# NETWORK ARCHITECTURE: LESSONS FROM THE PAST, VISION FOR THE FUTURE

François Fluckiger, CERN, 1211 Geneva 23, Switzerland

*Abstract*

The design principles of the Internet have dominated the past decade. Orthogonal to telecommunications industry principles, they dramatically changed the networking landscape because they relied on iconoclastic ideas. First, the Internet end-to-end arguments, which stipulate that the network should intervene minimally on the end-to-end traffic, pushing the complexity to the end-systems. Second, the ban on centralized functions: Internet techniques (routing, DNS) are based on distributed mechanisms. Third, the domination of stateless protocols.

However, when facing new requirements such as multimedia traffic, security, Grid applications, these principles appear sometimes as architectural barriers. Multimedia requires Quality of Service (QoS) guarantees, but stateless systems are not good at QoS. Security requires active, intelligent networks, but dumb routers are insufficient. Grid applications require intermediary overlay networks.

Attempts to overcome these deficiencies may lead to excessively complicated solutions, distorting the initial principles (e.g. the myriad of QoS options; and after all, do we need them, why not "throw bandwidth at the problem"?). Middleware solutions are sometimes difficult to deploy (e.g for PKIs). "Lambda on-demand" technologies are conceptually nothing else than old switched circuits, that we never managed to satisfactorily integrate with IP networks.

Where is all this going? To help forming a vision of the future, this paper refers to several observations (marked as "Author's statement") made by the author over the past 30 years.

## REQUIREMENTS, ARCHITECTURE AND TEHNICAL DESIGN

*Networking Architecture* is usually understood as a set of abstract principles for the *Technical Design* of Networking Systems. By Networking System we mean here the implementation of a complete suite of technologies which provide services to end users, such as the Plain Old Telephone Network, or the Internet together with its major applications.

However, these abstract principles which form the architecture are themselves dictated by the *requirements* which are placed on any new network architecture. The logical development process of a new network system should therefore follow the top-down chain: Requirements -> Architecture -> Technical Design. We shall see in the next sections what may be understood by each of these three steps and to what extent the logical process has been respected in the past.

### *Requirements*

What are these requirements, theoretically at the root of the process? They include:

- *Functionality* (the prime function of the network to support point-to-point communications, or also point-to-multipoint such as multicasting or broadcasting).
- *Robustness* and *reliability* (or even *survivability*, such as exemplified with the Arpanet requirements)
- *Scalability* (such as in number of connected end-systems, nodes, or traffic volume)
- *Adaptability*: the ability to accommodate new requirements, and to evolve the architecture and the Technical Design accordingly)
- S*ervice levels* (e.g. two levels only: busy-tone or base service as with the telephone or continuum from full to none as with the initial Internet)
- *Predictability*: the level or required predictable behaviour, in particular in times of heavy usage
- *Performance guarantees*

To get a sense of the difference between networks in term of their requirements, the reader may think of how the telephone network (point-to-point, high robustness, two service levels, low adaptability, …) compares to the cable-TV network (broadcasting, no point-to-point, fairly adaptable, …).

### *Architecture*

The abstract principles which form the architecture are, in a purely logical process, derived from the requirements. Network architects have defined and organized over the past twenty years a range of principles, proposing groupings and taxonomies. For the sake of simplicity, we will formulate only seven architectural principles, what certainly constitutes a simplification of the reality.

1. The degree of *centralization* of the necessary functions, including management

2. The degree of *Intelligence of the Network* (as opposed to the degree of intelligence of the end-systems)
3. The identification of the *network constituents* (e.g.: "The network is made of nodes and hosts")
4. The *state* nature (e.g. strongly stateless, strongly stateful)
5. The *Naming* and *Addressing* principles (in particular, hierarchical, or flat address space)
6. The *traffic prioritization* (e.g. no priority, or traffic discrimination)
7. The location of the *security boundaries* (who is in charge of what in terms of security)

### *Technical Design*

The Technical Design is the last step of the design process, and it precedes directly the implementation of the hardware and software systems and their deployment. The Technical Design translates the architectural principles into practical technical specifications directly useable for the implementation. This is where the Network Protocols are specified, and where the addressing and naming schemes are laid down.

### *Evolving the Architecture*

The architecture of a Networking System is usually revisited to integrate new requirements. Examples include the introduction of the "Intelligent Network" concept in the Public Switched Telephone Network, or the security extensions in the Internet.

### *Reference Modelling*

We haven't talked so far of what is called the reference models, such as the well-known OSI model. Probably because it is often one of the first thing taught in networking lectures and in text books, reference modelling is sometimes thought as the major upstream step in the design. This is certainly not the case. As any model, network reference models provide an a-posteriori abstract view of the reality, but some reality must already exist for it to be modelled. This was the case of the OSI reference model, which provides a means for describing any communication system as formed of seven successive layers with relatively well defined functions in each.

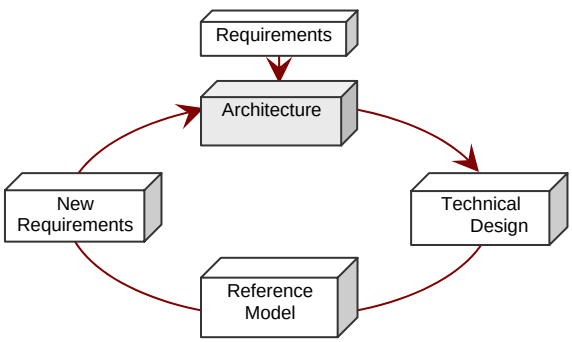


Figure 1: Design process of Network Systems

## INTERNET ARCHITECTURE

The history of the Internet as well as the successive evolutions of its architecture have been described by many Internet veterans [1].

### *Original Requirements*

The requirements of the original Internet were inherited from those of the ARPANET network, laid down in the 60's. They may be summarized as follows: the base function is to provide one-to-one (what was later called *Unicast*) bidirectional communication between pairs of end-systems; the robustness and survivability should be maximum; the scalability should be good and no absolute performance guarantees was required.

### *"Design philosophy"*

However, these initial requirements led more to a "design philosophy" than to a truly articulated architecture. As explained in [10], it was only in the mid-70's that real architectural discussions started, though at that stage, the documents ([2]: The TCP/IP specification by Cerf and Kahn, and [3]: Internetworking Issues by Cerf and Kirstein) were more technical designs (focussed on the TCP/IP protocols) with some architectural considerations than pure top-down Architectures. The Technical Design itself underwent successive modifications, as exemplified with the separation between the two major protocols (TCP and IP) which only occurred in 1974 and was implemented in the ARPANET in 1981.

### *First Architecture*

It was only in the 80's that the architecture itself started to be documented. A first key contribution [8] focussed on one aspect (the End-to-End argument, to which we come back later), and it was in 1988 that Clark published a rather comprehensive description of the original Interment Architecture. This was translated in 1996 only into a formal Internet standard document (Request for Comments RFC 1858, Carpenter Editor). Therefore, it was only when most of the Technical Design (that is, the protocols, the addressing scheme) had been completed that the "philosophy" was articulated into clear architectural principles. According to our limited list of seven principles listed above, the original Internet Architecture may be summarized as follows:

1. The network management should be fully distributed as far as possible.
2. The network should be as simple as possible, pushing the complexity to the end-systems.
3. The network is formed of two logical components: the nodes (later called routers) and the hosts (the end-systems).
4. No state should be maintained within the nodes.
5. The addresses should be numerical and of fixed size.
6. The treatment of data units should be egalitarian (no prioritization).
7. The end-systems should be in charge of their own security.

### *Extensions of the Architecture*

The original principles posed increasing difficulties in the 90's when new requirements emerged. The need to transport no only elastic traffic (for applications where the recipient can always wait for delayed data) but also real traffic (where data are unusable by the receiving application after a certain offset delay) led to the definition of traffic discrimination principles. Increased security concerns led to the development of techniques (IPSEC, Firewalls) which do not respect the "smart host / dumb nodes" principle. The shortage of network layer (IP) addresses gave rise to Network Address Translators (NATs) which also violate one of the corollaries of the End-to-End principles (namely that the address is carried unchanged from source to destination).

### *Evolving the Internet Architecture*

The beginning of the present decade saw initiatives from several of the original Internet designers to analyse the evolutions, the extensions and the violations of the architecture that occurred in the previous decade [6], [10] and to propose new avenues. As part of the analysis, they concluded that some of the extensions were not posing major architectural threats (IPSEC, MPLS, DiffServ / IntServ) whereas others had very negative effects because of having been developed without any architectural consideration (NATS, Firewalls, Web caches which alter the content of packets).

Another important outcome of these efforts was the rethinking of the End-to-End argument [8], the discussion of its major consequences. The argument was once described by B. Braden as "*Wonderfully ambiguous! The closest thing to a sacred text for the Internet Architecture*".

As a result of the "smart host / dumb nodes" principle, the network does not perform any flow control or buffering, or any error recovery or format conversion; also, addresses are carried end-to-end unchanged. The latter corollary implies, amongst other things, that it is up to the sending hosts to set their own address in the packets they inject into the Internet (unlike with the brave old X.25 network, where switches were inserting the source address in the call set-up packet). This was one of the reasons why IPSEC had to be developed.

## ON HOSTS AND NODES

We explained that Reference Modelling usually follows the Technical Design phase, as it uses the specified protocols as the reality corpus to be represented by an abstract structure. The OSI Reference Model laid down by Gien and Zimmermann in 1978 was no exception. The reference model relies on the layering principle which stipulates that communication functions are clustered into layers and that each layer has only two neighbours.

### *Layering Metaphors*

The usual metaphoric visual representation of the layering principle is that of a cupboard where the drawers represent the layers. However, one of the first representations of the concept, in the early 70's, used a different metaphor, the onion one where the concentric onion skins mimic the layers.

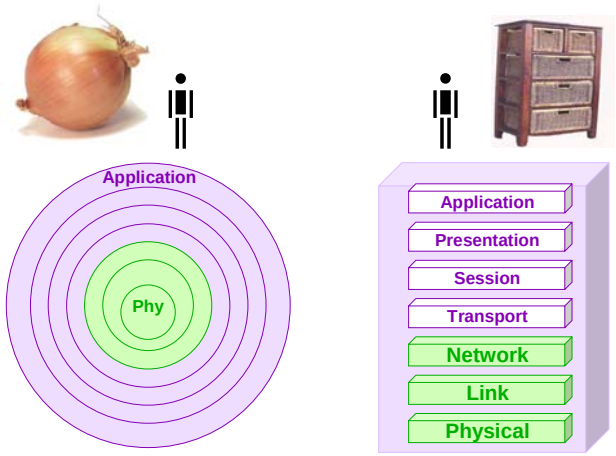


Figure 2: Onion and Cupboard layering metaphors

Amazingly enough, it is the onion metaphor that has been widely used so far to represent networking topologies, in particular that of the Internet, with the Nodes in the centre (the core) and the Hosts at the periphery.

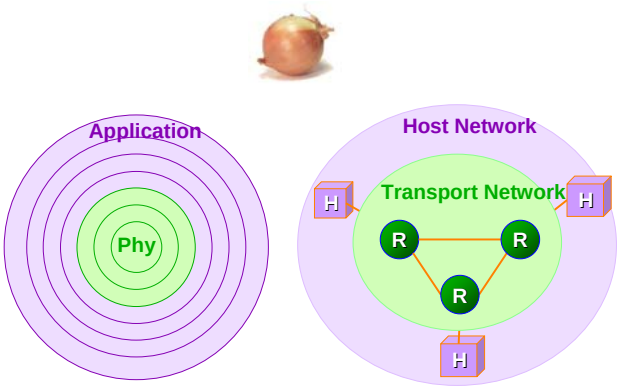


Figure 3: Onion metaphor to represent core/edge networking topologies

### *Network constituents*

The onion representation with two layers (the Transport Network layer, sometimes called the Communication Sub-network and the Host overlay layer) accredited the architecturally dangerous idea that networks in general and the Internet in particular have only two major logical

constituents. This let to neglect the key role of the Intermediate layers.

## INTERMEDIATE LAYERS

The dual-layer representation of network topologies with a single layer of hosts assumes that all non-node constituents, that is all hosts, are equivalent. But the collection of hosts is itself broken down into systems which deliver a service to the actual end user, and systems which only help to provide these services and belong in effect to intermediate layers.

An intermediate layer is a set of intermediary systems which are:

- invisible to the end-user
- topologically located on-top of the base transport network
- conspiring to deliver a specific service
- forming a topology (that is logically connected together, and communicating via a dedicated set of protocols)
- essential but not generally compulsory.

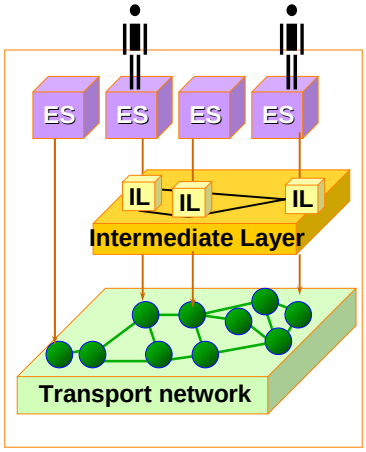


Figure 4: Intermediate Layer systems: they form a topology of communicating systems (IL)

### *Success and disappointments*

The existence as well as the importance of the Intermediate Layers has been largely overlooked in the architectural considerations of the Internet. In fact, it seems that there is one Intermediate Layer which is:

- truly universal
- invisible to the end user
- well managed, with a topology under control
- unchallenged
- of an undisputed non-proprietary technology.

Which one? The Internet Domain Name Server (DNS)!

But besides this success story, how many disappointments, because the resulting intermediate plane turned out

- to be fragmented and thus not supporting a universal service, or
- to use proprietary technology, or
- to be difficult to manage, with erratic topologies.

This has been the case for the IP multicast overlay, the Web caches, bandwidth brokers, or even the emerging Public Key Infrastructure (PKI), though in that particular case, it is arguable whether a universal service would actually be desirable.

### *Grid Middleware*

The Grid initiatives are developing middleware systems to perform functions such as data replica management, resource location and brokering, or authentication and authorization. One simple implementation model is that these functions are collapsed into a single central system. However, should actual intermediate planes formed of logically interconnected middleware servers be deployed, there are lessons to be drawn from the past. To conclude, Author's statement:

> ***The past has told us that except for the DNS, the other intermediate planes did not reach universality or openness, and that management issues are central to their success. The latter includes topology management (configuration changes, monitoring and optimization) as well as inter-domain management.***

## THE ROLE OF STATES

One of the central architectural principles of the Internet is the stateless nature of the transport network.

### *Back to basics*

In *stateless* (also called *connectionless*) networks data units can be sent at any time without prior authorization of the networks. Data units (packets in the case of the Internet) are the routed independently, carry the full address of the destination end-system, and may be lost or mis-ordered by the network. Ethernet and Internet IP are examples of protocols implementing the stateless principle. This is analogue to the service provided by the postal service or by the road network. Stateless is opposite to *stateful* (also called *connection-oriented*), where no data can be sent before being formally authorized by the network, usually by means of the creation of a connection between the end-systems. Examples of stateful network technologies include X.25, SNA, Frame Relay, ATM, ISDN and more recently "Lambda on-demand". The analogy is the Telephone service.

With stateful systems, the traffic is more predictable, it is easier to reserve resources and to guarantee a minimum quality of service. With stateless systems, there is no call set up delay, the routing is more dynamic and the resilience is higher.

### *IP and HTTP*

Not only Internet IP is stateless. The protocol designed by CERN to regulate the dialogue between web clients and web servers, HTTP, is also stateless. This is one of the reasons - rarely mentioned - for the success of the World-Wide Web technology: the application protocol (HTTP) and the transport protocol (IP) are of the same nature, and thus provide a very coherent basis for future developments.

### *States, memory and prediction*

What is the consequence for an IP router or an HTTP server of the stateless principle? The behaviour of a router may be summarized as follows: “take a packet, forward it, forget it”, and the behaviour of a web server as “take a request, serve it, forget it”. Of course these are minimal behaviours, and most routers or servers have modes where they can be cleverer, in particular by not immediately forgetting the recent traffic or requests. But by so doing, they de-facto maintain states, and therefore cease to behave as strict stateless systems.

How can stateless systems which forget past activities predict the load? This is indeed extremely difficult for them. Author’s statement:

***It is a fact of life that when you have no memory of the past, you cannot predict the future!***

## APLLICATIONS TYPES

To discuss the properties of network architectures, researchers and designers have proposed numerous taxonomies of network applications. To contrast the stateless and stateful approach, one classification is particularly relevant: that based on *bit rate* (sometimes also called *bandwidth*) types. It distinguishes between:

- Constant Bit Rate (CBR) applications
  They are conventional real-time applications - e.g. the traffic generated by PABXs
- *Available Bit Rate (ABR)*
  They are traditional bulk data applications - e.g. file transfer, mail
- *Variable Bit Rate (VBR)*
  Modern real-time applications - e.g. compressed audio or video

### *Bit rate and user satisfaction*

If we wish to plot the satisfaction of the users against the bit rate actually available from each of these application types, the results are significantly different.

- For CBR applications, below a given threshold, there is no service at all, and the user is fully unhappy. Beyond the same threshold, the service is established at a flat level of quality. The satisfaction of the user jumps from none to maximum, but increasing the bit rate will not change the quality, nor the level of satisfaction.
- For ABR applications, the more bit rate there is, the happier the user is (not necessarily linearly as shown on the over-simplified representation in Figure 5).
- With VBR applications, below a first threshold, there is no service at all, and the user is fully unhappy. Beyond a second threshold, the service is established at a maximum flat level of quality and the user satisfaction jumps reaches a maximum. Between the two, the satisfaction ramps up from none to maximum.

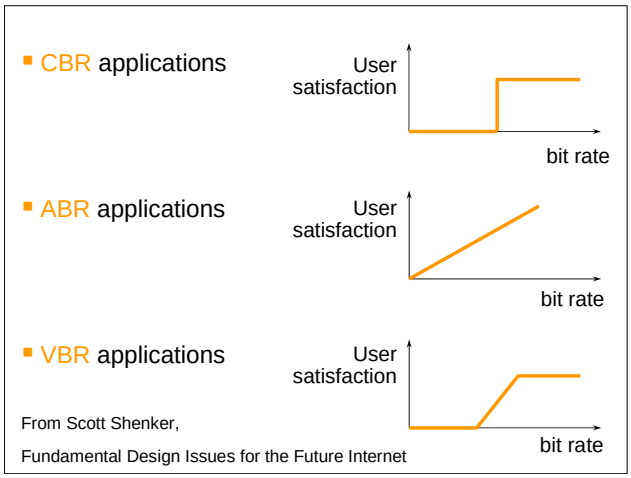


Figure 5: User satisfaction against bit rate for differing application types

### *Who is good at what?*

This simple classification of applications is extremely discriminatory when considering the state nature of network architectures.

As a matter of fact, stateful networks are good at supporting CBR applications, but this task is difficult for stateless networks. Conversely, stateless networks are very good at providing all the available bandwidth to end users (if a user waits to two a.m. to transfer a very big file, he or she may benefit from all the bandwidth available at that time over the local Ethernet, or the long distance IP VPN). But it is extremely difficult for stateful networks to offer all the available bandwidth between two points, even in the absence of third party traffic (in particular, but not only, because part of the bandwidth may be reserved by other communications, even in idle times). To summarize, Author’s statement:

***Stateful Networks are good at CBR, bad at ABR. Stateless Networks are good at ABR and VBR, bad at CBR.***

## QUALITY OF SERVICE

### *Why improve the quality of Service?*

The objective of the efforts undertaken since the beginning of the 90s about the Internet Quality of Service

is to improve the predictability of the service. Indeed, the historical "best effort datagram service" results in a somewhat unpredictable behavior. There are multiple reasons why this has become no longer desirable.

- Users may wish to set up Virtual Private Networks (VPN) over the shared Internet, such as the bandwidth of the pipes between sites part of the VPN is guaranteed. .
- Organizations which connect to an Internet Service Provider (ISP) at a given access rate may wish to have a secure aggregate bandwidth out of this access link, irrespective of the destination of their traffic.
- More and more multimedia applications use the Internet, in particular, audio and video streams. These streams usually need a minimum bit rate, below which it makes no sense to try and send audio or video traffic. These are requirements that do not apply to aggregates of traffic as in the above case, but to point-to-point flows between two end-systems.

## *Service discrimination*

Thus, the efforts for improving the Quality of Service (QoS) guarantees aim at moving away from the historical model of traffic where all packets are handled with the same priority by the network. By abandoning the pure egalitarian treatment of the datagrams, the new Quality of Service techniques create discrimination between packets. This is called *service discrimination*.

Service discrimination does not create any resource by itself –we do not get more bit rate on a link because some packets have higher priorities - therefore, it does not solve all problems of Quality of Service. If a network, or a portion of a network (a link), has not enough capacity, service discrimination will not help for all the traffic. However it will help for some. Indeed, the objective of service discrimination is to give better service to some traffic. But this is done at the expense of giving a worse service to the rest. Hopefully, this only occurs in times of congestion.

## *Integrated Services*

The first substantial work on Quality of Service in the Internet started in the early ‘90s in the framework of what was called the *Integrated Services* (IS) model. The first release was made in ‘93.

The Integrated Services model is based on the statement that a single class of packets is no longer sufficient, and that new classes with higher priorities are needed, in the same way as we have the economy, business and often first class with airlines. How many new classes were needed? The Integrated Services model opted for two new classes of packets, resulting in a total of three possible classes in the new discriminated Internet world:

- The *best effort service class* (*BE*)
  This is the default class
- The *controlled-load service class* (*CS*)
  There, if the sender respects a certain traffic profile (that is a certain bit rate) for a given flow, then the network promises to behave as though it was unloaded, but without quantitative guarantees in particular of the latencies of the packets.
- The *guaranteed service class* (*GS*)
  There, packets are promised to be delivered within a firmly bounded delay. This is for special applications with very stringent time delivery requirements.

## *Resource Reservations*

The guiding principles of the Integrated Services model are the following:

- Resource reservation is necessary.
  To improve the guarantees, the key resources needed in the network must be reserved in some way.
- Reservations operate on flows.
  A flow is a stream of packets between one source and one destination. For every flow that needs to benefit from either the CS or the GS service, reservations need be made.
- Routers have to maintain flow-specific states.
  By state, we mean in practice a block of memory in the router where information about the flow and its requirements are stored: the service class (CS or GS), the bit rate to guarantee for that flow, the conditions for delay if applicable, etc.
- Dynamic Reservations need a signaling (set-up) protocol.
  This protocol has been specified and is called *Resource Reservation Protocol*, or *RSVP*

RSVP is the mechanism defined by the Integrated Services for reserving resources in the network. It is called a signaling protocol, because its aim is to signal to the network that a given flow is going to require certain guarantees for latencies and loss ratio, if the flow respects a certain bit rate. RSVP is based on a number of principles.

- RSVP has to co-exist with regular datagram services
  Any router which supports RSVP also supports the regular best effort datagram service
- RSVP does not set hard connections
  Instead, the connections are said to be "soft", as the originator has to periodically refresh the state by repeating the reservation request.

To summarize, RSVP is an attempt to extend the Internet architecture towards the stateful world, in order to compensate one of the deficiencies of the stateless philosophy: the difficulty to provide QoS guarantees.

## *Diffserv*

*Diffserv*, which stands for *Differentiated Services*, is another technique aiming at overcoming the problem of heavy classification - that is the process for routers to

know to which service class a packet belongs. The idea here is to "mark" the packets with an indication of their priority in order to avoid having routers examining multiple fields. This mark is called a "differentiated mark" or a *Diffserv Code Point* (*DSCP*) and serves to map to a differentiated treatment to be applied to the packet. For a fast classification, the "mark" must be:

- of fixed length
- located at the beginning of the packet
- in a fixed position
- to be used as a direct pointer to find out what the differentiated treatment is to be.

The use of the mark is a technique which assumes that there is a core in the network which is "Diffserv-capable", that is, made of routers which understand the Diffserv marks and know how to exploit them for efficiently determining the packet priority. At the edge of this Diffserv core, the edge routers must be provisioned with the appropriate instructions to mark the packets (e.g. based on identification of flows such as source and destination addresses). However, this technology is not provided with mechanisms to reserve resources or decide whether a flow may be granted a high priority mark (what is called *Admission Control*).

Therefore, Diffserv is usually combined with RSVP in the core of the Internet.

## FUNDAMENTAL ANTAGONISM

### *Scalability*

The central positive consequence of not having to maintain states is that such stateless systems have extremely good scaling properties. As a matter of fact, maintaining states implies memorizing parameters of the states (e.g. for network nodes, the identity of the communication end-systems, quality of services parameters attached to the connection, measuring of the traffic, authorized traffic profiles; for servers: user login information, user profiles, activity log, …), which consumes memory space. In addition, the establishment of the states (that is, the call set-up in networks, or the login process in servers) creates a definitive processing overhead, which also constitutes a limiting factor to the scalability of such systems.

This property of excellent scalability of stateless systems may be illustrated with the analogy of transportation systems. Trains operating with no reservation systems may scale very well in terms of transported travellers. Peaks of traffic may be absorbed (by packing passengers in corridors, …). Systems with strict reservation mechanisms can only offer a fixed number of seats.

### *Quality of Service*

Conversely, stateful systems, in particular when provided with resource reservation mechanisms can inherently provide the best Quality of Service guarantees. Stateless systems, being incapable - when strictly implemented - of predicting the load, and having no admission control mechanisms, have difficulties to avoid congestion situations systems as well as securing resources for particular types of traffic, that is to provide QoS guarantees. This is again exemplified with the train analogy mentioned above: the TGV-like services with mandatory reservations to ensure to those who can book a seat, a predictable, well-known quality of service, or train services with no reservations, such as in certain emerging countries, which scale extremely well with no quality guarantees.

In summary, Author's statement:

***A fundamental antagonism exists in all systems between scaling and Quality of Service.***
***Stateless systems (no reservations) scale well, but are bad at QoS.***
***Stateful Systems (reservations) are good at QoS, but bad at scaling.***

## COMPLEXITY AND SCARCITY

Quality of Services Technologies as introduced in the Internet (RSVP, Diffserv, …) have definitely complicated the Internet Architecture, such as importing stateful behaviour and service discrimination practices. Note in passing that these technologies are themselves insufficient to support a full QoS service: other functionalities such as Capacity Admission Control ("Are there enough resources to satisfy a new request?), Policy Admission Control ("Is this User authorized to request these resources?"), or Parameter Provisioning are also necessary.

Is all this complexity needed? Is "Throwing bandwidth at the problem" not the true solution?

Author's statement: It is indeed a fact of life that:

***When resources are scarce, complex systems are needed to manage them. When resources are abundant, simple systems may suffice***

## RESOURCES AND USAGE

Therefore, the question arises: are we, for the network resources, in an era of abundance or in an era of scarcity?

### *Oscillating mismatch*

Usually, in networking and more generally computing, the resources that users can acquire at a definite cost do not evolve linearly. Instead, they tend to follow a stepped function were periods of moderate growth are followed by phases of accelerated increase, generated by the advent of disruptive technologies or massive infrastructure investments by service providers.

Similarly, if the curve of the demand starts at a lower level than the offer, this period of abundance where the

demand grows at a moderate pace is also generally followed by a phase of abrupt increase, often triggered by the development of new application made possible by the context of abundance and the impression of "unlimited capacity". But this demand curve generally collides with the offer curve, creating a period of scarcity. It has therefore been observed by the author that the evolution of the Wide Area Network (WAN) bandwidth offer/demand follows two mismatching curves, resulting in a succession of phases of abundances and phases of scarcity.

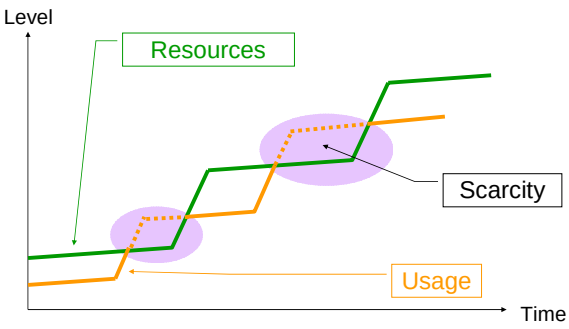


Figure 6: Simplified view of the mismatch oscillation between WAN capacity and WAN usage

### *Today's situation*

Where are we today? The core part of the Internet Service Providers (ISP) infrastructure is in most cases over-provisioned and the average load may be estimated as lying between 10 and 25%. Most Local Area Networks (LANs) are largely over provisioned.

However, there still exist a number of bottleneck causes in today's networks. Les us just cite a few.

- An increasing number of distributed organizations (such as multi-site companies) contract their Wide Area Network to ISPs in the form of Managed Network Services with guaranteed but limited bandwidth (this is often implemented by combining the MPLS, RSVP and Diffserv technologies).
- Because of this, the LAN-WAN interface is becoming again a frequent bottleneck.
- A portion of Internet infrastructures relies on slow links (slow as opposed to fast fibre connections). This is the case for the radio-transmission sections necessary for mobile applications.
- Voice over IP (VoIP) infrastructures may become suddenly congested if, due to exceptional reasons (such as an accident on a motor way), a number of portable telephone users start calling simultaneously.

For these reasons, Quality of Service technologies are and will continue to be used, in order to support Managed Services (QoS-secured VPNs), Voice over IP, protection of critical traffic, …

However, taking a more general perspective, it is the author's opinion that we are currently in a phase where network bandwidth is abundant. The next swap to scarcity will probably be due to the conjunction of the generalization of existing fast access technologies (e.g. ADSL-2) and the advent of disruptive applications such as broadcast-quality video steaming or Grid computing.

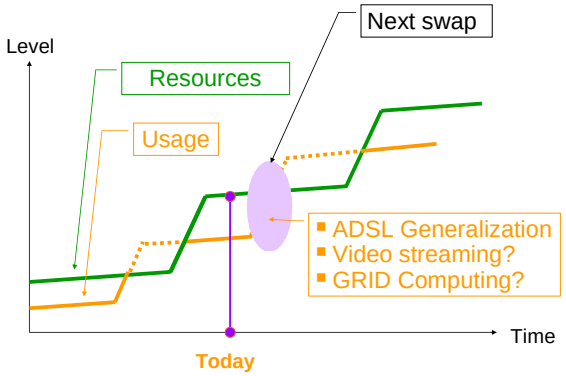


Figure 7: Nest swap to WAN capacity scarcity phase may come from new, disruptive applications

In summary, Author's statement:

***If history repeats itself, the next swap to network bandwidth scarcity will take place in the coming years, triggered by the advent of disruptive applications.***

## ON SMOOTH TRANSITION

The transition to IP version 6 (IPv6) is underway. This is a smoothed transition as it is expected that the population of IP version 4 (IPv4) end-systems will progressively convert to IPv6. When this is completed, in theory, the IPv4 infrastructure may simply disappear. However, in such a phased transition, until the full conversion is completed, two networking systems have to coexist, that is to be maintained in parallel, together with gateway mechanisms between the two.

When observing the past, we may note that this process proved to be a difficult one, and the most notable attempts of network transition ("any network" to OSI, Decnet Phase IV to Decnet Phase V) have failed. Conversely, the "flag-day" conversion worked generally well. This is the case of the successful ARPANET transition from NCP to TCP/IP in 1983. As a matter of fact, this was facilitated by the small size of the network at that time. But in telecommunications, large scale D-Day conversions also worked well, as exemplified by the swap of the French Telephone Network from 8 to 10 digits, affecting millions of subscribers.

The Flag-day transition to IPv6 is no-longer an option for the Internet of course. But we may predict that the

smooth transition will remain complex and costly. Some specialists are now talking of co-existence between the two versions instead of transition to the new one, highlighting the potential open-ended aspect of the process. In summary, Author's statement:

***Notable smooth transitions in Networking have failed so far, whereas flag-day swaps have succeeded. Transitions may lead to endless co-existence***

## EPILOG

The author tried in this article to exploit some of the lessons from history to help forming a vision of the future. Additional observations and Author's statements may be found in his Reference Text Book on Networked Multimedia [11].

Let us finish this discussion with a general comment on the evolution of network technologies. After more than a century of telecommunication progress, the technologies still divide into two broad categories: the stateful class and the stateless class. The former is excellent at controlling the load, predicting the behaviours, providing Quality of Service guarantees, supporting stable routes, but is bad at scaling or supporting broadcasting. This is where we find the Telephone Network, ATM, or Lambda-on-demand. The latter is good at scaling, can easily provide to users all their available bandwidth, can support broadcasting but is bad at guaranteeing quality of service. This is where IP and Ethernet stand. Whilst willing to keep their respective strengths, each of them develops complex adds-on (MPLS or RSVP for IP; ABR for ATM) to feature some of the goodies of the other class.

We will keep observing the evolution of the technology to see whether new paradigms finally emerge that would break this duopoly. Until then, each camp, though being persuaded of its own merits, may still look longingly at the other.

***The grass is always greener on the other side of the hill.***

## REFERENCES


[1] B. M. Leiner et al, "A Brief History of the Internet", http://www.isoc.org/internet/history/brief.shtml, Intrenet Society, December 2003

[2] V. Cerf and R. Kahn, "A Protocol for Packet Network Intercommunication". IEEE Trans on Comm, COM-22, No. 5, May 1974, pp. 637-648

[3] V. Cerf and P. Kirstein, "Issues in Packet Network Interconnection", Proc. IEEE, v.66, 11, November 1978.

[4] D. Clark, "The Design Philosophy of the DARPA Internet Protocols". Proc SIGCOMM 1988, September1988

[5] B. Carpenter, Editor, "Architectural Principles of the Internet". Internet Architecture Board, RFC-1958, June 1996

[6] M. S. Blumenthal et al. "Rethinking the design of the Internet: the end-to-end arguments vs. the brave new world", ACM Transactions on Internet Technology (TOIT), Volume 1, Issue 1, August 2001, Pages: 70 - 109

[7] D. Clark and D. Tennenhouse, "Architectural Considerations for a New Generation of Protocols". Proc ACM SIGCOMM, September 1990

[8] J. Saltzer, D. Reed, and D. Clark, "End-To-End Arguments in System Design". 2nd International Conf on Dist Systems, Paris France, April 1981

[9] D. Clark, L. Chapin, V. Cerf, R. Braden, and R. Hobby, "Towards the Future Internet Architecture". Network Working Group RFC-1287, December 1991

[10] R. Braden et al, , "Developing a Next-Generation Internet Architecture", July 2000

[11] F. Fluckiger, "Understanding Network Multimedia", Reference Text Book, Prentice Hall, ISBN 0-13-190992-4