Address family matters in end-to-end latency

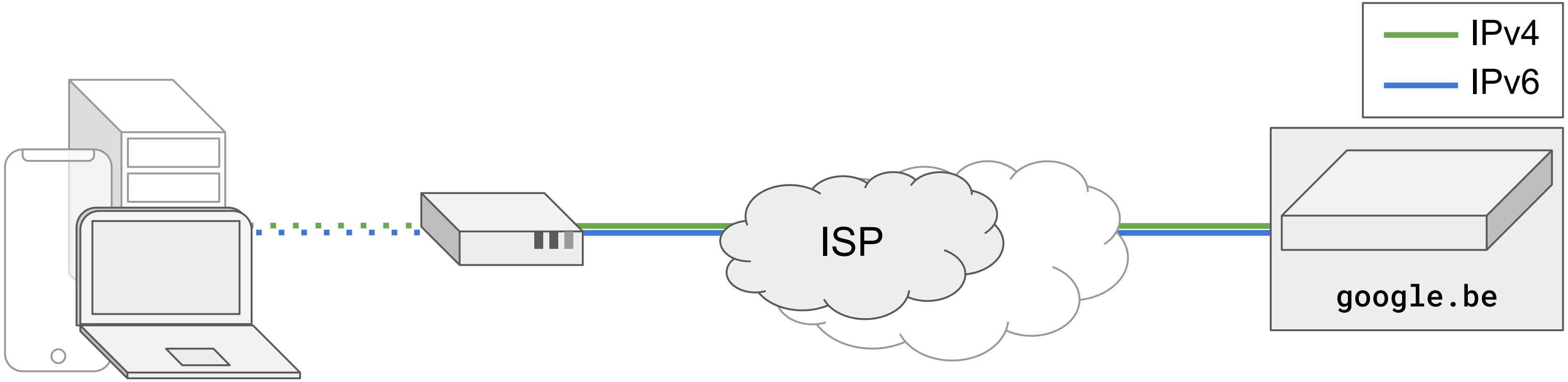
Maxime Piraux, Olivier Bonaventure

UCLouvain

ICTEAM Research Institute

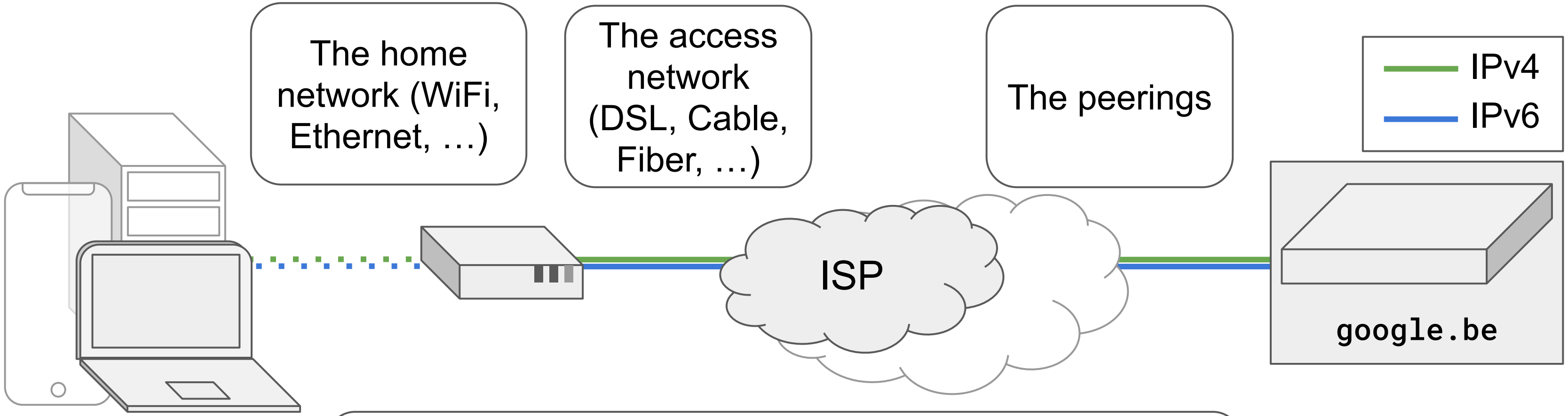
A naive question

“I subscribed to a DSL line in Belgium. I want to **improve latency** towards **google.be**. Should I switch to **fiber**?”



A naive question

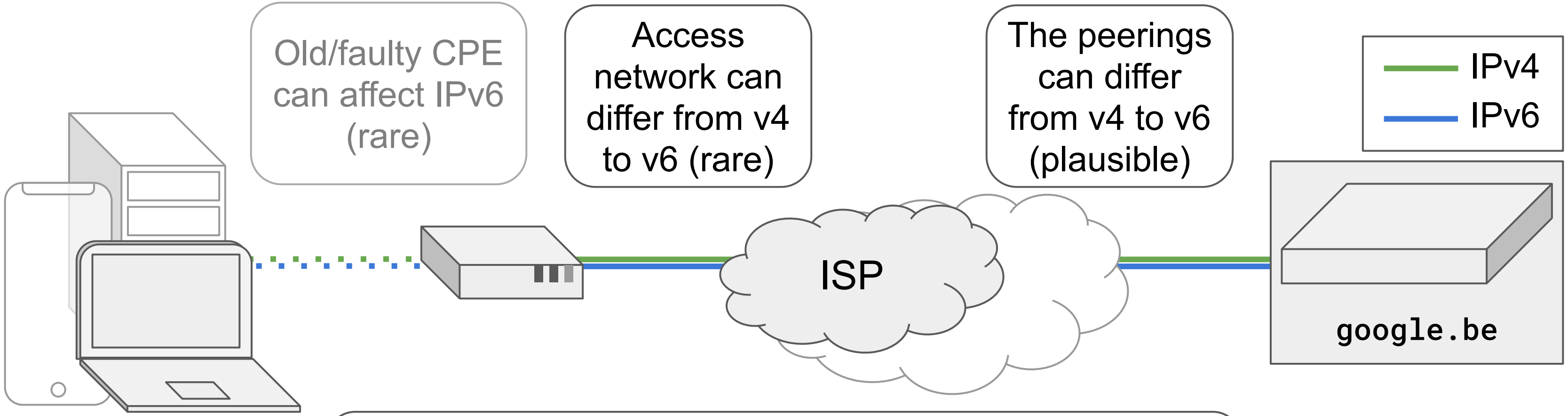
“I subscribed to a DSL line in Belgium. I want to **improve latency** towards **google.be**. Should I switch to **fiber**?”



What does impact latency?

A naive question

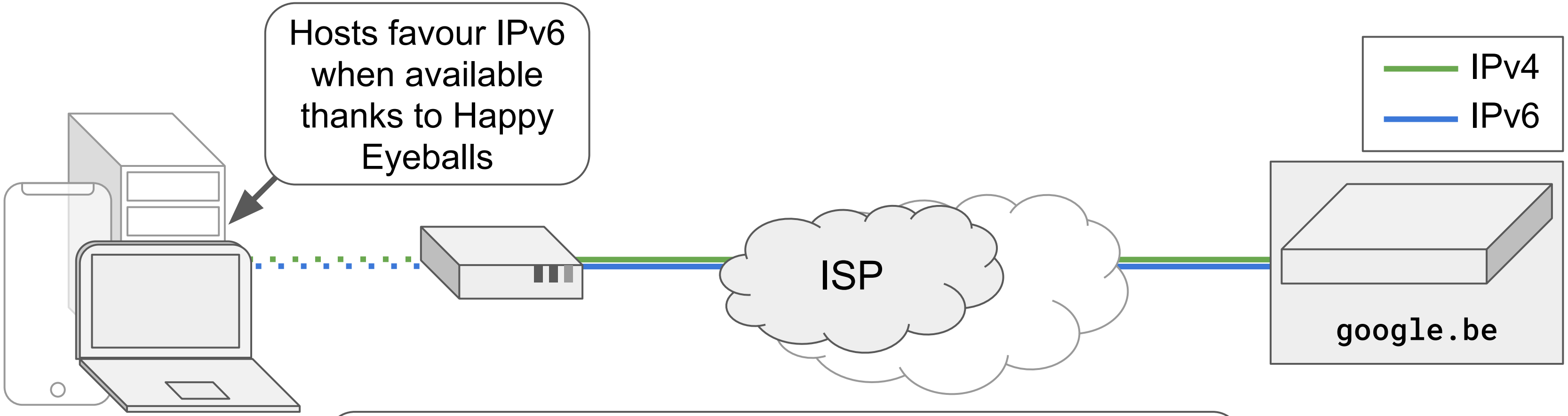
"I subscribed to a DSL line in Belgium. I want to **improve latency** towards **google.be**. Should I switch to **fiber**?"



Does the address family matter?

A naive question

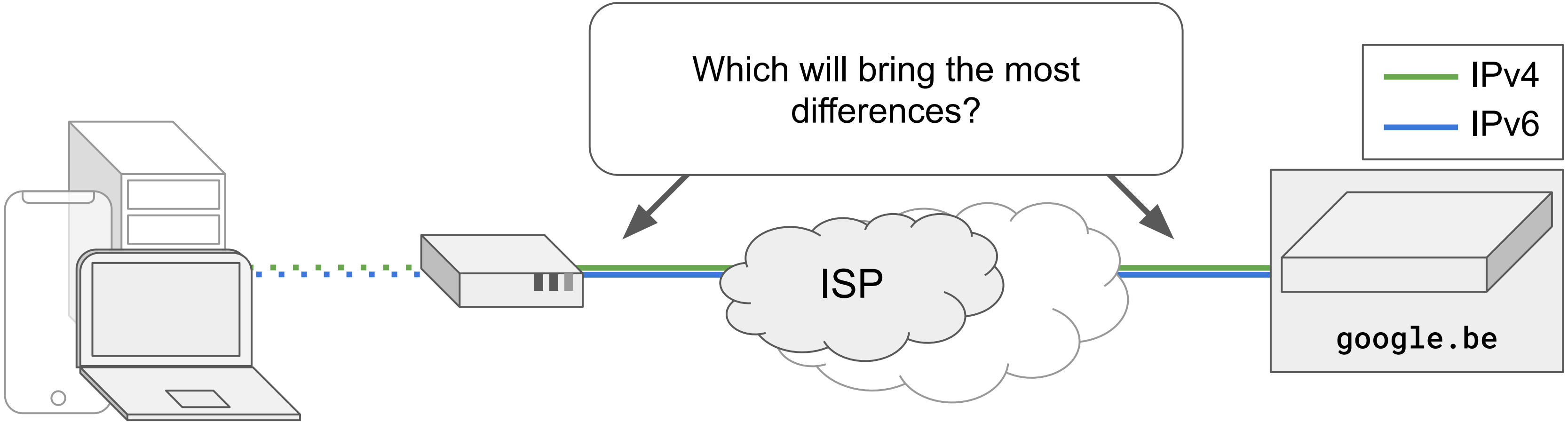
“I subscribed to a DSL line in Belgium. I want to **improve latency** towards **google.be**. Should I switch to **fiber** or **use IPv4**?”



How is the address family chosen?

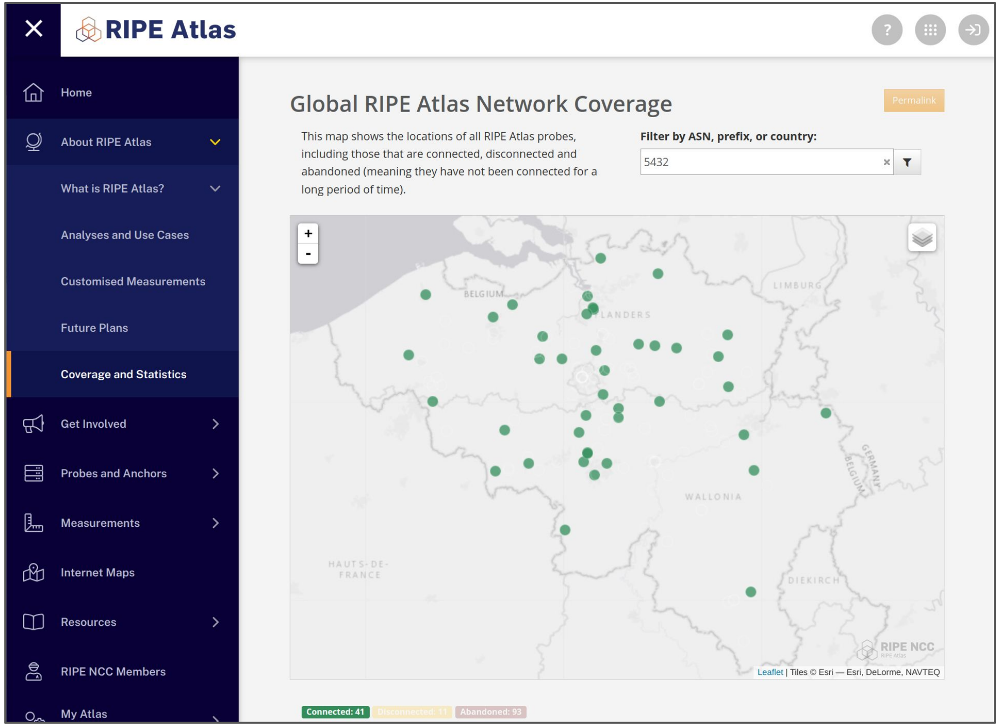
A naive question

"I subscribed to a DSL line in Belgium. I want to **improve latency** towards **google.be**. Should I switch to **fiber** or **use IPv4**?"



Measuring latency differences

- RIPE Atlas is a worldwide measurement platform.
- RIPE probes are operated by individuals and can perform network tests on demand.
- 41 probes are connected to AS5432.



RIPE Ping test

- Let’s request pings towards `google.be` using IPv6 and IPv4.

IPv6

Probe	ASN (IPv4)	ASN (IPv6)			Time (UTC)	RTT
62020	5432	5432			2023-10-10 07:25	11.249
11639	5432	5432			2023-10-10 07:25	13.373
1005964	5432	5432			2023-10-10 07:25	13.552
25348	5432	5432			2023-10-10 07:25	13.746
1006234	5432	5432			2023-10-10 07:25	14.613
17394	5432	5432			2023-10-10 07:25	18.578
50750	5432	5432			2023-10-10 07:25	19.363
15019	5432	5432			2023-10-10 07:30	19.468
24320	5432	5432			2023-10-10 07:25	19.507
55565	5432	5432			2023-10-10 07:25	19.883

RIPE Ping test

- Let’s request pings towards `google.be` using IPv6 and IPv4.
- Fiber improves the latency.

IPv6

Probe	ASN (IPv4)	ASN (IPv6)			Time (UTC)	RTT
62020	5432	[illegible]			2023-10-10 07:25	11.249
11639	5432				0-10 07:25	13.373
1005964	5432				0-10 07:25	13.552
25348	5432				0-10 07:25	13.746
1006234	5432				0-10 07:25	14.613
17394	5432				0-10 07:25	18.578
50750	5432				0-10 07:25	19.363
15019	5432				0-10 07:30	19.468
24320	5432				0-10 07:25	19.507
55565	5432	5432			2023-10-10 07:25	19.883

These seem to be connected to fiber

These seem to be connected to DSL

RIPE Ping test

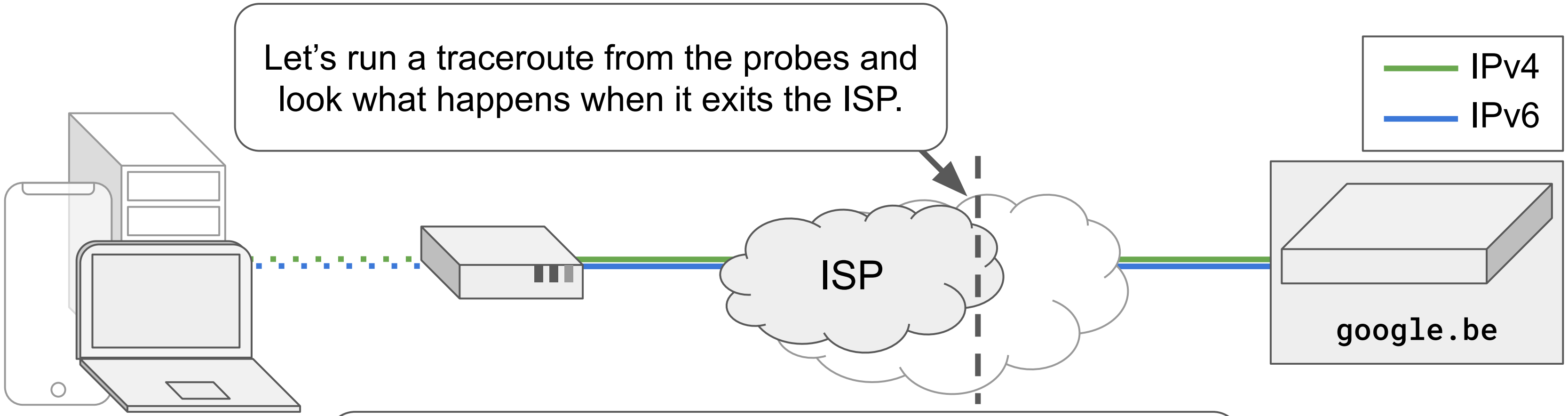
- Let's request pings towards `google.be` using IPv6 and IPv4.
- Fiber improves the latency.
- Changing the address family does too!

IPv4

Probe	ASN (IPv4)	ASN (IPv6)		RTT	RTT
62020	5432	5432		11.249	6.629
11639	5432	5432		13.373	7.527
1005964	5432	5432		13.552	8.372
25348	5432	5432		13.746	8.409
1006234	5432	5432		14.613	8.568
15019	5432	5432		18.578	12.694
17394	5432	5432		19.363	13.060
50750	5432	5432		19.468	13.099
24320	5432	5432		19.507	13.135
24519	5432	5432		19.883	13.238

A naive question

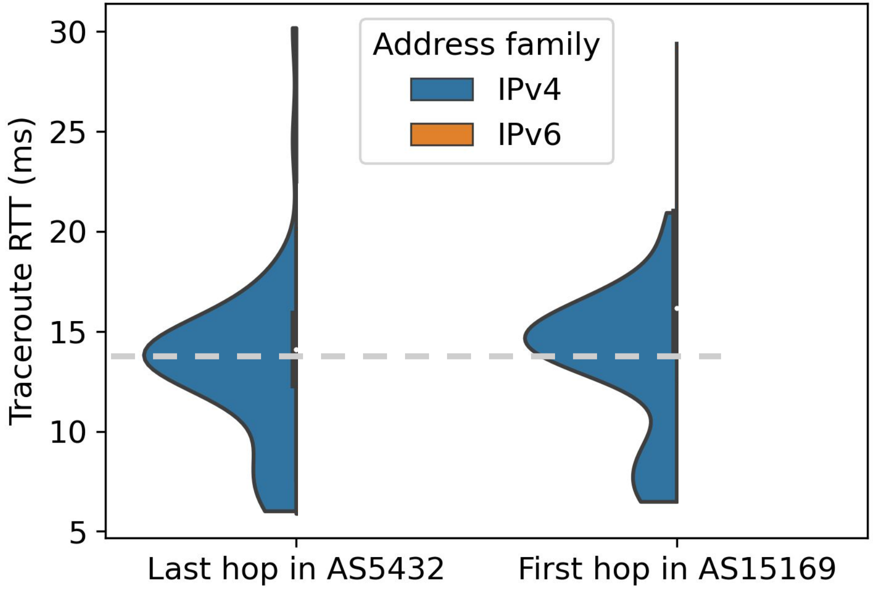
"I subscribed to a DSL line in Belgium. I want to **improve latency** towards **google.be**. Should I switch to **fiber** or **use IPv4**?"



Where does the address family difference come from?

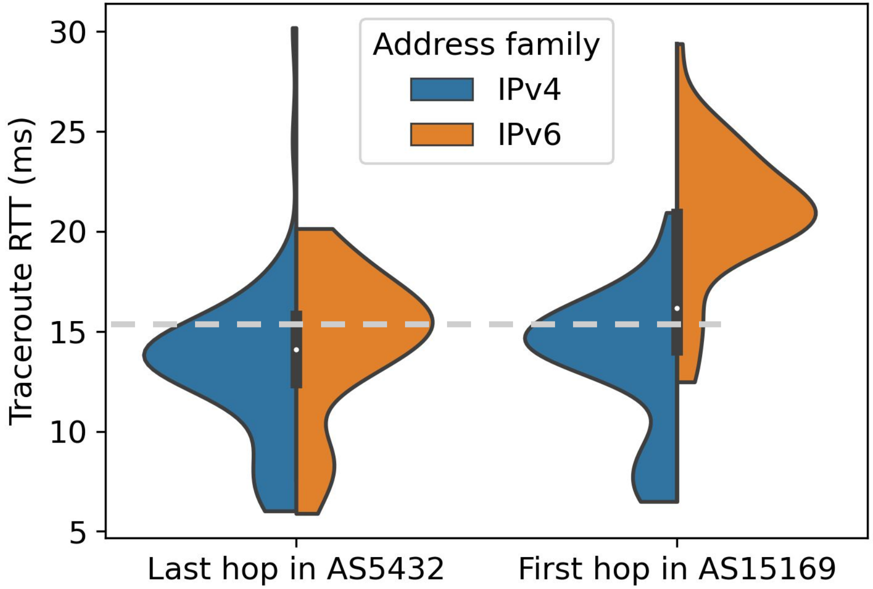
RIPE Traceroute test

- Running a traceroute reveals a difference when exiting the ISP and entering the Google AS.



RIPE Traceroute test

- Running a traceroute reveals a difference when exiting the ISP and entering the Google AS.
- Their peering is the major cause of address family latency differences.

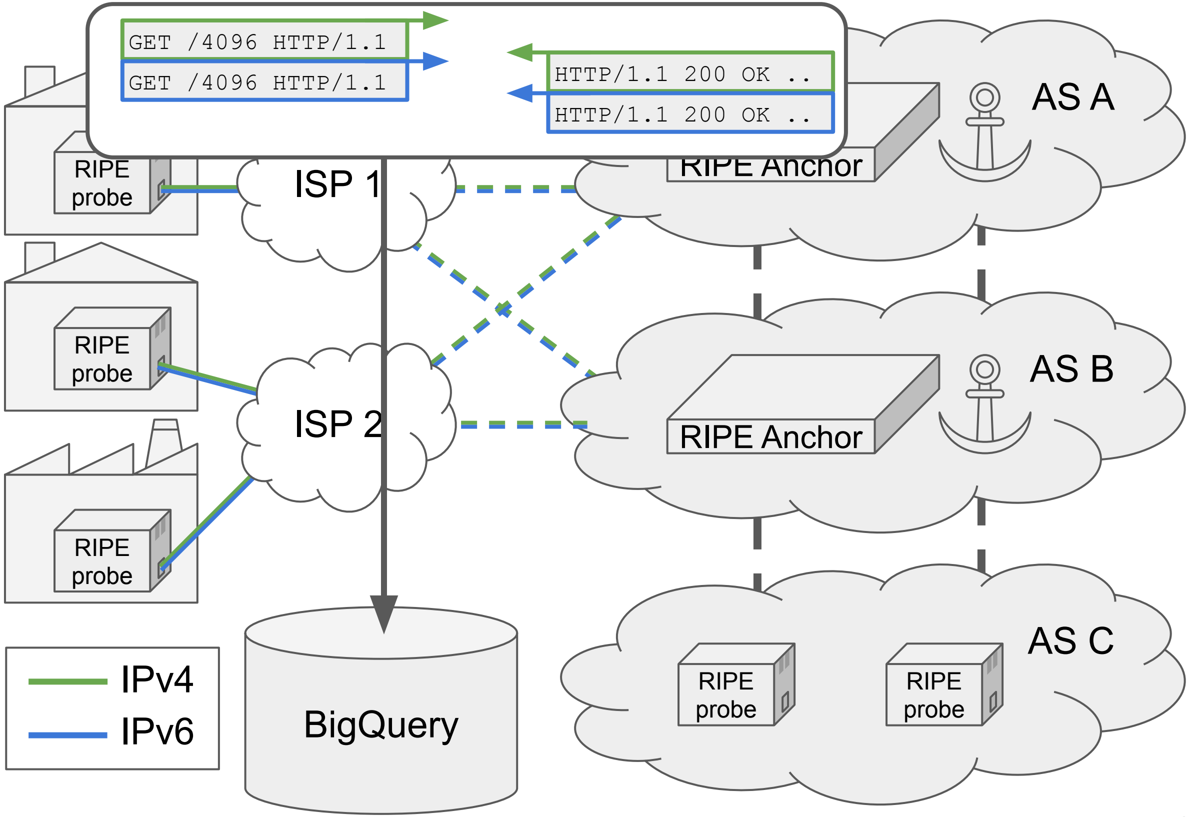


Taking a step back – Let’s ask smart questions now

- Is there an address family that has globally a lower latency?
- How are these differences spread?
  - Are they common to the source?
  - Do they depend on the destination?
- Are these differences stable over time?

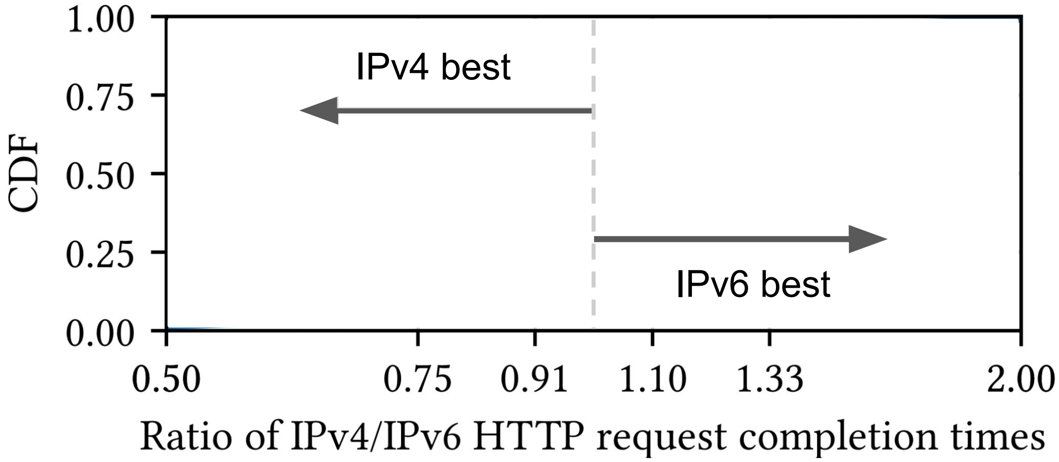
RIPE Atlas

- RIPE has about 12,000 probes and 780 anchors spread in 3600 ASes.
- Probes regularly perform automated network tests towards anchors.
- The tests results are collected and published through BigQuery



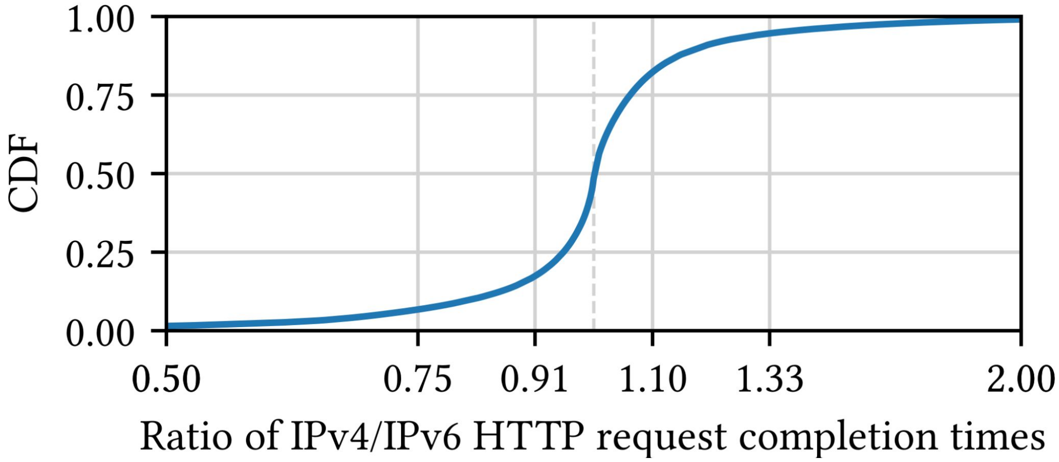
IPv4 and IPv6 end-to-end latency

- Probes performed 156 millions of HTTP v4/v6 tests between the 1st and 8th of June 2023.



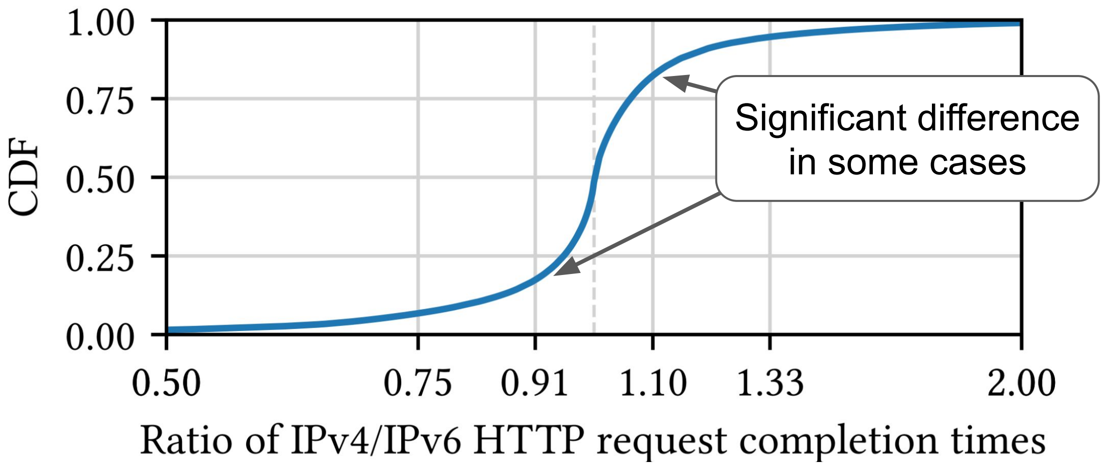
IPv4 and IPv6 end-to-end latency

- Probes performed 156 millions of HTTP v4/v6 tests between the 1st and 8th of June 2023.
- **No address family has a global advantage in terms of latency.**



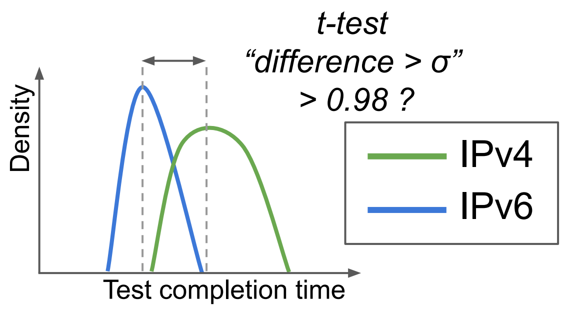
IPv4 and IPv6 end-to-end latency

- Probes performed 156 millions of HTTP v4/v6 tests between the 1st and 8th of June 2023.
- **No address family has a global advantage in terms of latency.**



IPv4 and IPv6 end-to-end latency

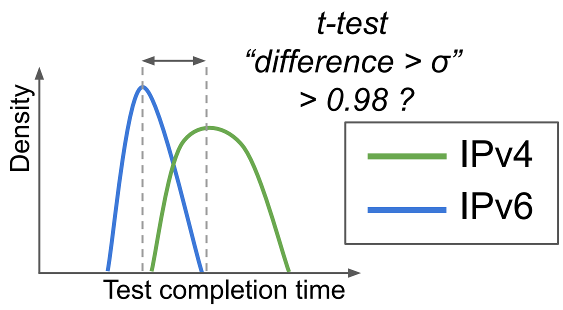
- There are 400 000 probe-anchor pairs with at least 300 HTTP tests.
- For each pair, a statistical test determine whether a difference larger than the standard deviation exists.
- Results are spread rather homogeneously.



IPv4 is best	IPv6 is best	None strongly better
113092 (28.4%)	129070 (32.4%)	156212 (39.2%)

IPv4 and IPv6 end-to-end latency

- There are 400 000 probe-anchor pairs with at least 300 HTTP tests.
- For each pair, a statistical test determine whether a difference larger than the standard deviation exists.
- Results are spread rather homogeneously.

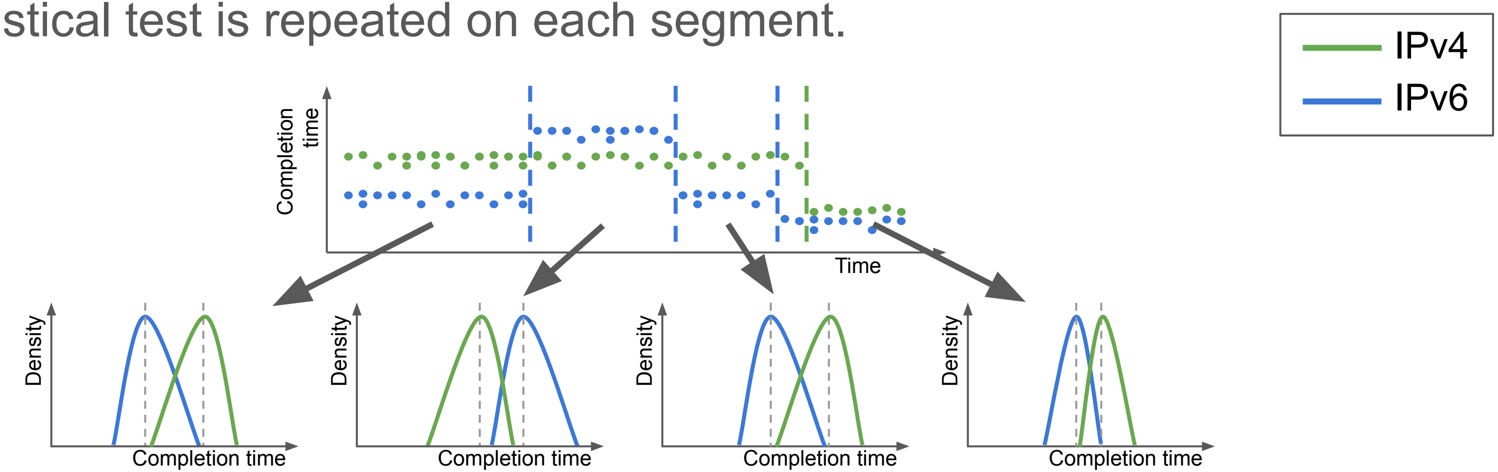


IPv4 is best	IPv6 is best	None strongly better
113092 (28.4%)	129070 (32.4%)	156212 (39.2%)

Are these results stable over time ?
Are these results consistent per probe ?

IPv4 and IPv6 end-to-end latency

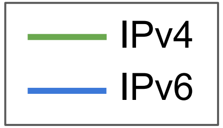
- We used a change-point detection algorithm to split probe-anchor timeries.
- The statistical test is repeated on each segment.

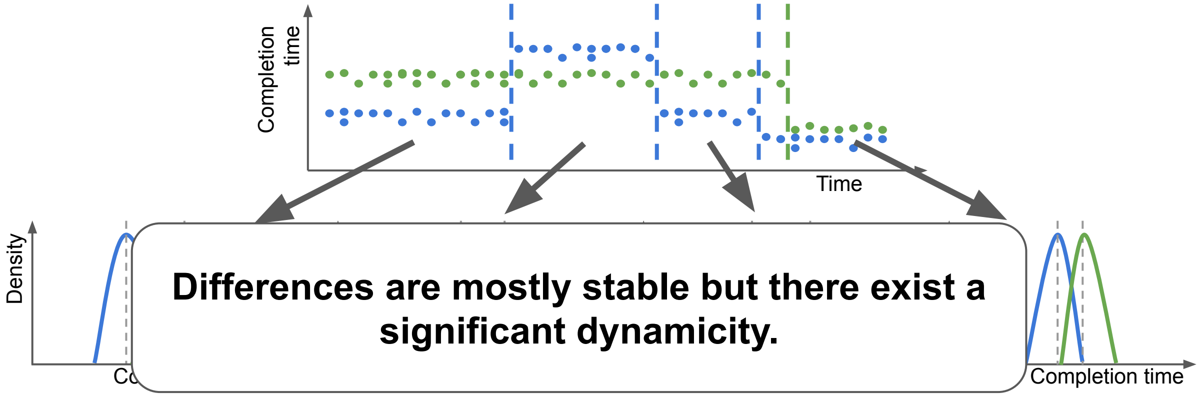


Pair category	IPv4	IPv6	IPv4	IPv6	None	None	Consistent
∃ segment w/ category	IPv6	IPv4	None	None	IPv4	IPv6	
Pairs total	4 569 (1.15 %)		32 459 (8.15 %)		27 312 (6.86 %)		334 034 (83.85 %)

IPv4 and IPv6 end-to-end latency

- We used a change-point detection algorithm to split probe-anchor timeries.
- The statistical test is repeated on each segment.





Differences are mostly stable but there exist a significant dynamicity.

Pair category	IPv4	IPv6	IPv4	IPv6	None	None	Consistent
∃ segment w/ category	IPv6	IPv4	None	None	IPv4	IPv6	
Pairs total	4 569 (1.15 %)		32 459 (8.15 %)		27 312 (6.86 %)		334 034 (83.85 %)

Interlude

- End-to-end latency differences between IPv4 and IPv6 are real.
    - Sometimes they play in favor of IPv6, sometimes they don’t.
- With the rise of latency-sensitive applications, ISPs and content providers need to make IPv6 as good as IPv4 for the transition to happen.
    - Test for IPv6 latency
    - Improve your peerings and infrastructure

Opportunities for latency-sensitive applications

- Latency-sensitive applications should carefully select the address family.
- A selection technique optimising for latency should:
  - give no a priori preference.
  - distinguish destinations.
  - be able to make its choices evolve over time

Adaptive Address Family Selection for Latency-Sensitive Applications on Dual-stack Hosts

Maxime Piraux
maxime.piraux@uclouvain.be
UCLouvain
Belgium

Olivier Bonaventure
olivier.bonaventure@uclouvain.be
UCLouvain
Belgium

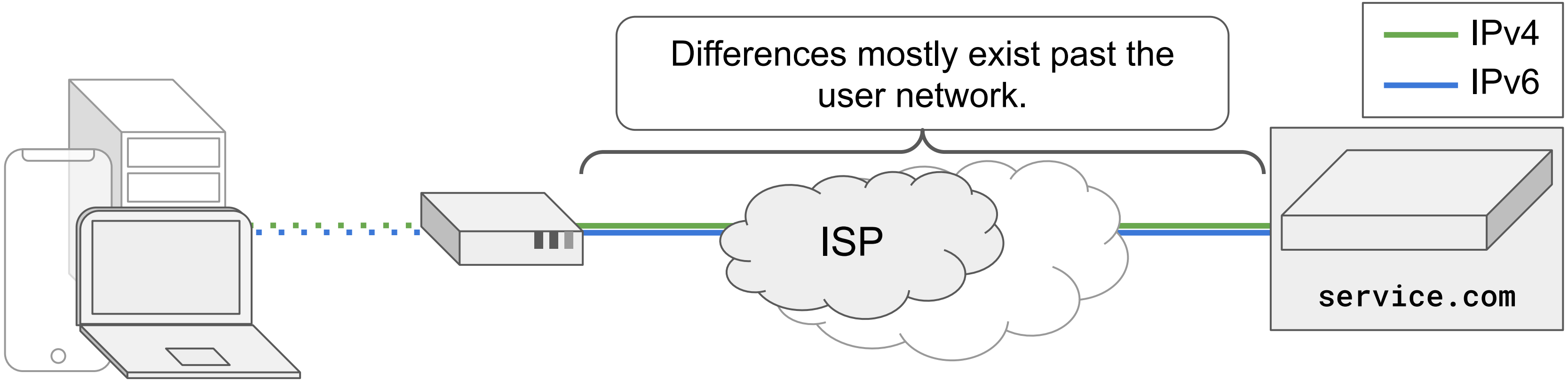
ABSTRACT

Latency is becoming a key factor of performance for Internet applications and has triggered a number of changes in its protocols. Our work revisits the impact on latency of address family selection in dual-stack hosts. Through RIPE Atlas measurements, we analyse the address families latency difference and establish two requirements based on our findings for a latency-focused selection mechanism. First, the address family should be chosen per destination. Second, the choice should be able to evolve over time dynamically.

Given that the adoption of IPv6 on devices, operating systems and networks is heterogeneous [30, 31, 11], very few service providers completely transitioned to IPv6 but rather became dual-stack. As a result, when an application establishes a transport connection, it needs to select one address family. This problem has seen a number of solutions over the years [38, 40, 35]. All of them made the hypothesis that IPv6 should be favoured to foster its transition and include a fallback mechanism in case of a broken IPv6 path. At the early stages of the IPv6 deployment, several transition solu-

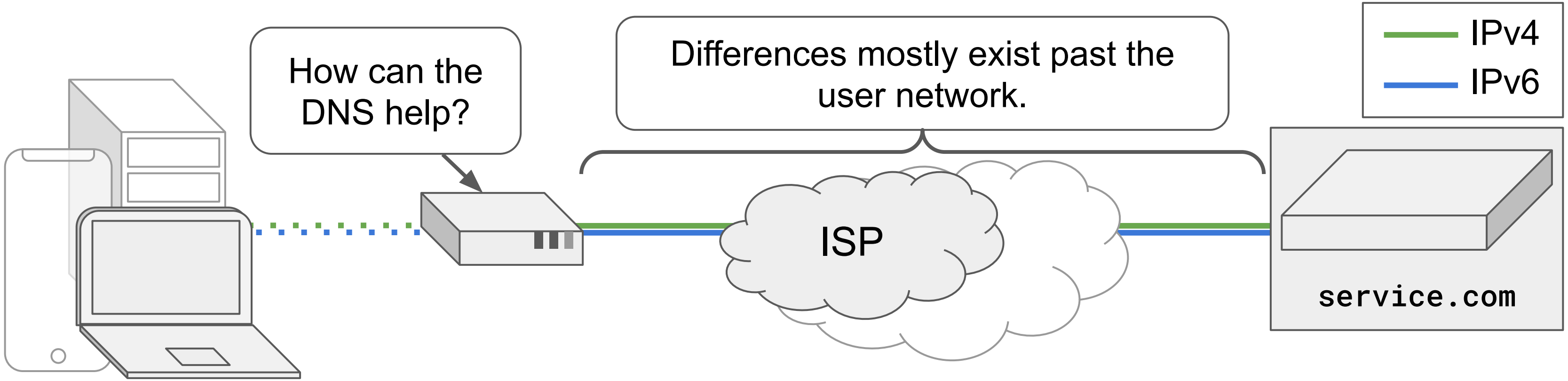
https://arxiv.org/pdf/2309.05369.pdf

Design



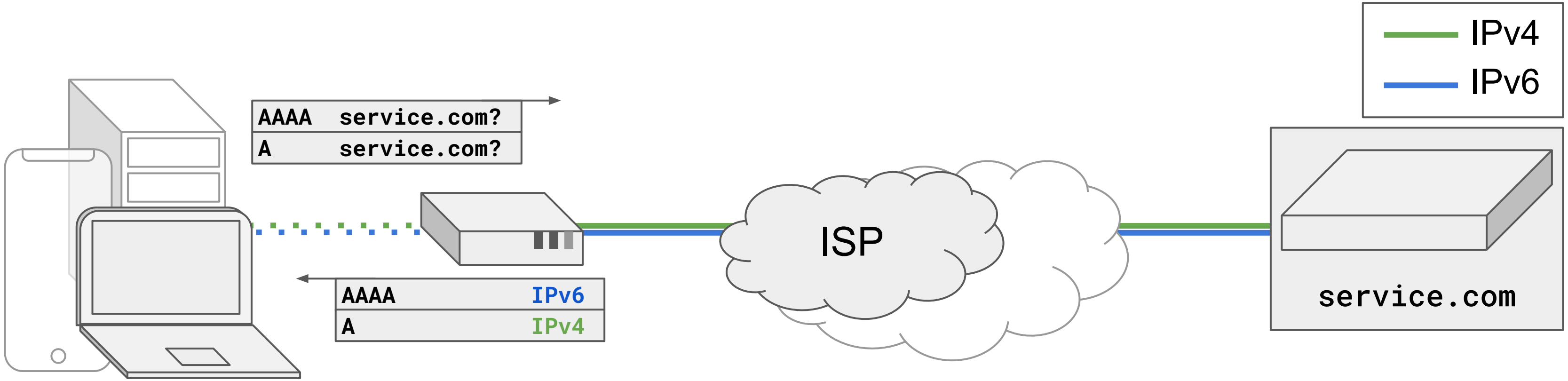
- How to steer hosts?

Design



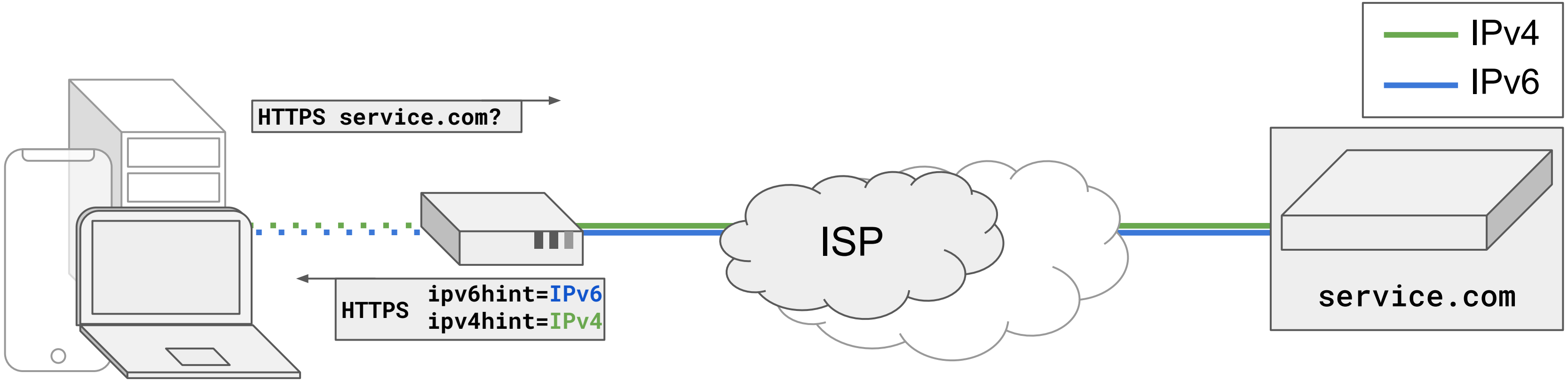
- How to steer hosts?
- The DNS resolver is at the boundary between:
  - User network and WAN.
  - Domain names and IP addresses.

Design



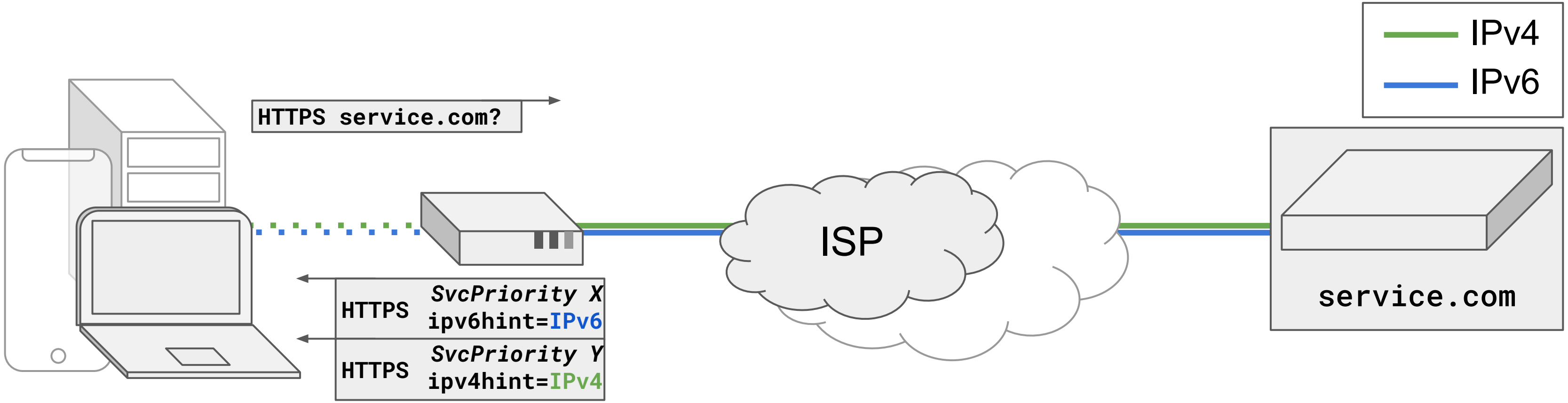
- Hosts with Happy Eyeballs version 2 prefer IPv6.

Design



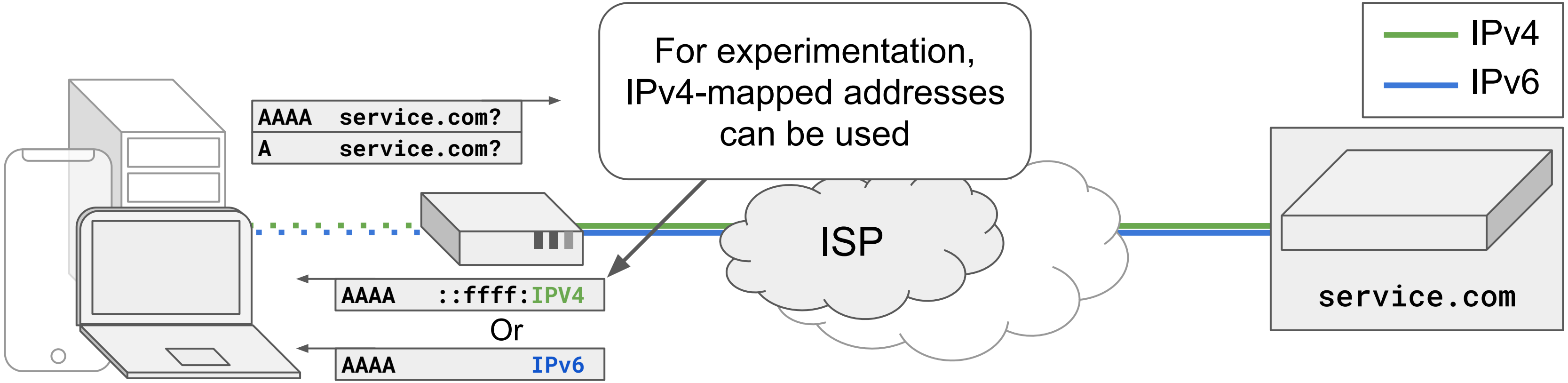
- Hosts with Happy Eyeballs version 3 (draft-pauly-v6ops-happy-eyeballs-v3) can use HTTPS SCVB RRs.

Design



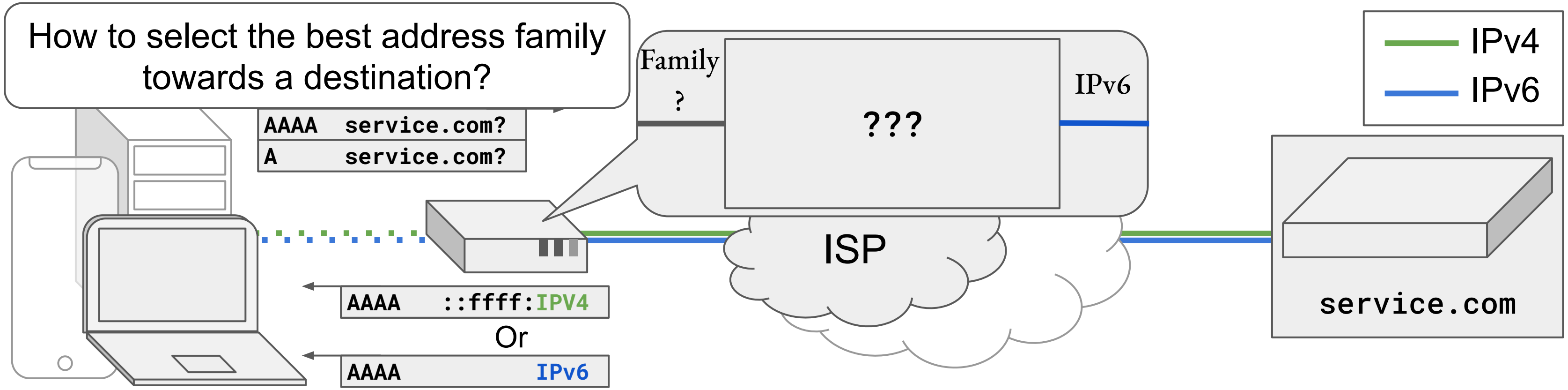
- Hosts with Happy Eyeballs version 3 ([draft-pauly-v6ops-happy-eyeballs-v3]()) can use HTTPS SCVB RRs.
- The resolver can influence the order established by HE by changing the priority of HTTPS RRs.

Design



- For experimentation, IPv4-mapped addresses can be used at the expense of preventing fallback.

Design



- The resolver must balance choices exploration and exploitation.
  - Reinforcement learning problem

More to read in the paper

- We explore how the address family selection can be formulated as a multi-armed bandit problem.
- We validate our design using the RIPE data.
- We implement and evaluate a DNS resolver prototype with Chrome loading popular web services on real networks.

Adaptive Address Family Selection for Latency-Sensitive Applications on Dual-stack Hosts

Maxime Piraux
maxime.piraux@uclouvain.be
UCLouvain
Belgium

Olivier Bonaventure
olivier.bonaventure@uclouvain.be
UCLouvain
Belgium

ABSTRACT

Latency is becoming a key factor of performance for Internet applications and has triggered a number of changes in its protocols. Our work revisits the impact on latency of address family selection in dual-stack hosts. Through RIPE Atlas measurements, we analyse the address families latency difference and establish two requirements based on our findings for a latency-focused selection mechanism. First, the address family should be chosen per destination. Second, the choice should be able to evolve over time dynamically.

Given that the adoption of IPv6 on devices, operating systems and networks is heterogeneous [30, 31, 11], very few service providers completely transitioned to IPv6 but rather became dual-stack. As a result, when an application establishes a transport connection, it needs to select one address family. This problem has seen a number of solutions over the years [38, 40, 35]. All of them made the hypothesis that IPv6 should be favoured to foster its transition and include a fallback mechanism in case of a broken IPv6 path. At the early stages of the IPv6 deployment, several transition solu-

https://arxiv.org/pdf/2309.05369.pdf

Next steps

- We are seeking collaborations regarding IPv6 multihoming.
- Full article on arxiv.org, under revision in a journal.
  - Datasets and code will be made public.
- Reach out to me at maxime.piraux@uclouvain.be.
- Let’s discuss extending the use of DNS and improving latency in IPv6 multihoming scenarii.