

インテリジェントLANの試作

戸井哲也⁺ 高橋範泰[#] 沖田明光[#] 相原玲二[#] 阿江忠[#]
⁺富士ゼロックス [#]広島大学工学部 [#]三菱レーヨン

1. まえがき

分散処理システムはすでに普遍的な処理形態として一般化しているが¹⁾、そのなかの一つの形態であるローカルエリアネットワーク(以下、LANと略す)を用いるシステムも、同一構内や建物内のシステムとして急速に脚光をあびてきている²⁾。

LANを用いるシステムの代表例の一つとして、ワークステーションに係わるユーザ層、装置間の情報伝送を行なうネットワーク層、そしてワークステーションでは処理しきれない特別な処理や大容量記憶(データベース)を受け持つサーバ層の3つの層からなる階層モデルで表現されるものがある(図1)。

本論文では、このような階層モデルで表現される分散処理システムで用いられるLANに対し要求される機能について、特にデータベースへの応用を念頭に考察を行なう。たのち、高機能化をめざした一つのLANとしてDRAGON(Dedicated Real-time Application Oriented Network)³⁾を提案する。さらに、そのプロトタイプ的设计、試作例について報告する。

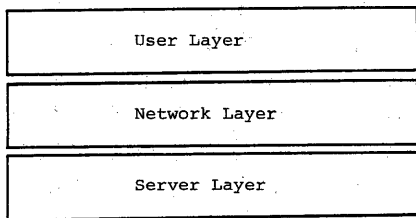


図1. 3層からなる分散システム階層モデル

2. 設計思想

2.1 従来型LANの問題点

従来のLANは、実際の通信処理に関してホスト計算機上のオペレーティングシステム(以下、OSと略す)への依存度が高い。したがって、ホスト計算機の処理負担が大きく、処理オーバヘッドのためシステム全体の実効スループットは伝送路の高速化においても十分に向上しないという問題が生じることがある⁴⁾。

むしろ、それは処理内容に依存し、単なるメッセージ転送程度では許容できることも多い。しかしながらリレーショナルデータベース⁵⁾のように相互に関連したデータベースが複数のファイルサーバに分散して存在する場合には、ワークステーションが複数資源の専有使用を要求することになり、デッドロックの問題⁶⁾が生じる。特に分散処理システムにおけるデッドロックの問題は手数のかかる厄介なものであり⁷⁾、システムのオーバヘッドを飛躍的に増大させる危険性がある。

一方、現状のLANを眺めるに、その多くは、ネットワーク自体は通信プロトコルとして、ISOのOSI参照モデル⁸⁾(図2)で言うところのデータリンク層までしかサポートしておらず、また、分散システムにおいて不可避な共用資源に関するシステムデッドロックについて、ネットワーク自体は何らの対策も持たないのが普通である。したがって、ひとたびデッドロックが生じると、その解決をネットワークの各ノードにつながれたホスト計算機において行なわなければならない。

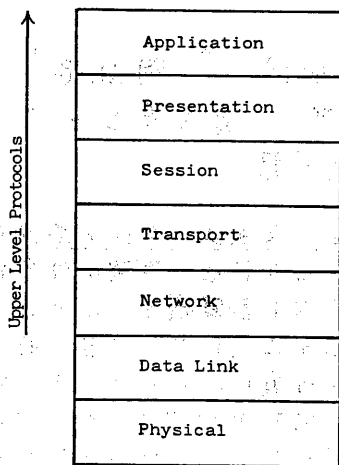


図2. ISO OSI参照モデル

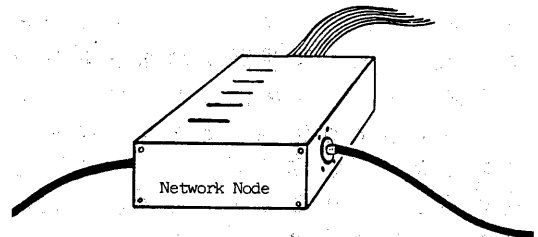
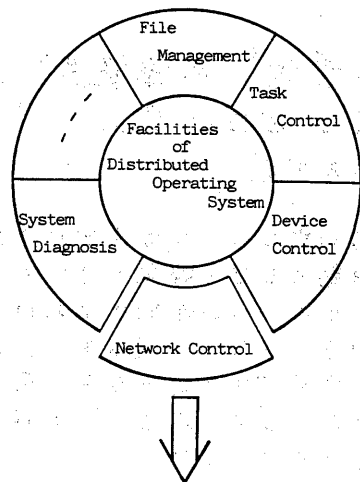


図3. インテリジェントネットワーク

2.2 解決策

このような従来型LANの抱える問題を解決するため、

- (1) ネットワークを構成するノードに、より上位の通信プロトコルまでを実行する通信プロセッサを組み込む。
- (2) 共用資源に関するデッドロックを回避するための基本的なアクセス制御機能を各ネットワークノードに持たせる。

の対策をとり、ネットワークのホスト計算機への依存度を低く抑え、その自立性を高めたのがここで紹介するインテリジェントネットワークである。

つまり、インテリジェントネットワークは、分散処理システムにおいて、従来、分散型OSが担っていたさまざまな機能のうち、ネットワークの通信制御に関する部分の大きさをネットワーク自体が備えた、新しいタイプのLANである(図3)。

3. 上位層プロトコル

ネットワークは通常、階層的に構成

される。その各層は自分より下位の層の提供する機能を利用して、自分より上位の層にいくつかの機能を提供する。たとえば、先に示したISOの参照モデルでは7つの層が存在する。

従来型LANのように、ネットワーク自体がデータリンク層までの通信プロトコルしかサポートしていない場合、実際のデータ通信に必要なより上位のプロトコルは、ネットワークに接続されているホスト計算機が実行しなければならない。そのためホスト計算機上では通信メッセージ毎の複雑な処理が必要となり、処理速度の面で伝送路の速度に十分に対応できない場合がある。

そこで、ネットワーク側に上位層プロトコルまでを高速度に実行する通信プロセッサを付加すれば、ホスト計算機はメッセージ毎の処理から解放され、

それに伴って処理の高速化によるエンドツウエンドのプロセス間の実効スループットの向上が期待できる。特にLANの場合、ネットワークの物理層から最上位の応用層までが同一の管理者の下におかれるので、広域ネットワークのように通信業者の管理する通信サブネットと利用者の管理するホストというように明確に分離して考える必要がなく、ネットワークノードに上位プロトコルを組み込むのに適している。

もっとも、通信プロトコルの最上位部は通常、処理業務の実体であったり、また、ホスト計算機に依存したプロセス毎の処理であったりして、ネットワーク側が独自に実行することは不可能である。

そこで、汎用性を著しく失わない範囲で、インテリジェントネットワークのノードに組み込むことが可能なのは、部分的にプロセス間のコネクションを意識する程度までということになり、ISOの参照モデルに置き換えれば、セッション層プロトコル⁽⁴⁾を境に、ネットワーク側とホスト側（つまりはOS）を分けることになる。

4. DRAON

DRAONは集中制御局をいっさい持たない分散制御型インテリジェントネットワークである。DRAONの各ネットワークノードDIN (DRAON Intelligent Node) はセッション層に対応する通信プロトコルを実行する能力を持ち、また、資源の共

用時に発生するデッドロックを回避するためのプロセス制御をホスト計算機とは独立して行なう。これによりホスト計算機はパケット毎の処理から解放され、またOSは、DRAONの提供するデッドロック回避機能を利用して、容易にデッドロックフリーなシステムを実現できる。

4.1 論理構造

DRAONは図4に示すような論理構造をしている。ここで、ネットワークマネージャはネットワーク全体の管理、リンクコントローラは通信リンクの確立・管理・切断を行なう。なおリンクコントローラは同時に複数の通信リンクを取り扱うことができる。

DRAONに接続されたホスト計算機上のアプリケーションプロセスは、自らの要求に応じて、システムコマンドをネットワークマネージャに提出する。システムコマンドは一種のレコード型データであり、次に示すような構成をとる。

action(arg1, arg2, arg3)

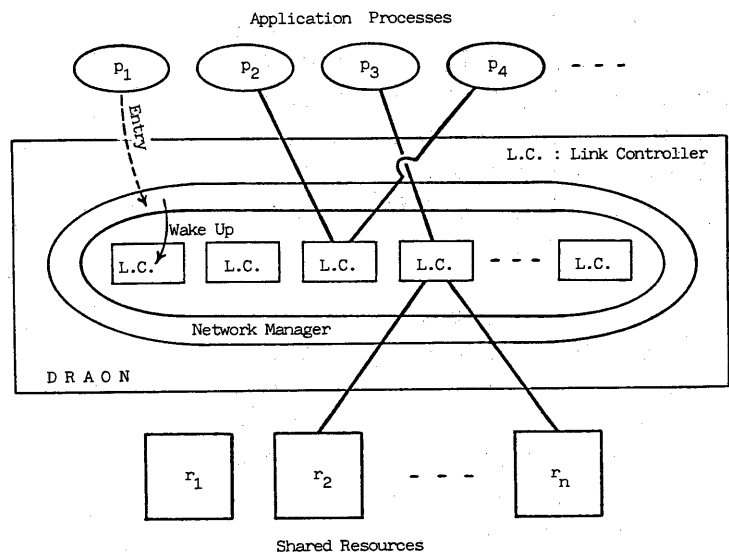


図4. DRAONの論理構造

ここで、

- action はコマンドの指定する動作、
- arg1 はコマンド動作の相手方、
- arg2 はコマンド動作の対象、
- arg3 はコマンド実行時の時間制限 (指定しなければシステム規準値)

具体例を示すと、たとえば次のようになる。

```
Send (printer.PEN, disk.NOTE:
      textfile, 20)
```

—— ファイルサーバNOTE内のtextfileを20秒以内にプリントサーバPENに対して転送開始せよ。

```
Order (disk.NOTE, /ABORT/, )
```

—— ファイルサーバNOTEが提出済コマンドに従って実行している仕事を中断せよ。

アプリケーションプロセスからコマンドを受け取ったネットワークマネージャは、待機状態にあるリンクコントローラを起動し、制御をわたす。一方、起動されたリンクコントローラは自らの管理するローカルタイマを始動した後、コマンド内のarg1に指定された通信リンクの設定を試みる。実行制限時間内に通信リンクが設定できなかった場合、またコマンドの実行を完了した場合は、それぞれネットワークマネージャに対してその旨を報告した後、待機状態にもどる。

4.2 システム階層

図5はDRAONのシステム階層を示している。ここでネットワークマネージャまでの下位4層がネットワークノードDINに実装される。データトランシーバは伝送誤りを意識しなくともより転送サービスを提供し、リンクコントローラはノード間の通信リンクを管理し、順序制御、フロー制御を行な

う。また4.3で述べる共用資源に対するアクセス制御を行なう。そしてネットワークマネージャはこれらすべての動作を監視する。DRAONでは共用資源はそのタイプごとにグループ化され、グループ名と個別名からなる論理名により管理されるが、論理名からネットワークの物理アドレスへの変換は各DINのネットワークマネージャにより行なわれる。

4.3 デッドロック回避メカニズム

ここで対象とするシステムデッドロックとは、分散処理システムにおいて並行に動作する複数のプロセスが、それぞれが既に使用権を持っている資源を互いに要求し合うことによつて、永久に処理を進めることのできない待ち状態に陥ることを意味している(図6)¹⁾。したがって、システムはこれらのデッドロックに対して、何らかの

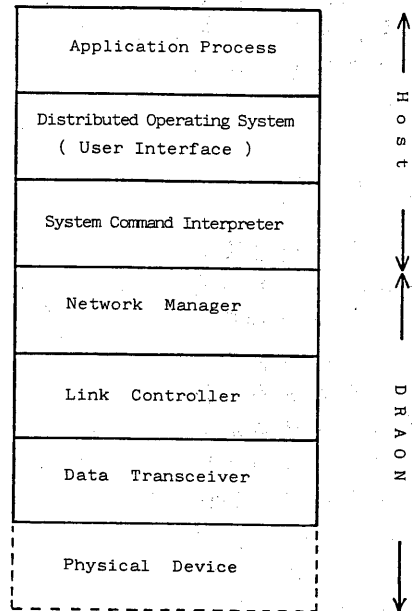


図5. システム階層図

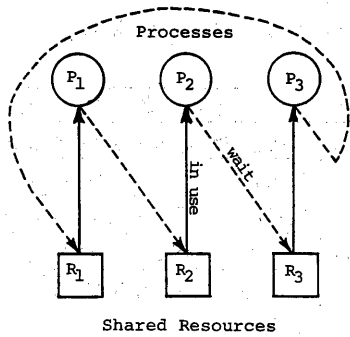


図6. 資源共用におけるシステムデッドロック

対策法を持たなければならぬ。

DRAONのデッドロック回避メカニズムとしては、ノードDINがデッドロック回避のために保持する情報量により大別すると、次の2つの方式が考えられる。一つは、(1)各DINがシステム内のすべての共用資源のステータスを保持しておく方式、他方は(2)各DINは自分自身のホスト計算機上のプロセスの資源使用状況のみを保持しておく方式である。

(1)の方式では、すべてのDINは、各資源の使用権を現在持っているプロセスの識別子が記録されたテーブル(Resource Access Right Table)を持ち、このテーブルの対象資源のステータスを"idle"から自分の識別子に書き換えることのできたプロセスが、その資源の使用権を得る。このRAARTに対する操作はすべてネットワークマネージャによって行なわれシステムがデッドロック状態に陥いるのを未然に察知することが可能である。

一方、(2)の方式においては、プロセスは同時に使用する資源を一括してその使用要求を出さなければならない。リンクプロセスは要求された資源のす

べての使用権を一括して得られた場合にのみ、プロセスにそれらの使用を認める。DINはローカルなプロセスの資源使用状況を把握しており、プロセスが資源の使用を終了した時には対象資源を自動的に解放する。またプロセスが別の資源を追加要求した場合には、リンクプロセスは同様にして、既の使用権を得ている資源を一度すべて解放した後、改めてそれらを一括要求する(図7)。この一括要求・一括獲得方式によってDRAONはシステムデッドロックを未然に回避する。

これらの2つの方式は、(1)が基本的にプロセスの挙動を制限しないかありに、内容的に重複した複数テーブルの内容一貫性を保つ困難さと、それによるオーバーヘッドがある¹²⁾のに対し、(2)では、同時要求という制限はあるものの、実システムへのインプリメントは比較的容易であるという特徴をもつ。

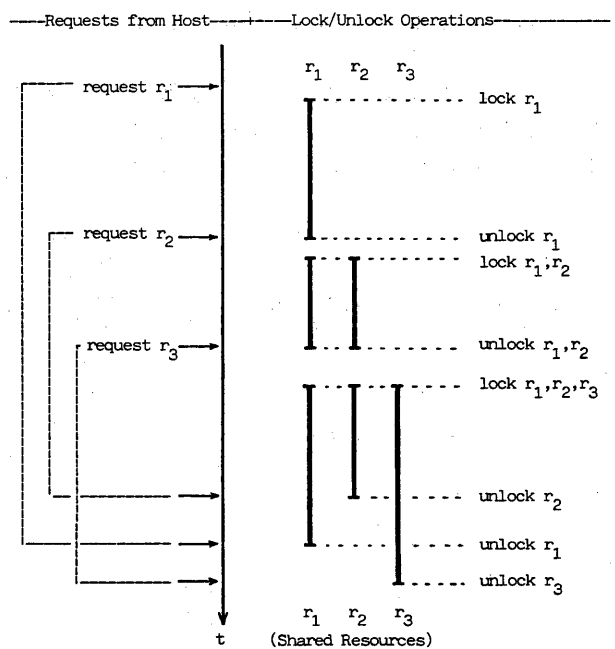


図7 一括要求・一括獲得方式

5. プロトタイプ

試作したDRAONは基本ネットワークとしてCSMA/CD方式のバス型ネットワークSimple Net¹⁵⁾、ノードDINとして8ビット・マイクロプロセッサ・ボードを使用しており、4章で述べた一括方式によるアクセス制御によって共用資源に関するデッドロックを回避する能力を有している。なお、今回の試作では、ハードウェア上の制約から、DRAON本来の機能の一部が割愛されている。

5.1 仕様

図8はワークステーション2台、共用資源としてファイルサーバ、プリントサーバ各1台が接続された試作システムの構成を示している。本試作システムにおいては、現在のところ表1に示すコマンドが使用可能となっており、ユーザはワークステーション上でアプリケーションプログラム、およびコンソールからの直接コマンド入力によりシステムを利用できる。

さらに試作システムには、多数台のワークステーションを接続した場合を想定して擬似トラヒックを発生するダ

表1. 試作システムにおけるコマンド

コマンド	処理内容
nload	ファイルサーバからのファイルのロード
nsave	ファイルサーバへのファイルのセーブ
ndelete	ファイルサーバ上のファイルの削除
nrename	ファイルサーバ上のファイルの名前変更
nprint	プリントサーバによるファイルのプリントアウト
nmessage	ワークステーション間のメッセージの交換
nmove	リソース間の直接ファイル転送
nabort	コマンド動作の中断

ミーステーションと、システムの挙動をネットワーク上のトラヒックにより監視するモニタステーションが接続されている。これらはシステムに対してさまざまな状況設定を行ない、そのときのDRAONの動作およびその有効性を検討するためのものである。

各ネットワークノードDIN内のソフトウェア構成は図9のようになっており、ここでLock Tableにはローカルなプロセスが現在どの資源を使用中であるかが登録されている。ユーザ側から共用資源の使用要求を含むシステムコマンドが提出された場合にはLock Requestメッセージが対象資源に対し送られる。要求資源すべてから規定時間内に肯定応答があった場合にのみ、Lock Tableの対応箇所にコマンド発行プロセス名が登録され、そのプロセスは要求資源の使用権を得る。使用権の獲得においてはネットワークのデータリンク層のCSMA手順による競合が行われ、各ノード間では資源の獲得に関して確率的に公平さが保たれる¹⁶⁾。

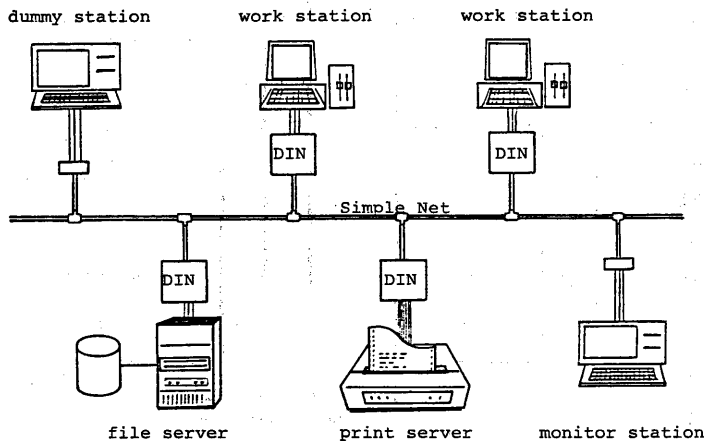


図8. 試作システム構成

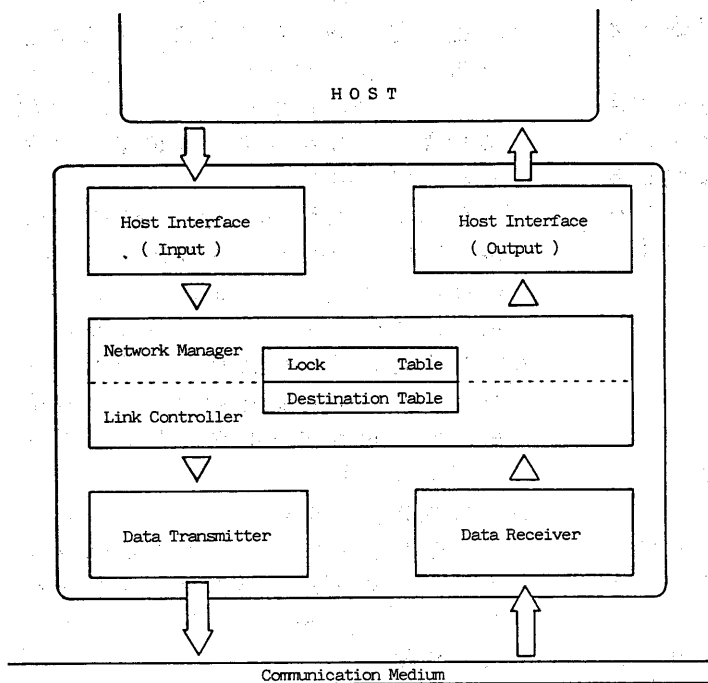


図9. DRAIN内のソフトウェア構成

5.2 問題点と課題

今回の試作は、DRAONの動作の検討のために行なったものであり、ノードDRAINの処理能力も十分ではなく、アドレス管理などの一部機能は簡略化されている。しかし、ダミーステーションおよびモニタステーションを用いた線動実験において、その有効性は確認されており、次の段階として、高速な伝送路（たとえば、Ethernet¹⁵⁾）と高い処理能力を持ったDRAINを採用して、リアルタイム性の強いアプリケーションにも対応できるシステムを製作することを計画している。

なお、デッドロック回避に関してはDRAONでは一括要求・一括獲得という最も自己規制の強い方式を用いているが、一方、4.3で大別した(1)、(2)のデッドロック回避メカニズムの中間に位置づけられる方式も考

えられる。つまり、現在獲得している資源はそのままに、加えて、ある条件を満足する資源に対しては新たに要求を出すことを可能とする方式である。たとえば、すべての資源（総数 m ）に1から m までの番号をつけ、現在獲得している資源の最大値が i ならば x （ただし、 $x > i$ ）なる値の資源のみ要求可能としてもデッドロックは起こらない。

むしろ、固定した番号づけがシステムの柔軟性を損なう場合には、ある周期でサイクリックに番号をまわすような工夫も必要になる。いずれにしても、種々の回避アルゴリズムは少なくとも一括

要求・一括獲得よりインプリメントが複雑になり、その効果とのトレードオフを見極める必要があるが、今後、順次検討したいと考えている。

6. むすび

最近、急速に普及してきたLANのもつシステム的な問題点の一つを取りあげた。デッドロックはデータベース・システムでは不可避なものであり、その解決のためのオーバヘッドはシステムのパフォーマンスに大きく影響する。本論文では、デッドロック回避をネットワークのレベルで行なうLANとしてDRAONを提案し、プロトタイプとして最もシンプルな解決法である一括要求・一括獲得方式をインプリメントした例を紹介した。むしろ、データベース・システムとしては非常に

強い制限のあるデッドロック回避法になっており、今後、具体的な問題ごとにその改善を試みたい。

なお、DRAONのプロトタイプはバス型ネットワークで実現されているが、LANのメディアや形態は特に限定しているわけではなく、DRAON自体はいろいろなものに柔軟に対応できるように考えられている。

参考文献

- 1) 特集：分散処理，情報処理 Vol. 20, No. 4 (1979)
- 2) 特集：ローカルエリアネットワーク，情報処理 Vol. 23, No. 12 (1982)
- 3) 阿江忠，戸井哲也：インテリジェントネットワークDRAON，情報処理ローカルエリアネットワークシンポジウム論文集，pp. 27-34, (1983)
- 4) 寺田松昭他：高速パケット伝送路用前置処理装置の一構成法，情報処理学会論文誌 Vol. 23, No. 6, pp. 707-715 (1982)
- 5) 牧え内顕文：関係モデルとその実現技術，情報処理 Vol. 23, No. 10, pp. 907-915 (1982)
- 6) C. Weitzman : Distributed Micro / Minicomputer Systems, Prentice - Hall, N. J. (1980)
- 7) Ibaraki, T. et al. : Design of Minimum - Cost Deadlock - Free Systems, J. ACM, Vol. 30, No. 4, pp. 736 - 751 (1983)
- 8) Tanenbaum, A. S. : Network Protocols, ACM Comput. Surv., Vol. 13, No. 4, pp. 453 - 489 (1981)
- 9) 松下温：コンピュータ・ネットワ

- ーク、培風館、東京 (1983)
- 10) 鈴木健二他：OS参照モデルにおけるトランスポート・セッション・レイヤ・プロトコルについて，情報処理学会，分散システム研資，19-4, (1983)
- 11) Chandy, K. M. et al. : Distributed Deadlock Detection, ACM Trans. Comput. Syst., Vol. 1, No. 2, pp. 144 - 156 (1983)
- 12) 西原義之他：分散型データベースシステムにおける重複データ制御方式，電子通信学会技報，EC78-6 (1978)
- 13) 阿江忠，戸井哲也：SIMPLE NETを用いるメッセージ伝送システム，情報処理第25回全国大会，19-10, pp. 737 - 738 (1982)
- 14) Tropper, C. : Local Computer Network Technologies, Academic Press, N.Y. (1981)
- 15) DEC, Intel, Xerox : The Ethernet, A Local Area Network : Data Link Layer and Physical Layer Specifications, Version 1.0 (1981)