Session 2: Rethinking the Internet

Transcribed by: M. Taimoor Tariq, Ammar Tahir (UIUC)

The Case for an Internet Primitive for Fault Localization

William Sussman (MIT); Emily Marx (UC Berkeley); Venkat Arun (MIT); Akshay Narayan (UC Berkeley), Mohammad Alizadeh, ***Hari Balakrishnan*** *(MIT); Aurojit Panda (New York University); Scott Shenker (ICSI and UC Berkeley)*

Summary:

Fault messages are not intuitive and do not have enough information for the user to localize the fault. Users care about localizing faults. All components maintain and emit health bits, and we have a graph of all components. Inspect health bits and infer fault locality when fault localization is activated.

Fault localization should be cross layer (network stack), cross domain (all the entities involved) and cross application (any application involved in the system).

Questions:

1. **What are the incentives? Why would an organization/entity want to take the blame for the fault?**
   Fixing the problem is the incentive. Negative publicity about faults, so organizations don't want to own responsibility. But it does affect your users, + revenue so that should be a motivation/incentive.

Tango or Square Dance? How Tightly Should We Integrate Network Functionality in Browsers?

Alex Davidson (Brave Software); Matthias Frei (ETH Zurich); Marten Gartner (OVGU Magdeburg); Hamed Haddadi (Brave Software); Jordi Subirà Nieto, ***Adrian Perrig*** *(ETH Zurich); Philipp Winter (Brave Software); Francois Wirz (ETH Zurich)*

Summary:

On some properties e.g. about path networking, OS cannot make the appropriate decision, the browser can based on user context. SCION allows such control using a proxy-based design that is directly integrated with the browser.

Questions:

1. **How do users come up with policies?**
   The default would be a high-performance path but there could be policies like preferring a low carbon footprint path at cost of e.g. latency. Parameters such as

CO2 consumption are already being added to path costs, path footprints.

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

***Gina Yuan**, David K. Zhang, Matthew Sotoudeh (Stanford University); Michael Welzl (University of Oslo; Keith Winstein (Stanford University)*

Summary:

Modern protocols like QUIC are encrypted and do not work well with middle-boxes and proxies. Proxies are useful in many cases to improve performance especially when system performance is coupled with the underlying transport protocol.

Questions:

1. **Does the sidecar use the same port/connection?**
   Depends on the context and application. Open question.
2. **Do we need to know intermediaries exist?**
   By quACKing you can discover intermediaries.
3. **What are the advantages of quACKing instead of putting QUIC in another layer? E.g. break one QUIC connection into multiple.**
   QuACK can work with what we have now instead of modifying all protocols
4. **In my opinion per flow fairness isn't a good policy. How 2 adjacent hops in a path can use quack differently based on different interests.**
   Haven't thought about it extensively, interesting question.
5. **Does this open up more opportunities for the usage of middleboxes**
   Having this kind of separation can help ensure middleboxes help us.
   There's a question to be asked whether middleboxes are hurting us or helping us.
   Having this functional separation can help us increase the use of middleboxes and make them less harmful and more beneficial.

DIP: Unifying Network Layer Innovations using Shared L3 Core Functions

***Ziqiang Wang** (Southeast University); Zhuotao Liu (Tsinghua University and Zhongguancun Laboratory); Xiaoliang Wang (Capital Normal University); Songtao Fu (Tsinghua University); Ke Xu (Tsinghua University and Zhongguancun Laboratory)*

1. **What is the incentive for ISPs to allow end-users to construct arbitrary protocols when it goes against their business model/incentives?**
   Attractive for users, that's why it will incentivize ISPs as they will be able to support new devices.

End of Session Discussion

Browser Proxy, if I change my policy, the network will not be changed but once more and more users change their proxies, in that case different users will have different policies so that will lead to unfairness in policies. Users might open and reserve more and more resources.
In a multipath system, life should strictly be equal or get better. Compared to a single path, you now have multiple paths to choose from and you should choose the optimal one. As far as different proxy and different policies are concerned, that is kind of already happening, with browsers opening 10 sessions.

Use of fault localization.
Today it would require some level of widespread tracing process but using the fault localization would allow us to speed up the process

Are we as users making more space for attackers by giving users more control.
Kind of like a tradeoff, but if the defender/app gains more than the attacker, then there is kind of like a net benefit.
Find a balance between usability and security.

Incentive for participating components in fault localization paper. Is accountability essentially an incentive for an entity to participate.
Yes, health bits would not just be localized. Making your peers accountable will be helpful, there should be less pessimism.

Does the system (fault localization) work if not all entities participate.
Open question, not clear yet

How generalizable are the primitives? E.g. header modification? Can health bits be application specific?

Changing the header fields is the primary issue which needs to be avoided.
As far as tracing frameworks are concerned, they are application specific. We don't want to expand the interface, we just want a narrow framework with a very narrow interface (limited number of bits).

It's super hard to convince these big orgs to change/add code to browsers. How do you plan on convincing/incentivizing them to allow users to define network configs?
It is hard but the speaker believes that you can enable it. It can be a simple Chrome extension, you don't even have to make extensive changes to the code.

One could argue that it's easier to add features to a browser than an OS.

Getting the information of fault is harder than fixing it. Issue of observability.
Yes, because the way we do observability is too verbose. We don't start with a general problem and then try to narrow it down. Today developers just log everything and that's terabytes of data and it's just too much information to localize.

Can health bits be used to perform other related tasks such as ECN, Load Balancing, etc.?
It’s an interesting direction. Have not thought about it yet. Could work out.

Having a lot of data is a problem, but cross-service understanding of the data is also a problem?
This is one of the key things, understand the data cross layer. Purpose of WTF is to identify where the problem is, it is the job of that component to identify what the problem is.

Internet paths are becoming shorter, how does your vision fit into this reality.
The use case is limited for shorter paths however most of the bytes today are going through the public cloud.