Center for Networked Systems

TECHNOLOGIES AND FOUNDATIONS
FOR ROBUST AND SECURE NETWORKED SYSTEMS

UNIVERSITY OF CALIFORNIA, SAN DIEGO

SUMMER 2012 NEWSLETTER

QUARK: DESIGNING A SECURE BROWSER

Over the past 15 years, Web-based applications have become commonplace. They are used for a wide variety of activities, from interacting with a bank or healthcare provider to managing business functions or spending leisure time interacting on a social network. As a result of this ubiquity, hackers have learned to exploit security flaws in order to access the wealth of private information tracked and accessed by Web browsers. Indeed, security breaches have become so frequent that there are now annual competitions where security experts show off their hacking chops by breaking into the latest versions of browsers, and companies such as Google pay cash prizes to people who report bugs that pose a security threat.

Because users generally want a browser that is trustworthy, secure browsing is a high priority for software developers. Yet current browsers are in fact very fragile. They are complex pieces of software with rich features that allow for flexibility and programmability, and even small bugs can make the browser vulnerable to attack. Indeed, browser vulnerabilities have been used to infiltrate the internal networks of American defense contractors and leading tech firms. Attempts to improve browser security are often ad-hoc engineering efforts; and even when formal guarantees are provided, they come in the form of proofs over a model or idealization of the browser, not the browser itself. A buggy implementation can invalidate intended guarantees and still leave users open to attack. CSE prof. Sorin Lerner and co-authors Dongseok Jang and Zachary Tatlock explore a new approach to secure browser design in a paper* presented in August 2012 at the 21st USENIX Security Symposium, the foremost research conference on computer network security. Continued on page 4

SORIN LERNER

STEFAN SAVAGE

CNS HOSTS SECURITY DAY

In May 2012 CNS Director Stefan Savage hosted a special one-day workshop to highlight systems security research currently being conducted by researchers at UC San Diego. While much of the work was presented by current CNS faculty, the workshop was also an opportunity for CNS to engage with industry guests and non-affiliated UCSD faculty and researchers who are pursuing interdisciplinary approaches to solving security problems.

The day was broken into four sessions. Three sessions focused on sub-themes: security in embedded systems; web security; and e-crime. The fourth featured a graduate student poster session. The smaller size of the event compared to the larger-scale CNS Research Reviews provided a more intimate atmosphere for informal networking and collaborative discussion. Building on the success of the workshop, CNS aims to host other future workshops around singular themes.

Some Security Day highlights:

- In his paper, “Bit-Tight Design: A Clean-Slate Approach to Hardware-Assisted Security and Resiliency,” Professor Ryan Kastner proposed a new hardware foundation that can ensure that private keys are never leaked and that untrusted information will never be used in the making of critical decisions.
- Professor Sorin Lerner presented research on “Establishing Browser Security Guarantees through Formal Shim Verification” (see companion article this page). Continued on page 8

CNS MEMBERS

CISCO ERICSSON Google Microsoft NSF ORACLE

IN THIS ISSUE

GOOGLE GRANTS CNS RESEARCHERS TWO AWARDS

In July, Google announced that recipients in the latest round of grants from its prestigious Research Award Program included two researchers from the Center for Networked Systems (CNS).

- Research Scientist Kenneth Yocum garnered Google support for his project "An App-Store Framework for Data Center Networks;" and
- CSE Professor Hovav Shacham received funding to support his work on "Quantifying Browser Fingerprinting on the Web."

Google's Research Award Program aims to "fund world-class research at top universities, facilitate interaction between Google and academia, and support projects whose output will be made openly available to the research community." Shacham and Yocum join an accomplished group of CNS faculty and researchers who have received similar awards from Google in recent years.

Google has been a close collaborator and member of CNS since 2004.

GOOGLE RECOGNIZES CNS ALUMNI

Google recently honored a team of current and former CNS graduate students for their work on the paper, "TCP Fast Open." It was published in 2011 in the Proceedings of CoNEXT, the 7th International Conference on Emerging Networking Experiments and Technologies. In its notification to the researchers, Google stated that the purpose of the award was "to better recognize scientific contribution/excellent papers by Googlers." The researchers collaborated on the design and implementation of TCP Fast Open, a protocol that decreases the delay experienced by short TCP transfers. The research team consisted of Sivasankar Radhakrishnan, Yuchung Cheng, Jerry Chu, Arvind Jain, and Barath Raghavan. Cheng, Chu, and Jain are UC San Diego alumni currently employed by Google. Radhakrishnan is currently enrolled in the CSE department as a Ph.D. student, and Raghavan is a CSE alumnus who now works as a research scientist at UC Berkeley.

Their paper is available online at http://cseweb.ucsd.edu/~ssradhak/Papers/TCP_Fast_Open-CoNEXT_2011-camera_ready.pdf.

GRADUATING STUDENTS

Abhijeet Bhorkar, who studied with ECE Prof. Tara Javidi, received his Ph.D. in March 2012 after defending his dissertation on "Multi-hop Routing for Wireless Mesh Networks."

Stephen Checkoway defended his doctoral dissertation, "Low-Level Software Security: Exploiting Memory Safety Vulnerabilities and Assumptions," in June 2012. His advisor was CSE Prof. Hovav Shacham. Dr. Checkoway will become a Research Professor at John Hopkins University.

Christopher Kanich was co-advised by CSE Prof. Stefan Savage and Prof. Geoffrey Voelker, and earned his Ph.D. in June 2012. His dissertation was on "Characterizing Internet Scams through Underground Infrastructure Infiltration." Kanich accepted a position as an Assistant Professor in the University of Illinois at Chicago.

John McCullough received his Ph.D. in July 2012 after defending his dissertation on "Enabling Efficiency in Data Center Systems." His advisor was CSE Prof. Alex Snoeren. McCullough joined Google as a Software Engineer.

Shervin Sharifi was advised by CSE Prof. Tajana Rosing prior to receiving his Ph.D. in June 2012. His dissertation was on "Accurate Temperature Sensing and Efficient Dynamic Thermal Management in MPSoCs." Sharifi accepted a position as a Staff Engineer at Qualcomm.

Cynthia Taylor defended her doctoral dissertation in June 2012 on "The Networked Device Driver Architecture: A Solution for Remote I/O." She earned her Ph.D. under CSE Prof. Joseph Pasquale. Taylor is now an Assistant Professor of computer science at Oberlin College.

Meg Walraed-Sullivan earned her CSE Ph.D. in July 2012. She was co-advised by CSE Prof. Amin Vahdat and Prof. Keith Marzullo. Walraed-Sullivan's dissertation was on "Scalable, Efficient, and Fault-Tolerant Data Center Networking." This fall she becomes a postdoctoral researcher at Microsoft Research.

Ding Yuan, a visiting Ph.D. student from the University of Illinois at Urbana-Champaign, was advised by CSE Prof. YY Zhou, and graduated in July 2012. He has accepted a position as Assistant Professor of computer science at University of Toronto.

Changkun Chen graduated in June 2012 with an M.S. degree, advised by CSE Prof. Tajana Rosing. He accepted a position at Qualcomm as a Developer.

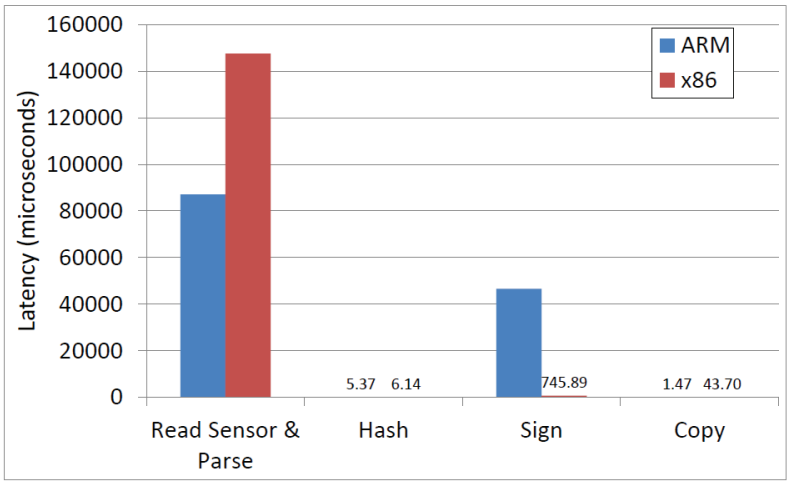
Utpal Kumar, advised by CSE Prof. Keith Marzullo, graduated with an M.S. in June 2012. He is now a Software Engineer at Arista Networks.

MICROSOFT INTERNSHIP YIELDS 'TRUSTED' RESEARCH

In summer 2010, CSE graduate student **Lonnie He Liu** left San Diego for a three-month internship at Microsoft Research in Redmond, WA. Little did he know when he left that the work he would complete over the summer would result in a paper presented at a major academic conference.

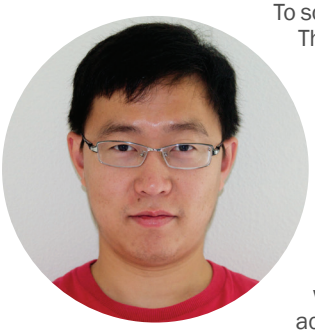
Liu was first author on the paper "Software Abstractions for Trusted Sensors" with Microsoft employees Stefan Saroiu, Alec Wolman, and Himanshu Raj. It was presented at the 10th International Conference on Mobile Systems, Applications, and Systems (MobiSys) in June 2012.

The research addressed the continuing need for trusted sensors used for mobile applications. Currently, mobile devices gather information about their surrounding environment or about their user and apply that information to a number of purposes. For example, location information can be used in a number of GPS-related applications, such as helping the user navigate his or her way to a restaurant, or to search for movie listings at the nearest movie theater. In another rapidly expanding field, health-related data can be gathered for purposes such as monitoring a person's blood-sugar level or calorie intake.



Latency breakdown on ARM and x86 platforms shows the latency of reading and parsing the sensor output, then hashing, signing and copying the results.

However, the usefulness of an application's end product is only as good as its input, and it is currently a trivial feat to fabricate sensor readings. False readings can be produced by the user or even, as Liu darkly describes, by a malicious outside actor who "by compromising a smart phone's software stack, for example by deploying a piece of malware" could fabricate GPS locations, camera shots, or health readings. One way to guard against this threat would be to validate the authenticity of sensor information at the time that it is taken, but this requires nuance in its implementation. For example, the authentication system must allow for the fact that there are occasions when the user might legitimately want to alter the original sensor information, such as when someone crops a photograph taken by their smart phone and then uploads it to a social networking site.



LONNIE HE LIU

To solve this obstacle, Liu and his Microsoft colleagues designed two software abstractions to expose trusted sensors. The first they call sensor attestation, which works by assigning a signature that becomes responsible for authenticating both the code produced by a given sensor reading as well as the sensor configuration that existed when the reading was made. They dubbed the second abstraction sensor seal. This accepts input as a secret, encrypts it, and marks it with a binding sensor policy. When the application attempts to un-encrypt the information, it checks to see if the policy specified at the time of sealing remains intact. If so, then the application is permitted to unseal the secret.

Because the team believed that trusted sensors should be developed for smart phones, tablets and laptops, they tested the implementation on both x86 and ARM platforms. And because of the complexity of issues related to different modes of sensor data, the team opted to work only with GPS-related sensor information where the user might want to provide invalid location information. For instance, an application that grants a user access to a secure file or database only when the user has been authenticated as being present within a specified location might be modified to report that the user is present in the secure venue even when he or she is not.

The researchers soon realized, however, that there was a potential problem related to user privacy that arises whenever signatures are attached to sensor data because, as Liu explains, "the sensor readings produced by a device are signed by the same entity." To address this issue, the team created a privacy layer on top of the sensitive data that manipulates and makes less identifiable the source of the data prior to it being attested and sealed. However, the manipulation of data introduces noise that can affect the reliability of the sensor readings, and this can become a liability when dealing with applications that utilize GPS and depend upon the accuracy of the readings. To test the severity of the problem, the group looked at GPS data of varying levels of differential privacy settings taken from smart phones over a period of time and plotted the work commutes of their users. The result was that the strongest privacy settings yielded unsupportably inaccurate results while moderate settings were acceptable for most uses.

The group emphasizes the preliminary nature of their work and the great complexity of the broader problem. Their research demonstrates the feasibility of sensor data authentication while taking some first steps to ensure the privacy of the mobile device user.

To read Lonnie He Liu's paper, visit http://research.microsoft.com/en-us/um/people/ssaroiu/publications/mobisys/2012/tenor.pdf.

NEWEST MEMBER OF CNS: MICROSOFT

In July 2012 CNS welcomed Microsoft as its newest Sponsor-level corporate member. Microsoft has long engaged CNS faculty on various levels of research collaboration. The Redmond, WA-based company has been an active participant in CNS events and also provided numerous summer internships. And of course, many CNS graduates at the Master's and Ph.D. levels went on to jobs at Microsoft. Becoming a CNS Sponsor is the next step in the natural evolution of a longstanding and mutually beneficial relationship.

Microsoft®

QUARK: DESIGNING A SECURE BROWSER (CONTINUED FROM PAGE 1)

CSE Ph.D. students Jang and Tatlock also previewed their work in a presentation to the first CNS Security Day in May 2012. They explained that previous verification techniques for browser security operate on a *model* or *abstraction* of the browser, and not on its actual implementation. This has created what Tatlock calls a 'formality gap', a discrepancy between what is verified and what is implemented. It is through this gap that hackers can infiltrate a browser even if it has been verified using strong formal methods.

DONGSEOK JANG

There is one known way to bridge this formality gap: implement the software in a proof assistant and use the proof assistant's interactive environment to formally prove, in full formal detail, that the software implementation is correct. More specifically, the programmer defines a *specification* stating what the code should do, and then uses the proof assistant to formally prove that the code in fact satisfies this specification, beginning with the most basic axioms and then building on them. Because of the precise way in which the program has been constructed, and the foundational nature of the proof, this kind of formal verification provides extremely strong guarantees.

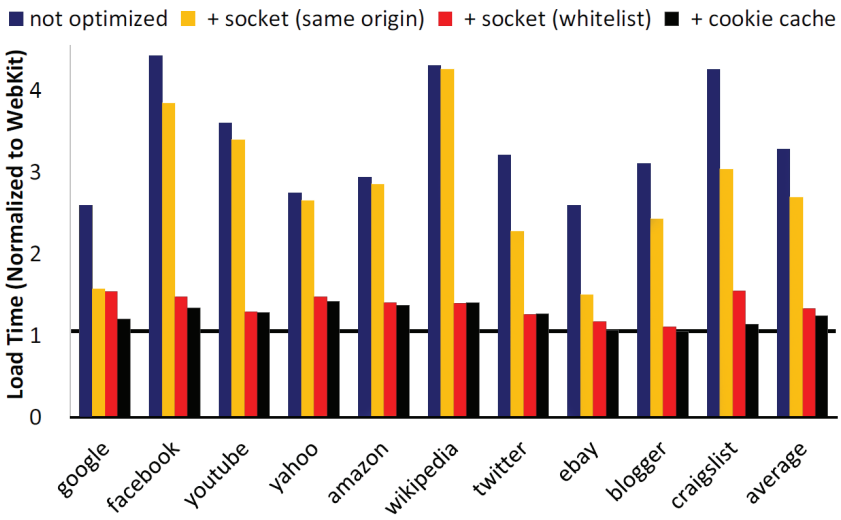
"Smaller is better."

In the past, however, this kind of verification has faced a number of practical barriers. One of the main challenges is that building formal proofs for applications with millions of lines of code is extremely time consuming, if not completely impossible. As a result, programmers using proof assistants have either restricted themselves to verifying stripped-down versions of their applications, or have had to expend *heroic* effort to perform the proofs, spending much more time and programmer-power than would be practical.

ZACH TATLOCK

Jang, Tatlock and Lerner speculated: What if there were ways to make the proofs easier by restricting the code that must be verified to a few hundred lines (as opposed to a few million)? The research team devised a technique, dubbed 'formal shim verification', for doing just this. Formal shim verification, says Lerner, consists of "creating a small browser kernel which mediates access to resources for all other browser components, and then formally verifying that this browser kernel is correct in a proof assistant." In other words, only a small part of the application that is vulnerable to outside scrutiny and attack will be verified using a proof assistant.

Following these design parameters, the team created Quark, a Web browser that uses a kernel-based architecture similar to Google Chrome. In particular, the heart of Quark is a small kernel responsible for mediating access to the rest of the application. Unlike Chrome's kernel, however, the Quark kernel has been verified in full formal detail using a proof assistant, allowing it to make strong guarantees about the security of the browser. As Jang explains, Quark "exploits formal shim verification and enables us to verify security properties for a *million* lines of code while reasoning about only a few hundred lines of code in the kernel." This critical distinction between a verified kernel and the non-verified components allowed the researchers to incorporate a number of state-of-the-art implementations into the non-verified parts of Quark, while still maintaining strong security guarantees. For example, Quark is able to use the WebKit open-source layout engine, the same layout engine used in Safari and Chrome. Using such realistic components has made Quark into a practical and usable browser, which can successfully run complex pages like Gmail, Google Maps, Facebook and Amazon.



Quark load times for the Alexa Top 10 Web sites, normalized to stock WebKit's load times. In each group the blue bar shows the unoptimized load time, the black bar shows load time in final, optimized version of Quark, and center bars show how additional optimizations improve performance.

During the prototyping phase of Quark, one thing became clear: "When forced to choose between adding complexity to the browser kernel," says Jang, "it was *always* better to keep the kernel as simple as possible." As a result, the current version of Quark is in some cases too simplistic. For example, it does not support some standard features of the web, such as third-party cookies, and in some cases it enforces non-standard security policies. Despite these current limitations, Quark is still capable of running many complex Web applications, including Facebook and Gmail.

Tatlock, Jang and Lerner are determined that this is only the first implementation of Quark. They already have some ideas about how to include a number of standard browser features without severely complicating their kernel or having to work through a fundamental redesign. Concludes Tatlock: "Our approach will handle more standard policies given design changes to our prototype and further engineering effort."

*__Establishing Browser Security Guarantees through Formal Shim Verification__, Dongseok Jang, Zachary Tatlock and Sorin Lerner, *Proceeding of the 21st USENIX Security Symposium*, August 2012. http://goto.ucsd.edu/quark/usenix12.pdf

STUDYING LOGGING IN THE WILD

Ever since the earliest development of technology, artisans have enumerated best practices for maintaining the quality of their craft. Following in this tradition, software developers evolved a number of best practices to assist in managing the software they write. One of the most conventional of these is the practice of logging.

Software log messages were developed to record information during a program's execution. Log data is used for a variety of purposes: to diagnose outages, to analyze trending failures, to develop a granular understanding of user experiences, to audit functionality, and more. Logging's value to the running and maintenance of software is underscored by its universal use – especially by commercial operators who presumably would not engage in logging if its costs outweighed the benefits.

Despite the universal agreement over its usefulness and necessity, however, logging has yet to become a standardized practice. When, where, how often and what to log are decisions left completely up to the each developer's discretion. In the commercial arena, logging is usually a fringe feature provided by vendors, and generally in this case, logging is used to document failures as an 'afterthought.'

YY ZHOU

To computer scientists in CSE whose research attempts to improve logging methods, this level of arbitrariness is unsustainably inefficient. According to a CSE Professor Yuanyuan (YY) Zhou and Ph.D. students Ding Yuan and Soyeon Park, more should be known about logging given its importance to developers. They embarked on the largest study to date to gather information about logging practices in the real world.

Among the first questions they tackled: Does it make a difference how logging is executed? If not, then there would be little need for taking the time and effort to impose order on the process. On the other hand, if certain patterns and problems are identified, they could point the way toward the areas of greatest need for the development of best practices.

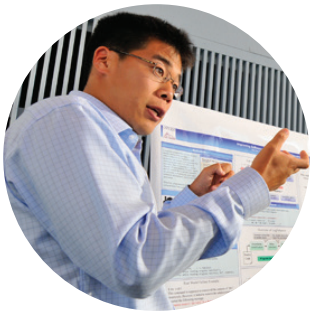
In their study, Zhou's group looked at logging in parts of Apache http, OpenSSH, PostreSQL, and Squid, four widely-used, open-source pieces of software.

By analyzing the density of log messages in the source code and in the revision record, they estimated the pervasiveness of logging. Then, by studying developers' modifications to their own log messages, they were able to estimate how efficient logging is as a process. Zhou, Yuan and Park could also extrapolate which logging aspects are most important to correct by looking at which ones developers spend the greatest time and effort correcting.

SOYEON PARK

Their findings confirmed what the software development community had already intuited: logging is a pervasive and inherently important practice. Across all the software studied, it was shown that on average every 30 lines of code contain one line of logging code. Proof of the usefulness of logging emerged when it was demonstrated that log messages reduced the median time it took to diagnose failures between 1.4 to 3 times. Despite the vital nature of the activity though, log messages are frequently incorrect the first time they are entered – and must be subsequently modified.

In fact, 18 percent of all committed revisions to software are made to logging code, which is completely out of proportion to the ratio of logging code to the body of code that comprises a piece of software. Perhaps most disturbingly, the researchers discovered that 33 percent of modifications to logging code are afterthoughts that occur after a failure happens and logs are needed. This means that one third of logging code is entered in a subjective, non-systematic way that adversely affects log quality, which developers must often correct several times over.



DING YUAN

Having established the importance of a practice that contains potential for improvement, the team set out to develop a tool to detect problem cases, such as log messages that were afterthoughts and had yet to be identified by developers. The checker operates on the assumption that if logging code within similar snippets of code is inconsistent with average verbosity levels, then either the logging or the source code must be incorrect. Using this approach, the team succeeded in detecting 138 problematic cases in the software studied – 24 of which had been confirmed and fixed by the software developers.

By using the checker, the team revealed that opportunities exist for the developers of tools, compilers, and languages to improve the efficiency of logging as currently practiced, and the resulting improvements could save developers' time and expertise that could be better spent elsewhere.

To read "**Characterizing Logging Practices in Open-Source Software**," visit http://opera.ucsd.edu/paper/log_icse12.pdf.

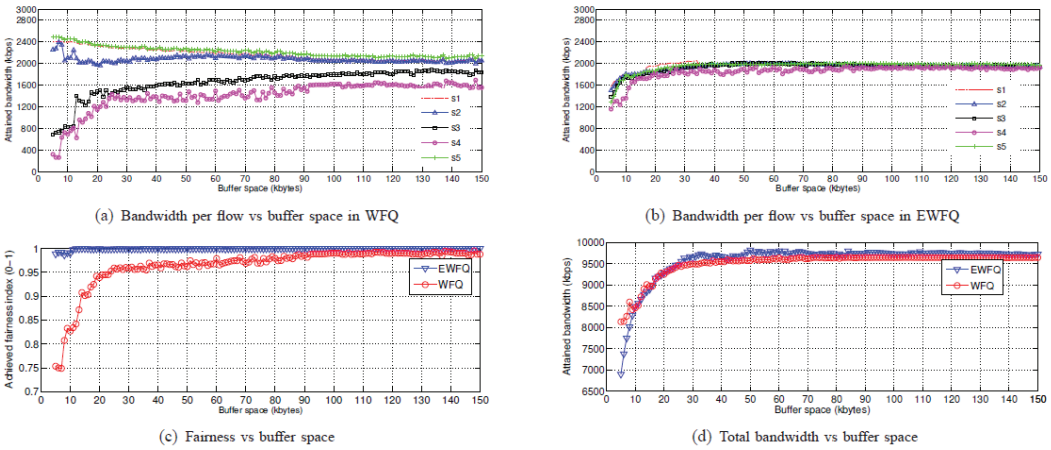
WEIGHTED FAIR QUEUING WITH DIFFERENTIAL DROPPING

The explosion of traffic on the Internet has forced service providers to rationalize more carefully the way they apportion bandwidth to their users. Whereas bandwidth once was meted out evenly according to consumer demand, providers have begun to create classes of users and applications based on their priorities of importance and perspective on profitability. "High value" applications (e.g., Voice-over-IP or games) tend to be given more bandwidth than "low value" traffic such as peer-to-peer file sharing or multimedia streaming. These are issues facing not just the commercial Internet, but also multi-tenant data centers and enterprise-wide networks.

In order to manage traffic more efficiently, different algorithms have been implemented. One such algorithm, Weighted Fair Queuing (WFQ), was successful in providing a fine-grained imposition of fairness, but its exceptional complexity demands on both CPU and buffer space when run at scale prohibit its use from some switches and routers. In order to reduce complexity, therefore, a packet-dropping scheme has been proposed, and while it helps WFQ to reduce resource utilization, the scheme also degraded the precision with which fairness was allocated.

In a recent paper, CSE Ph.D. student Feng Lu, with CSE faculty co-authors Geoffrey M. Voelker and Alex C. Snoeren, proposed an Enhanced Weighted Fair Queuing (EWFQ) algorithm that successfully integrates the effective traffic management of WFQ with a differential packet-dropping scheme that reduces CPU usage and the unreasonably onerous buffering requirements of WFQ, even when scaling to large numbers of flows and traffic classes.

EWFQ achieves this integration by dynamically repartitioning the buffer to share bandwidth fairly in a far wider variety of traffic-flow scenarios than WFQ or any other comparable algorithm is capable of handling. Additionally, the research team designed EWFQ to possess other advantages. It is self-tuning, does not require active queue management or operator intervention, and achieves more with a greatly reduced demand being placed upon buffer memory. With further testing and advanced iterations of the current algorithm, the research team hopes to reintroduce a higher level of fairness to resource management in the Internet and large-scale data centers.



Comparison between Weighted Fair Queuing (WFQ) and Enhanced WFQ (EWFQ) on fairness and attained bandwidth with respect to buffer space

To read the paper, "**Weighted Fair Queuing with Differential Dropping**," visit http://cseweb.ucsd.edu/~snoeren/papers/ewfq-infocom12.pdf.

INITIALLY NEGLECTED, WORK NOW SEEN AS "MOST INFLUENTIAL"

Innovation has been a constant fact of computer science ever since its inception, with the pace of change multiplying as the applications of computer science increased. Any moment in the development of this field has presented a host of problems and each problem has attracted a gamut of proposed solutions, only a few of which are attempted and even fewer of which become widely-adopted, next-generation practices. Sometimes the winning solutions are obvious at the moment of their proposal, while others take time to prove their worth.

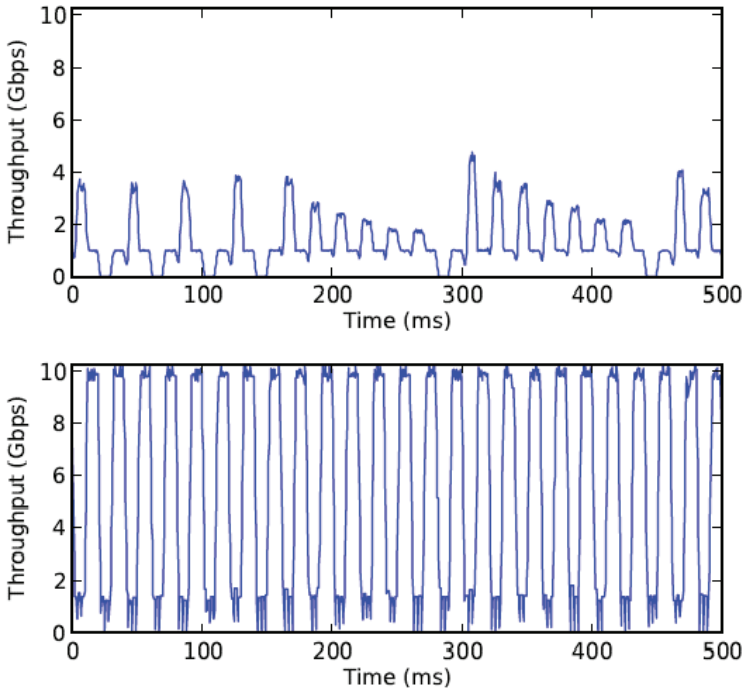
AMIN VAHDAT

In February, the ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC) celebrated its 25th year by identifying 20 of the "most influential" papers published in the history of its proceedings. They include a paper presented in 1997 whose lead author is CSE Professor and Google Fellow Amin Vahdat, entitled "WebOS: Operating System Services for Wide Area Applications."

In the paper, Vahdat and his colleagues at UC Berkeley and the University of Texas, Austin described a service, WebOS, which provided "basic operating systems services needed to build applications that are geographically distributed, highly available, incrementally scalable, and dynamically reconfiguring." The ideas outlined in the paper were a radical proposal for solving the then-growing problem of how to provide ease of use for wide-area resources. Though rejected from a number of conferences and widely overlooked at first, the basic framework described in Vahdat's paper is now commonly employed by such on-demand cloud service providers as Amazon Web Services's EC2 and Microsoft's Windows Azure.

"Cloud computing is placing increasingly stringent demands on datacenter networks," explains Bhanu Vattikonda, a Ph.D. student in the Computer Science and Engineering department who is trying to solve some of the biggest problems concerning modern datacenters.

Vattikonda is looking into features that prevent datacenters from meeting the demands placed upon them. In particular, he is concerned with the most commonly used architecture and protocols that were developed for the needs of wide-area or enterprise networks. While these technical parameters made sense for the hardware and application needs of the time when they were instituted, these older technologies and programs are beginning to fail to meet new constraints. The requirements of modern applications exceed and go far beyond the scope of the original capacities of these networks. The diversity of applications supported in datacenters is also constantly proliferating, and the requirements of each of these can vary radically from one application to the next. This means that the parameters are constantly widening for functionality and the means by which resources and workloads are balanced. The diversity of needs and constraints also means that designers cannot rely upon uniform requirements for supporting the applications they run.



Bandwidth seen by the receiver (top) in case of regular TCP adapting to changing link capacities, and (bottom) when the TCP flow is controlled by TDMA.

For some time, 'boutique' commercial hardware solutions that improve datacenter performance have been available for those who operate datacenters with especially high-level demands. However, when compared to the standard market practice of employing relatively low-cost Ethernet switches, the cost of equipping a large-scale datacenter with specialty solutions has been prohibitively high. Additionally, the capabilities of Ethernet have increased impressively in recent years, further ensuring its status as the default technology employed by datacenter designers.

Because of this, it made sense to Vattikonda and his colleagues, including Research Scientist George Porter and CSE Professors Amin Vahdat and Alex C. Snoeren, not to look for a hardware solution to the problem, but to accept the widely employed and economical Ethernet switches as the hardware of choice for the datacenter of the near future. This meant that their most profitable path for devising a solution would be to rethink the most troublesome aspects of current datacenter architectures and protocols that operate on this hardware. For example, more recent datacenter designs have made the standard use of TCP transport a liability because it impedes low-latency, high-throughput intercommunication. Therefore, the researchers asked themselves, what if the use of TCP could be foregone entirely?

"TCP was initially applied to problems of moving data from one network to another," says Vattikonda. "However, in certain environments... alternative transports have emerged to better suit the particular characteristics of these networks." The research team reasoned that the new patterns of communication currently found in datacenters no longer resemble the traditional wide-area networks for which TCP was designed – suggesting that now may be a good time to develop something to replace its use (just as Facebook reportedly eschews TCP in its operations).

"In order to dispense with TCP however," explains Vattikonda, "one must either replace its reliability and congestion control functionality, or remove the need for it." How to do this? The team considered what their goal parameters should be and ran across the solution. They wanted to minimize the possibility for congestion, queuing delays, and packet loss, and to do so they could impose upon the Ethernet network a time-division multiple access (TDMA) MAC layer. By using link-layer flow control protocols, they make explicit when to send packages and guarantee exclusive access for end hosts to the path they are assigned at any point in time.

As a result of their work, the team showed that they could reduce in-network queuing and relieve conflicts over buffer resources. The consequences of these changes are, Vattikonda says, "better performance for all-to-all transfer workloads and lower latency for request-response-type workloads."

Vattikonda emphasizes that one of the major advantages of this approach to the real-world consumer is that it has been developed to work with commodity network switches and it only requires minor modifications to the software running on the end hosts. Therefore, it could theoretically be adopted with a small investment of time, money and expertise.

Much work remains to be done to evaluate the efficiency and effectiveness of the centralized scheduler. "While our system architecture should allow the scheduler to react to switch, node and link failures," concludes Vattikonda, "we defer the evaluation of such a system to future work."

To read the paper, "**Practical TDMA for Datacenter Ethernet**," visit http://cseweb.ucsd.edu/~snoeren/papers/tdma-eurosys12.pdf.

CNS Summer Research Review

Wed-Thurs, August 15-16, 2012
Qualcomm Conference Center
First Floor, Jacobs Hall
UC San Diego

UPCOMING EVENTS

The CNS Summer 2012 Research Review will feature talks on the current technology challenges facing our member companies, updates on CNS research grants, a graduate-student research poster session and reception, and numerous opportunities for informal interaction and networking.

Attendance at the Research Review is limited to industry sponsors, invited guests, as well as CNS faculty, staff and graduate students. If you are interested in attending, please contact Kathryn Krane at kkrane@ucsd.edu or call 858-822-5964 with your inquiry.

CNS HOSTS SECURITY DAY (CONTINUED FROM PAGE 1)

• CSE postdoctoral scholar Nadia Heninger revealed unsettling results from the largest network study ever conducted of TLS and SSH servers: vulnerable keys are surprisingly widespread. In addition to a discussion of the survey findings, she outlined possible defenses and lessons for developers and the wider security community.

The presentations by Lerner and Heninger at CNS Security Day provided background on their respective papers accepted to the 21st USENIX Security Symposium, the leading computer systems and networking security conference. USENIX 2012 takes place Aug. 8-10 in Bellevue, WA. The previous day, at the 7th USENIX Workshop on Hot Topics in Security, Prof. Savage and Ph.D. students Feng Lu and Jiaqi Zhang will present a paper about "When Good Services Go Wild: Reassembling Web Services for Unintended Purposes."

Video of the CNS Security Day talks and PDF copies of the student posters are available for viewing by CNS Affiliates and Sponsors on the Members' Only section of the CNS site at http://bit.ly/MXhL9x.

For affiliates and sponsors who have difficulty logging in, please contact Kathryn Krane at kkrane@ucsd.edu.

MISSION AND OBJECTIVES OF CNS

The mission of CNS is to develop key technologies and frameworks for networked systems. By combining our research talents and strengths in partnership with industrial leaders, CNS achieves critical mass and relevant focus, accelerating research progress and creating key technologies, frameworks and systems understanding for robust, secure networked systems and innovative new applications. CNS also works to educate the next generation of top students with a perspective on industry-relevant research and to train students on how to continue their leadership throughout their careers. This is accomplished by bringing together leading faculty, students, and companies to investigate the most challenging, interesting and important problems in computer networks.

If you are interested in joining the Center, please contact Interim Director Stefan Savage at savage@cs.ucsd.edu.

GET CONNECTED

Stay up-to-date about upcoming CNS events, including lectures and Research Reviews, by signing up for the CNS Events RSS feed. To do so, visit the CNS website at http://cns.ucsd.edu and click on the link "CNS Events RSS Feed."

Calit2

UCSD Jacobs | Computer Science and Engineering