WAN-in-Lab: Motivation, Deployment and Experiments

George S. Lee, Lachlan L. H. Andrew, Ao Tang and Steven H. Low

Abstract— **WAN-in-Lab is a hardware testbed for the design, development, testing and evaluation of high speed network protocols. It uses real carrier-class networking hardware to avoid the artifacts introduced by network simulation and emulation, while being localized to allow detailed measurement of network performance. WAN-in-Lab is an open resource, available for use by the networking community. This paper describes the structure of WAN-in-Lab and the rationale behind it, issues encountered, and experimental results that illustrate its applications.**

I. INTRODUCTION

It has recently become apparent that very high speed wide area networks (WANs) are pushing the limits of existing protocols, such as the transmission control protocol (TCP). This has led to many proposed replacements for TCP [1], [2], [3], [4], [5], [6], [7], [8], which must be evaluated, optimized and eventually developed into the next generation of TCP. Although initial design and testing can be performed using mathematical modeling and software simulation, there is ultimately a need to implement the selected algorithms in real networks. WAN-in-Lab is a test facility consisting of a complete WAN in a laboratory environment, which is to be available for use by the global networking research community.

Developing protocols requires many forms of testing. Existing tools are listed below, in decreasing order of abstraction.

1) Mathematical modeling can explore infinite classes of networks but requires simplification of actual protocols, ignoring important implementation issues.
2) Simulation is a very flexible first step in evaluating proposals. However, simulating high speed networks is many times slower than real-time and does not allow testing and optimization of actual implementations.
3) Emulation [9] of large delays using Dummynet or Netem also works well for links below 1 Gbit/s. Unfortunately, software emulation introduces artifacts, notably in the form of burstiness, which become increasingly severe at higher link rates.
4) Production networks [10], [11] are the ultimate testing ground for new protocols, allowing black-box evaluations through limited end-to-end measurements. This is suitable for tests which will not disrupt other traffic, but not for testing many failure modes, such as heavily overloaded networks, or equipment failures.

All of these play a role in developing congestion control protocols. However, as shown in Figure 1, there is a big
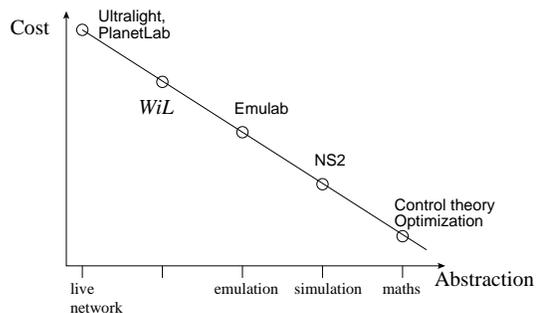
The authors are at Caltech, Pasadena, USA

Fig. 1. Tradeoff between cost and level of approximation.

gap between emulation and production networks. It is this gap which WAN-in-Lab seeks to fill. WAN-in-Lab is like a "wind tunnel" for networking, allowing real experiments in a controlled environment.

II. WAN-IN-LAB INFRASTRUCTURE

Due to the hardware complexity of building a large testbed, significant planning was required throughout the construction and configuration of WAN-in-Lab (WiL). Our goal was to build a multi-gigabit, 200 ms+ round trip time, and dynamically reconfigurable WAN using real metropolitan grade routers and fiber. This section describes the current status of the WiL infrastructure.

A. Layers 1 and 2: Physical Network

WiL's core network consists of 13 Cisco Optical Network Systems (ONS) 15454 Multiservice Transport Platform (MSTP) interconnected by 24 spools of 100 km Corning LEAF single mode fiber. The ONS 15454s are designed for metro and regional DWDM networks up to 40 Gbps. WiL has ONSs populated with OC192 transponders, optical add/drop multiplexers and amplifers, and multi-channel OC48 muxponders.

To maximize the delay obtained from the fiber, signals make multiple passes over it, as shown in figure 2. The ONSs use Wavelength Division Multiplexing (WDM) to achieve 40 Gbps over the fiber. Each ONS configured as an endpoint uses an Optical Add Drop Multiplexer (OADM) to merge or separate four 10 Gbps OC192 streams from the fiber. Each one of these 10 Gbps streams is then divided up into four separate 2.5 Gbps OC48 streams using a Muxponder card.

The 40 Gbps multiplexed signal leaves the endpoint ONS, traverses 100 km of fiber, and is received by another ONS configured as an optical amplifier. The regenerated 40 Gbps signal is then sent over another 100 km of fiber to be received
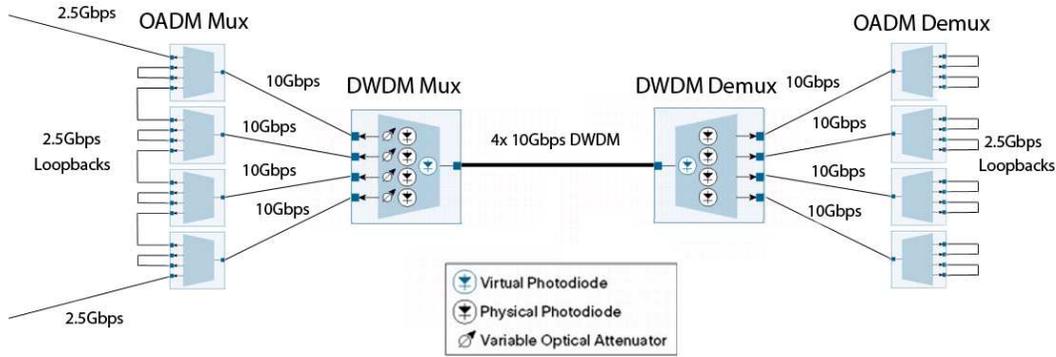
Fig. 2.   Multiplexer configured to maximize round trip time delay by looping OC48 streams over a fiber 16 times

by another endpoint ONS. This three ONS configuration uses a total of 4 spools spanning of 200 km and up to 16 separate OC48 channel capacity.

To increase delay the 2.5 Gbps channels can be connected as a loopback such that a single 2.5 Gbps stream must traverse 16 times before it leaves the system. This configuration yields a total distance of 3200 km, giving approximately 30 ms of round trip delay. Currently, WiL has three of these looped ONS configurations yielding a total of almost 100 ms.

B. Layers 3–7: Routers, servers and management

WiL has four Cisco 7609 series routers designed for high end Metropolitan Ethernet and ISPs. The routers are populated with OC48 blades (OSM-1OC48-POS-SS+ and OSM-1OC48-POS-SL) that support most enhanced WAN features such as QoS, VLANs, and VPNs. The OC48 transponders are connected into the ONS depending on the network topology. Each router also has multiple optical 1 Gbps Ethernet connections for servers and traffic generators.

WiL has over 20 Dual Intel Xeon 1U servers that generate traffic over the network. The servers are configured for PXE network boot so that they can be reconfigured quickly with different kernels and network settings. Each server has at least two network interfaces: one for experiments and one for management. Furthermore, each server is connected to a console server for alternate direct connection.

Each network device in WiL has an Ethernet management port that resides on the private network 192.168.0.0/16. In order to house all of this equipment, WiL hardware has been mounted on ten 8 foot 51U 19inch 4 post racks linked side by side. Each rack has been assigned a number that corresponds to the IP addresses allowed. For example, the left most rack is rack 1 and has 192.168.1.0/24 allocated to it. Several management servers have been installed on this network for DHCP, DNS, remote access, file storage, etc.

C. Machine room design

Physically constructing WiL was a significant challenge, due to the amount of equipment used. A lab of this size requires at least 800 sq. ft. of space that is nearby and has enough
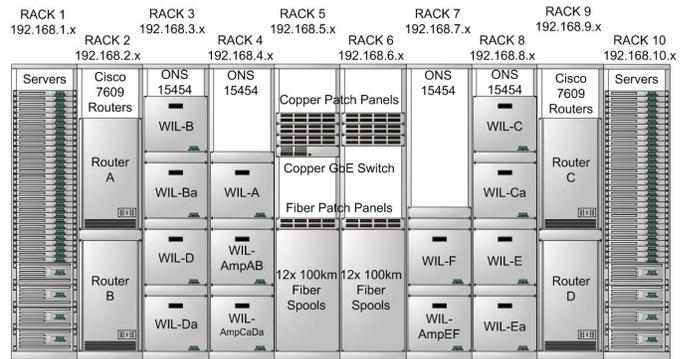

Fig. 3.   Layout of equipment over 10 racks

electrical power, chilled water for AC and noise isolation, and is secure.

Due to the wiring complexity of WiL, the physical layout is critical and has been chosen to visually represent the hierarchy of network, as in Figure 3. On the outer racks, 1 and 10, are servers and traffic generators. On racks 2 and 9 reside the routers that the servers will connect into. Racks 3, 4, 7, and 8, house all of the ONS chassis that the routers connect into. Finally, the middle two racks 5 and 6, contain all of the fiber spools and central patch panels.

On every rack, both 120/240V AC and 48V DC electricity are available via a ceiling drop. Copper and fiber network connectivity is provided on the first 3U of each rack that route to the center patch panel. Cables are individually numbered and color coded to each rack to make managing several hundred cables possible.

D. Automatic network topology reconfigurability

In January 2006, a Calient DiamondWave PXC 144x144 optical switch was added to WiL. All OC48 ONS spans in groups of 200 km, 400 km, 800 km, and 1600 km along with all OC48 interfaces on the routers have been attached to the optical switch. The optical switch will enable WiL to automatically change any OC48 fiber path resulting in minor or major changes network topology in a just a couple of seconds.

Rerouting is particularly important for delay-based protocols such as FAST. More vitally, this capability also allows the delay between routers to be adjusted from 0 ms to 100 ms (round trip), in increments of 2 ms (200 km). Using IP loopbacks, it is possible to achieve delays up to 180 ms, exceeding trans-Atlantic distances. TCP protocols behave very differently depending on the round trip delays, and so it is very important to be able to test and evaluate them in this wide range of conditions.

Automatic network reconfiguration for 1 Gbit/s copper and multimode fiber based servers are provided by Cisco 3950 switches using VLANs.

When a network topology change occurs, WiL must also reconfigure IP addresses and subnets. WiL uses a regular IP assignment scheme to help manage the many interfaces. The WiL experiment network is a collection of /24 or /28 subnets in the private 10.0.0.0/8 network.

The routers are numbered 1 to 4. The interface on router x connecting to router y will be 10.0.xy.x if $x < y$, or 10.0.yx.x if $y < x$. For example, the interface on router 1 between router 1 and 2 would be 10.0.12.1, connecting to 10.0.12.2 on router 2. Setting x to be the lower numbered router results in consistent class C subnets for each link.

Servers form tree subnets connected to the routers. All interfaces in the subtree homed on router w are numbered 10.w.xy.z/28. Digits x and y respectively specify the physical slot and port numbers of the interface on the router card. The home interface on the router has $z = 1$, and the server interface connected directly to it is numbered $z = 2$. Subsequent interfaces on that server have z values $z = 16 + 1, 32 + 1, \ldots$, corresponding to the first address in subsequent /28 subnets. For example, if server 10 was connected to Ethernet port 4 on slot 2 of router 1, and also to server 3, then the router interface would be 10.1.24.1, the attached interface of server 10 would be 10.1.24.2, the second interface on server 10 would be 10.1.24.17, and the interface on server 3 would be 10.1.24.18. This hierarchical naming simplifies the routing tables.

E. Scripted Configuration

In order to fully automate the reconfiguration of WiL's network topology, we have created scripts to execute those changes in sequence. WIL-NS is the main management server which oversees the network topology state. In addition it communicates to the optical switch, copper switch, routers, and servers, via TL1 and other scripting languages to initiate a change.

The management of WiL is based on "projects", corresponding to different users. Each project may have multiple "toplogies", corresponding to configurations of the hardware. Because WiL is concerned with performance measurements, the entire infrastructure is dedicated to one project at a time, in contrast to the time-slicing of PlanetLab [11].

Creating a topology requires describing how to configure the routers (such as interface names and routing tables), and also how to reset them to a known state afterwards. The command to change topologies then runs the "reset" scripts for the current topology, runs the configuration scripts for the new

topology, and reboots the servers into the appropriate kernels if necessary. It also calculates which spans of fiber to use to implement the delays for this topology, and configures the Calient switch accordingly.

For each topology, each server has a separate copy of the `/etc/sysconfig` directory, which configures such things as IP addresses, routing and firewall rules. The appropriate version is automatically copied into place during the boot process.

To simplify the depolyment of new kernels, all servers network-boot from WIL-NS, and their root and home filesystems are NFS mounted over the management network. Each server has additional local storage for real-time logging, or for testing disk-to-disk data transfer.

The management system is continuing to evolve, with increasing numbers of tasks being automated.

III. DIFFICULTIES ENCOUNTERED

Many researchers need to build testbeds. This section describes the pitfalls faced by WAN-in-Lab, to help other researchers who face similar challenges.

The initial motivation for WAN-in-Lab was to provide a more realistic network than a DummyNet, but one in which we understood and could control each component. Using commercial routers with no access to the firmware and physical fibers for delay inevitably reduces the level of control over the network, but the following approaches allow meaningful experiments to be carried out.

1) Multiple buffers: It is common to model a link as having a single buffer. However, 7609 routers have multiple stages of buffering, which are not clearly documented, and not configurable; in particular, the command to configure the output buffer size executes without error, but does not actually change the size of the buffer used.

Work around: The POS-SS+ and POS-SL blades both support token-bucket traffic shaping, with variable-sized buffers. By shaping the traffic to below 2.5Gbit/s per link, the buffer size can be controlled.

2) Discrete buffer sizes: Buffer sizes cannot be set in increments of one packet. The size in packets must be either a power of two from 128 to 32k, or 18, 25 or 42.

Work around: None; this is a typical artifact of real routers, and we are interested in the behavior of protocols with such restrictions.

3) Delay granularity: WAN-in-Labs delays are produced by passing data multiple times over spools of fiber. The granularity is at best the delay due to a single fiber. Moreover, there are 48 individual segments to be switched, requiring 96 optical switch ports, in addition to those used by the routers.

Work around: Delay segments are hard-wired into groups of 1, 2, 4 or 8 segments, which can be switched as single units with a limited number of optical switch ports.

4) Topologies and limited delay: The hardware supports a maximum delay under 100 ms, and only four OC48 links. (Although this is many more than the single link often used, there are interesting effects which only become apparent with five or more links [12].)
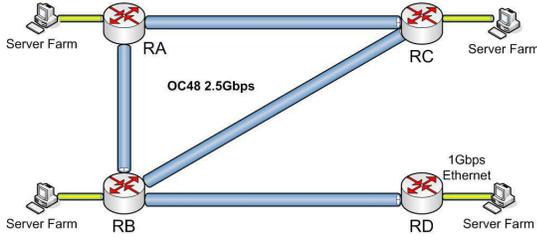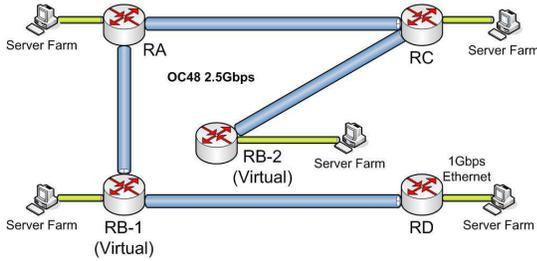
Fig. 4. WiL's physical topology.

Fig. 5. Four-hop "virtual" topology.

Work around: By setting static routes in the routers, virtual IP-level topologies can be constructed which traverse the physical links multiple times (in addition to the 16 loopbacks at layer 2), giving delays well over 100ms, and allowing topologies of at least six hops.

5) Combinations of features: The IOS configuration language allows multiple traffic shapers on a single port by assigning flows to different access lists. This would allow independent queues on each virtual link using that interface. Although the SS+ and SL blades have support for both access-lists and traffic shaping, they do not support this combination. This is a lesson for anyone constructing a network: always check explicitly that your chosen combination of features will work, rather than just ensuring that each is supported.

Work around: Do not virtualize bottleneck links. Virtual links can still be used to give increased delay, provided there is no traffic shaping on them.

IV. TOPOLOGIES

WAN-in-Lab can implement a wide range of topologies using both its hardware and software routers. Considering only the hardware routers, it has four core nodes, with degrees 1, 2, 2 and 3. This yields the physical topology shown in Figure 4, which contains sub-topologies of a three hop-parking-lot topology, a three-hop ring or a three-hop star.

The five-node, four-hop parking-lot topology of Figure 5 can be obtained by "virtualizing" router RB using static routing tables. In this mode, the router has a separate routing table for each input port. Traffic from servers connected to RB-2 is all routed via RC, even though RB is physically connected directly to RA and RD. Similarly, all traffic from any node to one of the IP addresses of routers on RB-2 is routed via RC. In the interface numbering scheme, described in Section II-D, virtual routers RB-1 and RB-2 would be numbered $w = 21$ and $w = 22$ respectively. This topology uses each physical link only once, and so does not introduce any traffic artifacts.

At the expense of slight artifacts, links can also be virtualized, yielding up to a six-node parking-lot. The artifact arises because each physical link is used as the forward path for one virtual link and the reverse path for another virtual link. Thus, the queueing (delay and loss) experienced by packets on one link is also experienced by acknowledgements on a different link. This artifact is harmless for testing the functionality of protocol implementation, but must be avoided for performance testing. As noted above, delays can also be extended by virtualising links; this does not introduce significant artifacts as long as the virtual links are not bottlenecks.

V. EXAMPLE APPLICATIONS

The first research performed on WiL was the development of a Linux implementation of the MaxNet protocol. That is described in [13], and briefly recapitulated here. This is followed by a description of more recent research, investigating the interation of protocols responding to different measures of congestion.

MaxNet is a TCP using explicit congestion signaling, in which sources respond to the maximum congestion level of any router along their path, thereby giving max-min fairness. It was designed to be provably stable for arbitrary network topologies, and incur minimum signaling overhead; unlike competing protocols, it explicitly signals only one value, the congestion level.

Since MaxNet uses assistance from routers, two of WiL's Linux servers were configured as software routers, and the OC48 links were used primarily to provide delay at high bandwidths.

A. Bistability of heterogeneous protocols

Unlike most laboratory testbeds, WiL has multiple bottleneck links. Several interesting phenomena only occur in multi-hop networks. One such is the possibility of multiple equilibrium rate allocations between flows when delay-based and loss-based protocols are mixed. This has been observed in NS2 simulations [14] and DummyNet networks [15], and this paper presents results demonstrating it using WAN-in-Lab's hardware routers.

1) Experiment: Multiple equilibria arise when the same traffic load can cause different combinations of links to be bottlenecks. This cannot happen [16] when all flows respond to a common congestion signal (e.g. loss), even if their responses differ, like H-TCP, BIC and Reno. However, it can occasionally happen if some protocols respond to loss and others to delay.

Consider the symmetric three link network show in Figure 6. The links use Random Early Discard (RED), with the parameters on link 2 set for low loss and high delay, but those on links 1 and 3 set for high loss and low delay.

In this topology, it is possible for link 2 to be the sole bottleneck since it carries the largest number of flows, or for links 1 and 3 to be bottlenecks since they have the smallest capacity. With mixed feedback signals, parameters can be set such that both are stable equilibria. The 3-hop Reno flows can create sufficient delay in links 1 and 3 that they remain
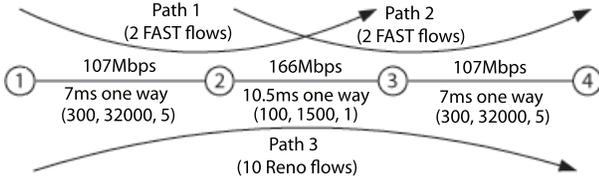
Fig. 6.   Network with links with different loss/delay tradeoffs

bottlenecks. Conversely, the 2-hop FAST flows can cause so much queueing at link 2 that Reno flows will see heavy loss and link 2 will remain bottlenecked.

Note that these two equilibria correspond to different sets of bottleneck links. This is more than just "sustained unfairness" due to old flows not making room for new flows.

The aim here is not to prove that these are distinct equilibria, since this was established in [15], or to compare FAST with Reno. The aim is to show that WAN-in-Lab provides a new proving-ground for protocol research, which can either verify or contradict results from DummyNet experiments. More specifically, this experiment aims to demonstrate that having multiple equilibria is not simply an artifact of, say, the way that RED is implemented in DummyNet, but can also be exhibited in a real network.

2) Settings and Results: Figures 7 and 8 show traces of the aggregate rate (calculated as window size/RTT) on each of the routes. In Figure 7, links 1 and 3 are bottlenecks for the entire experiment, while in 8, link 2 remains a bottleneck. However, note that fluctuations such as synchronised loss can cause the system to flip between equilibria.

As shown in Figure 6, two FAST flows traverse links 1 and 2, another two traverse links 2 and 3, and 10 Reno flows traverse links 1, 2 and 3. Links 1 and 3 had a round-trip delay of 14 ms, while that of link 2 was 21 ms. All FAST flows had tried to keep $\alpha = 10^5$ bytes in flight.

The RED parameters used are shown in Figure 6: A triple $(t_{\min}, t_{\max}, d)$ indicates that packets are dropped with probability

$$\frac{1}{d} \frac{q - t_{\min}}{t_{\max} - t_{\min}},$$

where q is an exponentially averaged queue size. The averaging parameter in each case was $w_q = 2^{-16}$, the longest averaging supported by the routers.

The link rates were 107.325 Mbit/s for links 1 and 3, and 165.866 Mbps for link 2; these were the closest approximations allowed by the traffic shaping to the 111 Mbit/s and 166 Mbit/s used in [14].

The brief dip in rate at about 5 s in Figure 8 is where a burst of cross traffic was introduced, which allowed the FAST sources to observe the propagation delay without queueing

3) Lessons: The parameters used in this experiment, notably the RED loss rates, were significantly different from those used in [14]. Initially, the parameters of [14] were used, but did not yield distinct equilibria. This demonstrates that WAN-in-Lab does indeed provide different (and hopefully more realistic) results from those of emulated experiments.

During the experiments, it was found that Reno's window seemed to reduce by more than half on many occasions, and
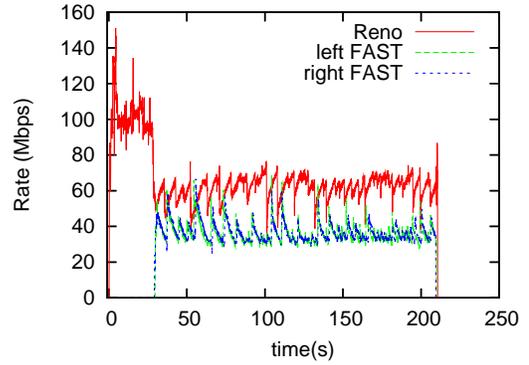

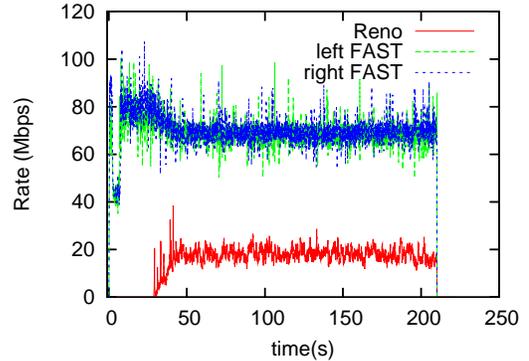Fig. 7.   Links 1 and 3 bottlenecked. (Reno starting first.)

Fig. 8.   Link 2 bottlenecked. (FAST starting first.)

that for some parameter settings (notably with small α), up to 15% of all losses resulted in timeouts. The reason for this is still under investigation, but may be related to the significant packet reordering which occurs when traffic shaping is enabled on the routers.

A consequence of this was that loss synchronisation became important. If links 1 and 3 are bottlenecked, but the Reno flows simultaneously reduce their windows, then FAST will rapidly respond, and the equilibrium will revert to link 2 being the bottleneck. To enable links 1 and 3 to be stably bottlenecked, most drops had to be random drops, requiring a larger $1/d$ than used in [14] (1/5 rather than 1/10 000). This may also be a consequence of using a smaller round trip time than used in [14], since smaller bandwidth-delay products required higher drop rates.

In the process of performing this experiment, several other limitations of "real" routers became aparent. Although [14] used a RED maximum dropping probability of $1/d = 1/10\,000$, the smallest value implemented in the 7609 routers is 1/255. Moreover, other routers impose the restriction that the difference between the maximum and minimum thresholds must be a power of 2. Neither of these affected the current experiment. However, discovering these constraints in a lab environment is easier than in on a live WAN, where many other factors cause deviations from the expected results.

Although still very preliminary, this study demonstrates the usefulness of WAN-in-Lab as an intermediate between experiments on emulated networks and live networks.

VI. Future Directions

Both the hardware and software components of WAN-in-Lab are still evolving, in the following ways.

A. Tools for simpler reconfiguration

WAN-in-Lab is able to switch automatically between different configurations. However, specifying the individual configurations currently requires users to specify specific router interfaces, IP addresses and in some case routing tables. Much of this can be automated, reducing the work required to specify a new experiment and, more importantly, simplifying the learning-curve for new users.

B. Benchmarking

Our current research involves developing a higher layer software infrastructure for performance benchmarking of different TCP variants. This will involve developing a measurement capability which can identify the queueing and burstiness of traffic, and correlate these measurements with the internal state of the protocols, such as window sizes and timeout thresholds.

A website will be created, at which a developer of a new protocol can submit a patch relative to a recent Web100-enabled Linux kernel, select from a standard set of tests, and then receive a comparison of the performance of the new protocol against existing protocols. While WAN-in-Lab will provide a standard platform for all researchers to use to compare their results, it is only one point in the price-abstraction tradeoff. We encourage the operators of networks based on emulated links to implement the same tests, to investigate the difference in performance that may result.

C. Monitoring

The aim of WAN-in-Lab is to provide a realistic yet controlled environment, which enables detailed monitoring of all aspects of protocol operation. At low speeds, `TCPdump` is useful for examining raw traffic, but at higher speeds, additional hardware is required. Arrangements are being made to attach real time analysis equipment, such as Endace DAG-MON systems, directly onto the OC48 stream. This will allow line-rate measurements of queue sizes and the burstiness of data anywhere in the core network.

Another approach is to monitor cumulative statistics such as total packet loss, aggregate throughputs, RED/QoS statistics, and queue levels. The benchmarking system described above will incorporate an SNMP system that periodically queries our routers and other network elements to obtain such statistics.

D. Community involvement

WAN-in-Lab is an open resource, intended for use by the entire networking community. This may require stricter enforcement of the time-sharing mechansims. WiL currently has a booking system to co-ordinate different projects, and automatically reconfigure the hardware between experiments. However, users from other projects can interfere by manually reconfiguring resources. More systematic resource allocation may involve the adoption of the eWAN framework developed at INRIA [17] and Caltech's MonALISA framework [18].

By integrating easy network topology reconfiguration along with powerful benchmarking and monitoring tools, we hope to provide a single platform where theory development, implementation, experiments, and deployment inform and influence each other intimately. Design, construction, and configuration details are available at the website ⟨http://WiL.cs.caltech.edu⟩.

VII. Acknowledgements

References

[1] C. Jin, D. X. Wei, S. H. Low, G. Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot, and S. Singh. "FAST TCP: From theory to experiments". *IEEE Network*, 19(1):4–11, 2005.

[2] C. Jin, D. X. Wei, and S. H. Low. "TCP FAST: Motivation, architecture, algorithms, performance", in *Proc. IEEE Infocom*, March 2004.

[3] S. Floyd. "HighSpeed TCP for large congestion windows". Internet draft draft-floyd-tcp-highspeed-02.txt, work in progress, http://www.icir.org/floyd/hstcp.html, February 2003.

[4] R. N. Shorten and D. J. Leith. "H-TCP: TCP for high-speed and long-distance networks". in *Proc. PFLDnet*, Argonne, 2004. http://hamilton.ie/net/htcp3.pdf.

[5] T. Kelly. "Scalable TCP: Improving performance in highspeed wide area networks". *Computer Communication Review*, 32(2), April 2003.

[6] L. Xu, K. Harfoush, and I. Rhee. "Binary increase congestion control for fast long distance networks", in *Proc. INFOCOM*, March 2004.

[7] H. Shimonish, T. Hama and T. Murase, "TCP-Adaptive Reno for improving efficiency-friendliness tradeoffs of TCP congestion control algorithm", in *Proc. PFLDnet*, Feb. 2006, pp 87–91.

[8] B. Wydrowski, L. L. H. Andrew, and M. Zukerman, "MaxNet: A congestion control architecture for scalable networks", *IEEE Commun. Lett.*, 7:511513, Oct. 2003.

[9] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb and A. Joglekar. "An integrated experimental environment for distributed systems and networks", in *Proc. Fifth Symp. Operating Systems Design and Implementation*. Dec. 2002, pp. 255–270.

[10] Ultralight: An ultrascale information system for data intensive research. http://www.ultralight.org.

[11] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. "A blueprint for introducing disruptive technology into the Internet", in *Proc. ACM Workshop on Hot Topics in Networking (HotNets)*, October 2002.

[12] A. Tang, J. Wang and S. Low. "Counter-intuitive throughput behaviors in networks under end-to-end control", *IEEE/ACM Trans. Networking*, 14(2):355-368, Apr 2006

[13] M. Suchara, R. Witt and B. Wydrowski. "TCP MaxNet – implementation and experiments on the WAN in Lab", in *Proc. IEEE International conference on Networks (ICON)*, p.6, 16–18 Nov 2005.

[14] A. Tang, D. Wei, S. H. Low and M. Chiang. "Heterogeneous congestion control: Efficiency, fairness and design", in *Proc. IEEE International Conference on Network Protocols (ICNP)*, 12–15 Nov 2006.

[15] A. Tang, J. Wang, S. Hegde and S. H. Low. "Equilibrium and fairness of networks shared by TCP Reno and Vegas/FAST", *Telecommunications Systems*, 30(4):417–439, Dec 2005.

[16] S. H. Low. "A duality model of TCP and queue management algorithms". *IEEE/ACM Trans. Networking*, 11(4):525–536, Aug 2003

[17] http://www.ens-lyon.fr/LIP/RESO/software/EWAN.

[18] MonALISA web page, http://monalisa.cern.ch/MONALISA.