPv4+NAT	IPv4	IPv6/IPv4 IPv6/UDP/IPv4
Case 2	IPv6/IPv4	-	-	IPv4	IPv4	IPv6/IPv4
Case 3	IPv6/IPv4	-	-	IPv4	IPv4	IPv6/IPv4
Case 4	IPv6/IPv4	-	-	IPv4+NAT	IPv4	IPv6/IPv4 IPv6/UDP/IPv4
Case 5	IPv6/IPv4	IPv4	IPv4+NAT	IPv4	IPv4	IPv6/IPv4 IPv6/UDP/IPv4
Case 6	IPv6/IPv4	IPv6/IPv4	IPv6/IPv4	IPv4	IPv4	IPv6/IPv4
Case 6b	-	IPv6/IPv4	IPv6/IPv4	IPv4	IPv4	IPv6/IPv4
Case 7	IPv6/IPv4	IPv6/IPv4	IPv6/IPv4	IPv6	IPv6	IPv4/IPv6
Case 7b	-	IPv6/IPv4	IPv6/IPv4	IPv6	IPv6	IPv4/IPv6
Case 8	IPv6/IPv4	IPv6	IPv6	IPv6	IPv6	IPv4/IPv6
Case 9	IPv6/IPv4	-	-	IPv6	IPv6	IPv4/IPv6

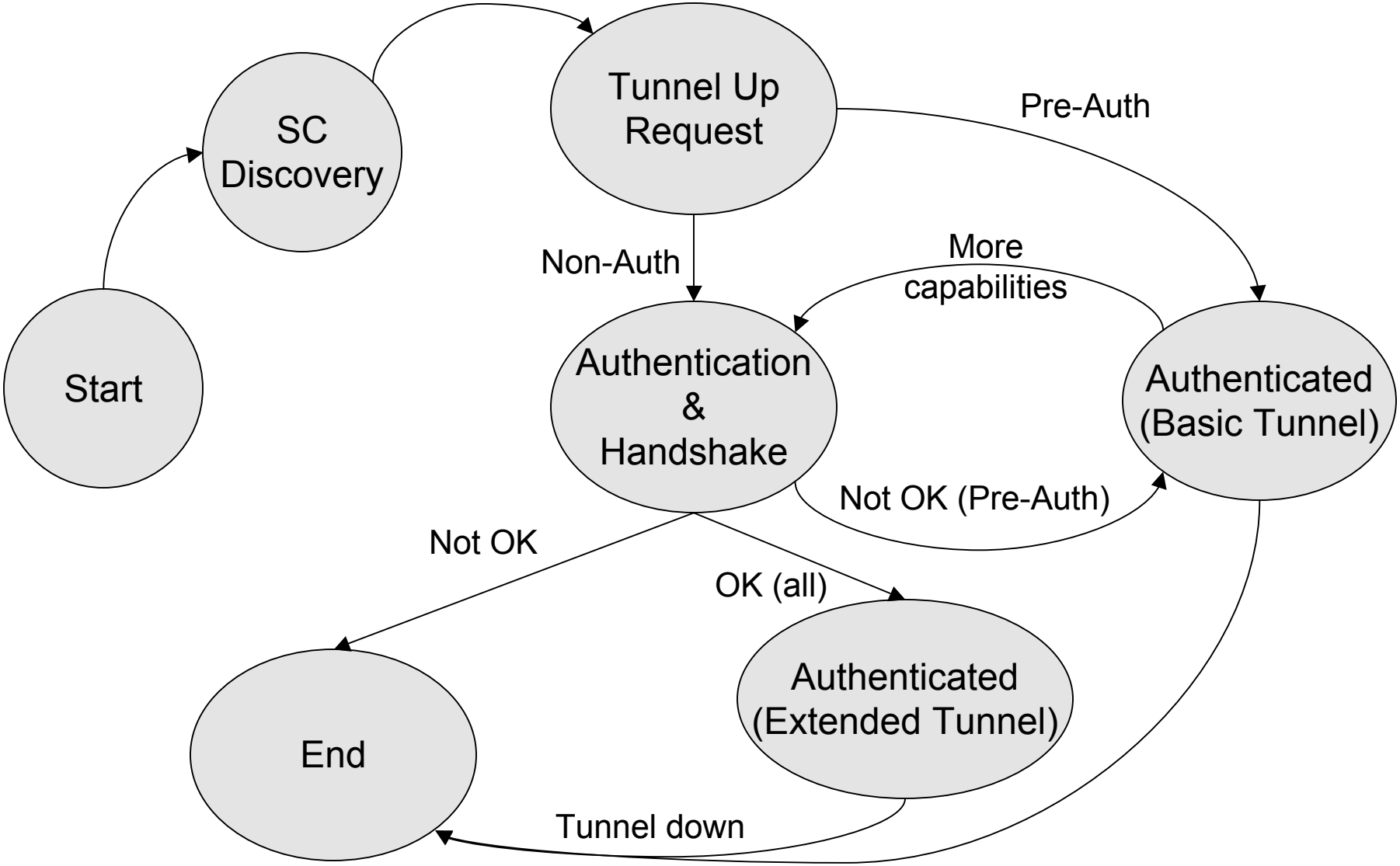
IPv4 -> IPv4 (IV)

o./d.	1	2	3	4	5	6	6b	7	7b	8	9
1	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4/SC	IPv4/SC	IPv4/SC	IPv4/SC
2	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4/SC	IPv4/SC	IPv4/SC	IPv4/SC
3	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4/SC	IPv4/SC	IPv4/SC	IPv4/SC
4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4/SC	IPv4/SC	IPv4/SC	IPv4/SC
5	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4/SC	IPv4/SC	IPv4/SC	IPv4/SC
6	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4/SC	IPv4/SC	IPv4/SC	IPv4/SC
6b	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4/SC	IPv4/SC	IPv4/SC	IPv4/SC
7	IPv4/CPE	IPv4/CPE	IPv4/CPE	IPv4/CPE	IPv4/CPE	IPv4/CPE	IPv4/CPE	IPv4	IPv4	IPv4/CPE	IPv4/CPE
7b	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	IPv4	IPv4	4/6/SC	4/6/SC
8	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	IPv4	4/6/SC
9	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	4/6/SC	IPv4

IPv6 -> IPv6 (V)

o./d.	1	2	3	4	5	6	6b	7	7b	8	9
1	IPv6	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC
2	6/4/SC	IPv6	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC
3	6/4/SC	6/4/SC	IPv6	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC
4	6/4/SC	6/4/SC	6/4/SC	IPv6	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC
5	6/4/SC	6/4/SC	6/4/SC	6/4/SC	IPv6	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC
6	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	IPv6	IPv6	IPv6/CPE	IPv6/CPE	IPv6/CPE	IPv6/CPE
6b	6/4/SC	6/4/SC	6/4/SC	6/4/SC	6/4/SC	IPv6	IPv6	IPv6/CPE	IPv6/CPE	IPv6/CPE	IPv6/CPE
7	IPv6/CPE	IPv6/CPE	IPv6/CPE	IPv6/CPE	IPv6/CPE	IPv6/CPE	IPv6/CPE	IPv6	IPv6	IPv6	IPv6
7b	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6	IPv6	IPv6	IPv6
8	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6	IPv6	IPv6	IPv6
9	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6/SC	IPv6	IPv6	IPv6	IPv6

Tunnel activation: state diagram



Start State

- Only represents the initial state.
- The user’s device is ready to start the activation of the tunnel

SC Discovery State

- Discovery of the Softwire Concentrator (SC) is out the scope of this mechanism specification
- It is assumed that SC is discovered by other external means
- Discovery mechanism will be integrated on the final specification of AST6 protocol
- draft-palet-v6ops-solution-tun-auto-disc-01 could be taken into account

Tunnel Setup Request State

- Once the SC has been discovered, the SI sends a request for the automatic tunnel setup
- The request will be done slightly different depending on
  - the available infrastructure
  - the kind of required tunnel
- If the device is already authenticated (Pre-Auth), then the tunnel request is automatically accepted and a transition to the Authenticated state is done
- If the device is not yet authenticated (Non-Auth), an Authentication and Handshake procedure is required before accepting the tunnel request

Authenticated (Basic Tunnel) State

- This state represents the status on which the SI is already authenticated
- Tunnel is active (up) on both sides and the SI is ready to send/receive data
- It sends/receives all the data by means of the tunnel, as usual
- Periodical keep-alive packets are sent to:
  - detect whether the SI's IP address changes. If so, a transition to the "End State" is forced in order to try to build a new tunnel
  - be sure the tunnel continues up. If don't so, SC does garbage collection
  - refresh NAT/PAT/Firewall tables
  - In IPv6 tunnels ➔ NS
  - In IPv4 tunnels ➔ ping4
- If the user's device won't use the tunnel anymore, it will transit to the "End State"

Authentication & Handshake State

- This state represents the state where the authentication and handshake process is done
- In Pre-Auth
  - Tunnel is up but user might desire extending the features (type of tunnel different to 6in4, prefix delegation, etc.)
  - SC could need extra authentication in order to confirm if user can obtain the solicited extra-features
- In Non-Auth
  - Requires to be authenticated before setting-up the tunnel
- The actions done are:
  - Authenticated and not-authenticate realms:
    - User authentication
    - Handshake to obtain extra-features on the tunnel
    - Transition is done towards either:
      - Authenticated (Basic Tunnel) state if negotiation doesn't succeeds
      - Authenticated (Extended Tunnel) state if negotiation succeeds
  - Only in non-authenticated realms:
    - Getting the IP address of the SI
    - Setting-up the tunnel on both the server and client sides
    - Transition is done towards either:
      - Authenticated (Basic Tunnel) state if negotiation succeeds
      - Authenticated (Extended Tunnel) state if negotiation succeeds
      - End state if negotiation doesn't succeed

Authenticated (Extended Tunnel) State

- Either Pre-Auth and Non-Auth SIs can transition to this state
- SI was successfully authenticated and authorized to set-up a tunnel with extended features
- SI is ready to send/receive data through the tunnel according to such extended features

End State

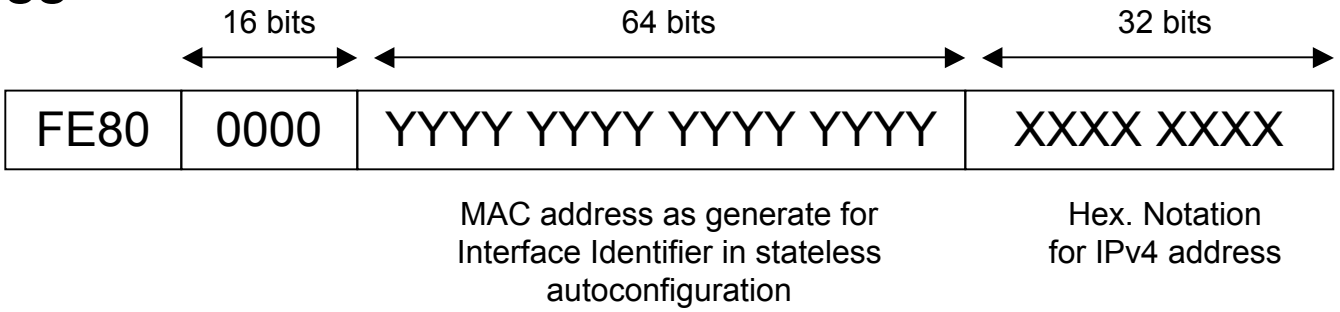
- This state represents the status on which the user's device wants to shut down the tunnel
- No messages to the SC is required because the tunnel is down if it timeouts and no more NA have been received
- END message can be used to indicate the SC that the SI wants to shut down the tunnel
  - This message speeds up the garbage collection process

Authentication & Handshake Options

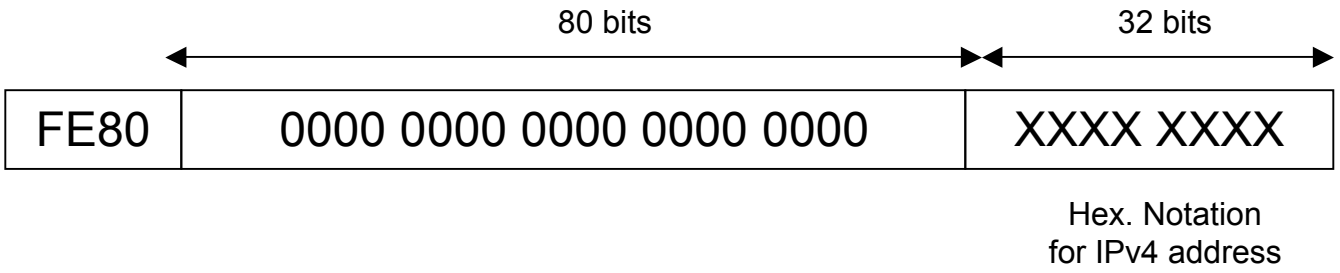
- IPv6 prefix:
  - Using DHCPv6-PD
  - ATS6 built-in capability
- Dynamic/Static prefix
- Keep-Alive Periodicity
  - periodicity of the keep-alive packets may be set to infinite, which in practice means that no keep-alive packets are delivered at all
  - other values are also possible
- NAT type
  - If SC knows details about NAT type, it is indicated
- Ciphering Type
  - Hash function to be used for signing the packets
- Encapsulation Type
  - Different encapsulations are possible: IPv6-in-IPv4, IPv6-in-UDP-IPv4, IPv4-in-IPv6, etc.

Behavior in IPv4-only networks (I)

- SI makes an IPv6 link-local address which has embedded both its public IPv4 address and the MAC address



- SC makes an IPv6 link-local address which has embedded only its public IPv4 address



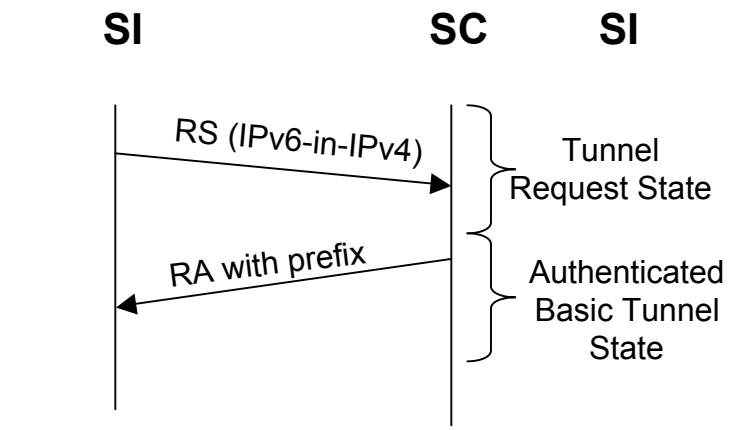
Behavior in IPv4-only networks (II)

- In IPv6 link local:
  - IPv4 address is included ➔ saves routing tables
  - MAC address is included ➔ differentiates several SIs located behind the same NAT
- Basic IPv6 tunnel is automatic
  - it does not require manual configuration
  - it is built by using only a link-local address at each of the tunnel end points
- IPv6 global address is built by using stateless autoconfiguration and the link local address just formed
- Depending on the user's realm, the process is different

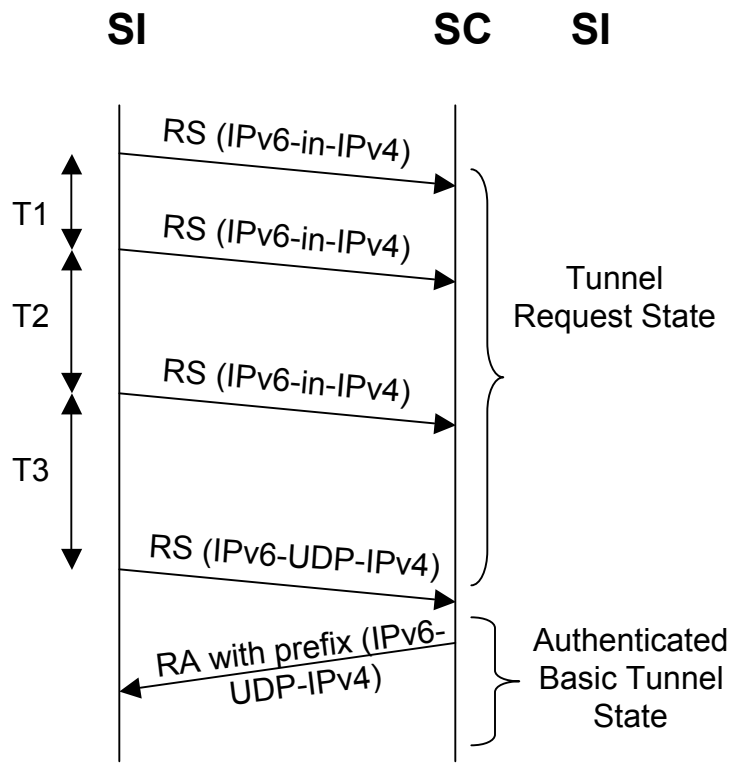
Behavior in IPv4-only networks (III): Pre-Auth realm

- No need to handshake for a Basic Tunnel
- Once the SC receives the IPv4-encapsulated RS from the SI, a transition to the Authenticated (Basic Tunnel) state is done
- SC replies a single IPv4-encapsulated RA and setup the tunnel in its side
- If RA is received ➔ no NAT or proto-41 forwarding one
  - SI builds the global IPv6 by appending the 64 lower bits of the link-local address (Interface Identifier) to the prefix received in the RA
    - IPv4 address is included into the IPv6 one ➔ saves routing tables
    - Part of MAC address is included ➔ differentiates several SIs located behind the same NAT
- If no RA is received ➔ there is a NAT non-proto-41 forwarding
  - All the IPv6 packets are encapsulated into UDP rather IPv4
  - The process for tunnel request is repeated
- If there are more than one SI behind the same NAT, the SC replies RA with M bit set in order to force the transition to the A&H state and negotiating UDP encapsulation

Behavior in IPv4-only networks (IV): Pre-Auth realm

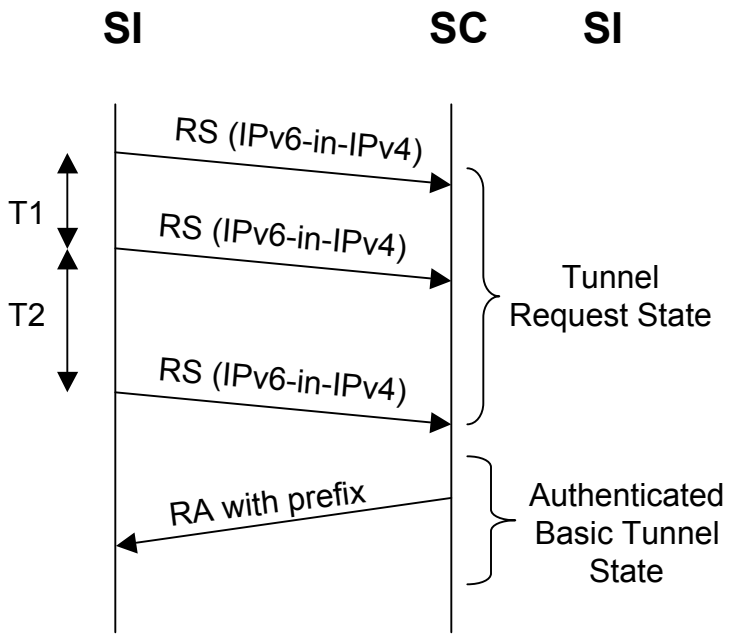


Tunnel Request with no NAT or proto-41 forwarding one

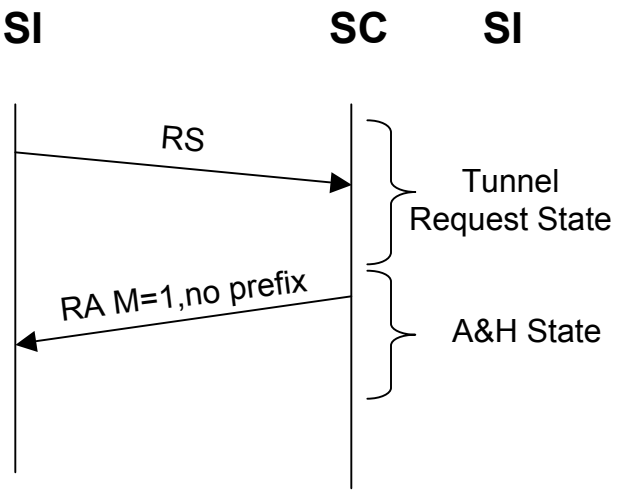


Tunnel Request with NAT non-proto-41 forwarding

Behavior in IPv4-only networks (V): Pre-Auth realm



Tunnel Request with no NAT and two RA lost

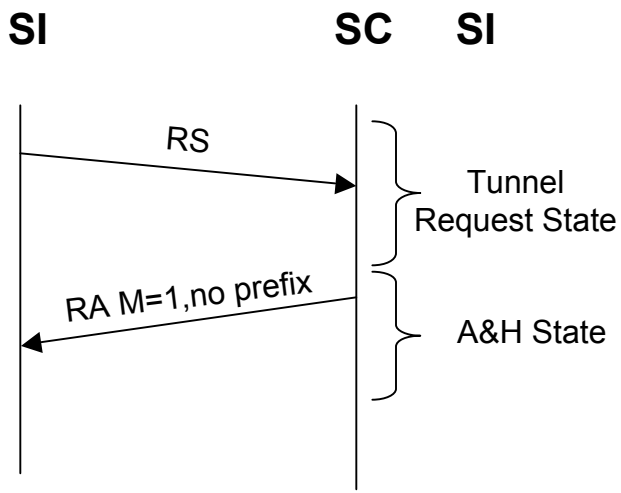


Tunnel Request with more than one SI behind the same NAT

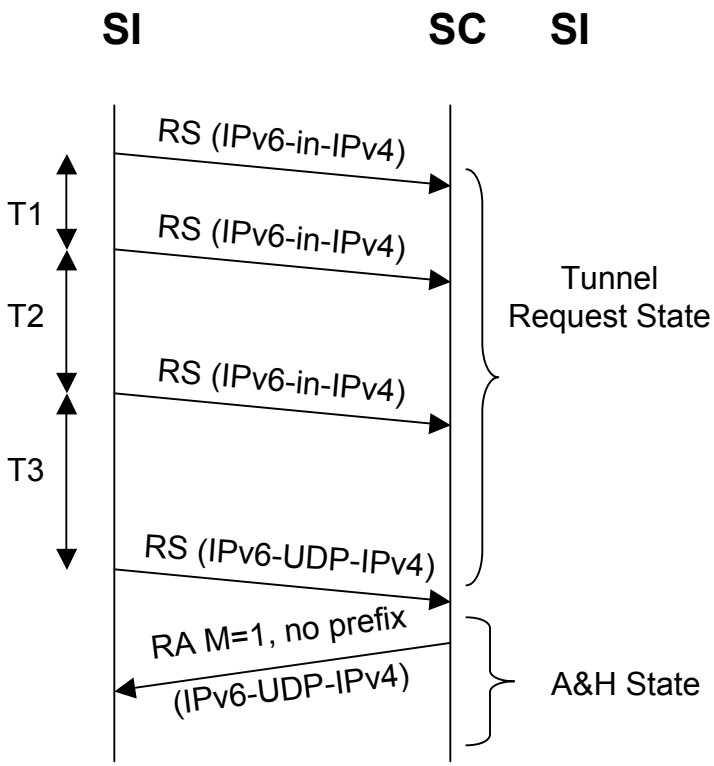
Behavior in IPv4-only networks (VI): Non-Auth realm

- Similar procedure to Pre-Auth
  - Tunnel built with the link local addresses
  - Global address is built by using stateless autoconfiguration
- SI needs to be authenticated
  - Transition to A&H state is forced by replying a RA with M bit set and no prefix
- If RA is received ➔ no NAT or proto-41 forwarding one
  - SI builds the global IPv6 by appending the 64 lower bits of the link-local address (Interface Identifier) to the prefix received in the RA
    - IPv4 address is included into the IPv6 one ➔ saves routing tables
    - Part of MAC address is included ➔ differentiates several SIs located behind the same NAT
- If no RA is received ➔ there is a NAT non-proto-41 forwarding
  - All the IPv6 packets are encapsulated into UDP rather IPv4
  - The process for tunnel request is repeated
- If there are more than one SI behind the same NAT, the SC also replies RA with M bit set in order to force the transition to the A&H state and negotiating UDP encapsulation

Behavior in IPv4-only networks (VII): Non-Auth realm



Tunnel Request with no NAT or more than one SI behind the same NAT

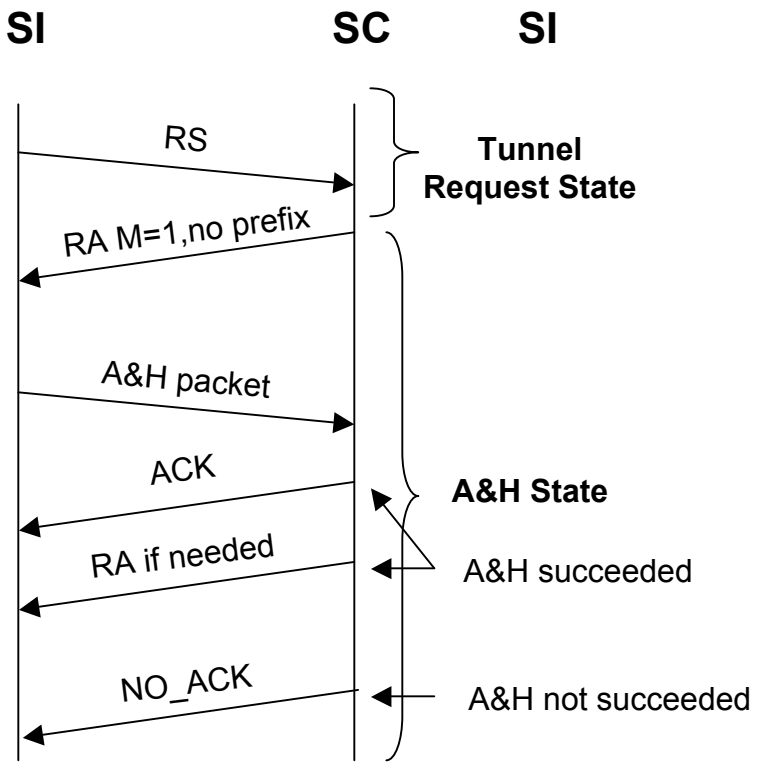


Tunnel Request with NAT non-proto-41 forwarding

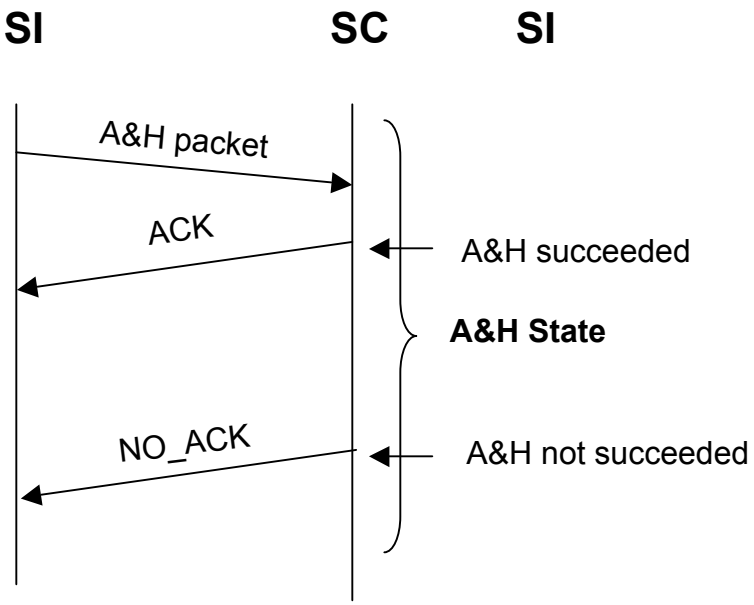
Behavior in IPv4-only networks (VIII): Handshake

- A&H is made by sending new ICMPv6 packet/s (unspecified yet) to the server. Packet/s will contain user's identification, authentication and parameter information
- Packet exchange between SC and SI will be short in time to keep the process as simple as possible
- In Non-Auth realms
  - the SI starts the A&H process during the "Tunnel Setup Request" when the SC returns a RA with the M bit set and no Prefix
- In Pre-Auth realms
  - the SI can start the A&H process at any time from "Authenticated" state
  - SC can also force the A&H from the Tunnel Request state if it detects more than one device behind the same NA. It returns a RA with the M bit set and no Prefix
- A&H packet sent by the SI indicates the options that the SI wishes for setting-up the tunnel
- If user has appropriate rights, SC sends
  - an ACK packet with the setup that is granted
  - RA if required to transition to the Authenticated State
- If user has no rights, SC replies with information about what is wrong by means of a NO_ACK packet

Behavior in IPv4-only networks (IX): Handshake



Handshake for Non-Auth realms and Pre-Auth realms with more than one SI behind a NAT



Handshake for Authenticated users (both Pre-Auth and Non-Auth) requesting for Extended Tunnel

Behavior in IPv4-only networks (X): Choices to Require in Handshake

- Different choices are available to require/provide:
  - Type of encapsulation
  - IPv6 prefix
    - How the IPv6 is delegated
    - Prefix static/dynamic
  - NAT type
  - Keep-alive periodicity

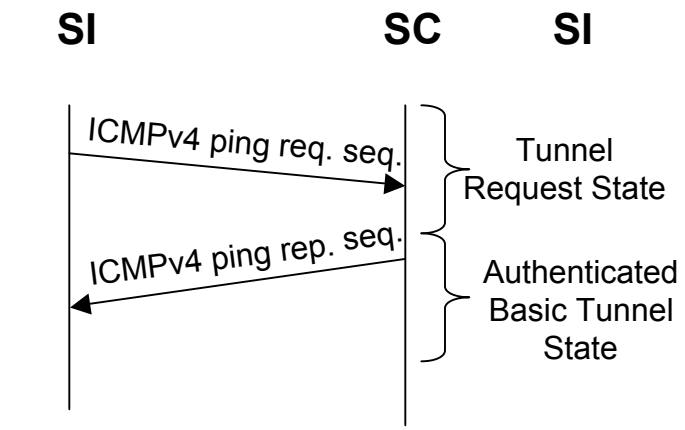
Behavior in IPv6-only networks (I):

- Three choices are available to get the IPv4 address:
  - IPv4 address derived from the IPv6
    - using hash function to be mapped to a private network (10/8)
    - Duplicate Address Detection required
  - DHCP
    - SC should be DHCPv4 server/relay
    - DHCPv6 also to be considered
  - ATS6's built-in mechanism
    - IP address is provided by using ATS6's A&H and ACK signaling packets
- Handshake (if required) is done by using the same A&H packets (ICMPv6) as in IPv4-only networks
  - Native IPv6 support available
  - Other alternatives can be explored (UDP, PPP, etc.)

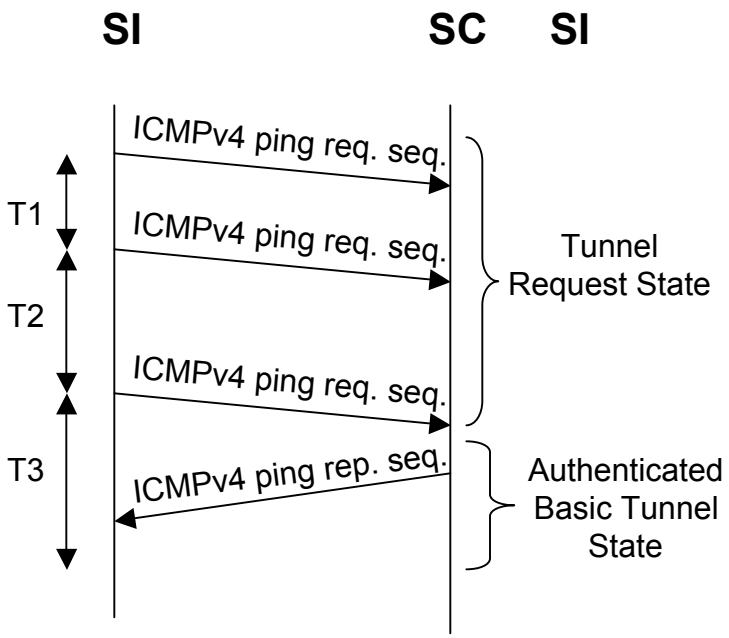
Behavior in IPv6-only networks (II): Pre-Auth realm

- IPv4 address derived from the IPv6 (Basic Tunnel)
  - Tunnel Request is indicated to the SC as a sequence of three predefined-length ICMPv4 ping request packets
  - Source IPv4 address of ping packets is the one extracted from the global IPv6 one
  - IPv4 packets are directly encapsulated into IPv6 packets
  - SC replies with ping reply packets as the user is already authenticated (Pre-Auth realm)
    - If IPv4 address is duplicated, SC doesn't reply
  - If no echo replies are received, SI tries again
  - If no echo replies are received after the third try, SI transitions to A&H state by sending an A&H packet

Behavior in IPv6-only networks (III): Pre-Auth realm

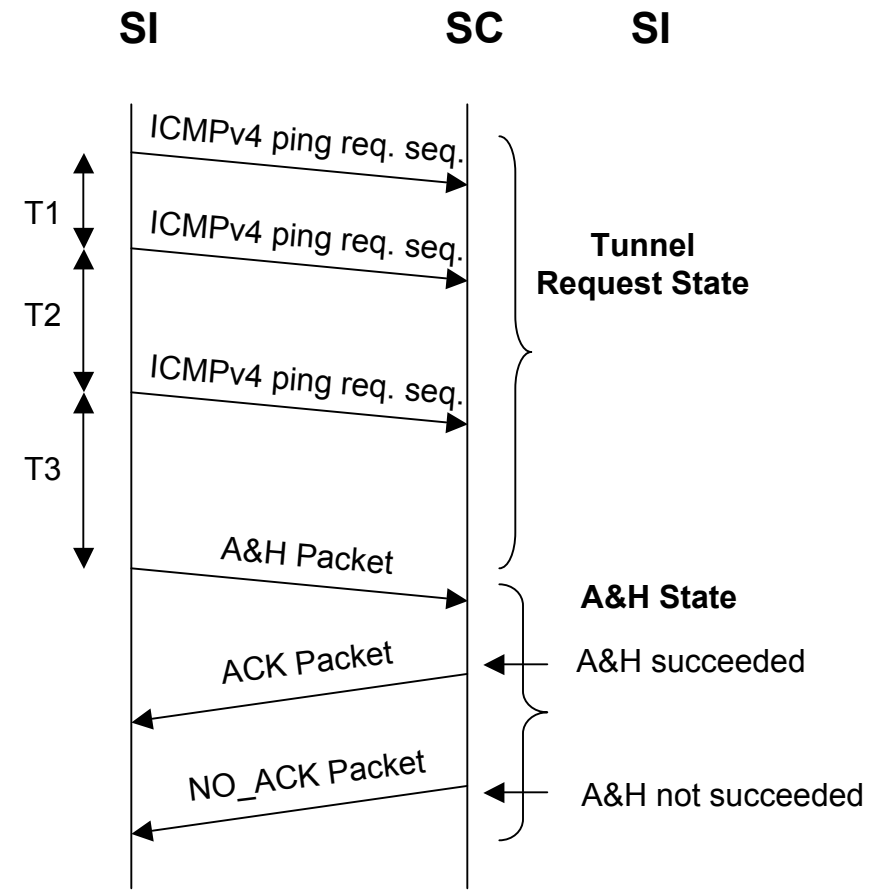


IPv4 Tunnel Request in IPv6-only infrastructures



IPv4 Tunnel request in IPv6-only infrastructures with lost ping replies

Behavior in IPv6-only networks (IV): Pre-Auth realm

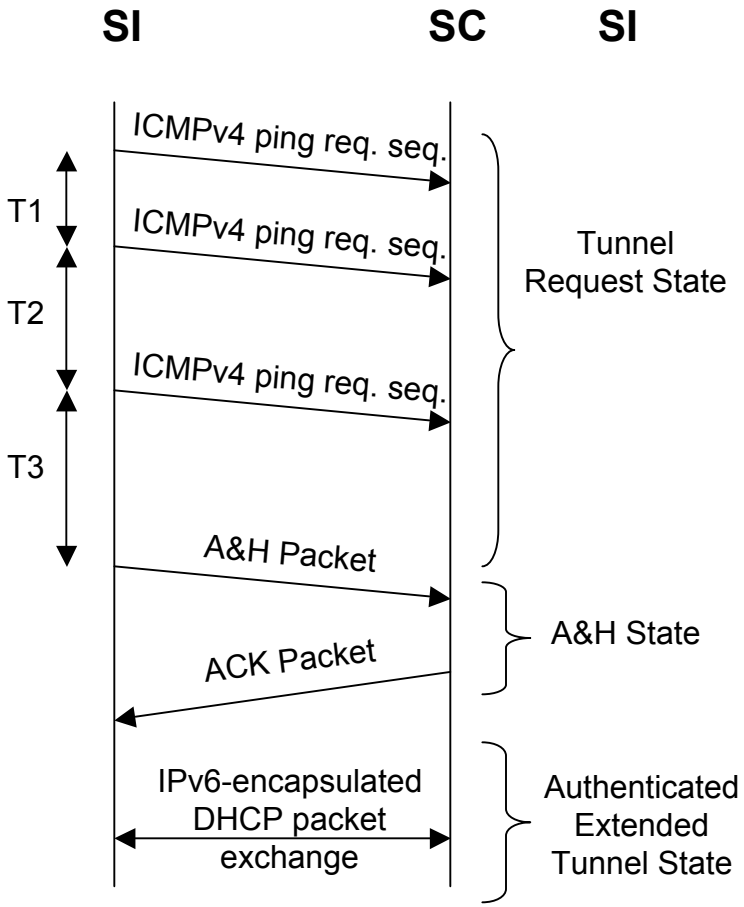


IPv4 Tunnel request in IPv6-only infrastructures with duplicate IPv4 address

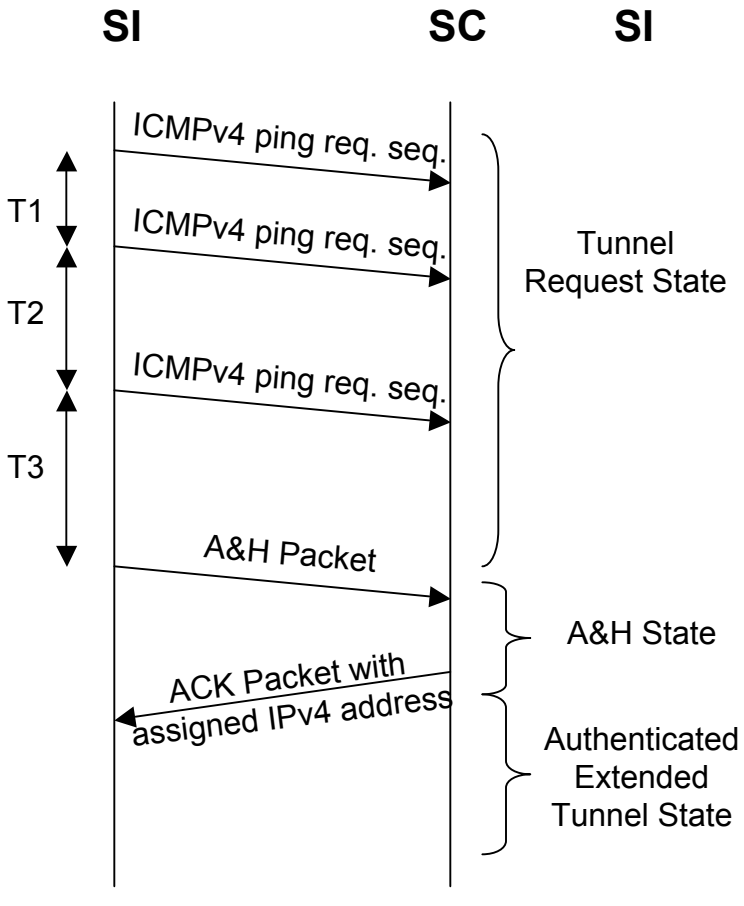
Behavior in IPv6-only networks (V): Pre-Auth realm

- DHCP (Extended Tunnel)
  - Tunnel Request as in Basic Tunnel but the process is done by using an IPv4 link-local address (169.254.0.0/16)
  - SC doesn’t reply to echo ping request and a transition to the A&H state is done
  - IPv6-encapsulated DHCP packets exchange is done
- ATS6 built-in mechanism (Extended Tunnel)
  - Tunnel Request as in DHCP case
  - IPv4 address is provided by the ACK packet within the A&H State

Behavior in IPv6-only networks (VI): Pre-Auth realm



IPv4 Tunnel request in IPv6-only infrastructures with DHCP



IPv4 Tunnel request in IPv6-only infrastructures with ATS6 built-in mechanism

Behavior in IPv6-only networks (VII): Non-Auth realm

- IPv4 addresses derived from the IPv6 (Basic Tunnel)
  - Similar to the Pre-Auth realm but SC doesn't reply to the ping echo request sequence to force the transition to the A&H State
  - Once the ACK is received, tunnel can be considered as activated in both sides, so there is no need for further ICMPv4 reply packets from the SC
- DHCP (Extended Tunnel)
  - Same as the DHCP case in Pre-Auth realms
- ATS6 built-in mechanism (Extended Tunnel)
  - Same as the ATS6's built-in mechanism case in Pre-Auth realms

Behavior in IPv6-only networks (VIII): IPv6 tunnels

- IPv6-in-IPv6 tunnel might be needed in IPv6-only infrastructures
  - i.e. to provide Multicast support
- The process is done as explained in IPv4-only infrastructures
- It is simpler because the IPv6 link-local address is already built

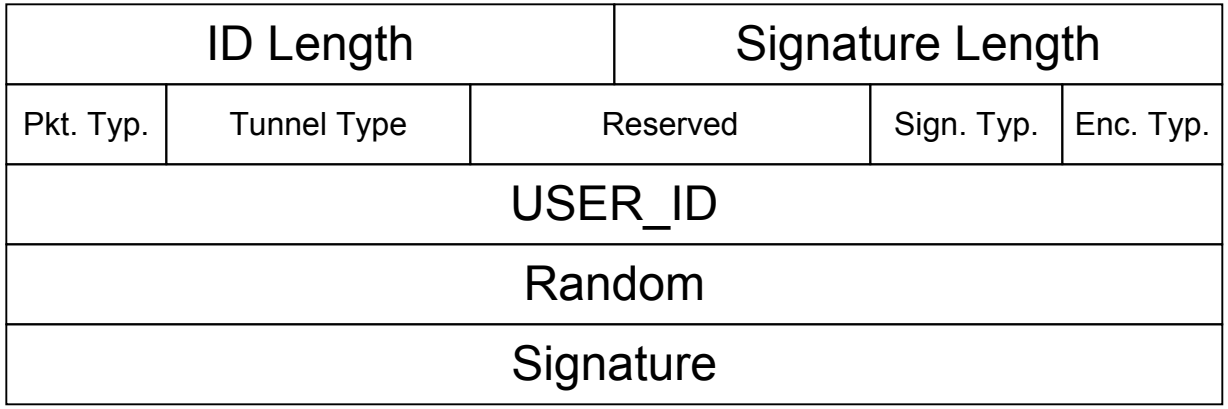
Behavior in IPv6-only networks (IX): Keep-alive packets

- Default keep-alive periodicity will be 60 seconds
  - Configurable in the A&H state
- Two different types
  - IPv4 tunnel: ping packets from SI to SC
    - Reply needed to communicate that the tunnel is up in the SC's side
  - IPv6 tunnel: NS packets as in IPv6 tunnels in IPv4-only infrastructures
    - NA needed to communicate that the tunnel is up in the SC's side

Signaling Packets

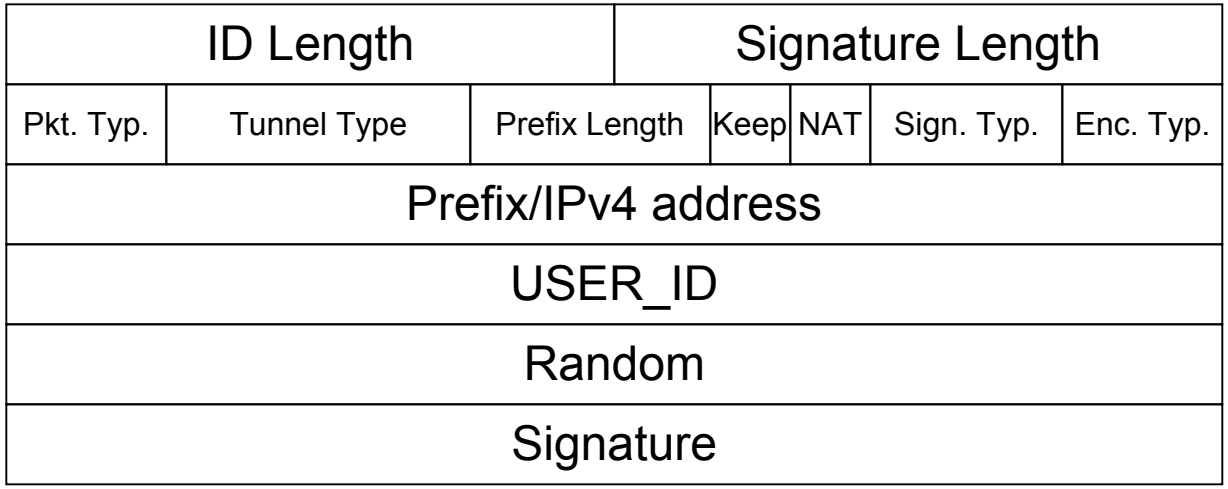
- ICMPv6 packets to be standardized.
  - Other choices can be also explored (UDP, PPP, etc.)
- A&H packet
  - Pre-Auth realms: to extend the capabilities of the basic tunnel
  - Non-Auth realms: to create the basic tunnel and/or to extend the capabilities of the basic tunnel
- ACK packet
  - To acknowledge the SI request
  - To inform about the granted choices
- NO_ACK packet
  - To not-acknowledge the SI request
  - To inform about a failure in the A&H request

A&H Packet



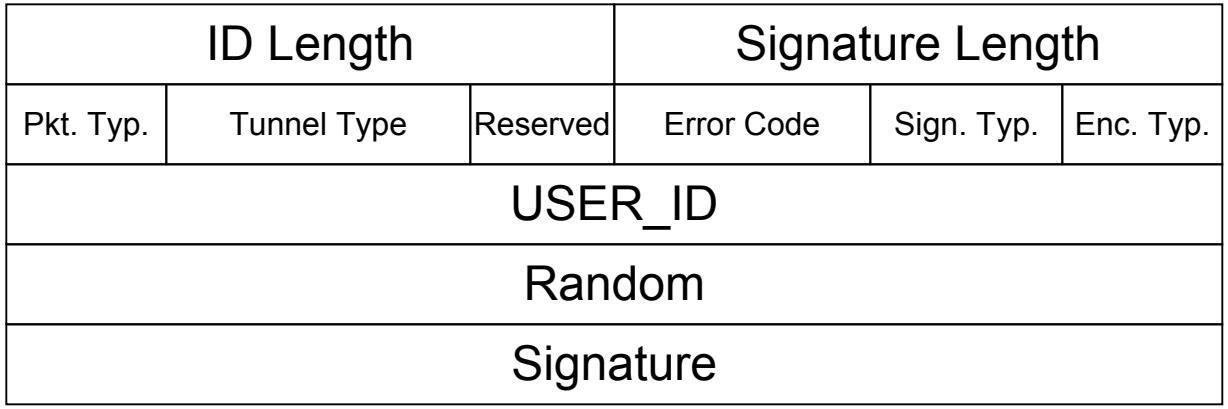
- ID Length (16 bits): The length of the USER_ID field
- Signature Length (16 bits): The length of Signature field
- Packet Type (4 bits): Information about the packet type (A&H, ACK or NO_ACK)
- Tunnel Type (5 bits): Five flags indicating the required tunnel
- Reserved (13 bits): Reserved bits for future use
- Signature Type (4 bits): The type of signature to be used in the handshake process
- Encapsulation Type (6 bits): Six flags indicating the required encapsulation type
- USER_ID: The user login. It is assigned during the registration process. To be further defined
- Random data: Data used to be included on the packet to prevent duplicate signatures. Either a random number, date, etc. To be further defined.
- Signature: It is the field that actually authenticates the user. It is the result of ciphering with the private key the result of hashing the packet with a hash function (MD5, SHA1, ...). To be further defined.

ACK Packet



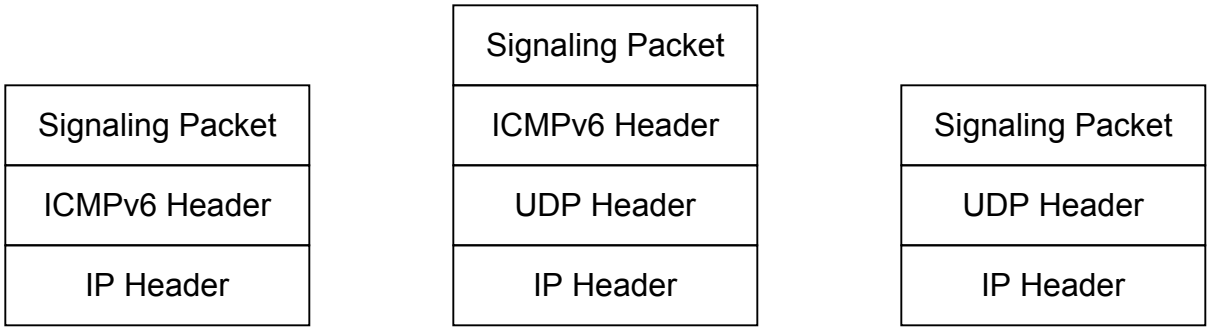
- ID Length (16 bits): The length of the USER_ID field
- Signature Length (16 bits): The length of Signature field
- Packet Type (4 bits): Information about the packet type (A&H, ACK or NO_ACK)
- Tunnel Type (5 bits): Five flags indicating the granted tunnel
- Prefix Length (7 bits): Indicates the desired prefix length
- Keep-Alive Periodicity (3 bits): The periodicity defined by the SC for the keep-alive packets
- NAT Type (3 bits): NAT type known by the SC
- Signature Type (4 bits): The type of signature to be used in the handshake process
- Encapsulation Type (6 bits): Six flags indicating the required encapsulation type
- Prefix/IPv4 address (64 bits): IPv6 prefix or IPv4 address assigned to the SI
- USER_ID: The user login. It is assigned during the registration process. To be further defined
- Random data: Data used to be included on the packet to prevent duplicate signatures. Either a random number, date, etc. To be further defined.
- Signature: It is the field that actually authenticates the user. It is the result of ciphering with the private key the result of hashing the packet with a hash function (MD5, SHA1, ...). To be further defined.

NO_ACK Packet



- ID Length (16 bits): The length of the USER_ID field
- Signature Length (16 bits): The length of Signature field
- Packet Type (4 bits): Information about the packet type (A&H, ACK or NO_ACK)
- Tunnel Type (5 bits): Five flags indicating the required tunnel
- Reserved (13 bits): Reserved bits for future use
- Error Code (8 bits): The reason of the "no acknowledgment"
- Signature Type (4 bits): The type of signature to be used in the handshake process
- Encapsulation Type (6 bits): Six flags indicating the required encapsulation type
- USER_ID: The user login. It is assigned during the registration process. To be further defined
- Random data: Data used to be included on the packet to prevent duplicate signatures. Either a random number, date, etc. To be further defined.
- Signature: It is the field that actually authenticates the user. It is the result of ciphering with the private key the result of hashing the packet with a hash function (MD5, SHA1, ...). To be further defined.

Signaling Encapsulation



- In principle, the signaling information is encapsulated as ICMPv6 payload, type to be standardized
  - simpler because networks will tend to have over the time, more native IPv6 support
  - simplifies both the resources and the implementation of ATS6 capable SCs and SIs
- Other choices are also possible
  - ICMPv6 packets encapsulated in UDP ones
  - UDP packets
  - TCP packets
  - combination of them
- The way how the signaling packets are encapsulated is not as important as the signaling itself

Peer-to-peer optimization

- ATS6 provides a way to avoid all the encapsulated traffic being handled by the SC
  - direct peer-to-peer among SIs is wanted, when they are connected in the same IPv4-only infrastructure
- Lots of resources (network, memory, CPU load, etc.) are saved at the SC
  - Also lower RTT, improving the protocol scalability
- An specific IPv6 prefix could be reserved for that purpose
  - Teredo prefix should be explored to study compatibility with Teredo clients
- Peer-to-peer optimization should use IPv6-in-UDP-IPv4 encapsulation

Non-technical requirements

- 0) Is the solution based upon any existing technology (reuse)?
  - Yes:
    - Autoconfiguration, DHCPv6, DHCPv6-PD, DHCP, 6in4, 4in6, 6in6, PPP, etc.
- 1) Is the solution documented (published)?
  - Yes:
    - http://www.ietf.org/internet-drafts/draft-palet-softwires-ats6-01.txt
- 2) Are there any known issues in the solution (completeness)?
  - No
- 3) Has the solution been fully implemented (status idea)?
  - No by now. Work being done.
- 4) Do two independent, commercially supported, demonstratetively inter-operable implementations of all the components of the underlying technology exist (interop)?
  - No by now
- 5) Have ISPs experimented with all the components of the solution successfully (deployment)?
  - No experimentation with the protocol, but there is much experimentation with the protocols used by ATS6 (tunnels, DHCPv6, etc.)

Technical requirements (I)

- 0) Support Hub & Spoke cases
  - a. NAT traversal
    - Yes
  - b. Nomadicity (outer address may change)
    - Yes
- 1) Address allocation
  - a. End point
    - Yes
  - b. Prefix delegation
    - Yes
- 2) Scalability
  - a. To the millions
    - Yes
  - b. Set-up time
    - Minimum case: 2 packets exchange for pre-authenticated users behind proto-41 NAT or not NAT
    - Typical case 1: 5 packets exchange for pre-authenticated users behind non-proto-41 NAT
    - Typical case 2: 5 packets exchange for non-authenticated users behind proto-41 NAT
    - Typical case 3: 8 packets exchange for non-authenticated users behind non-proto-41 NAT
- 3) Multicast support
  - Yes

Technical requirements (II)

- 4) Authentication/Security
  - a. Integration with deployed AAA solutions
    - Yes
  - b. Control/signaling
    - Yes
  - c. PDU
    - It depends on the type of tunnel: Examples:
      - 6in4: PDU at layer 2 minus IPv4 header
      - IPv6-UDP-IPv4: PDU at UDP minus IPv6 header
- 5) OAM
  - a. Keep alive for NAT traversal
    - Yes
  - b. Logging / accounting
    - Yes
  - c. End point failure detection (inside the softwire)
    - Yes (keep alive)
  - d. Path failure detection (outside the softwire)
    - Yes

Technical requirements (III)

- 6) Available encapsulations
  - a. IPv6/IPv4
    - Yes
  - b. IPv6/UDP/IPv4
    - Yes
  - c. IPv4/IPv6
    - Yes
- 7) L2 and L3 connectivity
  - Yes