Sidecar: In-Network Performance Enhancements in the Age of Paranoid Transport Protocols

Gina Yuan, David K. Zhang, Matthew Sotoudeh, Michael Welzl[^], Keith Winstein

HotNets 2022, November 14-15, 2022



Once upon a time, there was TCP/IP.



But the "right way" to implement TCP can depend on the particulars of the network path.



Retransmission problems when combining a high-delay link with a lossy link.

In the 1990s, middleboxes tried to transparently help with in-network TCP acceleration.



Connection Splitting [EUNICE '12, LCN '19], Satellite PEPs [VETECS '06, IJNGC '11], Cellular PEPs [ACSSC '17, MobiCom '95], Virtualized Congestion Control [SIGCOMM '16 x2]

But having to conform to the expectations of TCP middleboxes resulted in protocol ossification.



Multipath TCP [NSDI '12], Unordered TCP [NSDI '12], De-Ossifying the Internet Transport Layer [IEEE CS&T '17], It's Time to Replace TCP in the Datacenter [arXiv '22], Now, transport protocols are "paranoid," hiding even protocol fields, and transparent proxies cannot help.

Before: TCP Header

Source Port		Destination Port
Sequence number		
Acknowledgment number		
DO RSV	/ Flags	Window
Checksum		Urgent Pointer
Options		
TLS-Encrypted Data		

Now: UDP Header w/ QUIC



Proxies could still be useful, but only in ridiculous ways that tie the performance enhancement to the underlying protocol.



Problem: How do we enable in-network performance enhancements for paranoid transport protocols without ossification?

Proposal: sidecar protocol that leaves the original protocol unchanged on the wire, and can freely be ignored.



Proposal: sidecar protocol that leaves the original protocol unchanged on the wire, and can freely be ignored.



The **quACK** communicates exactly which packets the sidecar has received, WITHOUT looking at protocol fields, WITHOUT compromising the underlying protocol.

Example: Congestion Control Division



Customize the congestion control mechanism at the proxy and the host.

Challenge: Without sequence numbers, there is no language to refer to the packets in the underlying transport protocol.

Refer to packets by another **identifier,** e.g., the first 32-bits of randomly-encrypted data.

Challenge: It is hard to refer to a set of identifiers in a way that is both concise and efficient. (Collisions? 0.000023% chance.)

Refer to packets by another **identifier,** e.g., the first 32-bits of randomly-encrypted data.



Parameters: quACK once per 60ms RTT, 200 Mbps link, 2% loss rate ==> 1000 sent packets, 20 missing packets

Solution: The quACK uses power sum polynomials¹ to *concisely* and *efficiently* communicate the subset of received packets.



[1] David Eppstein and Michael T. Goodrich. 2011. Straggler Identification in Round-Trip Data Streams via Newton's Identities and Invertible Bloom Filters. IEEE Trans. Knowl. Data Eng. 23, 2 (2011), 297–306. https://doi.org/10.1109/TKDE.2010.132

Solution: The quACK uses power sum polynomials¹ to *concisely* and *efficiently* communicate the subset of received packets.



quACK: The first *t* power sums of the identifiers of all received packets for some threshold number of missing packets *t*, and a count.

• For each $x \in R$, the receiver accumulates:

```
y_1 = \Sigma x \pmod{p}y_2 = \Sigma x^2 \pmod{p}\dotsy_t = \Sigma x^t \pmod{p}
```

```
count += 1
```

- Given S and the quACK, the sender decodes S / R:
 - Calculate $y'_i = \Sigma_{x \in S} x^i$ for $i \in 1,...,t$.
 - Calculate $d_i = y'_i y_{i.}$
 - Solve the system of *t* polynomial equations in at most *t* variables.

[1] David Eppstein and Michael T. Goodrich. 2011. Straggler Identification in Round-Trip Data Streams via Newton's Identities and Invertible Bloom Filters. IEEE Trans. Knowl. Data Eng. 23, 2 (2011), 297–306. https://doi.org/10.1109/TKDE.2010.132

The quACK can be used in many more sidecar protocols...



Example: ACK Reduction

Example: In-Network Retransmission

- 1) Who quACKs to whom?
- 2) What does the sender do with the information in the quACK?
- 3) Parameters: number of identifier bits *b*, threshold *t*, quACKing frequency.

Conclusion



- Sidecar protocols enable transparent, in-network performance enhancements for paranoid transport protocols. QuACKs make it possible to *concisely* and *efficiently* refer to encrypted packets for several sidecar protocols.
- Next Steps: How much do we trust information from the sidecar? Can we handle adversarial proxies? What other theoretical guarantees could the quACK provide? What other protocols can emerge from this proposed functional separation?