# Network Fingerprinting: TTL-based Router Signatures

Yves Vanaubel[1], Jean-Jacques Pansiot[2], Pascal Merindol[2] and Benoit Donnet[1]

[1]ULg (Belgium), [2]UDS (France)

March 12, 2014

# Summary

- Motivations
- TTL-based router signatures
- Measurement campaign
- Signatures distribution and consistency
- Use cases
- Conclusions

# Motivations

### Network fingerprinting

Action of grouping network devices into (disjoint) classes.
*Equivalent to* ***nmap*** *but for routers instead of host OSes.*

### Signature

Set of information collected thanks to the fingerprinting.

- Understanding the characteristics of the Internet:
  - hardware distribution (CISCO, Juniper, etc.)
  - routing operating systems distribution (IOS, OS-XR, JUNOS, JUNOSE, etc.)
  - abnormal behaviors
  - vulnerabilities
  - ...
- Topology discovery
- ...

# Time To Live (TTL)

- Field in the IP header (avoid routing loops)
- Maximum number of hops for an IP packet

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

# TLL - Initial Value

- Should be initialized to 64 (RFC 1700)
- However, in practice, the initial value of the TTL (iTTL) may depend on:
  - the hardware (CISCO, Juniper, ...)
  - the operating system
  - the protocol used for the message (ICMP, UDP, ...)
  - the type of the message (information packets versus errors)

**Idea:**

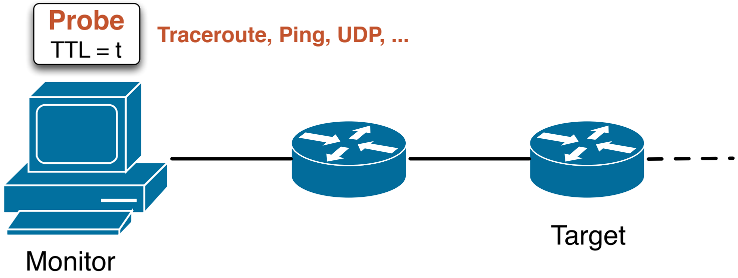
Solicit routers with several probes in order to receive $n$ different types of (ICMP) replies, infer their initial TTL value and derive a signature of the type

$$< iTTL_1, iTTL_2, iTTL_3, ..., iTTL_n >$$

# ICMP Messages

- We consider three types of ICMP messages:
  1. **Time-exceeded** messages (obtained with `traceroute`)
  2. **Echo-reply messages** (obtained with `ping`)
  3. **Destination-unreachable messages** (obtained with `UDP probes` sent to a very high destination port)
- Initial values of TTLs used by nodes: 32, 64, 128, 255

# Initial TTL Value: Inference




Initial TTL inference:

Smallest integer in {32, 64, 128, 255} larger than the received value

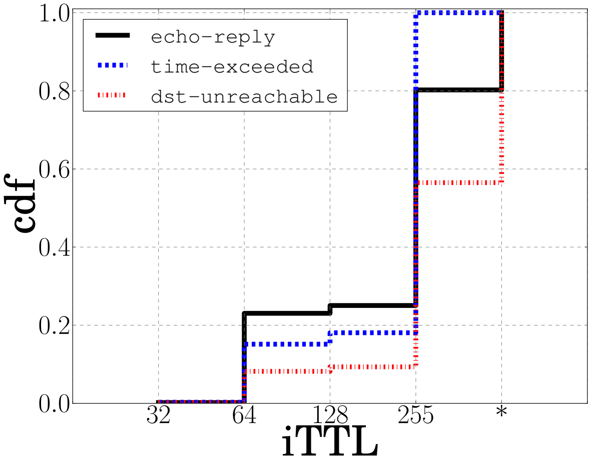
In the example:

- 63 in the TTL field of the ICMP response
- 64 is the corresponding inferred iTTL

# Measurement Campaign

- Measurement campaign on the PlanetLab platform
- 1M of destinations from CAIDA data
- 200 vantage points (VP), i.e. 5000 destinations/VP
- Each IP on a trace pinged 6 times
- Scamper with paris-traceroute
- About 8h of probing per VP
- About 3 days of campaign due to the PlanetLab instabilities
- 335,646 unique IPs collected with 13,437,896 `traceroute` replies
- Marginal probing cost overhead (14,803,614 `ping` replies)

# Initial TTL Value: Distribution



# Generic Router Signature Construction Algorithm

- For each destination:
  1. Send `traceroute` probes to detect the nodes on the path
  2. Foreach received `ICMP time-exceeded` message:
     - Check if the corresponding node was not already probed
     - Infer the first iTTL of the signature
     - Send other types of probes (`Ping`, `UDP`, ...)
     - Infer the other iTTLs based on the responses

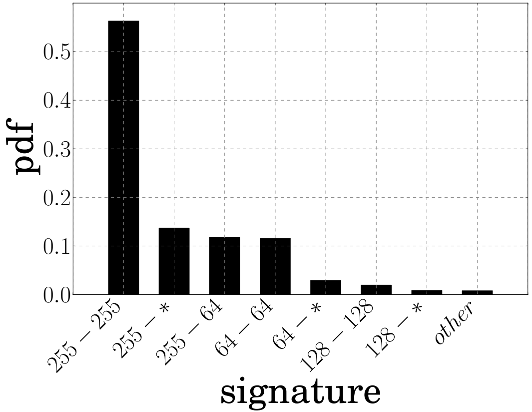
# TTL-based Router Signatures

- Consists in a n-tuple of initial TTLs
- As a first try, $n = 2$ (marginal gain with UDP probes):

**<Time-exceeded, Echo-reply>**

- Signature diversity: in theory up to $4 \times 5^{n-1}$, $n$: # probes
- The symbol $*$ means an absence of iTTL (no answer to the corresponding probe). The signature is **incomplete**
- Examples : <255-255>, <255-$*$>, <255-128>, ...

# Signatures Distribution




| | |
|---|---|
| Cisco | <255-255> |
| Juniper (Junos) | <255-64> |
| Juniper (JunosE) | <128-128> |
| Brocade, Alcatel and Linux boxes | <64-64> |

Table : Some router manufacturer mapping examples

# Signatures Consistency

**Assumption:**

The signature associated to a given IP address is unique

- Considering only IP addresses probed by at least two VPs...
- ... a signature may be (for a given IP address):
  - **Coherent**: signatures always the same (in 95.92%)
  - **Weakly incoherent**: signatures sometimes complete, but also sometimes incomplete (in 4.94%) (e.g. <255-255> and <255-*>)
  - **Incoherent**: complete signatures but different (in 0.14%)

# Signatures Consistency

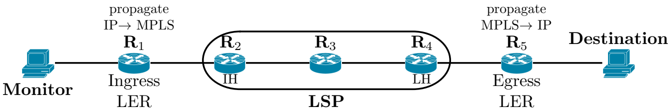
- In the vast majority, coherent signatures.
- Causes of the (rare) inconsistency:
  - our initial TTL inference?
  - anycast?
  - middleboxes?
- Possibility to complete weakly incoherent signatures (e.g. $<255\text{-}*> \Rightarrow <255\text{-}255>$)

$\Rightarrow$ Our assumption is correct:

**The signature associated to a given IP address is unique**

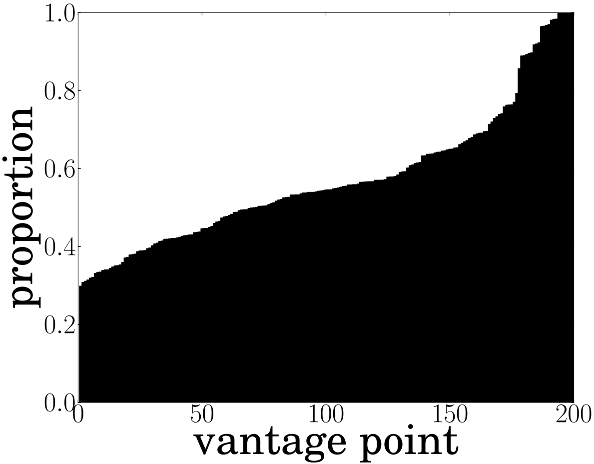
# MPLS use case

- Donnet et al.: "Revealing MPLS tunnels obscured from traceroute" ACM SIGCOMM CCR, 2012.
- Measurement-based classification of MPLS tunnels (traceroute)
  - TTL-propagate × RFC4950:




| Explicit | Implicit | Opaque | Invisible |
|---|---|---|---|
| 1. $R_1$ | 1. $R_1$ | 1. $R_1$ | 1. $R_1$ |
| 2. $R_2$ - MPLS | 2. $R_2$ | 2. $R_4$ - MPLS | 2. $R_4$ |
| 3. $R_3$ - MPLS | 3. $R_3$ | 3. $R_5$ | 3. $R_5$ |
| 4. $R_4$ - MPLS | 4. $R_4$ | 4. Destination | 4. Destination |
| 5. $R_5$ | 5. $R_5$ | | |
| 6. Destination | 6. Destination | | |

# Proportion of MPLS tunnels

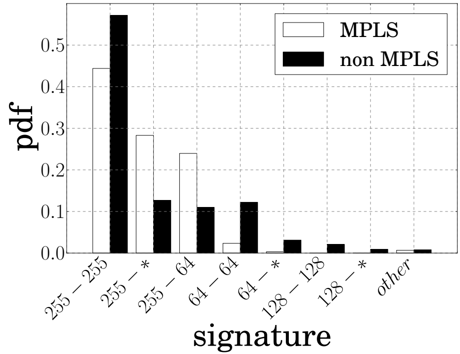



- From about half of the VPs, at least half of the paths reveal MPLS tunnels

# Proportion of MPLS tunnels

- Proportion of IP addresses in
  - explicit tunnels: 14.23%
  - implicit tunnels: 25.51%
  - opaque tunnels: 0.33%
  - all MPLS tunnels: 30.37%
- Some addresses belongs to different types of tunnels
- MPLS seems well deployed in the Internet today

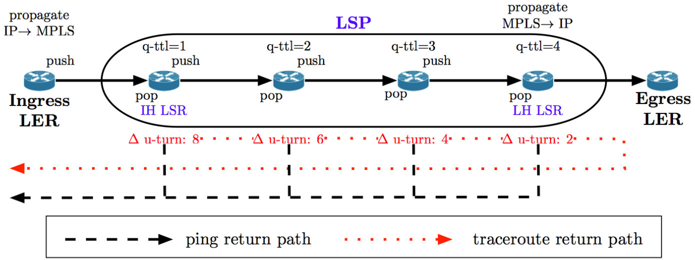
# MPLS Repartition: Global TTL-overview




- The increase of Juniper routers seems significant
- Decrease of signature <64-64>
- Decrease of signature <255-255> while <255-∗> and <255-64> increase their share

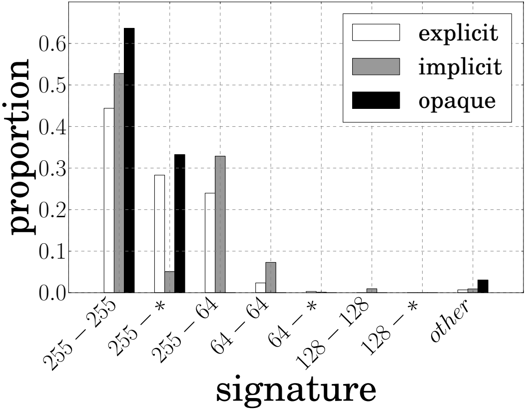
# MPLS: Implicit tunnels discovery

- Inference methods: `q-TTL` and `Uturn`



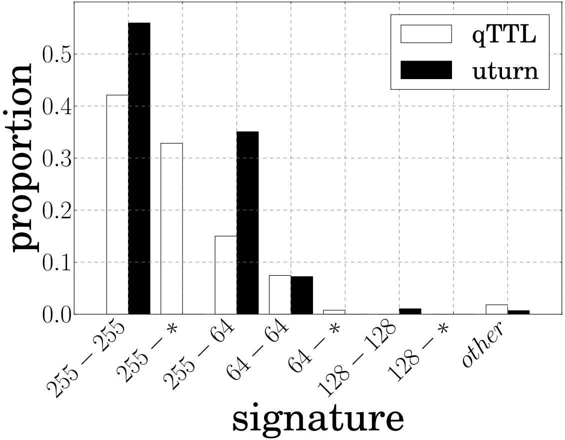
# MPLS Repartition: Refined view

- Signatures distribution in the different tunnel classes:




- Opaque tunnels : only one signature : <255-255> (and <255-*> but may be completed)

# MPLS implicit tunnels: TTL-overview




- Only slight differences due to technique definition
  - Our `Uturn` heuristic does not show abnormal behavior

# Use Cases

- Checking if a set of network devices is heterogeneous
- Understanding the architecture of an AS
- (In)validation of measurement hypotheses (e.g. MPLS tunnels discovery)
- Helping alias resolution (clustering approach)
- ...

# Conclusion

- Each IP (router?) has a unique TTL-based fingerprint
- The distribution of signatures is already valuable with 2 iTTLs
- Work still in progress: refine the signatures distribution
- Help alias resolution and so IP network mapping
- Help to improve any active probing methods and analysis such as MPLS discovery and quantification

Thank you!