There AND back
Designing reverse traceroute

Valentin Heinrich
Rolf Winter

Hochschule Augsburg University of Applied Sciences

Neulich im Netz

conntac
The self-service company

Traceroute (TR)

- Traceroute (TR) is sometimes referred to as "the number one go-to tool for troubleshooting problems on the Internet"
  - Quote is from a NANOG talk that is being held sort of regularly[1]
  - DENOG folks use Traceroute regularly, too
  - Last mail from the DENOG mailing list including traceroute output was on the Thread "Hilfe bei Eingrenzung Packetloss zu DTAG" (10.11.2022)
- While it appears simple, it can be challenging to interpret its results
- This talk is about an ID we have submitted recently to the IETF
  - Reverse Traceroute
  - https://datatracker.ietf.org/doc/html/draft-heiwin-intarea-reverse-traceroute
- You (and every Internet user) are the "customers" of this work

[1] A Practical Guide to (Correctly) Troubleshooting with Traceroute, Richard Steenbergen, NANOG 80, https://youtu.be/L0RUI5kHzEQ

Collecting feedback

- Everybody (online and at the venue) go to https://twbk.de
- Enter the following session ID:

1234

- Feedback is anonymous, but you'll see the aggregated results

Analyse this!

You suspect a problem. You run traceroute. You get the following output.

```
1   routerA.aug.net-a.com  (10.10.10.10)  1ms     2ms     1ms
2   routerB.muc.net-a.com  (20.20.20.20)  5ms     6ms     12ms
3   routerC.fra.net-a.com  (30.30.30.30)  11ms    21ms    14ms
4   routerD.fra.net-b.com  (40.40.40.40)  340ms   320ms   350ms
5   www.example.com      (50.50.50.50)  345ms   310ms   360ms
```

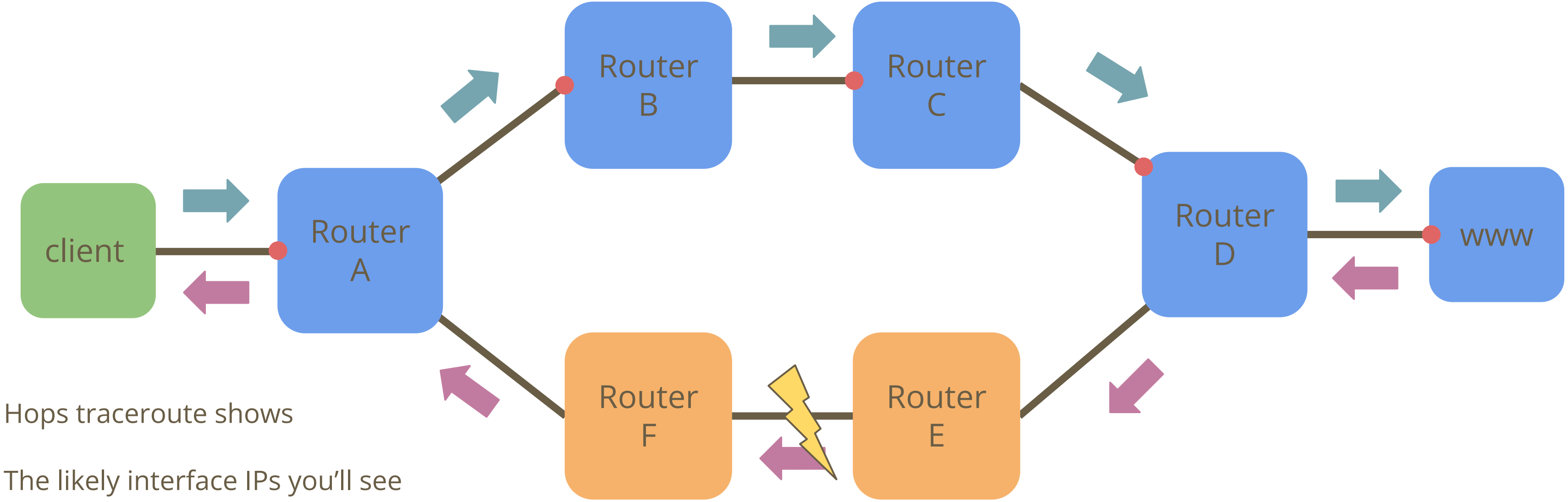
What is your conclusion?

A. Problem? What problem? This is how I would expect the output to be.
B. There is something wrong between routers C and D (hops 3 and 4).
C. You cannot really tell given this output alone.

twbk.de → 1234

Well...

```
1     routerA.aug.net-a.com    (10.10.10.10)     1ms    2ms    1ms
2     routerB.muc.net-a.com    (20.20.20.20)     5ms    6ms    12ms
3     routerC.fra.net-a.com    (30.30.30.30)     11ms   21ms   14ms
4     routerD.fra.net-b.com    (40.40.40.40)     340ms  320ms  350ms
5     www.example.com          (50.50.50.50)     345ms  310ms  360ms
```



Hops traceroute shows

The likely interface IPs you'll see

Packets on the forward path

Packets on the reverse path

Routers on the reverse path

Remember the DENOG mail from 10.11.22

- “Hat jemand von Euch einen DTAG Anschluss und könnte den umgekehrten Weg (z.B. zu *a.b.c.d*) mal prüfen?

Translates to: Does anybody amongst you have a DTAG internet connection and could check the return path for me?

Our **goal** is to design and implement a **reverse traceroute** mechanism for problems just like this one, that hopefully becomes as **ubiquitously available** just as traceroute is today.

One past attempt

- "Traceroute Using an IP Option", RFC 1393, January 1993
  - A special IPv4 option is added to TR packets (incl. the IP address of the originator)
  - Causes a router to send a special TR message to the originator
  - Packet with the option is simply forwarded
  - The receiver also sends a packet incl. above option with the originators address
- Why don’t we have this yet?
  - Well, likely the need for router support and the use of IP options
  - It teaches us to be careful with design choices
  - RFC 1393 was obsoleted in 2012

Design goals

No direct control over the remote host. 1

What makes ping and traceroute so successful, is that they work without control over the host replying to the messages sent.

Safe to use 2

Reverse traceroute should not be usable as a DoS tool, neither for the host nor for the network.

Deployability 3

Reverse traceroute should be designed in a way in which it can be widely deployed on today’s ossified internet, e.g. work through common middleboxes.

Policability 4

Reverse traceroute should be easily policable at network boundaries, even at line-rate.

Design goals

Awareness of load-balancing
5

Load-balancing is the norm on today's internet. We need to control load-balancing as part of the protocol.

No hackery
6

Reverse traceroute should not resort to practices that are frowned upon such as source IP address spoofing.

No router changes
7

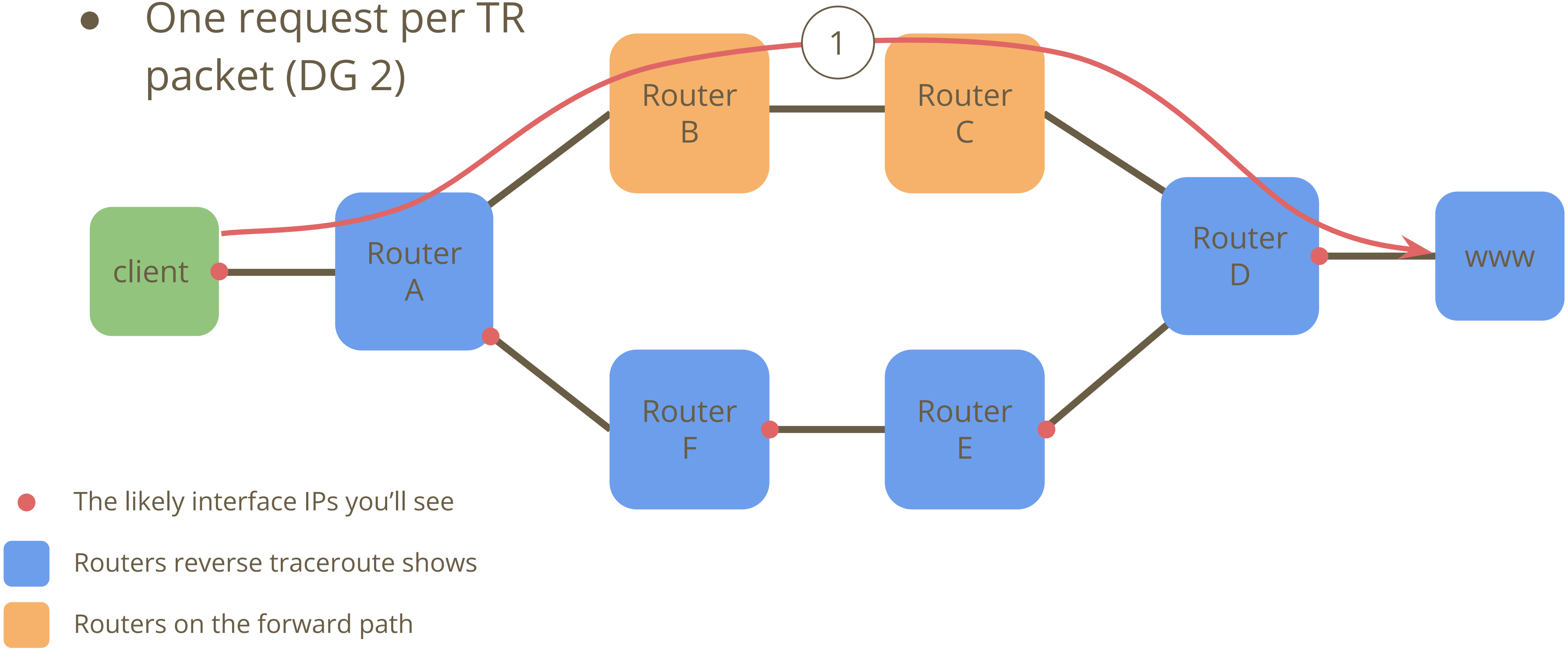
Routers should remain untouched. Things will become much more difficult if routers are involved.

Mimic traceroute
8

Reverse traceroute should allow to measure both the hops along the path and the RTT towards these hops, just as traceroute does for the forward path.

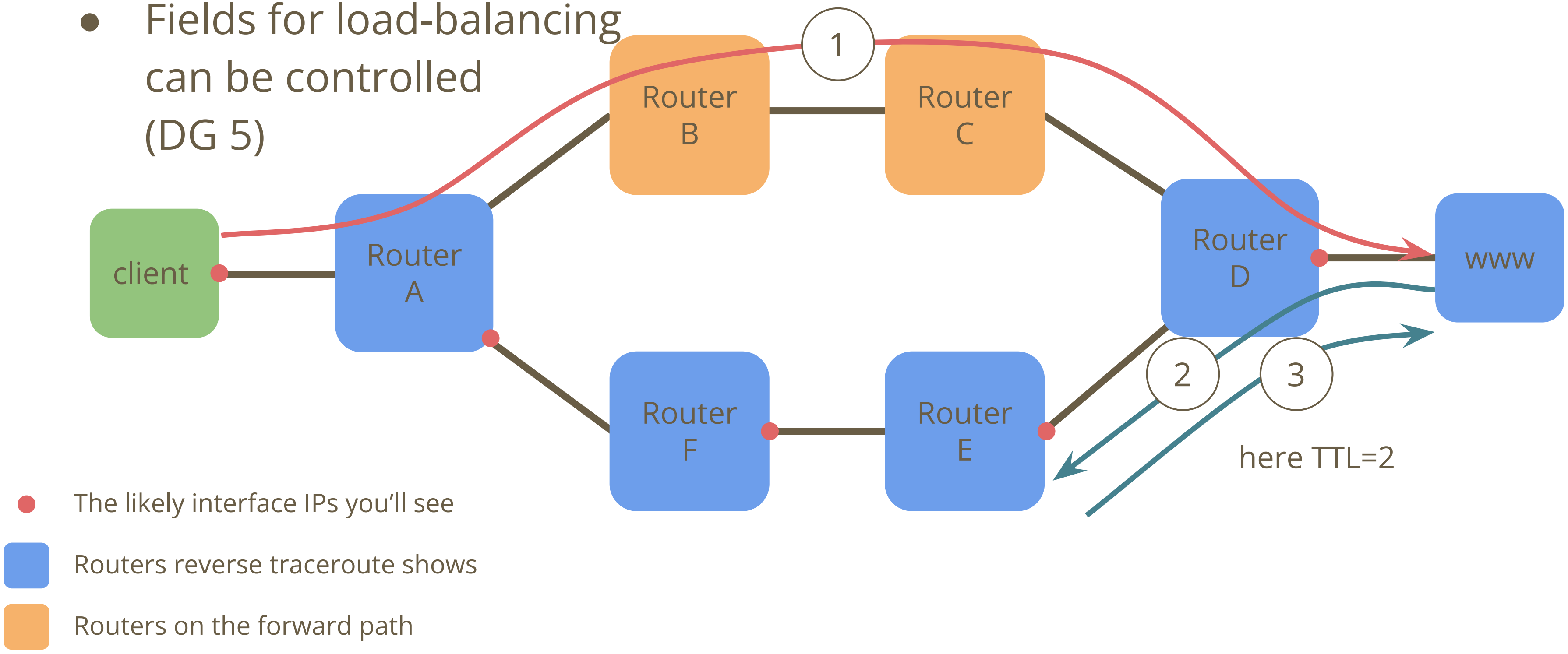
Meet reverse traceroute

- Uses a new ICMP request to trigger a reverse traceroute (DG 1, 4, 6)
- One request per TR packet (DG 2)



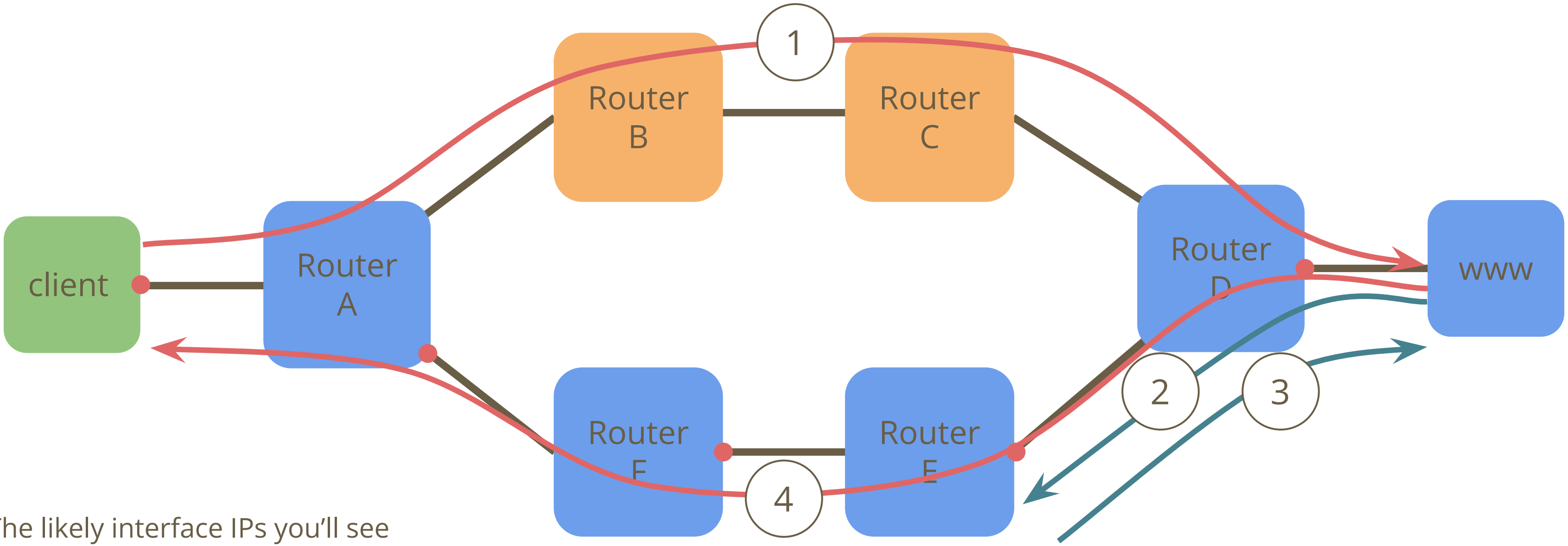
Meet reverse traceroute

- A regular TR packet is sent (UDP, ICMP or TCP) (DG 3, 7, 8)
- Fields for load-balancing can be controlled (DG 5)



Meet reverse traceroute

- For that single probe, an ICMP response is sent back



- The likely interface IPs you'll see
- Routers reverse traceroute shows
- Routers on the forward path

How do you feel about this?

A.
B. This seems sensible
C. OMG, there are more packets generated at th

Headers, code points ... oh my

- Reverse Traceroute is defined for both ICMP and ICMPv6
- ICMP messages typically start like this:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Question, which `Type` and `Code` to use:
  - Option A: New types and codes
  - Option B: Existing type and new codes
- Real question: which ones work on today's internet (DG 3)

What about middleboxes?

- The internet is ossified, mainly thanks to middleboxes
  - NATs e.g., are a pretty common middlebox
- Question: which packets go through NATs
- Tested 12 NAT implementation:
  - We sent two packets with type 8 (used by ping request) and codes 1 and 2 (standard ping uses 0), replies matched the code but used type 0
  - And two unassigned types (7 and 252) with code 0 each

ICMP request	forwarded	filtered	bypassed
Type 8, code 1	11	1[a)]	0
Type 8, code2	11	1[a)]	0
Type 7, code 0	1	7	4
Type 252, code 0	1	6	5

[a)] Response dropped

But what happens to those packets on the internet?

- We picked ten million IPv4 addresses at random and send an ICMP Echo request there (good old Ping)
- For each host that responded, we sent an ICMP Packet with the Echo type but a different code (code 1)

Filtered	Reflective	Unreflective	Erroneous
39.993	931.427	32.478	659[a)]

[a)] mostly dest. unreach.

Conclusion

- Call for action
  - Read the draft and join the discussion at the IntArea WG (IETF)
  - Offer to host a reverse traceroute end-point
  - Use our reverse traceroute client and send us the output
- We could use old home gateways
  - More NAT implementations
  - Other research work
- Website: https://net.hs-augsburg.de/en/project/reverse-traceroute/
- Github: https://github.com/HSAnet/reverse-traceroute
- Contact: rolf.winter@hs-augsburg.de
- If you liked this, you'll love "Neulich im Netz - der Internet-Podcast"

Gefördert durch:
Bundesministerium für Wirtschaft und Klimaschutz
aufgrund eines Beschlusses des Deutschen Bundestages