Paper Critique RON

Kevin Cho
Peter Szolovits 1PM

April 2016

I Introduction

The Resilient Overlay Network (RON) is a routing system that is fully connected to each and every node through a virtual link. Through path redundancy, RON is designed to solve the problems wide-area routing protocols such as BGP-4 have. Problems include recovering from path outages and vulnerability to faults. By trading-off scalability for reliability, RON serves to provide an alternative system through three main design goals.

The design goals are (1) failure detection and recovery in less than 20 seconds; (2) tighter integration of routing and path selection with the application; and (3) expressive policy routing.(Section 3.) The first requires RON to be fault-toleratn and resilient. The second requires RON to have a better performance through different trade-offs. The last requires RON to have flexibility.

II Design Goals of RON

Fault Tolerance. The biggest problem with wide-area routing protocols such as BGP-4 was that they were not very fault tolerant. As a result, the Internet was vulnerable to several errors (Sections 1). RON is able to detect errors such as packet floods and congestion that BGP could not. BGP's mechanism has packets run through pathways as long as they were deemed "live." This would cause congested paths to become even more congested (Section 3.1).

RON is able to find failures by aggressively probing and monitoring the virtual paths on each nodes. Through this probing, RON finds an adequate throughput path or forwards the packet to other RON nodes. Thus, in theory, if a error was found in a path to a certain node, RON is able to "hop over" by going through an intermediate node.

Resilience. Not only does RON have a fault-tolerant system, but also RON strives to achieve a resilient system. In other words, RON is designed to recover from an error in a matter of seconds (Section 3.1). This is a significant change from the previous system of BGP-4,

which took several minutes. RON achieves this goal is through its policy routing and virtual links.

Policy routing gives each node more options to go. In addition, each node is connected to every other through a virtual link. Thus, even if one pathway has an error, the node can "hop" to another node and reach the destination. In the evaluations of the paper (Section 6.3), the authors tell the audience that the single-hop method worked well in avoiding the paths with problems or errors. However, by connecting each node, RON trades off scalability with reliability.

Performance. Ron was created to choose a path that does not have low-throughput (Section 4.2.2). This means that a path chosen by RON is not necessarily the best path. By having a tighter integration with applications (Section 3.2), RON trades off certain performance criteria with others to find good alternative paths. For instance, a path that has the best throughput may have an unacceptable latency and, thus, cannot be used. These paths may not always be the best optimal path, but it will be a path that does not have a low-throughput.

RON achieves this path selection through a variation of steps. A path deemed "dead" through an outage will not be used. After considering "dead" paths, RON will provide a score to each path through a formula to avoid paths with high packet loss and long latency. The formula uses Ron's rtt (end-to-end round trip time) and one-way packet loss probability (Section 4.2.2).

Flexibility and simplicity. RON uses a flexible policy mechanisms that helps the system implement better network policies than previous versions (Section 8.0). Another aspect of flexibility ties in with the goal of simplicity. These two goals meet when clients are able to define their own membership mechanism (Section 4.5). The system is simple and flexible in that clients design the mechanism the way they want and that it chooses from several different paths.

III Analysis of the System

RON is a great solution to the problems that wide-area routing have. However, it is not necessarily an overall improvement from routing protocols such as BGP. RON does improve the fault-tolerance and resilience to take only seconds (Section 6.2). What the authors do not emphasize is that the reason for such an improvement is due to the large number of connections in nodes.

The biggest limitation to RON would be the scalability. The authors of the paper did evaluate RON with 50 nodes. However, it is implicitly stated that going above this would be impossible due to the sheer number of connections. RON trades off scalability for a more reliable system. Without the ability to scale to large numbers, RON provides a much more reliable and faster system for smaller areas.

Another limitation is security. As of now, users that violate the network policy have to be

dealt in person (Section 7). Thus, if a user wanted, he/she would be able to use the system with malicious intent. The authors do shake this limitation off by saying that these problems can be solved at the administrative level, but it is still lingering possibility.

Despite the limitations, RON can be useful in various cases. One particular case would be for a company that is dispersed around the world. Because there is a small number of nodes, these scattered location can benefit more with RON. Faster communication can be present and there would be less errors in connection. RON provides this case with a much more reliable, resilient, and safer system.

IV Conclusion

The paper about RON has validity to its claims. RON is resilient and fault tolerant in that it is able to detect and recover from faults and errors within seconds. This is a huge improvement from previous systems such as the BGP-4. RON also has better performance than previous systems. However, all these enhancements do come with a price. RON cannot scale to large areas like BGP-4 due to the vass number of connections. In short, RON is a resilient system that performs better and is more reliable in small use cases.

Works Cited

Andersen, David, Balakrishnan, Hari, Kaashoek, Frans, and Morris, Robert (MIT). "Resilient Overlay Netowrk" MIT Laboratory for Computer Science. ACM Symp. on Operating Systems Principles (SOSP) October 2001.