Session 6: Programming Networks at Scale

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

A Case for Remote Attestation in Programmable Data Planes

Nik Sultana *(Illinois Institute of Technology); Deborah Shands, Vinod Yegneswaran (SRI International)*

Summary:

Remote attestation defines protocols for building and using evidence and bootstrapping trust among mutually distrustful actors. How do we perform this by leveraging programmable networks?

Questions:

1. **How do you reason about packets leaving the network? What's the relevance with the Athens Affair? Why does programmability have to do anything with attestation?**
   In the paper we have a section we do have a discussion on this and we feel that it is very hard to address this. Analogous to the Athens Affair, using remote attestation we have a way of noticing that the software on the device is not the same as the one which was supposed to be running.
2. **What about the usability of these techniques? Linear Algebra is hard and not all users can manage to use it.**
   Writing a policy is hard and users aren't very good at that. A could work where a government entity or some NGO could write policies which the normal users can use then but yes it does require expert input.

Network Can Check Itself: Scaling Data Plane Checking via Distributed, On-Device Verification

Qiao Xiang, Ridi Wen, Chenyang Huang, Yuxin Wang (Xiamen University); ***Franck Le*** *(IBM Research)*

Summary:

Centralized approaches to data plane verification have multiple issues like bottlenecks and single point of failure. Using on-device abilities, data plane verification can be done in a distributed way.

Questions:

1. **How hard is it to support multipath?**
   As long as it can be expressed in regex, it can be supported
2. **What is the motivation for distributing the verification task? Verifier is not a user-facing component so how is it being a single point of failure an issue?**

It can be a performance bottleneck e.g. on some updates but we want the users to be able to verify fast.

Full-Stack SDN
Debnil Sur *(VMware); Ben Pfaff, Leonid Ryzhyk, Mihai Budiu (VMware Research)*

Paper argues for tighter binding between management, control and data plane in SDNs. This helps with scalability.

Questions:

1. **Limitations of expressiveness of modeling control plane**
   If you have parts of the control plane that are not incremental (?) it would not help.
2. **How did you do atomic matching of DB and HW?**
   Knowing your APIs well and the job of the sysadmin. We are still in the process of making the system and being able to quantify these factors.

It Takes Two to Tango: Cooperative Edge-to-Edge Routing
Henry Birge-Lee, *Maria Apostolaki, Jennifer Rexford (Princeton University)*

BGP does not expose all paths whereas path diversity can be helpful for real time applications. Moreover, conventional switches are not super useful for performance measurements to make choices about available paths (if exposed by BGP?).
BGP is not just a one path per destination but instead a one path pre prefix protocol. Tight collaboration between edge networks and use of programmable switches to address these issues can allow us to expose and explore different paths possible.

Questions:

1. **You can also do fault localization and demography through your technique.**
   Useful direction.
2. **Should we be trying to trick BGP or should we use something like SCION?**
   A Subset of SCION. This work has a plus point of quick deployability because we have BGP today.

End of Session Discussion

1. **How do you measure one way delay? Clock synchronization?**
   We can measure relatively accurate delay if we know clock offset and assume that clock offset does not shift at long enough timescales. Though it is true that this work does not completely address the problem of clock synchronization.

2. **In the past multipath selection could give us ms of improvement, but today things are really that bad today that this will be helpful since flattening of internet topology is already giving us good enough performance. What's the ultimate tangible benefit you can get from this?**
   Firstly there is an availability angle, e.g if a path appears to be hijacked in BGP, you can take another path. Moreover, inter domain routing does an atrocious jobs at picking up the right paths. Ingress, outgress paths don't really take too much performance into paths. The idea of moving paths closer to the edge is necessarily gonna get us good paths isn't something the author agrees with. There could still be bad paths.
3. **How do you measure the one way delay between 2 APs?**
   We do not do clock sync. Largely we have strictly 2 agents and there is a clock offset which is fixed so we do some level of approximation.
4. **If everyone uses Tango, what happens to convergence and scalability if everyone is picking from multipath?**
   Since we are on a smaller scale, scalability is not a concern.
5. **Most of the real time app optimizations are in the transport/ app layer. Are there BGP related challenges if you deploy on a large scale.**
   Since we are hacking BGP using community labels, it’s not a concern. If ISPs are unhappy that we are using a path, they can degrade the performance so we do not pick them. Usually ISPs would be happy if traffic is routed through them though.
6. **If each of you were to imagine another paper being deployed what would happen to your paper?**
   Verification and full stack SDN is complementary to Remote Attestation.
   Distributed verification can help catch bugs for full stack SDN paper, remote attestation is also very complimentary.
   Even if you don't believe in p4, it allows you to define the semantics for what each of the planes is doing, the planes become more amenable to attestation/verification.
7. **If the topology changes, how would you handle that?**
   We address in paper when we do not have full visibility of path, we are still able to operate.
   The network has dynamic states and some of these proposals suggest
   Even if you don't believe in p4, it allows you to define the semantics for what each of the planes is doing, the planes become more amenable to attestation/verification.
8. **For Tango, if you serve at scale, multiple servers/edge devices at a single edge?**
   We don't think this will be a big problem since we assume that Tango could be deployed using programmable networks. They are completely independent. Each of them is actually connected through different hops and you could make them coordinate.
9. **For Full Stack SDN: Comments on availability of runtime programmable switches to identify the incremental part of the program. Can they be combined together? When you want to dynamically change the program running on switches.**
   To start with, the system compiles all 3 together and runs them. If you change the dataplane program, you will have to recompile, and then re-run them. So more work is needed in this space to address this problem.

10. **For Tango: Clouds have a centralized managed RAN which they can manage and optimize it alot. Do you think you can match that performance with an edge based solution?**
    Our goal is not to compete. There are networks that are not as well served and there are enterprises which do not have edge network resources. We just want to provide an alternative where small enterprises are able to compete with the big players.
11. **For Tango: we need programmable switches, is there a necessity?**
    Not a necessity but rather something which makes sense as a solution. Based on the experiments we ran, we saw up to 30% improvement and if we want traffic to scale up to terabytes, it seems a viable solution.