Dynamically Switching Transport Protocols With STRAP

Jessica Berg, Sam Shih, Kevin Webb

{jberg1, sshih1, kwebb} @cs.swarthmore.edu

Introduction

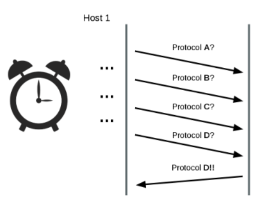
Today's Internet is limited in its modern uses because it uses only two ways to access the internet from the operating system, both of which were built in the 1980's. At that time, no one could have foreseen the changing needs of modern Internet applications.

Newer transport protocols, or rules that govern communication across the Internet, have been developed to provide services that better fit the needs of today, such as optimizations for smartphones, security, video streaming, and Wifi. However, they are not widely used because to communicate in a new protocol, talking hosts must agree on the protocol a priori, which introduces significant latency. So for the past 40 years we have been entrenched in using the same two protocols.

STRAP is a system that allows pairs of hosts to switch to any protocol based on application preferences without adding significant latency.

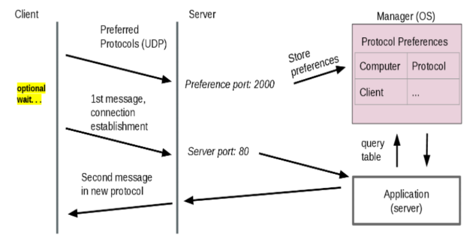
Avoiding Latency

If two hosts are trying to agree on a protocol to speak in, they must first agree on it. However, asking another host over and over whether they speak a certain protocol adds significant unwanted latency. That is a high price to pay for many who use the internet; in 2014, Amazon reported that even a 0.1 second delay in its services would have cost 889 million dollars in sales.



Design

At a high level, STRAP follows the following pattern to switch protocols: first, the client sends their preferred protocols. Then there is an optional wait, ensuring that the preferences have time to be stored on the other side. Second, the client sends the first message of their conversation. Upon receiving that message, the server side looks at its records and finds the client's preferred protocols. The server then sends back a message in the new protocol. From that message, the client will know that the transition went through and that both sides may now use the new protocol.



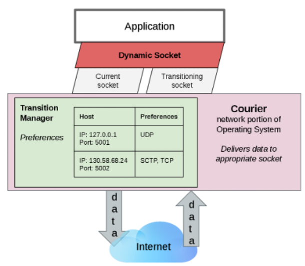
Architecture

There are three pieces of STRAP implementation:

1. *Courier:* The courier receives data from the Internet and directs it to the correct application on the computer. It exists as a part of the operating system.

1. *Transition Manager:* The transition manager sends, receives, and stores preference messages, creates new sockets for different protocols, and performs the logic necessary to switch between protocols.

1. *Dynamic Socket:* A socket is an application's interface with the internet; the kind of socket determines the protocol used. When transitioning, there is often a period of time spent making the new socket ready. During this time, the old socket should still function. The dynamic socket coordinates using both sockets simultaneously and switches out the old for the new.



Empirical Results

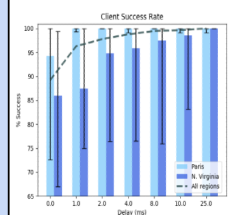
- STRAP currently supports three protocols
  - Transmission Control Protocol (TCP): the most widely used protocol on the internet; provides reliable communication
  - User Control Protocol (UDP): the second most widely used protocol on the internet; unreliable but fast
  - Lightweight and Multi-streamed Protocol (LAMP): a protocol we created, inspired by SCTP, to test STRAP; has more sophisticated start of conversation message exchanges, negotiates state between both parties, can have multiple peers and multiple streams
- STRAP can also support protocols built on top of UDP, the most widely used of these is QUIC, which is used for Google Chrome and YouTube.
- STRAP adds zero round trip time to a conversation whether or not the transition is successful (backwards compatibility).

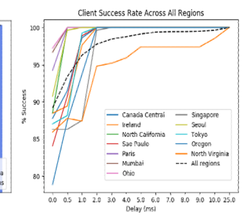
Evaluation

We deployed servers across 12 regions using Amazon Web Services, and had all pairs of computers send and process preference messages together to evaluate the optional wait discussed in *Design*.

The graph on the left shows the median percent successful preferences sent to all servers from clients in Paris and N. Virginia. It also shows the1th and 99th percentile within the distribution, the dotted line represents the median percent successful for clients in all regions.

The graph on the right shows the median percent successful preferences sent to all servers from each client.





Conclusion

We have characterized the different classes of transport layer protocols (connectionless, connection-oriented, one-to-one, one-to-many) and learned what state needs to persist in order to transition between them. We have implemented transitions among three protocols: TCP, UDP, and LAMP. This supports a proof of generalizability and allows future programmers to integrate new protocols into STRAP.

References

Adam Langley,et al. 2017. The QUIC Transport Protocol: Design and Internet-Scale Deployment. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (SIGCOMM '17). ACM, New York, NY, USA, 183-196. DOI: https://doi.org/10.1145/3098822.3098842

Armando L. Caro, et al. 2003. SCTP: A Proposed Standard for Robust Internet Data Transport. *Computer* 36, 11 (November 2003), 56-63. DOI: https://doi.org/10.1109/MC.2003.1244536