From: *Business Strategies for the Next-Generation Network*, Nigel Seel, Auerbach Publications, 2006.

Chapter 1: The strange death of Broadband ISDN

Introduction

In the early nineties I was working for Bell-Northern Research, then Northern Telecom's R&D organisation (now Nortel), as a a carrier network architect. I recall flying from the UK into the depths of the Canadian winter to attend a Broadband ISDN conference. These days, you may be forgiven for being somewhat hazy as to what Broadband ISDN actually was, but back in the early nineties it was absolutely the next big thing. B-ISDN - 'Broadband Integrated Services Digital Network' to spell out the full acronym, was the universally regarded architecture for the future multimedia carrier network, the 'Next-Generation Network' of its time, underpinned by a packet technology called Asynchronous Transfer Mode (ATM) [1]. Today, ATM is receding into history, but at the time we all knew it was the future of communications. All traffic: voice, video and data, was to be carried in 48 byte packets, with a 5 byte header for routing and control. This 53 byte packet was called an ATM cell. The magic (and small) numbers of 48 and 53 bytes came about because the main function of ATM, in the carrier view, was to carry voice, minimising cell-fill latency.

Most data packets, by contrast, are large (e.g. 1,500 bytes) so as to minimise packet header overhead. To carry a large data packet over ATM (figure 1) it is necessary to segment it into small chunks, each of which can be carried in one ATM cell. Extra control information is required to ensure it can be properly put together again at the far end in the right order. The significant extra overhead in segmentation and reassembly, plus the overhead of all those ATM headers was called the 'ATM cell tax'. IP people in particular deeply resented ATM for this reason. However, at the time, IP people did not do voice, and so were not especially influential.

Until quite recently, voice was the overwhelmingly dominant traffic on the network, so the networks were designed around it. In a traditional telephone call, the analogue voice signal from the telephone handset is sampled at the local exchange 8,000 times per second and the audio level of each voice sample is encoded in 8 bits (giving 256 possible amplitudes) for an overall 64 kbps signal. In this pre-ATM reality, each voice sample byte is then assigned a 'time-slot' and sent across the carrier network to the far-end exchange serving the conversational partner. The network has to be timed to exquisite accuracy - so there is virtually no possibility of losing a voice sample, or of incurring differential delay of successive time slots (called jitter). The overall delay, called the latency, is also minimal, as there is no queuing of timeslots. The circuit-switched telephone network is excellent for basic, vanilla voice. However, all of these desirable properties go by the board when we try to packetise voice in ATM.

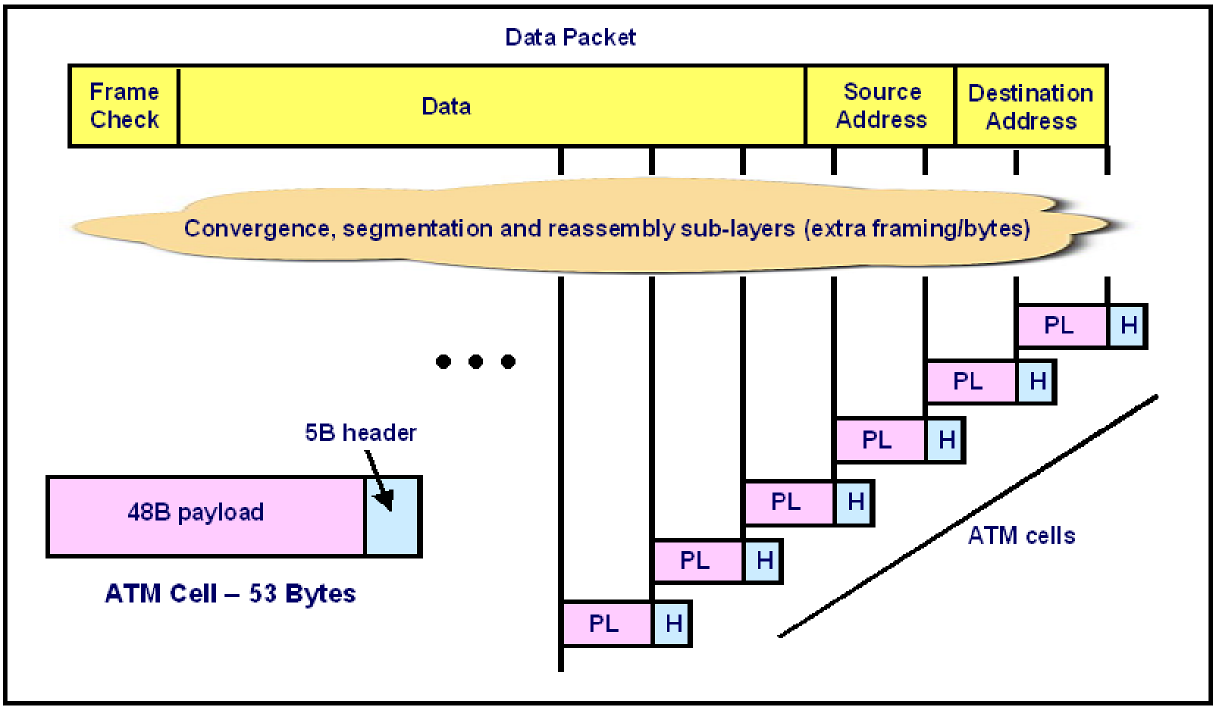

Figure 1. Adapting service traffic to ATM

In that stuffily-warm, lofty, log-cabin-like conference centre in frozen Quebec, leading ATM experts in the field were discussing how to maintain the end-to-end quality of voice calls given that:

- the queuing of ATM cells in ATM switches causes significant cross-network delay (latency)
- cell queuing times in the ATM switches vary randomly causing jitter
- cells are discarded when the queues in the switches get too big, causing clicks and gaps.

I put up my hand and asked why some of the smartest people in the industry were trying to figure out how to put back at the far end of the call the quality of service which they had gratuitously thrown away at the near-end (by ‘cellification’ and then ATM network transit). Particularly when we had a perfectly good circuit-switched network, already in service, which simply treated voice properly. I do not recall getting a compelling answer.

My question was both irritating and disingenuous. At the time everyone knew that the circuit-switched network had no future. It had been designed from top-to-bottom to do only one service well: to carry bandwidth-limited voice calls (around 180-3,200 Hz). For reasons later recounted by David Isenberg [2] trying to get this *one-product network* to do anything else was either impossible or hideously expensive and inefficient. Service innovation was only possible by the carriers themselves, not third parties, and was glacial in pace. No, carrying all traffic in packets or cells was the *only* way to liberate services from hard constraints: if that made it harder to do real-time 'isochronous' services like voice, then so be it.

Why Broadband ISDN?

The carriers knew they had to packetise their networks, and that they needed a new architecture (B-ISDN) supported by a number of new *protocols*. Here are some of the functions carriers thought they wanted to support.

- Set up a multimedia video-telephony call between two or more people (involves signalling).
- Carry the call between two or more people (involves media transport).
- Permit access to music, TV programmes and varied computer applications.
- Allow computers to communicate efficiently at high speeds.

Each of the above functions would be a chargeable service, leading to billing the customer. Carriers also needed to provision, operate, assure, manage and monitor their networks as per usual.

When carriers contemplate network transformation or modernisation, they like to huddle amongst themselves in the standards bodies to agree their target services and design a standardised architecture. The latter consists of a number of components, implemented using switches and servers, plus standardised message formats - protocols - which allow the intercommunication. It's easy to see why this ends up as a monolithic and closed activity - it all has to be built by the vendors, slotted together and then work properly. Getting the architecture into service is usually a highly complex and multi-year activity, and the cost will have to be covered by more years of revenue-generating service. Supporters of the model have pointed to the scalability and reliability of modern networks, the accountability which comes from centralised control, and the sheer functionality which can be put in place by organisations with access to large capital resources and internal expertise.

Critics point to the monopolistic tendencies of capital-intensive industries with increasing returns to scale, the resistance of carriers to innovation and the overall sluggishness and inflexibility of the sector. They note that circuit-switched networking began in the 1860s and that it had taken a further 130 years to automate dialling and digitise calls. By the time I was asking my question in Canada, B-ISDN had already been in gestation for around 15 years with no significant deployment.

The Internet, of course, also took its time to get started. TCP/IP came into service in 1983 and by the late eighties research groups were using email and remote log-in. That fusion of hypermedia and the Internet which gave us web browsers and web servers in 1993-94 launched the explosion in general Internet usage. By 1996 there was already a debate within the vendor and carrier community: was the future going to be IP and was the B-ISDN vision dead? It took a further ten years for the industry to completely take on board the affirmative response.

The Internet always ran on carrier networks. More precisely, the basic model of the Internet comprised hosts (computers running an IP stack and owned by end-users) and routers (sometimes called gateways) which forwarded IP packets to their correct destinations. The routers could be operated by *any* organisation (often maverick groups within carriers) and were interconnected using standard carrier leased lines. Almost all hosts connected to the routers by using dial-up modems at each end across switched telephone circuits. So from a carrier perspective, the Internet was simply people buying conventional transport and switched services - the specificity of the Internet was invisible. In truth, the Internet was beneath the radar of the B-ISDN project.

The Internet as the next-generation network

We already mentioned the many complex functions which need to be integrated to make a carrier network work. It's like a highly-specialised car engine. So where was this function for the Internet? Who was doing it? In what is the central mystery of the Internet, no-one was doing it. The basic Internet is *unusable*, because it does nothing but provide protocols to allow packetised bits to be transferred between hosts (i.e. computers). It is pure connectivity. However, pure global connectivity means that any connected computer application can be accessed by any other computer on the network. We have the beginnings of a global services platform. Here are some of the things which were, and are, *needed* to bring global services into being, roughly in the order the problem came up, and was solved.

1. Connecting to a service

Hosts and gateways operate on IP addresses for routing purposes. It is problematic, however, to use IP addresses (and port numbers) as end-system *service identifiers* as well. Apart from the usability issues of having to deal with 64.233.160.4 as the name of a computer hosting a service, IP addresses can also be reassigned to hosts on a regular basis via DHCP or NAT, so lack stability. A way to map symbolic names, such as *www.google.com* to an IP address is required. This was achieved by the global distributed directory infrastructure of the Domain Name System, DNS, also dating back to 1983.

2. Interacting with a service

Part of writing an application is to write the user interface. In the early years of computing, this was simply a command line interpreter into which the user typed cryptic codes if he or she could recall them. The introduction of graphical user interfaces in the late eighties made the user interface designer's task considerably more complex but the result was intuitive and user-friendly. The introduction of HTML and the first Internet browsers in the early nineties created a standard client which could be easily used to access arbitrary applications via HTTP across the Internet.

3. Connecting to the Internet

Research labs, businesses and the military could connect to the Internet in the eighties. But there was little reason for most businesses or residences to connect until the web brought content and a way to get at it. Initially the existing telephone network was (inefficiently) used for mass connection by the widespread availability of cheap modems. We should not forget the catalysing effects of cheap PCs with dial-up clients and built-in modems at this time. More recently DSL and cable modems have delivered a widely available high-speed data-centric access service.

4. Finding new services

Once the web got going, search engines were developed to index and rank websites. This was the point where Altavista, Yahoo!, and later Google came to prominence.

5. Paying for services

There is no billing infrastructure for the Internet, although there have been a number of attempts to support, for example, micro-payments. In the event, the existing credit card infrastructure was adapted by providers of services such as Amazon.com. More recently specialist Internet payment organisations such as PayPal have been widely used (96 million accounts at time of writing).

6. Supporting application-application services

Computer applications also need to talk to other applications across the Internet. They do not use browsers. The framework of choice uses XML, and we saw detailed architectures from Microsoft, with .NET, and the Java community with Java EE and companion editions, mostly since 2000.

7. Interactive multimedia services

Interactive multimedia was the hardest issue for the Internet. The reason is that supporting interactive multimedia is a systems problem, and a number of issues have to be simultaneously resolved, as we discuss next. So while for Broadband ISDN, voice/multimedia was the *first* problem, for the Internet, it has been the *last* (or at least, the *most recent*) problem.

Multimedia Sessions on the Internet

Layering telephone-type functions onto the existing Internet architecture is a challenge. Some of the basics are just not there. For example, the web uses names asymmetrically. There are a huge number of web sites out there which can be accessed by anonymous users with browsers. Type in the URL, or use a search engine. Click and go. But the website doesn't normally try to find you, and you lack a URL. The Public Switched Telephone Network (PSTN) by contrast names all its endpoints with telephone numbers. A telephone number is mapped to a device such as a mobile phone or a physical line for a fixed telephone. Various companies provide phone number directory services, and the phone itself provides a way to dial and to alert the called user by ringing. The basic Internet structure of routers and computer hosts provides little help in emulating this architecture. Somehow users need to register themselves with some kind of telephony directory on the Internet, and then there has to be some signalling mechanism which can look up the called party in that directory, and place the call. The IETF (Internet Engineering Task Force) has been developing a suitable signalling protocol (SIP - Session Initiation Protocol) since around 1999 and many VoIP companies are using it (Skype is a conspicuous exception, using a distributed peer-to-peer architecture with a proprietary protocol as we discuss in chapter 9).

The next problem is a phone equivalent. A PC can handle sophisticated audio and video, multi-way conferencing and data sharing. It is not, however, something which can be easily carried in a small pocket. Lightweight and physically small portable IP hosts are likely to have only a subset of a PC's multimedia capabilities and cannot know in advance the capabilities of the called party's terminal - more problems for the signaling protocol. A further reason for the relative immaturity of interactive multimedia services is the lack of wide-coverage mobile networks and terminals which are optimised for IP and which permit Internet access. The further diffusion of WiFi, WiMAX and possibly lower charges on 3G cellular networks will hopefully resolve this over the next few years.

Can the Internet, and IP networks in general, really be trusted to carry high-quality isochronous traffic (real-time interactive audio-video)? Whole books have been written on the topic [3] and it remains contentious. My own view is as follows. In the access part of the network, where bandwidth is constrained and there are a relatively small number of flows, some of which may be high-bandwidth (e.g. movie downloads), some form of class of service prioritisation and call admission control will be necessary. In the network itself, traffic is already sufficiently aggregated so that statistical effects normalise the traffic load even at the carrier's Provider Edge router. With proper traffic engineering, Quality of Service (QoS) is automatically assured and complex, expensive bandwidth management schemes are not required. As traffic continues to grow, this situation will get better, not worse due to the law of large numbers. Many carriers, implementing architectures such as IMS (IP Multimedia Subsystem), take a different view today and are busy specifying and implementing complex per session resource reservation schemes and bandwidth management functions, as they historically did in the PSTN. My belief is that by saddling themselves with needless cost and complexity which fails to scale, they will succeed only in securing for themselves a competitive disadvantage. This point applies regardless whether, for commercial reasons, the carriers introduce and rigidly enforce service classes on their networks or not - the services classes will inherently be aggregated and will not require per-flow bandwidth management in the core.

After establishing a high-quality multimedia session, the next issue of concern is how secure that call is likely to be. By default, phone calls have never been intrinsically secure as the ease of wiretaps (legal interception) demonstrates. Most people's lack of concern about this is based upon the physical security of the phone company's equipment, and the difficulties of hacking into it from dumb or closed end-systems like phones. One of the most striking characteristics of the Internet is that it permits open access in principle from any host to any other host. This means that security has to be *explicitly* layered onto a service. Most people are familiar with secure browser access to websites (HTTPS) using an embedded protocol in the browser and the web server (SSL - Secure Sockets Layer) which happens entirely automatically from the point of view of a user. Deploying a symmetric security protocol (e.g. IPsec) between IP-phones for interactive multimedia has been more challenging, and arguably we are not quite there yet. IMS implements hop-by-hop encryption, partially to allow for lawful interception. Most VoIP today is not encrypted: again Skype is a notable exception. As I observe in chapter 9, Skype looked for a while to be proof against third-party eavesdropping, but following the eBay acquisition, I would not bet on it now.

Architecture vs. Components

The Internet was put together by many people and organisations, loosely coupled though standard protocols developed by the IETF. Some of it works well, some Internet services are beta or worse. The world of the Internet is exploratory, incremental and sometimes revolutionary and it's an open environment where anyone can play and innovate. The libertarian ideology associated with the IETF theorises this phenomenon. The IETF saw (and sees) itself as producing enabling technologies, not closed solutions. Each enabling technology - security protocols, signalling protocols, new transport protocols - is intended to open the door for new kinds of applications. To date, this is exactly what has occurred.

The Internet model is disaggregated - the opposite of vertically integrated. Because the Internet is globally accessible and presents support for an ever-increasing set of protocols (equating to capabilities), anyone with a new service concept can write applications, distribute a free client (if a standard browser will not do) and attempt to secure a revenue stream. This creates a huge dilemma for carriers. In the Internet model, they are *infrastructure providers*, providing ubiquitous IP connectivity. In the classic tee-shirt slogan "*IP over everything*", the carriers are meant to be the "*everything*". But "*everything*" here is restricted to physical fibre and optical networking in the network core; copper, coax and radio in the access network; plus an overlay of routing/forwarding and allied services such as DNS. When it comes to end-user services, whether ISP services such as email and hosting; session services such as interactive multimedia, instant messaging, file transfer; or eBusiness services such as Amazon, eBay, e-Banking, there is no special role allocated for carriers - the Internet model says anyone can play.

This thought is entirely alien to the carriers, who have long believed they were more in the services business than mere bit transporters. Carriers have always wanted to move 'up the value chain' whether they were offering network-hosted value-added services or integrated solutions to their enterprise customers. As the carriers came to terms with the success of the Internet, and the collapse of Broadband ISDN, they attempted their own theorisation of the Internet. Not in the spirit of the libertarian open model of the IETF, but more akin to the vertically-integrated and closed models they were used to. They proposed to integrate

- data and media transport
- interactive multimedia session management
- computer application support

into one architecture where everything could be pre-specified and would be guaranteed to work. And so arrived the successor to Broadband ISDN, the Next-Generation Network (NGN).

The advantages of the NGN, as the carriers see it, include a well-integrated set of services which their customers will find easy to use, and a billing model which keeps their businesses alive. The disadvantage, as their critics see it, is the reappropriation of the Internet by carriers, followed by the fixing-in-concrete of a ten-year roadmap for the global Internet. The predictable consequence, they believe, will be the stifling of creativity and innovation, especially if the carriers use their NGN architecture anti-competitively, squashing third-party Service Providers, which is technically all too possible.

We should be clear here: anyone offering an Internet service has to develop a service architecture. In the IETF's view of the world, it is precisely the role of Service Providers to pick and choose from the IETF's set of protocol components and to innovate architecturally. There is absolutely no reason why the carriers shouldn't do their architecture on a grand scale through the NGN project if they wish. Critics may believe it's over-complicated, non-scalable and ridiculously slow-to-market. If they are right, Service Providers with lighter-weight and nimbler service architectures will win in the market-place, and the all-embracing NGN initiative will fail. 'Let the market decide' is the right slogan, but the market must first of all exist, which means that the Internet's open architecture must be preserved and not be closed down. Many carriers have significant market power and might be tempted to use it in order to preserve what they take to be their NGN lifeline against effective competition, so this is an issue for both customers and regulators. Thankfully, there are reasons to be hopeful as well as fearful.

Why did Broadband-ISDN really die?

There are positive reasons for the smooth uptake of IP, such as the easy availability of the TCP/IP stack as compared with competing proprietary data protocols, the relative simplicity of the basic Internet architecture and the prior existence of enterprise multiprotocol routers which could used directly as Internet routers.

Perhaps more important, though, were the problems with ATM. In the 1980s when ATM was being designed, the dominant usage mode was seen to be the multimedia successor to the phone call - human beings making videophone calls. As we saw above, interactive multimedia is the most challenging application for packet networks, requiring a complex infrastructure of signalling, terminal capability negotiation and QoS-aware media transport. It was not until the mid-nineties that large ATM switches capable of supporting the required signalling and media adaptation came to market - too expensive and too late.

Even worse, the presumed 'videophone' usage model for B-ISDN was highly connection-oriented, assuming relatively long holding times per call. So ATM was designed as a connection-oriented protocol with substantial call set-up and tear-down signalling required *for every call* to reserve resources and establish QoS across the network. This required per-call state to be held in each of the transit network switches. For comparison, millions of concurrent calls - called sessions or flows - transit a modern Internet core router - which knows nothing whatsoever about them.

It turned out that critical enabling technologies for the Internet, such as DNS, require brief, time-critical interactions for which a connection-oriented protocol is inappropriate. Even for connection-oriented applications such as file transfer, which use TCP to manage the connection, connection state is held only in the end-systems, not in the network routers, which operate in a connectionless fashion. This has allowed the Internet to scale.

So in summary, ATM had too narrow a model for how end-systems would network, and backed the wrong connection-oriented solution which couldn't scale. Because ATM was designed against a very sophisticated set of anticipated, predicted requirements, it was very complex, which led to equipment delays, expense and difficulty in getting it to work. The world moved in a direction not anticipated by the framers of B-ISDN and it was stranded, and then discarded.

References

[1]. Bannister, J., Mather, P., and Coope, S., *Convergence Technologies for 3G Networks*, Wiley, 2005, chap. 7.

[2]. Isenberg, D., The rise of the stupid network. *Computer Telephony,* August 1997, pp 16-26. Also available at: http://www.hyperorg.com/misc/stupidnet.html.

[3] Crowcroft, J., Handley, M. and Wakeman, I., *Internetworking Multimedia*, Taylor & Francis, 1999.