Message Management in Message Handling Systems

Sandy Shaw
Edinburgh Regional Computing Centre
University of Edinburgh

The management policies of a message handling system are described. The principal concerns are protection of the messaging system from congestion and protection of its host facility. Other benefits include improved status information both for service users and the service provider.

Introduction

The new standards for Computer Based Message Systems (CBMS) owe much to the experience gained on previous messaging systems. With the goal of satisfying the widest range of requirements, to permit extensive interworking, the standards address the needs of interpersonal messaging, multi-media messaging, access to existing telematic and telex services and other matters of global concern. There remain a variety of issues where the standards, properly, do not attempt to prescribe the local operation of the messaging system. However some of the issues, particularly those concerned with the management and control of the local messaging system, have received attention within existing messaging systems and policies have been developed which remain relevant under the regime of new standards.

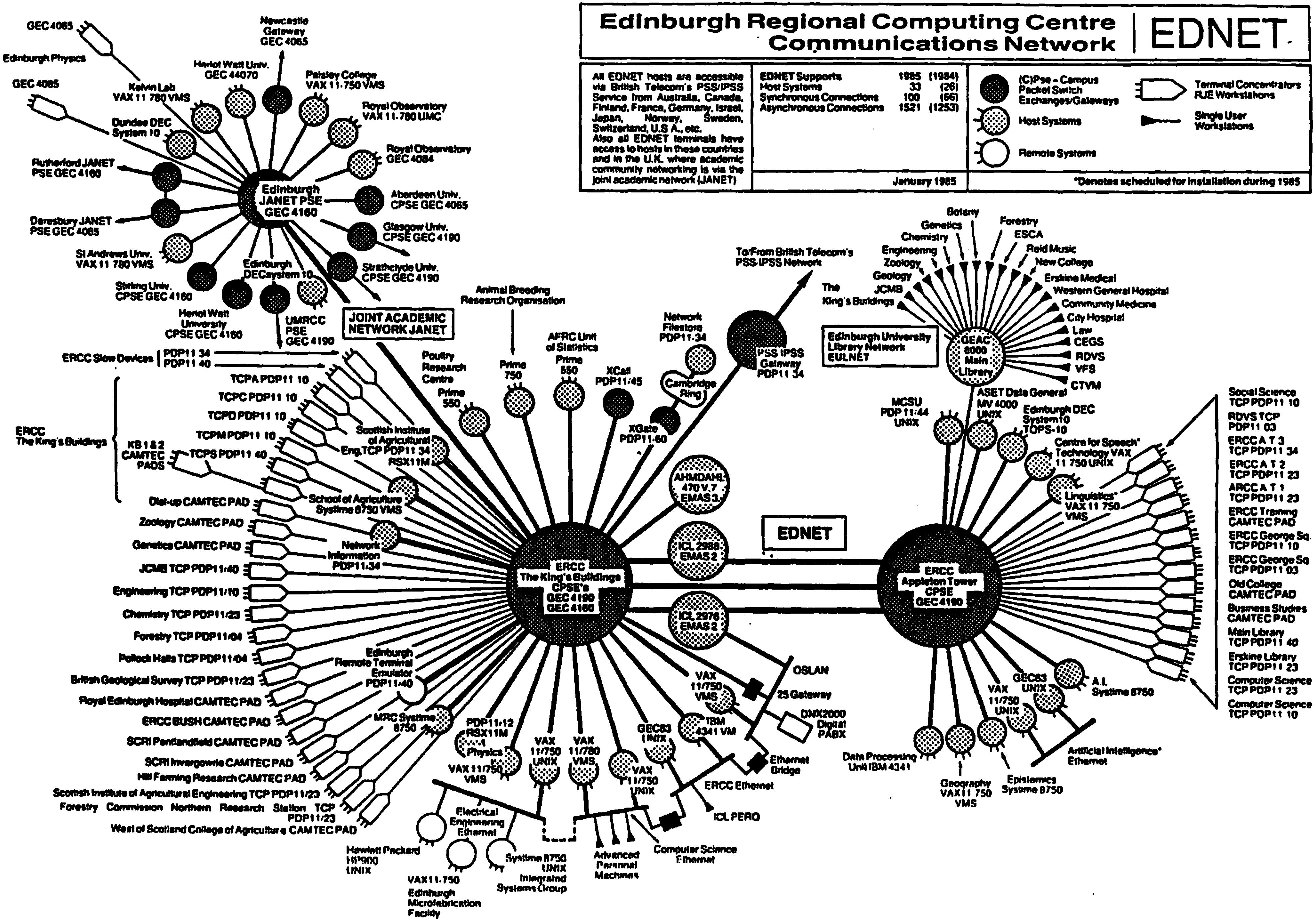
The messaging environment which provided the experience on which this paper is based lies within the services provided by the Edinburgh Regional Computing Centre. The client list includes Edinburgh University staff and students, research institutes, and numerous commercial organisations. The main computing facility is the Edinburgh Multi-Access System, EMAS, [1] which serves a community of around 5000 users. A wide-area packet switched network EDNET, supports over 1500 asynchronous connections to link users to the central services and to around 30 departmental machines. EDNET is connected to the Joint Academic Network, JANET, which links the major UK academic institutions, and to the British Telecom international packet switched service PSS/IPSS (Figure 1).

When the Computer Based Message System (CBMS) service was introduced on EMAS, it was required to conform to the operating principles and methodology of the system [2]. Hence the CBMS had to offer high availability; be self-regulating, to function with a minimum of operator intervention; be secure, to protect itself and its users from misuse; ensure reliable delivery of messages without loss or corruption; and supply a set of management tools, sufficient to regulate its use.

Given a computing environment with many users of differing experience, regularity of access and expectations of the service, the view was taken that the CBMS should offer message management services beyond the point of arrival of a message at its destination host system, and should continue this management until the acceptance of the message by its recipient was completed. In undertaking this additional responsibility, the CBMS acquires additional information on message traffic sufficient for it to implement mechanisms to protect and regulate the system.

Standards

The messaging standard in use within EDNET is the JNT Mail Protocol [3], an interim standard



Edinburgh Regional Computing Centre Communications Network | EDNET

All EDNET hosts are accessible via British Telecom's PSS/IPSS Service from Australia, Canada, Finland, France, Germany, Israel, Japan, Norway, Sweden, Switzerland, U.S A., etc. Also all EDNET terminals have access to hosts in these countries and in the U.K. where academic community networking is via the joint academic network (JANET)

EDNET Supports	1985	(1984)
Host Systems	33	(26)
Synchronous Connections	100	(66)
Asynchronous Connections	1521	(1253)

January 1985

*Denotes scheduled for installation during 1985

(C)Pse – Campus Packet Switch Exchanges/Gateways
Host Systems
Remote Systems
Terminal Concentrators RJE Workstations
Single User Workstations
GEC 4065
Edinburgh Physics
GEC 4065
Kelvin Lab VAX 11 780 VMS
Dundee DEC System 10
Heriot Watt Univ. GEC 44070
Newcastle Gateway GEC 4065
Paisley College VAX 11.750 VMS
Royal Observatory VAX 11.780 UMC
Royal Observatory GEC 4084
Aberdeen Univ. CPSE GEC 4065
Glasgow Univ. CPSE GEC 4190
Strathclyde Univ. CPSE GEC 4190
Edinburgh JANET PSE GEC 4160
Rutherford JANET PSE GEC 4160
Daresbury JANET PSE GEC 4065
St Andrews Univ. VAX 11 780 VMS
Stirling Univ. CPSE GEC 4160
Heriot Watt University CPSE GEC 4160
Edinburgh DECsystem 10
UMRCC PSE GEC 4190
JOINT ACADEMIC NETWORK JANET
ERCC Slow Devices
PDP11 34
PDP11 40
ERCC The King's Buildings
TCPA PDP11 10
TCPC PDP11 10
TCPD PDP11 10
TCPM PDP11 10
KB 1 & 2 CAMTEC PADS
TCPS PDP11 40
Scottish Institute of Agricultural Eng.TCP PDP11 34 RSX11M
School of Agriculture Systime 8750 VMS
Dial-up CAMTEC PAD
Zoology CAMTEC PAD
Genetics CAMTEC PAD
Network Information PDP11:34
JCMB TCP PDP11/40
Engineering TCP PDP11/10
Chemistry TCP PDP11/23
Forestry TCP PDP11/04
Pollock Halls TCP PDP11/04
British Geological Survey TCP PDP11/23
Royal Edinburgh Hospital CAMTEC PAD
ERCC BUSH CAMTEC PAD
SCRI Pentlandfield CAMTEC PAD
SCRI Invergowrie CAMTEC PAD
Hill Farming Research CAMTEC PAD
Scottish Institute of Agricultural Engineering TCP PDP11/23
Forestry Commission Northern Research Station TCP PDP11/23
West of Scotland College of Agriculture CAMTEC PAD
Edinburgh Remote Terminal Emulator PDP11/40
MRC Systime 8750
Poultry Research Centre Prime 550
Animal Breeding Research Organisation
Prime 750
AFRC Unit of Statistics Prime 550
XCall PDP11/45
Network Filestore PDP11:34
Cambridge Ring
XGate PDP11-60
To/From British Telecom's PSS IPSS Network
PSS IPSS Gateway PDP11 34
AHMDAHL 470 V.7 EMAS 3
ICL 2988 EMAS 2
ICL 2976 EMAS 2
EDNET
ERCC The King's Buildings CPSE's GEC 4190 GEC 4160
ERCC Appleton Tower CPSE GEC 4190
PDP11/12 RSX11M
Physics
VAX 11/750 VMS
VAX 11/750 UNIX
VAX 11/780 VMS
VAX 11/750 UNIX
GEC63 UNIX
IBM 4341 VM
VAX 11/750 VMS
OSLAN
25 Gateway
DNX2000 Digital PABX
Ethernet Bridge
ERCC Ethernet
ICL PERQ
Computer Science Ethernet
Advanced Personal Machines
Electrical Engineering Ethernet
Hewlett Packard HP900 UNIX
VAX11.750 Edinburgh Microfabrication Facility
Systime 8750 UNIX Integrated Systems Group
Botany
Genetics
Chemistry
Engineering
Zoology
Geology
JCMB
The King's Buildings
Forestry
ESCA
Reid Music
New College
Erskine Medical
Western General Hospital
Community Medicine
City Hospital
Law
CEGS
RDVS
VFS
CTVM
GEAC 8000 Main Library
Edinburgh University Library Network EULNET
MCSU PDP 11:44 UNIX
ASET Data General MV 4000 UNIX
Edinburgh DEC System10 TOPS-10
Centre for Speech* Technology VAX 11 750 UNIX
Linguistics* VAX 11 750 VMS
Social Science TCP PDP11 10
RDVS TCP PDP11 03
ERCC A T 3 TCP PDP11 34
ERCC A T 2 TCP PDP11 23
ARCC A T 1 TCP PDP11 23
ERCC Training CAMTEC PAD
ERCC George Sq. TCP PDP11 10
ERCC George Sq. TCP PDP11 03
Old College CAMTEC PAD
Business Studies CAMTEC PAD
Main Library TCP PDP11 40
Erskine Library TCP PDP11 23
Computer Science TCP PDP11 23
Computer Science TCP PDP11 10
A.I. Systime 8750
GEC63 UNIX
VAX 11/750 UNIX
Artificial Intelligence* Ethernet
Epistemics Systime 8750
Geography VAX11 750 VMS
Data Processing Unit IBM 4341

Figure 1. EDNET

which will be replaced by the CCITT X.400 Recommendations for Message Handling Systems [4]. The model originally proposed by IFIP WG 6.5, and developed within these CCITT recommendations is briefly described.

The CBMS comprises two classes of component, User Agents (UAs) and Message Transfer Agents (MTAs). UAs give the user access to the Message Transfer System (MTS) and offer local services for message preparation, storage and retrieval. The MTAs cooperate to convey a message through the MTS from an originator UA to one or more recipient UAs. The principal interactions between UA and MTA are the submission interaction where an originator UA requests the MTS to undertake responsibility for conveying a message to one or more recipients, and the delivery interaction where a destination MTA transfers a message to a recipient UA.

Delivery policy

The X.400 recommendations provide as part of the basic service to UAs, the ability to control the manner in which messages are delivered. The Hold for Delivery service specifies the encoded information types of the messages the MTA may deliver, the length of the longest message content, the minimum acceptable priority, and whether or not the MTA may currently deliver messages at all. These settings may be temporarily varied by the UA by means of a CONTROL request or permanently changed by a REGISTER request.

The provision of these facilities is in recognition of the needs of stand-alone UAs. However it is argued here, that even for UAs co-resident with the MTA, a default registration of Hold for Delivery (no messages deliverable) is appropriate. The justification is that message management policies can thereby be adopted which maintain control over the messaging system, to improve its reliability, security and functionality.

Self-protection

Where an MTA is operating as a relay between two remote MTAs and the source MTA operates at a faster rate than the sink MTA, then eventually congestion of the messaging system will occur. This congestion might take the form of exhaustion of MTA filespace, or exhaustion of some other database management resource. The result is that the throughput of the messaging system suffers in a way equally apparent to all its users.

To manage this problem, the system must adopt some form of flow control to apply back-pressure to a remote MTA which is responsible for consuming an excessive amount of local resource. When an incoming message is processed, its resource use is calculated, based on message size and the number of local recipients and remote MTAs to which it is to be relayed; this is decremented from the allocation granted to the remote MTA. When the allocation is exhausted, no further transfers are accepted from the MTA. As messages are disposed of, by local delivery and by completion of onward relaying, the allocation will eventually go back into credit and communication is resumed.

This method does not adequately deal with the case where a message is received from an MTA which itself has relayed it. Here we would like to apply back-pressure to the originating MTA rather than to the last relay, as this inhibits both the excessive and the well-behaved traffic from the relay. However, if the last relay is not able to make this distinction itself, then communication with it must be suspended.

It is noted that it is more difficult to evaluate the cost of handling a message when distribution list expansion is performed outside the MTS. Under this method, a message is delivered to a special-purpose UA which evaluates the distribution list name, and resubmits the message with the expanded set of recipient names, for onward processing [5]. The two-stage handling effectively masks the true cost of processing the message. Where distribution list expansion is performed within the MTS, no such problem arises.

The messaging system should also protect itself from locally-generated messages which may appear in large numbers through accidental or deliberate user action or through the misbehaviour of an agent which automatically generates messages. A similar solution applies: a record of the current resource use of each registered user is maintained. Once the user's allocation is exhausted, no further message submission interactions are permitted; when enough previously submitted messages leave the local system by relaying or by local delivery, then submission is again enabled. A simple variant of this has been used to curb the activities of undergraduate users, attempting to flood the system with messages. This involves imposing a delay, initially very small, each time message submission is attempted. The delay period is doubled each submission, effectively limiting the number of submissions that can be made each session.

It is important that these mechanisms be seen as operating only in exceptional circumstances, sufficiently serious to warrant alerting local site personnel. The allocations are set to such generous limits that in normal operation the flow control mechanisms do not come into effect.

Another aspect of this problem is where the messaging system while protecting itself against aberrant behaviour, fails to protect user filespace to which it has privileged access. If the delivery strategy is for unrestricted delivery of messages to filespace owned by and charged to UAs, then one user could set out to fill the allocation of every other user in the system. By contrast, where the messaging system offers Hold for Delivery, bearing the cost of this as an acceptable overhead, then the rest of the facility is protected from any potential misbehaviour.

Unaccepted messages

Where the users of a messaging service have individually subscribed to it, then it could be expected that they would activate their UAs with reasonable regularity; there is no particular problem associated with unaccepted messages. However, where a computing facility introduces a messaging service, and its existing users are perforce made potential recipients of messages, it is apparent that not all users visit the facility with the regularity necessary to make communication by CBMS productive. There may be a large number of registered users who access the system infrequently or at irregular intervals. On EMAS, only about half of the registered users access the system each month, although only 2% fail to access it within a year. A report on efficiency within the University concluded that E.R.C.C. did offer a fast, cheap and efficient electronic mail service, but that this was a neglected administrative tool, and would not reach its full potential until the majority of administrators became regular users [6]. For reasons of inconvenient access to network terminals, or perhaps simple conservatism, a proportion of users will not access the messaging system regularly, and the accumulation of a pool of unaccepted messages will result.

If the MTA performs unrestricted delivery of messages to recipient UAs, then the problem is ignored. The recipient may access a delivered message eventually, long after it has ceased to be relevant, but the originator is not made aware of this disparity. Alternatively, where the MTA retains responsibility for a message until a UA initiates delivery, it becomes possible to inform an originator when his attempt to communicate with a recipient has not succeeded within a locally determined period.

The existence of a Deliver By service element within X.400 would be one way of allowing the originator to make his intentions clear; the Expiry Date service element of the Interpersonal Messaging Service (IPM) is not suitable for this purpose. For the recipient, an additional parameter to the REGISTER request, to vary the time-out period which determines non-delivery, would permit him to inform the messaging system that his messages should be held for an extended period. The typical absentee user would have his messages declared undeliverable after the default time-out period.

A UA will occasionally be unable to accept delivery of a message where this would cause some local resource allocation (filespace) to be exceeded. In these circumstances, the MTA has two options. Where its policy is to divest itself of responsibility for a message at the earliest opportunity, by delivering it unconditionally, it abandons delivery and return a non-delivery notification to the message originator. Alternatively, where the MTA supports the Hold for Delivery service, the recipient is given the opportunity to acquire the resource necessary for delivery to be performed. Given that considerable cost may have been incurred within the MTS in conveying the message to the recipient, it is desirable that particular effort be made to enable its final delivery.

Delivery notification

In the X.400 recommendations, delivery notification provides a message originator with the means to request a positive confirmation that his message was delivered. Non-delivery notification is generated for any message that the MTS is unable to relay to its destination MTA, or unable to deliver at its destination, perhaps because of an error in the specification of a recipient name.

The circumstances under which delivery notifications are generated vary according to whether or not the recipient's UA employs the Hold for Delivery service. Where an MTA abandons its attempt to deliver a held message after some period, it issues a non-delivery notification indicating that the recipient UA is unavailable. If Hold for Delivery is not in force, then delivery is performed as soon as the message is received and non-delivery notification is not generated, even though the recipient may never subsequently access his UA. In neither case has communication successfully taken place between the originator and recipient, but only in the former case is the originator informed of this.

If a message for which delivery notification was requested is successfully delivered, then again different events are involved. In the held case, the notification is generated as a result of the recipient UA issuing the appropriate control request to enable delivery to take place. In the latter case, the notification results from error-free delivery by autonomous MTA action; the notification is issued even though the recipient UA may never be accessed by the user.

The X.400 recommendations state that no implication of any user or UA actions can be placed on delivery notification. However, these notifications, positive or negative, convey more accurate information when generated in an environment where Hold for Delivery is the default registration. Table 1 shows the delivery notifications generated under each regime, (positive, negative or none) where a user fails to access his UA within some locally-agreed time.

	Held	Not held
Delivery notification requested	-ve	+ve
Delivery notification not requested	-ve	0

Table 1. Delivery Notifications for unaccepted messages.

A receipt notification facility is offered by the Interpersonal Messaging Service, which operates at a higher protocol level than the delivery notification service. The use of this UA-UA facility is limited in that it is not mandatory within X.400 and some Administrations may choose not to support it. Hence a recipient UA may delete a message before the user has read it without generating a non-receipt notification even though the originator requested such notification. Also, a recipient UA might not generate receipt notification where this was requested, either because the facility is not locally supported or because the user declined to authorise such action.

Backup

The backup procedure copies message files currently resident in the on-line disc store to magnetic tape. Its purpose is to ensure that the messaging system is protected against loss or corruption due to hardware or software failure. A further use is to recover messages inadvertently destroyed by recipient users.

The EMAS messaging system uses the backup facilities provided by a system executive process which performs backup and archive services for all users of the filestore [7]. In practice, new message files and associated control information are backed-up daily and a complete backup of all files is performed weekly. Should a disc be lost, its contents can be restored to their state on the previous daily backup. However, those messages received by the system since the last daily backup and not yet disposed of are irretrievably lost. If the log file describing the messaging activity since the last backup is available, it is possible to generate minimal non-delivery notifications. That duplicate messages can occur is noted, but not considered a serious drawback.

The backup tapes are reused on a four week cycle. For recovery from system errors a period of one week would be sufficient, but to recover message files lost through user error, the additional backup capacity is occasionally useful. The ability to perform incremental backup of each message as it enters the messaging system would be desirable, but is not practical in a general purpose computing facility with many demands on its resources.

Message-space management

Having taken pains to ensure that messages are never lost, it may come as a surprise to the implementor of a messaging system to find he needs to take steps to destroy messages. Garbage collection is performed on several categories of message.

Unaccepted messages which are themselves delivery reports or status reports conveying IPM message receipt are deleted without notice. Unaccepted messages generated by other system processes to report completion of certain user-initiated activities such as file transfer or job execution are also deleted (the messaging system is especially useful here, in conveying reports of events in one part of the network to a user attached at a different point). Large numbers of messages may be generated in this way, of only transient interest. By declaring these messages undeliverable, after a period shorter than the period allowed for normal messages, saves message-space and frees recipients from a certain proportion of noise messages. At present, these messages are recognised in an ad hoc fashion. Under X.400 it may be possible to use the Expiry Date service element, optionally associated with a message, as grounds for deletion without notice. Strictly speaking, Expiry Date is a UA-level service element and so perhaps not suitable for this purpose. If such a message contained a non-receipt notification request, for example, it would not be correct to delete it without notice.

On a large University facility, several thousand undergraduate users are registered and deregistered at the beginning and end of the academic year. On deregistration, any messages queued for these users are declared undeliverable and non-delivery reports are generated for them. The process is repeated, so that any of these reports now queued for similarly deregistered users are declared to be dead letters and queued for the alternate recipient (the site postmaster).

Without these mechanisms, the messaging system would accumulate a large number of messages which in practical terms are undeliverable, or if delivery were eventually performed, of little interest to the recipient. In all cases where such messages are removed from the system, they are in fact requeued for the alternate recipient of the site rather than simply deleted.

Message-space structure

An MTA supporting Hold for Delivery will implement a more elaborate database structure than that required under a simpler delivery strategy. The model adopted for the EMAS messaging system borrows from the familiar concept of a spooler [8]. Messages are ordered into a series of message queues. Each queue is associated either with a locally registered recipient name, with a local MTA service agent (such as the deferred delivery agent), or with the name of a remote MTA.

These names are collected together as entries in a service directory, each entry acting as a queue head for its message queue.

A message queue consists of a linked list of message descriptors. The allocation of message descriptors occurs at message submission or on receipt of a relayed message from a remote MTA. Once validation of a message is complete, a series of descriptors are allocated:

- a head descriptor, containing submission details for a locally submitted message, or trace details for a remotely generated message. A copy of the message file is associated with this descriptor.
- a series of local recipient descriptors, one for each local recipient of the message.
- a series of remote MTA descriptors, one for each MTA to which the message is to be transmitted.

This block of descriptors defines, at any instant, the disposition of the message within the local system: a record of its origin, those local recipients who have accepted the messages and those still to do so, those remote MTAs to which the message has been transferred and those for which transfer is still pending. Each of the local recipient and remote MTA descriptors are, in addition, linked to message queues which are attached to corresponding entries in the service directory.

This organisation has a number of advantages:

- descriptors remain unused for a period after they have been deallocated, and provide a record of recent history within the messaging system.
- for a message with more than one local recipient, only a single copy of the message file is held.
- after a system failure, the queues can be completely rebuilt from the message descriptors.

As well as providing on-line information on message disposition, the set of message descriptor blocks could also be used to support possible future services such as Delivery Enquiry and Delivery Cancellation.

Conclusions

By undertaking to retain responsibility for unaccepted messages within the MTA, a messaging system acquires sufficient information to adopt management policies directed towards keeping the system under control. Failure to implement these protection mechanisms compromises the integrity of the messaging system and may compromise the general filestore management of the facility hosting it. The essential features are flow control, resource allocation, backup, and garbage collection. An important side-effect is that the quality of delivery notifications and non-delivery notifications is improved. The supporting structure lends itself to the provision of accurate state information for any message passing through the messaging system.

References

1. Whitfield, H. and A.S. Wight, "EMAS - The Edinburgh Multi-Access System", Computer Journal, 18, 331-346, 1973

2. Murison, J.M. (editor), "EMAS 2900: Concepts", E.R.C.C., 1978

3. Kille, S.E. (editor), "JNT Mail Protocol (revision 1.0)", Joint Network Team, Rutherford Appleton Laboratory, 1984

4. CCITT 5/VII, Message Handling Systems: System Model - Service Elements, Recommendation X.400, November 1983

5. Deutsch, D., "Implementing Distribution Lists in Computer Based Message Systems", IFIP WG 6.5 Conf., Nottingham, May 1984

6. Elliot, N.R., "Report on the University of Edinburgh - Jarratt Efficiency Enquiry", University of Edinburgh Bulletin, April 1985

7. Wight, A.S., "The EMAS Archiving Program", Computer Journal, 18, 131-134, 1975

8. Laing, W., "EMAS 2900 Spooler Notes", E.R.C.C., 1978