IPv4-in-IPv6

Massimo Bernaschi
m.bernaschi@iac.cnr.it

Massimo Vellucci
m.vellucci@unicampus.it

Luca Vollero
l.vollero@unicampus.it

Introduzione

Che cos'è IPv4-in-IPv6 ?

È una tecnica d'incapsulamento (stile VPN) che consente di trasferire pacchetti IPv4 su di un backbone IPv6

Tipologia di VPN

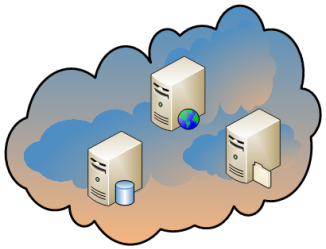
Di livello 3 senza la necessità di configurare in modo esplicito i tunnel (Tunneling Automatico)

Ambito di utilizzo

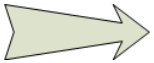
Interconnettere domini IPv4 isolati attraverso una rete IPv6. I domini IPv4 possono appartenere a distinti Autonomous System (AS)

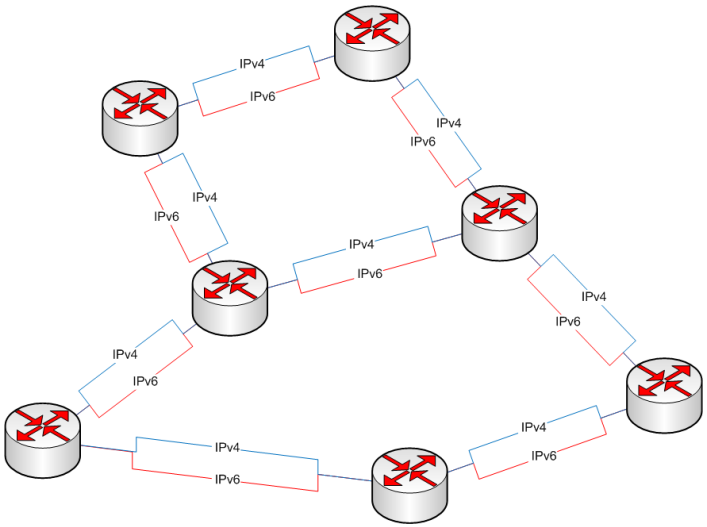
Ipotetica situazione di servizi in IPv4 e IPv6

Server / Servizi IPv4



Server / Servizi IPv6

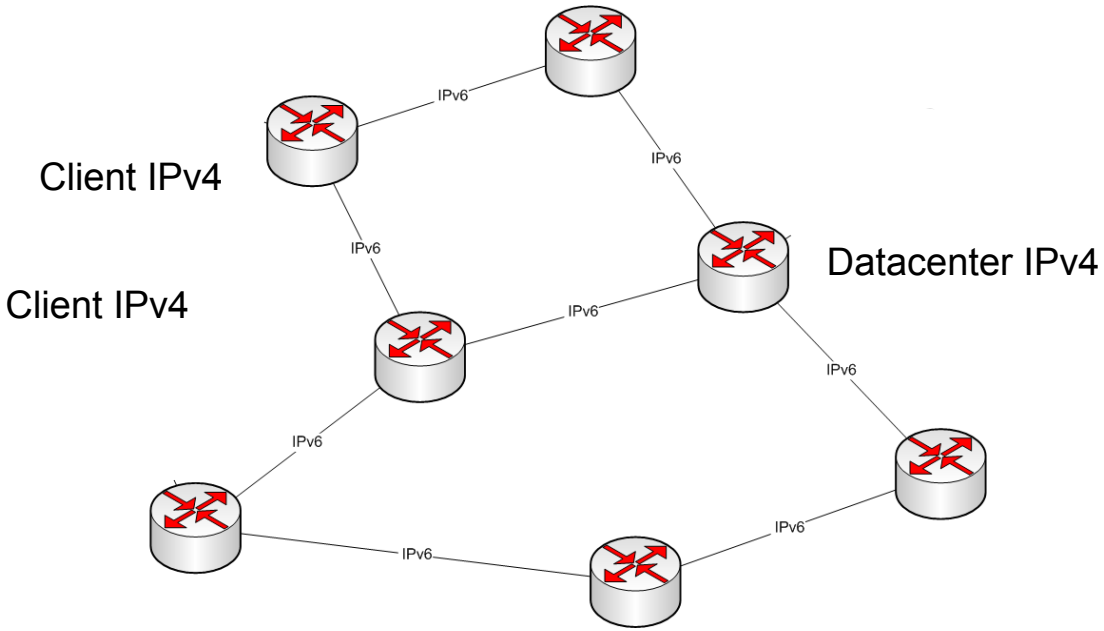




Backbone DualStack

Per consentire ai Client di accedere a Server in IPv4 è necessario che il Backbone gestisca IPv4

Visione per lo switch in IPv6



Svincolare il Backbone dall'IPv4 senza compromettere la fruizione dei servizi in IPv4

Caratteristiche principali della soluzione proposta 1/2

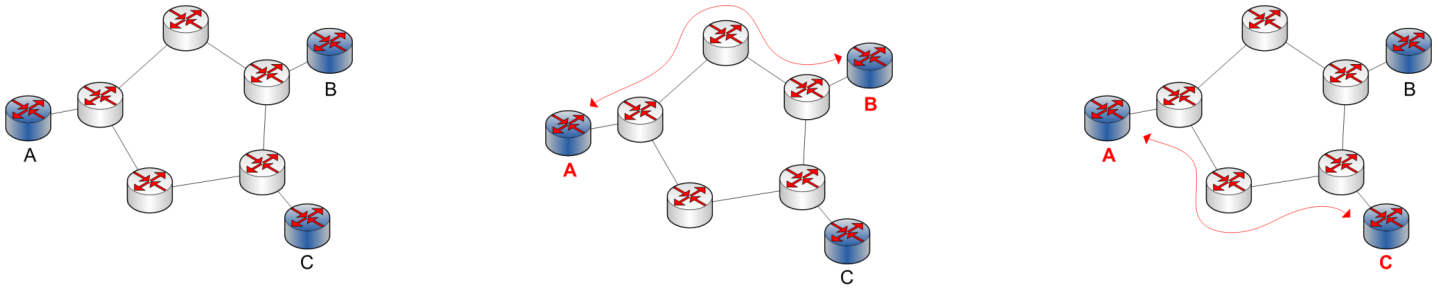
• Gli indirizzi IPv6 degli *endpoint* del tunnel sono autoconfigurati con gli indirizzi IPv4 sorgente/destinazione del pacchetto da incapsulare

IPv4 Sorgente 210.121.52.23 → IPv6 VPN Sorgente 2003::D279:3417

IPv4 Destinazione 74.125.71.94 → IPv6 VPN Destinazione 2003::4A7D:475E

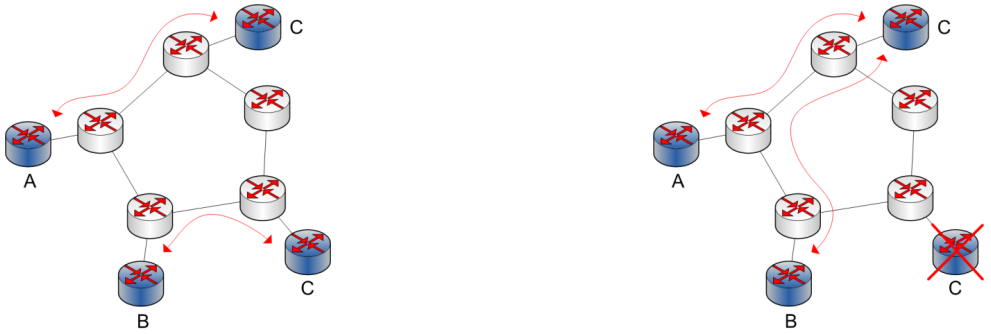
• Non richiede nessun protocollo di controllo. La soluzione si base sull'efficienza dei protocolli di routing come OSPF e BGP.

• Il traffico incapsulato raggiunge la destinazione tramite il percorso più breve



Caratteristiche principali della soluzione proposta 2/2

- Garantisce ridondanza e distribuzione del carico



- I pacchetti IPv6 ereditano i parametri QoS del traffico IPv4

- È possibile instaurare tunnel anche tra differenti AS

2003::IPv4

Andare oltre l’idea

Implementato un modulo kernel in ambiente Linux

Viene creata un’interfaccia virtuale (ip6encap) che si occupa di incapsulare/decapsulare il traffico in IPv4.

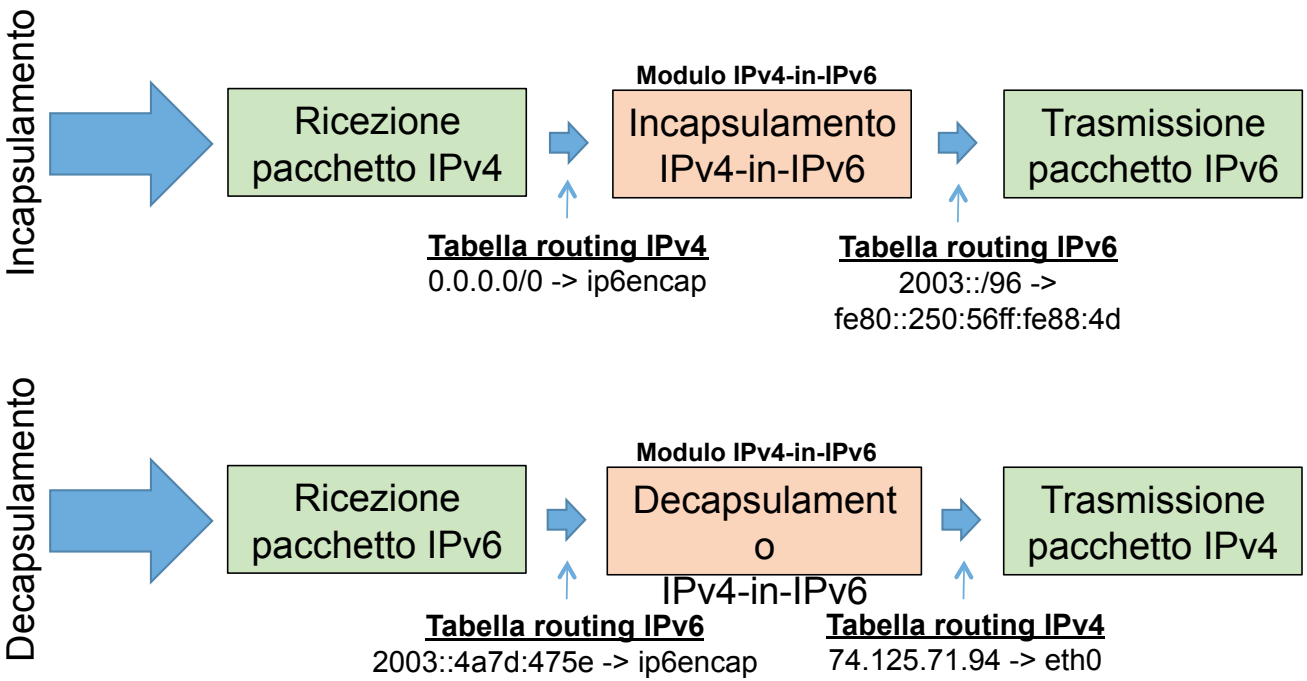
Nessun modifica al *core* di linux

Gestione tramite le tabelle di routing

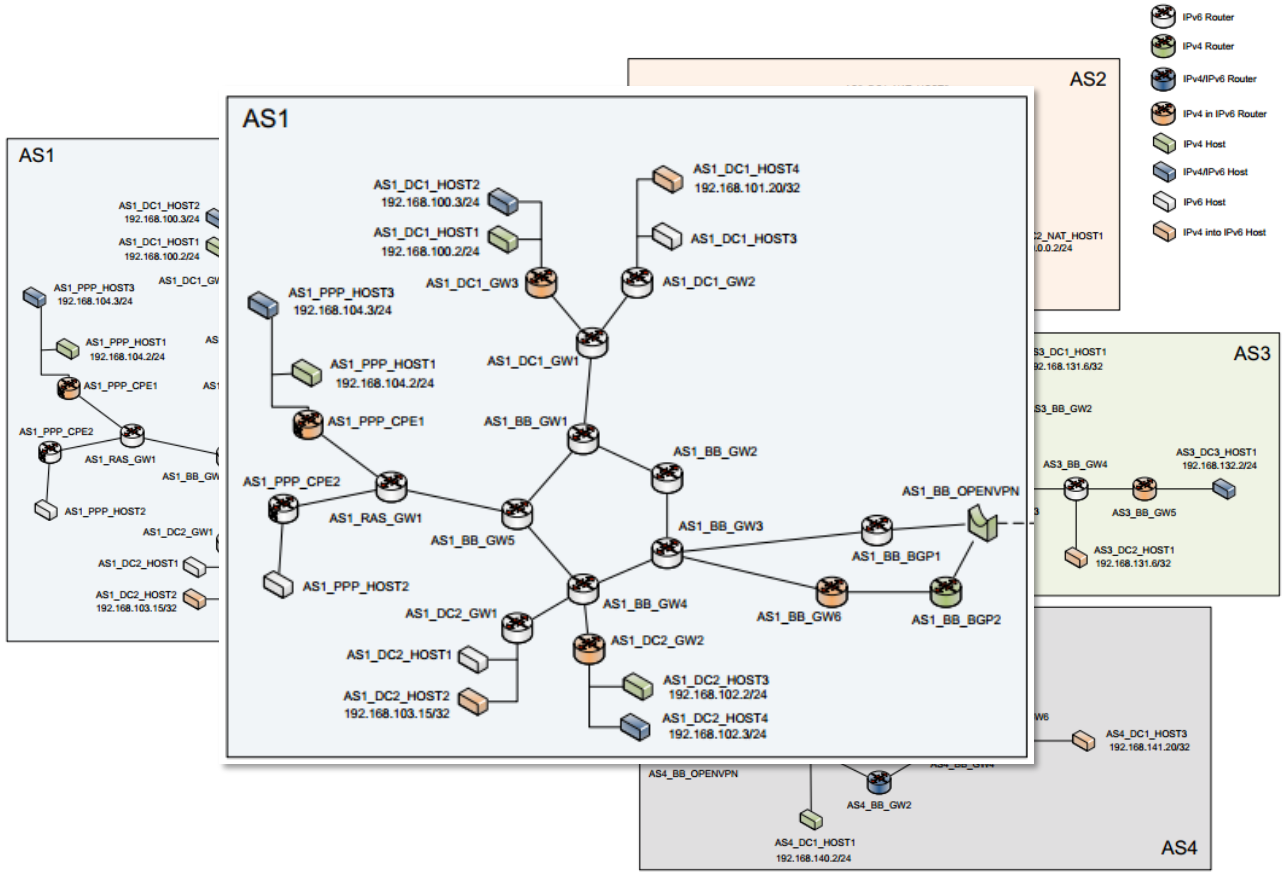
L’incapsulamento viene controllato esclusivamente tramite la tabella di routing dell’host

Tabella routing IPv4	**Tabella routing IPv6**
0.0.0.0/0 -> ip6encap	2003::4a7d:475e -> ip6encap
74.125.71.94 -> eth0	2003::/96 -> fe80::250:56ff:fe88:4d

Fasi del flusso dei pacchetti IPv4



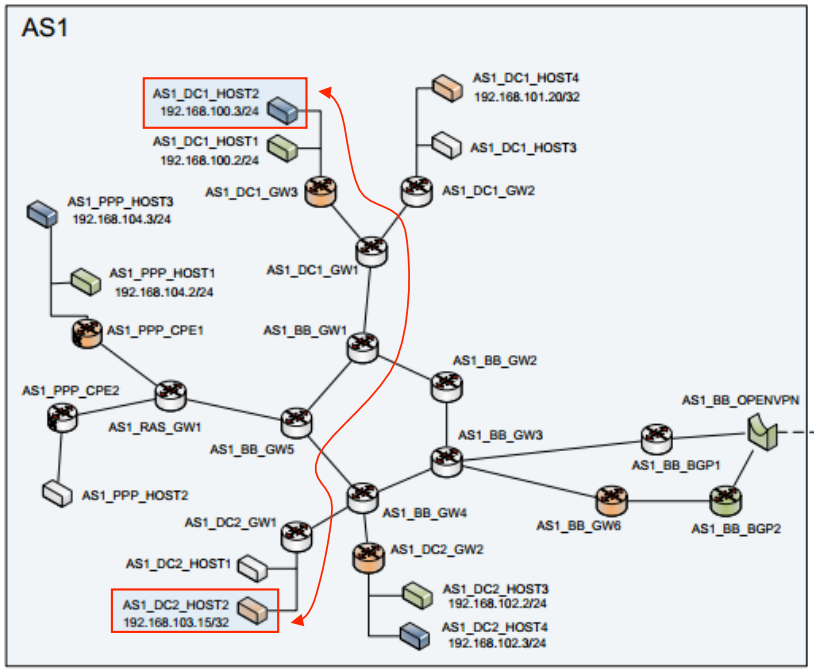
Testbed / Stress test 1/2



Testbed / Stress test 2/2

- Ogni AS è implementato su di un distinto server di virtualizzazione
- Ogni nodo è una macchina virtuale basata su OpenWRT con soltanto 8MB di Ram
- Gli AS sono dislocati in sedi differenti
- Per consentire il peering tra AS, il link fisico di interconnessione è composto da VPN Layer 2 create con OpenVPN
- Sono utilizzati i protocolli di routing OSPF e BGP tramite la suite Quagga
- Lo stress test consiste in flussi di traffico IPv4 (incapsulato) tra AS differenti per vari giorni senza interruzione
- I test di throughput sono svolti esclusivamente all'interno di un singolo AS
- I risultati sul throughput sono basati sull'analisi del traffico di rete catturato con l'applicazione tcpdump

Alcuni numeri 1/2



Traceroute IPv4

- AS1_DC1_GW3 (192.168.100.1)
- AS1_DC1_HOST2 (192.168.100.3)

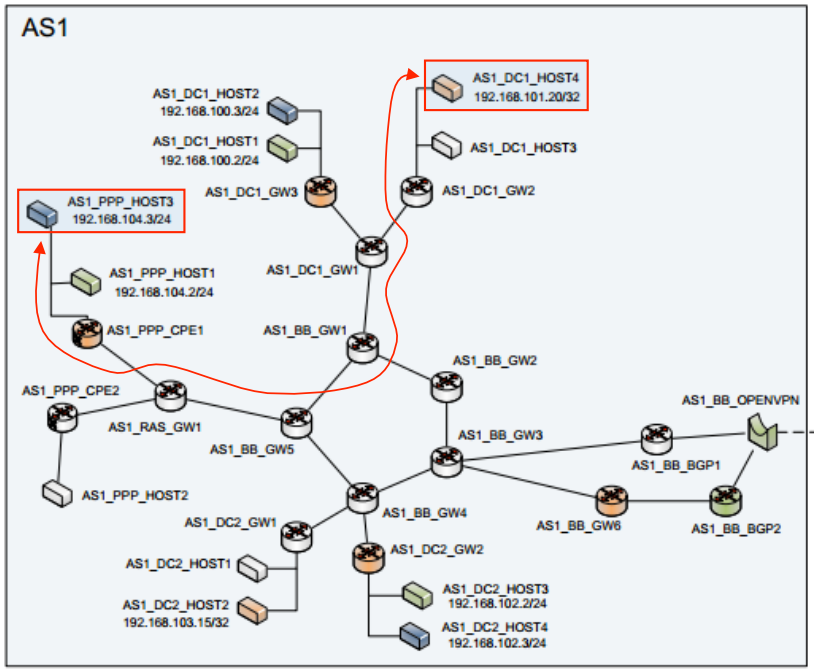
Traceroute IPv6

- AS1_DC2_GW1 (2001:0:0:3::1)
- AS1_BB_GW4 (fc00:0:0:1::4)
- AS1_BB_GW5 (fc00:0:0:1::5)
- AS1_BB_GW1 (fc00:0:0:1::1)
- AS1_DC1_GW1 (fc00:0:0:2::1)
- AS1_DC1_GW3 (2001:0:0:2::1)
- AS1_DC1_HOST2 (2001::2:250:56ff:fe88:2d)

Download tramite HTTP di un file da 500MB

- IPv4: 1,74 MBps – 9928 pkt/s
- IPv6: 1,80 MBps – 10254 pkt/s

Alcuni numeri 2/2



Traceroute IPv4

- AS1_PPP_CPE1 (192.168.104.1)
- AS1_DC1_HOST4 (192.168.101.20)

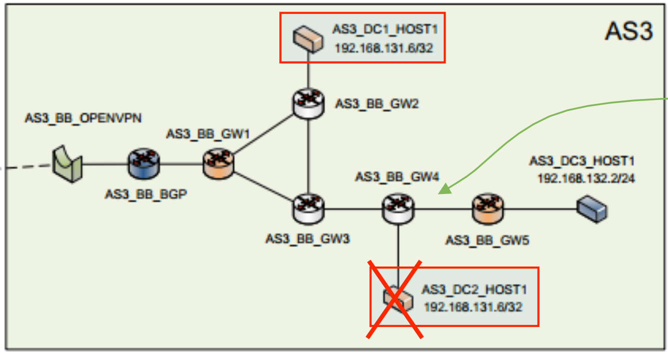
Traceroute IPv6

- AS1_PPP_CPE1 (2001:0:1:1::1)
- AS1_RAS_GW1 (2001:0:0:1::1)
- AS1_BB_GW5 (fc00:0:0:1::5)
- AS1_BB_GW1 (fc00:0:0:1::1)
- AS1_DC1_GW1 (fc00:0:0:2::1)
- AS1_DC1_GW2 (2001:0:0:1::1)
- AS1_DC1_HOST4 (2001::1:250:56ff:fe88:2b)

Download tramite HTTP di un file da 500MB

- IPv4: 1,53 MBps – 9187 pkt/s
- IPv6: 1,60 MBps – 9566 pkt/s

Ridondanza / Distribuzione dei Servizi




```
root@AS3_BB_GW4:~# ip -6 r | grep 2003
2003::c0a8:8306 via fe80::250:56ff:fe88:4d dev eth1  proto zebra  metri
1500 advmss 1440 hoplimit 0
2003::c0a8:8400/120 via fe80::250:56ff:fe88:49 dev eth2  proto zebra  m
root@AS3_BB_GW4:~# ip -6 r | grep 2003
2003::c0a8:8306 via fe80::250:56ff:fe88:4d dev eth1  proto zebra  metri
1500 advmss 1440 hoplimit 0
2003::c0a8:8400/120 via fe80::250:56ff:fe88:49 dev eth2  proto zebra  m
root@AS3_BB_GW4:~# ip -6 r | grep 2003
2003::c0a8:8306 via fe80::250:56ff:fe88:4d dev eth1  proto zebra  metri
1500 advmss 1440 hoplimit 0
2003::c0a8:8400/120 via fe80::250:56ff:fe88:49 dev eth2  proto zebra  m
mtu 1500 advmss 1440 hoplimit 0
2003::c0a8:8306 via fe80::250:56ff:fe88:45 dev eth0  proto zebra  metri
1500 advmss 1440 hoplimit 0
2003::c0a8:8400/120 via fe80::250:56ff:fe88:49 dev eth2  proto zebra  m
mtu 1500 advmss 1440 hoplimit 0
2003::c0a8:8306 via fe80::250:56ff:fe88:4d dev eth1  proto zebra  metri
1500 advmss 1440 hoplimit 0
2003::c0a8:8400/120 via fe80::250:56ff:fe88:49 dev eth2  proto zebra  m
mtu 1500 advmss 1440 hoplimit 0
```

```
64 bytes from 192.168.131.6: seq=33 ttl=63 time=0.344 ms
64 bytes from 192.168.131.6: seq=34 ttl=63 time=0.391 ms
root@AS3_DC3_HOST1:~# ping 192.168.131.6
PING 192.168.131.6 (192.168.131.6): 56 data bytes
64 bytes from 192.168.131.6: seq=0 ttl=63 time=0.539 ms
64 bytes from 192.168.131.6: seq=1 ttl=63 time=0.395 ms
64 bytes from 192.168.131.6: seq=2 ttl=63 time=0.354 ms
64 bytes from 192.168.131.6: seq=3 ttl=63 time=0.376 ms
64 bytes from 192.168.131.6: seq=4 ttl=63 time=0.455 ms
64 bytes from 192.168.131.6: seq=75 ttl=63 time=0.660 ms
64 bytes from 192.168.131.6: seq=76 ttl=63 time=0.625 ms
```

Criticità

Tabella di routing con rotte 2003::/96

Per mantenere l'incapsulamento tra differenti AS devo annunciare tramite BGP tutte le classi 2003::IPv4 ? No

Traceroute IPv4 incompleto

Non si può risalire al percorso assunto dal traffico incapsulato all'interno del Backbone IPv6. Si potrebbe avere come risultato che ho girato il mondo in un singolo *Hop*

Header IPv6 aggiuntivo

Calo del throughtput a causa dell'introduzione dell'header di incapsulamento IPv6

Multicast

Necessario gestire il protocollo IGMP

Evoluzioni

Fornire servizi in base al prefisso 2003::/96

- Rotte preferenziali di instradamento
- Interconessioni privata di domini IPv4

Cifratura del traffico

Compressione dei pacchetti IPv4

Mitigare il problema del header aggiuntivo IPv6 comprimendo il pacchetto IPv4

Oltre il virtuale

FRITZ!Box for WLAN 7390

Porting del modulo kernel su CPE fisico

Sviluppato un custom firmware partendo dal progetto Open Source Freetz.

Grazie !!!

Massimo Bernaschi
m.bernaschi@iac.cnr.it

Massimo Vellucci
m.vellucci@unicampus.it

Luca Vollero
l.vollero@unicampus.it