Network Functions Virtualization in Home Networks

Marion Dillon
Timothy Winters

Abstract

The current model of home networking includes relatively low-cost, failure-prone devices, requiring frequent intervention by largely non-technical users. This model is improved through the use of Network Functions Virtualization (NFV). Current home network gateways are becoming more sophisticated to support the many different network operator and user services. Software Defined Networking (SDN) and NFV offer a solution to the problem of the complex home gateway: Move the gateway to the cloud. We decided to create an SDN enabled home network gateway using currently available solutions to demonstrate NFV. We document the challenges and successes encountered during this project.

Keywords

Emerging technologies, Network topology, Local-Area Networks

Introduction

Home gateways today must support relatively complex applications on devices that must remain inexpensive for the technology to become ubiquitous in the home network. Therefore, the complexity of applications and hardware on a home gateway device is limited. The home gateway must have the ability to handle each type of traffic it sees to the home. The growing trend of Machine-to-Machine (M2M) technology requiring constant connection with a gateway or server, combined with proprietary communications protocols, means the current home gateway paradigm cannot be sustained. Home gateways must be up-to-date to support new protocols, or each different M2M service must have a separate gateway in the home. The home user may want to have a choice in service without changing hardware. This factored in with the constant innovation home gateways hardware will become obsolete far too quickly for the home user to keep up.

A virtual gateway as a service is adaptable to new technologies and protocols, and the service provider benefits from direct access, for management and customer service improvement. Network Functions Virtualization (NFV) can be leveraged to relocate some of the home gateway functionality into the virtual gateway giving both the user and service operator flexibility. For this project we use available off-

the- shelf solutions to place the home gateway services at various locations within the Provider Network and explore the testing methodology for verifying the Software Defined Network (SDN) solution.

SDN Solutions

A traditional Home Gateway device is used by service providers to deploy network services. The gateways provide functions such as security, network, and management to the user; including features such as firewalls, DHCP, and CWMP. An SDN approach can be taken using the OpenFlow protocol to convert home gateways to OpenFlow switches. This allows network operators to control flows through the entire network. The current home networking solution connects devices on its LAN interfaces and routes traffic to the WAN interface and ultimately the service provider's network and the Internet. Bridged Residential Gateways are devices in the home, which can communicate with a Virtual Gateway located on the WAN, and which has some subset of home gateway functionality. A vG is a logical service with a subset of home gateway functionality, located at one or more nodes in the service provider's access network, bridged network gateway, or cloud (Figure 1).

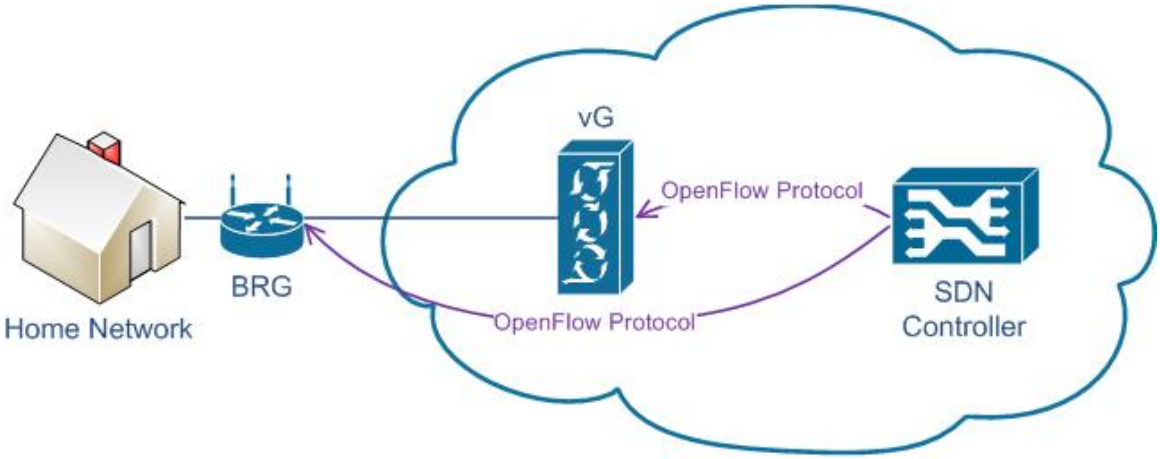

Figure 1 – Network Topology of a Virtual Gateway Scenario

How it works

The principal feature of Software Defined Networks is the separation, or removal, of the control plane from the data plane. Distributing the control plane and data plane functions allows networks to be more dynamic and intelligent – a centralized controller can make changes to all switches and routers while surveying the whole topology. In this use case, the SDN controller colors traffic from a home network to direct it to its virtual home gateway, located somewhere in the service provider's access network or cloud. The Bridged Residential Gateway need not be aware of the location of the virtual gateway, as long as it is located somewhere on the SDN.

SDN Prerequisites

There is an inherent bootstrapping problem with OpenFlow-controlled networks. There must be a working IP network among all switches, and the flows defined by

the controller must either be salient to this network, or completely separate. This management layer limits an SDN-controlled network, since a centralized controller cannot define the entire network.

Service providers already have an IP network setup to the home, since they are already setup to provide Internet. For this reason, the need for a working IP network is not prohibitive in implementing the virtual gateway scenario on top of existing infrastructure.

Use Cases

Quality of Service

The programmable Quality of Service that SDN provides opens the possibility of dynamic QoS that can be applied to many user applications. A QoS use case is defining service levels to different types of traffic over all devices in the home network. Using SDN, the same QoS rules must be programmed into both the vG and BRG. To confirm QoS is functioning properly a traffic flow must be selected to have higher priority, for example, a video stream. Network traffic is created to extend the BRG forwarding ability, so that BRG must forward the higher priority flows to the vG.

Failover

Using SDNs in a virtual home gateway scenario means that the location of the virtual service is immaterial to its connectivity with the home. Flows can be reprogrammed to redirect home network traffic to a new location, for use in failover situations or for maintenance and upgrades. If this service is stateful, such as a firewall or NAT, the state may be transferred as well. To validate failover, two matching vGs are created in different network locations, and using the SDN the flow is transferred from the BRG to the redundant virtual gateway while maintaining the correct services. The transfer should be tried using several different stimuli, such as power failure, network failure, or virtual machine misconfiguration.

M2M

As the Internet of Things becomes more of a reality, each low-power device needs a way to connect to the home gateway, and any web application and user accounts that may exist. Current home gateways are not equipped to receive and transmit traffic to these low-power devices. Using SDN capabilities, flows concerning M2M traffic leave the home to be directed to the correct Service Provider. For instance, if there is a smart meter that measures electricity use, the traffic reporting the usage can be directed at the electricity provider's cloud solution. This is an improvement on current M2M solutions, many of which require communication external to the home network, such as radio or cellular networks. By integrating the smart meter into the home network, there is new opportunity to widen its use. For instance, the smart meter, on the home network, could monitor the electricity usage of each smart appliance, for better reporting to the home user. This would also provide the

ability for energy-conserving applications to shut down or cap certain appliances in the home network. The most efficient way to test M2M applications is to make sure it at least has parity with a typical home deployment today. There will be some expected issues, latency that will need to be explored to see if the application is adversely effected by the moving the gateway into the cloud.

An Out-of-the-Box Solution

Adding OpenFlow support to open source home router software OpenWRT works well in theory, and even in practice, for an inexpensive, OpenFlow-only switch. The trouble comes when you want to use the OpenWRT forwarding software in tandem with OpenFlow. There is no mechanism, once a packet has gone through the OpenFlow flow tables and into the OpenWRT software, for it to go back through the same flow tables.

To harness the open source home router software in OpenWRT, the flow action LOCAL was used to redirect traffic to the local networking stack. This worked well in order to perform some actions through OpenFlow (such as stripping a VLAN header) and the rest of the actions through OpenWRT (such as processing DHCP traffic through the local DHCP server). When the packet cannot go back through the OpenFlow tables, it cannot get re-tagged in order for it to pass through the OpenFlow-controlled tunnel back down to the BRG (Figure 2).

This issue prompted a different solution. It is possible to integrate OpenFlow more completely with OpenWRT, but requires significant alterations to the OpenFlow plugin. An out-of-the box solution would require two autonomous pieces of software on a virtual network: Open vSwitch and virtual home gateway software. Open vSwitch is a virtual switch implementation, which includes an OpenFlow implementation. We placed this switch on the same virtual network as the virtual OpenWRT home gateway (Figure 2). Thus, we were able to obtain the two-way OpenFlow table processing needed for an OpenFlow-controlled tunnel to the BRG. Since the BRG need not be anything but an OpenFlow switch, a virtual machine running Open vSwitch sufficed. This software could be run on any hardware, including inexpensive chipsets such as the Raspberry Pi or Intel Galileo board.

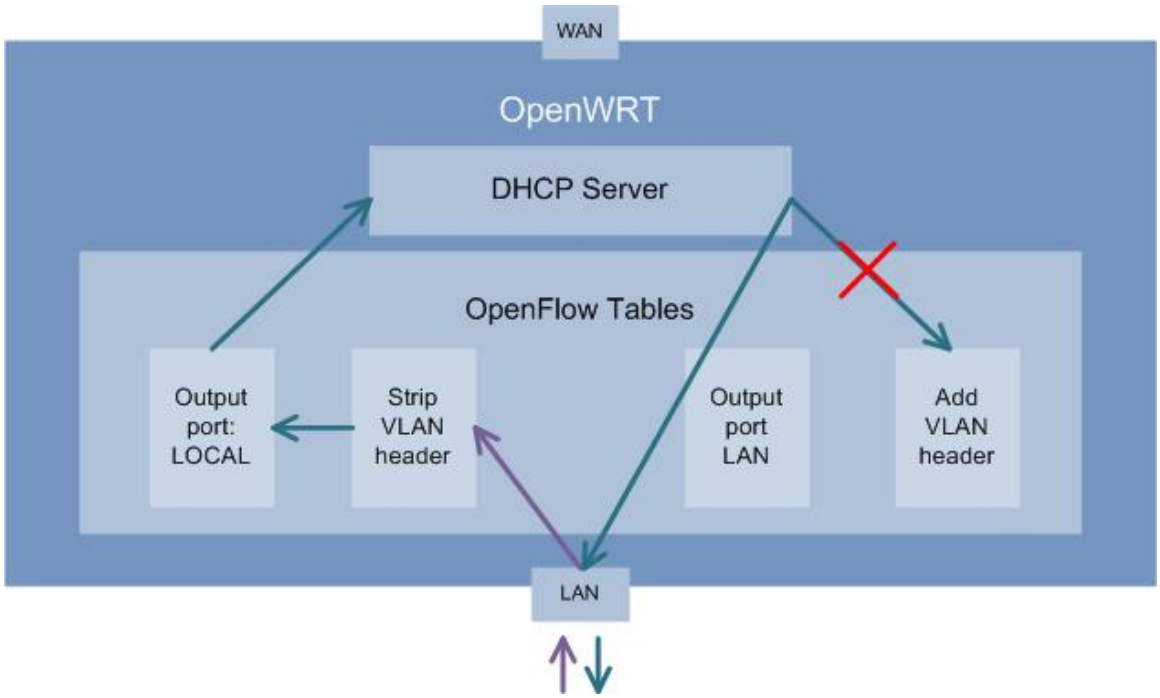

Figure 2a - Separating the OpenFlow switch from the gateway software solves VLAN tunneling issues. Here, the internal DHCP server is unable to send traffic back through the OpenFlow tables to get tagged.

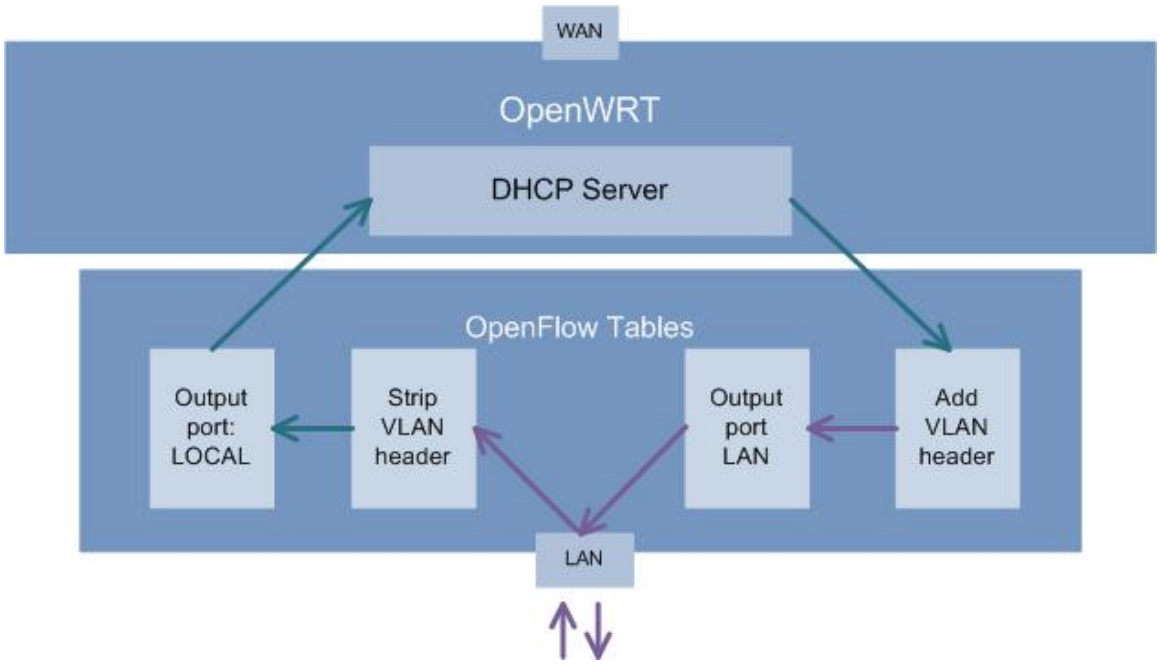

Figure 2b - Here, the problem is solved when the OpenFlow switch and the home gateway software are autonomous systems.

Conclusion

Each use case needs to be rigorously tested in order to determine the effects and implications of a relocated gateway. The location of the virtual gateway along with

the specific use case will affect network traffic. The aim of the implementation should be to minimize the adverse effects, and maximize improvement.

A smooth, out-of-the-box solution does not really exist for this problem. By piecing together assorted open-source software it is possible to emulate a rudimentary vG/BRG scenario, but there is work needed in order make the software cohesive. The gateway software and the OpenFlow software in the Virtual Gateway must be combined in a way that allows internal upstream and downstream OpenFlow table processing of all traffic. The Controller needs to make intelligent decisions based on the network to control flows to the vGs, and to alter them when needed. It needs to enable the BRG to route LAN sourced and destined traffic within the LAN, without sending it to the Virtual Gateway.

Using the granularity offered by the OpenFlow protocol, there is no need to color flows based on home network, but based on service. For example, a home with a security system could use this coloring to direct its security traffic to the security provider, rather than the Internet service provider. The virtualized nature of this solution implies room for expansion of the model and use cases.

Using Software Defined Networking to virtualize the home gateway offers users and providers versatility and room for growth. Home networking is changing rapidly. With the adoption of IPv6 and discussions of introducing routing in the home, the gateway will become more complex and important than ever. For this reason, it is important to provide that gateway with enough resources to perform the tasks it does now and all that it will do in the future. Virtualization is the answer to this problem.

Marion Dillon works at the University of New Hampshire InterOperability Laboratory. Dillon received a Bachelor of Science in Applied Mathematics: Computation from the University of New Hampshire. Contact her at mdillon@iol.unh.edu.

Timothy Winters is a technical lead at the University of New Hampshire InterOperability Laboratory. Winters received a Bachelor of Science in Computer Science from the University of New Hampshire. Contact him at twinters@iol.unh.edu.

Addresses:

Marion Dillon
121 Technology Drive
Durham, NH 03824
(603) 862-2911

Timothy Winters
121 Technology Drive
Durham, NH 03824
(603) 862-3332