SDN Based Congestion-Aware Routing for Wide-Area Sensor Networks

Korben DiArchangel, Jared Nelson, Bibek Shrestha, Engin Arslan
Department of Computer Science & Engineering
University of Nevada Reno
{kvdiarchangel, jaredLnelson1, bibek.shrestha}@nevada.unr.edu, earslan@unr.edu

Abstract—The adoption of the Internet of Things (IoT) phenomenon across many science and engineering applications has led to the wide-area deployment of sensor devices to gather and process large volumes of data. This data often needs to be moved to processing centers in a timely manner for (near) real-time analysis and archival. Current network management practices rely on static routing technologies, such as Open Shortest Path First (OSPF) and Multiprotocol Label Switching (MPLS), that are manually configured by experienced network administrators across the physical and virtual networks. This rigid network infrastructure is however a major hurdle in the effective use of available network resources despite degrading quality of service metrics due to dynamic nature of network congestion. In this paper, we present a practical network monitoring and management framework to detect congestion events and use Software-Defined Networking (SDN) to make real-time routing adjustments to provide quality of service for science flows with stringent networking needs.

I. Introduction

Recent advancements in sensing and wireless technologies led to the increased adoption of advanced IoT devices in wide-area settings such as environmental monitoring, public safety systems, and industrial management. In these hybrid networks, there are often multiple routes that a network flow can take from its source to the destination. However, current practice to control network routes involves manual link weight adjustments for OSPF protocol. This manual process limits routing policy updates to be done only when major events happen such as long-term link failures or network expansion, leading to suboptimal network utilization in the presence of network congestion events despite the existence of available bandwidth in alternate routes and backup paths.

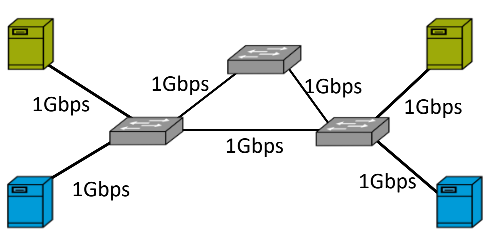


Figure 1. An illustration of a backup path between source and destination

Figure 1 illustrates a network topology in which there are two routes between data sources and destinations. While main route is used as a default route for both flows, the alternate route is activated only when the link between the switches fails. This in turn leads to suboptimal network resource utilization when both data sources want to transfer data at full speed. Figure 2 demonstrates this phenomenon as the start of the second flow nearly halves the throughput of the first even if backup route has available resources. As the high quality IoT sensors are deployed to wide-area networks and demand reliable network infrastructure to move data, there is a need for an automated network management approach to make abundant bandwidth available for network-intensive flows for improved resource utilization and enhanced QoS.

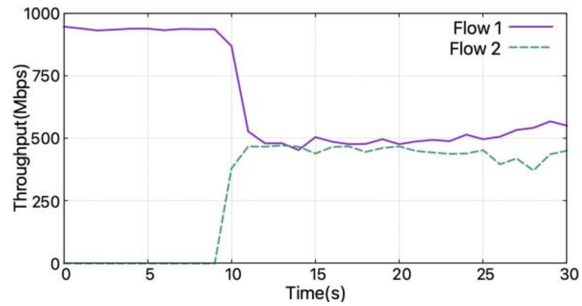


Figure 2. Failure to utilize available network resources effectively leads to poor QoS for network flows

Software Defined Networking (SDN) decouples the control and data plane to allow network administrators to automatically and dynamically manage and control network devices, services, topology, flows, QoS, and policies using high level programming languages. While SDN has been applied to different scenarios, this project makes a first attempt to integrate it to fine granular network monitoring architecture to capture congestion events timely and accurately. Specifically, we enable port mirroring on switches to obtain precise network usage information and use this information to adjust routing policies to increase resource utilization and meet QoS demands of sensor applications.

II. Proposed Solution

We introduce an integrated network monitoring and management framework to detect network congestion and update routing policies to improve network utilization. Figure 3 illustrates the proposed framework. Each switch is connected to a *Monitoring Server* which processes mirrored traffic from the switches. Although it is possible to capture some high level flow information from switches using OpenFlow, we adopt port mirroring based monitoring strategy for three major reasons. First, we can extract multitude of flow information

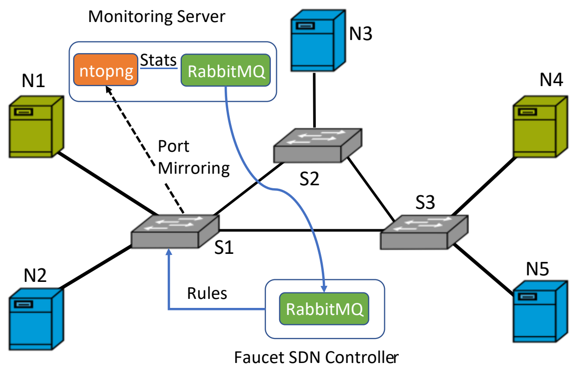


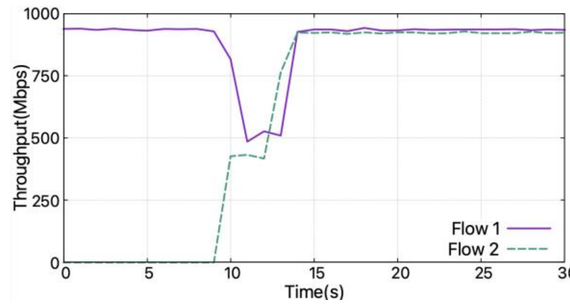
Figure 3. Proposed integrated monitoring and management architecture for congestion-aware path selection framework.

such as layer-7 dissection which is not readily available from SDN based monitoring using freely available tools such as ntopng [1]. Second, port mirroring makes it possible to fine granular packet level information that is otherwise not possible through statistics provided by OpenFlow. Third, while it may not be possible to replace every switch in the network with SDN-capable one, the information gathered by non-SDN switches can be used to adjust the routing for SDN switches.

Traffic statistics captured by *ntopng* is sent to Faucet SDN Controller [2] using RabbitMQ message broker. By default, the traffic statistics reported to the controller once in every five seconds. The controller then processes network statistics from each switch in the network to gain insights into capacity and utilization of each link as well as the performance of active flows. This information is finally used to make routing decision to fully benefit from available resources while maintaining QoS demands of critical flows.

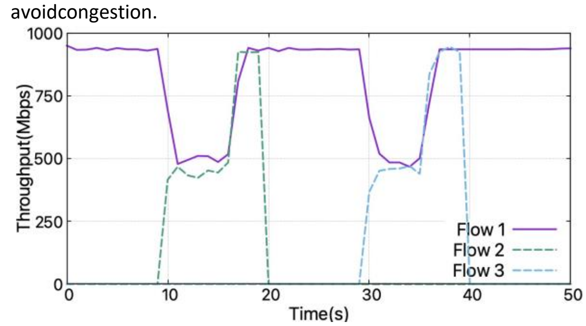
III. Preliminary Results

We conducted two experiments to demonstrate the effectiveness of proposed solution to detect network congestion and adjust routing. In the first one, the first flow is started between N1 and N4 nodes which travels through switches S1 and S3 based on default route selection algorithm which prefers shortest paths. After few seconds, we initiate the second TCP flow between N2 and N5 nodes that travels through S1 and S3 switches. Since the second flow interferes with the first one and alternate path over switch S2 has free resources, the path selection strategy is expected to divert one of the flows over the alternate path (S1-S2-S3) to increase network utilization. Figure 4(a) shows the throughput of each flow when the first experiment is executed. We can see that when the second flow is started at around $9s$ second, the throughput of first flow decreases to 500 Mbps. However, the controller detects that the link between S1-S3 is shared by two flows whereas the alternate path is not utilized at all. Thus, it diverts the first flow to take the alternate path thereby leading to increase in throughput for both flows.



The second experiments is similar to the first one except we terminate the second flow at around $20s$ and start a third flow from N3 to N5 at around $30s$. Since the controller diverts

(a) SDN Controller diverts Flow 1 to use the alternate path to avoidcongestion.



(b) SDN controller adjusts the path of the first flow twice (at$9s$ and $36s$) to mitigate network congestion and increase network utilization.

Figure 4. Controller running path selection strategy selects optimal path according to the overall network status.

the first flow to the alternate path to avoid the congestion at around $9s$, the first flow started using S1-S2-S3 route. When the third flow starts, however, the link between S2 and S3 becomes the bottleneck. Since we do port mirroring from each switch in the network, the controller can notice the congestion between S2-S3 and diverts the first flow to its original path by adjusting routing policies on S1 to take advantage of available resources. As can be seen in Figure 4, the throughput of the first flow declines when the third flow start but it recovers to the full speed upon the execution of routing adjustments by the controller. This experiment demonstrates that port mirroring can be used to detect congestion events even if some of the network equipment does not support OpenFlow. Specifically, the controller detects the congestion on S2 (which does not have to be SDN capable) through port mirroring but configures S1 (SDN capable) to divert the first flow.

IV. Conclusion and Future Work

Optimal path selection is critical to properly balance the available bandwidth among all the flows in the network. This work introduces congestion-aware routing for wide-area of sensor networks with the help of port mirroring and SDN. Port mirroring helps to gather accurate network usage statistics from network equipment which is then fed to SDN controller to adjust routing policies to maximize the utilization of network resources. The preliminary results obtained from realworld deployment of the proposed architecture show that the proposed framework leads to significant increase in resource utilization and increased QoS for performance-sensitive flows.

REFERENCES

[1] “ntopng,” https://www.ntop.org/products/traffic-analysis/ntop/, 2021.
[2] “Faucet,” https://faucet.nz/, 2021.