Tracing Internet Path Transparency

Mirja Kühlewind, Micheal Walter, Iain R. Learmonth, and Brian Trammell

June 28, 2018

TMA Conference, Vienna

mami

measurement and architecture for a middleboxed internet

measurement | architecture | experimentation

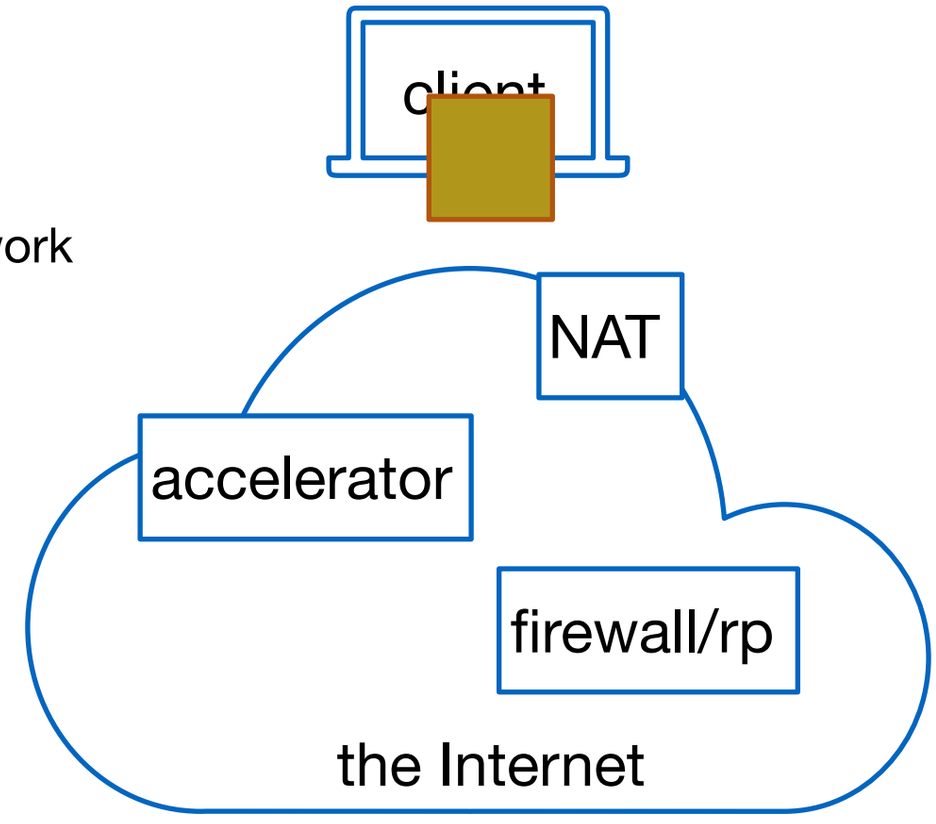
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688421.The opinions expressed and arguments employed reflect only the authors' view. The European Commission is not responsible for any use that may be made of that information.

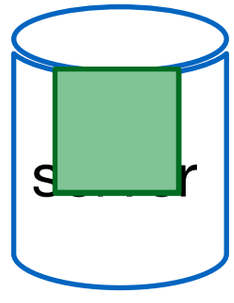
What is Path Transparency

- The Internet is notionally *transparent:*
  - packets come out the other end of the pipe unchanged.
  - based on the *end-to-end principle*: a maximally capable network made of smart endpoints connected by dumb pipes

- This is not how things actually are, especially at layer 4:
  - Network address translation
  - Extension and option blocking and stripping
  - TCP ACK/SEQ rewriting
  - etc, etc, etc, etc…

➡ Middlebox functions can impair the connectivity and treatment of end-to-end traffic



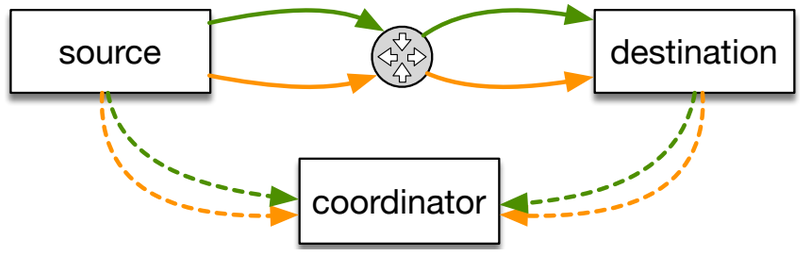


➡ Designing protocols and protocol extensions that can deal with interference require us to measure and understand the nature and prevalence of different kinds of impairments

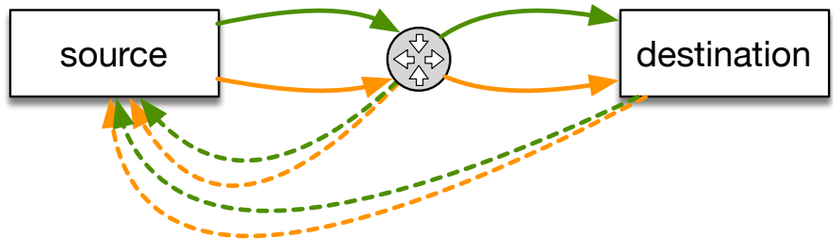
Active Measurement of Path Transparency

- Controlled experimentation (A/B testing): compare "vanilla" traffic to some feature under test.

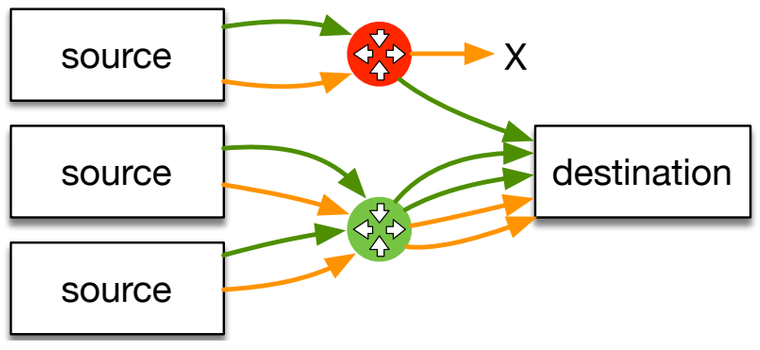
- Ideally, control both endpoints
  - compare packets send to packets received



- This scales poorly but can infer path behavior from the destination's response
  - or induce routers to send us a response (traceroute)



- Comparing results from multiple vantage points with different paths toward the same destination
  - infer on-path versus on-endpoint or near-endpoint interference

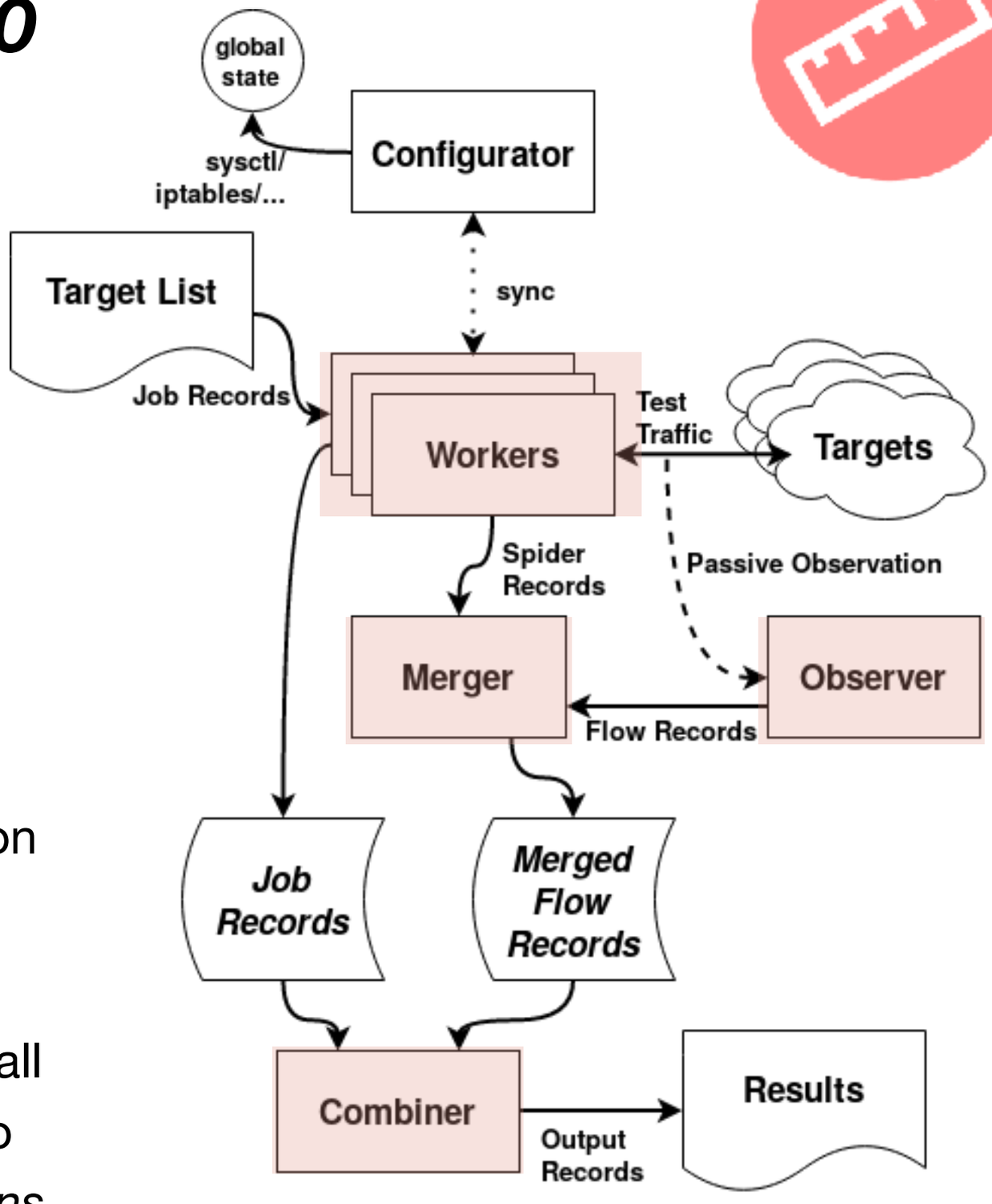


PATHspider 2.0

- Generalized framework for A/B testing
- Plugin-based architecture with plugins available for
  - ECN connectivity and negotiation
  - DiffServ Codepoints
  - TCP Fast Open
  - …
- Result outputs **Path Observations**
  - where a certain *condition* (e.g. `ecn.negotiation.succeeded`) has been observed at a certain point of *time* on a certain *path*

Design of PATHspider *2.0*

- Four main components
  - **Workers** open the connection and send test data for all target on target list
  - **Observer** passively monitors all out-going and in-coming packets
  - **Merger** appends record information from each worker to the passively observed flow records
  - **Combiner** analyses the results of all connections attempts belonging to one test and generates *observations*



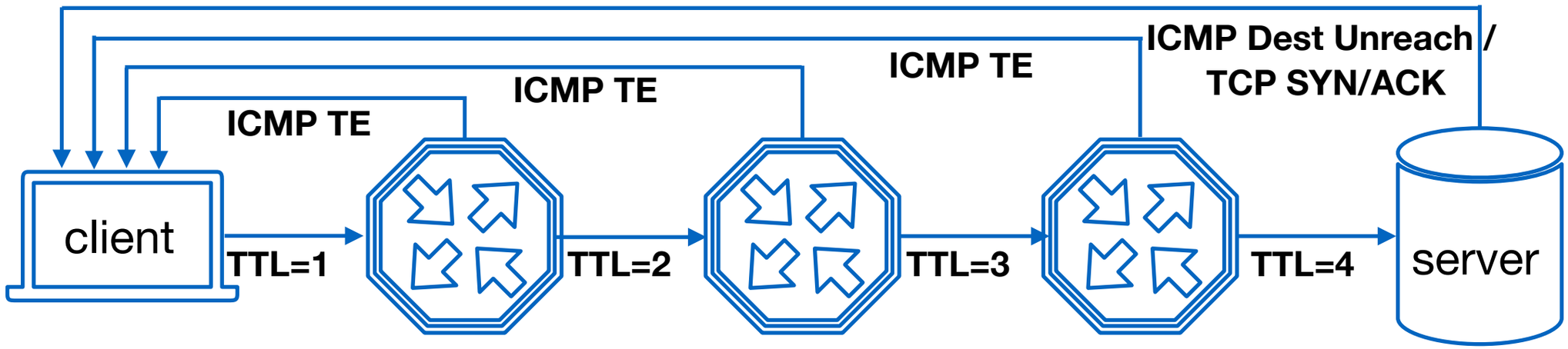
PATHspider 2.0 - New features

- Generalized to support more than just A/B testing
  - Any permutation of any number of tests now possible!
- PATHspider's is now using cURL for HTTP requests -> faster
- Framework for packet forging based plugins using Scapy
- Completely rewritten (in Go) target list resolver
  - Faster target list IP address resolution!
- Observer modules usable for standalone passive observation or analysis

- See https://github.com/mami- project/pathspider/tree/2.0.0/

Quick recap: traceroute and tracebox

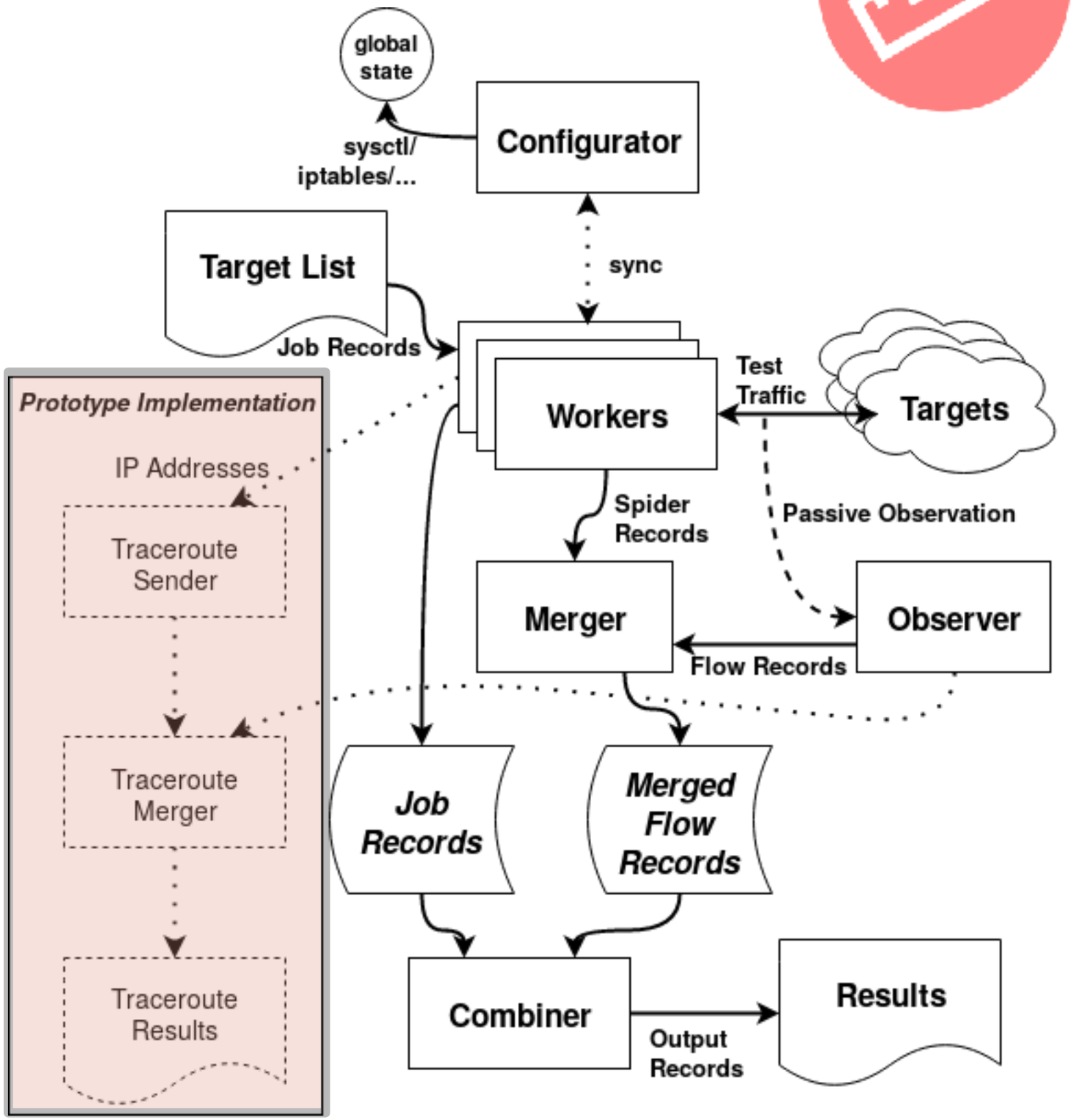
- traceroute is a active network measurement tool to identify path/hop information utilizing ICMP Time Exceeded messages



- tracebox in addition also inspects the returned content of the originally sent packet and analyses changed on the path
  - to locate the origin of the impairment observed
  - See http://www.tracebox.org/

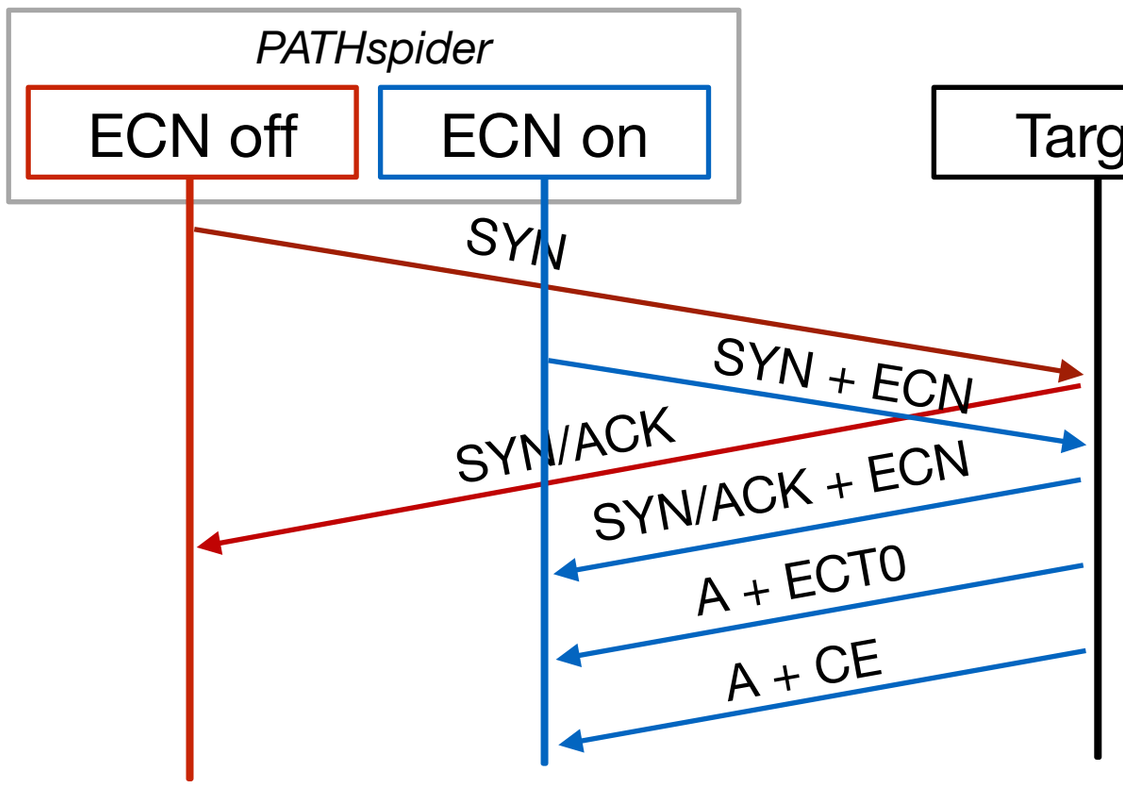
Integrating of traceroute/tracebox into PATHspider

- traceroute is automatically performed depending on results
- **Tracroute Sender** creates forged packet based on observed behavior
- **Traceroute Merger** analyses the returned ICMP messages (similar as tracebox)



Example study: Explicit Congestion Notification (ECN)

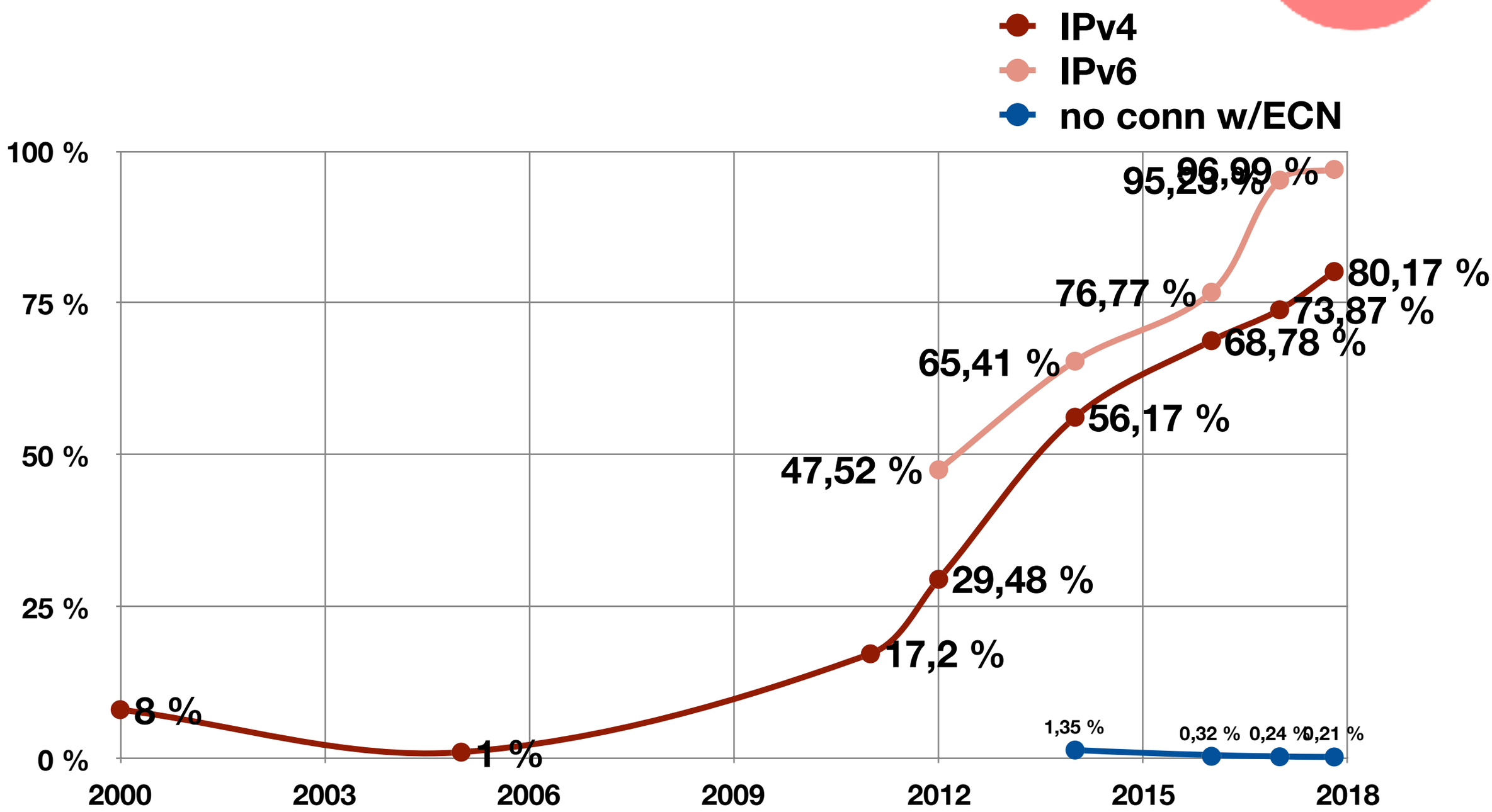
ECN is an TCP that allow router to signal congestion using to bits in the IP if successfully negotiated between both endpoints during the TCP handshake (SYN -> SYN/ACK)



1. `ecn.connectivity.`*`status`*
   - `works`: off + on OK
   - `broken`: off OK, on fails
   - `transient`: on OK, off fails
   - `offline`: no connection
2. `ecn.`*`negotiation`*`.succeeded/failed`
3. `ecn.ipmark.ECT`*`1`*`/ECT0/CE.seen/not_seen`

CE = Congestion Experienced; ECT = ECN Capable Transport

ECN support on webservers (Alexa 1Mio)



Enhanced ECN plugin

- 4 TCP connections
  - Baseline: TCP SYN without ECN negotiation attempt
  - TCP SYN with ECN + no ECN IP codepoit (Not-ECT)
  - TCP SYN with or without ECN + ECT(1)
  - TCP SYN with or without ECN + CE
- traceroute to detect mangling of the IP ECN codepoint as well as DiffServ codepoint (set to 46=Expedited Forwarding)

Used to be 8-bit Type of Service (ToS) field

Version	IHL	DSCP	ECN	Total Length
Identification			Flags	Fragment Offset
Time To Live		Protocol		Header Checksum
Source IP Address				
Destination IP Address				

ECN negotiation with IP ECN codepoint

- 69.35% only negotiated ECN when the ECN IP codepoint was set to zeros (non-ECT) but not if ETC0 or CE was set
- Only 12.79% of the hosts negotiated ECN no matter what codepoint was set
- 26 hosts negotiated ECN when ECT0 was set but not when CE was set

DSCP and ECN IP Codepoint Manipulation

without ECN nego but ECT(0) (for 201,854 hosts)

DSCP treatment	ECN → ECT0 (preserved)		ECN → Non-ECT (rewritten)		**total**	
	n	pct	n	pct	n	pct
→ EF (unchanged)	41850	20.7%	169	0.08%	42019	20.8%
→ 6 (three-bit bleach)	87182	43.2%	101	0.05%	87283	43.2%
→ 0 (bleach)	50031	24.8%	1665	0.82%	51686	25.6%
→ CSx	4883	2.42%	701	0.35%	5584	2.77%
→ AFxx/VA	9951	4.93%	68	0.03%	10019	4.96%
→ undefined value	5182	2.57%	81	0.04%	5263	2.61%
total	199079	98.6%	2775	1.37%	201854	100%

- Side note: 3,252 hosts reflected ECT(0) in the SYN/ACK even though ECN was not requested in the SYN
- Four of five paths see some DSCP manipulation

DSCP and ECN IP Codepoint Manipulation

without ECN nego but ECT(1) (for 201,854 hosts)

DSCP treatment	ECN → ECT0 (preserved)		ECN → Non-ECT (rewritten)		**total**	
	n	pct	n	pct	n	pct
→ EF (unchanged)	41850	20.7%	169	0.08%	42019	20.8%
→ 6 (three-bit bleach)	87182	43.2%	101	0.05%	87283	43.2%
→ 0 (bleach)	50031	24.8%	1665	0.82%	51686	25.6%
→ CSx	4883	2.42%	701	0.35%	5584	2.77%
→ AFxx/VA	9951	4.93%	68	0.03%	10019	4.96%
→ undefined value	5182	2.57%	81	0.04%	5263	2.61%
total	199079	98.6%	2775	1.37%	201854	100%

- For 2,775 (1.37%) hosts, the ECT0 codepoint was erased before received at server
  - about 50% of the codepoint removal in the last hop
  - more than 90% of the cases in the last 40% of the path
- The majority of ECN-manipulating paths also bleach the DSCP codepoint
- ECN codepoint is set to 0 while the DSCP codepoint is set to a CS value
  - indicates treatment according to the old ToS definition :-(

Summary and Conclusion

- Integration of path tracing + trace analysis into PATHpsider
  - automatically trigged in timely succession when configured failure conditions is observed

    ➡ detect middlebox impairments as well as location

    ➡ gives indications about the root cause of impairments

- Majority of on-path interference with the ECN IP codepoint is linked to older interpretation of the ToS byte
  - While ToS bleaching was observed on the whole network path (border routers), active ECN IP rewriting is more commonly performed at edge networks
- Also, connectivity breakage was still observed and is often linked to ICMP breakage -> no traceroute feedback :-(

Learn more about how to use PATHspider!

SIGCOMM Tutorial on

Repeatability and Comparability in Measurement (RCM)

on August 20, 2018, 2pm-5:45pm, Budapest,

- **Part I: Introduction and Topology Measurement**
  - Welcome and Introduction (Brian Trammell, ETH Zurich)
  - Tracebox: Topology Measurement and Impairment Discovery (Korian Edeline, U. Liege)
- **Part II: Path Transparency and Data Collection**
  - PATHspider: A Tool for Controlled Hybrid Measurement (Iain R. Learmonth, U. Aberdeen)
  - Observatories: Collection, Preservation, Metadata and Provenance for Active Measurement (Brian Trammell, ETH Zurich)
  - The Path Transparency Observatory (Brian Trammell, ETH Zurich)

See https://conferences.sigcomm.org/sigcomm/2018/tutorial-rcm.html