Self-configurable and Scalable Utility Communications Enabled by Software-Defined Networks

Young-Jin Kim
Bell-Labs, Alcatel-Lucent
young.jin_kim@alcatel-lucent.com

Keqiang He
University of Wisconsin
keqhe@cs.wisc.edu

Marina Thottan
Bell-Labs, Alcatel-Lucent
marina.thottan@alcatel-lucent.com

Jayant G. Deshpande
Bell Labs, Alcatel-Lucent
jayant.deshpande@alcatel-lucent.com

ABSTRACT
Utility communications are increasingly required to support machine-to-machine communications for thousands to millions of end devices ranging from meters and PMUs to tiny sensors and electric vehicles. The Software Defined Network (SDN) concept provides inherent features to support in a scalable and self-configurable manner the deployment and management of existing and envisioned utility end devices and applications. Using the SDN technology, we can create dynamically adaptable virtual network slices to cost-effectively and securely meet the utility communication needs. The programmability of SDN allows the elastic, fast, and scalable deployment of present and future utility applications with varying requirements on security and time criticality. In this work, we design a SDN-enabled utility communication architecture to support scalable deployment of applications that leverage many utility end devices. The feasibility of the architecture over an SDN network is discussed.

Categories and Subject Descriptors
C.2.1 [**Network Architecture and Design**]: Communications

General Terms
Management, Performance, Design, Experimentation

Keywords
Machine-to-Machine (M2M), Self-configurability, Scalability

1. INTRODUCTION
With Smart Grid roll-out, M2M communication networks supporting electric utility applications traffic is undergoing a tremendous change both in the increasing number of new grid applications, and a massive number of communication endpoints that the network must support [1]. Most of this increase in endpoints comes from deployment of sensors, currently limited to a few hundred Remote Terminal Units (RTUs), to thousands to several million sensors including Intelligent Electronic Devices (IEDs), Phasor Measurement Units (PMUs), smart meters, and sensors attached to Distributed Energy Sources (DERs) and Electric vehicles (EVs). In addition the new applications require self-configurable M2M communication networks that can adaptively and scalably meet the needs for performance, reliability, and security.

In this work, we design a new SDN-enabled M2M network architecture. Our M2M network architecture will not only provide a cost-effective and self-configurable network solution on commoditized SDN switches as network elements, but also has the ability to elastically define virtual network slices with each slice supporting an application (in one utility or across multiple utilities), or a group of similar applications.

e-Energy'14, June 11–13, 2014, Cambridge, UK.
ACM 978-1-4503-2819-7/14/06.
http://dx.doi.org/10.1145/2602044.2602074

Today's M2M Communication Networks

Today we could use the industry standard (i.e., IEEE 802.1Q Virtual LAN [2]) for virtual networking as a M2M communication solution that can accommodate grid applications. However, consider the following deployment scenario of a million scale communication-enabled measurement and monitoring end devices; a relatively-small number (e.g., 100~1000) of network switches; thus, a physical port (called port) in a switch must be logically (not physically) connected to more than one end device (i.e., multiple meters per port via a data concentrator); also, a device must subscribe to more than one VLAN. Unfortunately, the VLAN standard, IEEE 802.1Q, cannot scalably support the scenario due to: the small number of VLANs per-port (Port-based VLANs or Protocol-based VLANs). In the port-based VLAN, an access port between a switch and access devices (not a trunk port between switches) are assigned to a VLAN during a certain time period. In the protocol-based VLAN, one VLAN per protocol is supported. As a result, an access port must be concurrently used by multiple VLANs and only a small number of well-known protocols (i.e., IP, ARP, IPX) are supported. In addition from a security perspective in an IEEE 802.1Q network, all members that are authenticated can directly communicate with each other. As a result, compromising a device (such as computer malwares) can result in the propagation of security threats across the network..

2. BENEFITS OF SDN FOR M2M COMMUNICATIONS
All SDN can provide isolation of different traffic types, applications, and/or endpoint classification. E.g., virtual network slices may be defined for AMI, SCADA, DG/DS/EV, and PMU traffic. Network slices may also be based on geographical or domain considerations (transmission and distribution or security zones). The virtual network slices inherently enhance security with traffic isolation and enabling security, quality of service (QoS), and even network management policies for each network slice. So, a closed group of applications/application type/endpoint group can have its "own virtual network" that is its network slice. Note that the ability to rapidly create required functions with few changes in the physical network makes the network less vulnerable to potential network failures.

Our architecture design offers a programmable open interface to the applications as well as to the network elements for their control, configuration, and management. There is a deliberate shift from fixed network functions serving many applications to per application virtualized functions making introduction of new applications as well as connecting new endpoints in the network more efficient and manageable. The ability to reconfigure a

software defined network and rapidly deploy virtualized network functions allows for greater network utilization, global resource optimization, and enhanced scaling. Thus demand driven service and device activation and provisioning will directly lead to dynamic application velocity and scale.

3. SDN-enabled Virtual Utility Network

Our design has been inspired by the SDN concept and the publisher-subscribe (Pub-Sub) paradigm [3]. Fig. 1 represents the overall idea of the SDN-enabled Virtual Utility Network (SVUN) that consists of SDN switches (i.e., OpenFlow switches [4]), M2M clients (publishers or subscribers), and M2M control nodes. The M2M control plane consisting of M2M group manager, authenticator, network manager, and SDN controller provides dynamic and fine-grained membership management and authentication measures for establishing secure and QoS-aware M2M communications (i.e., VLAN per group). Please refer to [5] for details. We emphasize that our unique contribution against other SDN work achieves the complete automation of communication/security configuration by combining M2M group mangers with a SDN controller. Compared to other Pub-Sub work, our approach has the following distinct features: 1) line-speed packet processing and forwarding, 2) per-group VLAN traffic isolation (i.e., VLAN per group), 3) per-group QoS management (i.e., delay-sensitive), and 4) traffic-flow monitoring for load balancing and fail-over.

SVUN' key notions: (1) L4 flow match for access ports, (2) VLAN identifier tagging/stripping for trunk ports, and (3) Pub-Sub communication notion.

Scalability: One important thing is that the SVUN addresses the scalability issue as data from multiple VLANs can concurrently traverse over both access ports and trunk ports. The memory (i.e., TCAM) of SDN switches where flow entries are kept is a major resource since most SDN switches available in today's market have small-size flow tables (i.e., less than 4K flow entries). The scale of flows for M2M data traffic in an SVUN is dependent on the number K of application groups and the number M of group participants. However, the scaling impact of the number of group participants is bounded by the number of SDN switches N due to the effect of VLAN aggregation and multicast in SDN switches. The maximum number of flow entries per SDN switch is O(K) and K is independent of N and also typically smaller than M. This scaling is a unique characteristic of the SVUN.

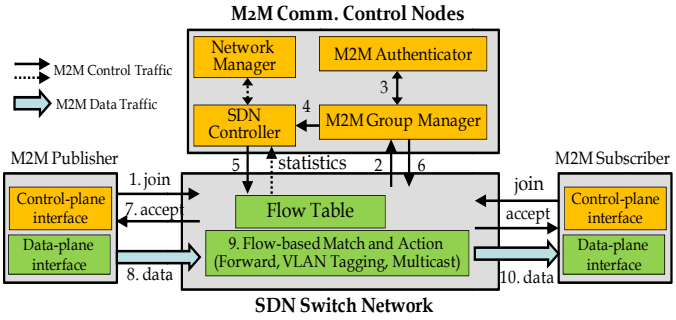

Fig. 1 Schema of an Instance of Our SVUN Architecture

4. FEASIBLITY STUDIES

We discuss a real implementation of SVUN written in Python and C++ and tested in our Lab test bed. In the implementation, an M2M group manager requests a SDN controller of installing VLAN flow-rules and also let M2M publishers know where publishing data is about to be sent. In the Lab (See Fig 2), we measure three metrics: 1) flow-table occupancy per switch, 2) end-to-end delay of M2M data plane and 3) M2M control plane. The flow-table occupancy shows scalability of our SVUN since the TCAM is critical but has limited resources; the end-to-end delay of M2M data plane is the time difference between when an M2M publisher sends data and when an M2M subscriber receives the data. This metric corresponds to the forwarding performance of SDN switches in M2M data traffic. The end-to-end delay of M2M control plane is the time difference between when a M2M client (as either a M2M publisher or a M2M subscriber) sends a join message to its M2M group manager and when it receive an accept message from its M2M group manager.

Flow Table Occupancy: the maximum number of hard-state flow rules for M2M data traffic is only two irrespective of the number of M2M devices. There exist a small number of soft-state flow rules (deleted after a timer is expired) for connectivity between devices and an M2M group manager.

Delay on aspects of M2M data plane: In principle, once VLAN flow rules for M2M data-plane have been installed, we see line-speed packet lookup and forwarding of TCAM. In the implementation of SVUN, we observed that the end-to-end delay from publishers to subscribers is never more than 150 ns irrespective of the size of data.

Delay on aspects of M2M control plane: M2M control traffic delay is either about 30ms or about 90ms. Compared with M2M data traffic delay, it is fairly high, even though it is tolerable. We observed the following delay sources: 1) flooding-based ARP discovery, 2) TCP connection setup between M2M group manager and M2M clients, 3) A VLAN flow setup for data-plane.

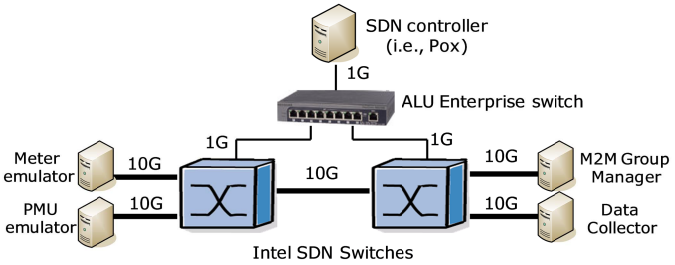

Fig. 2 Our Lab test-bed with two SDN switches

5. REFERENCES

[1] Budka, K., Deshpande J., and Thottan, M, Communication Networks for Smart Grids – Making Smart Grid Real, Springer, 2014.

[2] IEEE Std. 802.1Q-2011, Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks.

[3] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, The Many Faces of Publish/Subscribe, ACM Computing Surveys, vol. 35, no. 2, June 2003.

[4] OpenFlow Switch Specification Version. 1.0.0, Dec. 2013.

[5] Y-J. Kim, J. Lee, G. Atkinson, H. Kim and M. Thottan, SeDAX: A secure, resilient and scalable platform, IEEE JSAC, vol. 30, no. 6, July 2012.