Socio-technical Approach to Engineer Gigabit App Performance for PhysicalTherapy-as-a-Service

R. Bazan-Antequera, P. Calyam, D. Chemodanov,
W. de Donato[‡], A. Mishra, A. Pescape[‡], M. Skubic
University of Missouri-Columbia, USA
{*rcb553, calyamp, dycbt4, skubicm, akmm94*} *@missouri.edu*, University of Napoli "Federico II", Italy[‡],
{*walter.dedonato,pescape*}*@unina.it*

***Abstract*—The deployment of Gigabit Apps owing to their high-bandwidth and low-latency nature pushes the limits of today's end-to-end networking, and reveals new bottlenecks at multiple layers of networking, virtualization, application and user experience. In this paper, we use an exemplar smart health related Gigabit App use case viz., PhysicalTherapy-as-a-Service to show how a multi-layer instrumentation approach of measurement points was critical to successfully deploy our lab-tested App out to residential homes with Google Fiber connections. The salient instrumentation strategies involved an organized co-design method between the App Developer and Network Engineer roles, and a multi-domain network performance monitoring featuring perfSONAR extensions, both of which were realized through our Narada Metrics framework. Our instrumentation strategies engendered a "socio-technical tool" for co-ordination between multi-layer stakeholders in identifying and overcoming the intertwined bottlenecks, and in tuning the App performance. Our results highlight the new instrumentation and measurement challenges to foster multi-layer stakeholder collaboration, and provide rare insights to the budding Gigabit App developer community for performance engineering their Apps to serve residential users.**

***Index Terms*—Smart Health App, Gigabit Access Networks, Remote Physical Therapy**

I. Introduction

Ultra high-speed broadband networks are becoming economically feasible and increasingly available to residential user communities, especially with industry offerings (up to 1 Gbps Internet speeds) such as Google Fiber [1] from Google, Gigabit Pro [2] from Comcast, and GigaPower [3] from AT&T. Such an access gives users the opportunity to benefit from new Gigabit Apps in areas such as smart health, public safety, immersive shopping and distance education, in addition to the popular on-demand television and multi-player gaming entertainment purposes. Moreover, cities are increasingly making investments in large data centers and fiber assets to develop broadband-based economies that attract high-tech companies and high-paying jobs [4]. This emerging trend has given rise to a budding Gigabit App developer community that is growing through initiatives such as US Ignite, Mozilla Ignite and City-hosted GENI Racks [5] that are being supported by the National Science Foundation, and private funding sources.

However, the deployment of Gigabit Apps owing to their high-bandwidth and low-latency nature pushes the limits of today's networks and operating systems, and requires new

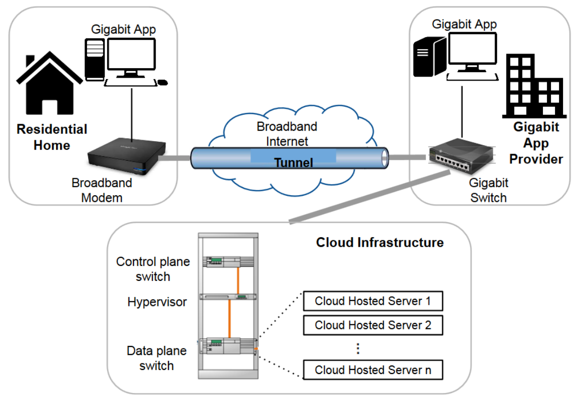

Fig. 1: Novel peer-to-peer Gigabit App deployment infrastructure between App provider (i.e. Physical therapist) and home user (patient) connected to cloud resources.

peer-to-peer or client-server architectures that involve cloud-hosted services. To accommodate scalable management of these services, computer and network virtualization technologies have to be adopted across the end-to-end infrastructure. Fig. 1 shows a novel peer-to-peer Gigabit App deployment infrastructure where the App provider's (e.g., Hospital, University) personnel or local resources need to be connected with a residential home desktop with e.g., Google Fiber connection for functioning of a Gigabit App. A cloud infrastructure integration may also be essential particularly for the large amounts of data storage associated with the Gigabit App, and for corresponding fast processing of the data to provide advanced analytic or multi-user collaboration support via HD videoconferencing.

Moreover, when sensitive information such as personal data is exchanged, overlay networks and tunnels may need to be setup between the home, provider and cloud data center sites for secure App services inter-communication. Thus, the distributed as well as data-intensive nature of the Gigabit Apps, along with high-performance expectations on the intermediate multi-domain network segments can be harder to satisfy due to the network virtualization, which adds complexity and introduces new bottleneck factors. This in turn demands new approaches/tools to monitor cloud-hosted services and tune App performance to foster suitable resource provisioning, and rapid bottleneck troubleshooting to ensure satisfactory user experience [6].

In this paper, we use an exemplar smart health related Gigabit App use case viz., PhysicalTherapy-as-a-Service to show how a multi-layer instrumentation approach of measurement points was critical to successfully deploy our lab-tested 'Interactive Interface for Physical Therapy' (PTaaS App) [7] out to residential homes with Google Fiber connections. The PTaaS App was developed in collaboration with the Center for Eldercare and Rehabilitation Technology at University of Missouri (MU) as part of a NSF-supported US Ignite project to connect a remote physical therapist at a clinic to a senior at home, in order to leverage ultra high-speed broadband to deliver in-home, personalized telehealth services. The PTaaS App is built to use Microsoft Kinect motion sensing capabilities for wellness coaching exercises and outputs a peak data rate of $\approx$ 200 Mbps in a controlled setting in a lab environment, which includes several data streams (i.e., RGB, depth, skeletal and audio) and requires large volume of storage (i.e., several GB of data for a simple exercise activity). The low-latency requirement stems from the fact that the therapist needs to be able to confidently assess whether non-ideal performance in the exercise forms of a senior is being impacted due to lag in network communications for the data-intensive interactive session, or in fact is due to the physical and cognitive limitations of the senior owing to aging.

The measurement and troubleshooting work presented in this paper was initiated when we deployed the lab-tested PTaaS App out to the senior homes with Google Fiber in Kansas City with the therapist located at the MU clinic in Columbia. Upon deployment, we immediately found the PTaaS App to be totally unusable by the users due to serious performance problems (i.e., audio and video impairments, interaction lag, frequent disconnections). We categorize this experience akin to a 'murder' of the PTaaS App performance and usability, and there were a number of murder suspects at the network, virtualization, and application levels.

In order to solve this murder mystery through a 'Sherlock Holming' styled crime investigation, we developed multi-layer instrumentation strategies to obtain performance visibility of the end-to-end system components. The first strategy we describe involves an organized co-design method between the App Developer and Network Engineer roles, which enabled us to analyze the application and network performance jointly through passive monitoring of the PTaaS App streams at the end-sites. The other strategy we developed was to use one of the widely-deployed multi-domain network monitoring solutions in the world viz., perfSONAR [8] along with custom-developed extensions available through our Narada Metrics framework [9]. The extensions allowed us to correlate the passive monitoring at the edges with active monitoring in the intermediate network hops at strategic locations within the MU campus, regional (MOREnet) and last-mile (Google Fiber) network segments.

The benefits of our multi-layer instrumentation strategies for our Sherlock Holming became evident when we were able to solve the crime by using a methodical troubleshooting approach of studying the intertwined murder suspects at the network, virtualization, and application levels. Our instrumentation strategies engendered a "socio-technical tool" for coordination between multi-layer stakeholders in identifying and overcoming the intertwined bottlenecks, and in tuning the App performance. The stakeholders included the App developers, performance engineers using Narada Metrics, network engineers of MU campus, MOREnet and Google Fiber, as well as the Brocade virtualization technology vendor. Our results highlight the new instrumentation and measurement challenges to foster multi-layer stakeholder collaboration that is essential throughout the Gigabit App life-cycle of design, development, testing, deployment and operations. The collaboration is needed mainly due to the fact that existing network infrastructures are primarily designed to support popular Apps such as Netflix, Skype and Dropbox, and need various parameter tuning at multiple layers (even at e.g., hardware selection and clock synchronization) to meet new Gigabit App demands.

The paper remainder is organized as follows: Section II describes the PTaaS App use case and the user experience 'murder'. Section III presents our multi-layer instrumentation strategies. Section IV presents the Sherlock Holming to study the murder suspects at the network, virtualization, and application levels. Section VI concludes the paper.

II. PTaaS App Use Case and Deployment

A. PTaaS App Overview

The PTaaS App is more than a typical videoconferencing application because it integrates 2D video feeds between the therapist and senior along with 3D sensing data to provide an immersive experience for in-home personalized physical therapy sessions. The current PTaaS App implementation uses Microsoft Kinect motion sensing capabilities and has been developed in C# language using the .Net Framework (4.5), the Microsoft Kinect SDK (1.8), and the Windows Presentation Foundation (WPF) library. It includes algorithms to calculate the bending/sway parameters from the 3D joint positions of the Kinect Skeletal tracking system. More specifically, the Sway and Joint alignment parameters are calculated to understand how good/bad the patients are in their postural balance while performing different physical therapy exercises as suggested by the therapist in a live interaction with the PTaaS App on both sides [10].

Two versions of the PTaaS App have been developed, one for the patient side, and the other for the therapist side. Despite each of them having a slightly different software configuration in order to suit user-specific perspectives and exercise activity tasks, the bandwidth consumption of the traffic streams on both sides is symmetrical. The therapist version has additional options such as session control, access to real-time network health information, voice commands control as well as access to real-time patient information e.g., exercise assessments and depth images. Moreover, both versions provide users with real-time access to skeletal data visualization, along with live audio and video conferencing capabilities.

B. Deployment Setup

We created a system design as shown in Figure 2 for a real deployment of the PTaaS App across the Internet to connect the patient side at a Kansas City home with the therapist side at MU clinic in Columbia. Owing to the fact that residential network connections of Google Fiber customers in Kansas

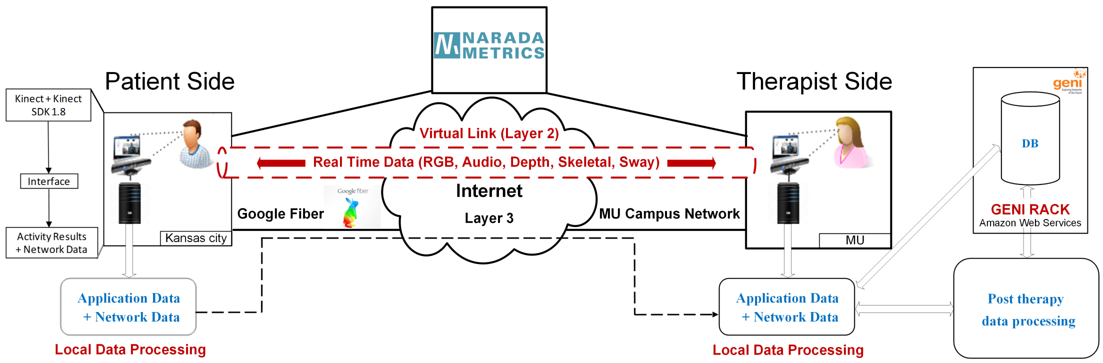

Fig. 2: PhysicalTherapy-as-a-Service System Setup: App communication is enabled through a virtual link between the patient and therapist sides.

City do not have public IP addresses (same is true for any other ISP case as well, as detailed in Section I), an overlay path using virtual link (Layer 2) technologies had to be setup for App services inter-communication with custom protocols. Moreover, PTaaS App requirements included that the therapist had to initiate several communications and the therapy sessions had to be secure end-to-end, because of which the virtual private network (VPN) tunnel option was considered. The overlay path allows for the fast data movement of video, audio, RGB, depth and skeletal data for real-time display of gait and other movement parameters at both sides.

On each side, the PTaaS App setup features a Kinect device along with a local computer that are mounted on a mobile cart that has a large HD display (specification: 1920x1080 px) showing the interactive interface. We intentionally used different specifications of the local computer on the therapist (Windows 7 64 bits, Intel Core $i5$ CPU, 4 GB RAM, 500 GB HDD, Gigabit NIC) and patient (Windows 7 64 bits, Intel Core $i7$ CPU, 8 GB RAM, 1 TB HDD, Gigabit NIC) sides to test the PTaaS App behavior in a heterogeneous environment with the patient having a slightly more powerful configuration. Additionally, we utilized the MU GENI Rack (configured with three virtual machines provisioned using the ESXi hypervisor) [11] to serve as the database of exercise activity reports, and also to host the peer-to-peer PTaaS App signaling coordination module. In order to guarantee effective App performance, both the patient and therapist sides had to be time-synchronized. To this end, we setup a Precise Time Protocol (PTP) [12] based synchronization client-server solution on the therapist and patient sides, respectively that was calibrated to produce accuracy at the level of $\approx$1 millisecond.

C. Laboratory Testing and Characterization

Once we had a minimal viable prototype of the PTaaS App, we conducted a series of controlled experiments in a laboratory environment at MU to study the App behavior, obtain a baseline for expected performance and understand the various factors that affect the baseline performance. Fig. 3 shows the empirical CDF plots of the bandwidth consumption of various stream types (audio, skeletal, color, depth) individually and in an aggregate as obtained from the Kinect API calls. We can see that a single PTaaS App session requires $\approx$ 200 Mbps end-to-end available bandwidth between therapist and patient sides to exchange the various data streams.

Moreover, given that the performance of the PTaaS App can be affected due to network health factors (e.g., delay, jitter, packet loss), we tested the controlled network path between the patient and therapist sides and confirmed expected and degraded behavior under good and bad network scenarios, respectively. Active measurement tools such as OWAMP and BWCTL available in perfSONAR were used in the network emulation tests between private IP hosts to verify the good and bad scenarios of network path status. In order to verify satisfactory user experience in our testing, we also had an approved Institutional Review Board (IRB) protocol that allowed us to collect user experience rankings from our MU physical therapist and human subjects in the widely accepted form of 'Mean Opinion Scores' that have a scale of 1 to 5, with 1 being Poor and 5 being Excellent. We particularly took precautions in using the PTaaS App with the recruited human subjects based on the guidance of our MU physical therapy expert to limit any risks such as physical or cognitive injury in the interactions with the PTaaS App.

D. First Deployment Experience

Once we had sufficiently tested the PTaaS App in our laboratory environment, we proceeded to deploy the system by connecting the MU therapist with 5 patients (healthy seniors who would benefit from PTaaS App delivered wellness coaching exercise activities) recruited from different Google Fiber locations in a particular Kansas City area. Upon deployment to the homes using a standard VPN solution (commonly used for security purposes), we immediately found the PTaaS App to be totally unusable by the users, and the user experience was extremely degraded due to serious performance problems (i.e., audio and video impairments, interaction lag, frequent disconnections). We then performed active measurements between the homes and MU clinic and obtained an example set of measurements reported in Table I that showed an unexpectedly low end-to-end available bandwidth speeds of $\approx$ 40 Mbps, along with $\approx$ 75% packet loss. We categorized our first deployment experience as a 'murder' of our PTaaS App, and found the need to identify the real suspect(s) amongst a number of possibilities at the network, virtualization and application levels.

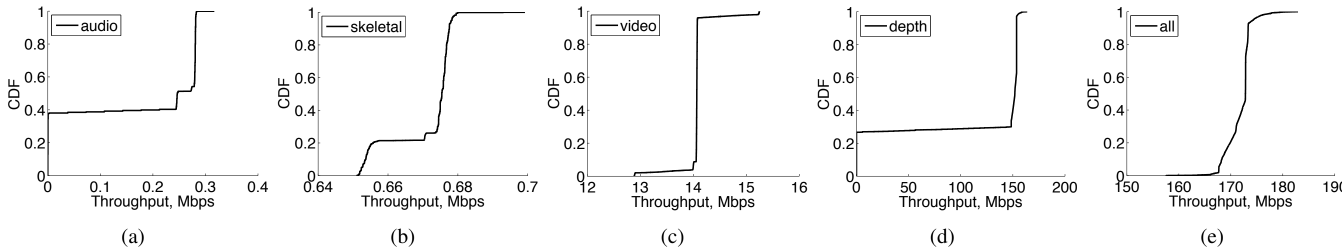

Fig. 3: Bandwidth consumption of different PTaaS data streams shown as empirical CDF plots: (a) audio stream; (b) skeletal stream that has the lowest bandwidth consumption; (c) video stream with medium bandwidth consumption; (d) depth data stream that shows the highest bandwidth consumption; (e) all streams together nearly require $\approx$ 200 Mbps end-to-end available bandwidth between provider and home sides.

TABLE I: Therapist side measurement sample collected with perfSONAR.

Activity under Test	Jitter [ms]	Packet Loss [%]	RTT [ms]
Single leg stance	14.53	**73.84**	30.67

III. Application and Network Measurement Co-Design

A. Motivation for Co-design

In order to solve the murder mystery observed during our first deployment experience, we decided to instrument the system with measurement points where possible, and at strategic locations across the end-to-end components. The current practice with perfSONAR for instrumentation in this context is shown in Fig. 4 (a), where the network measurements are performed by a Network Admin in isolation with the App over the common network APIs and infrastructure resources. The App measurements are used to indirectly inference network health, and the network measurements are used indirectly to assume App performance and user experience. To jointly and more effectively analyze the App and network performance trends, we took an approach shown in the Fig. 4 (b), which we hereto refer as the 'co-design method' that required us to engage both the App Developer and Network Engineer roles to integrate their views of performance. The co-design method was realized through our Narada Metrics framework, which provides a number of perfSONAR extensions such as: (i) *programmability* for conflict-free meta-scheduling of network-wide active measurements to cater to the App-specific monitoring objectives involving multiple active measurement tools [9], and (ii) *extensibility* that allows performance analysis for App-aware inferencing and drill-down of various network-wide metrics, in combination with 'custom metrics' specific to various App contexts.

B. perfSONAR Extensions in Narada Metrics

We used Narada Metrics' ability to instrument Measurement Point Appliances (MPAs) to perform on-going conflict-free active measurements with perfSONAR tools such as `iperf` and `ping` between the patient and therapist sides. The conflict-free active measurement data collection in Narada Metrics for tools such as e.g., `iperf` and `ping` tools is performed using a measurement scheduler that is aware of the measurement topology, and related measurement schedules. The measurement scheduling can be programmatically controlled (using RESTful API) by the App so that `iperf` tool tests that are computation and network resource intensive can be performed offline (e.g., before the use of a Gigabit App on a network path) in a periodic sampling manner with configurable inter-sample times, or conversely to initiate lightweight `ping` tool tests in-line with the App with a certain sampling rate. Due to the interoperable nature of Narada Metrics, we could also use Narada Metrics MPAs to test with other publicly accessible perfSONAR measurement points deployed across network domains (e.g., MOREnet, Internet2, Ohio State U.).

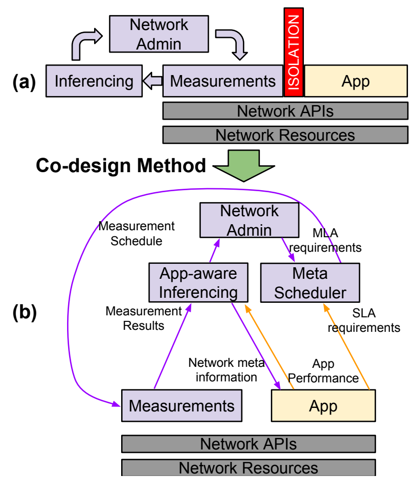

Fig. 4: Different approaches for App and Network Monitoring: (a) Isolated approach present in current perfSONAR practice, and (b) co-design method that demands measurement programmability and metric extensibility obtainable with Narada Metrics for a closely-integrated approach.

The custom metrics in our PTaaS App case within Narada Metrics related to the Kinect API calls for App-level performance status (e.g., upstream data transfer rate, downstream data transfer rate), which we collected using a passive measurement approach. Passive measurements were programmed to run in-line with the App traffic and were collected using the popular Wireshark packet capture tool, as well as the "Windows Network Interface" counters viz., (`BytesReceivedPerSec` and `BytesSentPerSec`). In order to keep passive measurements non-intrusive, we performed experiments and choose 10 seconds sampling rate setting within the App interface. Both the active and passive measurements were automatically collected using RESTful API calls and stored in JSON format. The RESTful API includes *publish* calls (from the PTaaS App to send raw measurements to Narada Metrics) and *subscribe* calls (from the PTaaS App to request processed/correlated measurements from the Narada Metrics) within a live physical therapy session. Finally, the collected measurements were analyzed

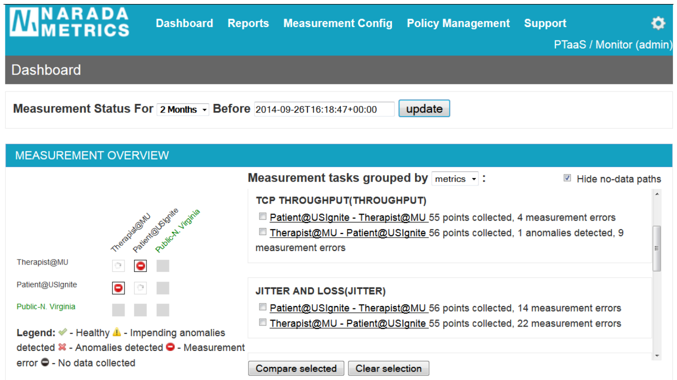

Fig. 5: Customized Narada Metrics Dashboard for PTaaS App performance visualization, and joint analysis of the App and network health metrics.

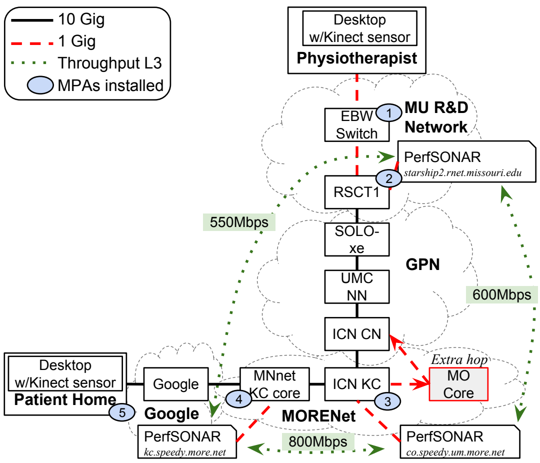

Fig. 6: Illustration of end-to-end 1 Gbps path across different ISPs responsible for PTaaS App data traffic forwarding between intermediate hops.

through a personalized Narada Metrics Dashboard shown in Figure 5 for the PTaaS App that we developed.

IV. Sherlock Holming PTaaS App Bottlenecks

A. *Murder at the Network Level*

Fig. 6 shows the end-to-end 1 Gbps (expected capacity) path between the therapist and patient sides along with the intermediate network hops belonging to the MU campus, regional (MOREnet) and last-mile (Google Fiber) ISPs. Communicating with MOREnet in our early troubleshooting stages helped us significantly because they connect the MU campus to national networks such as Internet2 as well as other commercial ISPs. Fortunately, we discovered that MOREnet had direct connectivity to Google Fiber network in Kansas City (KC), and their knowledge of both the MU network segment and Google Fiber network segment allowed us to obtain the intermediate hop sequences shown in Fig. 6.

We started troubleshooting at the network layer by first checking whether the access network performance was meeting expectations at both the access network sides. We performed measurements against publicly accessible perfSONAR MPAs at different locations and compared them with the measurements on the end-to-end network path between the therapist and patient sides. We found that the degraded network performance was found to be a common occurrence in both the cases, however we saw better performance going from the patient side to the therapist side than vice versa. We also investigated any obvious possible causes on both sides such as software misconfigurations in antivirus/anti-malware software, firewall rules, operating system patches and updates of the network interface card drivers. Additionally, we looked at obvious possible hardware misconfigurations in the Gigabit Ethernet ports in the desktops on the therapist and patient sides, as well as the network wiring cables.

When the performance problem remained unchanged, we shared the above set of observed measurement and system checking results with the MU campus networking, MOREnet and Google Fiber teams. In reaction to this issue notification, MOREnet immediately modified their forwarding setup by removing an extra hop (MO Core) in the end-to-end path and notified us that their new setup may improve our performance. They also helped us deploy 3 Narada Metrics MPAs (numbered 2, 3 and 4 in Fig. 6), and we deployed 2 additional Narada Metrics MPAs (numbered 1 and 5 in Fig. 6) to further ensure the intermediate network segments had expected performance (shown in green shaded boxes and arrows annotated with measured throughput values).

At the same time, MU campus network discovered that there was a duplex mismatch problem in the therapist building switch (EBW Switch). Half-duplex misconfiguration on one of the links was imposing the bottleneck by causing high packet loss rates from therapist to patient side. They then resolved the half-duplex misconfiguration issue by setting the appropriate network interface on the therapist side to be the desired full duplex configuration. Interestingly, the perfSONAR throughput measurements improved significantly i.e., to $\approx$ 600 Mbps on the therapist side at MU, almost by a factor of six as shown in Fig. 7.

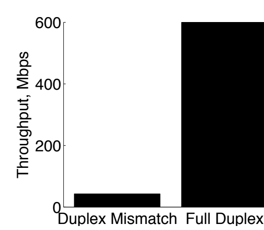

Fig. 7: Performance degradation by almost a factor of six due to duplex network mismatch in comparison with the desired full duplex condition.

Nevertheless, our troubleshooting at the network level prompted all of the intermediate ISPs to perform various modifications to tune the network setups, and we did find some improvement with perfSONAR throughput measurements, however we still were experiencing $\approx$ 40 Mbps PTaaS App throughput between the therapist and patient sides.

B. *Murder at the Virtualization Level*

In our troubleshooting efforts, we also had to deal with the virtualization layer, which enables a secure (private) point-to-point connection tunnel required for our PTaaS application as discussed in Section I. While setting up the tunnel, we chose to use the VPN encapsulation provided by the Brocade Vyatta vRouter [13] installed at both the therapist and patient sides. We opted for the vRouter software because it was the cheapest option (no need for additional hardware devices)

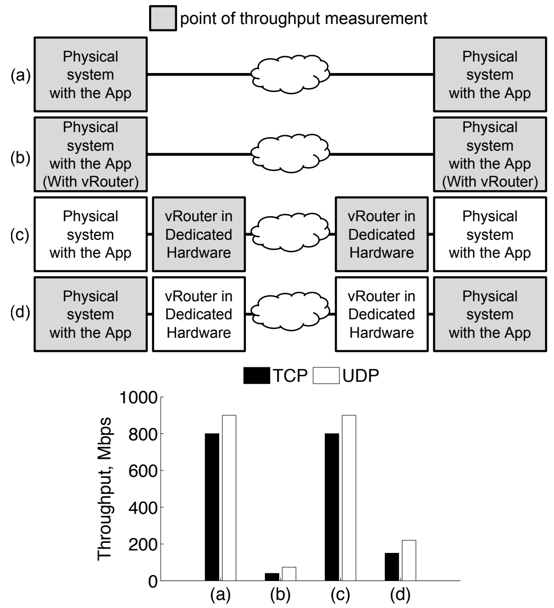

Fig. 8: vRouter experiments in emulated deployment representation: (a) throughput measurements between two Windows desktops to obtain baseline; (b) similar to previous experiment but with vRouter installed locally on the same desktop as the PTaaS App; (c) throughput measurements with vRouters running on dedicated hardware; (d) throughput measurements between two Windows desktops through vRouters running on dedicated hardware.

and easiest (simple configuration) to deploy a secure point-to-point connection tunnel. The vRouters were configured with OpenVPN to allow Layer 2 encapsulation over a Layer 3 communication channel, enabling both the senior and therapist desktops to use private IPs to communicate with each other.

Suspecting that the vRouter overhead affects the PTaaS throughput, we conducted a series of experiments and collected throughput measurements to understand the actual impact of virtualization over a high speed broadband connection with a Gigabit Apps that are shown in Fig. 8. In the case of Fig. 8 (a), we obtained a baseline by measuring throughput with perfSONAR between two desktops and obtained TCP throughput of $\approx$ 800 Mbps, and UDP throughput of $\approx$ 900 Mbps without the vRouter. Following this, we enabled the vRouter functionality between the two desktops and performed the measurement again and obtained TCP throughput of ≈ 40 Mbps, and UDP throughput of ≈ 45 Mbps as shown in Fig. 8 (b). Further investigation of packet capture analysis with Narada Metrics revealed an overhead caused by the virtualization process i.e., the end-system CPU and memory were becoming overloaded that clearly was impacting the throughput measurements. Upon consultation with Brocade, we decided to perform additional tests with vRouter using dedicated hardware on both sides. With the dedicated hardware configuration, we performed the measurement again and obtained TCP throughput of $\approx$ 800 Mbps, and UDP throughput of $\approx$ 900 Mbps between the measurement points of the dedicated hardware hosts as shown in Fig. 8 (c). Lastly, we repeated the perfSONAR measurement between the end-systems through the vRouters, thus involving Layer 2 over Layer 3 tunneling and obtained improved TCP throughput of ≈ 150 Mbps, and UDP throughput of ≈ 220 Mbps.

As a result, to eliminate the suspicions at the virtualiza-

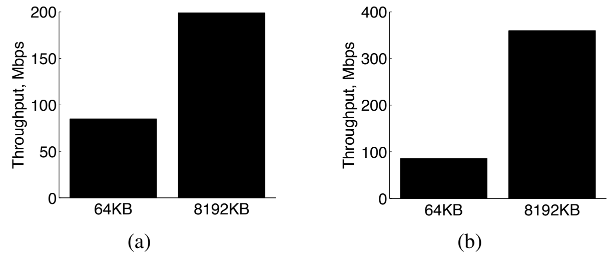

Fig. 9: Measurement results showing how TCP window size impacts throughput between two end-systems: (a) when VPN is used between patient and therapist desktops; (b) when data is sent directly through the desktops with public IPs.

tion level we ended using a dedicated physical VPN server deployed at the therapist side for a feasible (secure) solution to run our PTaaS App. At the same time, we have ended up using software VPN clients on patient sides to reach a more optimal solution in terms of cost and deployment complexity.

C. Murder at the Application Level

The earlier experiments on multi-domain networks and with the vRouter clearly indicated that they were no longer issues, and that our investigation had to focus on the application level. At this point, we started a troubleshooting process to evaluate how the TCP window size default configuration was affecting performance. Several packet capture traces were collected and the obtained TCP throughput measurements were analyzed using Narada Metrics, in conjunction with network measurements. Owing to the fact that our PTaaS App was developed for Windows 7 operating system, the default window size is 64 KB. The supported auto-tuning in Windows 7 should allow the window size to increase up to 16 MB. However, despite our tests that followed the TCP tuning process following guidelines in [14], we found that the auto-tuning was not occurring dynamically as expected.

By using network emulation between the end-systems with round-trip time of 5 ms (same value as in the real deployment between MU therapist and Kansas City home patient), we measured a maximum TCP throughput of $\approx$ 85 Mbps with the default TCP window size settings in Windows 7 over a single TCP connection. Note that the theoretical throughput achievable for 5 ms round-trip time and 64 KB window size is 105 Mbps. By running the same measurement with up to 4 concurrent TCP connections, we obtained a maximum aggregated throughput of ≈ 300 Mbps. However, since only the depth stream of the four PTaaS App data streams uses about 85% of the required bandwidth, improving the throughput of depth stream TCP connection was sufficient in our case. Hence, we incrementally increased the window size in order to achieve better performance outcomes. Among the many tests we performed, a window size of 8 MB on both sides gave use the most significant throughput increment (i.e., 199 Mbps), and increasing the window sizes to greater sizes did not yield any better results.

Equipped with this finding, we repeated the throughput measurements in the real deployment with the new VPN solution in place and obtained similar results as reported in Fig. 9(a). For completeness and comparison purposes, we repeated the same tests without the virtualization overhead using public IPs (i.e., a non-secure solution). For this case, the results we obtained that reached up to $\approx$ 360 Mbps are reported in Fig. 9(b).These results clearly confirm that - choosing appropriate VPN encapsulation technique and using “socio-technical” instrumentation strategies for both network and application tuning are essential to provide the expected baseline throughput that satisfies application requirements. Murder mystery was finally solved after several months of frustrating and sometimes exciting moments!

V. Related Work

Within the last five years, much attention is being focused on performance evaluations of broadband access networks from the edge of the network [15]–[18]. These studies are primarily driven by regulators, governments, researchers and operators pursuing wide deployments of instrumented routers through voluntary user participation in UK, USA, Canada, Brazil, Singapore and Europe [15]. A few of these studies have also analyzed the many aspects of performance as perceived by the end users [16]–[18]. Commonly, these deployments rely on periodic active measurements to monitor and assess the view on broadband performance evolution in different geographical territories, and could enable easier troubleshooting of network issues [17], [18]. Passive measurements are less common and require explicit user consent to avoid any privacy implications of inspecting personal user traffic that may contain unencrypted sensitive information [17]. The incentives for user participation in these broadband performance monitoring efforts are to enable them to more accurately verify their expectations of network speeds from their home, and to provide performance trends via a web-portal with statistics and interactive plots of collected measurements [15], [16], [18]. Our application and network co-design strategy takes such trends of monitoring the edge network further by having individual Gigabit Apps customize and exposes custom metrics that provide insights to users/operators on performance trends and possible bottlenecks. We remark that the approach for multi-layer and multi-domain co-design of measurements with application is also being adopted by other recent frameworks such as mPlane [19].

From the experiences gained in deploying the telehealth PTaaS App in this paper, many of the previously developed telehealth efforts can be revamped to take advantage of ultra high-speed broadband networks and cloud computing. For example, the mPHASiS project [20] that uses mobile medical sensors to provide continuous monitoring in supervised rehabilitation of patients can be transformed into a Gigabit App. Another context where a Gigabit App transformation in legacy telehealth implementations can benefit from ultra high-speed broadband networks and cloud computing can be seen in the REWIRE project [21]. Therein, at-home stroke patient rehabilitation was enabled by using wearable body sensors and inexpensive motion sensors (e.g., Microsoft Kinect or Sony PlayStation Eye) for real-time data collection and remote supervision by the hospital clinicians. Further, the Bush Babies Broadband project [22] that provides on-demand virtual neonatologist intensive care for babies at rural, remote or urban area hospitals can also take advantage of our Gigabit App transformation and high-speed network tuning approach for telehealth Apps. A cloud-based service deployment for a Gigabit App transformation in this case can allow prevention of cases such as e.g., immediate need to move the baby, through transfer to real-time monitoring and fast/high-definition data transfer of ventilator data, audiovisual streams and static physiological data such as X-ray images to a consulting Neonatologist.

Many Gigabit Apps, as with most of the current Apps (e.g., Skype, Dropbox), still rely on the TCP protocol for data transfer even though such a choice has several implications on the achievable performance at high-speeds speeds over long-distance and multi-domain networks [23]–[26]. Previous experimental studies have demonstrated the impact on performance and fairness when using different TCP congestion control algorithms [23]–[25], pointing out the importance of selecting the right algorithm depending on the App use case requirements and constraints e.g., the bandwidth-delay product is of paramount importance over long-distance high-speed links [25]. At the application layer, a deep understanding of the impact that different sized packet buffers have on various types of App traffic is essential to properly tune supporting TCP-based services [26]. Other works on monitoring and troubleshooting of multi-domain networks can be found in [27]–[32]. These works do provide valuable guidance, and our work is an exemplar case study where such TCP tuning need is intermixed with other factors when troubleshooting TCP performance problems of Gigabit Apps over long-distance high-speed networks. Particularly, our work highlights the importance of multi-domain monitoring for collaborative troubleshooting (i.e., it acts as a ‘socio-technical tool’) by having multiple vantage points of the performance through instrumentation at end-sites and intermediate network hops at strategic locations with perfSONAR-like frameworks [8]. Our work thus provides a set of design guidelines for entities to share measurement resources and collaborative measurement intelligence from multiple perspectives in order to effectively identify network anomaly events and diagnose end-to-end performance bottlenecks affecting Gigabit Apps.

VI. Conclusion

This paper highlighted the challenges in the life-cycle of an exemplar smart health enabling Gigabit App viz., PhysicalTherapy-as-a-Service starting from its design in the lab, to real-world deployment to users at homes with Google Fiber. We showed through our experiences that the distributed as well as the data-intensive nature of Gigabit Apps such as our PTaaS App, along with high-performance expectations on the intermediate multi-domain network segments and virtualization technologies, necessitate novel multi-layer instrumentation approaches to tune App performance and troubleshoot intertwined bottlenecks. The Gigabit App developer has to consider a co-design approach with other stakeholders, particularly those in network/performance engineering roles in order to ensure performance visibility is achievable through joint orchestration of active and passive measurements. Addressing the challenges using the multi-layer instrumentation strategies described in this paper can enable Gigabit App developers to effectively conduct performance tuning and bottleneck troubleshooting in both laboratory and operational environments with wide-area networking components and cloud services.

In addition, our strategy of using perfSONAR extensions through our Narada Metrics framework helped in the process of methodical troubleshooting of performance bottlenecks that had originally made the lab-tested PTaaS App totally unusable in a real-world deployment. It helped us to effectively correlate the passive monitoring at the edges with active monitoring in the intermediate network hops at strategic locations to gain performance visibility from multiple vantage points, and allowed us to overcome bottlenecks at multiple layers of networking, virtualization, application and user experience.

From the insights of our troubleshooting case study, we hope ISPs are better informed to more consciously deploy perfSONAR as a "socio-technical tool" and share performance data of their networks and systems with the budding Gigabit App developer community. The multi-layer stakeholder collaboration will be needed even as networking infrastructures and technologies mature, since they will continue to be only tuned to support popular Apps (e.g., Netflix, Skype), and newer Gigabit Apps such as our PTaaS App and others will need custom tuning of parameters at multiple layers to ensure satisfactory user experiences. Thus, transforming traditional measurement and monitoring practices to enable collaborative practices can surely shorten the time to deploy Gigabit Apps to residential users to reap tangible benefits of using ultra high-speed broadband in areas of national priority such as healthcare, public safety and education.

Acknowledgments

This work was supported in part by the National Science Foundation under awards: CNS-1346789, ACI-1246001 and ACI-1245795, and by the US Department of Energy under award numbers: DE-SC0001331 and DE-SC0007531. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the united states government or any agency thereof.

References

[1] "Google Fiber", *https://fiber.google.com*, Accessed 2016.

[2] "Comcast Gigabit Pro", *http://goo.gl/BmgYBx*, Accessed 2016.

[3] "AT&T GigaPower", *https://goo.gl/dQZZCt*, Accessed 2016.

[4] "DubLINK Network Supports Economic Development, Health Care, and Supercomputing", *http://muninetworks.org*, Accessed 2016.

[5] "US Ignite Initiative", *http://www.us-ignite.org*, Accessed 2016.

[6] G. Aceto, A. Botta, W. De Donato, A. Pescapé, "Cloud Monitoring: Definitions, Issues and Future Directions", *Proc. of IEEE CloudNet*, 2012.

[7] P. Calyam, A. Mishra, R. Antequera, D. Chemodanov, A. Berryman, K. Zhu, C. Abbott, M. Skubic, "Synchronous Big Data Analytics for Personalized and Remote Physical Therapy", *Elsevier Pervasive and Mobile Computing*, 2016.

[8] A. Hanemann, J. Boote, E. Boyd, J. Durand, L. Kudarimoti, R. Kapacz, M. Swany, S. Trocha, J. Zurawski, "PerfSONAR: A Service Oriented Architecture for Multi-domain Network Monitoring", *Proc. of Service-Oriented Computing-ICSOC*, 2005.

[9] P. Calyam, L. Kumarasamy, C. G-. Lee, F. Ozguner, "Ontology-Based Semantic Priority Scheduling for Multi-domain Active Measurements", *Journal of Network and Systems Management*, Vol. 22, No. 3, pp. 331-365, 2014.

[10] A. K. Mishra, M. Skubic, C. Abbott, "Development and Preliminary Validation of an Interactive Remote Physical Therapy System", *Proc. of IEEE Engineering in Medicine and Biology Society*, 2015.

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

[12] A. Vallat, D. Schneuwly, "Clock synchronization in telecommunications via PTP (ieee 1588)", *Proc. of IEEE Frequency Control Symposium*, 2007.

[13] B. Gillian, "Vyatta: Linux IP Routers", *http://freedomhec.pbworks.com/f/linux_ip_routers.pdf*, 2007.

[14] "ESnet Fasterdata Knowledge Base", *https://fasterdata.es.net/host-tuning/ms-windows*, Accessed 2015.

[15] M. Bagnulo, T. Burbridge, S. Crawford, P. Eardley, J. Schoenwaelder, B. Trammell, "Building a Standard Measurement Platform", *IEEE Communications Magazine*, 2014.

[16] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, A. Pescapé, "Measuring home broadband performance", *Communications of the ACM*, Vol. 55, No. 11, pp. 100-109, 2012.

[17] S. Sundaresan, S. Burnett, N. Feamster, W. de Donato, "BISmark: a testbed for deploying measurements and applications in broadband access networks", *Proc. of USENIX*, 2014.

[18] W. de Donato, A. Botta, A. Pescapé, "HoBBIT: a platform for monitoring broadband performance from the user network", *Proc. of Traffic Monitoring and Analysis*, pp. 65-77, 2014.

[19] B. Trammell, P. Casas, D. Rossi, A. Bar, Z. Houidi, I. Leontiadis, T. Szemethy, M. Mellia, "mPlane: An intelligent measurement plane for the internet", *IEEE Communications Magazine*, 2014.

[20] P. Kulkarni, Y. Ozturk, "mPHASiS: Mobile patient healthcare and sensor information system", *Elsevier Journal of Network and Computer Applications*, Vol. 34, No. 1, pp. 402-417, 2011.

[21] N. A. Borghese, R. Mainetti, M. Pirovano, P. L. Lanzi, "An intelligent game engine for the at-home rehabilitation of stroke patients", *Proc. of IEEE SeGAH*, 2013.

[22] C. McGregor, B. Kneale, M. Tracy, "On-demand Virtual Neonatal Intensive Care units supporting rural, remote and urban healthcare with Bush Babies Broadband", *Elsevier Journal of Network and Computer Applications*, Vol. 30, No. 4, pp. 1309-23, 2007.

[23] Z. Yue, X. Zhang, Y. Ren, J. Li, Q. Zhong, "The performance evaluation and comparison of TCP-based high-speed transport protocols", *Proc. of IEEE ICSESS*, 2012.

[24] H. Bullot, R. Les Cottrell, R. Hughes-Jones, "Evaluation of advanced TCP stacks on fast long-distance production networks", *Journal of Grid Computing*, Vol. 1, No. 4, pp. 345-359, 2003.

[25] M. Sosic, V. Stojanovic, "Resolving poor TCP performance on high-speed long distance links - Overview and comparison of BIC, CUBIC and Hybla", *Proc. of IEEE SISY*, 2013.

[26] E. Cocker, F. Ghazzi, F. Ghazzi, U. Speidel, M.-C. Dong, V. Wong, A. Han Vinck, H. Yamamoto, H. Yokoo, H. Morita, H. Ferreira, A. Emleh, R. McFadzien, S. Palelei, R. Eimann, "Measurement of buffer requirement trends for real time traffic over TCP", *Proc. of IEEE HPSR*, 2014.

[27] M. Fagan, M. M. H. Khan, B. Wang, "Leveraging Cloud Infrastructure for Troubleshooting Edge Computing Systems", *Proc. of IEEE ICPADS*, 2012.

[28] E. Kissel, A. El-Hassany, G. Fernandes, M. Swany, D. Gunter, T. Samak, J. M. Schopf, "Scalable integrated performance analysis of multigigabit networks", *Proc. of IEEE/IFIP NOMS*, 2012.

[29] C.-W. Chang, G. Huang, B. Lin, C. Chuah, "LEISURE: Load-Balanced Network-Wide Traffic Measurement and Monitor Placement", *IEEE Trans. on Parallel and Distributed Systems*, Vol. 26, No. 4, pp. 1059-70, 2015.

[30] F. Espinet, D. Joumblatt, D. Rossi, "Zen and the Art of Network Troubleshooting: A Hands on Experimental Study", *Proc. of Traffic Monitoring and Analysis*, 2015.

[31] A. Mahimkar, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, Q. Zhao, "Towards automated performance diagnosis in a large IPTV network", *In ACM SIGCOMM CCR*, Vol. 39, No. 4, pp. 231-242, 2009.

[32] H. Cui, E. Biersack, "Troubleshooting slow webpage downloads", *Proc. of IEEE INFOCOM Workshops*, 2013.