The (Un)Revised OSI Reference Model

by
John Day
BBN
10 Moulton St
Cambridge, MA 02139
day@bbn.com

1. Introduction

In 1988, SC21, the ISO committee responsible for the Open Systems Interconnection (OSI) Reference Model,[1] determined that it was time to undertake revising ISO 7498-1, The Basic OSI Reference Model. Since the model had been published in 1984, this was in accordance with ISO practice of reviewing and revising standards every five years, and there was good reason for considering the task. Over the intervening five years, the groups developing the OSI protocols had raised many questions about the architecture and these had been answered and documented in the Approved Commentaries [1994]. Many of these commentaries contained information that could be usefully incorporated into the Reference Model itself. In addition, the addendum describing connectionless mode, i.e. datagrams, had been completed several years before and needed to be incorporated. There was also considerable demand for interworking connection-mode and connectionless mode communication, something not supported by the Reference Model or any architecture. Also, when the original version of the Reference Model was frozen about 1983 some aspects of OSI such as the upper layers were not well understood and were only described in the most cursory manner. And while connection-mode and connectionless mode had been brought within OSI, there was no indication as to how broadcast and multicast were to be handled. Thus, the revision might be able to provide a more comprehensive description of these areas.

This paper describes how the revision was carried out, describes the changes and additions that were made, considers the effect and contribution this revision has made to our understanding, and describes the outstanding issues that were not addressed by this revision.

1. ISO is organized into Technical Committee's for major areas, like communications, wine glesses, nuclear reactors, safety signs, etc. which are broken up into one or more Study Committees (SC) which in turn might be broken up into one or more Working Groups (WGs); very similar to the organization of the IETF which would correspond to a TC, an SC to an Area, and a WG to Working Group.

Why Have a Reference Model?

Many of the issues here belie the fact that different people have different and not always compatible uses for a reference model or an architecture. This raises the question: if it causes so much trouble, why bother? Generally, four arguments can be made for developing an architecture:

Coordination. When developing a set of protocols intended to work together, it is useful to have a set of principles and conventions to ensure that the protocols will work together, to avoid duplication of functionality, to foster consistency among the specifications, and to allow parallel development.

Simplicity. Often if one works out the structure of a problem, one can find a solution that is much simpler than may first appear. This is the "potential well" phenomena, i.e. the best solution is not found by the path of least resistance. A good example is routing architecture. The Internet has generated 6 or more protocols for routing by solving problems as they appeared while OSI worked out a general architecture using the Internet experience and found that only 3 were required and the same theory solved several related problems as a consequence. Working out the architecture defines a module structure that will facilitate evolution and change.

Economic/Political. Technical solutions are not economically or politically neutral. No matter what solutions are chosen they will favor some political or economic position. For example, many of the debates in OSI were not so much technical as determined by where market boundaries would be drawn between carriers and computer vendors, or to create market opportunities for European manufacturers with respect to US or Japanese competition. This use may work for a while, but "nature" has a tendency to "assert" itself. For example, the European PTTs (and their governments) have tried to use the OSI Reference Model to legislate a particular technological direction based on a connection mode X.25 view of communications, which has been rejected by the market. (On the other hand, it is also the case that the connectionless approach represented by the Internet would not have been successful without substantial government subsidy as well.)

Educational. One would hope that research and experience has shown that there are certain principles in the design of communication protocols. While perhaps not as succinct as those of Newton or Maxwell, embodying these in a Reference Model can help guide the development and ensure consistency and facilitate creating a comprehensive whole.

Note that none of these require a standard.

Several of the issues described here show that the US and Europe had distinctly different ideas as to the purpose the Reference Model served. The Europeans, in the wake of the Nora-Minc Report [1977], the Alvey Report [1982] and similar initiatives by governments and the European community to drive information technology business, saw the Reference Model as the means to legislate a technical solution and a homogeneous computer and communications market for European companies.[1] The belief that the direction that market systems take can be legislated, ignoring internal forces acting on these systems, seems to be one peculiar to Europe.[2] Europeans attempted to use the Reference Model to legislate a particular technological solution on a broad and general problem. Europeans remain convinced that they can legislate technical solutions or simply ignore them and no one will develop product in that direction. Much of this can probably be attributed to the role of government in both standards and the PTTs and the relative weakness of the European computer industry.

North Americans tended to see the Reference Model as a place to record principles for network architectures, as it would seem do most people who approach the document for the first time. In some sense, the document should record the general principles for constructing interacting sets of protocols gleaned from research and experience. These principles could then be applied to the development of a set of standard protocols. A Reference Model should represent an ever improving goal that indicates where new work is headed, incorporates new understanding and not be concerned that it does not bless previous practice. It would certainly be unusual if as our understanding improved we did not learn that some of the solutions of the past were wrong or that better solutions did not exist or there were better ways of thinking about the problem. It should be assumed that some principles would apply to all layers and others would be specific to individual layers. Hence, the view that the document should not be a standard, so that it could be easily modified and added to. The lack of government participation in standards and the more "hands-off" role of government in communications even before deregulation and the strong computer industry did not allow one industry segment to dictate its positions to the standards committees. While the North American behavior may appear more aimed at seeking the "right" answer, this is only because the contending forces were more balanced and not for any altruistic reasons.

1. Remember all of that talk in the 80's about "level playing fields."
2. As we have seen recently, a similar attitude toward economic systems can be seen in the behavior of the EEC in other areas with similar results.

2. How It Was Done

The revision of the OSI model was approached very carefully. The Europeans[2] particularly did not want to see any major changes made. They argued strongly for stability.

The OSI Layers

Application: All applications such as File Transfer, Network Management, etc. reside here.
Presentation: Negotiates the transfer syntax to be used by the application. It does not do data compression or encryption as claimed by many books.
Session: Provides primitive functions for coordinating dialogs of an application. Note that the Session Layer has nothing to do with creating or maintaining Sessions as believed by many.
Transport: Provides end-to-end reliability and flow control for an application.
Network: Provides network routing and relaying.
Data Link: Provides sufficient error control over the media to make Transport error control cost effective. Has the same scope as the Physical Layer and generally less scope than the layers above.
Physical: That messy electrical stuff.

On the other hand, there was a desire to give the groups working on major areas some freedom to revise those parts of the Reference Model as they saw fit. To that end, the task was divided in three parts:

- WG1 which was responsible for the reference model as a whole, would revise clauses 1 to 6, the general principles on which the seven layer model was built.
- WG6, which was responsible for the upper layer architecture, was given the task of revising clauses 7.1 to 7.3, the descriptions of the Application, Presentation, and Session Layers.
- SC6, the committee responsible for the lower layers, was given the task of revising clauses 7.4 to 7.7, the descriptions of Transport, Network, Data Link, and Physical Layers.

To answer the questions on the Reference Model and assure that there was a consensus on the answers, WG1 had put in place a procedure for balloting the answers to questions that required a National Body vote on a draft answer and on a final answer. This procedure was to assure that there was a consensus on the interpretation of the Reference Model. This was mostly due to the contentious nature of many of the questions and the delicate wording that existed in the Reference Model to address many previously contentious issues. Allowing National Bodies to vote on the text of the questions ensured that no one was able to slip something outrageous into the Reference Model. To address the issue of stability, WG1 decided that for the parts of the revision for which it was responsible, the only changes that would be made to the Reference Model during the revision would be those which were either already approved commentaries or were approved during the process of revising the Reference Model. While this procedure works well for precise, tightly focused questions, it also guarantees that major issues that might lead to major changes would not be addressed.[3] However,

2. Throughout this paper we will refer to the “Europeans” as representing all Europeans since this was the role the European representatives to ISO committees serve. While there are experts, perhaps even a majority, in Europe who would disagree with the positions that were taken, their views were not represented in the deliberations, and thus in the tradition of representative democracy their views must be considered “the views of the Europeans.”
3. While it was continually stressed that the Reference Model was and should be a working document that recorded the principles that had been learned, as everyone knows, a sizable segment of the industry and those involved in the discussions soon treated it as a “bible” and inviolate, not unlike the adherence to the works of antiquity during the 16th Century. This ten-

WG6 and SC6 were given carte blanche to modify their sections in any way they saw fit and to send the approved revision to WG1 for incorporation into the revision.

To initiate the revision the Rapporteur[4] for the OSI Reference Model met with both WG6 and SC6 to outline the tasks that were being given to them and to ensure that both groups understood these tasks and that they were in agreement with the procedures for creating the revised text. The Rapporteur went through the approved commentaries up to the time the revision work began and made recommendations as to which ones contained material that might be useful to all three groups working on the revision and made this information available to WG1, WG6, and SC6. For the clauses that WG1 was responsible for, the Rapporteur proposed changes based on the material in the commentaries. The results of this compilation were produced both in the form of ballot comments (change pages) and as a marked up copy of the Reference Model. This material was then submitted to WG1 for approval.

This material was reviewed by National Bodies, comments were incorporated, and a new draft of the revised Reference Model was produced. In parallel, a number of questions were also progressed to a final answer and included in the revision. Let us briefly consider what the revised reference model contains.

3. What's New in the New Edition

3.1. Integration of the Connectionless Addendum

At the outset the most difficult task was the integration of the connectionless addendum. The inclusion of connectionless data transfer in the OSI Reference Model had been a very contentious and hotly argued topic. The Europeans in keeping with their position that the X.25 connection-oriented service provided everything required in the Network Layer[5] argued strongly that connectionless mode was a useless concept and, if allowed at all, should be tightly constrained. The US, on the other hand, believed that connectionless mode was absolutely essential to any set of communication standards and certainly to any useful Reference Model.

In 1980 as the Reference Model was being prepared for progression to DP,[6] there was a compromise struck to get the model out quickly. The initial version would include only connection mode communication and connectionless mode would be the subject of an addendum. The US only agreed to this compromise on the condition that the New Work

---

dency is not restricted to the participants in the OSI work.

4. Given that the Rapporteur is the author of this paper, a note is necessary to explain the task of a Rapporteur. That task is to facilitate arriving at a consensus. The Rapporteur does not determine the technical direction of the work; that is done by National Body contributions. The Rapporteur can indicate flaws and *try* to ensure the quality of the work, but can not dictate the direction. Thus, the opinions expressed in this paper are those of the author as a member of the technical community, not those of the Rapporteur for the OSI Reference Model. The Rapporteur may have entirely different opinions.
5. And the Transport Layer for that matter.
6. Draft Proposed standard. ISO standards are balloted at two levels. The DP is recommended when the Working Group is satisfied with the content for ballot by the Subcommittee as a whole. Once it is approved at this level it is then balloted as a Draft International Standard (IS) in a wider context by the Technical Committee.

Item for connectionless mode be approved at the same meeting. After three years of hotly contested arguments, the connectionless addendum was approved for its first ballot.[7] The addendum was drafted as a free-standing document independent from the Basic Reference Model itself. The decision to address connection mode first and connectionless as an addendum and as a separate document caused difficulty in integrating it into the Reference Model and also created the first of several major technical flaws in the Model.

The combination of reticence for wholesale change to the Model and the self-standing document made the Editor's task of merging the two a very touchy affair. Both the Model and the addendum contain text that has been very carefully drafted. Every phrase, word, and comma had special significance to someone. It was also clear that much of the text in the connectionless addendum that provided a rationale for its creation would have no place in a merged document. After considerable work and editorial gerrymandering, the Rapporteur prepared a proposal for incorporating the connectionless addendum into the body of the Reference Model that introduced no technical change in the text.

3.2. Incorporation of the Approved Commentaries

3.2.1. Definition of End System and Relay System – It was found useful especially in other standards to distinguish systems acting as hosts from systems acting as routers. An end system is defined to be a system which is the ultimate source or destination of data. A relay system is one which is not the ultimate source or destination of data. Relaying is allowed below the Transport Layer and in the Application Layer. It is important to note that the use of these terms is relative to "an instance of communication." For example, a router would generally be thought of as a relay system that relayed at layer 3. But for network management purposes, when a management application communicates with the Agent (an application) in the router, it is an end system. Most of the time the router will be considered a relay system, but once in a while it is an end system. Similar situations can arise for systems that are generally considered end systems or hosts.

3.2.2. No Transport Relays – In some sense, the debates in OSI began with the necessity of an end-to-end transport layer. The European PTTs insisted that X.25 was reliable in the face of both theoretical and practical results to the contrary. They also insisted that one did not need as robust a transport protocol over X.25 as over unreliable connectionless networks. (The argument was that one would use Class 2 over X.25 and Class 4 over the Connectionless Network Layer Protocol (CLNP).[8]) For connections that spanned both, one would do Transport Relays. The fact that it had been shown in the Network Layer that anytime relaying was introduced reliability could not be guaranteed, was somehow not supposed to apply to the Transport Layer. Once again theoretical and practical results showed that Transport Relays could not guarantee the Trans-

7. Even after it was agreed, the Europeans and PTTs argued that there should be no conversion between connectionless mode and connection mode. In fact, there was to be no "cross-over" at the network/transport boundary. At a crucial meeting in Ottawa in 1983, the US made it clear that if this cross-over was not allowed the US would walk out of OSI, and the Europeans grudgingly acquiesced a little.
8. There was also a belief that Class 2 would be more efficient than Class 4. While a Class 2 implementation is slightly smaller (the implementation effort is also only slightly smaller), the performance under the same error conditions are the same. The only difference is that there is a large class of errors from which Class 2 does not recover that Class 4 does.

port Service except under very constrained circumstances. Finally, it was agreed that relaying was prohibited in the Transport Layer and appropriate text was included in the Reference Model. Oddly enough, the usefulness of transport relays continues to be debated.

Classes of Transport

The OSI Transport Protocol defines 5 classes, essentially protocols with increasing functionality. The reasons can be seen as either technical or political depending on your point of view:

Class 0: Transport addressing or so that CCITT SGVIII could claim that Teletex was an OSI protocol.

Class 1: Error recovery from resets and errors explicitly signaled by the network, but no multiplexing or the protocol CCITT SGVII wanted to maximize the number of X.25 connections one would have.

Class 2: Class 1 with multiplexing and optional flow control, for the European computer companies that wanted to minimize X.25 connections, but thought X.25 was sufficiently reliable that they didn't need Class 4.

Class 3: The union of Classes 1 and 2 or the ECMA requirement for "robustness."

Class 4: Similar to TCP except with message sequencing, no graceful close, and less overhead. The only class you really need.

3.2.3. Inclusion of Type and Instance – Early in the history of the Reference Model it was found useful to distinguish types and instances. It was necessary to talk about concepts that applied to types of objects as well as concepts that only applied to instances of those objects. The intent was to adopt the classic definitions of type and instance found in mathematics or object-oriented programming and now used in the Open Distributed Processing work. However after a five year debate on the definition of type and instance(!), a strange three level structure for type and instance was finally agreed to and only applied to (N)-entities. *Entity-types* are descriptions that span the entire OSI Environment; *Entities* are instance of Entity-types within a particular subsystem, i.e., within a layer in one system; and *Entity-invocations* are instances of entities. After all this time was spent coming to this less than useful definition, no one had the energy to try to propagate the concepts throughout the model and in particular to identify which objects in the model are associated with which types of objects and which are associated with instances of objects. This has been especially unfortunate and could have greatly simplified the addressing architecture and would be especially necessary for a simple model of multicast. Had this been applied to the Reference Model, it would have become clear that primitive addresses must name instances of communication; and that the most common use of addresses, i.e. network addresses in either OSI or the Internet, are generic addresses in that they name a set of instances of communication.

3.2.4. Versions and Version Identification – The revised reference model includes new text covering the nature of versions and version identification of protocols. This work has been quite helpful in defining the conditions under which a new version is required: new version identification is required if the changes are sufficient to modify the behavior of the protocol state machine such that the changes can be detected by its peer.

3.2.5. New Definition of Address – In the course of developing Part 3 of the Reference Model on Naming and Addressing, a problem was encountered with the definition of (N)-connection. (N)-connections are defined as being from one (N+1)-entity to another.[9] For addressing, this created problems. Since connections are established between

9. In other words, the shared state between (N)-entities is an (N-1)-connection!?

(N)-entities, this would require that all addresses be allocated and bound all the way up the stack before an establishment attempt was made. The recipient system would not be able to assign addresses (and make resource allocation decisions) at establishment time.

The (N)-Notation

Clause 5 of the reference model develops a number of general concepts that are applicable to any layer. Often these concepts and principles require reference to the layer above or the layer below. Consequently, the (N) notation was adopted to refer to a layer in the Model, while (N+1) refers to the layer above the (N)-layer and (N-1) to the layer below the (N)-layer.

The group had a choice of changing the definition of (N)-connection which would have had far reaching effects on all parts of the Reference Model (many of which would have been helpful in resolving other problems) and other standards,[10] or changing the definition of (N)-address to finesse the problem and keeping the effects localized to the work on addressing. The group chose the latter route and defined (N)-SAP-address to be the name of a single (N)-Service-Access-Point and (N)-address to be a set of (N)-SAP-addresses. Thus at establishment, an (N)-address was sent in the protocol and the recipient could choose one element of the set to respond with, thus allowing the recipient some flexibility in associating addresses with resources to be allocated.

3.2.6. Expedited Data Transmission – Expedited mechanisms are found in X.25 and many of the early lower layer protocols. In these protocols, an expedited mechanism was provided to communicate a “small amount” of data outside the normal flow and with a higher priority. Expedited had been used to communicate a command to flush the data stream, a terminal break character, etc. In general, the method of “expediting” such data has been to place it at the head of every queue it encounters. In all cases, the mechanism provided a simple means to send a small amount of data. There was no desire to make the expedited “stream” have all of the properties of the normal stream with respect to segmentation, flow control, etc. If that were the case, it would be simpler to establish a connection with the appropriate quality of service. The mechanism had to remain simple. The use of expedited up to this time had involved only 2 layers. When it came to trying to apply these mechanisms in the general environment of a model with more than 2 layers, problems arose. With multiple levels of data transfer protocols, could one map (N+1)-expedited to an (N)-expedited? And what effects would it have?

Clearly, PDU encapsulation would require each expedited Protocol-Data-Unit (PDU)[11] in the lower layers to be larger to accommodate the upper layer expedited PDUs. Since the use of expedited is generated by the application, it was difficult to put a reasonable upper bound on what a “small amount” of data ought to be. Given that flow control was generally handled by allowing only one outstanding expedited PDU at a time, encapsulating would mean that sending an (N)-expedited PDU for one (N+1)-connection would inhibit sending an (N+1)-expedited on all other (N+1)-connections until a response was received for the first one. If the use of expedited by one connection was

10. While the change would have had a ripple effect on the text of many other standards, the effects were primarily cosmetic and would not affect the “bits on the wire.” If one listened to the discussions rather than what was written, (N)-connection was used as the shared state between (N)-entities. The definition was serving primarily to make the text less understandable.
11. OSI terminology for a message. One might find this terminology unnecessary, but it was found that so many people had specific ideas about what messages, packets, frames, etc. were that a completely different term was deemed necessary.

“less critical” and took longer, one could actually prevent an expedited from being sent that would alleviate a more critical situation at the receiver. Given these considerations and others and the requirement not to make the mechanism any more complex, it was recommended that expedited only be used within a single layer and that (N+1)-expedited not be mapped to (N)-expedited if multiplexing were present. Since the OSI Session and Transport Protocols ignore this recommendation, it is best not to use expedited at all.

As one can tell, the constraints on the expedited function are such to make it virtually useless. Relaxing any of the constraints, however, requires expedited to have most of the machinery required for a connection: fragmentation/reassembly, flow control, etc. This defeats the entire purpose of a simple mechanism for a small amount of data. Basically the result of this analysis is that expedited simply does not generalize to architectures where connection-mode multiplexing occurs in more than one layer.[12]

3.2.7. Orderly Release – Some early transport protocols had provided a graceful close mechanism, whereby if the sender had data queued for the receiver when the (N+1)-layer issued a release, the sender would not send a disconnect until all queued data had been sent. Orderly Release extended this by proposing that the sender could issue a disconnect and the receiver could refuse the request and keep the connection in the data transfer state. The question was raised whether either mechanism was beneficial and where in the architecture it should reside.

The US DoD argued (because of TCP) that they had to have this mechanism to support “security mechanisms.” However, since one cannot assume that a connection will always terminate gracefully (failures will occur), the successful operation of an upper layer mechanism, especially security mechanisms, could therefore not be predicated on the successful completion of a graceful close. Also, it was found that graceful close added considerable complexity to a transport protocol. Further, the application cannot be assured that all data was delivered or that some of the data that was delivered after the graceful close led to an error in the application which at that point has no means to notify its peer of the problem.[13] This problem is exacerbated in higher speed networks where considerable data can be “in-flight” when the close is issued, thus increasing the likelihood of an undetected error; so a graceful close is even less desirable in high speed networks or networks with high delay.

The only benefit of a graceful close is as a matter of convenience to the applications, and even then the application will never know if some unrecoverable error occurred after the release was issued. Consequently, it was decided that the benefits were not

12. TCP has a similar construct, i.e. Urgent, that does not generate a separate message, but is plagued with these problems and others that make it no more workable. It appears that the primary use of Urgent by current applications is to mark record boundaries (SDUs in OSI terms) to get around the stream nature of TCP.
13. A common argument against this view is that experience in the Internet indicates that such errors don’t happen that often. It is very difficult to confirm this claim. It may take days or even months to discover an error caused by a graceful close failure (making it difficult to trace the cause back to the graceful close), and few if any systems actually collect data on these sorts of failures, either in TCP or in the applications. It is virtually impossible to collect data on how often errors can be traced to a graceful close. So the only evidence that graceful close works is hearsay.

sufficient to burden any protocol below the Application Layer with such a facility. Applications should determine by an exchange of PDUs among themselves when they are finished and then disconnect. It is simply not prudent to do a graceful close, which can lead the user to believe that operations have occurred that did not; and worse, the fact something went wrong will not be discovered for some time. A common orderly release mechanism might be beneficial if defined as a common application module and used in those application protocols that need it, but to separate it from the application in a different layer merely creates unnecessary complexity and generates a false sense of confidence in the network.

3.2.8. Connection Establishment Through the Layers – In a layered model there are always questions of efficiency resulting from cascading multiple layers, especially in trying to limit the number of exchanges required to initiate communications. With a seven layer model, it could potentially require 7 or more round trips for a user to establish communication. The question arose, could it be done in fewer round trips by putting the connect request PDU of the (N)-layer in the user data field of the connect request PDU of the (N-1)-layer?

Actually, this question had been tackled by the research community in the mid 1970's in some detail [Belnes, 1974; Danthine, 1977; Spector, 1982]. Consequently, there was considerable analysis available about the conditions under which pathological conditions could arise or PDUs could be lost. First of all, it was noted that any single user would seldom if ever have to establish connections at the lower layers. Data Link connections will exist, if at all, from the time the network is made operational. Analysis showed that Transport connect PDUs could not be embedded in Network connect PDUs and similarly, Session connect PDUs could not be embedded in Transport connect PDUs without creating considerable additional implementation complexity for all of the recovery cases or creating the potential for error conditions that were either undetectable or unrecoverable by those layers.[14] However, no pathological conditions arise from embedding the establishment PDUs of the Application into the Presentation Layer PDUs and in turn into the Session Layer PDUs. So it is possible to reduce the number of establishment exchanges from 7 to 3. If a connectionless network layer is used, then only 2 are required.

3.2.9. Implicit Connection Establishment – The reference model now includes text to describe the modeling of permanent virtual circuits. The view taken by the Model is that these connections have an establishment phase just like normal (N)-connections; however, the establishment phase is performed by a means that is not in the scope of the layer. This may be some ad hoc mechanism, such as two humans on the telephone, a non-standard protocol, or network management. While this may seem only important to PTT networks, this extension (and the next one) show the generality of the basic concepts of an establishment and data transfer phase.

14. Also, constraints on maximum PDU size in Network and Transport protocols made it impossible to always be able to include a (N+1)-connect PDU in a (N)-connect PDU without fragmenting the (N+1)-connect PDU being encapsulated. This caused much of the supposed efficiency either in round-trips or processing to be lost and with the increased complexity in protocol implementation, not warranted.

3.2.10. Modeling Circuit Switched Networks – Out-of-band signalling networks, such as circuit-switched networks or Integrated Services Data Network (ISDN) are modeled in a similar vein. In this case, the protocol exchanges for setting up the connection are simply viewed as flowing on an (N-1)-connection with a different address than the (N-1)-connection used for data transfer. Clearly, the connections for the establishment protocol must be created by in-band mechanisms or by the implicit means described above. However, since the Reference Model is a general model and does not attempt to describe specific technologies; these results only manifest themselves as a "note" in the Network Layer description.

It should be noted that this model of out-of-band signalling differs considerably from the modelling of out-of-band signalling described by ISDN standards. The Reference Model approach simply applies the same model as for all lower layer protocols. The ISDN model has connection establishment occurring by sending the PDUs to establish communication on the B channels as exchanges on the D channel. ISDN models this establishment as occurring in the *Application Layer*.

There are two major reasons for the approach adopted by the Reference Model. Since connection establishment and release are considered a critical operation that must be accomplished even if the network is congested. Communication at the Application Layer is assumed to be non-critical. Application data may be subject to delay due to congestion in the network, but will be delivered. Since establishment/release will often entail the acquisition or release of the resources required to alleviate congestion, putting connection establishment in the Application Layer may lead to networks that deadlock or require special consideration to be given to certain traffic. Second, modeling establishment in the Network Layer greatly simplifies the modeling of interworking with other Network Layer protocols. For example, to interwork X.25 and ISDN, an X.25 connection establishment in the Network Layer is mapped to an ISDN connection establishment also in the network layer. The only difference is that in one protocol both establishment and data transfer are mapped to the same (N-1)-address, in the X.25 case, and are mapped to different (N-1)addresses in the ISDN case. For ISDN to model X.25/ISDN interworking, it is necessary to coordinate interactions between establishment mechanisms in the Network and Application Layers?! ISDN implementations that multiplex B and D channels on the same physical media may have problems under heavy load conditions and those that don't have much more invested in their physical plant than necessary.

3.2.11. Deletion of Interface Definitions – The original Reference Model contained a set of definitions for modelling interfaces, often referred to Application Program Interfaces (APIs) regardless of whether the interface is at the Application Layer. To avoid the implication that interfaces needed to exist at all seven layers, most of the standards have been done in terms of a higher level of abstraction based on a service boundary and service primitives. Interfaces, i.e. APIs, and protocols are at a level of abstraction between service definitions and implementations. Interfaces correspond to specific API specifications, while a service definition is an abstraction of an interface or API specification.[15] While the concepts of service primitive were widely used, the relation between service boundaries and service primitives, on the one hand, and interface definitions, on the other, was never made. Since (at the time) there was no work on APIs for OSI, for the revision it was determined to simply delete the interface definitions, rather than describe the relation between the two. Consequently, APIs are no longer modeled in the Reference Model, just at the point that there was industry interest in defining APIs.

The issue of interfaces and service definitions had been a contentious one since the beginning. In the early years, many companies, especially IBM, wanted no APIs specified. They believed that interface standards would put constraints on the internal system implementations, i.e. they would have to make interfaces available that they did not need or want. They wanted only protocol specifications. But many recognized the necessity of having something to hide the mechanisms of one protocol from another protocol or application operating on top of it. So, Service Definitions were invented as a higher level abstraction of interface definitions that only modeled the behavior seen across the layer boundary as a result of protocol actions. Local or implementation specific interactions, such as fragmentation/reassembly, flow control, status across the interface were not modeled. This gave an indication of what sorts of actions caused PDUs to be generated but left the implementor as much freedom as possible for the implementation strategy. Given the variety of implementation environments, this was a wise decision.

However, these precautions were insufficient. Many who did not take the time to understand the architecture argued strongly that APIs should contain only the calls in the service definition, no more or no less.[16] This, of course, is absurd: either is acceptable as long it supports what users of the API need. Some APIs may expose more functionality than the service definition (although this is rare); some may make the API simpler than the service definition. Similarly, others (often the same ones) insisted that every service definition should be a basis for testing. This became especially absurd in the upper layers and lead to the ultimate in the ridiculous, the layer-by-layer implementation of the upper layers.

3.2.12. Suspend/Resume – Another lower layer mechanism that was proposed was Suspend/Resume. There are applications that send data, have a long quiet period, and then send more data. It was considered to be advantageous if the application could tell the lower layers when these periods were starting so that network resources could be freed until they were needed. This not only allows more efficient use of network resources but also lowers the application's cost. The Reference Model now contains text on the

15. The abstraction of a service definition is invariant with respect to two properties of an API: First, interactions of a purely local nature are not represented. (This is why some useful aspects of an API, such as status or interface flow control, do not occur in service definitions.) Second, service definitions are language and operating system independent. Thus, a service definition is an abstract API and a service primitive is an abstract API system call or procedure.
16. Most of these discussions were held in the workshops which were generally held at overlapping times with the standards meetings, so that the people who had been involved in the development could not attend, not to mention the physical limitations on how many meetings one person can attend. There was also a bit of the attitude in the workshops that "real programmers" didn't need the architects to tell them how to implement. From the early implementation agreements produced, this was apparently not the case.

general nature of Suspend/Resume. While the Commentaries recommend that the function be in the Transport Layer, SC6 has never undertaken the work to introduce the functionality into Transport, and it was not part of the revision of the lower layers text.

3.2.13. QoS – The clauses describing the lower layers make considerable use of the term Quality of Service (QoS). However, there was nowhere in the Reference Model that the concept was described. Consequently, some attempt was made to introduce the concept with some text to act as a placeholder until more could be said. QoS remains a concept that everyone agrees must be accommodated in a meaningful and useful way, but no one seems to have any idea how it can be effectively done.

New work on QoS has been started but it is unclear whether this work will lead to any real progress in the area. While the work has developed a taxonomic structure for QoS, it is not clear how relevant such a structure is to implementation or solving real QoS problems. The work considers beyond its scope the fundamental question of how changes in QoS parameters are to be manifested as changes in protocol or system behavior. Until this issue is addressed, the work on QoS will be of only academic interest. It would appear that this problem can not be addressed without specifying considerably more about the details of the implementation, which runs counter to the idea that standards should not legislate implementations and that implementations of standards remain an area for competitiveness.

3.3. Incorporating WG6's Improvements in the Upper Layers

As we noted earlier, when OSI began, the architecture of the lower layers (Transport and below) were fairly well understood, but the nature of the upper layers was not so clear. When the seven layer model was chosen as the basis, it was assumed that it would be possible to work out what the structure should be and the upper three layers in the seven layer model seemed to provide a good starting point.[17] After considerable discussion, a picture began to emerge while the development of the upper layer protocols continued in parallel with the architecture work. It was determined that the Application Layer was concerned with the semantics of applications, while the Presentation Layer was concerned with the representation of application layer semantics, i.e. the syntax. The Session Layer was probably intended by the original author of the model to establish application sessions and to perform access control for the Application Layer. Unfortunately, the Session Layer was hijacked by CCITT for the synchronization primitives for Teletex and Videotex.[18] The creation of sessions and access control functions are provided by the Application Layer.[19]

However, about 1983 it became apparent that rigid structuring of the current upper layer architecture was causing numerous problems. SC21 set about to fix these problems so that the upper layer architecture could be completely revised in time for the revision of the

17. It is not at all clear how much the existence of the upper three layers influenced the outcome. While there were proposals during this period for fewer upper layers, none was very strong.
18. Teletex was the CCITT 1980s response to e-mail. Essentially, Telex with memory and rudimentary local editing. Videotex was an early CCITT attempt to offer information services, the French Minitel being a prime example. Several trials were done in the US in the 1980s, but predictably, did not meet an enthusiastic response.
19. A mistake often made in many textbooks whose authors apparently never bothered to read either the Model or the Session protocol to see what the Session Layer really did.

Reference Model. However, the attempt went astray and when it came time to revise the Reference Model, the revision of the upper layers had not been started. The revision of the description of the upper layers was reduced to bringing the description in the Reference Model into line with the emerging second edition of ISO 9545 (Application Layer Structure) and the current state of the Presentation and Session Layers. This in itself, was a fairly major improvement since in 1983 when the Reference Model text was frozen, the concepts for the upper layers were only just being formulated. Considering each layer in turn:

> *Application Layer* – In general, the Reference Model has attempted to describe the layers at the level of the service provided and the interactions and functions of the (N)-entities. Any structure more detailed than that was left to other standards, such as the Application Layer Structure (ISO 9545) or the Internal Organization of the Network Layer (ISO 8648). The revised text for the application layer maintains this convention and brings the text into line with ISO 9545 describing the generic functions provided for establishing application associations and managing abstract syntaxes.
>
> *Presentation Layer* – This text introduces the Presentation terms of abstract, concrete and transfer syntax and describes the function of the Presentation Layer as managing the negotiation of the transfer syntax. The abstract syntax is equivalent to the data representation description that might be used by a programming language. The concrete syntax defines the actual bit patterns used to represent the abstract syntax. An abstract syntax may have more than one concrete syntax, just as a compiler may have more than one set of procedures for code generation. The transfer syntax is the common syntax understood by two communicating applications. It may be the same or different from the local syntax used internally by either application. The local and transfer syntax are both described by an abstract syntax which is then used to generate the specific concrete syntax that is either used in the system (local) or on the wire (transfer). The Reference Model does correct an oversight in the Presentation Layer standards by defining Transfer Syntax as encompassing both the abstract and concrete syntax used on the presentation-connection, not just the concrete syntax.[20]
>
> *Session Layer* – Since very little has been changed in the Session Layer since Teletex was approved, the major change here was to delete the descriptions of services and functions that had never been provided, such as quarantine.[21]

3.4. Incorporating SC6's Improvements in the Lower Layers

While SC6 had the opportunity to make some sweeping changes to the lower layer architecture and to solve some long outstanding problems, they chose primarily to bring the text into line with current practice. The definitions of "real subnetwork" and "subnetwork" from the Internal Organization of the Network Layer[22] were introduced in the

20. Actually this is an important change. It is crucial to the understanding of the syntax translations that may be required to recognize that the concepts of abstract and concrete syntax are orthogonal to the concepts of local and transfer syntax.
21. Quarantine was proposed very early as a function withheld delivering data to the application until the peer directed it to be delivered. This was not pursued after it was shown that it was provided by existing mechanisms.
22. "Real" is used as an adjective in OSI terminology to distinguish objects in the real world from objects in the logical, abstract

Network Layer; routing was added to the Data-Link Layer in recognition of developments in local area networks; and multiplexing was added to the Physical Layer in recognition of new physical layer technologies, such as T1, Frame Relay, and ATM.

3.5. The Compliance Clause

The biggest change in the Revised OSI Reference Model is the inclusion of an new clause defining consistency and compliance with the Reference Model. The clause defines consistency as follows: a "referencing" standard is consistent with a "referenced" standard in a way which does not alter the meanings of concepts in the "referenced" standard. Compliance is defined in terms of the requirements a "referenced" document puts on a "referencing" document. For example, the Reference Model requires that boundaries of Service Data Units (SDUs)[23] be the same on both ends of a connection. A protocol that meets this requirement complies with the Reference Model; one that doesn't, does not comply. The introduction of this clause belies a major difference in opinion between Europe and North America in the role that the Reference Model plays and in the nature of the forces that create standards.

The impetus for this clause came entirely from Europe. The US had always opposed any sort of compliance statement for the Reference Model and had argued since 1980 that the Reference Model should not be a standard at all. It is difficult to assess how one is to view such a statement, especially at this juncture in the development of OSI. First and foremost, as a consequence of the compliance clause, most of the current OSI stack of protocols are not compliant with the OSI Reference Model. All of the current protocols can be viewed as violating one or more requirements imposed by the Reference Model. Why would the Europeans want to make the existing OSI protocols and their implementations, non-OSI? There is no intent by the committee to set up a "protocol police" to be on the look out for these infringement. There is not even any agreement on which statements in the Reference Model constitute requirements. (Some are more obvious than others). If the clause can't be used, what purpose is it serving? Since many of the changes implied by this clause would cause sufficient change to the protocols to require new versions, can we expect a whole new set of protocol versions of the OSI protocols?

4. What is the Result?

If we take the criteria for measuring change outlined in the OSI Reference Model itself, we would have to say that the changes are so minor that if it were a protocol it would probably not require a new version number. While the OSI Reference Model is facing some major technical problems, some due to wrong decisions made early on, some due to new understanding, or some to new technology, none of them have been addressed by the current "revision." The process that SC21/WG1 set up for revising the Model ensured that none of the critical issues would be addressed and that the current revision would serve only to do a "little clean-up."

There was a strong desire that stability should be maintained at all costs. The cost of this stability without addressing major technical problems has proven costly indeed. Consequently, the Revised OSI Reference Model offers nothing of significance that it didn't offer 10 years

world of the architecture. Thus, "subnetwork" is an abstraction used in the architecture to represent "real subnetworks."

23. SDUs are units in which user-data is passed across a service boundary.

ago. (Of course, it still solves problems that most current “architectures” are still struggling with, such as not having to change addresses because one changes providers.) But, there are three key problems which should have been addressed and would, if included, have gone far to make the revision a worthwhile effort.

5. What WG1 Didn’t Do

5.1. Connection/Connectionless Interworking

As we noted above, the Europeans had imposed a very strict boundary between connectionless mode and connection mode protocols. In particular, there was no means of interworking them at the Network Layer where it is most required. One either had to choose to operate connection mode over connectionless mode or vice versa. Regardless of which one was chosen, the service at both ends of the Network Service had to be the same and connection/connectionless was consider an aspect of the service. While this dichotomy was forced by the PTTs[24] to protect their X.25 market, it became clear by the late 80’s that this hard separation was costing the PTT’s money and new markets. Many of the new lower layer technologies can be considered connectionless or a hybrid of connection and connectionless protocols. A model that unified the two in a meaningful way would be a great advantage. By the end of the 1980’s, the PTTs had begun to use connectionless protocols and began to make noises that a solution was required. While the solution is simple, it has major ramifications throughout the Reference Model.

The primary distinction between connectionless and connection transmission is the amount of shared state. Connectionless transmission represents less shared state than connection mode. A connection mode functionality can be built quite simply on a connectionless functionality by adding mechanisms to the connectionless mechanisms. The converse is not true. Thus, connectionless mode is both more primitive and more basic to a theoretical model of communication. By doing a connection model first and introducing connectionless model later, OSI created a dichotomy where they could and should have created a unification. There is much more to say on this topic. However, it is beyond the scope of this paper to consider in any depth the nature of this unification.

5.2. Revision of the Upper Layer Architecture

As we noted above, SC21 realized in 1983 that a major revision of the upper layer architecture was required and put a program in place to correct the problem. Unfortunately, the first phase (describing the details of the application layer structure) went astray and has only recently been put back on track (even that effort was delayed 2 years by European stonewalling). Consequently today, the revision of the upper layer architecture is only one third complete. There are still problems that derive from the nature of the Session and Presentation Layers and their interactions with each other and the Application Layer.

The placement of synchronization mechanisms in the Session Layer make it difficult for multiple application components to use the mechanisms. The requirement for Presentation Layer to track the behavior of the Session Layer makes the architecture unnecessarily difficult to describe and an efficient implementation difficult to discern from the standard.

24. Post, Telephone and Telegraph, a generic term for The Phone Company.

The Presentation Layer created problems with application relays, etc. But probably worse was that while the structure could be implemented efficiently, naive or mediocre implementors slavishly followed the standards and produced horrendous layer-by-layer implementations.[25] The fact that the correct implementation strategy did not follow the architecture was a major problem. Further, the nature of the Session and Presentation Layers made building new applications out of old ones difficult.

The revised upper layer architecture was supposed to have been complete in plenty of time to be included in the revision of the Reference Model. While this solution is fairly straightforward and creates a much more robust upper layer architecture, the correct solution represents a fairly big change, i.e. a heresy, to the ardent OSI supporter. A more in depth treatment of this is beyond the scope of this paper.

5.3. Multicast

While the model describes connection and connectionless data transfer between two communicating entities, the other major form of communication that was not accommodated in the first version or the revision of the Reference Model was multicast/broadcast. In the mid 1980's, the OSI architecture group had done an initial architecture for multicast. However, because there was no work going on in the lower layers that would allow validation of the architecture, the work was suspended until such time that there was some multicast work that could validate and contribute to the architecture. In 1991 and 1992 in light of considerable multicast work in the lower layers, the US attempted to restart the effort at the international level, but the Europeans were not interested, and the effort failed. The US indicated they felt that the work was important and would continue to work out the architecture in their national committees. Finally in 1993 coincidental with EEC funding, the Europeans suddenly became interested in working on multicast. However, US work had progressed in other forums and there has been little interest in re-hashing the issues.

The inclusion of multicast is crucial to making the Reference Model complete. However, initial indications are that the strong emphasis on the narrow architecture of X.25 early in the development of the Reference Model will mean that major changes will be required in the Reference Model to adequately accommodate multicast. The new work being offered by the Europeans does not recognize these inherent limitations and attempts to develop a model that does not require changes to the Basic Reference Model. At this writing, there are many unanswered questions about this model, not the least of which is whether or not it is implementable.

6. Conclusions

In this paper, we have considered how the OSI Reference Model was revised and the new technical material included in it. The revision of the OSI Reference Model has led to a new edition of the document that is more unrevised than revised. Even though the three key areas of connection/connectionless interworking, upper layer architecture, and multicast needed major work, none of these have been considered.

The OSI Reference Model had 3 golden opportunities to make major contributions to net-

25. Another wonderful example that most implementors treat standards as a "bible" for implementation, rather than the minimal constraints an implementation must meet.

working. All 3 opportunities have been missed. In subsequent papers, we will outline how the problems the reference model chose not to solve can be solved in a manner that leads to a very powerful structure and finish with a somewhat different approach to network architecture that reveals some interesting "patterns" and provides a unifying approach that points to some major simplifications in our understanding of protocol architecture.

7. References

1. ISO/IEC JTC1, Information Technology - Open System Interconnection - OSI Reference Model, ISO 7498, 1984.

2. ISO/IEC JTC1, Information Technology - Open System Interconnection - OSI Reference Model: Part 1 - Basic Reference Model, ISO/IEC 7498-1, 1994.

3. ISO/IEC JTC1/SC21, The Approved Commentaries on the OSI Reference Model, SC21/N8342, 1993.

4. ISO/IEC JTC1, Information Technology - Telecommunications and Information Exchange Between Systems - The Internal Organization of the Network Layer, ISO/IEC 8648, 1988.

5. Nora, Simon and Minc, Alain. **The Computerization of Society**, MIT Press. 1977.

6. Alvey Committee, A Programme for Advanced Information Technology, Report of the Alvey Committee, 1982.

7. Belnes, Dag. "Single Message Communication," IEEE Trans on Communications, Vol Com-24, No. 2, Feb 1976.

8. Spector, Alfred. "Performing Remote Operations Efficiently on a Local Computer Network," CACM, Vol 25, No. 4, April 1982.

9. Danthine, A. Modeling and Verification of End-to-End Protocols, National Telecommunications Conference, Los Angeles, CA. 5 - 7 Dec. 1977.