CERN-DD 88-18

C1

CERN-Data Handling Division
DD/88/18
F. Flückiger
October 1988

CERN LIBRARIES, GENEVA

CM-P00059897

GATEWAYS AND CONVERTERS IN COMPUTER NETWORKS

Gateways and Converters in Computer Networks

François FLUCKIGER

CERN, European Laboratory for Particle Physics, CH-1211
Geneva 23, Switzerland (Tel.: +41 22 834 989; E-mail:
FLU.DN@GEN.CERN)

The term "Gateway" has a wide range of meanings. A clear distinction between the function of protocol translation (called "Converter") and the function of interconnection of networks through a small number of transit points (called "Gateway") avoids misleading statements on the desirability, merits and drawbacks of such systems. In the real world, not all Gateways are Converters and vice-versa.

Converters are never totally transparent and always entail a loss of functionality. Gateways must be more reliable and available than the networks they connect. This is not always the case, especially with Application Gateways, which also entail artificial multiplication of transit delays, traffic and cost on chargeable networks.

CERN has developed and operates two powerful multi-protocol Application Converters: MINT for Mail, and GIFT for File Transfer (GIFT is the result of an international collaboration). The main lessons are that operating store-and-forward Gateways or Converters is always manpower consuming, and development of on-the-fly Converters requires (when technically achievable) considerable investment, paid back by lower operational overheads.

As a conclusion, pure Gateways (or Relays) must be absolutely avoided, except when (a) simplifying drastically addressing (via "Late-binding"), or (b) mandatory for network management. Converters are mandatory for connectivity between interim protocols, and are a strategic component of OSI migration.

Keywords: Gateways, Converters, Relays, Bridges, MTA, Chain of Converters, Converter Transparency, Gateway Reliability, Converters at CERN, GIFT, MINT, On-the-fly Conversion, Store-and-forward Conversion, Late Binding, OSI Migration.

1. Gateways and Converters

Like many of the terms used in computing (the best example being the term "Network" itself), the term "Gateway" covers a wide variety of notions. As a consequence the statements made and conclusions drawn on the desirability, the needs, the advantages and drawbacks of such "Gateways" are often misleading when the definition of such systems is not clearly established.

1.1. Two Distinct Functions: Definition and Terminology

Two distinct functions associated with the term "Gateway" must be defined:

(1) *The interconnection function.* A system performs an interconnection function when it represents a particular (sometimes unique) point of data transit between 2 (or more) physical (say up to level 3) or logical (say "application") networks. Note that the two interconnected networks may perfectly well have the same technology, and use the same protocols.

In the following, this interconnection function will be called the “gateway” function, since this is the term which has been so far the most often associated with this notion.

(2) *The conversion function.* A system performs a protocol conversion (or translation) function when a mapping is realized at a given layer between two different communication protocols. Note that the mapping can be implemented in a variety of ways (software module, dedicated hardware, “on-the-fly” translation, temporary storage,...).

In the following, this conversion function will be called the “converter” function.

1.2. Examples of Gateways and Converters

(1) *Example of Gateways.* The gateway function is (to simplify) realized at two possible levels: the network level, and the application level.

An X.25 switching system between a private and a public X.25 network or between two public networks, and an interconnection box between two Ethernets are examples of “network level gateways” since they interconnect communication subnetworks (sometimes called “physical” networks) (in the above examples, of identical nature). When such gateways interconnect “LANs” at level 2, they are often called “bridges”.

An Electronic Mail X.400 Message Transfer Agent (MTA) interconnecting a campus mail network and a nation-wide network is an example of an “application level gateway” since it interconnects application (sometimes called logical) networks (in this example, using the same protocols). Such gateways, especially in the field of Mail, are often called “relays”.

(2) *Examples of Converters.* The converter function can be realized in two ways: within an end-system, and within an intermediate system.

A system which locally (within the end-system) transforms the mail (proprietary) protocol of a given manufacturer to X.400 is an end-system converter. It only implements the converter but not the gateway function (as defined above).

An Ethernet-X.25 interconnection box, a RSCS-UUCP converter physically connected to both EARN and EUnet are converters fulfilling also the gateway function (both protocol translator and point of transit between networks).

1.3. Combination of Functions

All the three possible combinations of the two functions exist in real networking systems:

(1) *Gateway function only* where no protocol conversion is performed because the technology and protocols used in the interconnected networks are the same;

(2) *Converter function only* when the converter system is implemented in an end-system;

(3) *Gateway and Converter functions* where systems interconnect two real networks with different technology and protocols.

Message 1

Not all “Gateways” are “Converters”
and vice-versa.

2. Problems with Gateways and Converters

2.1. Problems with Converters

The main two problems with protocol converters are

- the transparency, and
- the overall resulting functionality.

These two problems are inherent to any translation activity (not only in computing).

(1) *Protocol converter transparency.* The problem of the transparency of converters will be expressed by posing two simple questions: *When two systems using two distinct (sets of) protocols communicate through a converter system,*

- *can the software used in the two end-systems remain unchanged?*
- *can the user ignore the specificities of the remote system, and even not notice he crosses a converter?*

The answers to these two questions are (the rationale is not given here):

- 1: *Yes, it is possible to design converters such that the software of the end-systems using them remains unchanged; this should always be one of the main targets of any converter.*
- 2: *Yes, it is possible to implement Communication Subnetwork converters in such a way that the user can ignore he crosses them, but it is*

much more difficult to achieve this with application gateways (at least, in many cases, the gateway has to be “addressed” by the user).

(2) *Overall resulting functionality through converters.* All protocols targeting a given service do not offer the same functionality. The maximal theoretical functionality attainable through a converter translating between two protocols is just the intersection of the set of functions of these two protocols. In practice, it is always less (one of the most famous problems is that of “returning the error codes”).

Consequently, a protocol converter always entails a loss of functionality. Therefore the cascading of converters is worse, entailing along the chain a progressive degradation of functionality (again, this remark is general to any chain of translation, including obviously natural languages).

Message 2

Converters are never totally transparent.

2.2. Problems with Gateways

The major difficulties with gateways are of an operational nature. Three of the main problems are treated here:

- reliability (and performance),
- availability,
- multiplication of delays and traffic.

(1) *Reliability.* By nature, a gateway being a particular (often single) point of transit between two (or more) networks, constitutes

- a single point of failure,
- a single point of congestion.

Consequently, *Gateways should always be more reliable than the Networks they connect.*

Unfortunately, this is often the reverse; two of the many reasons are

- The hardware reliability is sometimes lower than that of the connected networks (e.g. gateways between “mechanics-less” subnetworks using mechanical components!).
- The gateways are not always professionally operated (especially in the research and academic world).

(2) *Availability.* This notion is distinct from the reliability: a failed system is unavailable; but reli-

able systems may have poor availability too. This is particularly true for Application Gateways for the following reasons.

Application gateways are often implemented on general-purpose computers, and not on dedicated hardware/firmware systems. Consequently, they may be unavailable for many reasons, including

- maintenance windows,
- development windows,
- software upgrades (e.g. move to new versions of operating systems),
- hardware upgrades.

Furthermore, such computers are sometimes used to offer other services than the Gateway service, entailing additional reasons for unavailability.

Message 3

Gateways must be more reliable and available than the Networks they connect.

(3) *Multiplication of delays and traffic.* In the case of Application Gateways (such as Mail Relays), crossing one or several such Relays connected often to the same Communication Subnetwork (e.g. an X.25 Network), multiplies artificially the transit elapsed time, the traffic on the Subnetworks (the same data cross the same network several times) and consequently the cost on chargeable networks.

3. Example: Converters at CERN

The objective of this short section is to draw a few conclusions from 5 years of experience in developing and operating Converters at CERN.

CERN launched 5 years ago a programme of Subnetwork and Application Converters to cope with the great diversity of protocols to be supported [1]. Two major Application Converters have been implemented:

- MINT for Electronic Mail,
- GIFT for File Transfer.

3.1. MINT: Multiple Mail Converter System

MINT [2,3] is a multiple protocol converter system for Mail, based on a central multi-protocol machine and a set of satellite bi-protocol systems.

It currently performs conversion between any pair of the 6 following protocols: UNIX Mail (UUCP), VMS Mail (DECnet), EARN Mail (RSCS), X.400 (EAN), Norsk Data Mail, TCP/IP Mail (SMTP).

MINT uses a store-and-forward technique to perform the conversion.

MINT has been entirely developed by CERN. Mint is operational since early 1986.

3.2. GIFT: Multiple File Transfer Converter System

GIFT [4,5] is a multiple protocol converter for File Transfer, based on a single multi-protocol machine.

It currently performs conversion between any pair of the 5 following protocols: CERNET FTP, DECnet FTP, JANET "Blue Book", TCP/IP FTP and CDC RHF. Developments are underway to add the DFN FTP and the ISO FTAM.

GIFT uses (almost uniquely in the word) the technique of the "on-the-fly" conversion (no intermediate storage of the files, but protocol translation at the time the transfer progresses).

GIFT is the result of an international collaboration, initially centred around the HEP community. Main contributors were sofar INFN (I), CERN, RAL-Oxford-JNT (UK), SARA (NL).

GIFT is operational at CERN since 1985, and at SARA since 1988.

3.3. Lessons from MINT and GIFT

Unlike pure Gateways, Application Converters need to be developed. Table 1 compares for MINT and GIFT the two usual key figures for project management: manpower for development and manpower for operation.

The Conclusion is that MINT (6 protocols) consumed 10 time less manpower for the develop-

ment than GIFT (5 protocols sofar), but requires 10 times more manpower to be operated. The explanation being that on-the-fly translators are terribly difficult to design and implement (unlike store and forward based converters). But once done, they almost never risk to be overloaded or congested, unlike store and forward systems.

Message 4

Operating store-and-forward gateways and converters is manpower consuming.

Message 5

For converters (as for other human realizations) the more one invests (development), the less one spends (operation).

4. Why Gateways and Converters?

If Gateways and Converters are unreliable and unavailable, non-transparent, difficult to develop, expansive to operate, why are there so many of them around the world? What are they good and bad for?

4.1. Reasons for Gateways

The question is: if two networks use the same technology and protocols, why interconnect them through a single (or a small number) of gateway(s) which will be single point of failure and congestion, instead of creating a single general fully interconnected network? The two main reasons are

- addressing, and
- management.

(1) *Addressing*. In several cases, pure gateways (between similar networks) are only set up to facilitate addressing by using what is called the "late binding addressing technique". With this technique, within a given network, the communicating systems only have to know the addresses of the systems within their network. Beyond it, they only know the address of a Gateway (also called

Table 1

	Development (Man years)	Operation (Full time equiv.)
MINT		
Store-and-forward CERN alone	2	2
GIFT		
On-the-fly collaboration	15/20	0.2

Relay), which in turn knows the addresses of the systems in the other network(s), and maps a reference given by the source system (often a name) to the actual destination address.

This technique is widely used in Electronic Mail where Message Transfer Agents only resolve addresses and relay traffic. But modern mail protocols, such as X.400, although allowing mail relaying for addressing simplicity, can perfectly work straightforward without artificial gatewaying, provided the adequate directory services are available.

(2) *Management*. When interworking between two networks using the same technology, but administratively distinct, is sought, single (or a small number of) interconnection points are often required for

- access control (security, authentication, ...),
- traffic monitoring,
- accounting,
- simple problem diagnosis.

4.2. Reasons for Converters

The three main reasons for having converters now and in the future are

- they increase the connectivity;
- they minimize the number of protocols to be supported and used;
- they are a strategic component of the migration to OSI.

(1) *Increased connectivity*. Some systems are only provided with protocols which are not supported by others. The only way to achieve interworking between such sets is via converters.

(2) *Less protocols to support and use*. Certain end-systems could be provided with almost all the protocols that exist for a given service (because the software is available). Such systems could therefore in theory communicate with all the others, using each time the far-end protocol, and without need for protocol converters.

Such a solution is unrealistic, because of the overhead to support all these protocols on the end-systems, and the reluctance of the end-users to have to learn so many user interfaces. Converters therefore minimize the number of protocols to be supported and used on the end-systems.

(3) *Strategic component of the OSI migration*. The migration to OSI (in which continuity of service is mandatory) can be viewed as composed of two logical successive phases:

(a) Phase 1, where all the existing systems are

still provided with the interim protocols, and a few of them already also provided with OSI. During this phase, Converters between interim and OSI protocols are highly desirable (for the above reasons), but not mandatory.

(b) Phase 2, where some of the existing systems are no longer provided with the interim protocols (because support given up on old systems, or no implementation of interim on new systems), but not all yet have OSI. In this phase converters between interim and OSI protocols are mandatory for interworking.

5. Conclusions

The conclusion will take the form of the two last messages:

Message 6

Converters are mandatory for
(1) connectivity between interim protocols,
(2) migration to OSI.

Message 7

Pure Gateways (or Relays) must be absolutely avoided except when
(1) simplifying drastically addressing,
(2) mandatory for management.

References

- [1] B. Carpenter et al., Two Years of Real Progress in European HEP Networking: A CERN Perspective, *Proc. Computing in HEP Conference*, Asilomar (1987).
- [2] D. Heagerty, Practical Experience with High Level Gateways for Mail Transfer, *Computer Networks and ISDN Systems* 13 (1987) 195-200.
- [3] D. Heagerty, Experience with Mail Gateways and Transition to X.400, *Proc. DECUS Symposium*, Rome (1987).
- [4] M. Ferrer et al., GIFT: An HEP Project for File Transfer, *Proc. Computing in HEP Conference*, Amsterdam (1985).
- [5] F. Fluckiger, A Brief Presentation of GIFT: The General Internetwork File Transfer Project, CERN Document CERN/DD/CS-G40.