# The Sidecar: “Opting in” to PEP Functions

**Michael Welzl**
*(includes joint work with Gina Yuan, David K. Zhang, Matthew Sotoudeh, and Keith Winstein [2] )*

*We need visible, not transparent PEPs*

For congestion control, end-to-end is a dead-end street. This has become particularly clear with the advent of wireless links such as 5G cellular or, worse, mmWave, where a bottleneck’s available capacity can suddenly undergo strong fluctuations. A downward capacity change will make the end host’s congestion control react, but the end host is never told when capacity becomes available again. Previous efforts tried to address this by sharing information from the network to end hosts, but this keeps the control end-to-end, which is the wrong scope for this problem (how would a host know that the information truly represents a path’s bottleneck?).

In principle, TCP Performance Enhancing Proxies (PEPs) could help, as they execute control in the network, especially when they split TCP connections – but they are not doing this right: they have to “cheat” TCP and therefore cannot make per-application decisions, and they also cause ossification. These PEP problems could be avoided with a proxy that is *visible* and explicitly addressed (as in the “0-RTT TCP Convert Protocol“, RFC 8803, which can be seen as giving a hint about how the proposal in the present document could be implemented).

*These visible PEPs should offer opt-in services*

When a PEP is visible, there is an opportunity to implement its various possible performance-enhancing mechanisms as a “sidecar”, which:

- splits out general functionality, thereby minimizing the necessary changes for the main protocol (such as TCP or QUIC),
- offers mechanisms as *opt-in* services,
- and provides a similar interface to all protocols.

These functions could include, e.g.: sending ACKs on behalf of the receiver, to reduce the number of ACKs that a receiver must send, or sending ACKs that make the sender increase its cwnd (i.e., transmit more data to the proxy faster than the receiver could accept, such that it could be buffered at a proxy adjacent to a fluctuating-capacity wireless link).

Such services could be chosen via a local interface, directly on the involved hosts, e.g. only on the server side (depending on the service in question). The proxy itself could produce acknowledgments by hashing over a fixed number of bytes from the main protocol’s transport header, without ever requiring to understand anything about it (thereby avoiding ossification).

This idea was first presented to at IETF-113 [1]; a more detailed description is given in [2].

## References

[1] “Service Awareness rather than Path Awareness“, presentation, PANRG, IETF-113: https://datatracker.ietf.org/meeting/113/materials/slides-113-panrg-service-awareness-rather-than-path-awareness-00

[2] *Gina Yuan, David K. Zhang, Matthew Sotoudeh, Michael Welzl, Keith Winstein: "Sidecar: In-Network Performance Enhancements in the Age of Paranoid Transport Protocols", accepted for publication, ACM HotNets, November 14-15, 2022.*