Testing the Feasibility of a Low-Cost Network Performance Measurement Infrastructure

Scott Chevalier, Jennifer M. Schopf
International Networks
Indiana University
Bloomington, IN 47408
{schevali, jmschopf}@indiana.edu

Kenneth Miller
Telecommunications & Networking Services
The Pennsylvania State University
University Park, PA 16802
kdm193@psu.edu

Jason Zurawski
Energy Sciences Network
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
zurawski@es.net

***Abstract*—Todays science collaborations depend on reliable, high performance networks, but monitoring the end-to-end performance of a network can be costly and difficult. The most accurate approaches involve using measurement equipment in many locations, which can be both expensive and difficult to manage due to immobile or complicated assets.**

The perfSONAR [11] framework facilitates network measurement making management of the tests more reasonable. Traditional deployments have used over-provisioned servers, which can be expensive to deploy and maintain. As scientific network uses proliferate, there is a desire to instrument more facets of a network to better understand trends.

This work explores low cost alternatives to assist with network measurement. Benefits include the ability to deploy more resources quickly, and reduced capital and operating expenditures. We present candidate platforms and a testing scenario that evaluated the relative merits of four types of small form factor equipment to deliver accurate performance measurements.

I. INTRODUCTION

Networks are essential to modern research and education [28]. Distance education requires stable network performance to facilitate audio and video. Research innovation relies on bulk data movement that is free of transmission errors and plentiful bandwidth.

Almost all current research collaborations depend on stable networks that are reliable and error free in order to be successful. In fact, the average network user is generally unaware of the specifics of why a network experience may not go smoothly [30], but can detect deviation from their expectations [34].

Initiatives such as the National Science Foundation's Campus Cyberinfrastructure program [18] have brought new focus to the plight of the state of R&E networks in the United States. These programs have collectively invested $82 Million via 170 awards in 46 states and territories [36]. The goal of these programs is to upgrade and rethink network architectures via the seminal work on the Science DMZ [23], as well as to encourage network monitoring using tools such as the perfSONAR Monitoring Framework [26] to better gauge network performance, locate problems, and bring them to faster resolution.

Ensuring end-to-end performance, e.g. observed from the point of view of a network user, is complex even with intelligent tools due to the complexity of the path [32]. The environment features many layers [38] and different administrative domains, which can complicate the path and result in a reduction in the overall performance. Debugging problems of this nature is equally challenging, and requires knowledge of a myriad of components: applications, communications protocols, computational hardware, and network hardware, to name several broad areas of focus.

Visibility into performance characteristics is crucial. Keeping with the theme of Metcalfe's Law, a monitoring infrastructure becomes more useful as the deployed instances grow [33]. However, potential deployments need to keep the costs associated with long term operation low in order to be feasible for network operators. There is a need to ensure that the initial cost and long term maintenance requirements of network measurement equipment remains low, and usability of the resulting framework remains high.

The difficulties in deploying and using network monitoring software is being addressed by the perfSONAR project, which has invested considerable resources into simplifying the task of software deployment. Early incarnations required building dedicated machines with a customized operating system. Recent improvements [12] now facilitate deployment via a series of software "bundles": one of which is specifically targeted towards use on "low cost" hardware offerings. The rationale is simple: if the software is easy to deploy and maintain on inexpensive resources, the number of these resources will grow and benefit the original deployment site as well as the community at large. Responding to community feedback, the project is addressing the desire for operation on devices with a price point of around $200 [21].

However, simplifying the software is only one part of the deployment issue. For large scale deployments which will have many test instances, we must ensure that low cost resources used in this environment will offer observations that are free of self-inflicted error and are designed to be free of internal bottlenecks. Additionally, the resources must be capable of

This manuscript has been authored by an author at Lawrence Berkeley National Laboratory under Contract No. DE-AC02-05CH11231 with the U.S. Department of Energy. The U.S. Government retains, and the publisher, by accepting the article for publication, acknowledges, that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes.

continuous operation for a number of years, otherwise the investment, no matter how small it may be, will be wasted. We investigated whether single board technologies, also referred to as Small Form Factor (SFF) machines employing the Micro, Mini, Nano, or Pico ITX motherboard technology, might be capable of being used for network measurement.

This paper presents several options with an evaluation to better understand the choices of available hardware for network performance measurement activities. Starting with a selection of hardware offerings, we show a comparison of cost, performance, maintenance, and overall usability when deploying a network measurement infrastructure. We describe our comprehensive study of perfSONAR operation on these devices in several pragmatic environments. We conclude with some preliminary guidance on purchasing and maintaining a deployment of inexpensive testing resources.

The rest of the paper proceeds as follows. Section II discusses similar measurement projects, and how they relate to this experimentation. Section III talks about network measurement preliminaries, and Section IV discusses possible deployment strategies. Section V describes our experiments plan as part of the SC15 SCinet [14]. Section VI offers commentary on the observed results after the deployment was tested within the SCinet [15] infrastructure. Section VII discusses the experience and outlines future work by the perfSONAR project and similar community efforts.

II. RELATED WORK

Deployment of network testing resources is a well researched topic. In [20], the authors perform a comprehensive review of available technologies, many of which are targeted towards a broad deployment of dedicated devices. These projects have a common goal to better understand network traffic from the point of view of an end user, as well as to locate and fix architectural bottlenecks. Many of these solutions are inexpensive, meeting at least some of the criteria that drove our current work. Some of these solutions are designed to be black boxes, with little programmatic interaction or insight into the underlying reasons for why the results of a test are as good, or as poor, as reported.

Many small node solutions are geared toward the home user, and are not designed (or capable) of handling the higher speeds and requirements of the Research and Education (R&E) network infrastructure, which is our focus area. Few of these measure network throughput, which is required for our work, and instead focus on traceroutes or ICMP measurement data. These related projects include:

- *BISmark* [35] is a platform to perform measurements of Internet Service Provider (ISP) performance and traffic inside home networks. This device functions as a traditional broadband router, performing normal functions in addition to periodic network performance measurements of throughput and latency.
- *RIPE Atlas* [19], [27] is a global network of probes that measure Internet connectivity and reachability, and is primarily deployed by home users to provide an understanding of the state of the commercial Internet (not R&E networks) in real time.
- *NetBeez* [6] is a product designed for network managers primarily interested in early fault detection and quick troubleshooting of networks, primarily in a LAN, not a WAN, environment. Via broad deployment of small network monitoring agents at each office, it is possible to quickly detect and fix network and application issues at that scale.
- *CAIDA* deploys and maintains a globally distributed measurement called *Archipelago (Ark)* [25], based on the Raspberry PI platform [13]. This infrastructure is formed by distributing hardware measurement devices with geographical and topological diversity, but is not collecting throughput data due to limitations in the hardware.

Each of these approaches offers an inexpensive platform, easy integration with a proprietary central management system, and the ability to collect a variety of measurements. An unfortunate downside is the lack of ability to federate instances that span different domains of control or easily share results for visualization and analysis.

perfSONAR is designed to provide federated coverage of paths using common network tools that are accessible via a common API. It can help to establish end-to-end usage expectations. There are 1000s of perfSONAR instances deployed world wide, many of which are available for open testing of key measures of network performance. This global infrastructure helps to identify and isolate problems as they happen, making the role of supporting network users easier for engineering teams, and increasing productivity when utilizing network resources.

It is desirable to adopt a solution that integrates easily with this global framework, knowing that we can use it to address local and remote performance observations. The perfSONAR approach differs from other frameworks because:

- Each probe is individually owned and operated;
- Federation of resources, within and between domains, is available by default;
- "Open" testing and data access policies may be set by the local sites;
- The software is designed to work on commodity hardware;
- There are several broad possibilities for use: end-users, network operators, and application developers.

By adopting perfSONAR as the measurement software, it is also possible to integrate into other real-time debugging frameworks such as OnTimeDetect [22], Pythia [29], or UNIS [24].

There is ongoing work examining the use of very small nodes, such as the Raspberry Pi [13] or BeagleBone [1] with perfSONAR distributions installed on them [37]. However, we focused on links that needed to be tested at close to 1 Gigabits per second (Gbps), which is beyond the capability of this type of hardware.

III. BACKGROUND AND METHODOLOGY

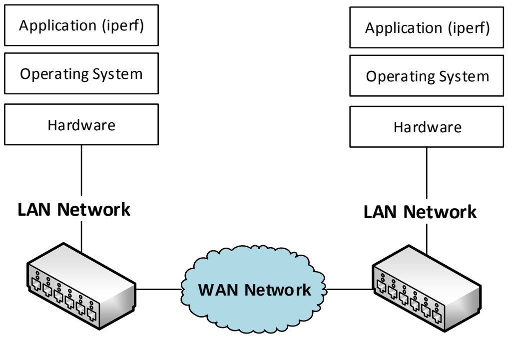


Fig. 1. Layers of Performance

To accurately measure performance of the underlying network infrastructure, it is crucial to remove imperfections caused by the measurement device. This is traditionally done by ensuring that the measurement infrastructure is performing at peak efficiency in terms of both the hardware and software. It is possible to compensate for experimental error that the test infrastructure introduces to the resulting measurement if enough information is available, although this adds significant complexity to the environment.

As part of the measurement functionality of perfSONAR, the tools can estimate an error range for latency and throughput. These calculations are based on the complete end-to-end picture, as shown in Fig 1, and are not indicative of any one one component. It is challenging to know precisely which factor on the end to end path (the end hosts, the software, the intermediate network devices, the protocols, etc.) is causing any form of error, but the error can be used as a confidence interval when evaluating the final result.

Care has been taken to optimize the perfSONAR software platform at both the operating system and application layers to ensure that these are always operating at peak efficiency. Tools will always give more accurate measurements of network behavior if they are not bottlenecked by the measurement devices themselves, either hardware or software. Since the software product is designed to run on commodity hardware, the initial hardware choices have a large impact on resulting measurement. In practice, the performance of the hardware is a function of the design and cost characteristics. We consider three classes of hardware: traditional servers, virtualized environments, and low cost hardware.

A. Server Class Hardware

In the world of computing a “server” is often distinguished as a device that is capable of providing service to multiple clients simultaneously. The hardware used is often more powerful and reliable than standard personal computers, and thus capable of more intense activities over a longer period of time. Modern servers feature an architecture that can support a single or multiple processors, with fast clock speeds, on a motherboard that can support communication with peripherals. The main memory is measured in Gigabytes, and can support the needs of the operating system and concurrent service requests. Network interfaces may range in size from 1 to 100 Gbps.

The bottlenecks of this computing architecture occur in four common places:

- **Processor Speed and Availability**: A single stream of TCP can only be bound to a single processor core. The performance you achieve with a software measurement tool will always be limited by the performance of the CPU. A system may have many other tasks to do simultaneously in a multi-threaded environment, thus is is highly beneficially to have a CPU with a high clock speed and multiple cores available for system operation.
- **Contention for system bus**: The system bus handles all communication between peripherals. If there are other devices that are using this limited resource during a measurement, the background noise can impart additional error.
- **Improper tuning of Network Interface Card (NIC)**: Modern NICs feature processors on-board that can attempt to offload the task of network communication away from the main CPU. Knowing the performance characteristics of the NIC is important, and how it will interact with the system as a whole.
- **Memory Speed and Availability**: Most network test tools are “memory” based testers, meaning they create, store, send, and receive directly from the main memory of a system. If the memory is slow in relation to the CPU, bus, or NIC, it will become a bottleneck in testing.

In most cases server class hardware is able to perform at or near “line rate”, i.e. the maximum throughput given protocol overhead, due to the nature of the components. Tuning of the operating system and components can result in moderate gains over a standard configuration.

B. Virtual Hardware

In computing, the act of “virtualization” refers to creating a virtual (rather than physical) version of computing components. This approach is not new, and has proliferated in computing since the 1960s when large shared resources (i.e. mainframe computers) were divided up to support multiple users. Modern virtualization focuses on delivering clonable environments that are identical to a physical resource, emulating a complete hardware and software stack through the dedication of a small number of physical resources.

However, when used in a network measurement environment, virtualization can strongly affect the accuracy and stability of measurements, particularly those that are sensitive to environmental considerations on a host or operating system. perfSONAR was designed to“level the playing field” when it comes to network measurements by removing host performance from the equation as much as possible. The use of virtualized environments can introduce unforeseen complications into the act of measurement.

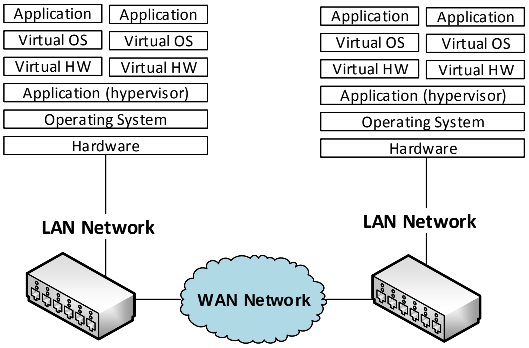


Fig. 2. Virtual Layers of Performance

As shown in Fig 2, additional layers are added to the end-to-end measurement. These additional layers impart several challenges:

- **Time Keeping**: Some virtualization environments implement clock management as a function of the hypervisor (i.e. software that manages the environment) and virtual machine communication channel, rather than using a stabilizing daemon such as NTP [31]. This could mean time skips forward, or backward, and is generally unpredictable for measurement use.
- **Data Path**: Network packets are timed from when they leave the application on one end, until arrival in the application on the other end. The additional virtual layers add latency, queuing, and potentially packet ordering or transmission errors, to the calculation.
- **Resource Management**: Virtual machines share the physical hardware with other resources, and may be removed (“swapped”) from physical hardware to allow other virtual machines to run as needed. This exercise of swapping virtual resources could impart additional error measurement values that may only be seen if a long term view of data is established and matched against virtualized environment performance.

These challenges show up in the resulting measurement and can often be difficult to fully justify, even with the availability of error estimation. For these reasons, virtual measurement platforms are not encouraged for the type of deployments we use to gain a full understanding of network performance.

C. Low Cost Hardware

Thus far two extremes have been presented: dedicated hardware capable of line rate performance that comes with a large investment in hardware and maintenance, and virtualized shared resources that are inexpensive to deploy and maintain, but do not deliver on performance goals.

A third option is the use of single-board computers. These are not a new innovation, the first of which appeared at the dawn of personal computing [17]. Small Form Factor (SFF) personal computers, e.g. those that were smaller than the Micro-ATX, became popular in the latter part of the 2000s. Utilizing commodity processors found in consumer electronics, such as cell phones, it was possible to construct small, cost effective, devices for common computing tasks such as serving media files or controlling stand-alone hardware and software tasks. Coupled with the release of Linux distributions complied specifically for these computing architectures, these devices have proliferated [1], [3], [13].

Indeed the SFF environment has grown quickly, which is both a blessing and a curse. Currently it is possible to purchase a turn-key device for less than $200 that promises 1Gbps network speeds and computing power similar to PC. These devices offer dedicated resources, a step better than virtualization, but may feature some of the same types of bottlenecks, if not orders of magnitude worse, then in their larger server-class relatives. In particular, the shared system bus, single core processor with a slower clock speed, and limited memory footprint are all reasons for concern when it comes to network measurement.

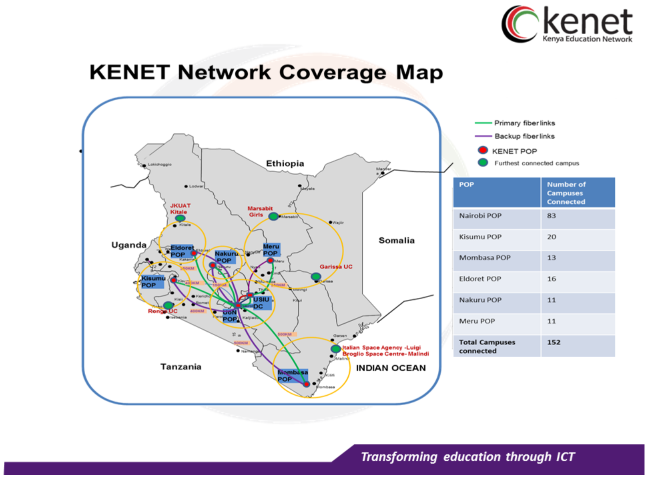
There are many available options in this space given the current size and growth pattern, so it is not feasible to examine all of them. However, in the next section we detail guidance on the broad requirements that will lead to accurate and reliable network measurement activities and discuss several options.

IV. DEPLOYMENT SCENARIOS

When planning the deployment of a measurement framework, the most important factor is to position measurement equipment along highly utilized paths. This principle holds true for deployments that span a continent, a region, or a campus. Having measurement equipment along critical junctions makes them more useful for ensuring performance or during a debugging exercise.

As an example of a continental scale network, the Energy Sciences Network (ESnet) [2] is a high-performance, unclassified network built to support scientific research. ESnet provides services to more than 40 Department of Energy (DoE) research sites, including the entire National Laboratory system, its supercomputing facilities, and its major scientific instruments. ESnet maintains measurement equipment that supports throughput and latency tests at ESnet points of presence (PoPs) as well as near the site network boundary of many DoE facilities for a total of nearly 60 locations. The ESnet measurement resources are high-end servers. Each of the 60 measurement resources are connected to the ESnet network via a top-of-rack switch. The throughput measurement equipment are also connected via 10Gbps fiber connections to the hub/site router at each PoP. The initial capital expenditure investment into measurement hardware totaled approximately $\$300,000$. That number does not include ongoing support or refresh expenses.

Not all networks span an entire continent. KENET [5], the Kenya Education Network, is Kenya's National research and education network. KENET connects facilities throughout the country to the leading internet exchanges. Working with partners International Networks at Indiana University (IN@IU) [4] and the Network Startup Resource Center (NSRC) [7], KENET researched and deployed components of an 11 node measurement infrastructure using a low-cost server solution as shown in Fig 3. The measurement equipment was designed to be a lower price point, costing around $\$10,000$, than an over-provisioned server and more stable than a SFF device.



POP	Number of Campuses Connected
Nairobi POP	83
Kisumu POP	20
Mombasa POP	13
Eldoret POP	16
Nakuru POP	11
Meru POP	11
Total Campuses connected	**152**

Fig. 3. KENET Network

Networks that span a small physical area, e.g. a single campus or those joined via metro area connectivity, can also benefit from having a smaller scale monitoring infrastructure in place. At the Pennsylvania State University [16], the Real-Time Measurement (RTM) service continuously monitors and measures the University Enterprise Network (UEN) to identify issues and enhance performance [9]. Within each of the 23 campus locations there are numerous network PoPs. Each campus is home to a dedicated full scale server used to measure parameters that characterize network performance back to the main campus.

As a part of an NSF Campus Cyberinfrastructure award, the campus is working toward core network upgrades, resilient paths, and a Science DMZ, as shown in Fig 4. The updated design features additional SFF devices at key intersections. These additional measurement resources will enable testing along any section of the path associated with the end to end performance across the multi-campus system. This deployment highlights using the right resource in the right setting: full scale servers for the heavily used connections, and smaller, cheaper nodes to pull out additional information, with more flexibility, wherever needed. The SFF nodes may also be used to provide extra information on demand as part of the network debugging process.

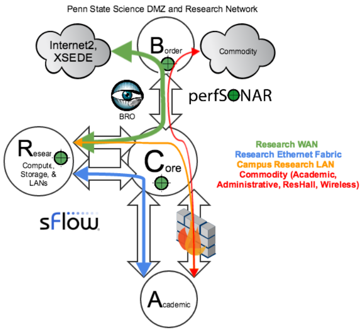


Fig. 4. Penn State Network

V. QUANTIFICATION AND EXPERIMENTATION

To better understand the feasibility of deploying SFF measurement equipment in a network measurement infrastructure, an experiment was devised to validate several varieties during operation of a large scale network that mixed both enterprise and research traffic profiles. A set of criteria was established to compare the relative merits of each platform when running perfSONAR measurement components as they operated for approximately a week under typical working conditions.

A. *Environment*

The 27th annual International Conference for High Performance Computing Networking, Storage, and Analysis (SC) was held in Austin, TX in November of 2015. SCinet is the conferences dedicated high performance research network. It is the fastest and most powerful network in the world, built by volunteer expert engineers from industry, academia, and government for the duration of SC: just over one week. SCinet network traffic peaked at more than 500Gbps, supporting high-performance research demos, wireless traffic for over 6000 simultaneous wireless clients daily, and connected meeting room connections over 89 miles of optical fiber. This environment, pictured in Fig 5, was chosen as a crucible for SFF testing due to the magnitude of the measurement challenge, along with the at-scale qualities of the network.

A total of 18 SFF resources were targeted for deployment locations within the SCinet infrastructure. These locations were chosen to cover several use cases:

- Near the demarcation of the SCinet network within the conference center;
- Within the core infrastructure, where all traffic traverses upon entry, egress, or transit; and
- Near key locations of congestion for the wireless and wired client connections.

We selected 4 types of SFF hardware to evaluate for deployment, all with a price point below \$200. Note we did not include Raspberry Pi or equivalent equipment in this experimentation, as a design criteria was that each node would be able to test close to 1Gbps of throughput. The four evaluated technologies were:

- 3 **BRIX** by *GigaByte*, model $GB-BXBT-2807$;
- 9 **LIVA** by *ECS*, model $batMINI$ [1];
- 3 **NUC** by *Intel*, model $NUC5CPYH$; and
- 3 **ZBOX** by *ZOTAC*, model $CI320nano$.

Each instance was configured to mimic a traditional perfSONAR testing resource built on top of a supported Linux platform. Of the initial 18 total machines, 6 of the devices experienced issues during transit (likely due to poorly connected components coming lose from jostling) or during operation. Due to this, 12 machines were deployed and organized into a single "mesh" where each resource tested against all other nodes to provide a full set of measurements.

B. Evaluation criteria

To fully evaluate the SFF performance, we considered several factors related to the physical qualities of the devices, their overall performance, and their ability to be easily integrated and maintained over a period of time. The following factors were evaluated for each:

- **Usability**
  - *Unit Cost*: The total cost to purchase all components (case, power supply, board, processor, memory, storage, and basic peripherals such as networking). Some devices were sold "as is", others required purchasing additional components.
  - *Operating System Support*: The operating systems that are known to work for the hardware platform, along with any abnormalities with device support.
  - *Hardware Capabilities*: The number of cores, along with the clock speed, of the Central Processing Unit. The amount of memory available. The capacity of the NIC.
  - *Power Delivery*: The mechanism for power delivery: external brick or enclosed in the device.
  - *Ease of Installation*: A subjective evaluation on the process to assemble (if required), install, and configure each tester.
  - *Ease of Operation*: A subjectively evaluation of the process to use and maintain each tester.
- **Performance**
  - *Observed Throughput*: Observed average throughput versus the maximum interface capacity.
  - *NTP Synchronization*: Ability to measure time accurately and precisely.
  - *Device Stability*: Does not introduce jitter or other systematic error into measurements.

These factors are not a panacea, but provide a useful metric for the feasibility of SFF as measurement infrastructure on deployments large and small.

C. Usability results

Table I discusses the results of the usability survey. Factors are rated between 1 and 3 stars, with 3 being the best. Several trends emerged during testing:

- The **LIVA** devices experienced the greatest number of issues during testing. They were only able to support a single operating system, which limited functionality. Driver support for peripherals, including an issue observed with SELINUX, remained a problem during testing. Additionally, it it required a larger external power source (delivered via a brick, although 5V USB is a possibility) and that a keyboard and monitor be present at bootup, e.g. "headless" operation was not possible as in other testers.
- The **BRIX** and **NUC** both required the use of a larger external power source (that caused plug blocking), but were stable and straightforward to assemble.
- The **ZBOX** had a standard grounded power cable and required no assembly.

D. Performance results

Table II discusses the results of the performance survey. Factors are rated between 1 and 3 stars, with 3 being the best. Data analysis revealed several trends. Overall these devices, when operating and reporting data, showed fairly similar results and achieved acceptable throughput, in aggregate. One BRIX and one NUC device reported lower average throughput. Investigation at the time of collection could not find a fault in configuration of the device or network, thus the reason why these two individual machines would report lower performance remains unsolved.

As previously noted, there were many instances of mechanical and software failure for the tested devices. These included:

- Several **LIVA** devices were damaged during transit (in a padded bag for shipment) to the event and could not be

---

[1] This model is no longer in production, and has been replaced by newer X and $X2$ models

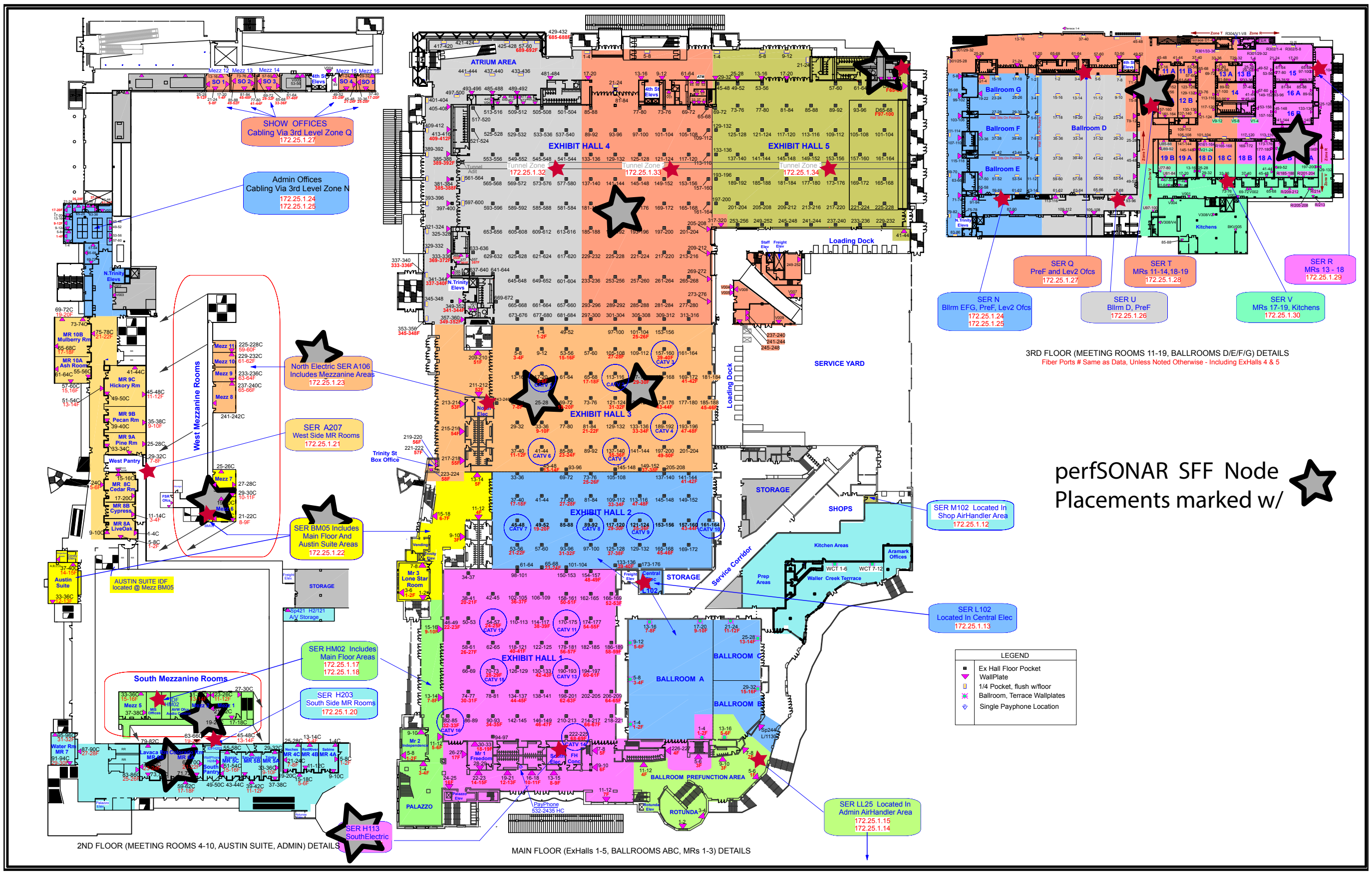


Fig. 5. SC15 Showfloor

installed. Others were configured, but ceased to respond after a number of days. Investigation found that they had not booted properly due to a lack of a keyboard and monitor (e.g. “headless” operation was not functional).

- A **BRIX** device was configured before the event, but would not respond to credentials upon deployment. Memory corruption during transit was assumed.
- 2 different **BRIX** devices reported significant TCP packet loss when testing to one another, and little to no TCP packet loss to all other nodes. Investigation at the time could not determine if this was related to the path or the configuration.
- 2 **ZBOX** devices were never capable of testing to each other, but tested to all other devices without issue. Investigation was inconclusive regarding a cause, although configuration could not be ruled out as a possibility.

VI. UNDERSTANDING THE RESULTS

After evaluation of the results in Section V, we detail the following several key findings regarding this hardware. Universal findings are:

- Each of the small nodes may be inherently more fragile and cheaply constructed than a server. The price is an attractive option that may outweigh the durability.
- Hardware and operating system interaction is still challenging, due to rapid changes in design and support.
- Prices fluctuate widely for the SFF devices the authors tested in the weeks before and after purchase.

Model availability continues to expand, and there will always be newer, smaller, faster devices available on the market. This testing hopes to assist in selecting models or even help to determine if these SFF nodes are the right choice for your organization.

Model	*Cost*	*OS*	*Number of Cores*	*Power*	*Install*	*Operation*
BRIX	$99 Base; < $200 as tested	CentOS & Debian	Dual-core	Power brick or 5V USB	Requires tools	As expected
LIVA	$160	Debian only; Driver Issues	Dual-core	Power brick	Snap together	Boot issues; Hardware failures
NUC	$130 Base; < $200 as tested	CentOS & Debian	Dual-core	Power brick	Requires tools	As expected
ZBOX	$125 Base; < $200 as tested	CentOS & Debian	Quad-core	Grounded cord	Tool-free assembly	As expected

TABLE I
USABILITY RESULTS

Model	*Throughput*	*NTP*	*Stability*
BRIX	$> 900 Mbps$ average, one unit reported $725 Mbps$ average	No NTP Sync issues found	Two units reported significant TCP packet loss
LIVA	$> 800 Mbps$ average, found to be processor limited	No NTP Sync issues found	Several units inexplicably halted during operation or showed damage due to poor manufacturing
NUC	$> 900 Mbps$ average, one unit reported $845 Mbps$ average	No NTP Sync issues found	No errors reported
ZBOX	$> 900 Mbps$ average	No NTP Sync issues found	No errors reported

TABLE II
PERFORMANCE RESULTS

A. BRIX

The **BRIX** by *Gigabyte* was found to be a solid performer in the tests. The device was found to support both CentOS and Debian operating systems. There are different processor options available, both are Intel Celeron Processors, either a 1.58GHz or 2.16GHz dual core. As tested the unit retails for $99, and required additional components bringing the total to around $200. This was found to be a good middle of the road tester, despite some of the observations for performance of certain units.

B. LIVA

The **LIVA** by *ECS* was the smallest, most inexpensive (at around $160), and least feature rich device that was tested. The hardware was found to only support the Debian operating system, and in particular only supported one variant that featured a specific driver that supported the flash memory. The build quality was questioned, given the number of units that did not boot upon arrival, or failed during operation. The inability to operate without a keyboard and monitor was considered a major flaw in operation. The under powered processor limited usefulness as a throughput tester. The device was found to have a low power draw at 15W, which makes it a candidate for for Power over Ethernet (PoE). It is recommended for deployments that require an extremely low price point and need only limited test capability.

C. NUC

The **NUC** by *Intel* was also found to be a solid performer in the tests, and a good value for the money. Like the **BRIX** described in Section VI-A, the NUC supported both CentOS and Debian operating systems and retails for $130, but with additional components the cost is around $200. There are different processor options available, both are Intel Celeron Processors, either a 1.6GHz or 2.16GHz dual core. The positive aspects of this tester were that it featured a well known manufacturer and did not experience any unexplained mechanical failures.

D. ZBOX

The **ZBOX** by *Zotac* performed best across the board and featured the most conveniences, including several USB ports and no assembly required. There are different processor options available, both are Intel Celeron Processors, either a 1.8GHz or 2.16GHz quad core. The processor has the highest stability and performance of measurements for all testers. The device supported both CentOS and Debian operating systems and retails for a base price of $125. Including required additional components raises the price to just under $200.

VII. CONCLUSION & FUTURE WORK

With the growing dependence on high performance networks to support collaborative research, there is a need for extended networking monitoring in order to ensure good performance. Historically, this has been challenging for both software deployment and hardware cost issues. The perfSONAR suite of tools offers a solution to the former problem, and in this paper we address possible approaches to the second.

Our experience with Small Form Factor (SFF) technology emphasizes the need to consider many factors when selecting a test environment, including but not limited to cost, deployability, management, and measurement accuracy. Different settings will require an emphasis on different aspects, but all of them impact the end goal of usefulness in debugging problems and ensuring performance.

In the pragmatic setting of SCinet, we evaluated SFFs including BRIX, LIVA, NUC, and ZBOX, although acknowledge that the offerings in this space are numerous. We found that although slightly cheaper than the others in price, the need for the LIVA to have a keyboard and monitor at boot time was very limiting when it came time for the actual deployments, and it only supported a single variant of Debian. LIVA machines also had numerous failures during the week. The BRIX were more reliable, but also had two nodes with significant stability issues over the week. The NUC had some operational issues and also variable results when testing throughput. Overall the ZBOX, with its stability and quad core design (as well as the tool-free assembly) seemed to be best suited for our environment.

Going forward, there are many additional hardware offerings currently available, and prices change even more rapidly. This evaluation is only the first of many, and other community efforts [8], [10] continue to test state of the art technology offerings to answer questions about network performance.

VIII. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under grant no. 0962973.

The authors are grateful for the assistance provided by the perfSONAR collaboration. Their support and suggestions greatly enhanced this experimentation.

IX. DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

REFERENCES

[1] BeagleBoard. http://beagleboard.org/.

[2] ESnet - The Energy Sciences Network. http://www.es.net/about/.

[3] Intel NUC. http://www.intel.com/content/www/us/en/nuc/overview.html.

[4] International Networks at IU. http://internationalnetworking.iu.edu/.

[5] KENET - Kenya Education Network. https://www.kenet.or.ke/.

[6] NetBeez. https://netbeez.net/.

[7] Network Startup Resource Center. https://www.nsrc.org/.

[8] NTAC Performance Working Group. http://www.internet2.edu/communities-groups/advanced-networking-groups/performance-working-group/.

[9] Pennsylvania State University's Science DMZ Research Network. http://rn.psu.edu/design/.

[10] PERFCLUB - A perfSONAR User Group. http://perfclub.org/.

[11] perfSONAR. http://www.perfsonar.net/.

[12] perfSONAR 3.5 Release. http://www.perfsonar.net/release-notes/version-3-5-1/.

[13] Raspberry PI. https://www.raspberrypi.org/.

[14] SC: The International Conference for High Performance Computing, Networking, Storage, and Analysis.

[15] SCinet: The Fastest Network Connecting the Fastest Computers. http://sc15.supercomputing.org/scinet/.

[16] The Pennsylvania State University. http://www.psu.edu/.

[17] Build a Dyna-Micro 8080 Computer. *Radio-Electronics*, pages 33–36, May 1976.

[18] Campus Cyberinfrastructure - Data, Networking, and Innovation Program (CC*DNI). https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=504748, 2015.

[19] Vaibhav Bajpai, Steffie Jacob Eravuchira, and Jürgen Schönwälder. Lessons Learned From Using the RIPE Atlas Platform for Measurement Research. *SIGCOMM Comput. Commun. Rev.*, 45(3):35–42, July 2015.

[20] Vaibhav Bajpai and Jürgen Schönwälder. A Survey on Internet Performance Measurement Platforms and Related Standardization Efforts. *IEEE Communications Surveys and Tutorials*, 17(3):1313–1341, 2015.

[21] E. Boyd, L. Fowler, and B. Tierney. perfSONAR: The Road to 100k Nodes. 2015 Internet2 Global Summit, perfSONAR: Meeting the Community's Needs, 2015.

[22] Prasad Calyam, Jialu Pu, Weiping Mandrawa, and Ashok Krishnamurthy. OnTimeDetect: Dynamic Network Anomaly Notification in perfSONAR Deployments. In *MASCOTS 2010, 18th Annual IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Miami, Florida, USA, August 17-19, 2010*, pages 328–337, 2010.

[23] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski. The Science DMZ: A Network Design Pattern for Data-Intensive Science. In *IEEE/ACM Annual SuperComputing Conference (SC13)*, Denver CO, USA, 2013.

[24] Ahmed El-Hassany, Ezra Kissel, Dan Gunter, and D. Martin Swany. Design and Implementation of a Unified Network Information Service. In *IEEE SCC*, pages 224–231. IEEE Computer Society, 2013.

[25] M. Fomenkov and K. Claffy. Internet Measurement Data Management Challenges. In *Workshop on Research Data Lifecycle Management*, Princeton, NJ, Jul 2011.

[26] A. Hanemann, J. Boote, E. Boyd, J. Durand, L. Kudarimoti, R. Lapacz, M. Swany, S. Trocha, and J. Zurawski. PerfSONAR: A Service-Oriented Architecture for Multi-Domain Network Monitoring. In *International Conference on Service Oriented Computing (ICSOC 2005)*, Amsterdam, The Netherlands, 2005.

[27] Thomas Holterbach, Cristel Pelsser, Randy Bush, and Laurent Vanbever. Quantifying Interference Between Measurements on the RIPE Atlas Platform. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, IMC '15, pages 437–443, New York, NY, USA, 2015. ACM.

[28] W. Johnston, E. Chaniotakis, E. Dart, C. Guok, J. Metzger, and B. Tierney. The Evolution of Research and Education Networks and their Essential Role in Modern Science. Technical Report LBNL-2885E, Lawrence Berkeley National Laboratory, 2010.

[29] Partha Kanuparthy, Danny H. Lee, Warren Matthews, Constantine Dovrolis, and Sajjad Zarifzadeh. Pythia: Detection, Localization, and Diagnosis of Performance Problems. *IEEE Communications Magazine*, 51(11):55–62, 2013.

[30] Matt Mathis, John Heffner, and Raghu Reddy. Web100: Extended TCP Instrumentation for Research, Education and Diagnosis. *SIGCOMM Comput. Commun. Rev.*, 33(3):69–79, July 2003.

[31] D. L. Mills. Internet Time Synchronization: The Network Time Protocol, 1989.

[32] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end Arguments in System Design. *ACM Trans. Comput. Syst.*, 2(4):277–288, November 1984.

[33] Carl Shapiro and Hal R. Varian. *Information Rules: A Strategic Guide to the Network Economy*. Harvard Business School Press, Boston, MA, USA, 2000.

[34] Leigh Shevchik. The Bandwidth Dilemma: Exceeding Low Expectations. https://blog.newrelic.com/2012/01/30/the-bandwidth-dilemma-exceeding-low-expectations/, 2012.

[35] Srikanth Sundaresan, Sam Burnett, Nick Feamster, and Walter De Donato. BISmark: A Testbed for Deploying Measurements and Applications in Broadband Access Networks. *USENIX Annual Technical Conference. USENIX*, June 2014.

[36] Kevin Thompson. CC*DNI and a Few Other Updates. http://www.thequilt.net/wp-content/uploads/KThompson_Quilt_Feb2016.pdf, 2016.

[37] Alan Whinery. UH SWARM: Dense perfSONAR Deployment With Small, Inexpensive Devices. 2015 Internet2 Global Summit, perfSONAR: Meeting the Community's Needs, 2015.

[38] H. Zimmermann. Innovations in internetworking. chapter OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection, pages 2–9. Artech House, Inc., Norwood, MA, USA, 1988.