Tunnel Discovery in IPv6
Methods, results, and security

Lorenzo Colitti

Roma Tre University – RIPE NCC

Outline

- Background on tunnels
- Tunnel discovery methods
- Current state of tunnels in the Internet
- Tunnel discovery in TTM
- Tunnels and security

What is a tunnel?

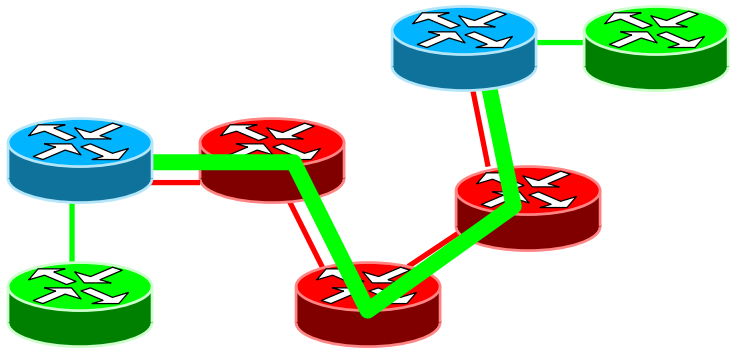
- Point-to-point link between two routers
- IPv6 uses IPv4 as its “link layer”
- IPv6 packets are encapsulated in raw IPv4 packets (Protocol = 41)
- Tunnel MTU ≤ IPv4 MTU - 20

IPv4 Header

Ver	IHL	TOS	Length
Identification		F	Fragment Offset
TTL	**Protocol**		Hdr checksum
Source Address			
Destination Address			

Ver	Class	Flow Label
Length	Next Hdr	Hop Limit
Source Address		
Destination Address		
Data		

IPv6 Packet



IPv4 link

IPv6 link

IPv6-in-IPv4 tunnel

IPv4 router

IPv6 router

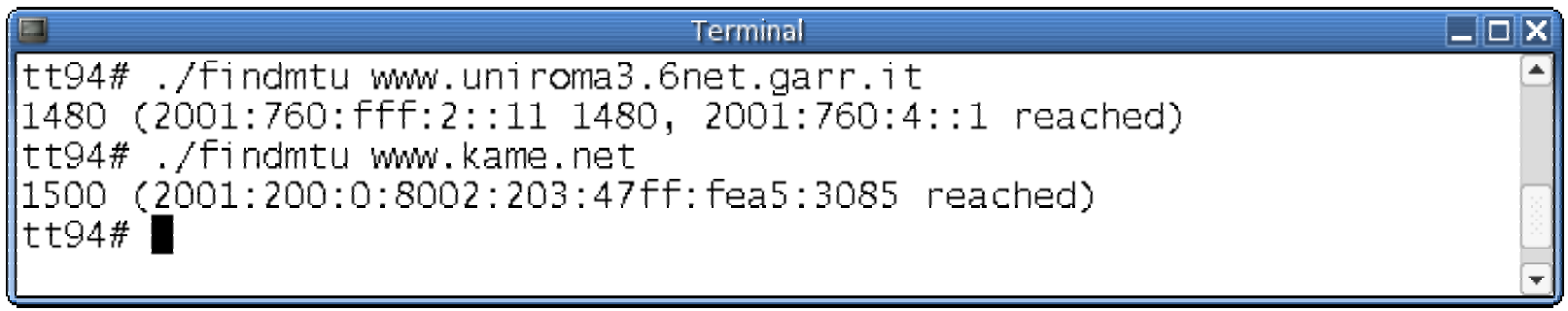
Dual stack router

Problems with tunnels

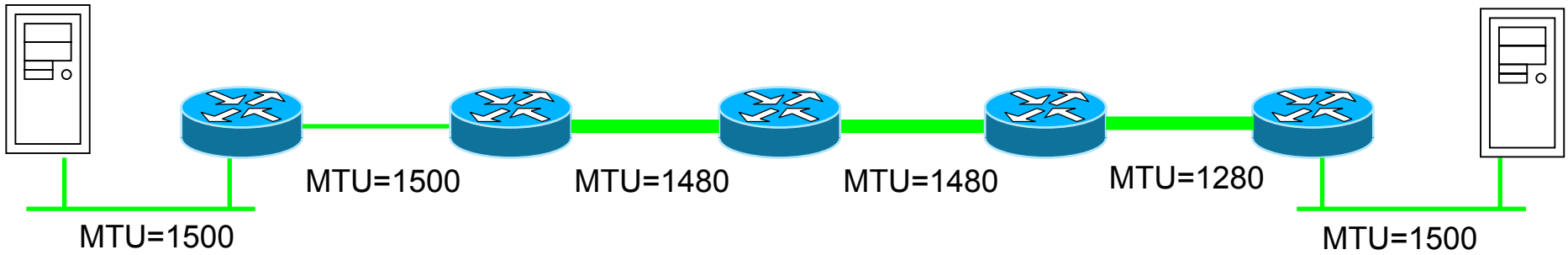
- Low performance
  - Heavy on routers
  - Encourage inefficient routing
- Difficult to troubleshoot
- Pose security problems
- To avoid them we must know they're there
  - Transparent to IPv6, "single-hop"
    - Traceroute doesn't see them
  - What can we do?
  - (What we can't do: DNS)

Finding tunnels

- Path MTU discovery can spot a tunnel
  - MTU of tunnel usually lower than native links
  - Certain MTU values typical of tunnels: 1480, 1280, 1476
- Allows us to find (first) tunnel in a path
  - Often we only want to see if there is a tunnel or not
- Does not distinguish between “short” and “long” tunnels
- Tool: findmtu (linux, freebsd, …)
  - Finds MTU drops on path to user-specified destination



Path MTU discovery and tunnels



1500 byte packet

ICMPv6 Packet Too Big

MTU=1480

1480 byte packet

ICMPv6 Packet Too Big

MTU=1280

1280 byte packet

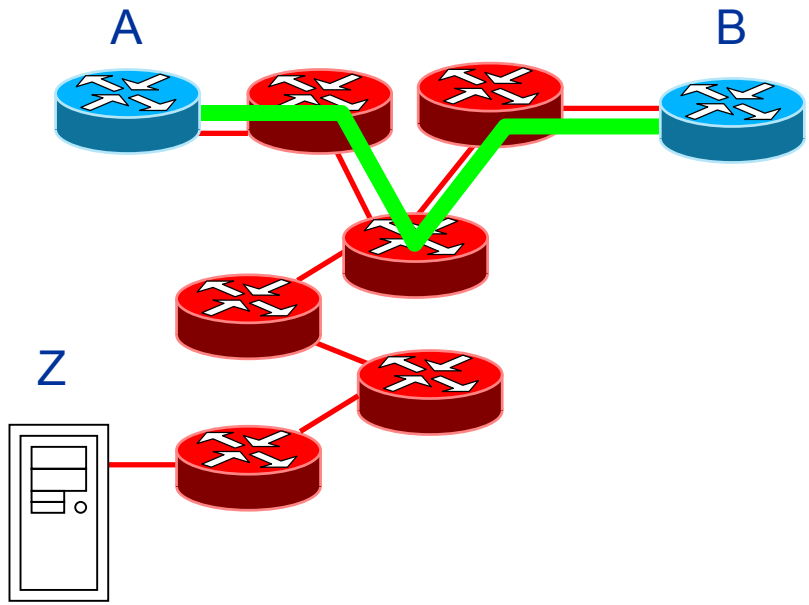
reaches destination

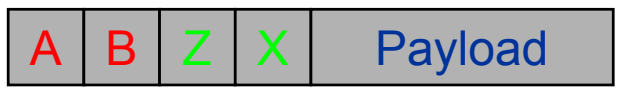
We see a tunnel only if it has a lower MTU than the previous tunnel

Native link Tunnel

Packet injection

- Tunnels provide no authentication mechanism
- If Z knows the IPv4 endpoints of the tunnel, it can source IPv6 packets from B
  - Z spoofs A's IPv4 address and sends an encapsulated packet to B
  - B thinks the packet is from A
  - Since B has a tunnel to A, it decapsulates the IPv6 packet and processes it normally
- As if Z had a direct L2 link to B





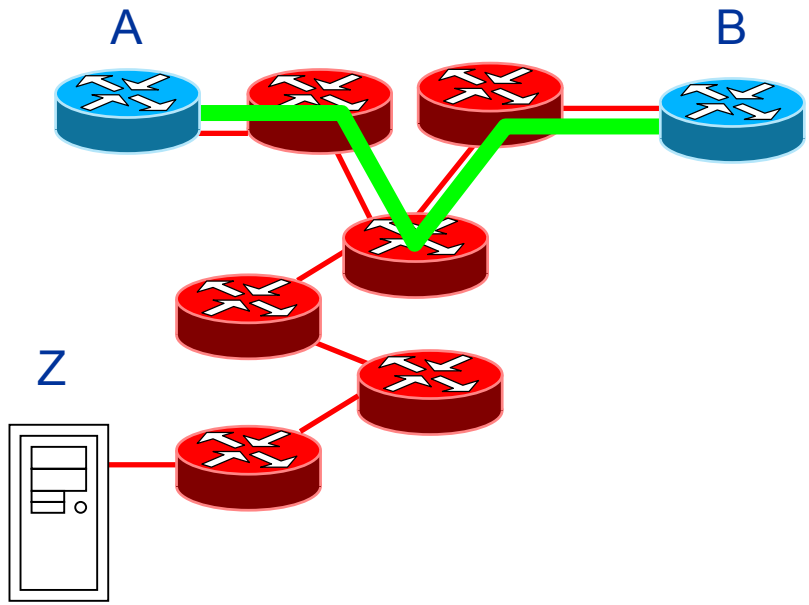
Encapsulated IPv6 packet

A = IPv4 address of A

A = IPv6 address of A

Packet injection for discovery

- Allows Z to:
  - Confirm the presence of a tunnel
    - Inject a packet addressed to itself
  - Discover the IPv6 addresses of the endpoints
    - Send hop limited source routed (or ping-pong) packets
  - Find more tunnels from B
    - IPv6 packet size ≤ MTU of tunnel
    - But IPv4 packets can be fragmented
- A tunnel is a **vantage point** from which Z can explore the rest of the network



A	B	Z	X	Payload

Encapsulated IPv6 packet

A = IPv4 address of A

A = IPv6 address of A

How many tunnels out there?

- We can measure from:
  - Tunnels in the 6bone registry
    - Over 4000 tunnels
      - ~43% nonexistent
      - ~32% down or filtered
    - **~1000 vantage points**
      - Mostly in tunneled networks
  - IPv6-enabled TTM test-boxes
    - **~ 20 vantage points**
      - Mostly in native networks
- Basic idea: find MTU from each vantage point to all prefixes in BGP table

Lorenzo Colitti . RIPE 47, 27 January 2004 . colitti@dia.uniroma3.it – lorenzo@ripe.net

Tunnels seen from the 6bone

- Scan all prefixes from all vantage points, aggregate values
- Results from Aug 2003
- Tunnels dominant
  - Cisco/Linux (1480) and BSD (1280) about the same
  - GRE is much less common
- **Only 8% of paths are native**
  - The 6bone vantage points are biased towards tunnels as they are themselves tunnels.
  - What about native networks?

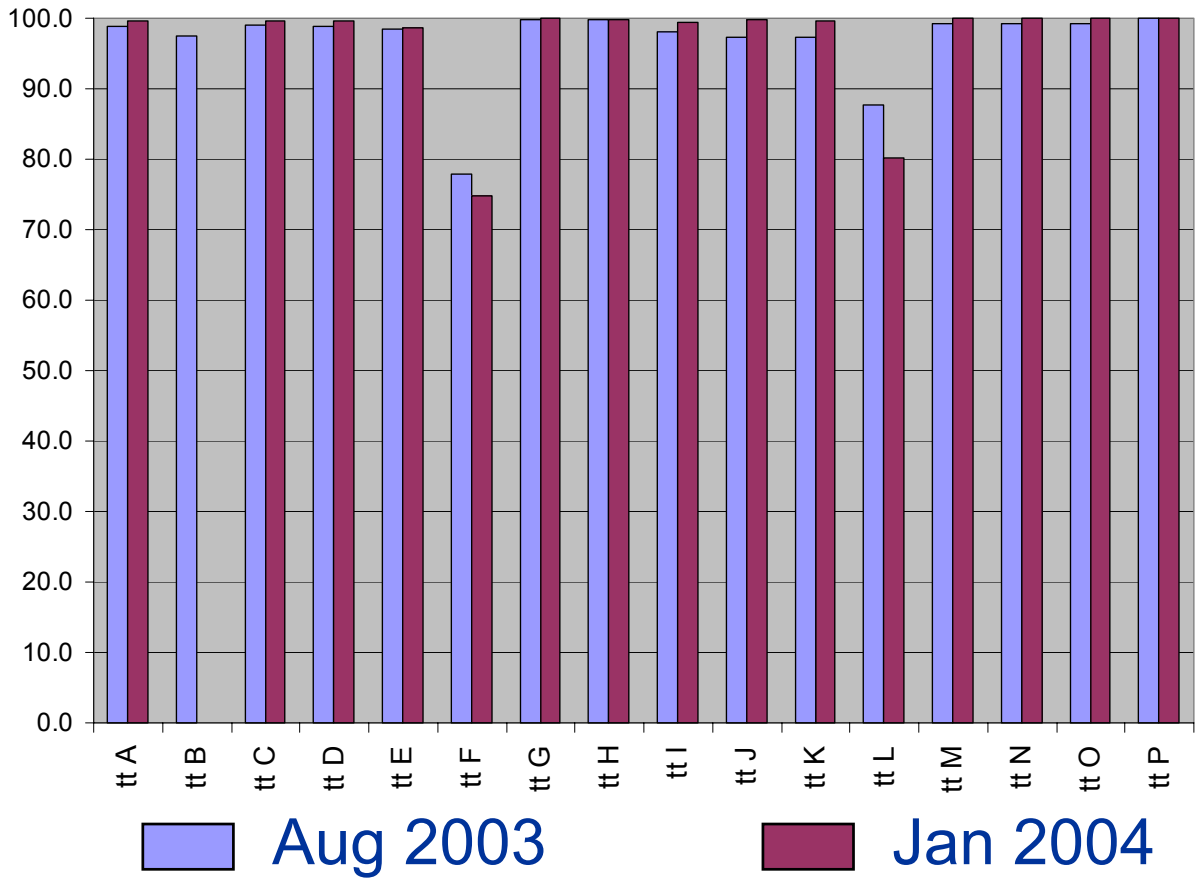
MTU	# paths	%
1480	150946	39.4
1280	138358	36.1
1476	44404	11.6
1500	31525	8.2
1428	13619	3.6
Other	4104	1.1
Total	382956	100.0

How native is “native”?

- Look at IPv6 Internet from TTM boxes, GARR, RIPE NCC networks
- Find MTU to all BGP prefixes
  - Use same BGP table for all vantage points
  - Eliminate errors (unreachable, hop limit expired, …) from routers in same /32
  - Find how many prefixes are definitely tunneled
    - This is a lower bound!
- Measured in Aug 2003 and in Jan 2004
  - Might be interesting to do on a regular basis?

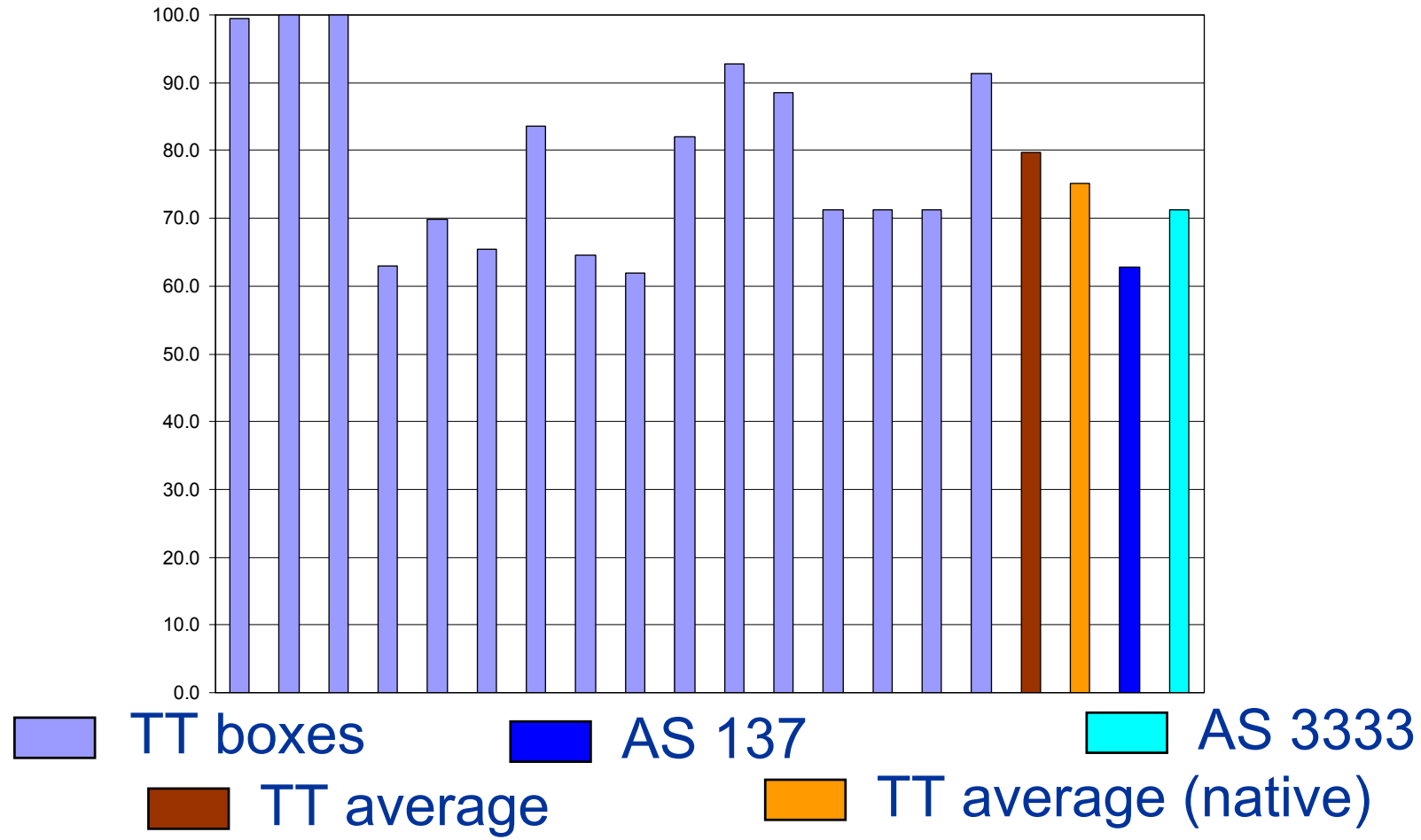
Reachability

Not all prefixes are reachable by all boxes

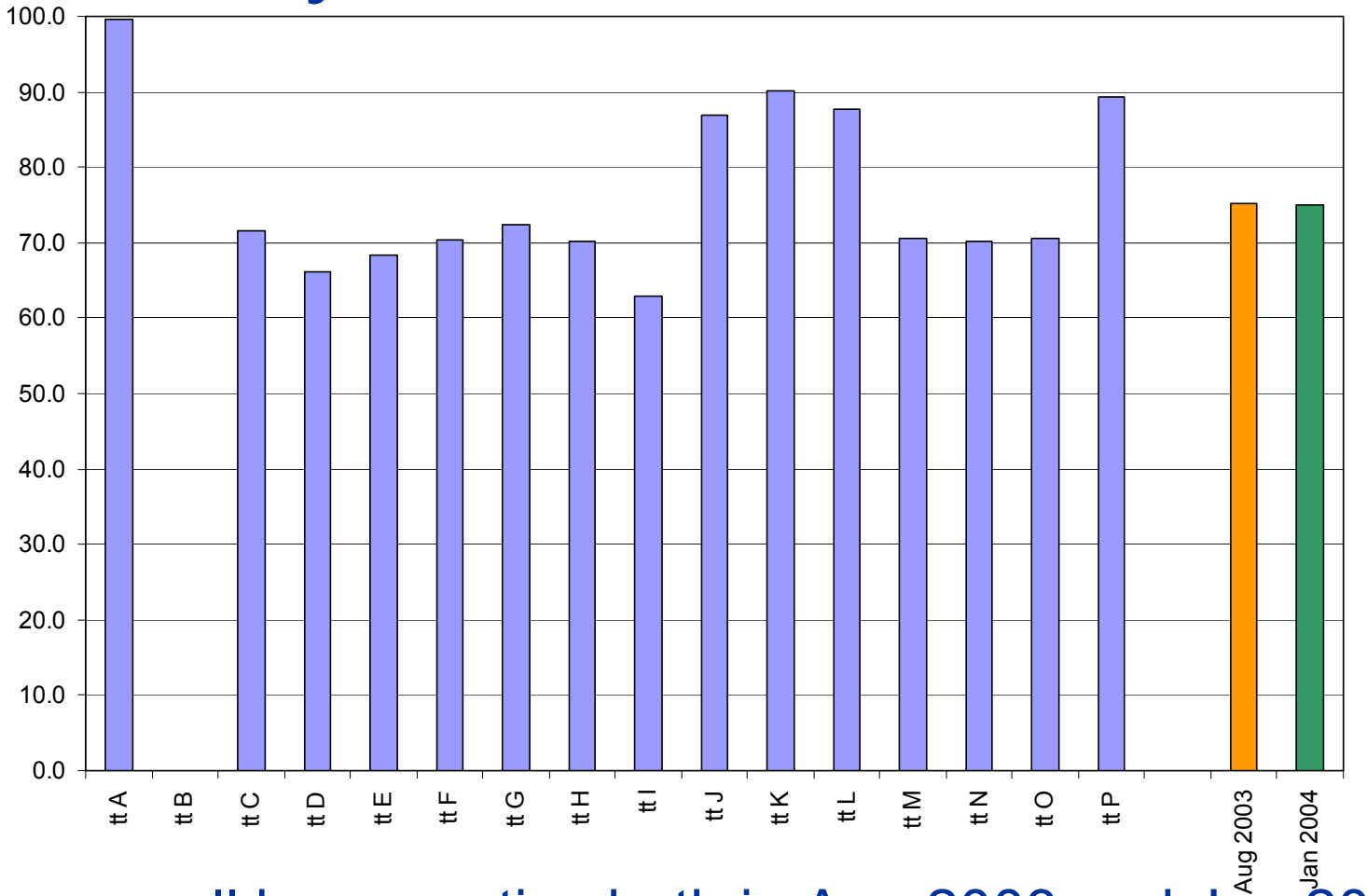


August 2003

- Even the “best” networks are ≥ 62% tunneled



January 2004: is it better now?



- Compare all boxes native both in Aug 2003 and Jan 2004
- Tunnel percentage stable at ~ 75%

Tunnel discovery in TTM

- TTM uses tunnel discovery to better qualify other measurements
  - A high delay might be caused by a tunnel
- Uses path MTU discovery to detect tunnels
  - Find MTU from every testbox to every other testbox
  - Measurements once per hour
- Query data via web interface
  - Can choose set of testboxes, time
  - Full history (for now)
  - Click on MTU value shows traceroute with MTU values

TTM testbox MTU matrix

Source Testbox \ Destination Testbox	tt01	tt103	tt13	tt25	tt35	tt42	tt52	tt55	tt56	tt72	tt73	tt77	tt85	tt86	tt94
tt01		1280	1500	1480	1500	1500	1500	1500	1480	1500	1500	1500	1500	1500	1500
tt103	1280		1280	1280	1280	1280	1280	1280	1280	1280	1280	1280	1280	1280	1280
tt13	1500	1280		1480	1500	1500	1500	1500	1280	1500	1500	1500	1500	1500	1500
tt25	1476	1280	1476		1476	1428	1476	1476	1280	1476	1428	1480	1476	1476	1476
tt35	1480	1280	1500	1476		1500	1500	1500	1480	1500	1500	1462	1476	1476	1480
tt42	1500	1280	1500	1480	1500		1500	1500	1500	1500	1500	1500	1500	1500	1500
tt52	1500	1280	1500	1480	1500	1280		1500	1280	1500	1500	1462	1476	1476	1500
tt55	1500	1280	1500	1480	1500	1280	1500		1280	1500	1500	1462	1476	1476	1500
tt56	1476	1476	1476	1280	1476	1280	1476	1476		1476	1476	1476	1476	1476	1476
tt72	1500	1280	1500	1480	1500	1280	1500	1500	1480		1500	1462	1476	1476	1500
tt73	1500	1280	1500	1480	1500	1500	1500	1500	1280	1500		1500	1500	1500	1500
tt77	1500	1280	1500	1476	1476	1500	1476	1476	1280	1476	1500		1500	1500	1500
tt85	1500	1280	1500	1280	1476	1500	1280	1476	1280	1476	1476	1500		1500	1500
tt86	1500	1280	1500	1476	1476	1500	1476	1476	1280	1476	1476	1500	1500		1500
tt94	1500	1280	1500	1480	1500	1500	1500	1500	1280	1500	1500	1500	1500	1500	

Legend: native | tunnel | no value

Click for traceroute6 vector

Example: two tunnels

from tt85				
Hop	**IPv6 address**	**Hostname**	**AS**	**MTU**
0	2001:620:0:9::85	tt85.ripe.net		
1	2001:620:0:9::1	swiCS4-V27.switch.ch	559	1500
2	2001:620:0:20::6	swi6T1-F0-1.switch.ch	559	1500
3	2001:780::b	tu-viagenie.ipv6.noris.de	12337	1480
4	3ffe:1001:1:f00d::2	*no response*	5609	1280
5	2001:7f8:1::a501:2859:1	sara.ams-ix.ipv6.network.bit.nl	29145417	1280
6	2001:7b8::290:6900:1cc6:d800	jun1.kelvin.ipv6.network.bit.nl	12859	1280
7	2001:7b8:3:32:201:2ff:feb0:c737	tt52.ripe.net	12859	1280

Click for whois info

First tunnel

Second tunnel

Example: symmetric route

from tt103

Hop	IPv6 address	Hostname	AS	MTU
0	2001:240:0:400::2497:101	tt103.ripe.net		
1	2001:240:0:400::1	*no response*	2497	1500
2	2001:240::204	*no response*	2497	1280
3	2001:240:100::2	*no response*	2497	1280
4	2001:240:100:fffd::21	*no response*	2497	1280
5	2001:504:1::a500:3257:1	ge-1-0-0.nyc10.ip6.tiscali.net	*N/A*	1280
6	2001:668:0:2::330	so-3-0-0.nyc30.ip6.tiscali.net	3257	1280
7	2001:668:0:2::1a1	so-1-0-0.nyc31.ip6.tiscali.net	3257	1280
8	2001:668:0:2::1e0	so-4-0-0.lon12.ip6.tiscali.net	3257	1280
9	2001:668:0:2::30	so-6-0-0.ams10.ip6.tiscali.net	3257	1280
10	2001:7f8:1::a501:2859:2	telecity.ams-ix.ipv6.network.bit.nl	2914/5417	1280
11	2001:7b8::290:6900:1cc6:d800	jun1.kelvin.ipv6.network.bit.nl	12859	1280
12	2001:7b8:3:32:201:2ff:feb0:c737	tt52.ripe.net	12859	1280

to tt52

to tt103

Hop	IPv6 address	Hostname	AS	MTU
12	2001:240:0:400::2497:101	*no response*	2497	1280
11	2001:240:0:400::1	*no response*	2497	1280
10	2001:240:100::204	otm6-gate0.IIJ.Net	2497	1280
9	2001:240:100:fffd::ff	*no response*	2497	1280
8	2001:504:1::a500:2497:1	*no response*	*N/A*	1500
7	2001:668:0:2::331	so-3-0-0.nyc10.ip6.tiscali.net	3257	1500
6	2001:668:0:2::1a0	so-1-0-0.nyc30.ip6.tiscali.net	3257	1500
5	2001:668:0:2::1e1	so-7-0-0.nyc31.ip6.tiscali.net	3257	1500
4	2001:668:0:2::31	so-2-0-0.lon12.ip6.tiscali.net	3257	1500
3	2001:7f8:1::a500:3257:1	ams-ix.ip6.tiscali.net	2914/5417	1500
2	2001:7b8::205:8500:120:7c1f	jun1.sara.ipv6.network.bit.nl	12859	1500
1	2001:7b8:3:32::1	*no response*	12859	1500
0	2001:7b8:3:32:201:2ff:feb0:c737	tt52.ripe.net		

from tt52

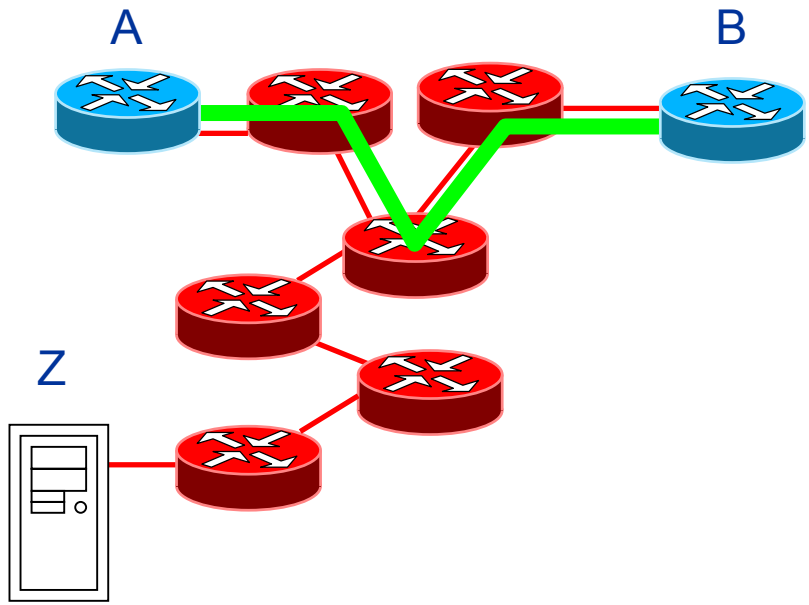
Legend: native tunnel/unknown route invalid

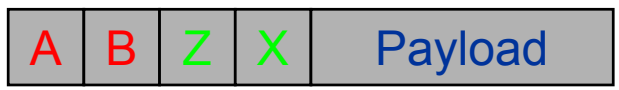
Native

By comparing traces in opposite directions, we can see both the start and the end of a tunnel

Tunnels and security

- Packet injection is bad for security
- Z can source arbitrary IPv6 packets from B
  - More effective than IPv6 spoofing
    - Bypasses IPv6 filtering
    - Z can use its real IPv6 source address and receive replies
  - More effective than source routing
    - When packet arrives at B, Hop Limit is untouched
      - ND packets can be spoofed
    - Can't be turned off on routers





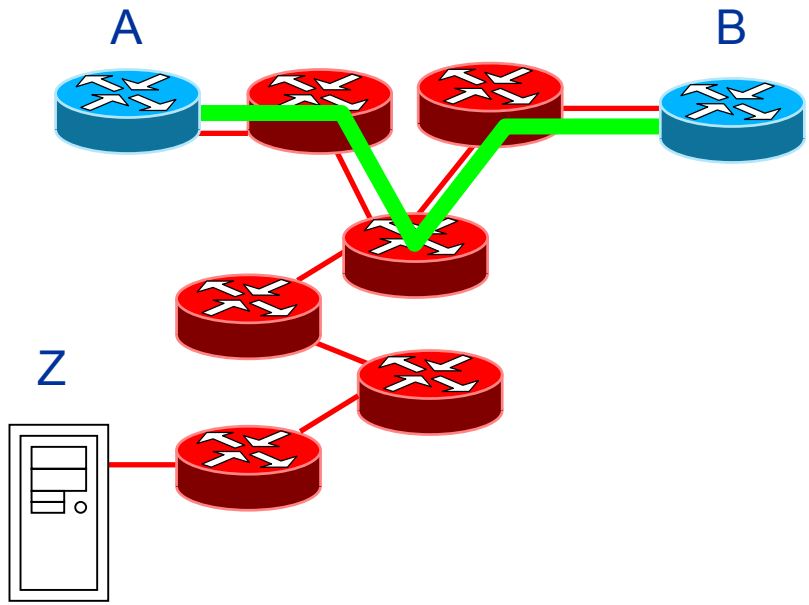
Encapsulated IPv6 packet

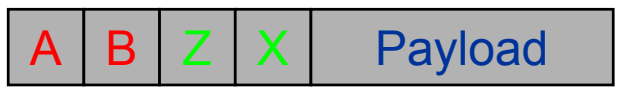
A = IPv4 address of A

A = IPv6 address of A

Tunnels and security (2)

- Packet injection allows Z to:
  - Bypass firewalls / ingress filters
  - Spoof ND packets
    - Redirect, L2 address spoofing, …
    - Not tested, but possibly dangerous
  - …
- What can be done?
  - IPv4 filtering helps
    - But not for interdomain tunnels
  - Don't trust tunnels and keep them at the edge
  - Use GRE / keyed GRE tunnels





Encapsulated IPv6 packet

A = IPv4 address of A

A = IPv6 address of A

References:

- Tunnel discovery @Roma Tre:
  http://www.dia.uniroma3.it/~compunet/tunneldiscovery/
- Tunnel discovery in TTM:
  http://www.ripe.net/ttm/Plots/pmtu/tunneldiscovery.cgi
- RIS IPv6 update counts
  http://www.ris.ripe.net/ipv6-updates/

Questions