Abstract

This document specifies some extensions to the Active Network Encapsulation Protocol (ANEP) when it is used over IPv6 [RFC1883].

1. Raisons d’etre

The reasons for extending the ANEP protocol are:

a) Taking into account the jumbo payload of the IPv6 packet.

b) Be compliant with IPv6 upper-layer checksuming scheme.
3. Jumbo payload

The format of the ANEP header is:

```
+---------------------------------------------+---------------------------------------------+---------------------------------------------+
|    Version     |     Flags     |            Type ID            |
+---------------------------------------------+---------------------------------------------+---------------------------------------------+
| ANEP Header Length | ANEP Packet Length |
+---------------------------------------------+---------------------------------------------+
| Options                  |                               |
+---------------------------------------------+---------------------------------------------+
| Payload                  |                               |
+---------------------------------------------+---------------------------------------------+
```

When an ANEP packet is carried by an IPv6 jumbo-gram, the fields header and packet length of the ANEP packet can be too short to express the true size of the packet.

One can use the same method as for TCP and set the ANEP Packet Length field to 0 when using a jumbo-gram, the packet size would be determined by using the IPv6 jumbo payload option.

This is easy to implement, but doesn’t remove the limitation on the ANEP Header Length size which could also be too short.

What we propose is to use the ANEP Packet Length as a way to extend the ANEP Header Length when there is a need to have more than $2^{18}$ bytes for the ANEP Header (the length is expressed in blocks of 4 bytes). So that the size of the ANEP Header would be computed by:

$$(\text{ANEP Header Length} \ll 2) + (\text{ANEP Packet Length} \ll 18)$$

and the ANEP Packet Length could not exceed a value of $2^{14} - 1$.

This has the advantage to be compatible with the TCP method when the ANEP Header is short enough.
4. Integrity Checksum

One need to ensure that the source and destination addresses as well as the packet type included in the IPv6 header have not been altered during the transfer.

This could be done for the addresses by using the ANEP options source and destination identifiers; in this case the redundancy is used as a way to check the addresses haven’t change, but this doesn’t preserve from altered packet type.

One will take the same method that the one used with UDP by making the checksum mandatory and computed by including an IPv6 pseudo header.

The format of the checksum option is:

```
+-------------+-------------+-------------+-------------+
| FLG | Option Type | Option Length | Option Payload (Option Value) |
+-------------+-------------+-------------+-------------+
```

For the Integrity Checksum, the Option Type field value is 3. For computing the checksum, the payload of this Option must be set to zero. The Option Length field must be 2.

The payload of this option contains the 16 bit one’s complement of the one’s complement sum of the entire ANEP packet, starting with the ANEP Version field [RFC1071] [RFC1141] [RFC1624], AND including the IPv6 pseudo header [RFC1883].

This option should be:
- Public (Bit 0 of FLG must be set to 0),
- Processed by the node (Bit 1 of FLG set to 1).

If this option is missing or appeared more than once the node should discard the packet.
5. References


Authors' Addresses:

LORIA/ INRIA Lorraine
615 rue du Jardin Botanique
54602 Villers-les-Nancy CEDEX
France
tel: +33 3.83.59.20.00
Email: Surname.Name@loria.fr