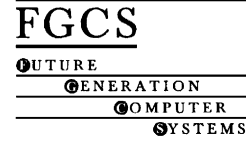




ELSEVIER

Future Generation Computer Systems 19 (2003) 191–197



www.elsevier.com/locate/future

Future applications and middleware, and their impact on the infrastructure

Brian E. Carpenter

IBM Zurich Research Laboratory, Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland

Abstract

The Internet today has progressed from being “just a network” through phases as a human communications mechanism and an unrivalled information system to the mere beginning of being a true services network. On the way, it has encountered a series of technical and external challenges that have created some barriers to further development. For the Internet to release its known potential and develop as a resource-sharing computing services platform, these barriers must be torn down.

© 2002 Published by Elsevier Science B.V.

Keywords: Grid computing; Web services; Internet transparency; IPv6; Open Grid Services Architecture

1. The Internet today

The Internet was initiated as a genuine peer-to-peer network, with a single transparent addressing scheme, in which any host computer could initiate an application session with any other without prior formality. Indeed, this concept arose precisely to overcome the barriers to transparent communication observed in the earliest computer networks [5,15]. TCP/IP and all its associated protocols grew up in this context. The Internet’s first wave of success outside a narrow community was when this transparency made electronic mail and news groups available to hundreds of thousands and then millions of users. The second wave was when the Web built on this same transparency. As Berners-Lee wrote “There is a freedom about the Internet: as long as we accept the rules of sending packets around, we can send packets containing anything to anywhere” [2]. Thus the Internet became an information web—the normal mode is for clients (users) to suck down bits from a server, like young birds in a nest

suck down food from their parents. Indeed some service providers have taken to referring to their users as “consumers” and to what they consume as “content”. From the viewpoint of a network conceived as a transparent connection between peers, this is a strange choice of terminology. Using the web to achieve positive results (buy, sell, play, work)—in fact, to *do stuff*—is still somewhat the exception. Using the web on the move is still the exception. Most seriously, fully trusting the web is still the exception.

It seems that a third wave, known as Web Services, is just starting. In this we return to the multi-party model rather than a simple client/server model, and one sees signs of organised, rather than random, access to computing resources across the open network. In particular, services are described in a standard format (Web Services Description Language, WSDL) and service requestors are bound to service providers using a standard protocol (Simple Object Access Protocol, SOAP) [8]. Another view of this is as a form of distributed computing which, unlike previous implementations, claims to operate just as well across the open Internet as it does across a private network.

E-mail address: brian@hursley.ibm.com (B.E. Carpenter).

2. Challenges and barriers

The growth and success of the Internet, especially over the last 10 years, have led to challenges of several kinds: those directly due to success, rapid scaling, and apparently, thoughtlessness. The responses to several of these challenges have unfortunately led to the erection of artificial barriers to further growth and development.

2.1. Challenges of success

A first challenge of success, beyond straightforward commercial activity, was the arrival of Internet gold diggers starting around 1995. This has led to many unintended or unexpected uses of the technology, some of doubtful legitimacy: use of the Domain Name System (DNS) as a product directory, domain name squatting and speculation, and unsolicited commercial email (spam) are only the most blatant examples. In turn, spam, viruses, and other security threats have led to extensive use of email filtering and reluctance to allow corporate users direct Internet access.

The Internet's explosion as a commercial medium led to the creation of the Internet Corporation for Assigned Names and Numbers (ICANN) to take over a few administrative tasks:

- administer protocol parameters;
- coordinate allocation of address blocks to the regional registries;
- coordinate allocation of TLD names to TLD registries;
- coordinate root server operations.

Unfortunately these matters, formerly handled part-time by one person, have come to consume 30 staff while distracting government officials around the world, a response that is quite disproportionate to the objective scale of the tasks involved. This unnecessary complexity makes innovation much harder.

ICANN is not a regulator, but national and international telecommunication regulators now find the Internet very tempting, even if hard to get hold of. However, they are persistent, and their instinct, when in doubt, is to make a regulation. Politicians also find the success of the Internet very tempting, and sometimes threatening (it allows free speech, displays unwelcome material, and is largely tax free). Politicians

are even less predictable than engineers and regulators, and when in doubt, they pass a law, regardless of engineering reality. Although in general the Internet is flexible enough to “route around damage” caused by regulators and politicians, this is not guaranteed, and in some cases straightforward use or discussion of technology has even been criminalised.

The success of the Internet has led to notorious excesses of “hype”, i.e. exaggerated claims made about relatively mundane inventions and business models, apparently in the belief that making such claims is a requirement for success. Yet unjustified hype always leads to disappointment. A closely related non-technical challenge of success is the risk of Internet people suffering from hubris, exaggerated pride or self-confidence. Those who created the Internet have reason to be proud, but should not lose sight of the real problems and should not ignore the impact of success on the original design principles of the network. We cannot expect things to continue exactly as they were. In some cases, technical principles that have served us well for many years will need to be revised.

The main technical challenge of success is internationalisation, i.e. making the Internet useable in any language and character set. The plan was to rely on ISO 10646/Unicode [1]. However, some uses of text are hidden entirely in protocol elements and need only to be read by machines (protocol engines), while other uses are intended entirely for human consumption (presentation). Many uses lie between these two extremes, which lead to conflicting implementation requirements:

- humans can handle ambiguity, while protocol engines cannot;
- humans care about cultural aspects, while protocol engines are indifferent to them.

Thus, matching and folding requirements are different in the two cases and the choice of a single standard such as Unicode is inevitably a compromise.

2.2. Challenges of scaling

The need to scale up the IP address space has been a known problem since 1992, and the solution, IP version 6, was chosen in 1994, with products existing since 1997, and the basic standards stable since about 1999. Nevertheless, operational deployment of IPv6

has been surprisingly slow to start. We will return to this question below, but we can mention several reasons:

- Operational costs of conversion; operational conservatism.
- Lack of strategic incentives in a fundamentally short-term industry.
- The cost of *not* converting is spread too thinly and not understood by decision makers.

Another problem known since 1992, but far harder in principle than scaling the address space, is the need to scale up the Internet's backbone routing system [10]. The current interdomain routing protocol, BGP4, will not be adequate much longer. This difficult topic is still a research problem.

Scaling leads to congestion and unreliability, leading to strong calls for "Quality of Service" (QOS). The technical community has specified session-oriented (integrated services) and stateless (differentiated services) models for Internet QOS, and both technologies are available in widely used products, but neither has swept the world. Most users still receive exclusively "best effort" service, i.e. no defined service level whatever. As with IPv6, the question arises of how we can get a new technology into the current practice of every network operator [11].

A final challenge of scaling is the sheer number of standard organisations affecting the Internet today. Apart from well-known bodies such as the IETF (Internet Engineering Task Force), the W3C (World Wide Web Consortium), the GGF (Global Grid Forum), and the ITU (International Telecommunications Union), one can find almost 50 other relevant standard-related bodies. An implementer of a complete software suite must consider all of these.

2.3. *Challenges of thoughtlessness*

The most worrying class of challenges are those that, viewed from a long-term perspective, appear to be simply the result of thoughtlessness about the strategic impact of a tactical decision. Network Address Translation (NAT) is the first example. It was such a tempting quick fix for the impending shortage of IPv4 addresses and the undoubted difficulty of renumbering a site when its Internet Service Provider was changed. NAT could even be marketed as a

security system, by pre-configuring NAT boxes to disallow almost all traffic by default. Yet NAT breaks many non-client-server applications as well as preventing network level security, while causing innumerable operational complexities [9]. In some cases, such as India or China, the shortage of IPv4 addresses would, if handled thoughtlessly, lead to the deployment of not one but several layers of NATs. The operational costs of this are enormous, since NAT-induced failures of connections or application sessions are frequent and very hard to diagnose, even without considering the opportunity cost of being unable to deploy the innovative applications that NAT prevents.

Systematic "layer violations" marketed as performance enhancements also pose serious challenges to smooth operation. Boxes that interfere with traffic, by peeking into application layer headers, and then sending the packet to a different server, proxying a request without being asked, or worst rewriting applications data, may cause unpredictable, inexplicable glitches and failures. Boxes performing intermediate functions between source and destination should do so by using protocols designed for this purpose, not by abusing protocols designed to work end-to-end [4].

The DNS has also been thoughtlessly abused and functionally overloaded. The DNS was narrowly designed, as a replacement for simple tables of names and addresses, with distributed update and distributed lookup. It was also designed to be extensible, but it was not designed as a directory. In recent years, despite being insecure, it has been abused as a directory and indeed built into many if not all recent proposals for distributed computing as a fundamental namespace [13].

The HyperText Transport Protocol (HTTP) was also narrowly designed, to carry HyperText Markup Language (HTML) requests and responses between a Web client and a Web server. It was also designed to be easy to use, but it was not designed as a general-purpose transport protocol (indeed, it can only run by relying on a transport protocol). Yet because HTTP is easy, it has been widely abused as a transport protocol and firewall penetration technique, two purposes for which it is hardly suitable [14].

Related to this is the thoughtless "crunchy outside, soft inside" view of security engendered by the firewall model. The Internet was originally designed (in the 1970s) with a universal trust model; all users were

trusted, or mistrusted, equally. This model could not survive the combination of commercial and public usage of the network, so the firewall model was quickly invented in the late 1980s. Corporate firewalls attempt to divide the world into a trusted inside or “intranet” and a mistrusted outside (often with a half trusted “demilitarised zone” between the two). While providing a measure of gross security, this approach masks many real risks (e.g. vulnerability to dishonest employees, penetration by “safe” protocols such as HTTP used as a Trojan Horse). Most seriously, it fails to meet, and even blocks, new requirements: compartmentalised and dynamic end-to-end trust relationships across administrative boundaries [12].

The mythical Public Key Infrastructure (PKI) is the final challenge we consider. We thoughtlessly imagined that by creating technology capable of supporting a universal PKI, such an infrastructure would come into existence. This is not the case, and we have a big challenge in actually deploying public key based solutions except within closed worlds.

2.4. Barriers to progress

Unfortunately, some of these challenges and the Internet industry’s responses have created artificial barriers to progress beyond the “information web” stage:

- Inevitably, we must overcome the scaling challenges.
- Shortage of addresses, NATs, firewalls, and layer-violation boxes inhibit deployment of “any-to-any” distributed computing solutions (as opposed to simple client/server solutions built mainly around HTTP).
- The firewall/intranet model and the PKI problem inhibit deployment of any-to-any trust and fine-grain security.
- The current level of dependency on HTTP and DNS will eventually be found unacceptable in terms of reliability.

We cannot deal with these barriers by simply denying their existence and insisting that the Internet cannot be changed. There will always be intermediate boxes, and the original vision of a fully transparent Internet is dead [3]. So an architecturally sound solution is required, in which the barriers are either removed, or selectively opened.

3. The Internet as a computing services platform: tear down the barriers

The main reason that the barriers must be torn down is the continued need for change and growth: the marketplace has new requirements, technology and the appetite for technology feed on each other, and both benefit from the Internet culture of open standards.

The most important marketplace requirements concern the need for more efficient use of IT resources for computation, storage, and transactions. Industrial and commercial users, in particular, place even more importance on resource management (including security) and on the total cost of ownership. They are chasing out hidden costs and wasted resources, and they expect industrial strength infrastructure, running round the clock, which is fully secure, robust under attack, and can quickly recover from disaster. Furthermore, they now expect solutions that are integrated, but flexible. In other words, applications must run in a uniform environment, with uniform data access, whether they are distributed, centralised, out-sourced, or a mixture. The days of application “silos” in which a given application can run only on a given department’s computer are numbered. However they are spread across a network, computing resources must become virtualised and be managed in a uniform way. Middleware is required to mediate between the applications that see this virtual environment, the actual physical resources (servers, storage systems, networks), and the operational staff managing them.

At the same time, and despite current macro-economic difficulties, growth refuses to slow down. Network capacity is no longer a scarce commodity, server and storage costs decline, low cost wireless devices are imminent, and emerging economies are showing a strong interest in the Internet and in distributed IT. Given current trends, it seems reasonable to target a 10 billion node Internet within a few decades. Our technology must certainly allow for this.

In this context, two important computing trends are converging. Grid computing today [6] is not the same as Web Services, but it was driven in the scientific world by the same forces that drove Web Services for dynamic e-business:

- evolving costs of network capacity, servers, and storage;
- systems convergence on Internet protocols and Unix/Linux operating systems;
- the value of resource sharing on the network;
- the need for guaranteed service levels;
- the need for any-to-any security rather than the simplistic firewall model.

Thus the idea has emerged of the Internet as a computing services platform in itself, rather than simply a data transport mechanism. An important effort has started to converge the ideas and standards behind Web Services and those behind Grid computing into the Open Grid Services Architecture (OGSA) [7]. In OGSA, Grid services will be invoked using WSDL and SOAP. The result, when complete, will be standards and middleware toolkits for building an infrastructure that is both open and secure. OGSA will allow computing resources to be efficiently managed and shared within or between organisations in a secure fashion, very often as a paid utility service rather than by purchase of equipment. A key aspect of OGSA is that an entity may perform both as a service provider and a service requestor simultaneously; thus the rigid separation of devices into clients and servers no longer exists. At the same time, OGSA assumes that users belong to virtual as well as physical organisations, and security boundaries will be created as a function of both; the traditional division of users into “inside” (trusted) and “outside” (untrusted) fails.

It is hard to imagine deploying OGSA, or any other generic any-to-any distributed computing architecture, across a 10 billion node network without removing the barriers identified earlier. By removing them:

- True end-to-end network security, and true any-to-any computing, will be possible.
 - For Web Services, Grid services (OGSA), and distributed computing in general, it will then be possible to achieve massive deployment by enabling all nodes to be service providers as well as requestors. This will open up computing as a pervasive utility service for every enterprise, small, medium, or large.
 - As a result, distributed and virtual enterprises will be possible.
- Perhaps surprisingly, mergers and acquisitions will be simplified (merging two private networks is a major cost, especially with NATs; merging IT systems is an enormous cost; merging the usage of a utility is much easier).
 - Adequate security and address space will allow pervasive networking of homes and schools.
 - It will be possible to generally deploy Voice over IP services, today seriously inhibited by NATs.
 - Similarly, it will be possible to deploy third-generation cell phones, and other wireless solutions, with a clean addressing and routing scenario for “Internet on the run”.

We saw that, apart from general scaling problems that are being tackled, most of the barriers to growth come from a single source: intermediate boxes (NATs, firewalls, and layer-violation boxes) that constrain us to use client/server solutions based on HTTP and DNS even when the problem is not client/server. Solutions are on the way.

First, eliminating NATs and providing enough address space for 10 billion nodes is straightforward: simply deploy IPv6. There is no need to try to squeeze a 10 billion node network into the four billion addresses we have today, and there is no need to force the new any-to-any network into the old client/server model intrinsic to NAT. It is better to invest in the strategic benefit of IPv6 for the next 50 years than to pay the operational costs of struggling on with IPv4.

Secondly, we must move to a security model in which direct transport level connections, secured by a well-known mechanism such as IPSEC or TLS, are allowed to traverse administrative boundaries rather than being blocked by firewalls. In this model, the trust relationships and security barriers are no longer exclusively at some “external” boundary, but are actually where they need to be—inside the servers they are designed to protect. Recently started work on OGSA security [16] is an example of the class of architecture that will be required. Apart from improved security, a side benefit of such a change is that we can cease to use HTTP as a Trojan Horse protocol for firewall penetration, and decide on merit which transport protocol to use to carry distributed computing protocols such as SOAP. SOAP itself is designed to run over a variety of transport protocols, so the Web Services architecture is ideally placed to take advantage of this

future improved security environment, and place intermediate boxes and security boundaries where they are truly needed instead of at arbitrary administrative boundaries.

Thirdly, where intermediate boxes are needed to perform important functions (such as relaying, proxying, caching, transcoding, or load balancing) or where firewall or NAT traversal is still required, properly architected solutions based on standard protocols should be used. In many cases, Web Services or OGSA may be suitable, or solutions already exist, e.g. SMTP. At the time of writing, the MIDCOM and OPES working groups in the IETF are tackling specific requirements in this area, and no doubt others will arise in future.

4. Conclusion

The Internet has progressed as far as the early deployment of Web Services, and Grid computing, with IPv4, a firewall-based security model, and some thoughtless quick solutions (NAT, HTTP as a Trojan Horse, etc.). As growth continues, distributed computing solutions such as the OGSA will transform the Internet into a computing platform, but unless the barriers are removed, OGSA and its peers will get stuck on rough edges of NAT boxes, firewalls, and layer-violation boxes.

As just described, ways of removing the barriers are emerging: IPv6, any-to-any security models, and architected approaches to intermediate boxes. The Internet community should finalise and deploy these solutions. For the future, Internet engineers should avoid responding in a thoughtless, short-term way to the next round of challenges.

Acknowledgements

Useful comments on this paper were made by François Flückiger and Philippe Janson.

References

- [1] H. Alvestrand, IETF Policy on Character Sets and Languages, RFC 2277, January 1998. <http://www.rfc-editor.org/>.
- [2] T. Berners-Lee, M. Fischetti, *Weaving the Web*, Harper-Collins, 1999, p. 208.
- [3] B. Carpenter, Internet Transparency, RFC 2775, February 2000. <http://www.rfc-editor.org/>.
- [4] B. Carpenter, S. Brim, Middleboxes: Taxonomy and Issues, RFC 3234, February 2002. <http://www.rfc-editor.org/>.
- [5] V. Cerf, The Catenet Model for Internetworking, IEN 48, Information Processing Techniques Office, Defense Advanced Research Projects Agency, July 1978.
- [6] I. Foster, C. Kesselman, S. Tuecke, The anatomy of the grid: enabling scalable virtual organizations, *Int. J. Supercomput. Appl.* 15 (3) (2001).
- [7] I. Foster, C. Kesselman, J. Nick, S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, 2002, in progress.
- [8] S. Graham, S. Simeonov, T. Boubez, G. Daniels, D. Davis, Y. Nakamura, R. Neyama, *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI*, Sams, 2001.
- [9] T. Hain, Architectural Implications of NAT, RFC 2993, November 2000. <http://www.rfc-editor.org/>.
- [10] G. Huston, Commentary on Inter-domain Routing in the Internet, RFC 3221, December 2001. <http://www.rfc-editor.org/>.
- [11] G. Huston, Next Steps for the IP QoS Architecture, RFC 2990, November 2000. <http://www.rfc-editor.org/>.
- [12] S. Ioannidis, A.D. Keromytis, S.M. Bellovin, J.M. Smith, Implementing a Distributed Firewall, in: *Proceedings of the ACM Conference on Computer and Communications Security*, Athens, Greece, November 2000.
- [13] J.C. Klensin, Role of the Domain Name System, 2002, in progress.
- [14] K. Moore, On the use of HTTP as a Substrate, RFC 3205, February 2002. <http://www.rfc-editor.org/>.
- [15] L. Pouzin, A Proposal for Interconnecting Packet Switching Networks, in *Proceedings of the Conference of EUROCOMP*, Brunel University, May 1974, pp. 1023–1036.
- [16] F. Siebenlist, V. Welch, S. Tuecke, I. Foster, N. Nagaratnam, P. Janson, J. Dayka, A. Nadalin, *OGSA Security Roadmap*, 2002, in progress.



Brian E. Carpenter is an IBM Distinguished Engineer working on Internet Standards and Technology. He is currently based at the IBM Zurich Research Laboratory. From 1999 to 2001 he was at iCAIR, the International Center for Advanced Internet Research, sponsored by IBM at Northwestern University in Evanston, IL.

Before joining IBM, he led the networking group at CERN, the European Laboratory for Particle Physics, in Geneva, Switzerland, from 1985 to 1996. This followed 10 years' experience in software for process control systems at CERN, which was interrupted by 3 years teaching undergraduate computer science at Massey University in New Zealand.

He holds a first degree in physics and a PhD in Computer Science, and is a Chartered Engineer (UK) and a Member of the IBM Academy of Technology. He is an active participant in the IETF, where he co-chairs the Differentiated Services Working Group. He

served from March 1994 to March 2002 on the Internet Architecture Board, which he chaired from July 1995 to March 2000. He was Chairman of the Board of Trustees of the Internet Society from June 2000 through June 2002.