Secure Affordable Sustainable Edge Clouds (SASEC) for Smart Cities and Enterprises

Malathi Veeraraghavan*, Dan Kilper†, Xiao Lin*‡, Rider Foley*, Weiqiang Sun‡
* University of Virginia, Charlottesville, VA 22903, {mv5g, xl5h, rwf6v}@virginia.edu
† University of Arizona, Tucson, AZ, dkilper@optics.arizona.edu
‡ Shanghai Jiao Tong University, Shanghai, China, sunwq@sjtu.edu.cn

***Abstract*—This paper describes a novel architecture that addresses the need for a low-cost alternative to today's solution for residential Internet access for households in cities left behind by the digital revolution. Given the costs and security vulnerabilities of all home-owners purchasing and maintaining their own PCs, application software and Internet access, our Secure Affordable Sustainable Edge Clouds (SASEC) solution consolidates computers into an edge cloud, and requires only inexpensive I/O devices, such as Keyboard, Video, Mouse (KVM) terminals and audio input/output devices, in user homes. The term Dumb Connected Devices (DCDs) is used to describe these user-owned systems. SASEC is also a suitable candidate for enterprises, such as university campuses, as it offers a strong security advantage. SASEC vastly reduces the attack surface since users' DCDs have no processor and no operating system, and hence cannot be compromised and used in botnets. The challenge lies in designing a high-speed network to interconnect the Edge-Cloud (EC) to the DCDs. Initial experiments show that with compression, 50 Mbps is required for 1080p HDTV video. If the EC-DCD network needs to support 100 simultaneous web-browsing sessions with video, then Gb/s wireless solutions are required.**

***Keywords*: Smart cities, Disadvantaged communities, Network applications and services, Network architecture and protocols, Dumb terminals**

I. Introduction

Most cities in the world have neighborhoods that fall on the wrong side of the digital divide. Lack of broadband Internet access is often blamed for this problem. However, a more basic problem is that many households do not have computers at home. For example, a 2014 report from the office of the New York City (NYC) Comptroller [1] notes that 27% of NYC households lack broadband Internet at home, and 17%, which was 532,902 households, did not have a computer at home. This second figure increases to 20% if the heads of households did not graduate from high school.

Research shows that children's grades improve when schools supply computers for students to access the Internet at home [2]. A 2014 study showed that students who do not have adequate online access at home risk falling behind their peers [3]. These studies and statistics show how the lack of computers and Internet access at home contribute to the lack of social mobility even within developed economies.

To address this problem, we propose a technical architecture called Secure Affordable Sustainable Edge Cloud (SASEC), which could lower the capital expenditures (capex) and operating expenditures (opex) for households to acquire Internet access at home.

The SASEC architecture consists of (i) an edge cloud (compute-and-storage clusters) that runs commonly used desktop applications, such as web browsers and Microsoft Office, and offers other shared services, and (ii) a per-user Keyboard-Video-Mouse (KVM) terminal with audio input/output devices. In a smart-city/smart-community deployment of SASEC, the hardware and software licensing costs of the edge cloud would be shared by a large number of households. Each household would require just a KVM terminal that is connected via network links to the edge cloud. KVM terminals are cheaper than standard desktops/laptops, and hence are more affordable for households without computers. The term Dumb Connected Devices (DCDs) is coined to describe these user-owned devices, which could include cameras and other IoT type devices. These devices are intended to be truly "dumb," which means that unlike thin clients [4], these devices will have no processor and hence no operating system or software.

The SASEC architecture may seem like a throwback to the pre-PC-on-the-desktop architecture that was used in the days of mainframe computing. But, in the days of mainframe computing, the screen output was ASCII text, while today, with graphical interfaces and the use of video, an architecture that separates the servers from the monitors requires a completely new design.

What makes this architecture *secure* is that the devices controlled by users do not have processors, which means these users do not have to install operating system and application patches to fix security vulnerabilities. This solution vastly reduces the attack surface, i.e., the number of hosts that need protection from becoming compromised and recruited, via implanted malware, into large-scale botnets. Botnets are a menace to the entire Internet user population, and hence the larger the number of SASECs deployed, the better for society-at-large. Edge-cloud servers are easier to monitor for viruses and security breaches, and to detonate detected malware in a VM in safe mode [5].

The SASEC architecture is more *affordable* than current solutions, in which each household is expected to purchase its own laptop/desktop, purchase its own software licenses for applications, and purchase its own Internet access, for the simple reason of increased sharing. Also, energy management would be easier in the edge cloud than in individual households, e.g., servers could be kept powered-off when demand is low.

The rise of the sharing-economy for many types of services, e.g., AirBnB for renting out rooms, RelayRides for renting personal cars, was described in a 2013 Economist article [6]. SASEC offers a method for sharing computer hardware, licensed software, and Internet access.

While security, affordability, and its potential to reach more households, are significant advantages of SASEC, what are the disadvantages? The *key design challenge* lies in achieving low-delay performance for the transfer of video output from the servers in the edge cloud (which would ideally be located in a datacenter in one of the residential buildings of the participating households) to the large number of KVM terminals located within apartments in that same building, or in other residential buildings of the participating households. Similarly, keyboard/mouse input from the KVM terminals should be delivered rapidly and securely to the edge-cloud servers. Sub-standard service would be detrimental to user adoption of SASEC. We propose a dedicated hardware solution for interconnecting the KVM terminals to the edge cloud.

A second challenge lies in determining who (what organization) owns and operates the edge clouds? One option is for SASECs to be deployed for such communities as part of Smart City initiatives, and *sustained* by city governments, at least until, commercial entities see sufficient economic value proposition in operating such edge clouds, or communities self organize and take over operations just as home-owner associations manage private roads. Public policy experts could evaluate whether taxes from increased household income (a 2013 study showed that even small increases in Internet uptake rates are strongly correlated with increases in employment and household income [7]) could pay for subsidies offered to residents of SASEC-user communities.

The SASEC architecture offers advantages over the current architecture even for *enterprises*, such as corporations, university campuses and government agencies. The cybersecurity advantage of SASEC is significant for enterprises. Additionally, a reduction in computer-administration staff costs is enabled. Today, many university departments have their own computer administrators to manage the PCs used by their faculty, staff and students. If some of these users find that the applications hosted on the edge-cloud servers are sufficient for their needs, and agree to give up their PCs, this Human-Resource (HR) cost can be reduced.

Section II describes the SASEC architecture when deployed for communities within smart cities. Section III shows how SASEC can be deployed in enterprises. Section IV presents the results of two experiments that were conducted to determine the amount of bandwidth required for the network between the edge cloud and the dumb connected devices. Section V describes our dedicated hardware solution for connecting the DCDs to the edge cloud. Section VI addresses three dimensions of sustainability. After reviewing related work in Section VII, the paper is concluded in Section VIII.

II. Secure, Affordable, Sustainable Edge Clouds (SASEC)

The SASEC architecture consists of a community/organization-owned edge-cloud with compute and storage servers and user-owned Dumb Connected Devices (DCDs). For basic computing and Internet applications, the DCD is a KVM terminal, which consists of a keyboard and a mouse for a user to provide input, and a video monitor for the user to receive output from the edge-cloud servers. Beside this user input-output device, there are other devices such as video cameras, microphones and speakerphones for audio input and output, respectively, Internet-of-Things (IoT) sensors and actuators, Virtual Reality (VR) cameras and headsets, and, in the near future, robots.

In the SASEC architecture, we propose to make all these devices *dumb*, i.e., lack processors, to lower costs and avoid security attacks. The recent Mirai distributed denial of service (DDoS) attack exploited vulnerabilities in the software installed on IoT devices such as digital cameras [8]. Unlike DCDs, thin clients have processors and operating systems, with typically no disk space. For the cyber-security advantage, we propose the use of DCDs for most users, while some users may want the flexibility of owning PCs or thin clients and still belong to the community served by the edge cloud.

Our proposal for using DCDs goes against the trend, which is to make all these devices "smart," i.e., by equipping all these devices with processors, operating systems and software. For example, web-connected cameras run http servers to allow a user to connect to the camera from a web browser. Such cameras leverage video compression software to reduce the amount of bandwidth required to send the video signals. In SASEC, cameras and other IoT devices should ideally be dumb, but this means a dedicated-hardware solution (see Section V) is required to handle video compression. The web server can be run on the edge-cloud to stream the video received from the camera.

Section II-A describes the network between the Edge Cloud (EC) and the DCDs as applied to a smart-city community deployment. Section II-B describes the hardware and software of the edge cloud.

A. EC-DCD network

Fig. 1 illustrates a three-building SASEC deployment. The edge cloud, located in Building 2, consists of compute clusters and storage clusters in racks, each of which has computers and Top-of-Rack (TOR) Ethernet switches, and one or more IP routers. For reliability, high-speed access links are shown as connecting the edge cloud IP router to multiple ISPs.

All three buildings are shown to have multiple floors, with apartments on both sides of a central shaft, e.g., elevator shaft. Located within apartments are KVM terminals, other types of DCDs, and PCs (some households may prefer to purchase and maintain their own desktops/laptops and still use some of the services of the edge cloud). Since some buildings may not be wired with Ethernet to each apartment, WiFi is an option. WiFi access points can be configured to forward packets between each other in a wireless-mesh configuration as illustrated in

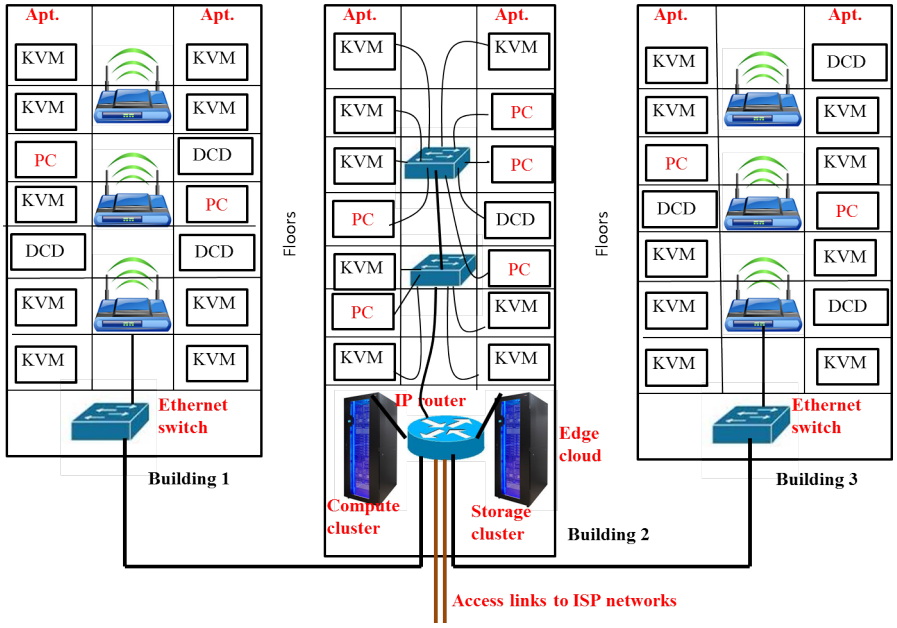


Fig. 1: Secure Affordable Sustainable Edge Cloud (SASEC)

Buildings 1 and 3 of Fig. 1. As an example of a wired building, Building 2 is shown to have Ethernet wiring to each apartment.

The key point to note in Fig. 1 is that the KVM terminals and other DCDs located in various apartments connect via the Ethernet switches in the basements of their buildings to the edge cloud located in Building 2. All applications, from web browsers to Microsoft Word, are executed on the edge-cloud servers, and KVM terminals within the apartments serve as simple I/O devices through which users interact with these remotely located applications.

To realize this distributed KVM architecture of SASEC, we looked for use cases in which a single server generates video that is visualized on multiple monitors. We found *four use cases* as shown in Fig. 2: (i) office workers who use multiple monitors for increased productivity (see Fig. 2a), (ii) video walls used for scientific and other big-data visualization (see Fig. 2b) (iii) graphics cards that allow video gamers to connect multiple monitors to a server (see Fig. 2c), and (iv) reverse-KVM switch (see Fig. 2d).

For the first use case, commodity workstations and laptops have built-in video cards that have one or two external ports (e.g., HDMI, DisplayPort or VGA), which along with USB ports, allow for multiple monitors to be connected to one host. For the second use case, visualization clusters [13] are used to drive large video walls. As an example of the third use case, Fig. 2c shows an Nvidia GeForce GTX 650 Ti card that can drive four display monitors via its two DVI ports, one DisplayPort and one HDMI port. In the fourth use case, a reverse KVM switch [12] allows multiple KVM terminals to be connected to a single server. Fig. 2d shows the front and rear views of a reverse KVM switch. It has classical PS/2 ports for the keyboard and mouse, and a VGA port for the display, on the front to connect to a server, and two sets of keyboard, mouse and VGA ports on the rear to connect to two KVM terminals. KVM extenders are also available. For example, a fiber-optic KVM extender [14] allows a DVI video signal to be carried over single-mode fiber for up to 4 km.

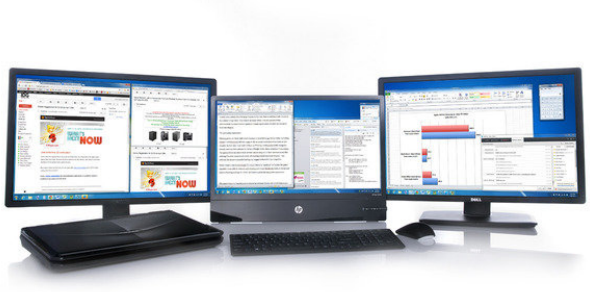
But in general, for all cases of video cables, there are restrictions on the length, and for the purposes of SASEC, where the servers can be located in one building and the KVM terminals in another, we need other networking technologies such as Ethernet and WiFi. The challenge is that the TCP/IP protocol stack that is commonly used on top of Ethernet/WiFi is implemented in operating systems and run on processors. Since the KVM terminals do not processors, and hence operating systems, a solution such the dedicated-hardware board described in Section V is required to extract the monitor signals from the received Ethernet/WiFi packets for delivery to the user, and to format keyboard/mouse input from the user into payloads suitable for transmission over Ethernet/WiFi. A corresponding board is required at the edge cloud.

B. Edge cloud design

The edge cloud can offer users four types of services:

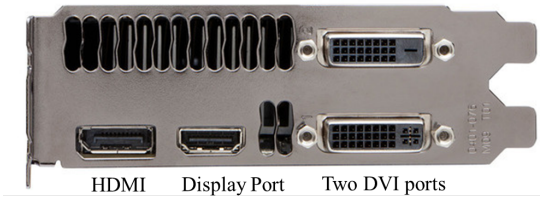
1) access to applications that are typically run on user PCs, such as word processors, web browsers, conferencing applications, and chat applications,
2) virtual machines for users who want to install their own O/S, and other software,
3) storage services, and
4) Internet access.

The edge-cloud hardware required to support these services consists of compute and storage clusters and IP routers, as illustrated in Fig. 1. Multi-homed Internet access is also illustrated in this figure.



(a) Office worker station [9]

(b) Video wall [10]



(c) Gaming: Nvidia card [11]

(d) Front and rear views of a reverse-KVM switch [12]

Fig. 2: Four use cases for connecting multiple monitors to a server

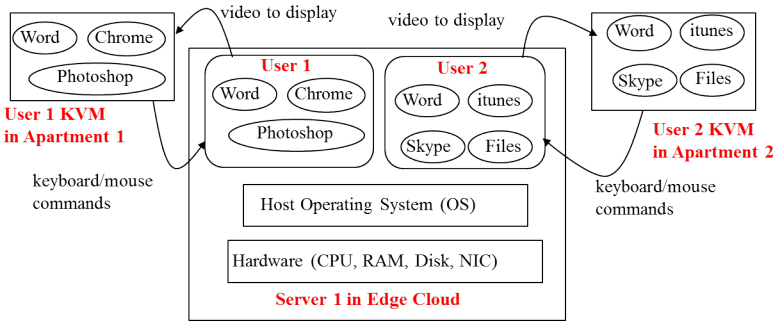


Fig. 3: Shared applications service

Edge-cloud software can be classified into four categories: (i) access to shared user applications, (ii) virtualization software, (iii) storage software, and (iv) operations software.

Shared-user applications such as OpenOffice, Microsoft Office (Word, Powerpoint, Excel), Adobe Acroread, Acrobat, PhotoShop, Apple applications such as iTunes and Quicktime, web browsers such as Chrome and Internet Explorer, conferencing applications such as Skype and Google Hangouts, and other common applications could be hosted on a set of servers. Multiple users could simultaneously access the same or different applications on one server. For example, Fig. 3 illustrates two users simultaneously running applications on Server 1. The Word application is being run by both users. The video output for each users' applications is being sent over the EC-DCD network to the corresponding user KVM terminals, and corresponding keyboard/mouse commands input by the two users are delivered to Server 1 via the same network.

Fig. 4 shows that other servers in the edge cloud, such as Server 2, could be used to run hypervisors to offer users *virtual machines (VMs)*. In this example, User 3 has chosen to run Linux on VM1, while User 4 runs Windows on VM2. With this service, users have the flexibility to install their own applications.

In SASEC deployments, since most users will only have KVM terminals, which do not have disk storage, it will be important to support storage services in the edge cloud. While commercial storage providers such as Dropbox and Box offer some amount of free storage, some users will need higher amounts of storage. For these users, the edge cloud can offer storage services with and without backup as is done in large enterprises. *Storage software* will need to be installed on the edge cloud.

Operations software include the types of software run by the IT division of large enterprises. A DHCP server is required

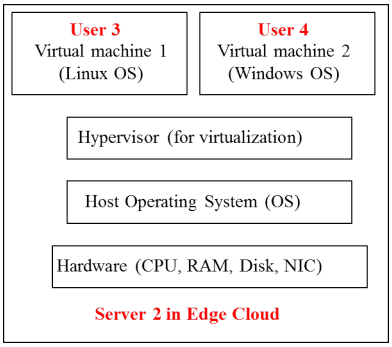


Fig. 4: Virtual Machines (VMs) service

for private IP address allocations to users' own PCs and hand-held devices, as well as to the servers within the edge-cloud clusters. Software for login management, such as OpenLDAP, is required to support users. Security systems, such as firewalls, intrusion detection systems, virtual private network (VPN) servers, can be purchased as standalone appliances. But Network Function Virtualization (NFV) [15], with its promise of capex and opex savings, is enabling many of these network functions to be implemented in software for execution on commodity servers. OpenDNS service can be used to store resource records for the few servers that will need static public IP addresses. For example, if the edge cloud offers web hosting services, then the computers on which the web server software, such as Apache http server, is executed will need to be assigned static public IP addresses, and information about their domain names will need to be added to the DNS system. Given the ubiquitiy of free commercial email services, such as gmail and yahoo mail, SMTP servers may not be needed in such edge clouds.

Fault management, Configuration management, Accounting, Performance monitoring, and Security (FCAPS) systems are an indispensable part of any network deployment. There are open-source tools for these functions. Without these management-plane systems, services on the edge-cloud will not be trustworthy, which could lead to user frustration and disillusionment.

As examples, we describe two of these functions: authentication, which is part of security, and accounting. For *authentication*, we propose the use of Shibboleth software [16], which is federated identity management system. For example, the NSF Global Environment for Network Innovations (GENI) [17] testbed uses Shibboleth. Researchers in a federation called InCommon [18] operated by Internet2, can use their own university authentication system to access GENI testbed resources. Currently, the GENI testbed has 58 computer racks, located at various universities across the US, and these resources are shared by 9000 users [19]. GENI requires a project leader, who is typically a faculty member or senior researcher, to first submit an application. If approved, the project leader then creates a project, and students can submit requests to join a project, which results in an automatic email being sent to the project leader, who can then approve/disapprove the request. A similar model can be used in communities with the project leader being replaced by an Account holder, who is head of a household. Members of the household can then submit requests to join the household account. This distributed management of responsibilities works well with NSF GENI, Chameleon, CloudLab and other such projects. It is well suited for SASEC.

Accounting will be essential for sustainability, which is discussed in Section VI. Software is required to collect measurements of usage of edge-cloud resources on a per-account/per-user basis. These measurements will be used in the development of a sustainable economic model to set pricing for the four types of services.

III. SASEC usage in enterprises

The major advantage of SASEC for enterprises is cybersecurity. The cyber attack surface is vastly reduced with SASEC since the number of user-owned desktops and laptops will be smaller. For example, if an enterprise user clicks on a URL in a phishing email, any executable that is downloaded to the edge-cloud server can be first tested in a separate VM to check for malware. In contrast in today's PC-at-the-desktop solution, since users have root access to their PCs, downloaded malware in zero-day attacks gets installed on the PCs, allowing for those PCs to be recruited into botnets for other attacks. Also concerns about users upgrading their OS and applications to patch security vulnerabilities are eliminated with SASEC.

Enterprises, especially university campuses, may not have the network bandwidth constraint for KVM-terminal connectivity in offices since most campuses have Ethernet copper, and in some cases fiber, connectivity. However, wireless support for DCDs will still be required. Consider a university classroom with 200 students, all carrying a DCD instead of a laptop. If they simultaneously try to connect to web browsers or their email clients running on edge-cloud servers, the amount of bandwidth required could be on the order of Gb/s. IEEE 802.11ac and newer 5G technologies will be required to support SASEC deployment in large enterprises.

The sustainability problem faced with smart-city deployments for disadvantaged communities is also not likely to be problem in enterprises. As noted in Section I, significant HR savings are possible by eliminating the need for a large number of desktops to be maintained in departments and divisions within the organization. Nevertheless, SASEC represents a significant change in the type of services an Information Technology (IT) division of an enterprise offers today. Most enterprises do not maintain edge clouds that offer their employees Virtual Machines (VMs). But some research universities maintain High-Performance Computing (HPC) clusters with support for VMs. Given the presence of commercial cloud providers, such as Amazon Web Services (AWS), HPC-computing managers at universities often face a challenge with designing sustainable pricing models for their services. Lessons learned from this community can be applied by enterprise IT divisions to support SASEC deployments.

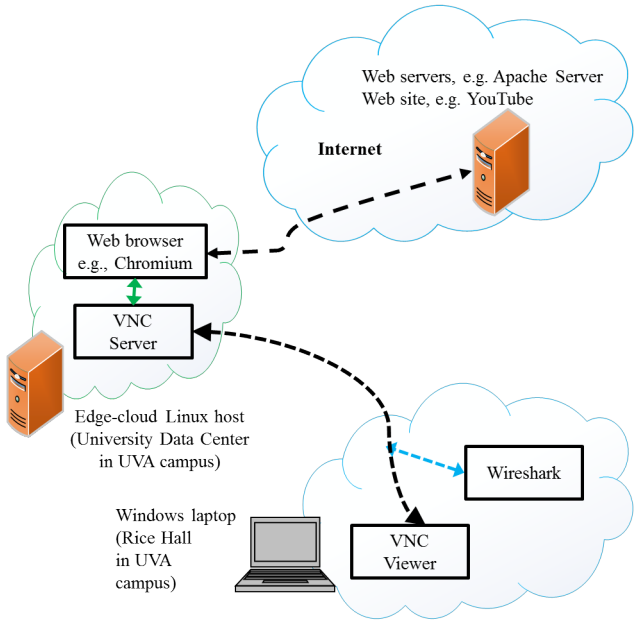


Fig. 5: Experimental setup; UVA: University of Virgina

IV. Experiments

Two experiments were conducted. The purpose of the first experiment was to estimate the amount of bandwidth (bits/s) required between the edge cloud and the KVM terminals for video. The purpose of the second experiment was to determine the impact of network delay (latency) on the quality of video when viewed on KVM terminals.

Section IV-A describes the experimental setup. Section IV-B describes and presents results for Experiment 1, while Section IV-C describes and presents results for Experiment 2.

A. Experimental setup

Fig. 5 illustrates our experimental setup. We used a Linux host located at our University Data Center (UDC) to emulate an edge-cloud server, and a Windows laptop at Rice Hall to emulate a KVM terminal. The distance between UDC and Rice Hall is roughly 2 miles, and there are two IP routers and multiple Ethernet switches in between the laptop and the Linux host. CentOS 6.7 with kernel version 2.6.32 was run on the Linux server, and Windows 7 was run on the laptop.

To the best of our knowledge, there is no mechanism to directly transfer the monitor output from the Linux host at the UDC to a KVM terminal at Rice Hall across our campus Ethernet/IP network. Therefore, we used a Windows laptop at Rice Hall, ran a Virtual Network Computing (VNC) client [20] on this laptop and a VNC server on the Linux host. VNC uses the Remote Frame Buffer (RFB) protocol [21] to transport video from the server to a remote monitor, and receive input from a remote keyboard and mouse. In a real SASEC deployment, dedicated hardware, as described in Section V, is required since DCDs, lacking processors and operating systems, do not run TCP/IP required for VNC.

Next, we installed the Chromium web client on the Linux host at the UDC, and Wireshark on the Windows laptop to capture the RFB packets sent back and forth between the laptop at Rice Hall and Linux host at the UDC.

For the second experiment, we installed the MobaXterm [22] client on the Windows laptop.

B. Experiment 1: Bandwidth requirements for video

Execution steps: First, we connected the VNC client on the laptop to the VNC server on the Linux host. Then we started packet capture in Wireshark at the laptop. Next, from the Chromium web client on our Linux host, we connected to a YouTube server and played a video. The video was viewed on the laptop. After a short duration, we stopped the video on the Linux host, and the Wireshark packet capture on the laptop, and analyzed the packet capture (pcap) file.

The *input parameter* that was varied in this experiment was video resolution. The YouTube website allows the user to select the video resolution. We ran the experiment with the following video-resolution settings: 360p (SD VGA, 480 × 360), 480p (SD VGA, 640 × 480), 720p (HDTV, 1280 × 720) and 1080p (HDTV, 1920 × 1080).

The *output measure* computed from pcap-file analysis was bandwidth used by the VNC client-server connection in the upstream (laptop to Linux host) and downstream (Linux host to laptop) directions.

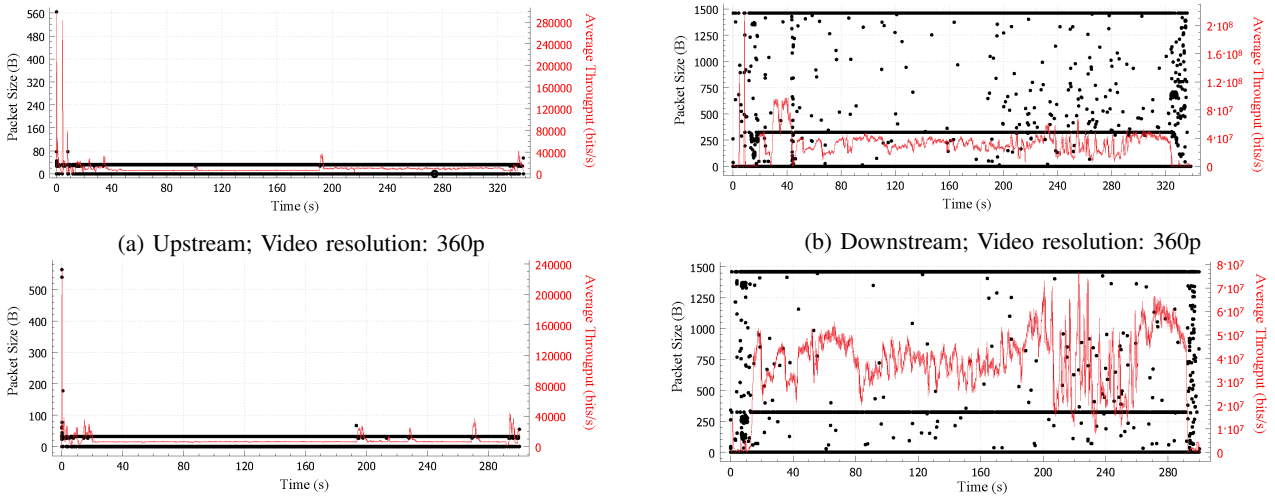
Results: Fig. 6 shows the throughput (right y-axis) of the VNC flows in the upstream and downstream directions for the two extreme settings (360p and 1080p) of video resolution. Since only keyboard and mouse input is carried in the upstream direction, the throughput in this direction is small. However, in the downstream direction, the throughput, and hence the required bandwidth, is significant for high video resolutions, sometime reaching more than 70 Mbps. Compression is being used on the VNC connection because the 1080p color video at 60 Hz requires 2.9 Gb/sec if it is uncompressed.

The left y-axis shows the packet sizes. Most of the packets in the upstream direction are small, with TCP acknowledgments having no payload (and hence are reported to have 0 packet size). In the downstream direction, the packet size distribution is trimodal, with dominant values of 0, 324, and 1460 bytes.

Average throughput	Video resolution			
	360p	480p	720p	1080p
Upstream (kbps)	6	6	7	8
Downstream (Mbps)	35	38	43	49

TABLE I: Required bandwidth

Table I shows the average bandwidth requirement for different video resolutions. If a single KVM terminal requires 50 Mbps, the amount of bandwidth required on a wireless network between the basement of a building and the users in apartments as shown in Fig. 1 is significant. If 50 households out of say 500 apartments in a building are simultaneously active,



(a) Upstream; Video resolution: 360p

(b) Downstream; Video resolution: 360p

(c) Upstream; Video resolution: 1080p

(d) Downstream; Video resolution: 1080p

Fig. 6: Packet size and Throughput; Upstream: laptop to Linux host; Downstream: Linux host to laptop

Input parameter					
Added latency (ms)	0	10	50	100	500
Output measures					
`ping` delay (ms)	1	10	50	100	500
`ssh`-typing delay (ms)	2.18	12.07	52.66	102.22	502.57
Subjective `ssh` quality	Excellent	Excellent	Excellent	Acceptable	Unacceptable
Subjective video quality	Excellent	Excellent	Acceptable	Unacceptable	Unacceptable

TABLE II: Impact of network latency on ssh and video quality

then IEEE 802.11ac will be required for the WiFi network. Another implication is that the KVM terminals in homes need an external device that offers compression service, which is part of our hardware design (see Section V).

C. Experiment 2: Impact of network latency

In this experiment, we used the Linux `traffic control (tc)` utility at the edge-cloud host to deliberately add packet delay to outgoing packets. The amount of delay added was varied to measure the impact of latency on video quality and on remote `ssh`.

Execution steps consist of the following:

1) Connect the VNC client on the laptop to the VNC server on the Linux host.
2) Run the `tc` utility at the Linux host to modify the added delay for VNC packets.
3) Execute the `ping` command on the laptop to measure the round-trip delay to the Linux host.
4) Start packet capture in Wireshark at the laptop to collect only `ssh` packets.
5) Start MobaXterm client on the laptop and remote login via `ssh` into the Linux host.
6) Use the Chromium web client to access a video on Youtube with the video resolution set to 1080p.
7) Stop the video, packet capture and analyze the data.

The *input parameter* that was varied in this experiment was the added `tc` delay for VNC packets at the Linux host. The values used were 0ms, 10ms, 50ms, 100ms, and 500ms.

Four *output measures* were used in the experiment: `ping` delay, `ssh`-typing delay, subjective `ssh` quality, and subjective video quality. The `ping` command sends Internet Control Message Protocol (ICMP) packets directly over IP, and is used to verify that the `tc`-utility set value took effect. The `ssh`-typing delay was measured by taking the time difference between the outgoing packet carrying a typed-character from the laptop to the Linux host and the incoming packet carrying the same typed-character back from the Linux host to the laptop. The average values were computed for 10 packets within one `ssh` session. In all cases, the standard deviation was small.

The subjective quality assessments were not rigorous since we do not as-yet have the federal-government Institutional Review Board (IRB) clearance that is required for experiments involving human subjects. Instead the reported values are assessments of authors of this paper. Three levels were used: Excellent, Acceptable, and Unacceptable. Methods for translating Quality-of-Service (QoS) into Quality-of-Experience (QoE), such as those described by Chen et al. [23], will be used in future work.

Results: Table II shows the impact of network latency on the four output measures. These are preliminary results, but

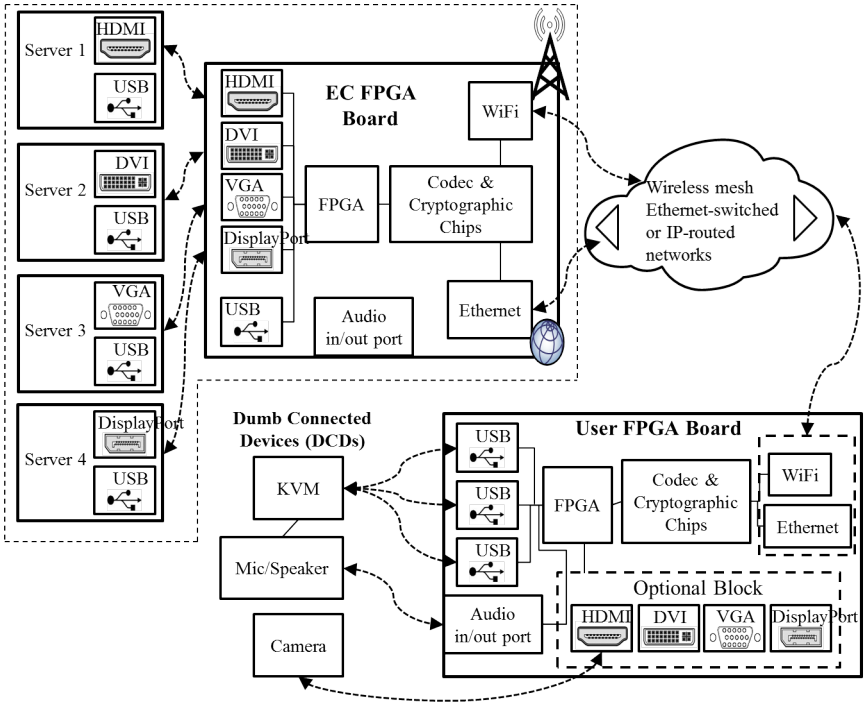


Fig. 7: Hardware design for EC-DCD network

they offer insights into how latency impacts QoE perceived by users. It appears that edge clouds are required for low propagation delays, unless a commercial cloud provider has a data center located within a round-trip time of 10 ms.

V. Hardware design

Fig. 7 illustrates a hardware design for connecting the edge cloud to the DCDs. Edge-cloud (EC) servers are shown to have HDMI, DVI, VGA, DisplayPort and USB ports. Different variants of the EC FPGA board can include different types and numbers of these ports. The FPGA can be used for the EC-DCD protocol, e.g., a simplified version of the RFB protocol, as well as for video coding [24] and encryption [25]. An alternative option is to use dedicated ASICs for video coding and encryption. Audio input/output ports are implemented on the EC FPGA board and will be connected to the corresponding ports on the servers. In addition, the EC FGPA board has WiFi and/or Ethernet for communication with the DCDs.

At the user end, a User FPGA board has WiFi and/or Ethernet interfaces, and USB interfaces to connect to KVM terminals. Optional video interfaces such as HDMI, DVI, VGA and DisplayPort can also be made available on the User FPGA board. Audio input/port ports will be used to connect microphones and speakers at the user end. As with the EC FGPA board, the FPGA on the User FPGA board can implement the protocol, video coding, and encryption/decryption functions, or ASICs can be used for the latter two functions.

Security is important on this EC-DCD network to protect the keyboard input from users to the edge cloud, and frame-buffer information that can carry sensitive data from the edge-cloud servers to the users. Video encoding/decoding is required since uncompressed video signals, especially at high resolution, require large amounts of bandwidth.

VI. Sustainability

The three dimensions of sustainability [26] are: (i) environmental dimension, (ii) economic dimension, and (iii) social dimension.

For the *environmental dimension*, we consider energy efficiency. The shared edge-cloud infrastructure of SASEC makes it easier to implement smart power management techniques. For example, CPU load of the edge-cloud servers can be monitored, and during periods of low load, user sessions can be concentrated to a few servers (if it possible to do so without impacting performance), which would allow for other servers to be powered off. Day-night usage patterns can also be leveraged.

With regards to the *economic dimension*, we raise the question of who owns and operates the edge clouds in the SASEC solution. Commercial cloud and Internet service providers have typically bypassed such communities for economic reasons. For example, Verizon's FiOS network bypassed so many households in New York City (NYC) that NYC government has publicly raised the possibility of litigation [27]. One option is for communities to self organize, raise the capital required for the initial deployment of the edge cloud hardware and software, and then create a sustainable flow of income from the participating households to maintain operations. Such community based efforts have succeeded as reviewed in Section VII, but it could be challenging in some cases as civic engagement in disadvantaged communities is

typically low [28]. Another solution is for smart cities to make the investment and deploy and operate edge clouds in disadvantaged communities. Increased taxes raised from the communities through increases in employment and household income enabled by Internet access [7] should be considered in economic models.

While smart cities can take the lead, for the *social dimension* of sustainability, democratic governance is essential, i.e., the community served by the edge cloud should be invested in its success. Democratic governance requires transparency and responsiveness to users by the provider [29]. One of the drawbacks of the SASEC architecture is that users with just a KVM terminal have less freedom in choosing software. Decisions on what hardware and software licenses to purchase, whether to use community members for operating the edge cloud or to outsource edge-cloud operations, should involve the user community. Also, the user community should be involved in setting pricing for the services. Households should be billed based on the services used, although perhaps at subsidized levels initially. For example, households who own their own PCs may purchase only the Internet access service of the edge cloud. VM users could be charged more than application users. The user base size should be controlled carefully. The larger the edge cloud, the lower the costs, but scale should not be increased without consideration of delay and bandwidth performance.

VII. Related work

Underserved communities around the world are deploying their own networks. For example, Guifi WiFi network was built in a rural area in Catalonia, Spain [30] because, as noted the paper: "A significant part of the population, especially in Catalunya, feels a deep rooted resentment towards Telefónica, Spain's incumbent operator." A WiFi mesh network was deployed in Mankosi, a rural, impoverished part of South Africa [31]. A 2016 paper [32] describes a Community-Lab Testbed effort, which aims to provide solutions to encourage the adoption of community networks. This paper also provides examples of collaborative federations of microISPs formed in the United Kingdom, France, South Africa, Mexico and India, all of which afford users and citizens greater voices in the network and make microISPs more responsive to their needs.

Another track of work related to this paper is the study of thin clients. A 1997 article [33] discussed "network computers," which was a term popularized by Oracle and Netscape, while Microsoft and Intel were pushing the desktop PC approach. In 2005, the THINC [4] architecture was proposed for thin clients. Another 2005 paper [34] described skinny devices without operating systems called Sun Ray clients, which are still available from Oracle. A 2008 performance study [35] of the quality of experience for users of thin clients was presented for Microsoft office applications. The setup included a Windows Terminal Server (WTS) to run Microsoft Office 2003 products, and Citrix Presentation Server 4.0 was used to make these applications accessible by thin-client users. A 2016 paper developed CloudBrowser 2.0 [36], which is just a rendering and I/O module, while the presentation state of a user's browsing session is maintained on the web server. This browser is comparable to thin clients. SASEC differs from these thin-client solutions in that DCDs are dumb, i.e., they have no processors, a necessary feature for the cybersecurity advantage.

VIII. Conclusions

Our proposed Secure Affordable Sustainable Edge Clouds (SASEC) architecture offers two key advantages: (i) cost and (ii) cyber-security. SASEC proposes that computers be grouped together into an edge cloud, along with applications, which are then shared by a large number of households, or users in a large enterprise. Each household or user has a simple KVM terminal. These terminals lack processors, and hence operating systems and software. Removing the CPU from the edge devices leads to increased *cybersecurity* for all by vastly reducing the attack surface. Centralizing the processing hardware and software licenses into the edge cloud allows the community to *share the cost burden*, use smart power management techniques to reduce energy usage, keep up with security updates, and offer centralized intrusion detection and prevention for better cyber-security. Finally, significant HR savings are possible if SASEC is adopted in large enterprises, such as universities. SASEC offers a means to broaden the *sustainability* goals to beyond energy efficiency by supporting greater livelihood opportunities and opening up opportunities for democratic governance to shape client-provider relationships. Experiments demonstrated that for video-intensive applications, even with compression, about 50 Mbps is required per user for 1080p video. Also, latency experiments demonstrated the need to make the round-trip delay between the user KVM terminals and the edge-cloud servers less than 10 ms. Finally, a hardware design was proposed to carry video signals from the edge cloud to the KVM terminals, and keyboard and mouse input from the KVM terminals to the edge cloud via Ethernet/WiFi using FPGAs to implement the protocols, video compression and encryption/decryption.

IX. Acknowledgment

The University of Virginia part of the work was supported by NSF grants ACI-1340910, CNS-1405171, CNS-1531065, and CNS-1624676.

References

[1] Scott M. Stringer, Office of the New York City Comptroller, Bureau of Policy and Research. (Dec. 2014) Internet inequality: Broadband access in NYC. https://comptroller.nyc.gov/wp-content/uploads/documents/Internet_Inequality.pdf.

[2] Y. Benkler, *The Wealth of Networks: How social production transforms markets and freedom*. Yale University Press, New Haven, CT, 2006.

[3] (Sept. 2014) Connected Texas: Broadband and Education Connecting Students in Texas. http://www.connectedtx.org/sites/default/files/connected-nation/Texas/files/tx_education_report.pdf.

[4] R. A. Baratto, L. N. Kim, and J. Nieh, "Thinc: A virtual display architecture for thin-client computing," *SIGOPS Oper. Syst. Rev.*, vol. 39, no. 5, Oct. 2005.

[5] Computer Viruses: What is the safest way to detonate an executable? https://www.quora.com/Computer-Viruses-What-is-the-safest-way-to-detonate-an-executable.

[6] The Economist. (Mar 9, 2013) Peer-to-peer rental: The rise of the sharing economy; on the Internet, everything is for hire.

[7] Connect Michigan: Broadbands Economic Impact in Michigan. http://www.connectmi.org/sites/default/files/connected-nation/Michigan/files/mi_economic_impact_final.pdf.

[8] DDoS attack that disrupted internet was largest of its kind in history, experts say. https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet.

[9] Image of three display monitors. https://cms-images.idgesg.net/images/article/2013/04/displayport_productivity_1160-100033745-orig.jpg.

[10] Narrow Bezel Ultra Thin Bezel Video Wall Display With Samsung LCD. http://www.360digitalsignage.com.

[11] Nvidia GeForce GTX 650 Ti. https://cms-images.idgesg.net/images/article/2015/05/geforce-gtx-650-ti-ports-100586024-orig.jpg.

[12] Reverse KVM Switches. [Online]. Available: http://www.adder.com/products/categories/reverse-kvm-switches

[13] G. P. Johnson, G. D. Abram, B. Westing, P. Navr'til, and K. Gaither, "Displaycluster: An interactive visualization environment for tiled displays," in *2012 IEEE International Conference on Cluster Computing*, Sept 2012, pp. 239–247.

[14] ADDERLink XD150FX. http://www.adder.com/products/adderlink-xd150fx.

[15] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 236–262, Firstquarter 2016.

[16] Shibboleth. https://shibboleth.net/.

[17] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "GENI: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, pp. 5 – 23, 2014.

[18] Incommon. [Online]. Available: https://www.incommon.org/

[19] M. Berman. GENI Update and Transition, Dec. 12, 2016. http://www.geni.net/nice2016/docs/2016-12-12BermanWelcomeandUpdate.pdf.

[20] CentOS. VNC (Virtual Network Computing). https://wiki.centos.org/HowTos/VNC-Server.

[21] T. Richardson and J. Levine, "The Remote Frame Buffer Protocol," RFC 6143 (Informational), Internet Engineering Task Force, Mar. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6143.txt

[22] MobaXterm X server and SSH client. http://mobaxterm.mobatek.net/.

[23] Y. Chen, K. Wu, and Q. Zhang, "From QoS to QoE: A Tutorial on Video Quality Assessment," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 1126–1165, Secondquarter 2015.

[24] J.-M. Cloquet. Barco-Silex, Enabling Video Equipment to Compress and Transport High-Quality Content. https://www.xilinx.com/publications/prod_mktg/applications/barco-silex-xilinx-backgrounder-FINAL.pdf.

[25] A. M. Deshpande, M. S. Deshpande, and D. N. Kayatanavar, "FPGA implementation of AES encryption and decryption," in *2009 International Conference on Control, Automation, Communication and Energy Conservation*, June 2009, pp. 1–6.

[26] United Nations Environment Programme (UNEP). The three dimensions of sustainable development. http://web.unep.org/ourplanet/march-2015/unep-work/three-dimensions-sustainable-development.

[27] The New York Times (P. McGeehan). (2015) As service gaps remain, city says verizon broke promise on fios. https://www.nytimes.com/2015/08/27/nyregion/new-york-city-and-verizon-battle-over-fios-service.html.

[28] T. Ali, DNAinfo. (2016) How every New York City neighborhood voted in the 2016 presidential election. https://www.dnainfo.com/.

[29] R. W. Foley, M. J. Bernstein, and A. Wiek, "Towards an alignment of activities, aspirations and stakeholders for responsible innovation," *Journal of Responsible Innovation*, vol. 3, no. 3, pp. 209–232, 2016.

[30] M. Oliver, J. Zuidweg, and M. Batikas, "Wireless commons against the digital divide," in *2010 IEEE International Symposium on Technology and Society*, June 2010, pp. 457–465.

[31] C. Rey-Moreno, Z. Roro, W. D. Tucker, M. J. Siya, N. J. Bidwell, and J. Simo-Reigadas, "Experiences, challenges and lessons from rolling out a rural WiFi mesh network," in *Proceedings of the 3rd ACM Symposium on Computing for Development*, ser. ACM DEV '13. New York, NY, USA: ACM, 2013, pp. 11:1–11:10.

[32] L. Navarro, R. B. Vinas, C. Barz, J. Bonicioli, B. Braem, F. Freitag, and I. V. i Balaguer, "Advances in wireless community networks with the community-lab testbed," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 20–27, July 2016.

[33] S. J. Vaughan-Nichols, "To nc or not to nc?" *ACM netWorker*, vol. 1, no. 1, Mar. 1997.

[34] A. Schwartz and C. Guerrazzi, "You can never be too thin: Skinny-client technology," in *Proceedings of the 33rd Annual ACM SIGUCCS Conference on User Services*, ser. SIGUCCS '05. New York, NY, USA: ACM, 2005, pp. 336–337.

[35] B. Staehle, A. Binzenhoefer, D. Schlosser, and B. Boder, "Quantifying the influence of network conditions on the service quality experienced by a thin client user," in *14th GI/ITG Conference - Measurement, Modelling and Evalutation of Computer and Communication Systems*, March 2008, pp. 1–15.

[36] X. Pan and G. Back, "Rich cloud-based web applications with cloud-browser 2.0," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, ser. SAC '16. New York, NY, USA: ACM, 2016, pp. 758–765.