Virtual Public Networks

Arjuna Sathiaseelan[1], Charalampos Rotsos[1], Sriram C S[2], Dirk Trossen[1], Panagiotis Papadimitriou[3], Jon Crowcroft[1]
[1]Computer Laboratory, University of Cambridge, UK
[2]Paxterra Solutions, USA
[3]Leibniz Universität Hannover, Germany
{arjuna.sathiaseelan, charalampos.rotsos, dirk.trossen, jon.crowcroft}@cl.cam.ac.uk
sriram@paxterrasolutions.com
panagiotis.papadimitriou@ikt.uni-hannover.de

ABSTRACT

Universal access to Internet is crucial. There have been several initiatives recently to enable wider access to the Internet. The Public Access WiFi Service (PAWS) is one such initiative that enables free Internet access to all and is based on Lowest Cost Denominator Networking (LCDNet) – a set of network techniques that enable users to share their home broadband network with the public. LCDNet makes use of the available unused capacity in home broadband networks and allows Less-than-Best Effort (LBE) access to these resources. LCDNet also enables third party stakeholders like local government to become virtual network operators, reducing the costs of network operators to setup and manage new infrastructures to extend access to their Internet backhaul. With the advent of software defined networking (SDN), there are more opportunities for network operators to create, deploy and manage in large scale such open home networks. In this paper, we present Virtual Public Networks (VPuN), home networks created, deployed and managed through an evolutionary SDN control abstraction. This offers more flexibility to users and network operators, allowing them to share and control the network, while providing opportunities for new stakeholders to emerge as virtual network operators.

1. Introduction

Internet access in the recent years has become an important resource for the global population. The Human Rights Council considers Internet access as an important enabler of human expression and a potential human right [3]. However the Internet is seriously “challenged” (geographically, socio-economically and technically) to ensure universal coverage [2].

Lowest Cost Denominator Networking (LCDNet) [2] introduces a novel network paradigm for global Internet access, by utilising unused network resources. LCDNet architects multi-layer resource pooling Internet technologies to support new low-cost access methods that could greatly reduce a network operator’s direct investment in local infrastructure to support wider Internet access.

A recent initiative to enable universal access to the Internet is Public Access WiFi Service (PAWS) [7]. PAWS is based on LCDNet that makes use of the available unused capacity in home broadband networks and allows Less-than-Best Effort (LBE) [2] access (lower quality compared to the standard Internet service offered to paying users) to these resources. PAWS adopts an approach of community-wide participation, where broadband customers are able to donate controlled but free use of their high-speed broadband Internet to fellow citizens.

Large-scale deployment and management of such open networks will impose several challenges in terms of scalability, security, accountability and performance to both network operators and users. This will in turn increase operating expenditures for network operators to manage such networks. In [2], we argued that the stakeholder value chain should be extended for incentivizing donated Internet access by including more than the two traditional parties (consumer and Internet Service Provider) [2]. Such addition of third parties (e.g., for local government or NGO) responsible for managing these networks can in turn reduce operating expenditures for network operators.

For users sharing their network, there will be major concerns with respect to security, performance, and network management. Home network are already complex to setup and manage. This is a major obstacle for realizing our vision of wider deployment of a service like PAWS.

With the advent of software defined networking (SDN), there are more opportunities for network operators to deploy and manage in large scale such open public wireless networks. SDN has enabled open and programmable networks by isolating the control plane of the network and providing abstractions in it.

In this position paper, we present an SDN enabled architecture for creating, deploying and managing Virtual Public Networks (VPuN). We lay out our architectural vision and further define the network design which includes defining the access point control abstractions that can be used by different stakeholders (users, network operators (NO) and third party virtual network operators

(VNO)) to provide a third party VNO federated Internet access as well as the ability to dynamically control resources.

2. Virtual Public Networks

In this section, we present Virtual Public Networks (VPuN): home networks created, deployed and managed through an evolutionary SDN control abstraction. VPuN are envisaged to achieve the following high-level objectives:

- Expose a dynamic and user-friendly abstraction to stakeholders at various levels of the network to specify network resource requirements.
- When explicit requirements are not specified or if external factors compel it, the network must be able to reprogram itself automatically. This will be useful in scenarios during emergency situations where access points can automatically mesh with access points of other users or personal devices based on online social networking trends (e.g., earthquake or tsunami in the area) or network operators can dynamically allocate capacity based on real-time needs (for e.g. to support sudden increase in traffic due to flash crowds).

To fulfil these high-level objectives, VPuN must:

- Provide the ability for stakeholders to specify their requirements and the parameters that affect these requirements in a simple manner.
- Provide the ability to translate these requirements into control flows to be installed in various components of the network.
- Aggregate and curate data from authorized social media and news feeds and form the network itself to ascertain status of the environment.
- Using the information gathered, make intelligent decisions to automatically reprogram the network.

In addition to these feature requirements, VPuN must fulfil the following design requirements so that it can be integrated seamlessly with any SDN stack.

- Provide backwards compatibility with existing network protocols.
- Provide ability to extend and support other SDN specifications that might evolve over and above OpenFlow.
- Provide transparent APIs and libraries that can be used to build external value added applications (e.g., a reward point tracker for capacity sharing).

VPuN require the following components:

- SDN enabled home routers that can be configured using open APIs (e.g., OpenFlow [5]).
- Controllers (e.g., POX, NOX [4]) that allow users to access and modify the flow table of a home router.
- A slicing layer (e.g., FlowVisor [6]) for the home router allowing both the home network user and the VNO to control their respective slices while enforcing isolation between these slices.
- A sharing policy expression language (SPEL) that allows each home network user to specify, amongst others, the amount of bandwidth and the period over which his network will be shared with VPuN clients.

2.1 Architectural Vision

Hereby, we present our vision of the VPuN architecture. As illustrated in Figure 1, VPuN architecture is distributed across home networks, the network operator (NO) and the virtual network operator (VNO). According to the VPuN requirements, an SDN-enabled home router, a controller, a Access Point Manager (APM) and a SDN slicing layer and run-time system for the translation of sharing policy expressions are deployed in each home network. The VNO relies on SDN principles to retain control of an isolated network slice used for VPuN. To this end, the VNO deploys the status and rule aggregator whose functionality is discussed below.

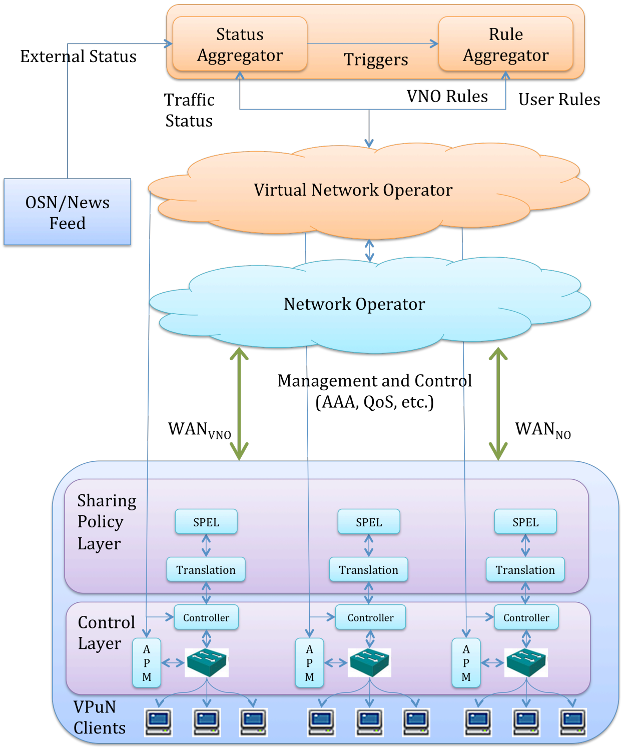

Figure 1: VPuN Architecture Overview

2.1.1 Access Point Manager (APM)

The APM is the system that contains the router control API described in Section 2.2. In addition to the basic APIs described there, the APM will also provide the ability to create, modify and configure wireless SSIDs. This APM is controlled via a simple SDN controller (e.g., POX) which resides inside the router along with the APM. The APM is the only software module that must be obligatorily present inside the home router of the end user.

2.1.2 Sharing Policy Expression

We propose a sharing policy expression language (SPEL) for the specification of network sharing policies. As such, a user can specify the amount of bandwidth that is willing to contribute and the period that he desires to share his network. Network sharing may be subject to other conditions and events which can be comprehensively expressed using SPEL. Besides the language specification, VPuN requires a run-time system for the translation of sharing policy expressions into network programming actions sent to the SDN controller (Figure 2).

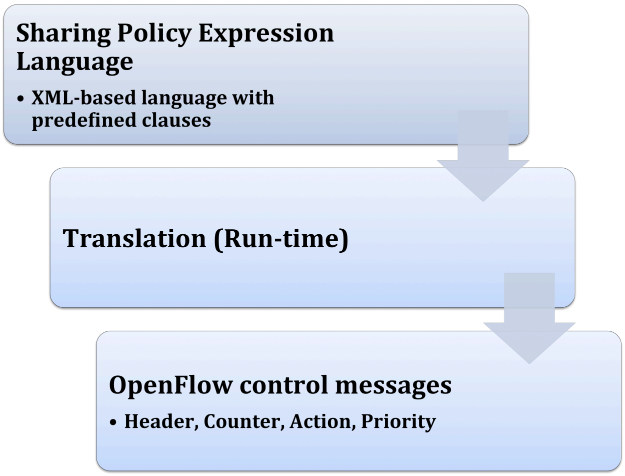
Sharing Policy Expression Language
• XML-based language with predefined clauses
Translation (Run-time)
OpenFlow control messages
• Header, Counter, Action, Priority

Figure 2: Translation of Sharing Policy Expressions

SPEL employs an XML-based schema, as shown below:

```
<rule_id>1</rule_id>
<name>"Share when I sleep"</name>
<condition>
  "$time_of_day > 8.00 PM AND $time_of_day > 5.00 PM"
</condition>
<action value="SHARE">
  <data_cap value=60>
     units = "GB"
  <data_cap>
  <rate value=4>
     units = "Mbps"
       <rate>
</action>
<expire>"00:00:00 31 Nov 2013 IST"</expire>
<priority>1</priority>
<watch value="My event cancelled">
  <action>EXPIRE</action>
  <handle value="@AuthorizedHandle">
     <type>"Twitter"</type>
  </handle>
</watch>
```

As shown in the example above, a rule consists of following clauses.

- A condition clause that tells what condition must be met for this rule to become active. This clause will support a few variables like $time_of_day, $current_location, $my_subnet, $my_vlan_id, etc., and comparison operators like >,<,==,=~, etc., to check for greater than, lesser than, equals to and matches.
- An action clause that will state what must be done when the condition is met. The action can be "SHARE", "DENY" etc., for sharing capacity on the access point created by the access point manager or denying traffic etc. The sub clauses for an action will be rate and data_cap given with supported units.
- A watch clause that tells the SPEL what expression to watch for in authorized handles. The handle will be specified as a type (Twitter, Facebook, News, etc.,) along with an id. The watch clause will set triggers in the status aggregator described in the next section resulting in rules being installed automatically. An action sub clause will be added to this watch clause stating what must be done when a certain watch pattern is found on the handle. The action sub clause will support an extra action other than those listed above which is EXPIRE. This action will cause the rule to expire immediately. In addition to social media handles, an additional handle type of "network_info" will also be supported to listen for network events and information such as node down, capacity updates, etc.
- Besides these, certain parameters like expiry timeout, priority to resolve conflicts, rule id, name, etc., will also be supported.

A rule thus created can be shared easily amongst various nodes. This also allows for a VNO to aggregate all the rules installed by various users for the purpose of analysing trends and managing the network.

It is not required that the SPEL translator and the SDN controller are collocated with the home router. Instead, they can reside on a server hosted by the NO or the VNO.

2.1.3 Status Aggregator

The status aggregator is a curated service that gathers information such as natural disasters, crowd distribution, weather changes, etc., about the external world via social media and other authorized news feeds and information such as network load, traffic conditions, load distribution, patterns, etc., about the network itself via distributed controllers (for SDN enabled network components) or SNMP (for traditional networks). The information that is scanned for is determined by filters applied by the rule aggregator (illustrated in Figure 3).

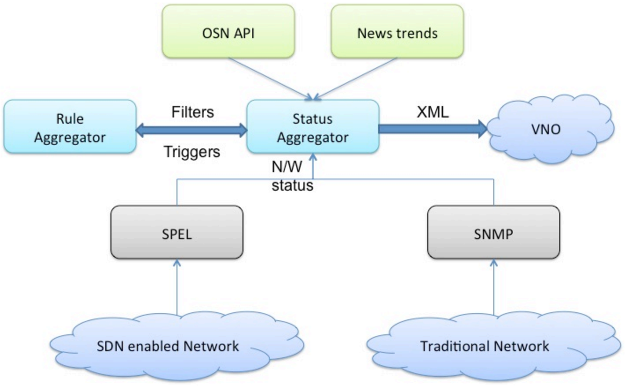

Figure 3: Status Aggregator Flow

Data received from various state providers is combined into a unified JSON/XML format and provided to the VNO. This format is described below.

```
<update>
        <type>external | network</type>
        <sub-type>
          facebook | twitter | google | switch | router
        </sub-type>
        <source_identifier>
          <handle id> | <mac_address> | <vlan_id>
        </source_identifier>
        <info type>
          expression | bandwidth | load distribution |
        node status
        </info type>
        <info>
          "<actual info>"
        </info>
</update>
```

This information can be monitored manually by the VNO so that corrective actions can be taken in situations that demand them. Alternatively, the VNO can also enable a rule aggregator which takes into account all rules in the system and takes corrective actions automatically.

Note that the status aggregator can be a distributed service that can split aggregation across individual status aggregator nodes and share status data thereby enabling efficient use of resources.

2.1.4 Rule Aggregator

The rule aggregator is an intelligent decision making engine that can act as a proxy agent for an end user or VNO, take informed decisions and install rules in the system for them.

The rule aggregator (Figure 4) is just an extension of the SPEL with two additional functions:

a. A listener port that receives data from the status aggregator.

b. A listener port that consolidates rules from distributed client rule engines.

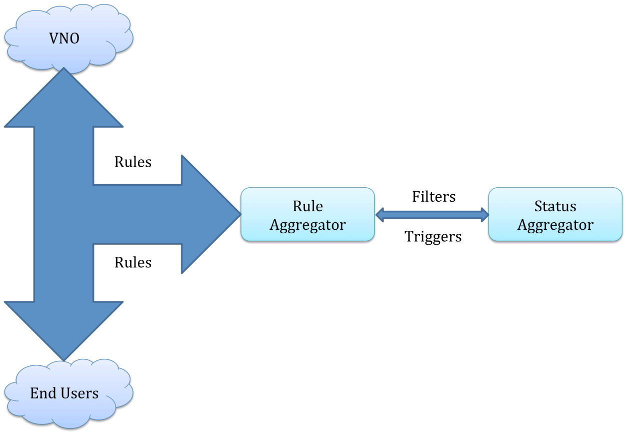

Figure 4: Rule Aggregator Flow

Based on the rules aggregated from VNOs and end users, the rule aggregator pushes filters to the status aggregator(s). The status aggregator in turn pushes triggers to the rule aggregator to install rules automatically as and when the system demands it.

This provides the rule aggregator the ability to scan the network and external world for relevant information and make informed decisions on behalf of both the VNOs and the end users.

As with all other components, the rule aggregator too can be centralized or distributed as per deployment demands.

2.1.5 Third Party applications

While technically not a part of the architecture, all the components in the architecture will expose APIs that can be used to build external applications that can provide value added services to end users and VNOs. As examples, a few applications are suggested below.

- A reward point management application that tracks information received from the rule aggregator and status aggregator to decide actual bandwidth shared by users and reward them redeemable points.
- A social media based access enabler that determines APs that belong to friends.
- An intelligent AP identifier that talks to local status and rule aggregators to determine which of the free APs will be available based on
  - Most bandwidth,
  - Highest duration (based on rules),
  - VNO/NO preference

While the architecture shows these applications on the top of the stack, these applications could reside anywhere from the datacentre of a VNO to a mobile device of an end user.

2.2 Router Control Abstraction

In this section, we present the router control abstraction that we envisage for creating, deploying and managing VPuN. The home router enables connectivity between end-users and the network and enforces resource allocation policy. The router enables end-user device connectivity over a distinct wireless SSID (e.g., for PAWS). This design approach enables progressive deployment of the technology in existing networks, through a simple replacement of the home router.

Our home router software consists of an OpenFlow-enabled switch and a controller application. The controller application exposes a minimal JSON-RPC API, enabling third-party VNOs access and resource allocation policy configuration and usage logging. In addition, the routing platform must expose to the controller a configuration API for traffic shaping queues and Access Point (AP) setup.

A VPuN must enable three functionalities: accurate resource control, user-level privacy and effective control.

2.2.1 Defining the router control abstraction

VPuN requires an isolated wireless network within each home network. In the data-link layer we enable virtualization through the multi-SSID functionality of modern wireless chips. The home router exposes two distinct wireless networks: the home and the VPuN. We rely on a SDN slicing layer to enforce bandwidth isolation between the two wireless networks. FlowVisor can be used as a building block for network slicing, allowing the home network user and the VNO to manage their slices [6, 1] using their own SDN controllers. Slicing policies can be expressed by users based on SPEL, as exemplified in Section 2.1.2.

For the VPuN, we modify the network control logic, in order to develop a new abstraction that matches the requirements of our system. We expose through the controller a JSON-RPC API to enable VNO policy expression. The API comprises of four functions and is presented in Table 1. User_On and User_Off functions are implemented by the VNO services and provide guest user connection and disconnection notification. In order to enable Internet connectivity for a guest user, the VNO must modify router policy, using the User_Config method. The method uses as a parameter a structure to define guest user service accessibility (e.g. permitted domain name or IP addresses) and resource allocation (aggregate amount of data and pick rate limit). Finally, the VNO can aggregate billing information on a per user basis, using the Get_user_stats method. The method provides accounting and accountability information per guest user to the VNO. We believe that the API provides sufficient primitives to control user access, while the usage of a local controller to enforce network policy minimizes forwarding plane performance degradation.

Function	Description
User_On/User_Off	Router -> VNO: notify VNO on the connection/disconnection of an authenticated user
User_Config	VNO -> router: VNO configures resource parameters for the user.
Get_user_stats	VNO -> router: VNO requests flow level user traffic usage information.

Table 1: Router Control API

2.2.2 Defining the access network

Enforcing resource control policies on the edge network is not sufficient for end-to-end resource allocation. On one hand, the highly asymmetric nature of home network traffic, makes it impossible to control the downlink through control of the uplink. This is a direct consequence of the asymmetric link speed, as well as, due to the traffic pattern of Internet applications. On the other hand, network bottlenecks that affect traffic prioritisation, are beyond the control of the end-user and usually found within the backhaul of the ISP network, in the majority of current broadband network. In order to establish sufficient end-to-end resource allocation, we need to express resource allocation policy within the ISP network.

Our design provides an evolutionary deployment mechanism. In order to establish clean separation between homeowner and VPuN traffic in the ISP network, the NO must support multi-addressing for each household. Specifically, each access point has a public IP (WAN_{NO}), for the home Internet traffic, and multiple private IP addresses WAN_{VNO} (depending on the number of VNOs), which are used to route traffic (the VPuN wireless interface is bridged to the respective WAN_{VNO} interface) for VNO clients. The IP level virtualisation of the endhost traffic matches finely with the routing control abstraction of the ISP network. The NO through its routing policy aggregates incoming and outgoing traffic in a single point within the network to apply NAT translation, as well as, enforce per-VNO aggregate resource allocation and accounting. In addition, since the VNO traffic constitutes an aggregated IP subnet, the network operator can apply QoS polices to marks all VPuN traffic at a lower QoS (Less than Best Effort (LBE)) compared to the paid user traffic (Best Effort (BE) or higher) or higher QoS depending on the agreed SLA between the NO and the VNO. As the deployment of SDN control augments within the ISP network, the NO can support more fine control and provide dynamic resource slices within the network to the VNO.

A SDN enabled router can trivially support this forwarding mechanism. OpenFlow control provides primitive to translate source and destination IP addresses on a per-flow basis, thus tagging traffic on the end-nodes. In addition, the controller is responsible to enforce short-term resource control for each user. The User_Config configuration API call contains rate limiting configuration parameters, which are translated to per-user queue setup. Using the User_Config API, the VNO can coordinate the assignment of VPuN clients to access points based on information available to rule aggregators and status aggregators, such as traffic loads. In case the VPuN client is disconnected, a connection with a another proximate access point can be re-established.

Figure 5 presents an example VPuN network where the VNO is the local government council.

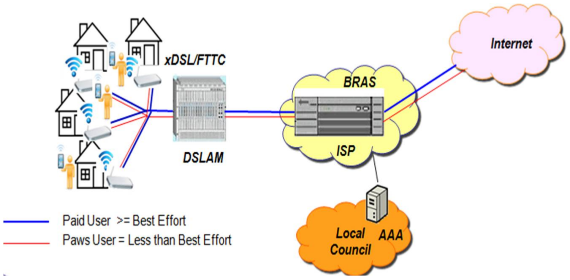

Figure 5: Virtual Public Network: Local government participating as a Virtual Network Operator

Our architecture allows much more finer grained control and flexibility to VNOs: from lower priority LBE free Internet access to higher quality services. This provides new opportunities for charities, NGOs and local government bodies to become VNOs. The NOs can lease their network at a lower cost to these organizations and allow access to unused capacity.

Our major vision for such an evolutionary architecture is that during emergencies such as natural disasters or terrorist activities, home networks have the provision to be opened up to the public to communicate. The VNO has the choice on which access points can be opened up: the home user could have agreed to open up his entire access network during such events. Using the external data feeds from OSN and news trends, the home networks can then be opened up either through manual VNO setup or automatically.

3. Conclusions

With the advent of software defined networking (SDN), there are more opportunities for network operators to create, deploy and manage in large scale open home networks. In this paper, we proposed Virtual Public Networks (VPuN), home networks created, deployed and managed through an evolutionary software defined network control abstraction. We showed that the proposed control abstraction enables more flexibility for users and network operators to share and control the network and to allow new stakeholders to emerge as virtual network operators.

4. Acknowledgement

This work was supported by the EPSRC Grant EP/K012703/1.

5. References

[1] Y, Yiakoumis et al., Slicing Home Networks, ACM SIGCOMM HomeNets 2011.

[2] Sathiaseelan, A. and Crowcroft, J. LCD-Net: lowest cost denominator networking, SIGCOMM CCR 43, 2 (April 2013), 52-57.

[3] La Rue, F. Report of the Special Rapporteur on the promotion and protection of the right to freedom of opinion and expression, Human Rights Council, UN General Assembly, 16 May 2011, http://goo.gl/MDjS7

[4] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker, “NOX: Towards an operating system for networks”, SIGCOM CCR, vol.38, no 3, 2008.

[5] N. McKeown, et al., “OpenFlow: Enabling Innovation in Campus Networks”, ACM SIGCOMM CCR, 38(2), 2008.

[6] R. Sherwood, et al., Can the Production Network Be the Testbed?, USENIX OSDI 2010.

[7] A. Sathiaseelan, J. Crowcroft, M. Goulden, C. Greiffenhagen, R. Mortier, G. Fairhurst, D. McAuley, Public Access WiFi Service (PAWS), Digital Economy All Hands Meeting, Aberdeen, October 2012.