Homenet3D: A New View on Home Network State

Grenville Armitage and Dominic Allan
garmitage@swin.edu.au,domallan8@gmail.com
Centre for Advanced Internet Architectures, Swinburne University of Technology, Australia

Abstract—**Although it performs an increasingly important role in our lives, the home network remains a bit of a mystery when end-users wish to know what is ‘happening on the inside’. This paper introduces Homenet3D, an open-source project that allows users to view their home network’s state as objects in 3D space within a web browser window. Homenet3D maps quantitative state to the shape, size, spin, bounce and/or colour of selected objects to qualitatively communicate what is happening on the network. We describe our Homenet3D implementation for routers running OpenWRT, and discuss the potential for visualising multi-router/multi-subnet home networks.**

I. Introduction

Today’s modern, Internet-connected residential home has an increasingly busy network within its borders. Typically there will be a fixed broadband connection into the home, providing Internet access to PCs, laptops, tablets, smartphones, network-enabled TVs and streaming-media clients, security cameras, and so on. As the *Internet of Things* (IoT) takes hold, everyday devices (such as such light switches, washing machines, air conditioning systems, and so on) are also being augmented with IP-based network interfaces and remote control systems [1]. Yet despite its increasingly important role in our lives, the home network remains a bit of a mystery when end-users wish to know what is ‘happening on the inside’.

Our premise is that a useful, qualitative, relatively non-technical view of the home network’s current state may be created through the use of suitably designed & animated objects in a virtual 3D environment. Furthermore, network configuration changes might be effected through an end-user’s interaction with objects inside the virtual world.

The Homenet3D project [2] envisages the typical home user utilising their laptop, tablet or smart phone to view their home network’s state as objects in 3D space within a browser window. Particular visual attributes of these objects (such as shape, size, spin and/or colour) will vary over time to qualitatively communicate what is happening on the network.

The rest of this article is structured as follows. Section II places Homenet3D within the broader context of efforts to define and manage home networks. Section III discusses the state of 3D rendering support in consumer devices. Our prototype of Homenet3D for OpenWRT is introduced in section IV and configuration options are described in section V. Sections VI and VII explore the implications of multi-router home networks, controlling network devices through the 3D interface, and potential applications outside the home environment. We conclude in section VIII.

II. Home networks

For many current homes, a single IP gateway sits between the home and the broadband modem providing ADSL, DOCSIS, optical fibre or fixed wireless service to the outside world. Communication within the home will involve one or more link layer technologies such as wired Ethernet, 802.11 WiFi or Homeplug (Powerline) networking. The gateway, wired switch ports, 802.11 access point (AP) and Homeplug interfaces may be separate devices or integrated within the home’s broadband modem.

Despite the diverse layer-2 technologies, current consumer products often encourage a single-subnet topology centered around a single home gateway. Network address translation (NAT) then enables all the in-home devices to (relatively transparently) communicate with the wider Internet using the home’s single ISP-provided IP address.

Industry efforts, such as the Home Gateway Initiative (http://www.homegateway.org) and the UPnP Forum (http://www.upnp.org), are developing technologies for home networks to self-configure and operate in as much of a plug-n-play manner as possible. Not surprisingly, existing standards largely assume IPv4-based home networks. More recently the IETF’s Homenet working group (http://tools.ietf.org/wg/homenet) has explicitly chosen to focus on an IPv6-based future where homes may have multiple internal subnetworks and routers and support multiple ISP connections to the outside world (*multihoming*) [3].

Homenet3D is not directly related to these other efforts. Rather than developing protocols for instantiating home networks, Homenet3D focuses on providing non-technical users with a qualitative sense of their home network’s internal state. This might include the aggregate bandwidth in or out of the home, clients currently associated with their WiFi AP, DHCP-assigned internal IPv4 or IPv6 addresses, ad-hoc or auto-configured internal IPv6 hosts, active NAT sessions and firewall rules and so on. Homenet3D augments current browser-based interfaces to home gateways, which usually provide terse and limited quantitative details (largely cryptic to the average non-technical end user).

III. Immersive 3D Monitoring

Dynamic 3D environments enhance the information that we may concurrently present to a viewer. Relationships may be qualitatively expressed by the spatial positioning of visible 3D objects relative to each other (distant vs close, clustered vs scattered, and so on). Underlying quantitative values may be approximated by a 3D object’s static characteristics (such as scale, radius, colour or shape) or dynamic characteristics (such as rotation speed or bounce rate).

Presenting system state in 3D is not an intrinsically new idea. The SGI File System Navigator turned up in 1993's Jurassic Park, sporting a navigatable 3D representation of the unix file system. Subsequent examples include network activity visualisation [4], virtual-world metaphors for interacting with computer process space [5] and virtual world collaboration systems [6]. A recent survey of many ideas in 2D and 3D visualization to assist with network security monitoring can be found in [7].

A. From game engines to WebGL

Some of our own previous work focused on re-purposing the server and client engines of a late-1990s era first person shooter (FPS) game to provide a multi-party monitoring of network state using 3D "worlds" [8]. Inside the virtual world multiple network states were represented in real-time using 3D objects having *visually orthogonal* attributes. (An example of visual orthogonality might include making a pyramid shaped object's spin rate and colour proportional to packets per second and number of flows at a measurement point respectively, as spin is unlikely to be mistaken for colour.) We utilized the Quake III Arena game engine for efficient use of network resources between clients and server, and 3D-capable clients across multiple platforms [9].

However, dedicated multiplayer game clients are no longer the only place to find 3D rendering and networking capabilities. The emergence of HTML5 and WebGL standards opens up the potential for 3D virtual environments being created in-browser on modern consumer devices. Javascript toolkits like Three.js (http://threejs.org) ease the creation of dynamic and interactive in-browser 3D applications. HTML5's WebSockets (http://tools.ietf.org/html/rfc6455) enable browser-based applications to engage in continuous, two-way communication with remote servers. In other words, new technologies are making it easier for residential home gateways to present their dynamic state inside 3D worlds displayed by HTML5/WebGL-compliant browsers.

B. The rise of smartphones and tablet computers

Ten years ago we still needed decent PCs or laptops to provide high resolution colour graphics and significant computing power. Today, sub-$150 smartphones and tablet computers are sporting colour touchscreens with 800x600 or higher resolutions, hardware accelerated graphics, 802.11g/n WiFi connectivity and 1GHz+ processors.

As productivity enhancer, fashion statement and the focal point of many people's social lives we might reasonably assume one or more such devices will be present in many home networks. Given the steadily increasing availability of mobile browsers capable of WebGL-based 3D rendering, Homenet3D arguably moves from theoretically interesting to plausibly useful.

IV. Homenet3D for OpenWRT

To demonstrate Homenet3D we targeted OpenWRT (https://openwrt.org), a Linux distribution commonly used to replace the factory firmware on many commercial, residential WiFi gateways and similar embedded systems. Replacing the factory firmware typically provides users with additional configuration options and extended functionality.

We have Homenet3D running on ARM and MIPS machines emulated in QEMU, and a physical TP-Link WR-1043ND wireless gateway. The OpenWRT SDK can build Homenet3D-capable firmware for other platforms that OpenWRT supports.

A. System state as objects in a 3D world

Figure 1 illustrates a Homenet3D client (web browser window) presenting the network as seen by a single OpenWRT router and WiFi access point (AP). Different internal OpenWRT system states are mapped to individual on-screen *entities*. Homenet3D's entities are objects whose appearance and behaviour qualitatively represent some underlying system state. The user's mouse (or touch screen controls) are used to rotate, zoom and pan the view. Hovering the cursor over an entity triggers a pop-up subwindow with more specific (quantitative) information about the associated system state.

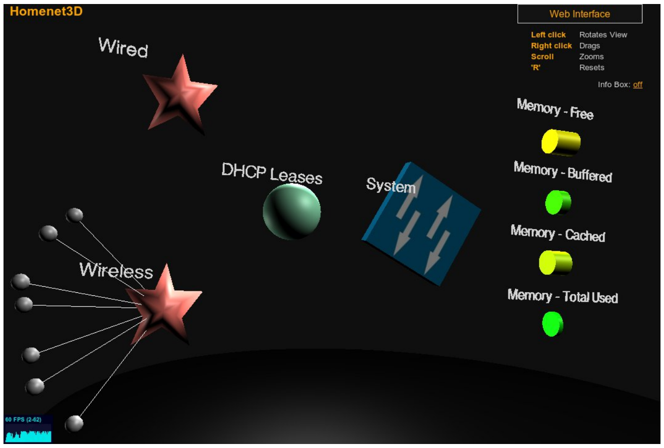


Figure 1. Homenet3D for OpenWRT viewed on a WebGL-enabled browser. Hovering the mouse over an object reveals detailed state information

Figure 2 shows the categories of OpenWRT system state tracked by Homenet3D. There is one entity each for System, Memory, DHCP Leases, aggregate Wired devices, aggregate Wireless Devices and individual Wireless Devices. Each entity is associated with specific information. For example, for the System entity there is the device's hostname and uptime as well as the network interface information for WAN and LAN. Wired and Wireless entities house all information about the wireless and wired devices currently connected to the system. Both the Wired and Wireless entities spawn child device entities that detail each individual attached device. The Wired, Wireless and Device entities leverage the lease information housed in the Leases entity.

Homenet3D currently provides a number of specific mappings. The blue switch model's spin rate and size is proportional to network traffic on the WAN interface. Each

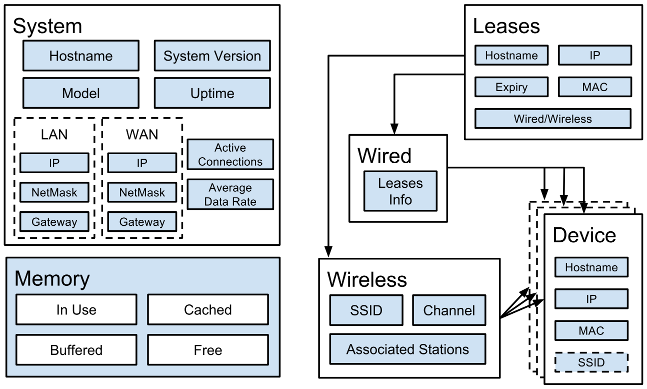


Figure 2. OpenWRT System States tracked for Homenet3D

memory category is presented along a cylinder colour-coded to represent different types of memory use. Wireless clients of the AP, and wired DHCP clients, are represented by small spheres-on-strings hanging off two large stars. More spheres cluster around each star as more clients arrive. Hovering one's mouse over either Wired or Wireless stars will bring up detailed information on client MAC addresses and assigned IP addresses in a pop-up window. Hovering one's mouse over an individual sphere-on-a-string will bring up information about that specific client.

B. Turning browsers into Homenet3D clients

Viewing is as simple as pointing your browser at a specific page hosted by the OpenWRT device's internal web server (for example, `http://192.168.0.1/hnet3d` if your OpenWRT gateway sits at 192.168.0.1). The returned `index.html` directs your browser to download and execute additional Javascript files from the OpenWRT device via regular HTTP on port 80. From this point on the browser is executing the Homenet3D client, and communication continues via Web-Sockets on port 10001.

As illustrated in Figure 3, Homenet3D supports multiple Homenet3D clients being connected at the same time. Key OpenWRT system state is streamed to all attached Homenet3D clients over their individual WebSocket connections. All 3D rendering and viewer controls are instantiated on the client side, allowing each end-user to view and navigate around the Homenet3D world independently.

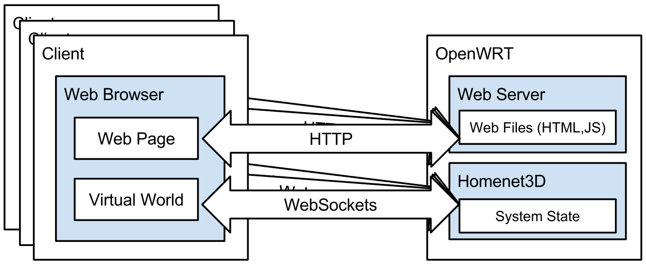


Figure 3. Multiple Homenet3D clients may concurrently view system state

C. Resources consumed by Homenet3D

OpenWRT is typically used on embedded systems having Flash memory in the 4MB to 8MB range and 16MB or more RAM. In this context Homenet3D cannot afford to take up significant additional space. A monitoring system should also add little to existing home network traffic.

The current Homenet3D client consumes ~972KB in the Flash image. A similarly-sized burst of local network traffic is generated when downloading the Homenet3D client to a browser. (Many browsers will cache the initial download, eliminating much of this traffic from later reconnections to the Homenet3D server.) Subsequent traffic to each Homenet3D client is short ~5-10KB bursts per change in monitored system state (e.g. when a WiFi client comes or goes). Such bursts of traffic will be fairly infrequent (tens of seconds or minutes between events).

V. Mapping state to 3D representations

Homenet3D's ultimate utility depends on the meaningful mapping of network states (such as aggregate bandwidth, or number of active WiFi clients) to the dynamic behaviour of in-world entities representing those states.

A. Configuring entities and objects

A single configuration file on the Homenet3D server defines the mapping of different OpenWRT system states to on-screen entities, and the initial visual characteristics (attributes) of the 3D objects representing each entity (such as orientation, position, size, colour, spin rate and bounce rates). The configuration file further defines how changes in system state map to changes in one or more of an entity's attributes (for example, how large an object will grow in response to increased bandwidth use). Table I shows a range of per-entity attributes that may be set.

Table I
Configurable in-world Object Attributes

Attribute	Format	Description
state_types	string vector	One or more internal states the entity represents
position	[x,y,z] vector	Object position in 3D space
bounce_height	integer	Height of object bounce in Z-axis
bounce_freq	integer	Period of bounce in Z-axis
rotate_speed	float	Rotation rate around Z axis
colour	[R,G,B] vector	Object colour
label	string	Text displayed with object
text_colour	[R,G,B] vector	Text label colour
radius	integer	Object radius
rotation	[rx,ry,rz] vector	Object's 3D orientation
shape	integer	Select pre-defined (1..6) or custom (7) shape
objfile, mtlfile	string	Names of custom OBJ and MTL files
scale	float	Ratio of original size to display size

We can illustrate this process using Algorithm 1, a fragment of Homenet3D configuration file defining the 'switch' entity used to represent System state in Figure 1. This entity is at location [0,0,0] in 3D space, rotated to a specific initial orientation [0.5,0,0] (expressed in radians around the x, y and z axes respectively) and has a text label of "System" written in 3D space next to the object. The shape, objfile and mtlfile options (Section V-C) indicate that the object itself is a custom design, and the mappings=() option (Section V-B) indicates that the system's average datarate (Homenet3D's calculation of network traffic through the OpenWRT system's WAN interface) is mapped to the object's rotation speed.

Algorithm 1 Fragment of Homenet3D configuration file

```
name = "Homenet3D"
version = "0.3"
application =
{
// Global settings
text_colour = [211,211,211];
// Configuration example for System entity
 entities = (
 {
  position = [0,0,0];
  label = "System";
  state_types = ["system","network"];
  shape = 7;
  objfile = "models/switch.obj";
  mtlfile = "models/switch.mtl";
  scale = 0.5;
  bounce_freq = 0.0;
  rotation = [0.5,0.0,0.0];
  // Mappings
  mappings = (
    {
      metric = {
        metric_name = "datarate";
        upper_thresh = "100000.0";
        lower_thresh = "300.0";
      }
      attrs = (
        {
          attr_name = "rotate_speed";
          attr_max = "5.0";
          attr_min = "1.0";
          attr_gran = "0.5";
        }
      )
    }
   );
  }
 );
}
```

B. Mapping states to attributes

Homenet3D uses the mappings=() configuration option to define a list of linear mappings from measured system state values to displayed object attributes. Pairs of metric={} and attrs=() sub-parameters identify the system state and target attributes respectively.

For example, Algorithm 1 shows the System entity configured such that values of datarate (metric_name) between 300 (lower_thresh) and 100000 (upper_thresh) are mapped to a rotation speed (attr_name) between 1.0 (attr_min) and 5.0 (attr_max), with the rotation speed quantised to the nearest 0.5 (attr_gran). If datarate falls outside either upper or lower thresholds, the object's rotation speed will be capped at the max or min rates respectively.

Similar syntax can be used to establish other mappings between Homenet3D-monitored system state and numeric object attributes in Table I. Being easily configurable, Homenet3D enables further research into the utility of different mapping strategies (such as whether spin, bounce or colour better capture the intuitive importance of state such as network traffic, memory consumption, or numbers of clients). The potential benefits of non-linear mappings will also be explored in later versions of Homenet3D.

C. Using custom 3D objects

In addition to six pre-defined objects, Homenet3D allows the use of OBJ (object geometry) and MTL (material template) definition files to specify custom 3D objects for each entity. Created in the 1990s by Wavefront Technologies, OBJ and MTL files are easily generated by many 3D design tools such as Milkshape3D, Blender, Shade or SketchUp Pro. (A diverse range of example objects can be found at sites such as http://tf3dm.com/3d-models/all.)

As illustrated in Algorithm 1, a custom object is selected using shape=7 and the associated OBJ and MTL files are identified using the objfile and mtlfile options. By default custom entity objects are stored on the OpenWRT device, and retrieved during initial startup of each Homenet3D client.

Custom objects must be relatively simple and small to minimise consumption of the OpenWRT device's Flash memory. However, objfile and mtlfile may also be full URLs. This enables hosting of larger custom objects on another web server (also accessible from the home network) that has more local storage space and allows cross-origin content retrieval from Homenet3D clients.

VI. Visualising Multi-router environments

Homenet3D can also encompass future homes that contain multiple internal networks, multiple routers, and (potentially) multiple connections to different ISPs.

A. Complex home environments

The single-gateway/single-subnet model is a convenient simplification that increasingly fails to reflect a modern home's network environment. It is increasingly likely that a home's primary network will be divided at the IP layer into wired and wireless subnetworks (e.g. to isolate guest and trusted WiFi access, or to separate home backup server traffic from IP telephony and games traffic, and so on). This implies multiple WiFi networks linking back to the common ISP connection, potentially multiple routers, and certainly more complex in-home topologies.

Complicating the situation further are networks created through *tethering*, where WiFi-equipped smartphones act as their own local WiFi access points and share their 3G or 4G cellular internet access with other nearby WiFi devices. Such networks are transient, and independent of the home's primary Internet access network. Yet they can provide an entirely real service to the tethered WiFi devices (albeit slower, and with higher latencies, than Internet access through the home's fixed-line broadband service).

More advanced homes may also implement multihoming, where the primary in-house network has active gateways to multiple ISPs at the same time. Multihoming can ensure a home network stays connected to the Internet when one (or more) ISPs suffer outages. The cost is additional complexity inside the home network – someone must make and distribute routing policies that control how the home's router(s) will spread traffic across the available ISP links.

Significant work is currently underway in the IETF's Homenet working group for automating the discovery and self-configuration of devices within multi-router, multi-subnet and/or multi-homed networks. Core to this is the Home Network Control Protocol (HNCP) – both the specification [10] and an actual implementation for Linux routers (https://github.com/sbyx/hnetd) are under active development.

B. Presenting multiple routers with Homenet3D

Homenet3D provides one possible approach to visualising the state of a network having multiple OpenWRT devices. First we assume Homenet3D is installed on each OpenWRT device. In their Homenet3D configuration files, one OpenWRT device is designated the Homenet3D 'master' while the other OpenWRT devices are designated Homenet3D 'slaves'. The master maintains system state information for itself and all slaves (who send the master regular updates of their own system states).

When a Homenet3D client connects to a Homenet3D master server the user is presented with state information for all the slave devices rendered in 3D space. Figure 4 shows how this might look for a system of two slaves and one master router. Each Homenet3D master or slave has its own mappings between system state and on-screen entity attributes. Each device might, for example, use a different custom 3D object to represent their 'system' state entity, or use different colour, bounce or spin rate mappings.

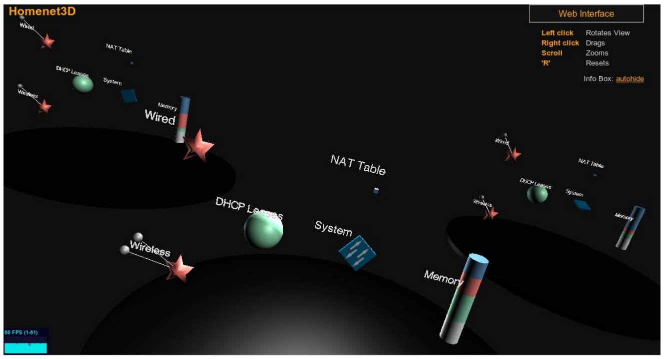


Figure 4. Homenet3D presenting a multi-router home network

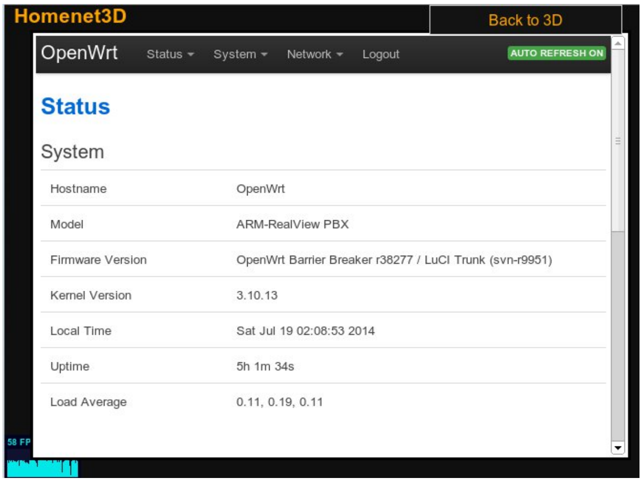
Each device must have Homenet3D pre-installed, with the identities of master and slave nodes specified in each devices' Homenet3D configuration file. As HNCP matures, future versions of Homenet3D will leverage HNCP functionality to auto-generate Homenet3D master and slave configuration files for discovered devices. As HNCP allows auto-discovery of network topology, we expect the spatial arrangement of entities within the Homenet3D world can reflect topological relationships between devices in the home network.

VII. Interacting with devices on the home network

The next obvious step is to offer *control* of network devices or systems through their in-world entities. We consider two avenues – control of actual home network infrastructure, and control of household devices that are interconnected by the home network.

A. Home network infrastructure

Homenet3D currently provides easy access to OpenWRT's conventional web interface by clicking on the words Web Interface in the top right corner of the Homenet3D screen. The conventional interface is then presented on-screen as shown in Figure 5. After the user interacts in detail with the underlying OpenWRT router (potentially making specific changes to their home network) they click on Back to 3D to return to the 3D world view. The user may freely alternate between 3D view and 2D GUI modes, with the conventional web interface remaining logged while the user is back in 3D view mode.



Status

System

Hostname	OpenWrt
Model	ARM-RealView PBX
Firmware Version	OpenWrt Barrier Breaker r38277 / LuCI Trunk (svn-r9951)
Kernel Version	3.10.13
Local Time	Sat Jul 19 02:08:53 2014
Uptime	5h 1m 34s
Load Average	0.11, 0.19, 0.11

Figure 5. OpenWRT's conventional web interface inside Homenet3D

A key future enhancement will be to enable triggering of macros – pre-scripted control actions – when the user interacts with entities within the 3D view. For example, previous work using 3D game engines re-purposed the in-game notion of 'shooting' an object to trigger specific firewall rule updates, without the end-user needing to worry about detailed firewall rule syntax [8].

In a Homenet3D environment we envisage pull-down menus appearing in 3D space when a user clicks on individual objects. These menus then allow selections from actions such as renewing or releasing DHCP leases, rebooting individual devices, changing firewall rules in response to unexpected traffic, and so on. The selected actions would be parameterised by information associated with the 3D object's entity.

For example, consider an object whose spin rate indicated what fraction of WAN link capacity was being consumed by the entity's associated device. One of the pull-down menu options might be to instantiate Quality of Service (QoS) rules that limit the device to one or more pre-configured fractions of the WAN link. End-users can focus on the conceptual task rather than the detailed syntax underlying each network or system management action.

B. Networked household devices

The *Internet of Things* (IoT) envisages a world where everyday devices and systems are augmented with IP-based network interfaces and embedded remote control systems [1]. IoT sees devices such light switches, washing machines, air conditioning systems, semi-autonomous vacuum cleaners, pacemakers, in-car navigation and entertainment systems, and so forth, being remotely monitored and/or controlled via their network interfaces.

The "internet" in IoT may imply connectivity between devices and the public Internet, or it may simply mean a multitude of devices within an organisational domain (such as a factory, shopping centre or home) sharing an isolated, IP-based communications infrastructure. It is beyond the scope of this paper to address the many questions surrounding security and authentication in an IoT world. But Homenet3D offers interesting opportunities in an IoT world.

In section VI-B we discussed how Homenet3D supports slave OpenWRT devices feeding their system state updates to a master server, and the master server controlling how each slave's state is mapped into each Homenet3D client's 3D world. It is easy to generalise the Homenet3D slave to be any network-attached IoT device generating a stream of telemetry data indicating the state of its current environment.

Combine this with custom 3D objects that evoke everyday IoT devices being monitored (section V-C), flexible placement of all entities with the 3D world, and the potential to trigger scripted actions on slave devices. It is not hard to see Homenet3D being applied to monitoring and controlling a wide range of networked devices around the home, from the browser of your smartphone or tablet.

VIII. Conclusion

Modern homes see an increasingly complex mix of technolgoies and devices making up their home network. Usually one IP gateway (but potentially more) sits between the home and broadband service(s) to the outside world. In-home communication involves one or more link layer technologies such as wired Ethernet, 802.11 WiFi or Homeplug (Powerline) networking. Sometimes these technologies are integrated in a single device or spread across multiple devices.

Homenet3D is a project aimed at letting end-users observe aspects of their home network infrastructure using increasingly-common, WebGL-enabled browsers in laptops, tablets and smartphones. Our goal is to present a useful, qualitative view of the home network's current state through the use of suitably designed & animated objects in a virtual 3D environment. To help the broader community explore this in detail we have developed and released an open-source, BSD-licensed prototype that runs on OpenWRT-based devices. In this paper we have discussed our prototype's initial mappings of OpenWRT system state to visually-orthogonal attributes of 3D entities viewed by the end-user, and Homenet3D's support for customisation. Finally, we outlined ways in which our Homenet3D concept may be relevant to the Internet of Things by providing a user-friendly way to monitor and control a wider range of networked devices.

Acknowledgement

This project has been made possible in part by a gift from the Cisco University Research Program Fund, a corporate advised fund of Silicon Valley Community Foundation.

References

[1] J. Zheng, D. Simplot-Ryl, C. Bisdikian, and H. Mouftah, "The internet of things [Guest Editorial]," *Communications Magazine, IEEE*, vol. 49, no. 11, pp. 30–31, November 2011.

[2] G. Armitage, "Homenet3D," Sep 2014. [Online]. Available: http://caia.swin.edu.au/urp/homenet3d

[3] T. Chown, J. Arkko, A. Brandt, O. Troan, and J. Weil, "IPv6 Home Networking Architecture Principles," IETF Secretariat, Internet-Draft draft-ietf-homenet-arch-17.txt, July 2014. [Online]. Available: http://tools.ietf.org/wg/homenet/draft-ietf-homenet-arch/

[4] P. Abel, P. Gros, C. Russo dos Santos, D. Loisel, and J.-P. Paris, "Automatic construction of dynamic 3D metaphoric worlds: an application to network management," in *Proceedings of SPIE, Visual Data Exploration and Analysis VII*, vol. 3960, San Jose, USA, Feb 2000, pp. 312–323. [Online]. Available: http://www.eurecom.fr/publication/276

[5] D. Chao, "Doom As an Interface for Process Management," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '01. New York, NY, USA: ACM, April 2001, pp. 152–157. [Online]. Available: http://doi.acm.org/10.1145/365024.365078

[6] B. Kot, B. Wuensche, J. Grundy, and J. Hosking, "Information Visualisation Utilising 3D Computer Game Engines Case Study: A Source Code Comprehension Tool," in *Proceedings of the 6th ACM SIGCHI New Zealand Chapter's International Conference on Computer-human Interaction: Making CHI Natural*, ser. CHINZ '05. New York, NY, USA: ACM, 2005, pp. 53–60. [Online]. Available: http://doi.acm.org/10.1145/1073943.1073954

[7] H. Shiravi, A. Shiravi, and A. Ghorbani, "A Survey of Visualization Systems for Network Security," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 8, pp. 1313–1329, Aug 2012.

[8] W. Harrop and G. Armitage, "Real-time Collaborative Network Monitoring and Control Using 3D Game Engines for Representation and Interaction," in *Proceedings of the 3rd International Workshop on Visualization for Computer Security*, ser. VizSEC '06. New York, NY, USA: ACM, 2006, pp. 31–40. [Online]. Available: http://doi.acm.org/10.1145/1179576.1179583

[9] G. Armitage, "L3DGE – Leveraging 3D Game Engines project," Feb 2007. [Online]. Available: http://caia.swin.edu.au/urp/l3dge

[10] M. Stenberg and S. Barth, "Home Networking Control Protocol," IETF Secretariat, Internet-Draft draft-ietf-homenet-hncp-01.txt, June 2014. [Online]. Available: http://tools.ietf.org/wg/homenet/draft-ietf-homenet-hncp/

Biographies

GRENVILLE ARMITAGE (garmitage@swin.edu.au) earned a B.Eng. in electrical engineering (Hons) in 1988 and a Ph.D. in electronic engineering in 1994, both from the University of Melbourne. He is a full professor of telecommunications engineering and founding director of the Center for Advanced Internet Architectures at Swinburne University of Technology. He authored Quality of Service In IP Networks: Foundations for a Multi-Service Internet (Macmillan, April 2000) and co-authored Networking and Online Games — Understanding and Engineering Multiplayer Internet Games (Wiley, April 2006). He is also a member of ACM and ACM SIGCOMM.

DOMINIC ALLAN (domallan8@gmail.com) completed his double degree B.Eng in telecommunications and network engineering and B.Eng in computer science and software engineering in 2014 at Swinburne University of Technology. He worked as a research assistant at Swinburne University's Center for Advanced Internet Architectures in 2013 and 2014 on the "Homenet3D" project.