The Internet Protocol Journal

September 2012 Volume 15, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

Internet devices use various forms of timers and timestamps to determine everything from when a given e-mail message arrives to the number of seconds since a particular device was rebooted. Most systems use the *Network Time Protocol* (NTP) to obtain the current time from a large network of Internet time servers. NTP will be the subject of a future article in this journal. This time we will focus our attention on the *Leap Second*, which is occasionally applied to *Coordinated Universal Time* (UTC) in order to keep its time of day close to the *Mean Solar Time*. Geoff Huston explains the mechanism and describes what happened to some Internet systems on July 1, 2012, as a result of a leap second addition.

The Internet of Things (IoT) is a phrase used to describe networks where not only computers, smartphones, and tablets are Internet-aware, but also autonomous sensors, control systems, light switches, and thousands of other embedded devices. In our second article, David Lake, Ammar Rayes, and Monique Morrow give an overview of this emerging field which already has its own conferences and journals.

The *World Wide Web* became a reality in the early 1990s, thanks mostly to the efforts of Tim Berners Lee and Robert Cailliau. The web has been a wonderful breeding ground for new protocols and technologies associated with access to and presentation of all kinds of media. The phrase *Web 2.0*, coined in 1999, has, per Wikipedia, "...been used to describe web sites that use technology beyond the static pages of earlier web sites." David Strom argues that the term is no longer appropriate and that we have moved on to a new phase of the web, dominated by mobile devices and Social Networking.

The last few years have seen great advances in Internet-based collaboration tools. Sometimes referred to as *Telepresence*, these systems allow not only high-quality audio and videoconferencing, but also the use of shared whiteboards and other presentation material. In our final article, Pat Jensen describes one important component of such systems, namely the *Binary Floor Control Protocol* (BFCP), which the IETF's XCON Centralized Conferencing working group has developed.

As always we welcome your feedback on anything you read in this journal. Contact us by e-mail at ipj@cisco.com

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Leaping Seconds

by Geoff Huston, APNIC

The tabloid press is never lost for a good headline, but in July 2012 this one in particular caught my eye: "Global Chaos as Moment in Time Kills the Interwebs."[1] I am pretty sure that "global chaos" is somewhat "over the top," but a problem did happen on July 1 this year, and yes, it affected the Internet in various ways, as well as affecting many other enterprises that rely on IT systems. And yes, the problem had a lot to do with time and how we measure it. In this article I will examine the cause of this problem in a little more detail.

What Is a Second?

I would like to start with a rather innocent question: What exactly is a *second?* Obviously it is a unit of time, but what defines a second? Well, there are 60 seconds in a minute, 60 minutes in an hour, and 24 hours in a day. That information would infer that a "second" is 1/86,400 of a day, or 1/86,400 of the length of time it takes for the Earth to rotate about its own axis. Yes?

Almost, but this definition is still a little imprecise. What is the frame of reference that defines a unit of rotation of the Earth? As was established in the work a century ago in attempting to establish a frame of reference for the measurement of the speed of light, these frame-of-reference questions can be quite tricky!

What is the frame of reference to calibrate the Earth's rotation about its own axis? A set of distant stars? The Sun? These days we use the Sun, a choice that seems logical in the first instance. But cosmology is far from perfect, and far from being a stable measurement, the length of time it takes for the Earth to rotate once about its axis relative to the Sun varies month by month by up to some 30 seconds from its mean value. This variation in the Earth's rotational period is an outcome of both the Earth's elliptical orbit around the Sun and the Earth's axial tilt. These variations mean that by the time of the March equinox the *Solar Day* is some 18 seconds shorter than the mean, at the time of the June solstice it is some 13 seconds longer, at the September equinox it is some 21 seconds shorter, and in December it is some 29 seconds longer.

This variation in the rotational period of the Earth is unhelpful if you are looking for a stable way to measure time. To keep this unit of time at a constant value, then the definition of a second is based on an ideal version of the Earth's rotational period, and we have chosen to base the unit of measurement of time on *Mean Solar Time*. This mean solar time is the average time for the Earth to rotate about its own axis, relative to the Sun.

This value is relatively constant, because the variations in solar time work to cancel out each other in the course of a full year. So a second is defined as 1/86,400 of mean solar time, or in other words 1/86,400 of the average time it takes for the Earth to rotate on its axis. And how do we measure this mean solar time? Well, in our search for precision and accuracy the measurement of mean solar time is not, in fact, based on measurements of the sun, but instead is derived from baseline interferometry from numerous distant radio sources. However, the measurement still reflects the average duration of the Earth's rotation about its own axis relative to the Sun.

So now we have a second as a unit of the measurement of time, based on the Earth's rotation about its own axis, and this definition allows us not only to construct a uniform time system to measure intervals of time, but also to all agree on a uniform value of absolute time. From this analysis we can make calendars that are not only "stable," in that the calendar does not drift forward or backward in time from year to year, but also accurate in that we can agree on absolute time down to units of minute fractions of a second. Well, so one would have thought, but the imperfections of cosmology intrude once again.

The Earth has the Moon, and the Earth generates a tidal acceleration of the Moon, and, in turn the Moon decelerates the Earth's rotational speed. In addition to this long-term factor arising from the gravitational interaction between the Earth and the Moon, the Earth's rotational period is affected by climatic and geological events that occur on and within the Earth[2]. Thus it is possible for the Earth's rotation to both slow down and speed up at times. So the two requirements of a second—namely that it is a constant unit of time and it is defined as 1/86,400 of the mean time taken for the Earth to rotate on its axis—cannot be maintained. Either one or the other has to go.

In 1955 we went down the route of a standard definition of a second, which was defined by the *International Astronomical Union* as 1⁄31,556,925.9747 of the 1900.0 *Mean Tropical Year.* This definition was also adopted in 1956 by the *International Committee for Weights and Measures* and in 1960 by the *General Conference on Weights and Measures*, becoming a part of the *International System of Units* (SI). This definition addressed the problem of the drift in the value of the mean solar year by specifying a particular year as the baseline for the definition.

However, by the mid-1960s this definition was also found to be inadequate for precise time measure-ments, so in 1967 the SI second was again redefined, this time in experimental terms as a repeatable measurement. The new definition of a second was 9,192,631,770 periods of the radiation emitted by a Caesium-133 atom in the transition between the two hyperfine levels of its ground state.

Leaping Seconds

So we have the concept of a second as a fixed unit of time, but how does this relate to the astronomical measurement of time? For the past several centuries the length of the *Mean Solar Day* has been increasing by an average of some 1.7 milliseconds per century. Given that the solar day was fixed on the Mean Solar Day of the year 1900, by 1961 it was around a millisecond longer than 86,400 SI seconds. Therefore, absolute time standards that change the date after precisely 86,400 SI seconds, such as the *International Atomic Time* (TAI), get increasingly ahead of the time standards that are rigorously tied to the Mean Solar Day, such as *Greenwich Mean Time* (GMT).

When the *Coordinated Universal Time* (UTC) standard was instituted in 1961, based on atomic clocks, it was felt necessary that this time standard maintain agreement with the GMT time of day, which until then had been the reference for broadcast time services. Thus, from 1961 to 1971 the rate of broadcast time from the UTC atomic clock source had to be constantly slowed to remain synchronized with GMT. During that period, therefore, the "seconds" of broadcast services were actually slightly longer than the SI second and closer to the GMT seconds.

In 1972 the *Leap Second* system was introduced, so that the broadcast UTC seconds could be made exactly equal to the standard SI second, while still maintaining the UTC time of day and changes of UTC date synchronized with those of *UT1* (the solar time standard that superseded GMT). Reassuringly, a second is now a SI second in both the UTC and TAI standards, and the precise time when time transitions from one second to the next is synchronized in both of these reference frameworks. But this fixing of the two time standards to a common unit of exactly 1 second means that for the standard second to also track the time of day it is necessary to periodically add or remove entire standard seconds from the UTC time-of-day clock. Hence the use of so-called leap seconds. By 1972 the UTC clock was already 10 seconds behind TAI, which had been synchronized with UT1 in 1958 but had been counting true SI seconds since then. After 1972, both clocks have been ticking in SI seconds, so the difference between their readouts at any time is 10 seconds plus the total number of leap seconds that have been applied to UTC.

Since January 1, 1988, the role of coordinating the insertion of these leap-second corrections to the UTC time of day has been the responsibility of the *International Earth Rotation and Reference Systems Service* (IERS). IERS usually decides to apply a leap second whenever the difference between UTC and UT1 approaches 0.6 second in order to keep the absolute difference between UTC and the mean solar UT1 broadcast time from exceeding 0.9 second.

The UTC standard allows leap seconds to be applied at the end of any UTC month, but since 1972 all of these leap seconds have been inserted either at the end of June 30 or December 31, making the final minute of the month in UTC, either 1 second longer or 1 second shorter when the leap second is applied. IERS publishes announcements in its *Bulletin C* every 6 months as to whether leap seconds are to occur or not. Such announcements are typically published well in advance of each possible leap-second date—usually in early January for a June 30 scheduled leap second and in early July for a December 31 leap second. Greater levels of advance notice are not possible because of the degree of uncertainty in predicting the precise value of the cumulative effect of fluctuations of the deviation of the Earth's rotational period from the value of the Mean Solar Day. Or, in other words, the Earth is unpredictably wobbly!

Between 1972 and 2012 some 25 leap seconds have been added to UTC. On average this number implies that a leap second has been inserted about every 19 months. However, the spacing of these leap seconds is quite irregular: there were no leap seconds in the 7-year interval between January 1, 1999, and December 31, 2005, but there were 9 leap seconds in the 13 years between 1985 and 1997, as shown in Figure 1. Since December 31, 1998, there have been only 3 leap seconds, on December 31, 2005, December 31, 2008, and June 30, 2012, each of which has added 1 second to that final minute of the month, at the UTC time of day.

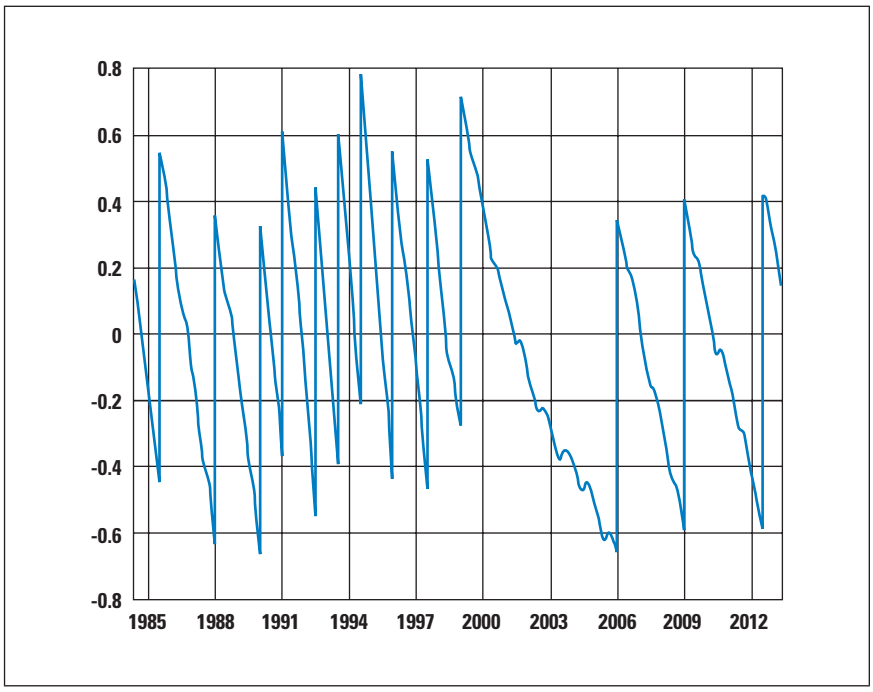


Figure 1: The difference between UT1 and UTC 1984–2012

Leaping Seconds and Computer Systems

The June 30, 2012 leap second did not pass without a hitch, as reported by the tabloid press. The side effect of this particular leap second appeared to include computer system outages and crashes—an outcome that was unexpected and surprising. This leap second managed to crash some servers used in the Amadeus airline management system, throwing the Qantas airline into a flurry of confusion on Sunday morning on July 1 in Australia. But not just the airlines were affected, because LinkedIn, Foursquare, Yelp, and Opera were among numerous online service operators that had their servers stumble in some fashion. This event managed to also affect some *Internet Service Providers* and data center operators. One Australian service provider has reported that a large number of its Ethernet switches seized up over a 2-hour period following the leap second.

It appears that one common element here was the use of the Linux operating system. But Linux is not exactly a new operating system, and the use of the *Leap Second Option* in the *Network Time Protocol* (NTP) [7–10] is not exactly novel either. Why didn't we see the same problems in early 2009, following the leap second that occurred on December 31, 2008?

Ah, but there *were* problems then, but perhaps they were blotted out in the post new year celebratory hangover! Some folks noticed something wrong with their servers on January 1, 2009. Problems with the leap second were recorded with Red Hat Linux following the December 2008 leap second, where kernel versions of the system prior to Version 2.6.9 could encounter a deadlock condition in the kernel while processing the leap second.[3]

> "[...] the leap second code is called from the timer interrupt handler, which holds *xtime_lock*. The leap second code does a *printk* to notify about the leap second. The *printk* code tries to wake up *klogd* (I assume to prioritize kernel messages), and (under some conditions), the scheduler attempts to get the current time, which tries to get *xtime_lock* => *deadlock*."[4]

The advice in January 2009 to sysadmins was to upgrade the systems to Version 2.6.9 or later, which contained a patch that avoided this kernel-level deadlock. This time it is a different problem, where the server CPU encountered a 100-percent usage level:

> "The problem is caused by a bug in the kernel code for high resolution timers *(hrtimers)*. Since they are configured using the CONFIG_HIGH_RES_TIMERS option and most systems manufactured in recent years include the *High Precision Event Timers* (HPET) supported by this code, these timers are active in the kernels in many recent distributions.

> "The kernel bug means that the *hrtimer* code fails to set the system time when the leap second is added. The result is that the *hrtimer* representation of the time taken from the kernel is a second ahead of the system time. If an application then calls a kernel function with a timeout of less than a second, the kernel assumes that the timeout has elapsed immediately after setting the timer, and so returns to the program code immediately. In the event of a timeout, many programs simply repeat the requested operation and immediately set a new timer. This results in an endless loop, leading to 100% CPU utilisation."[5]

Leap Smearing

Following a close monitoring of its systems in the earlier 2005 leap second, Google engineers were aware of problems in their operating system when processing this leap second. They had noticed that some clustered systems stopped accepting work during the leap second of December 31, 2005, and they wanted to ensure that this situation did not recur in 2008. Their approach was subtly different to that used by the Linux kernel maintainers.

Rather than attempt to hunt for bugs in the time management code streams in the system kernel, they noted that the intentional side effect of NTP was to continually perform slight time adjustments in the systems that are synchronizing their time according to the NTP signal. If the quantum of an entire second in a single time update was a problem to their systems, then what about an approach that allowed the 1-second time adjustment to be smeared across numerous minutes or even many hours? That way the leap second would be represented as a larger number of very small time adjustments that, in NTP terms, was nothing exceptional. The result of these changes was that NTP itself would start slowing down the time-of-day clock on these systems some time in advance of the leap second by very slight amounts, so that at the time of the applied leap second, at 23:59:59 UTC, the adjusted NTP time would have already been wound back to 23:59:58. The leap second, which would normally be recorded as 23:59:60 was now a "normal" time of 23:59:59, and whatever bugs that remained in the leap second time code of the system were not exercised.[6]

More Leaping?

The topic of leap seconds remains a contentious one. In 2005 the United States made a proposal to the *ITU Radiocommunication Sector* (ITU-R) Study Group 7's Working Party 7-A to eliminate leap seconds. It is not entirely clear whether these leap seconds would be replaced by a less frequent *Leap Hour,* or whether the entire concept of attempting to link UTC and the Mean Solar Day would be allowed to drift, and over time we would see UTC time shifting away from the UT1 concept of solar day time.

This proposal was most recently considered by the ITU-R in January 2012, and there was evidently no clear consensus on this topic. France, Italy, Japan, Mexico, and the United States were reported to be in favor of abandoning leap seconds, whereas Canada, China, Germany, and the United Kingdom were reportedly against these changes to UTC. At present a decision on this topic, or at the least a discussion on this topic, is scheduled for the 2015 *World Radio Conference*.

Although these computing problems with processing leap seconds are annoying and for some folks extremely frustrating and sometimes expensive, I am not sure this factor alone should affect the decision process about whether to drop leap seconds from the UTC time framework. With our increasing dependence on highly available systems, and the criticality of accurate time-of-day clocks as part of the basic mechanisms of system security and integrity, it would be good to think that we have managed to debug this processing of leap seconds.

It is often the case in systems maintenance that the more a bug is exercised the more likely it is that the bug will be isolated and corrected. However, with leap seconds, this task is a tough one because the occurrence of leap seconds is not easily predicted. The next time we have to leap a second in time, about the best we can do is hope that we are ready for it.

For Further Reading

The story of calendars, time, time of day, and time reference standards is a fascinating one. It includes ancient stellar observatories, the medieval quest to predict the date of Easter, the quest to construct an accurate clock that would allow the calculation of longitude, and the current constellations of time and location reference satellites. These days much of this material can be found on the Internet.

[0] Wikipedia, "Leap Second,"
http://en.wikipedia.org/wiki/Leap_second

[1] Herald Sun online,
http://www.heraldsun.com.au/news/leap-second-crashes-qantas-and-leaves-passengers-stranded/story-e6frf7jo-1226413961235

[2] "The deviation of the Mean Solar Day from the SI-based day, 1962–2010," graph in the Wikipedia article referenced earlier[0],
http://upload.wikimedia.org/wikipedia/commons/thumb/2/28/Deviation_of_day_length_from_SI_day_.svg/1000px-Deviation_of_day_length_from_SI_day_.svg.png

[3] Red Hat Bugzilla - Bug 479765, "Leap second message can hang the kernel,"
https://bugzilla.redhat.com/show_bug.cgi?id=479765

[4] "Re: Bug: Status/Summary of slashdot leap-second crash on new years 2008–2009,"
http://lkml.org/lkml/2009/1/2/373

[5] "Leap second bug in Linux wastes electricity," *The H Open,* July 3, 2012,
http://www.h-online.com/open/news/item/Leap-second-bug-in-Linux-wastes-electricity-1631462.html

[6] "Time, technology and leaping seconds," Google Official Blog, September 15, 2011,
http://googleblog.blogspot.de/2011/09/time-technology-and-leaping-seconds.html

[7] Burbank, J., Kasch, W., and D. Mills, "Network Time Protocol Version 4: Protocol and Algorithms Specification," RFC 5905, June 2010.

[8] Mills, D. and B. Haberman, "Network Time Protocol Version 4: Autokey Specification," RFC 5906, June 2010.

[9] Elliott, C., Haberman, B., and H. Gerstung, "Definitions of Managed Objects for Network Time Protocol Version 4 (NTPv4)," RFC 5907, June 2010.

[10] Lourdelet, B. and R. Gayraud, "Network Time Protocol (NTP) Server Option for DHCPv6," RFC 5908, June 2010.

Disclaimer

GEOFF HUSTON, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005. He served on the Board of Trustees of the Internet Society from 1992 until 2001.
E-mail: **gih@apnic.net**

The Internet of Things

by David Lake, Ammar Rayes, and Monique Morrow, Cisco Systems

Until a point in time around 2008 or 2009, there were more human beings in the world than devices connected to the Internet. That is no longer the case.

In 2010, the global average of connected devices per person was 1.84. Taking only those people who use the Internet (around 2 billion in 2010), that figure becomes 6 devices per person.[1] Chip makers such as ARM have targeted developments of low-power CPUs and predicts up to 50 billion devices connected by 2020.[2]

Today, most of these devices are entities that the user interacts directly with—a PC or Mac, smartphone, tablet, etc. But what is changing is that other devices used every day to orchestrate and manage the world we live in are becoming connected entities in their own right.

They consist not just of users interacting with the end devices—the source and treatment of the information garnered will now occur autonomously, potentially linking to other networks of similarly interconnected entities.

Growing to an estimated 25 billion connected devices by 2015, the rapid explosion of devices on the Internet presents some new and interesting challenges.[3]

A Definition of the Internet of Things

The *Internet of Things* (IoT) consists of networks of sensors attached to objects and communications devices, providing data that can be analyzed and used to initiate automated actions. The attributes of this world of things may be characterized by low energy consumption, auto-configuration, embeddable objects, etc. The data also generates vital intelligence for planning, management, policy, and decision making. In essence, the five properties that characterize the Internet of Things are as follows:

- *A Unique Internet Address* by which each connected physical object and device will be identified, and therefore be able to communicate with one another.
- *A Unique Location—can be fixed or mobile—within a network or system* (for example, a smart electricity grid) that makes sense of the function and purpose of the object in its specified environment, generating intelligence to enable autonomous actions in line with that purpose.
- *An Increase in Machine-Generated and Machine-Processed Information* that will surpass human-processed information, potentially linking in with other systems to create what some have called "the nervous system of the planet."

- *Complex New Capabilities in Security, Analytics, and Management,* achievable through more powerful software and processing devices, that enable a network of connected devices and systems to cluster and interoperate transparently in a "network of networks."
- *Time and Location Achieve New Levels of Importance* in information processing as Internet-connected objects work to generate ambient intelligence; for example, on the *Heating, Ventilation, and Air Conditioning* (HVAC) efficiency of a building, or to study soil samples and climatic change in relation to crop growth.

The concepts and technologies that have led to the IoT, or the interconnectivity of real-world objects, have existed for some time. Many people have referred to *Machine-to-Machine* (M2M) communications and IoT interchangeably and think they are the same. In reality, M2M is only a subset; IoT is a more encompassing phenomenon because it also includes *Machine-to-Human* communication (M2H). *Radio Frequency Identification* (RFID), *Location-Based Services* (LBS), *Lab-on-a-Chip* (LOC) sensors, *Augmented Reality* (AR), robotics, and vehicle telematics are some of the technology innovations that employ both M2M and M2H communications within the IoT as it exists today. They were spun off from earlier military and industrial supply chain applications; their common feature is to combine embedded sensory objects with communication intelligence, running data over a mix of wired and wireless networks.

What has really helped IoT gain traction outside these specific application areas is the greater commoditization of IP as a standard communication protocol, and the advent of IPv6 to allow for a unique IP address for each connected device and object. Researchers and early adopters have been further encouraged by advancements in wireless technologies, including radio and satellite; miniaturization of devices and industrialization; and increasing bandwidth, computing, and storage power.

All these factors have played a part in pushing the boundaries toward generating more context from data capture, communication, and analytics through various devices, objects, and machines in order to better understand our natural and man-made worlds. In exploring the relationship between the IoT and *Information-Centric Networking* (ICN), embedded distributed intelligence will be an important attribute for ICN. Context that is distributed as opposed to centralized is a core architectural component of the IoT for three main reasons:

- *Data Collection:* Centralized data collection and smart object management do not provide the scalability required by the Internet. Managing several hundreds of millions of sensors and actuators in a *Smart Grid* network, for example, cannot be done using a centralized approach.

- *Network Resource Preservation:* Network bandwidth is scarce and some smart objects are not mains-powered, meaning that collecting environmental data from a central point in the network unavoidably leads to using a large amount of the network capacity.
- *Closed-Loop Functioning:* The IoT needs reduced reaction times. For instance, sending an alarm via multiple hops from a sensor to a centralized system, which runs analytics before sending an order to an actuator, would entail unacceptable delays.

Service Management Systems (SMS) (also known as Management Systems, Network Management Systems, or back-end systems) are the brain in the IoT. SMS interacts with intelligent databases that contain *Intellectual Capital* (IC) information, contract information, and manufacturing and historical data. SMS also supports image-recognition technologies to identify objects, people, buildings, places, logos, and anything else that has value to consumers and enterprises. Smartphones and tablets equipped with cameras have pushed this technology from mainly industrial applications to broad consumer and enterprise applications.

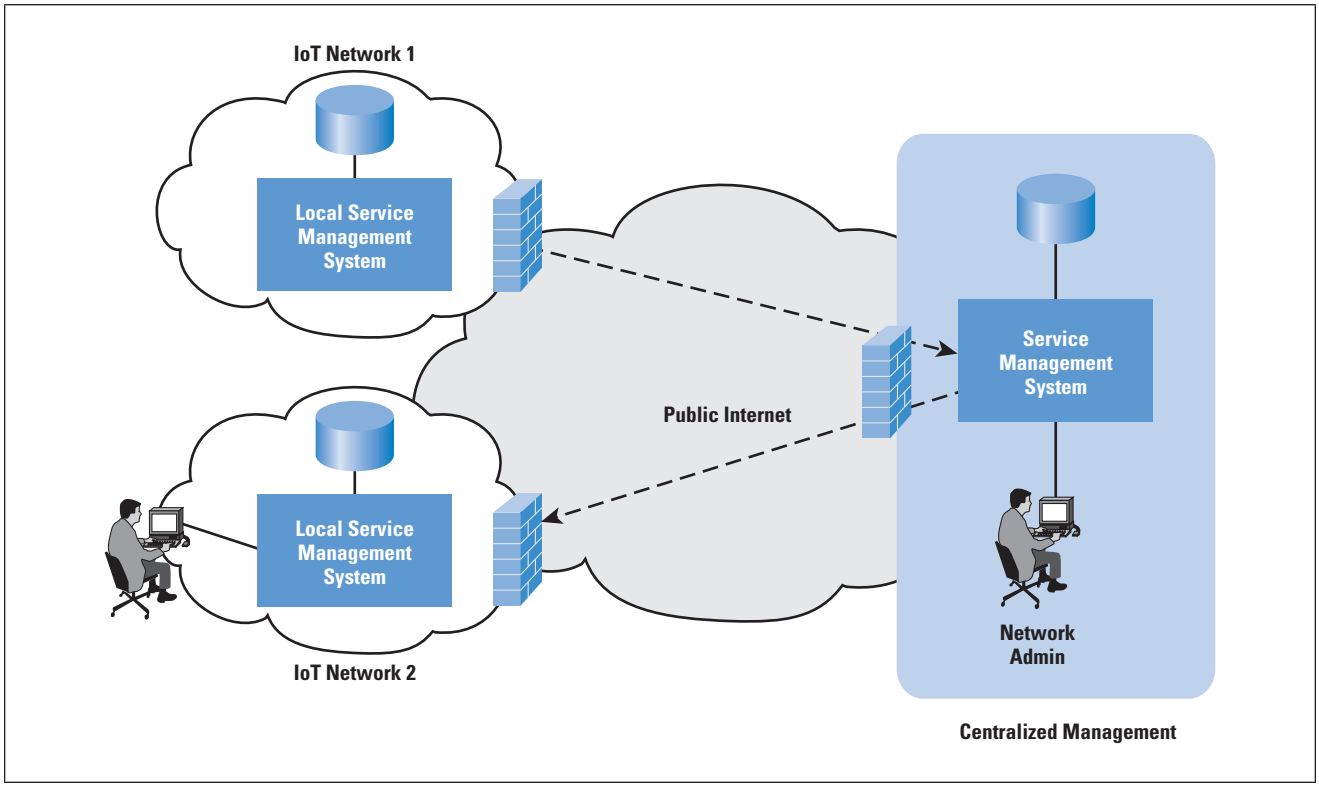
IC information includes intelligence of the vendor's (for example, Cisco) databases and systems such as contract DB, Manufacturing DB, and more importantly thousands of specific roles that are captured over the years by analyzing software bugs, technical support cases, etc.; that is, Cisco knows which devices were manufactured for which customers and with what features. Data collected by the collector is analyzed and correlated with the repository of proprietary Intellectual Capital, turning it into actionable intelligence to help network planners and administrators increase IT value, simplify IT infrastructure, reduce cost, and streamline processes.

Secure communications allow collected data to be sent securely from the agents or collection system to the SMS. SMS includes a database that stores the collected data and algorithms to correlate the collected data with Intellectual Capital information, turning the data into actionable intelligence that network planners and administrators can use with advanced analytics to determine the optimal solution for a problem (or potential problem) after the data is analyzed and corrected. More importantly, a secure mechanism allows the vendor to connect to the network remotely and take action. Secure communications also allows the SMS (automatically or via a network administer) to communicate back with the device to take action when needed.

However, centralized SMS for a large number of entities is very challenging given the near-real-time requirements and the effect on the network performance (see Figure 1). At the same time, centralized intelligence will be required for many IoT networks to interact with back-end centralized databases that are very difficult to distribute (for example, supplier Intellectual Capital databases).

This centralization is more demanding than the traditional multitier environments, servers, and back-end database types of applications where database caching was an effective approach to achieve high scalability and performance. Solution architects need to consider an optimal hybrid model that supports centralized and distributed systems at the same time. Distributed SMS may need to make sub-optimal decisions by using only narrow information to address real-time (or near-real-time) performance problems.

Figure 1: Typical Deployment of an IoT Network



Device and Data Security

The IoT will comprise many small devices, with varying operating systems, CPU types, memory, etc. Many of these devices will be inexpensive, single-function devices—for example, a temperature or pressure sensor—and could have rudimentary network connectivity. In addition, these devices could be in remote or inaccessible locations where human intervention or configuration is impossible.

The nature of sensors is such that they are embedded in what they are sensing—one can envisage a new workplace, hospital, or school construction project where the technology is introduced during the construction phase as part of the final fit rather than after completion as is common today. This paradigm in itself creates new challenges because the means of connectivity may exist only after the installation teams have left the site.

Additionally, methods must be taken to ensure that the authenticity of the data, the path from the sensor to the collector, and the connectivity authentication parameters cannot be compromised between the initial installation or configuration of the device and its eventual presence on the IoT infrastructure.

The challenges of designing and building IoT devices can be summarized as follows:

- IoT devices are typically small, inexpensive devices.
- They are designed to operate autonomously in the field.
- They may be installed prior to network availability.
- After deployment, these devices may require secure remote management.
- The computing platform may not support traditional security algorithms.

Because the IoT will not be a single-use, single-ownership "solution" with sources and the platform on which data may be consumed could be in different ownership, managerial, and connectivity domains, devices will be required to have equal and open access to numerous data consumers concurrently, while still retaining privacy and exclusivity of data where that is required between those consumers.

This requirement was neatly summarized by the IETF Security Area Directors as follows: "A house only needs one toaster even if it serves a family of four!"[4]

So we have seemingly competing, complex security requirements to be deployed on a platform with limited resources:

- Authenticate to multiple networks securely.
- Ensure that data is available to multiple endpoints.
- Manage the contention between that data access.
- Manage privacy concerns among multiple consumers.
- Provide strong authentication and data protection that cannot be compromised.

And we have to manage existing challenges that all network-attached devices have to contend with such as *Denial of Service* (DoS) attacks, transaction replays, compromised identity through subscriber theft, device theft, or compromised encryption.

These problems have particular relevance in the IoT, where the availability of data is of paramount importance. For example, a critical industrial process may rely on accurate and timely temperature measurement—if that sensor is undergoing a DoS attack, the process collection agent must understand that, and be able to either source data from another location or take evasive action.

It must also be able to distinguish between loss of data because of an ongoing DoS attack and loss of the device because of a catastrophic event in the plant. This ability could mean the difference between a safe shut-down and a major incident.

Authentication and authorization will require reengineering to be appropriate for the IoT. Today's strong encryption and authentication schemes are based on cryptographic suites such as *Advanced Encryption Standard* (AES), *Rivest-Shamir-Adelman* (RSA) for digital signatures and key transport, and *Diffie-Hellman* (DH) for key agreement. Although the protocols are robust, they make very high demands of the compute platform—resources that may not exist in all IoT-attached devices.

These authentication and authorization protocols also require a degree of user intervention in terms of configuration. However, many IoT devices will have limited access; initial configuration needs to be protected from tampering, stealing, and other forms of compromise between device build and install, and also for its usable life, which could be many years.

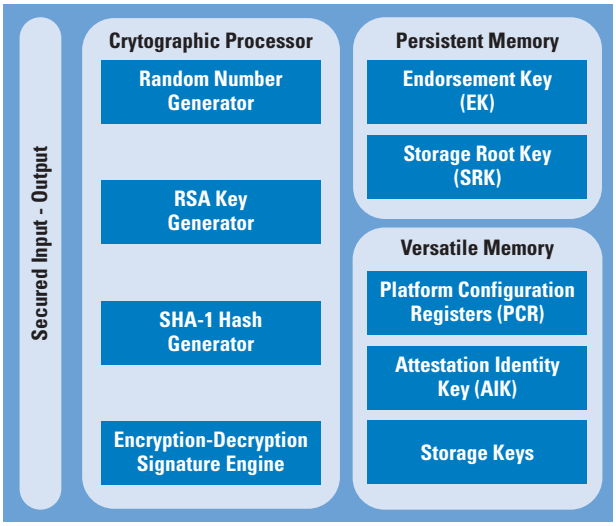
In order to overcome these difficulties, new authentication schemes that allow for strong authentication to many domains while building on the experience of today's strong encryption and authentication algorithms are required.

One possible approach could be to extend methodologies used in the PC industry such as the *Trusted Computing Group's Trusted Platform Model* (TPM).[5,6]

TPM-enabled devices are fitted at build time with a highly secure hardware device containing a variety of cryptographic elements. Keys and other factors known from this device by trusted third parties are then used in an attestation—a request to validate the authenticity of one device from known parameters.

Because the cryptographic keys are burned into the device during build and the signatures are known to a controlled, trusted third party, a high degree of confidence in the authenticity of the device being queried can be obtained. A typical TPM-compliant cryptographic chip is shown in Figure 2.

Figure 2: Trusted Platform Module



TPM has traditionally been limited by requiring access not only between the devices, but also to a trusted third party. In the IoT, where connectivity may be transient, this requirement is obviously a limitation. Extensions to the TPM to allow for high-confidence attestation between devices without involving a third party have been built; for example, *Direct Anonymous Attestation* (DAA).[7]

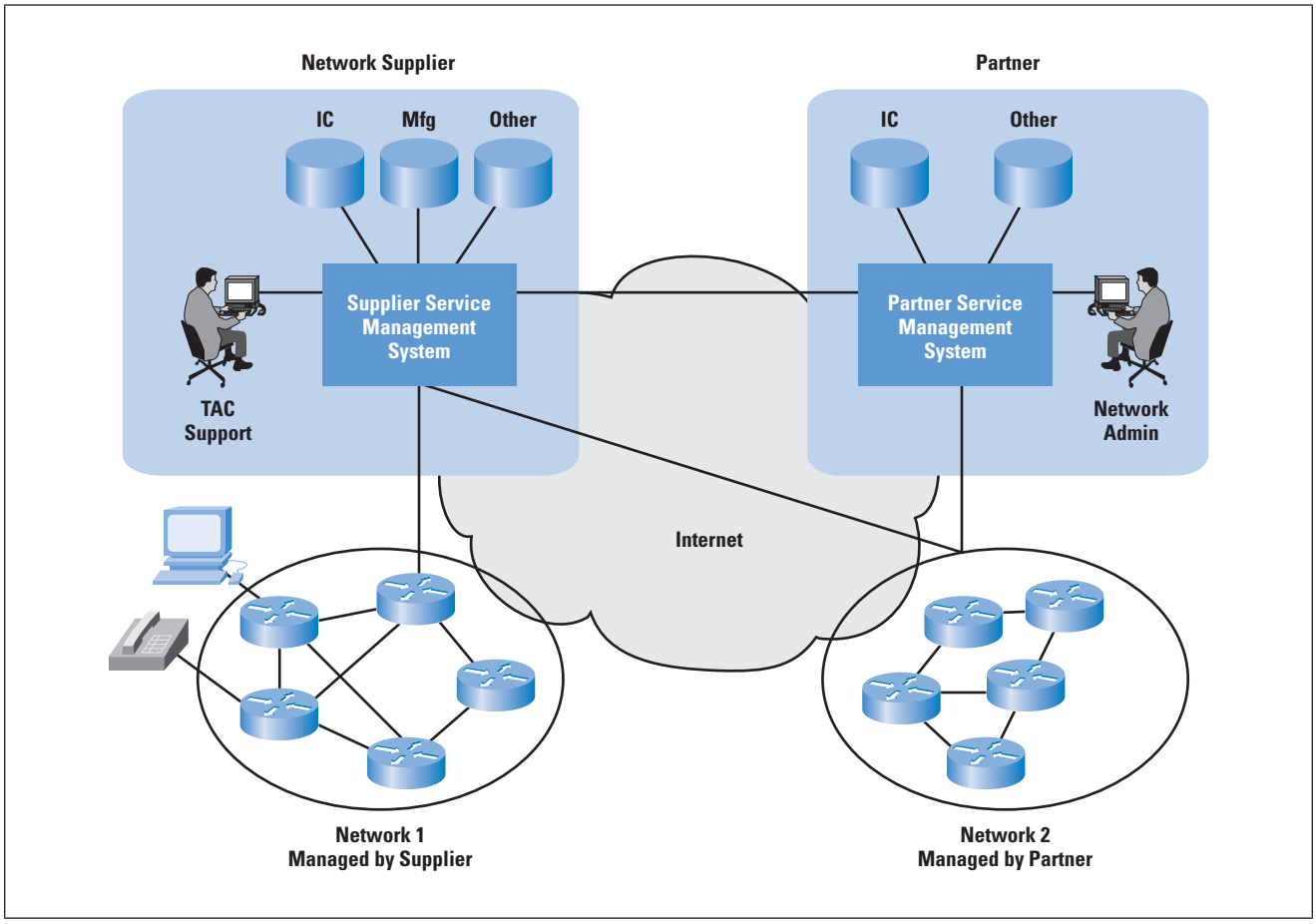
Other elements in security that could be considered include strong authentication between the device and the network attachment point (such as through electrical signatures at the *Media Access Control* [MAC] layer), application of geographic location and privacy levels to data, strengthening of other network-centric methods such as the *Domain Name System* (DNS) and the *Dynamic Host Configuration Protocol* (DHCP) to prevent attacks, and adoption of other protocols that are more tolerant to delay or transient connectivity (such as *Delay Tolerant Networks*).[8]

An IoT Case Study

The concepts behind the IoT allow management of assets within an enterprise with responsibility shared among customer, partner, and manufacturer in a manner that would previously have been difficult to control.

A typical IT network consists of routers, switches, IP phones, telepresence systems, network management systems such as call managers, data center managers, and many other entities (also known as "machines") with unique identification (for example, serial number, MAC address, or other address (for example, IP address). Such a solution is depicted in Figure 3.

Figure 3: Example of Smart Services



The system has the following components:

- The IT IP-based network: The network typically is owned by a business customer or an end customer (for example, a small business network). It includes IP devices that may be managed either by the supplier (via service contract), by a third party, Partner 2, or by the customer network administer.
- Smart agent or collection system (or sensor): An external collection system (for example, a server) or smart agent or collection systems on the managed devices gather the device and network information via numerous methods including *Simple Network Management Protocol* (SNMP) requests, *Command-Line Interface* (CLI) commands, syslog, etc. Collected information includes inventory, security data, performance data such as service-level agreement parameters, fault messages, etc.
- Supplier or partner back-end service management system: A service management system collects data from various devices and networks, correlates the collected data against intelligent Intellectual Capital rules and important databases (for example, Manufacturing database or Contact Management database), analyzes the results, and produces actionable and trending reports that examine the network and predict the performance.

- Two-way connectivity: Connectivity allows the front-end system (that is, smart agents and collection systems) to send data securely to the supplier or partner service management systems. It also allows the service management system to access the device or network secularly to take action when required.
- Secure entitlement and data-transfer capability to register and entitle customer networks and communicate securely (via encryption and security keys) with service providers or network vendors: Such capability is typically deployed on the collector and back-end systems.

A Smart Service provides a proactive intelligence-based solution addressing the installed-based lifecycle and *Fault, Configuration, Accounting, Performance, and Security* (FCAPS) management with the unique benefit of correlating data with the supplier's Intellectual Capital and recognized best practices. Using smart agents, Smart Services collects basic inventory information from the network in order to establish Install Base context.

Conclusions

The implications of the IoT on today's Internet are vast. With such a large number of devices and highly constrained network environments, provisioning and management of the IoT needs to be a part of the architecture. It is both unwise and impractical to provision each active device in the network manually throughout its lifecycle. Earlier technologies, including IP phones, wireless access points, or service provider *Customer Premises Equipment* (CPE), have demonstrated that provisioning can be carried out securely over the network.

The IoT encompasses heterogeneous types of devices that can be on public or private IP networks: from low-powered, low-cost sensors, to fully functioning multipurpose computers with commercial operating systems. For this reason, there can be no "one-size-fits-all" approach to IoT security. What is required is a series of architectural approaches that are dictated by specific IoT use cases. In certain industry solutions, most notably healthcare, security is not just important; information privacy is specifically mandated in many countries.

The challenges of designing, deploying, and supporting billions of IP-enabled endpoints, each producing data that needs to be analyzed and acted on, present exciting opportunities for the next generation of the Internet.

References

[1] Dave Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," April 2011, **http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf**

[2] "ARM targets Internet of Things with new low-power chip," Institute of Nanotechnology,
http://www.instituteofnanotechnology.co.uk/arm-targets-internet-of-things-with-new-low-power-chip/

[3] **http://share.cisco.com/internet-of-things.html**

[4] Tim Polk and Sean Turner, "Security Challenges for The Internet of Things," IETF Security Area Directors, Feb. 14, 2011,
http://www.iab.org/wp-content/IAB-uploads/2011/03/Turner.pdf

[5] Guillaume Piolle and Yves Demazeau, "Une architecture pour la protection étendue des données personelles," 2010,
http://guillaume.piolle.fr/doc/piolle10b.pdf

[6] "Trusted Platform Module," ISO/IEC 11889,
http://www.iso.org/iso/catalogue_detail.htm?csnumber=50970

[7] Jan Camenisch, "Direct Anonymous Attestation: Achieving Privacy in Remote Authentication," June 15, 2004.
http://www.zurich.ibm.com/security/daa/daa-slides-ZISC.pdf

[8] Delay Tolerant Networking Research Group:
http://www.dtnrg.org/wiki

DAVID LAKE, B.Sc., is a Consulting Engineer in the Research and Advanced Development Group at Cisco. He has more than 20 years of network design and deployment experience, ranging from X.25 and SNA, through the era of multiprotocol routing to IP, covering a wide range of networking technologies. He has extensive experience in transporting rich-media technologies across complex enterprise and service provider networks. David has worked as customer, network integrator, and manufacturer, and he understands the unique positioning of each of these areas in the IT industry. E-mail: **dlake@cisco.com**

AMMAR RAYES is a Distinguished Service Engineer at Cisco Systems focusing on Smart Service Technology Strategy. He has authored and co-authored over a hundred papers, patents and books on advances intelecommunications-related technologies. He is the President of the *International Society of Service Innovation Professionals* **www.issip.org**, an Associate Editor of ACM *Transactions on Internet Technology* and Editor-in-Chief of *Advances of Internet of Things Journal*. Dr. Rayes received his BS and MS Degrees in EE from the University of Illinois at Urbana and his Ph.D. degree in EE from Washington University in St. Louis, Missouri.
E-mail: **rayes@cisco.com**

MONIQUE MORROW is a Distinguished Engineer in the Research and Advanced Development Group at Cisco. She has over 20 years experience in IP internetworking that includes design, implementation of complex customer projects and service development for service providers. She has presented in various conferences on the topic of RFID, Grid Networking and Cloud Computing; and, she co-authored several books and publications. She is currently focused on the Internet of Things/ Machine-to-Machine Communications in eHealth and security and is active in various SDOS and forums such the IETF, ITU-T and the FTTH Council Asia-Pacific, holding leadership positions in these organizations. Monique is a Senior Member of the IEEE the and a Life Member of the ACM. Monique has a Masters of Science Degree in Telecommunications Management and an MBA.
E-mail: **mmorrow@cisco.com**

The Demise of Web 2.0 and Why You Should Care

by David Strom

The term Web 2.0 has been around for about a decade[1], but we are finally seeing its disuse. No, the web itself is not going away, but the notion that an interactive layer of applications, protocols, programming languages, and tools has become subsumed into a new kind of web—one where everything is a *service,* mobile browsing is more important, and social networking has helped discover and promote new content. As a result, we do not really need the term anymore, because it is so much of what the web has become.

Think of this concept as going beyond the 2.0 label of the web: now we have a richer world of interactions that is just the beginning of how we use that tired old TCP port 80. All these developments mean that the readers of *The Internet Protocol Journal* are well poised to help others take advantage of this new complex web environment, because it has become the norm rather than some fancy address in the better part of town. Understanding its new structure and purpose is critical to building the next generation of websites and interactive applications.

Back in the early days of the web in the mid-1990s, it was largely static content that a browser would access from a web server. The notion of having dynamic pages that would automatically update from a database server was exciting and difficult to accomplish without a lot of programming help.

But then came Web 2.0, where the interactive web was born. We had blogging tools such as Google's *Blogger* and Automattic's *Wordpress,* and anyone could create a website that could be easily changed and instantly updated. Web and database servers became better connected, and new protocols were invented to better marry the two.

Everything as a Service

The past few years have seen the rise of *Software as a Service, Infrastructure as a Service,* and even *Platforms as a Service.*[2] The coming of *Cloud Computing* has meant that just about anything can be virtualized and moved into a far-away data center, where it can be managed and replicated easily, obviating the need for any physical infrastructure in the traditional enterprise data center.

Why is this change relevant for the modern web era? Four reasons:

- The web browser is still used as the main remote-access tool to configure and manage a wide variety of applications, network equipment, and servers, including all kinds of cloud-based infrastructures.

- Most of these "as-a-service" entities still run over ports 80 and 443 and piggyback on top of web protocols, for better or worse. We have gotten used to having these ports carry all sorts of traffic that has nothing to do with ordinary web browsing, and we have to do a better job of sorting out the ways apps use the traditional web ports too.
- We do not need to buy any software or install it on our own desktops; everything is available in the cloud at a moment's notice. What is more, we have gotten used to having the web as the go-to place to get new tools, software drivers, and programs. Software repositories such as *GitHub* and open source projects such as *Apache* have blossomed into places that corporate developers use daily for building their own apps. And why not? They have large support communities and hundreds of projects that are as well tended as something out of Oracle or Microsoft (and some would argue better, too).
- The days of a simple web server serving up pages is ever more complex, with typical commercial websites having ad servers, built-in analytics to track page views and visitors, discussion forums to moderate comments, connections to share the post on *Twitter* and *Facebook* (more on these in a moment), and videos embedded in various ways. All of these websites require coordinated applications and add-ons to the basic web server that require various cloud services. For example, the sites that I run for *ReadWriteWeb* use *Moveable Type* for our content, *Google Analytics, Disqus* discussions, interactive polls from `PollDaddy.com`, and custom-built advertising servers, just to name a few of the numerous add-ons. The ever increasing numbers of add-ons means maintaining this system is not easy, and it requires a lot of detailed adjustments on a too-frequent basis.

The Rise of Mobile Browsers

According to the research firm NetApplications[3], the share of web browsing originating from mobile devices has more than doubled in the past year. Although desktops still account for more than 90 percent of the data accessed from browsers, mobile devices are consuming the web at an increasing rate.

Part of this trend is that we are using more devices and they have become more capable. Android-based phones constitute the largest market share, and they have the fastest-growing consumer mobile phone adoption rate.[4] Certainly, more and more of us are browsing more webpages from mobile devices these days.

Another part of the trend of increased roaming on mobile devices is that more people are creating and using more mobile apps, too. Hundreds of new mobile apps with a wide variety of content are created every day. Professors at major universities teach computer science students how to code mobile apps, and you can even take online courses on *Java* programming.

But mobile browsing poses a conundrum for web designers. One school of thought is to build custom tablet applications for your website, to show off the features of the tablet interface and to make it easier for tablet users to interact with your content. The U.K. *Guardian,* for example, is leading the way in this area.[5]

Another school of thought is to improve the mobile experience, by either building a separate site that is optimized for smaller screens and lower bandwidth connections or allowing the site to work automatically under the constraints of the mobile browser itself.[6]

One real challenge for the mobile web browsing experience is the role of Adobe Flash and the newest of the *Hypertext Markup Language* (HTML) standards, HTMLv5. Apple decided when it released its first iPads to not support Flash, and since then there has been additional effort and movement to migrate many Flash-based sites, such as **YouTube.com**, toward HTMLv5, which is supported by Apple's tablets and can be more efficient for lower-bandwidth connections. Although this topic could easily be the subject of an entire article for this journal, our point in mentioning it here is that displaying video and similar content is still a problem for the web, even today.

Our mobile traffic at *ReadWriteWeb* has increased tremendously in the past year, and I suspect our site is typical of other sites. But this increase in traffic presents challenges for content creators: is it better to sell ad units around the content, even ads that have sub-par browsing experiences on mobile devices? Or code up your own iPad app (or use Verve's tools [**http://www.vervewireless.com/**] or something equivalent)? Certainly the level of engagement with the custom mobile app is greater, but it amazes me that sites with just static pages still are not optimized for mobile browsers yet, with large image downloads or multiple included links, for example.

Let's consider the site **Remodelista.com** as a case study of how to properly optimize a site for mobile browsing. The owners have implemented tricks to adjust its layout for different screen sizes. As you make your browsing window smaller (or as you run it on a mobile device with a small screen), the integrity of the site content remains intact, meaning that font sizes change and ad blocks appear on wider, higher-resolution screens and disappear on smaller ones, but the overall content stream remains the same, no matter what device is used to view it. This consistency is achieved by adding a lot of special coding to the webpages, as the following snippet shows:

```
<!--[if IEMobile 7]> <html class="no-js iem7 oldie" itemscope itemtype="http://schema org/"><![endif]-->
<!--[if lt IE 7]> <html lang="en" class="no-js ie6 oldie" xmlns="http://www.w3.org/1999/xhtml"
xmlns:nectar="http://saymedia.com/2011/swml" itemscope itemtype="http://schema.org/"><![endif]-->
<!--[if (IE 7)&!(IEMobile)]> <html lang="en" class="no-js ie7 oldie" xmlns="http://www.w3.org/1999/xhtml"
xmlns:nectar="http://saymedia.com/2011/swml" itemscope itemtype="http://schema.org/"><![endif]-->
<!--[if (IE 8)&!(IEMobile)]> <html lang="en" class="no-js ie8 oldie" xmlns="http://www.w3.org/1999/xhtml"
xmlns:nectar="http://saymedia.com/2011/swml" itemscope itemtype="http://schema.org/"><![endif]-->
<!--[if gt IE 8]> <html class="no-js" lang="en" itemscope itemtype="http://schema.org/"><![endif]-->
<!--[if (gte IE 9)|(gt IEMobile 7)]> <html class="no-js" lang="en" itemscope itemtype="http://schema.org/">
<![endif]-->
```

The Social Web Is Now Everywhere

It used to be the odd person in your professional circle who did not have or use an Internet e-mail account. Now the odd person is the one who does not have an account on Facebook or some other social networking site. What began in a Harvard dorm room in this decade has turned into a juggernaut of more than a billion users—and it is growing rapidly.

But the social web is more than a bunch of college kids swapping photos of their party pictures. A recent study from the University of Massachusetts at Dartmouth[7] shows that nearly 75 percent of the Inc. 500 (the fastest-growing 500 American private companies) are using *Facebook* or *LinkedIn,* a level that is about twice the percentage that are using corporate blogs. "Ninety percent of responding executives report that social media tools are important for brand awareness and company reputation. Eighty-eight percent see these tools as important for generating web traffic while 81% find them important for lead generation. Seventy-three percent say that social media tools are important for customer support programs." Clearly, these tools have become the accepted corporate intranet, the mainstream mechanism for communications among distributed work teams, and the way that many of us share events in our professional lives as well.

The social web means more than a "Like" button on a particular page of content; it is a way to curate and disseminate that content quickly and easily. It has replaced the Usenet *news groups* that many of us remember with a certain fondness for their arcane and complex structure. Or maybe that is just nostalgia talking.

In the presocial web past, even in the days when Web 2.0 was the rage, sharing and curation was not easy. If you wanted to share something you found online, more than likely you would e-mail your colleagues a URL. Now you can *Tweet,* post on *Facebook* and *Google+,* add an update to your *LinkedIn* account, put up a page on your corporate **`Yammer.com`** or **`tibbr.com`** server, or use one of dozens more services that will stream your likes and notable sites to the world at large. Or you likely have to do all of these tasks.

Back in the days of yore (say 2000), when I wrote a freelance article, it was sufficient to post a link to the story on my own personal website, in addition to perhaps sending an e-mail message or two to the people I thought might be interested in reading the content. Those days seem so quaint. Today, the process of writing the article is actually just the beginning, not the end. When the article appears online, a whole series of promotional activities must take place, including monitoring online discussions and adding my own comments, posting on the various social media sites, and re-Tweeting a link to my article several times over the next several days—all to ensure generation of lots of traffic.

There are even services such as **Ping.fm** and **Graspr.com** that can coordinate batch updates to numerous services, so that at the push of a button all of your social media will get your news at once. Or services such as **Nimble.com** that attempt to coordinate your entire social graph (as it is called) of friends and admirers so you can track what is going out across all your various networks.

Where We Go from Here

I have just tried to touch on a few topics to show that the days of the simple static web are "so over," as Generation Y says. Clearly, we have a long and rich future ahead of us for more interesting web applications.

References

[1] **http://en.wikipedia.org/wiki/Web_2.0**

[2] See "Alphabet Soup in the Cloud": **http://www.readwriteweb.com/cloud/2011/10/alphabet-soup-in-the-cloud-und.php**

[3] NetApplications research cited in this September 2011 article in *Computerworld:* **http://www.computerworld.com/s/article/9219696/Apple_rules_phone_tablet_browsing_market**

[4] Nielsen's statistics are typical: **http://blog.nielsen.com/nielsenwire/?p=29786**

[5] See **http://m.guardian.co.uk/**, but you really need to view it on an iPad or other tablet device to understand what they are trying to do with their content. See also: **http://www.readwriteweb.com/archives/the_guardian_ipad_edition_hits_ios_5_newsstands.php**

[6] See Thomas Husson's May 2011 Forrester Research blog post here: **http://blogs.forrester.com/thomas_husson/11-05-03-why_the_web_versus_application_debate_is_irrelevant**

Also see my own January 2012 article in ReadWriteWeb here: **http://www.readwriteweb.com/hack/2012/01/do-you-really-need-your-own-mo.php**

[7] See their January 2012 study here: **http://www.umassd.edu/cmr/studiesandresearch/2011inc500socialmediaupdate/**

DAVID STROM has created dozens of editorial-rich websites for publications such as *ReadWriteWeb, Tom'sHardware.com, eeTimes,* and others, as well as written thousands of articles for numerous IT magazines. He was the founding editor-in-chief of *Network Computing* magazine and author of two books on computer networking. He lives in St. Louis, Mo. and can be found at **strominator.com**, on Twitter **@dstrom**, and **david@strom.com** for those that still prefer e-mail.

Binary Floor Control Protocol

by Pat Jensen, Cisco Systems

Over the last decade, communication technologies have evolved to encompass new modalities of collaboration across IP networks—from instant messaging on a personal computer, to being able to make *Voice-over-IP* (VoIP) calls and also now including the growing adoption of *High-Definition* (HD) videoconferencing.

Operating systems, device types, and physical locations now are less affected as continued growth in networking has evolved to promote high bandwidth across wireless and wired networks. An example is the emergence of growing network-access technologies such as *Multiprotocol Label Switching* (MPLS), *Very-High-Speed Digital Subscriber Line 2* (VDSL2), *Long Term Evolution* (LTE), and *Data over Cable Service Interface Specification* (DOCSIS). With both availability of bandwidth and broadband user penetration increasing, the user's expectation of delivering immersive collaboration now becomes more apparent.

This evolution includes modern use cases accelerating the adoption of videoconferencing, such as enabling telemedicine for remote surgeries and diagnostic procedures as well as distance learning applications being used to connect educators with students across the globe.

This article introduces the *Binary Floor Control Protocol* (BFCP) as a standard for managing floor control during collaboration sessions across dedicated video endpoints, mobile devices, and personal computers running collaboration software. These capabilities can be delivered using an enabled *Session Initiation Protocol* (SIP) standards-based endpoint or as a software implementation in a collaboration application stack.

History

BFCP is a deliverable developed as part of the *Internet Engineering Task Force* (IETF) XCON Centralized Conferencing working group. The IETF XCON working group was formed to focus on delivering a standards-based approach to managing IP conferencing while promoting broad interoperability between software and equipment vendors.[1]

This mandate includes defining the objects, mechanisms, and provisions to assist in scheduling conferencing resources. These resources could be consumed as a conference enabled in a web browser, via an audio conference call or during a videoconference.

As defined, privacy, security, and authorization are considered integral in protecting the ability to join, participate in, and manage each conference session. The IETF XCON working group's initial focus was on unicast media conferences.

The IETF XCON working group was proposed in August 2003, with work starting early in October of that year.[2] Early requirements for BFCP were defined in RFC 4376, which describes important concepts, including a model for floor control and how it should be integrated in a conferencing platform.[3] Other important aspects such as security, including using authentication and encryption to provide protection against man-in-the-middle attacks, were also outlined.

In November 2006, Gonzalo Camarillo, Joerg Ott, and Keith Drage authored RFC 4582, which defined the Binary Floor Control Protocol.[4]

Besides BFCP, other standardization efforts around conference role and content management also were defined, including the ITU-T H.239 recommendation.[5] Unlike BFCP, H.239 applies specifically to H.323-enabled *Integrated Services Digital Network* (ISDN) and IP conferencing endpoints, whereas BFCP is designed to be agnostic of the underlying signaling protocol.

Protocol Details

The basic concept of floor control is analogous to managing a live in-person presentation, where you want to control who is presenting, manage and transition your presenters, and maintain a feedback loop. Also important is the ability to allow a presenter to show slides and share with your audience a white board or transparency projector.

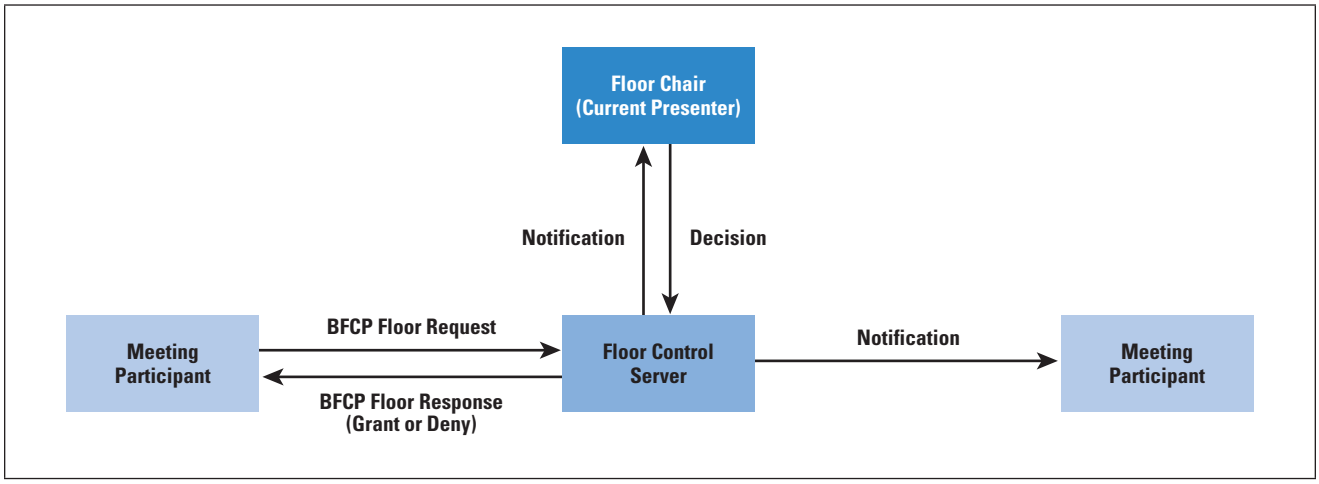
During an active collaboration session, a presenter may choose to present material to a remote user, or optionally to an audience on a call with multiple endpoints through a *Multipoint Control Unit* (MCU). This session could include many additional sources; for example, using a secondary video camera to show zoomed-in content (that is, an optical examination camera used in telemedicine) or any external video source.

This floor-control mechanism can also encompass functions available in a collaboration application stack, such as the ability to share the content of the presenter's desktop, application, or web browser.

BFCP provides the ability to manage multiple streams being presented during a collaboration session using floor control. BFCP accomplishes this management using a token-based mechanism where a single presenter can request control of the floor from the floor-control server.

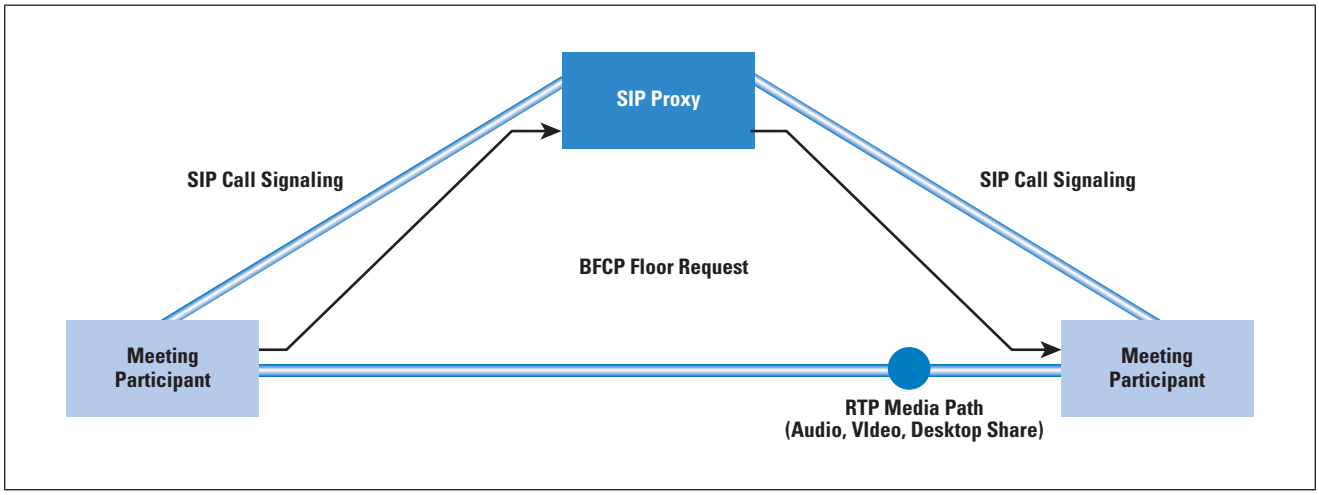
When this request is granted, the presenter holds the token and has the ability to open an additional stream to provide presentation data. Figure 1 examines this process in detail, with a meeting attendee requesting the token from the floor-control server to become an active presenter during the session.

Figure 1: BFCP Floor Request from Floor-Control Server



This same interaction can also take place during a point-to-point audio or video call with only two parties. In this case, a token can be used to signify which party will be presenting an additional stream, such as a secondary camera or application providing a desktop sharing session. Figure 2 shows an overview of this process. One of the critical differences here is that in a point-to-point call, the floor-control server capability is being provided by the user's device or application instead of using a multipoint control unit or conference server.

Figure 2: BFCP Floor Request in a Point-to-Point SIP Call



For instance, as a presenter, you can choose to present auxiliary streams via your application or endpoint and determine whether it is your primary, secondary, or tertiary stream. As a conference participant, you can also choose which stream you are currently viewing, also including the definition and quality of the secondary stream. In this case, current network conditions such as bandwidth and latency will also dictate the quality of additional streams.

BFCP is designed to be signaling protocol–agnostic, in that it is relying on the capabilities of the underlying signaling and transport protocols to set up each stream that is being managed, including whether voice, video, or content is being provided in the *Real-Time Transport Protocol* (RTP) stream.

For example, using a standards-based endpoint and *Session Initiation Protocol* (SIP), a SIP INVITE message is sent with the media capabilities line specifying the session description information about the stream. This data provides relevant information about the underlying video codec being used and the bit rate that is required to support the video and presentation streams.

In this case as multiple RTP media streams are transported across the network carrying audio and video traffic, *Call Admission Control* (CAC) and *Quality of Service* (QoS) tagging can be applied and enforced by the call-control platform, providing the ability to limit bandwidth usage and helping ensure that bandwidth is available on the network after the additional media stream is added.

Also important to note, BFCP can use *Transport Layer Security* (TLS) to provide encryption of floor information pertaining to each resource that is being controlled as well as the participants using and viewing them. BFCP provides the ability to support anonymous users as well for sessions where you may have a large audience or where anonymity is desired. An example of where this feature could be used is hosting a large web conferencing event where you have external attendees who may be outside of your organization.

One use case for BFCP includes the ability to focus on the presenter while the presenter is sharing a desktop application. With the ability to control the presenter's media stream, this feature adds additional immersion in a collaboration session, allowing you to both identify the presenter's visual cues and posture as well as focus on relevant content the presenter supplies.

Summary

The Binary Floor Control Protocol plays a very important role in helping manage diverse types of content being shared across multiple parties in a conference session. Today's modern implementations of BFCP span web conferencing applications as well as video and audio conferencing solutions across a wide array of vendors.

While these vendors are focused on delivering these capabilities across screen-led PC-centric types of devices, because of its inherent transport-agnostic capabilities, it is likely we will see BFCP being used to enable new modalities of content sharing across collaboration applications in the future.

Industry efforts are focusing on promoting collaboration applications across new arrays of devices, including using touchscreen technology on handheld computers and stationary LCD televisions to manipulate and visualize data in new ways.

Concepts such as manipulating session content using cognitive mapping as an evolution of electronic whiteboarding and transitioning an active conference from a tablet device to another type of room-based video-enabled endpoint during a collaboration session are two powerful examples of ways BFCP could be used in the future. On the horizon, touchscreen-enabled tablet and smartphone devices and HTML5-enabled web browsers also provide yet another avenue to enable rich standards-based multimedia conferencing with advanced content management.

Disclaimer

The views of this article do not necessarily represent the views or positions of Cisco Systems.

For Further Reading

[1] `http://datatracker.ietf.org/wg/xcon/charter/`

[2] `http://datatracker.ietf.org/wg/xcon/history/`

[3] Petri Koskelainen, Joerg Ott, Henning Schulzrinne, and Xiaotao Wu, "Requirements for Floor Control Protocols," RFC 4376, February 2006.

[4] Gonzalo Camarillo, Joerg Ott, and Keith Drage, "The Binary Floor Control Protocol," RFC 4582, November 2006.

[5] "Role management and additional media channels for H.300-series terminals," International Telecommunication Union Standard H.239, September 2005.

PAT JENSEN is a member of the Unified Communications consulting systems engineering team at Cisco Systems. Since 2010, he has designed collaboration architectures for Cisco's customers across the western United States. Prior to joining Cisco Systems, he served as an IP telephony design engineer designing and implementing unified communications and telepresence technologies at AT&T and SBC DataComm. E-mail, XMPP, SIP: `patjense@cisco.com`

Fragments

© Stonehouse Photography/Internet Society

Pierre Ouedraogo Receives 2012 Jonathan B. Postel Service Award

The Internet Society recently announced that its prestigious *Jonathan B. Postel Service Award* was presented to Pierre Ouedraogo for his exceptional contributions to the growth and vitality of the Internet in Africa. The international award committee, comprised of former Jonathan B. Postel award winners, noted that Mr. Ouedraogo played a significant role in the growth of the Internet in Africa and demonstrated an extraordinary commitment to training young engineers and participating in regional Internet organizations.

Mr. Ouedraogo is the Director of Digital Francophonie at *Organisation Internationale de la Francophonie* (OIF) based in Paris, France. Over the years, he has established networks of IT experts to coordinate African efforts to develop IT and use it as a tool for development. Mr. Ouedraogo initiated many IT technical workshops in Africa and is a founding member of numerous African regional organizations, including AfriNIC (the African Internet Registry for IP addresses); AfTLD *(African Internet Top Level Domain Names Association);* AFNOG *(African Network Operators Group);* AfCERT *(African CERT network)*, and AfrICANN *(African network of participants to the ICANN process)*.

"Pierre Ouedraogo is a highly-regarded technical leader in Africa, and he has been instrumental in bringing the Internet to Burkina Faso as well as other French-speaking African countries," said Lynn St. Amour, President and Chief Executive Officer of the Internet Society."

"His commitment to the expansion of the Internet and encouragement of young engineers to help them build their skills through training workshops has had a profound impact on the growth of the Internet across Africa."

The Postel Award was established by the Internet Society to honour individuals or organisations that, like Jon Postel, have made outstanding contributions in service to the data communications community. The committee places particular emphasis on candidates who have supported and enabled others in addition to their own specific actions. The award is focused on sustained and substantial technical contributions, service to the community, and leadership.

For more information about the Internet Society and the Postel award, see: **http://www.internetsociety.org/**

Atsushi Seike (L) with Vint Cerf and Jun Murai.

Vint Cerf Awarded Honorary Doctorate by Keio University

Keio University in Tokyo recently awarded Dr. Vinton Gray Cerf an honorary doctorate in Media and Governance for his work in the creation and governance of our modern Internet over the last forty years. On the recommendation of Professor Jun Murai, dean of the Faculty of Environment and Information Studies, Keio University president Atsushi Seike presented Dr. Cerf with the degree. The ceremony was held in the Enzetsu-kan, the historic public speaking hall on Keio's Mita Campus in Tokyo, and streamed live via the Internet to viewers around the world.

Professor Murai's recommendation for the degree, read during the ceremony, said that not only is Dr. Cerf the founding father of internetworking technology, "he is the global leader in many ways of the largest innovation for the 21st century, the Internet itself, which has become the core of today's information-based society." In addition to his work on TCP/IP with Robert Khan, Dr. Cerf's work in establishing the Internet Society and his stewardship of ICANN as its chairman were highlighted. Also mentioned was his role in *Delay/Disruption-Tolerant Networking* (DTN) and the first experiments connecting a space probe twenty million miles away using Internet protocols.

In his remarks, President Seike mentioned Dr. Cerf's forty-year commitment to advancing the role of networks in creating our global society, from the earliest days of the ARPANET through today's Internet. "[Dr. Cerf] understood quickly and clearly the international nature of the Internet and its potential for having a positive impact on the lives of not just the technical elite, but for all of the people of the world, as a tool for education, commerce, and the advance of democracy," he noted. Professor Seike compared Dr. Cerf's role in using technology to make the world a better place to the efforts of Yukichi Fukuzawa, the founder of Keio University, who in the mid-19th century was instrumental in bringing knowledge to Japan from the outside world, not as an academic exercise but in order to improve society.

Following the ceremony, Dr. Cerf gave an invited technical talk titled "Re-Inventing the Internet." He discussed the potential of DTN and *Mobile Ad Hoc Networks* as tools for disaster recovery. He presented his view of urgent technical problems, including the need for strong authentication and digital forensics. He also outlined society's need for preserving data, the programs that create and manipulate that data, and even the systems that are used to run those programs. Without such an effort, we will fail to preserve our own technical and cultural history for the thousands of years we have come to expect, he noted.

Dr. Cerf left behind the inscription, "I cannot imagine a greater honor than to be brought into this august and highly regarded university where contrary thinking is rewarded! I am most grateful to my good friend, Jun Murai, for his decades long commitment to the Internet."

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com