

## 協調分散型 LAN 間接続方式を適用した キャンパスネットワークシステム†

井田 昌之\*\* 田中 啓介\*\*

青山学院大学の三キャンパスを接続するネットワークシステムについてその設計原則と構築について述べる。幹線機能と利用者環境の提供の双方を担う点に特徴がある。キャンパス間の接続は二重結合とし、信頼度を向上させている。同時に各接続経路は機能分担をもち、定常時にはメールの送受経路とインタラクショングレートを分けるなど負荷の分散をはかった。また、外部のインターネットとのゲートウェイを持っている。このネットワークシステムを Trinity と呼ぶ。Trinity は TCP/IP をベースとしている。三キャンパスを結合するゲートウェイコンピュータ上に、中核となるシステムを UNIX の拡張により構築し、これにより利用者環境を提供する。その中核部を Apostle と呼ぶ。密結合をせずに 64 kbps の接続経路によるキャンパス間接続において同一環境を提供する機構を設計開発した。これによりどのキャンパスからアクセスしても同一の利用者環境を利用できる。また、異なる手順に従うホストコンピュータに対するマッチメカをサーバ/クライアント形式で作成し、二つの利用モデルに基づくアクセス手法を設計開発した。これにより Apostle 利用者環境からのホストアクセスを可能にさせた。

### 1. はじめに

青山学院大学は、青山キャンパス、世田谷キャンパス、厚木キャンパスという三つのキャンパスをもっている。これらのキャンパスの間では以前よりホストコンピュータとそれへの端末が配置されていたが、多様な利用形態への対応の要求と UNIX 導入への期待が 1984 年頃より顕在化してきた。これらに対応するべく 1987 年 11 月より具体的な構想の検討が開始された。これを Trinity\* 構想 (三キャンパス情報環境一体化構想) と呼ぶ。このための実験システムが Trinity ネットワークシステムである。1988 年に具体的なプランの実施をはじめ<sup>1)-3)</sup>、1989 年より公開実験<sup>4),5)</sup>、1990 年より試験運用に伴う再構成を経て、その後、実験/改良を加えながら現在に至っている。

大学キャンパスネットワークとしては、東北大学の TAINS<sup>6)</sup> その他の例がある。これらと同様に Trinity ネットワークは各キャンパスに LAN を構築し、それらを結合する。同時に外部とのゲートウェイをもちインターネットの一部を形成するものである。これに対して、枝線として各学部の個別の LAN をそれぞれの設計に従って接続するように意図している。例えば、世田谷キャンパスは理工学部だけであり、FDDI によ

り各部を接続する LAN が理工学部により形成されており、Trinity とそれとのゲートウェイを置いている。他学部においても順次それぞれの特徴を考慮した LAN の構築が進められている。Trinity は、このように各学部の LAN に対して学内幹線機能を提供するが、それに加えて幹線上の諸リソースに対して統合された利用者環境を与えるシステムを設計開発した点が特徴である。このために Apostle\*\* と呼ぶシステムを Trinity ネットワークの中核として開発した。

実現上の大きな課題の一つは、ホストコンピュータ環境との整合である。ネットワークプロトコルの整合をとると同時にそれだけでなく、会話の制御方式の違い、ファイルシステムの違いなどに対するツールが必要である。Trinity ではこれに対する機構の設計開発を行った点も特徴である。

本論文ではこの三キャンパスネットワークシステムの設計と構築について論じる。全体のネットワーク構想 (2章)、三キャンパスの結合方式 (3章)、中核となる Apostle システム (4章)、利用者サービス機能 (5章)、スーパーコンピュータ接続機構 (6章)、インターネット機能 (7章) の順に論述を進める。

### 2. Trinity 構想

#### 2.1 Trinity 構想の枠組

まず、設計上の原則を

- ユーザモビリティ: どのキャンパスからアクセスしても特別な手段なしに同一の環境を利用できる。

† A Campus Network System as a Cooperative Distributed Environment on the Inter LAN Connection by MASAYUKI IIDA and KEISUKE TANAKA (Computer Science Research Laboratory, Information Science Research Center, Aoyama Gakuin University).

\*\* 青山学院大学附属情報科学研究センター研究教育開発室

\* Trinity: 三位一体。キリスト教用語

\*\* Apostle: 使徒、弟子



- フォルトトレランス：リライアブルシステムを志向する。

という二つの命題に集約させた。

そして、次の要件を満たすものとして Trinity ネットワークの立案をすすめた。

- 広域性：三キャンパスを結合する。これにより、分散しているコンピュータリソースの有効利用を図る。距離的に離れた三キャンパス間の計算機設備を接続したネットワークを構築するのであるから、たとえば、10 Mbps あるいは 1.5 Mbps 等の速度により接続できるとすればもっとも良い。しかし、コスト便益比の観点から実質上不可能である。(青山・世田谷間が約 10 km, 青山・厚木間が約 45 km)

青山学院においては高速デジタル回線(青山・世田谷間, 青山・厚木間)によりキャンパス間接続をまかなっているため、その一部を切り出すことで接続経路とした。

この結果、実現上最高速であった 64 kbps を設計速度とした。この速度では、LAN での方式を用いて三キャンパスを密結合することは困難であるため、新しい方式の開発が必要になった。

- 異機種性と共通性：異機種の結合を図る。これにより、ホストコンピュータ環境からパーソナルコンピュータ環境にいたる連続性を持つ。

先端的な通信網の構築ではなく、コンピュータシステムとしての運用側面を重視する。この結果、現在可用性/共通性が高いと評価した TCP/IP を基本となるプロトコルとした。他のプロトコルに従う機器は、これとのプロトコル変換機構を用意し、全体との整合性を保つ。将来、OSI プロトコルが広く実用化すればそれに移行することとした。

また、使用する OS も既存システムとの親和性を重視し、独自の修正にあたっては極力モジュラにするようにした。

- 幹線機能と利用者機能：各学部における自律的な LAN に対する幹線機能を果たすと同時に、研究者用の利用者環境として、電子メール、ニュースシステム、および、データベースの共通アクセス等の学内情報交換の手段を UNIX 環境を中心に提供する。多数台を要する教育用ワークステーション群などは Trinity に接続する枝線ネットワークでの機能として位置付ける。

- 外部組織とのインターネット接続を行う。これにより、組織間コンピュータ結合および組織間電子コミ

ュニケーションの道具を実験の利用者に提供すると同時に、今後の当該分野における技術に対する知見をうる。

## 2.2 分散性を考慮した管理機能

物理的なキャンパス間の距離は資源管理にあたる管理者の配置についても十分考慮する必要を生じる。集中管理のほうが管理のしやすさがあるが、地理的な距離はそれを妨げる。分散志向により得られるメリットは大きい。三キャンパス全体を通した維持管理に対する機構が必要になってくる。このための仕組みを開発した。

## 2.3 要件を満たすシステムの概形

上記の要件に合うように中核となるワークステーションシステムを三キャンパスの各々に構築し、それを中心に、スーパーコンピュータなどのホスト環境、ならびにパーソナルコンピュータ環境との整合/結合をはかった。この中核となるシステムを Apostle と呼ぶ。

利用者に対しては「同一環境を提供する」という大原則をおいた。「Apostle 上の自分の環境をアクセスしさえすればすべてのコンピューティングリソースを利用でき、その使い方は使用するキャンパスによらず同一となっている。」ということを目指とした。これにより、専任教員(助手を含む)のおよそ 30% は、二キャンパス以上にまたがる職務を持ち、また、そのほとんどは計算機科学の専門家ではない、という状況に対処する。

## 3. 三つの LAN にまたがるシステム形態の選択

### 3.1 設計の選択枝と協調結合

Trinity の設計にあたっては次のような選択枝があった。

A) TSS システムを作り、それを三キャンパスからアクセスする。図 1(a) 参照。この場合、負荷集中の回避が課題となる。

B) 各キャンパスの LAN は独立したものとし、それらの間の違いを意識したままとする。図 1(b) 参照。この場合、一般利用者にとってのユーザモビリティがない。

C) 分散 OS による密結合。例えば Mach<sup>TM</sup> を用いて作業環境を提供する。図 1(c) 参照。この場合、結合経路のバンド幅が大きくなければ有効に機能しない。

このほか、NFS<sup>®</sup> などのネットワークファイルシス



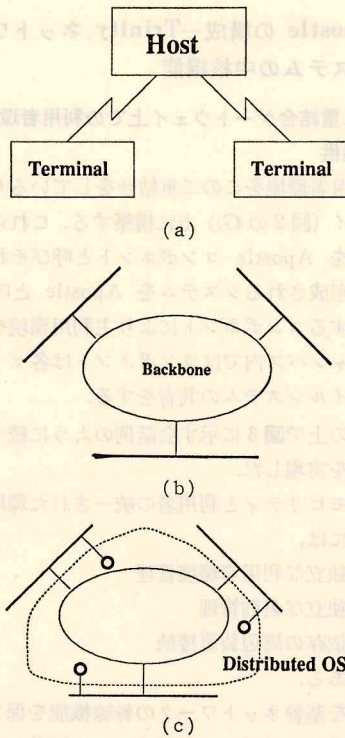


図 1 設計前段階で考慮した三キャンパス接続形態の選択枝  
 (a) ホストコンピュータへのリモートアクセス、(b)バックボーンを介する独立した LAN の接続、(c)分散 OS による統一環境  
 Fig. 1 Choice of the three campus connecting scheme on the early stage of the design.  
 (a) Remote access to a host computer, (b) Independent LANs connected via the backbone, (c) Unified environment supported by a distributed OS.

テムによるファイルシステムの共有なども一案であるが、大別すれば C) に属する。

2章で述べたような要件すべてを上記 A, B, Cは満足しない。したがって、いずれも適さない。

そこで、表 1 のような位置づけを持つ体系として、協調結合と呼ぶ新しい概念の確立を試みた。協調結合は利用者情報管理機能 (4.1 節に後述) を実現する機構である。同一環境の提供を特別な高速回線によらずに、実効的な密結合を達成できることを意図している疎結合型分散システム概念である。

協調結合は、分散 OS や NFS に比して

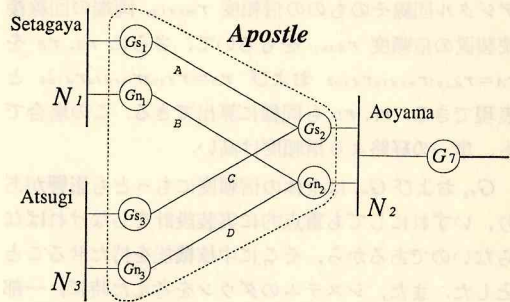


図 2 二重結合による三つの LAN の接続とその上の Apostle  
 Fig. 2 Connecting three LANs through the double links and the Apostle system on top of them.

少ないトラフィックで実用に足る環境の同一性をもつ。その実現とその上での利用者環境については 4 章で述べる。NFS と比較した評価について 8.1 節で述べる。

3.2 二重結合による LAN 間接続

接続の安定性と高信頼性のために、LAN 間接続に際して二重結合をおいた。図 2 を参照。

複数のゲートウェイによる二重結合の信頼度は、図 2 の 6 台のゲートウェイの信頼度をそれぞれ  $rg_i$  とし、経路の信頼度を、それぞれ  $r_A, r_B, r_C, r_D$  とすると、LAN  $N_1$  と LAN  $N_2$  間に対する信頼度  $R_{N_1N_2}$  は、 $R_{N_1N_2} = 1 - (1 - r_A r_{G_{s1}} r_{G_{s2}})(1 - r_B r_{G_{n1}} r_{G_{n2}})$  および、LAN  $N_2$  と LAN  $N_3$  間の信頼度  $R_{N_2N_3}$  は、 $R_{N_2N_3} = 1 - (1 - r_C r_{G_{s2}} r_{G_{s3}})(1 - r_D r_{G_{n2}} r_{G_{n3}})$  となる (ただし各経路は独立とする)。  $R_{N_1N_2}$  および  $R_{N_2N_3}$  は、それぞれ単一経路による接続の信頼度  $R = r_i r_{G_a} r_{G_b}$  より高い。なお、青山学院で実際に利用されたような同一の高速デジタル回線の分割利用の場合では、高速

表 1 協調結合の位置  
 Table 1 Harmonic connection among three connection styles.

	(密結合)分散 OS	協調結合	分散ファイルシステム
例	Mach	Apostle	NFS+NIS (YP)
結合媒体	バス Ethernet	同期回線 X. 25	Ethernet
システムトラフィック	大	小	中
環境の同一性	同一	論理的に同一	共通
分散度	小	大	中
管理ドメイン	単一	階層的分散	階層的分散



デジタル回線そのものの信頼度  $r_{media}$ , 両端の回線接続装置の信頼度  $r_{dsu_i}$  をもちいて, さらに  $r_A, r_B$  を  $r_A = r_{dsu1} r_{media} r_{dsu2}$  および  $r_B = r_{dsu3} r_{media} r_{dsu4}$  と表現できる.  $r_C, r_D$  も同様に算出できる. この場合でも, 単一の経路より信頼度は高い.

$G_{s_2}$  および  $G_{n_2}$  は全体の信頼度にもっとも影響があり, いずれにしても重点的に実装設計をしなければならないのであるから, そこに中核機能を持たせることとした. また, システムのダウンを考えた時に, 一部でも利用可能にすることを考えたので, これら  $G_{s_2}$  および  $G_{n_2}$  の先に外部へのゲートウェイをおき, 一方が故障していても他方がそれによって停止することのないようにした. これにより特に幹線機能の信頼度/可用性を高めた. スーパーコンピュータの配置もこの延長で考えるのが良いが, Setagaya (世田谷キャンパス) (図2) に利用者が多く, その結果そこに置かれたのでその場合, Atsugi (図2) からの利用は直並列型のアクセスモデルとなった.

二重接続のゲートウェイとなるコンピュータは SUN ( $G_{s_i}$ ) と NEC EWS ( $G_{n_i}$ ) を各一台ずつ組にして配置した.  $G_{s_1}-G_{s_2}$  および  $G_{s_2}-G_{s_3}$  を主路とし,  $G_{n_1}-G_{n_2}$  および  $G_{n_2}-G_{n_3}$  を交替路とした.

この二重結合に対して, 動的経路制御と, 定常時における機能分担を設計した. 主路には動的経路制御をしやすい Xerox Synchronous Point to Point Protocol を用い, これに従う SunLink Internetwork Router (INR)<sup>9)</sup> を利用した. 交替路には SLIP を利用した. 一方が故障した場合には自動的に他方に切り替わるようにした.

通常時には負荷の分散を図るようにし, 主路にはインタラクション, 交替路にはメールを通すようにした. なお6章で後述する DINA プロトコルパケットの送受には交替路を優先しその故障時には主路を利用するようにした.

なお,  $r_A, r_B, r_C, r_D$  はそれぞれ 0.998 から 0.999 の実績 (1990 年度一年間) をもっている.  $R_{N_i N_j}$  は, 各  $G_i$  が利用者をもっていることおよびシステム稼働時間が日中だけであったことなどから数量化は困難であるが, マシンドアンの頻度は非常に少なく, その設置目的に照らして高信頼なシステムであるといえる.

## 4. Apostle の構成—Trinity ネットワークシステムの中核機能

### 4.1 二重結合ゲートウェイ上での利用者環境の提供

統合利用者環境をこの二重結合をしている6台のゲートウェイ (図2の  $G_i$ ) 上に構築する. これらのゲートウェイを Apostle コンポネンと呼びそれら全体によって形成されるシステムを Apostle と呼ぶ. 主路を構成するコンポネンにより主利用環境を形成させる. キャンパス内ではコンポネンは各々 NFS によりファイルシステムの共有をする.

これらの上で図3に示す会話例のように統一された利用環境を実現した.

ユーザモビリティと利用者の統一された環境を実現するためには,

1. 位置独立な利用者環境管理
2. 位置独立な名前管理
3. 位置依存の周辺装置接続

が必要である.

これらを基幹ネットワークの幹線機能を保ちながら実現する手法を Information Server (IS)<sup>11)</sup> として開発し, 実現した.

Apostle の利用者は, 各々自分で選択したコンポネンにホームディレクトリを置く. 他のコンポネンでログインする場合, 自動的にそのホームディレクトリが割り当てられる. 同時に位置依存の環境については環境変数の自動設定により, 使用中のコンポネンがあるキャンパスにセットされる. これにより, 三キャンパスで職務を持つ教員も特別な操作なしに同一の自分の環境を使用することができ, 同時にプリント出力などは使用中のキャンパスのものとなる. 利用者情報は階層的に役割を担う管理者が各々の管理範囲の利

```
Apostle Atsugi-Component login: m-ida
password:
Last login: Tue Jun 14 11:05:01 from backabon
apostle% cd x3j13      --- ホームが他キャンパスにある例 ---
apostle% pwd
/home/aoyama/m-ida/x3j13
apostle% ls
cl.mbox      clos.mbox    senddoc.1  cl-edit.mbox  packmail    x3j13.mbox
apostle% packmail cl.mbox --- 遠隔実行 ---
apostle% lpr cl.mbox    --- ローカルなプリンタへ出力される ---
apostle% logout
```

図3 会話例 (本論文の骨子に無関係の部分はカットしてある)  
Fig. 3 Session example on Apostle, for case a user is on a different campus than his home.



用者の分を維持更新する。その情報は整合性を保ったまま全体に反映される。

#### 4.2 利用者の階層

利用者を 1) 通常の利用者である一般利用者, 2) 利用者登録や利用環境の設定等の一般的な管理作業を行うグループ管理者, 3) 各コンポーネントの OS の正常動作のための管理を行うコンポーネントシステム管理者, 4) 全体を統括する Apostle システム管理者の四段階に分割する。Apostle システム管理者の中には変更権限はないオペレータクラスも置く。管理作業のため, 2) は管理範囲にある利用者の個人ファイルと管理上関係するファイル, 3) は system file, 4) は Apostle システム管理のためのすべてのファイルに対して特別に変更権が認められる。この管理権の分散によって, 各管理作業が明確になり, かつ個々の管理者の負担は減少する。

### 5. 統合ユーザサービス

#### 5.1 概要

以下の項目をユーザサービス機能の中心とした。

1. 電子メール: 7章に後述する。
2. 電子掲示板: ニュースシステムを設置した。学内の掲示板も用意した。
3. パソコンファイルサーバ機能: パソコンファイルのセーブ/リストアのための簡易コマンドを準備した。これによって, キャンパス間で移動する教員が同一パソコン環境を得ることを容易にした。
4. スーパーコンピュータアクセス機能: 6章に後述する。
5. 文書整書: T<sub>E</sub>X および関連機能を充実させた。

#### 5.2 低速非同期回線によるユーザアクセスとの整合

通常の一般利用者からのアクセスをサポートするために端末サーバをおいた。また, モデムによる公衆回

線ならびに内線からのアクセス, ISDN 網からのアクセスを各キャンパスで許すようにした。そのための kermit ソフトを配布する体制をとった。また, パソコンやワープロの文書の転送を簡単にできるコマンド, 自動的にコード変換するプログラム等を準備した。

### 6. Apostle とスーパーコンピュータとの結合

#### 6.1 DINA-TCP/IP 変換による接続

スーパーコンピュータ SX-1EA およびその OS である SX-OS は, UNIX システムではないが, Ethernet に接続することができる。その上で, DINA というプロトコルを理解する。したがって, DINA-TCP/IP 変換をするプロトコル変換機構をおき, 合わせて画面制御, 鍵盤のマッピングの変換等を行い, TCP/IP 下の WS で SX-1EA を利用できるようにした。加えて, その変換機構を持つ WS を中継することで同一 LAN だけでなく他のキャンパスの WS/パソコンから SX-1EA を利用できるようにした。

この実現にあたっては EWS-UX にある dinadaemon<sup>12)</sup> と呼ぶデーモン機能を利用した。これは, DINA と TCP/IP との変換を行う基本機構である。

SX-1EA と同一の Ethernet (図 2 の  $N_i$ ) にある EWS ( $G_{n_1}$ ) では, dinadaemon は DINA 手順を Ethernet パケットにのせて直接通信する。 $G_{n_2}$  および  $G_{n_3}$  上の dinadaemon は DINA 手順を TCP の仮想回線にのせて  $G_{n_1}$  と通信することでホストを利用する (図 4)。

これにより, TCP で通信できる範囲のすべての EWS が, ホストと DINA 手順で交信できる。この EWS を Apostle 下の他のコンポーネントから利用することでたとえば, SUN-4 やその他の機械からスーパーコンピュータを利用できる仕組みを開発した。

#### 6.2 マッチメーカ

SX-1EA が要求する仕様と Apostle で標準として

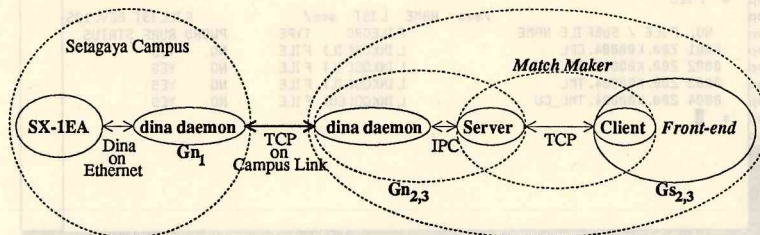


図 4 ゲートウェイを経由した他キャンパス WS からのホストコンピュータアクセス  
Fig. 4 A case for accessing a host computer from a different campus WS through gateways.



いる仕様との間のマッチメカをサーバ/クライアント形式により開発することで以上の仕様を実現した。

主環境 ( $G_{s_i}$ ) を利用中に SX-1EA 交信機能を持つコンポーネント ( $G_{n_i}$ ) に対して自動的にログインする仕組みを、 $G_{n_i}$  で動作するサーバデーモンである *acoserv* とそれに対するクライアントとして動作する起動コマンドを作成することで実現した。(図4)

クライアントは自動ログイン機能に加えて、キーボードの読みかえ、画面表示の対応(例えば、漢字コードの変換)などの処理を行う。これらをクライアント側に置くことで使用するさまざまな端末の特性と利用者によるカスタマイズに対応する。

これらの仕組みにより、SX-1EA 側のソフト/ハードに手を加えずに Apostle 統合環境の一部として SX-1EA を利用できるようになった。

### 6.3 スーパーコンピュータアクセス機能

#### 6.3.1 二つの利用モデル

SX-1EA の利用手段として異なる二つのモデルを用意し、それぞれに対応する機構を用意した。一つは、ネットワーク下の機器から端末としてアクセスするモデルである。これは従来からの端末利用者を想定したものである。もう一つは UNIX システムに付加された応用サーバとしてアクセスするモデルである。

UNIX 環境下での作業の中で必要に応じて SX-1EA へ仕事を投入する。

この実現にあたっては極力既存のツールを組み合わせる方針をとった。可能な限りメーカ提供の部分はそのサポートをうけるためである。

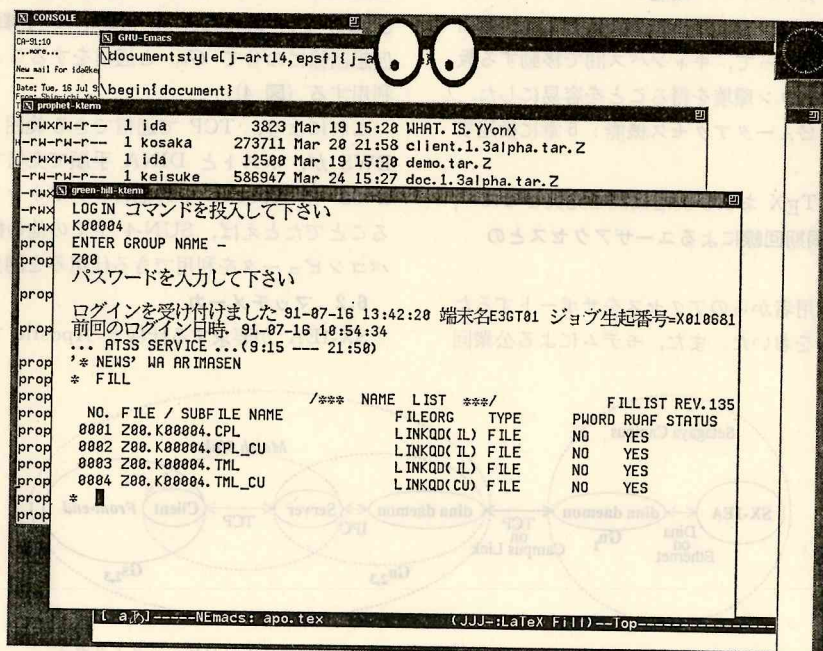
#### 6.3.2 ホスト端末エミュレーション機構

NEC EWS には ETOS 52G<sup>13)</sup> と呼ぶ DINA 手順でホストと会話するソフトウェアがある。画面制御は VT エミュレーションのできるもので、この結果、Apostle 下のパソコンおよび X-window 上の端末エミュレータでセッションを開くことができる。これらにより NEC EWS 以外の端末等でもホスト環境を利用できるようにした。利用者にとって専用端末に比して同程度の応答性能を提供した。

図5にセッション例を示す。

#### 6.3.3 Apostle に対する数値計算応用サーバとしてのアクセス機構

UNIX のコマンド体系の中で、もし、入力されたコマンドがスーパーコンピュータでの実行を要求するものであれば、必要なセットアップがされてそちらの機能呼び出すようにする。このために基本機能として NEC EWS とスーパーコンピュータの間で類似の機能を行う TML<sup>14)</sup> を利用した。TML (Couple) は、 $G_{n_i}$



```

CONSOLE
CP-21.10
...
New mail for 10d8k...
Date: Tue, 18 Jul 91
From: prophet@kern
\begin{document}
\end{document}
prop: prophet-kern
-rwx: 1 ida 3823 Mar 19 15:20 WHAT.IS.YYonX
-rwx: 1 kosaka 273711 Mar 20 21:58 client.1.3alpha.tar.Z
-rwx: 1 ida 12500 Mar 19 15:20 demo.tar.Z
-rwx: 1 keisuke 586947 Mar 24 15:27 doc.1.3alpha.tar.Z
-rwx: green-hill-kern
-rwx: LOGIN コマンドを投入して下さい
-rwx: K00004
prop: ENTER GROUP NAME -
prop: Z00
prop: パスワードを入力して下さい
prop: ログインを受け付けました 91-07-16 13:42:20 端末名E3GT01 ジョブ生起番号=X010681
prop: 前回のログイン日時: 91-07-16 10:54:34
prop: ... ATSS SERVICE ... (9:15 --- 21:50)
prop: * NEWS' WA ARIMASEN
prop: * FILL
prop: /*** NAME LIST ***/
prop: NO. FILE / SUBFILE NAME FILEORG TYPE PWORD RUAF STATUS
prop: 0001 Z00.K00004.CPL LINKQD(IL) FILE NO YES
prop: 0002 Z00.K00004.CPL_CU LINKQD(IL) FILE NO YES
prop: 0003 Z00.K00004.TML LINKQD(IL) FILE NO YES
prop: 0004 Z00.K00004.TML_CU LINKQD(CU) FILE NO YES
prop: * █
prop:
[ a ] ----NEmacs: apo.tex (JJJ-LaTeX FILL)--Top

```

図5 ETOS 52G による画面端末エミュレーション  
Fig. 5 Screen terminal emulation with ETOS 52G.

```

apostle% tml          --- スーパーコンピュータのログインコードの入力 ---
ENTER LOGIN CODE for ACOS: A123456
ENTER GROUP CODE for ACOS: XXX
ENTER PASSWORD for ACOS:

接続しました .....

<中略>
ログインを受け付けました 91-07-08 14:32:55 E3GT01 =X006810
前回のログイン日時. 91-07-05 16:59:57
... ATSS SERVICE ...(9:15 --- 21:50)
'* NEWS' WA ARIMASEN
ホストコンピュータにログインしました.....
apostle-tml% cd ACOS
apostle-tml% cat test.f77      --- 従環境のシェルへ移行 ---
      READ(5,*) I
      WRITE(6,*) 100*I
      END
apostle-tml% f77 test.f77      --- Fortran Compiler の呼びだし ---
FORTRAN 77 がはじまりました    --- f77の実行は SX_1EA 上で生ずる ---
エラーなし

< 中略 >
実行可能な LM を作成しました
正常終了しました
apostle-tml% a.out            --- コンパイル結果の実行 ---
?
5
500
*
apostle-tml% exit
DINA セッションが終了しました

apostle%                    --- 主環境のシェルへ戻る ---
    
```

図 6 Apostle からのスーパーコンピュータの呼び出し  
Fig. 6 Session example for accessing super computer from Apostle.

上で動作する。Apostle 主環境を使用中の利用者が、従環境利用を tml 投入により指示した後に、N<sub>1</sub> 上にあるスーパーコンピュータで実行するべきコマンドを起動

した場合には dinadaemon および ETOS 52G の核機能を利用してそちらを呼び出す。図 6 に例を示す。このセッションの応答速度は専用端末に比して同程度である。

## 7. インターネットとしての Trinity ネットワーク

### 7.1 Aoyama ドメインの形成

この Apostle を中心として構築されたネットワークの構成を図 7 に示す。133.2 というクラス B のインターネットアドレスを持つ aoyama ドメインを形成した。

ゲートウェイを担当する LAN を Apostle とは独立させ、これにより、学内相互の接続と外部ゲートウェイの機能を分離した。セキュリティの向上を図り、かつ機能の分担を行った。

なお、外部との具体的な接続は、64 kbps の専用線を当初 1989 年度は東大大型計算機センター、その後、1990 年度より東京 WNOC との間で行っている。

### 7.2 セキュリティの制御と IP レベルの外部アクセス

Apostle のコンポネントはすべての利用者に対して透過とした。研究開発系（ゲートウェイ部門）の機械はそれらの上でログインコードを持つもののみ透過とした。学外とは、研究開発系を除いては直接的な通信を行わないものとした。学内他部所との接続に対しては、個別に定めるものとした。学外との FTP のためには、anonymous FTP を許すようにした。また、ftp.csrl.aoyama.ac.

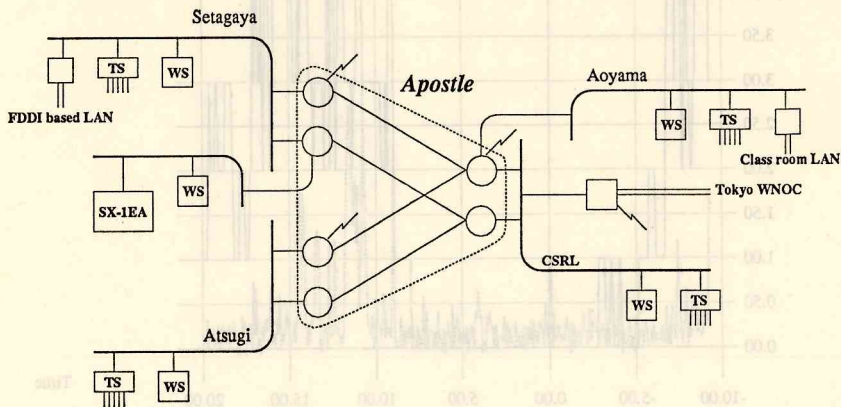


図 7 Apostle を中心とする Trinity ネットワークシステム構成図  
Fig. 7 The current configuration of the Trinity network system centering Apostle.



jp を用意し、公開するファイルを置いた。

外部からの IP 接続は、東京 WNOG との専用線接続のほかに、ISDN によるアクセスを実験的に CSRL に対して用意した。ISDN 網による IP 接続はセキュリティのために発信者電話番号の事前登録とそれとの照合によりアクセスを許可するようにした<sup>15),16)</sup>。

7.3 インターネットメールアドレス

Apostle 利用者の電子メールアドレスは、すべてマシンを特定しない形で表現する。これに対して cc というサブドメインを与えた。例えば、ida は、ida@cc.aoyama.ac.jp であり、どのコンポネントにメールボックスがあるかを問わない。IS は各利用者のホームディレクトリのあるコンポネントを知っているので、その情報をもとに電子メールの送り先を決定する。これによって cc サブドメインの指定だけで Apostle 利用者へメールを送ることができる。また、発信に際しては、マシン名を特定しないような指示をメーラに与えている。

8. 評価

8.1 NFS/NIS に対する Apostle 方式のメリット

ワークステーション A 上で操作をするが、その個人ファイル（ホームディレクトリ）は地理的には離れたワークステーション B 上にある利用者の利用を仮定する。この場合に対して、NFS/NIS<sup>10)</sup> による環境の統一と Apostle での方式との違いを明らかにする。議論すべき相違点を表 2 に示す。

NFS/NIS では、原則的に利用しているワークステーション A 上のリソースを利用し、ホームディレクトリだけをリモートマウントする。Apostle の方式ではホームディレクトリのあるワークステーション B

表 2 NFS/NIS と Apostle 方式の比較  
Table 2 A comparison between NFS/NIS and the Apostle method.

	NFS/NIS	Apostle
使用する CPU	local	remote
使用するホームディレクトリ	remote	remote
使用する環境	local	local

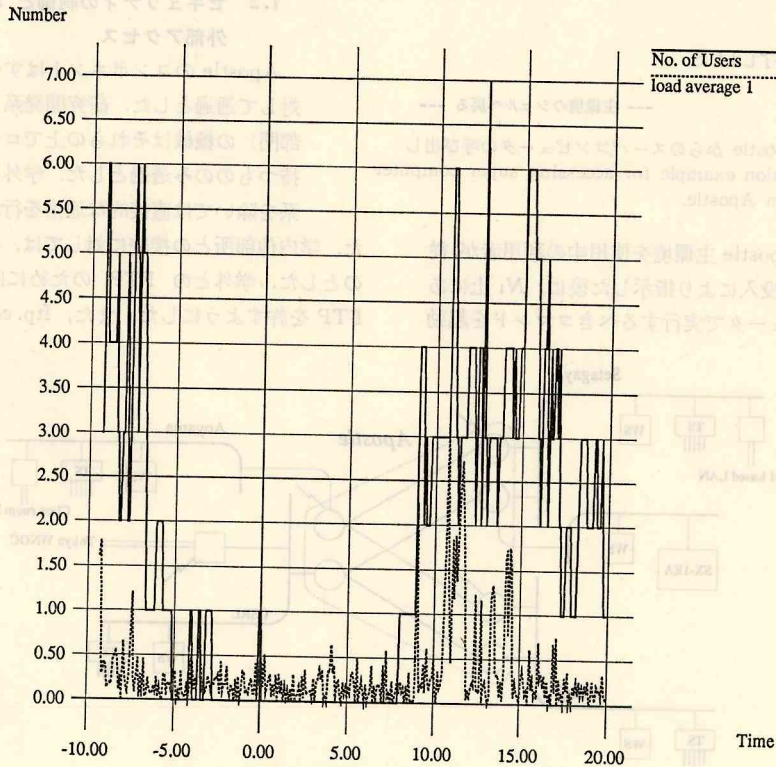


図 8 G<sub>s2</sub> での利用者数および CPU ロードの計測  
Fig. 8 A measurement of number of the users and the CPU load for G<sub>s2</sub>.



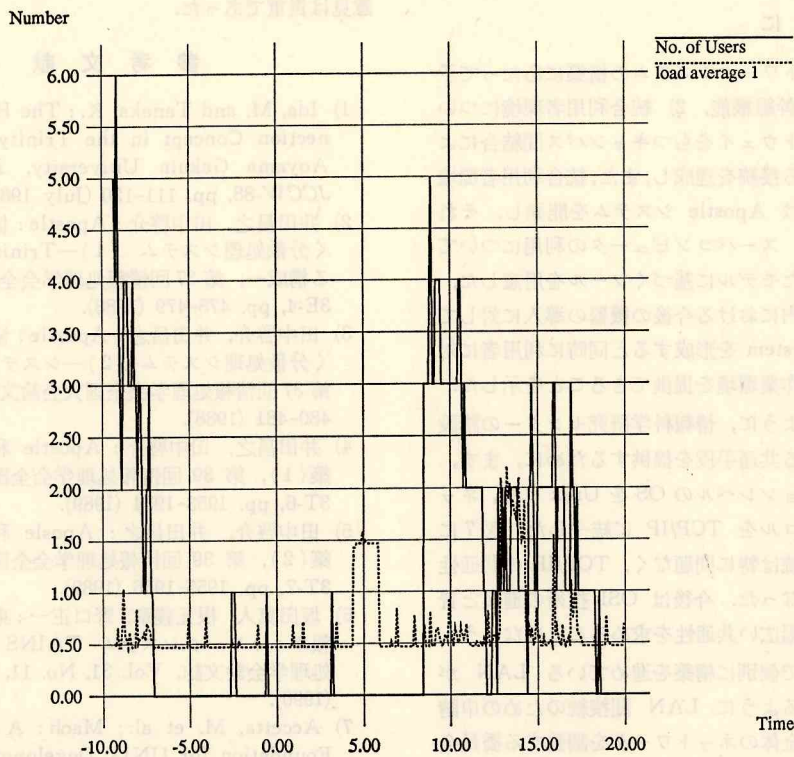


図 9  $G_{s1}$  での利用者数および CPU ロードの計測  
 Fig. 9 A measurement of number of the users and the CPU load for  $G_{s1}$ .

上で実行が進められ、ユーザインタフェースはワークステーション A 上で実行される。

基本的な違いは、ネットワークトラフィックは、「NFS/NIS の場合、該当する個人ファイルアクセスに対して生じる、Apostle 方式の場合、利用者との会話に対して生じる」、という点である。この得失を簡単な指標で示すことは困難であるが、一般にファイルの内容の転送のほうはるかに、会話のために必要な転送量（たとえば、キー入力データおよび表示情報の出力）に比べて大きい。Apostle 方式で大きなデータの転送を必要とするものは非同期でよいプリンタ出力や、応用プログラムが直接に画面を管理するような場合などである。一般的なウィンドウ制御などはすべてローカルで実行されネットワークトラフィックを必要としない。現在、速度の不十分さが感じられるのは  $T_{E}X$  の画面上でのプレビューなどの場合だけである。これらの点で一般利用者の多い Apostle システムでは本論文で述べた方式は NFS/NIS より適した方式であると結論づけられる。

## 8.2 ネットワーク性能

$G_i G_j$  間に対する RTT (Round Trip Time) およびスループットと、Apostle コンポーネントの CPU 負荷の三指標により Trinity ネットワークの性能を示す。

まず、RTT であるが、これは ICMP Echo パケットの応答速度を計測した。ping コマンドを約 10 分間実行させ、これにより測定した。平均 32 msec, 最小 29 msec, 最大 190 msec となっている。スループットは約 1 Mbyte のファイルの FTP により計測した。平均 7.3 Kbyte/sec の速度である。これらの値は 64 kbit/sec の接続としては十分に良い値であると評価できる。

各コンポーネントの CPU 負荷はログインユーザ数ならびにその時点で 1 分以上 run queue にあるジョブの個数で計測した。 $G_{s2}$ ,  $G_{s1}$  に対して 5 分間隔で約 30 時間計測した。この結果を図 8 および図 9 に示す。平均利用者数はそれぞれ 1.686 と 0.889, ロード平均は 0.308 と 0.632 である。各キャンパスに負荷が分散されているといえる。



## 9. おわりに

Trinity ネットワークシステムの構築にあたって予定した 1) 学内幹線機能, 2) 統合利用者環境について, 二重のゲートウェイをもつキャンパス間結合により高信頼性のある接続を達成し, また, 統合利用者環境の提供については Apostle システムを開発し, それにより達成した. スーパーコンピュータの利用についても二つの異なったモデルに基づくツールを用意した. これにより, 学内における今後の機器の導入に対して柔軟な Open System を形成すると同時に利用者に対しては安定した作業環境を提供できることを示した.

2章で述べたように, 情報科学研究センターの諸設備をアクセスする共通手段を提供するために, まず, ワークステーションレベルの OS を Unix にし, ネットワークプロトコルを TCP/IP に統一した. 図 7 に示した異機種接続は特に問題なく, TCP/IP の共通性を立証した形となった. 今後は OSI 技術の進展と普及によりさらに幅広い共通性を求めることになる.

各学部/学科で個別に構築を進めている LAN が有機的に機能するように LAN 間接続のための申請とそれを含めて全体のネットワークを調整する委員会を編成し, 今後の拡張に備えることとした.

今後の課題としては, 7章に述べた事項と関連するが, 学外の接続と学内の接続をどのようにするかが一番の課題である. 例えば, 外部の機関への直接ログイン機能などの課題が存在するが, これに対する技術的/管理的な検討はまだ済んでいない. 現状の接続では東京 WNOG の制御機能とも関係があり, 今後, 研究を進めたい.

また, Apostle の利用者に対して, さらに充実した同一性を与えることなどが内部的な課題である. 64kbps による接続は, 利用の度合に応じて高速化する予定である. 二重結合の経路も利用に比例して今後は全く異なる媒体を組み合わせることを考えている. これらの場合でも本論文に述べた機構はそのまま有効である.

謝辞 本研究の遂行にあたっては, 青山学院大学情報科学研究センターの高森所長ならびに大矢知前所長をはじめとする諸氏の支援を受けた. 東京大学との接続にあたっては大型計算機センターの石田晴久氏, 東京 WNOG との接続にあたっては慶応大学の村井純氏ならびに WIDE プロジェクトの諸氏, の支援を仰いだ. NEC および Sun Microsystems からは技術協力を受けた. 謹んで感謝する. 公開実験利用者からの

意見は貴重であった.

## 参 考 文 献

- 1) Ida, M. and Tanaka, K.: The Harmonic Connection Concept in the Trinity Initiative of Aoyama Gakuin University, *Proc. of 3rd JCCW-88*, pp. 111-120 (July 1988).
- 2) 井田昌之, 田中啓介: Apostle: 協調結合に基づく分散処理システム (1)—Trinity 構想における構成—, 第 37 回情報処理学会全国大会論文集, 3E-4, pp. 478-479 (1988).
- 3) 田中啓介, 井田昌之: Apostle: 協調結合に基づく分散処理システム (2)—システムの実現—, 第 37 回情報処理学会全国大会論文集, 3E-5, pp. 480-481 (1988).
- 4) 井田昌之, 田中啓介: Apostle 利用者環境の構築 (1), 第 39 回情報処理学会全国大会論文集, 3T-6, pp. 1953-1954 (1989).
- 5) 田中啓介, 井田昌之: Apostle 利用者環境の構築 (2), 第 39 回情報処理学会全国大会論文集, 3T-7, pp. 1955-1956 (1989).
- 6) 坂田真人, 根元義章, 野口正一: 東北大学総合情報ネットワークシステム TAINS の構築, 情報処理学会論文誌, Vol. 31, No. 11, pp. 1661-1671 (1990).
- 7) Accetta, M. et al.: Mach: A New Kernel Foundation for UNIX Development, *Proc. of USENIX 1986 Summer Conf.*, pp. 93-112 (1986).
- 8) Sun Microsystems Inc.: NFS: Network File System Protocol Specification, RFC 1094 (1989).
- 9) Sun Microsystems Inc.: SunLink Internetwork Router System Administration Guide (1987).
- 10) Sun Microsystems Inc.: The Yellow Pages Protocol Specification (1986).
- 11) Tanaka, K. and Ida, M.: The Information Server, Aoyama Gakuin CSRL Technical Report No. 88-002 (1989).
- 12) 日本電気: dinadaemon (1M), EWS-UX/V 利用者の手引ネットワーク説明書, EWS-52-3 (1989).
- 13) 日本電気: ETOS52G エミュレータ利用の手引, EWS-75-3 (1989).
- 14) 三上 理, 八木橋信一ほか: 複数のオペレーティングシステム環境とターゲットマシンリンクシステムの開発, 情報処理学会オペレーティングシステム研究会予稿集, 37-1 (1987).
- 15) Hieda, K.: An Experimental Network Using ISDN, *5th JWCC*, pp. 217-228 (1990).
- 16) 田中啓介, 吉田洋之, 佐藤 衛: ISDN を利用した IP ネットワークの構築, 第 42 回情報処理学会全国大会論文集, pp. 1-115-116 (1991).

(平成 3 年 7 月 18 日受付)  
(平成 4 年 4 月 9 日採録)





井田 昌之 (正会員)

1951 年生。1974 年青山学院大学理工学部卒業。1981 年同博士課程修了。工学博士。同年青山学院大学理工学部助手、専任講師、助教授を経て、現在、青山学院大学附属情報科学研究センター助教授。

キャンパスネットワークシステムの研究、Common Lisp 言語仕様に関する研究、分散協調型システムの研究に従事。著書「UNIX 詳説—基礎篇」(丸善、共著者田中啓介)ほか。訳書「Common Lisp」(共立出版、共訳者 8 名)ほか。ソフトウェア科学会、人工知能学会、ACM、IEEE CS などの会員。



田中 啓介 (正会員)

1963 年生。1986 年上智大学理工学部電気電子工学科卒業。1988 年同博士前期課程修了。同年青山学院大学附属情報科学研究センター助手。

学内ネットワーク構築に関する研究をはじめとして、分散環境に関する研究に従事。1992 年 4 月よりニチメングラフィックス株式会社勤務。ソフトウェア科学会会員。