マルチメディア通信と分散処理　47－6
（1990．9．20）

分散環境におけるネットワーク管理

貫井春美　筒井宏史

株式会社東芝　システム・ソフトウェア技術研究所

パーソナルコンピュータ（ＰＣ）やワークステーション（ＷＳ）の急速な普及に伴いソフトウェアの開発環境はＰＣやＷＳをネットワーク接続した分散型開発環境に移行している。このような環境においては、いままでのホスト計算機を中心とした集中型開発環境における管理方式とは異なった、分散型開発環境に適した管理方式が必要と考える。

筆者等は、ネットワーク管理を、管理に必要な情報を収集し、収集した情報を解析し、その結果をフィードバックするといった動作の繰り返しであると捉えている。

本稿では、分散型ソフトウェア開発環境におけるネットワーク管理に関する課題を整理し、管理に必要な情報を収集するためのモニタリング方式を提案し、それに基づいた試作システムと収集した情報の解析に関する考察について報告する。

Network management on Distributed Environment

Harumi Nukui　Hiroshi Tsutsui

System & Software Engineering Laboratory, Toshiba Corporation
70, Yanagi-cho, Saiwai-ku, Kawasaki, Kanagawa, 210 Japan

Distributed software development environment which connects personal computers(PC) and workstation(WS) by a network is popularizing. Such environment, there are several features, eg: independency of node , flexibility of network.

So network management system on distributed environment which is fit for these features is required.

In this papar, we described following ;

(1)problems network management on distributed environment

(2)monitoring method on distributed environment

(3)implementation monitoring system based on this method

(4)analize the information

1．まえがき

パーソナルコンピュータ（ＰＣ）の高性能化、ワークステーション（ＷＳ）の低価格化によりその普及は急速に進んでいる。また、通信技術の進歩によりこれらのコンピュータをネットワーク接続することが容易になった。こういった背景からソフトウェアの開発環境は、大型機やミニコンピュータをホスト計算機としたＴＳＳ中心の集中型開発環境から、ＰＣ・ＷＳといった小型で安価な開発装置をネットワーク接続した分散型開発環境へと移行してきている。

ユーザーは目の前のＷＳ・ＰＣとネットワーク上に分散された資源から必要なものを選択し、よりパーソナルな環境においてソフトウェア開発を行うことが可能になった。

しかし、その利点の反面、不当な資源アクセスが行われたり、特定の箇所に資源アクセスの頻度が集中するなど、ネットワークの円滑な運用を妨げるいくつかの問題も生じてきている。

こうした状況から、分散型開発環境にも管理の必要性が叫ばれている。

筆者等は，ネットワーク管理を

管理に必要な情報を収集する。

収集した情報を解析する。

解析した結果をフィードバックする。

という３段階の動作の繰返しであると考え、ネットワーク管理の研究・開発に着手した。

本稿では、分散型ソフトウェア開発環境におけるネットワーク管理に関する課題を整理し、管理に必要な情報を収集するためのモニタリング方式を提案し、それに基づいた試作システムと、そこで収集した情報解析に関する考察について報告する。

2．分散型開発環境におけるネットワーク管理

2－1．分散型開発環境の特徴

分散型開発環境の特徴をまとめると次のようになる。

（１）ノードの独立性

分散型開発環境は、各計算機が基本的に独立に機能し、これらの開発装置がネットワークを通して、他の計算機・サーバといった資源を必要に応じて相互に利用することができる。

（２）ネットワークの柔軟性

開発対象に応じた柔軟なネットワーク構成が可能であり、１ワークグループ単位のネットワーク（ドメインネットワークと呼ぶ）が、そのソフトウェア開発の規模に応じて構成できる。

（３）ネットワークの拡張性

（２）で述べたドメインネットワークがゲートウェイマシンを介して、他のドメインネットワークと相互に接続される。従って物理的にオープンエンディッドな拡張性を持つ。

2－2．管理上の課題

分散型開発環境の管理上の課題をまとめると次のようになる。

（１）非集中型管理

ホスト計算機を中心とした集中型開発環境では、管理についてもホスト計算機における一括集中管理が可能であった。しかし、各計算機が対等に接続されている分散型開発環境では、ネットワーク上の特定箇所にオペレータを配置するような集中管理は適さない。

（２）負荷の集中

分散型開発環境では、分散された資源が共有されることによってネットワークの特定の箇所に負荷が集中することがある。ネットワーク上のどの資源がどのように利用されているか、特定の箇所に負荷が集中していないか、といったネットワークの利用状況を把握することが必要となる。資源が分散されたことにより、ネットワーク上の資源の使用状況を把握することがむずかしくなる。

（３）アクセス制御

ネットワークを介することにより資源のアクセスが遠隔地から容易になった。これにより、不正なユーザの資源アクセスや、データの破壊といったセキュリティの問題も生じてくる。

特にオープンエンディッドな拡張性をもつために、他のマシンを介したリモートアクセスや、場合によっては、他のドメインネットワークを経由したリモートの資源のアクセスも可能であり、不正ユーザには侵入され易く、逆に不正を発見することは難しくなる。

（４）管理範囲の不明確さ

複数のドメインネットワークが相互接続された場合、誰がどこまでの範囲について責任を持つかが不明確であり、管理範囲に関して明確にする必要がある。

また、種々の管理情報やネットワークの構成情報をどのように持つべきか、またネットワーク内の資源の削除・追加・変更に伴う構成情報の更新、同期の取り方といった点も課題となる。

2－3．ネットワーク管理の構成要素

分散型開発環境におけるネットワーク管理を、管理要素とそれを支援する機能（エンティティ）により整理し、図1．のように定義する。

各管理要素に関して以下に説明する。

（1）利用者管理

資源の利用者の管理を行う。ネットワークを通して資源を利用している利用者の管理も含む。

ユーザＩＤ、パスワードといった利用者情報の登録や利用者情報の更新・削除といった利用者情報データベースの管理機能と、その情報に従いユーザーのログインの可否を判断する認証機能を備える。

また、利用者は許可された資源を正当に利用しているかどうかを管理するために、ネットワーク上のどこで・誰が・何をしているかといった利用状況を把握する監視機能を有する。

不正に資源を利用している利用者を発見した場合には、警告を発するとか資源の利用を強制的に禁止するといったアクションをとることが考えられる。

（2）資源管理

ネットワーク上の資源を管理する。

この場合の資源とは、利用者の利用できるすべてのものを意味し、計算機本体（ＣＰＵ，メモリ）・周辺機器（ディスク，プリンタ等）といったハードウェアから、ユーザーアプリケーションプログラム・各種ツールといったソフトウェアも含む。

ネットワーク上での資源の場所と名前を管理する名前管理機能や、資源へのアクセス権の管理、資源の稼動状況の監視機能といったことが含まれる。

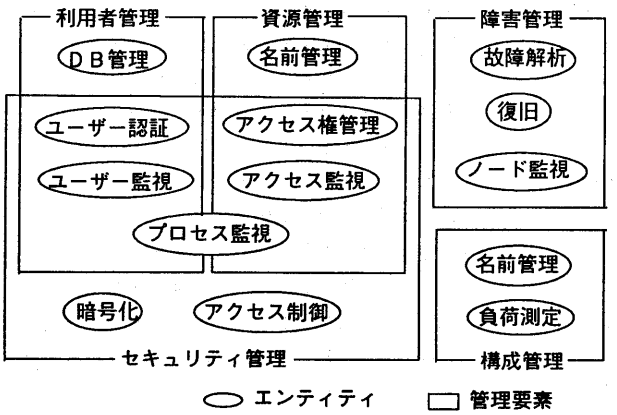

図1．ネットワーク管理の構成要素

（3）障害管理

ネットワーク上の資源の障害を管理する。障害の対象はＷＳやＰＣといった単位が中心となる。

ネットワーク上のＷＳやＰＣが動いているかどうかの監視機能や、障害を検出し利用者に通知する機能、障害復旧機能を備える。

（4）構成管理

ネットワーク上の資源の構成の管理を行う。

ネットワーク構成の変更（ノードの追加、削除、移動）に伴う各ノード間の構成に関する情報データベースの更新を行うといった情報データベースの管理機能を備える。

また、資源の利用状況やレスポンスタイムといったネットワークの性能に関するデータを収集するための監視機能、それをネットワークの再構成するためのデータとしてフィードバックするための解析機能を備える。

（5）セキュリティ管理

ネットワーク上の資源の保全を管理する。

ネットワーク上の資源へのアクセス権の管理機能や、ネットワーク上を流れているデータの保全をはかる暗号化機能といったことを備える。

また、資源のアクセス状況を把握し不正な資源アクセスがないかを識別し、その結果から、不正ユーザのアクセスを禁止するなどの資源のアクセス制御を行うための監視機能を備える。

3．モニタリング

モニタリングは、ネットワーク管理におけるplan－do－check －actionの“check”にあたるフェーズであり、図1．でいう各監視エンティティに相当する。

3－1．モニタリング情報

監視エンティティには3つのサブエンティティから構成される。各サブエンティティとその収集情報を以下に説明する。

（1）ユーザー監視エンティティ

利用者管理の基盤となる情報を収集する。

ネットワーク上のどこで・誰が・何をしているかの　中の”誰が、どこで”を把握するために、ネットワーク上のノードにログインしているユーザの情報を収

集する。

単にノード単位の利用者を把握するだけでなく、リモートからのログインを含めたネットワーク利用特有のユーザの位置（実際の居場所、どこからどこにログインしているか）といった情報を収集する。

（２）アクセス監視エンティティ

資源管理の基盤となる情報を収集する。

どの資源が、誰によって、どのように利用されているかを把握するために、各ノード上で動作しているプロセスの情報を収集する。

プロセスのＣＰＵ使用時間、起動ユーザ、起動端末といった情報を収集することによって、負荷や利用者管理の「何をしているか」といった点にも適用する。

（３）ノード監視エンティティ

障害管理の基盤となる情報を収集する。

どのノードが機能しているかを把握すために、ネットワーク上の各ノードが動作しているかどうかといったノード情報を収集する。

３－２．モニタリング方式

３－２－１．分散型情報管理

大型計算機やミニコンをホスト計算機とした従来の集中型開発環境では、ほとんどの場合ホスト計算機をマスターノードとし管理情報を一括して集中管理する方式が取られていた。また、ＰＣやＷＳを含んだ端末群の制御はすべてホスト計算機が行っていたためその障害はシステム全体に影響を及ぼす。

１つのノードの故障の影響が小さいとか、ネットワーク上に特定ノードを置くことは適さない、といったを分散型開発環境の特徴を考慮し、モニタリングにおいても特定のノードに依存しない方式が要求される。

そこで、各管理情報は各ノード単位に管理する分散型情報管理方式を用いる。

分散型情報管理とは、

・ネットワーク上の各ノードはそのノードローカルな管理情報を把握する
・ローカルな管理情報はＯＳが持つ管理テーブルから参照する

といったことから実現される。

３－２－２．オンデマンド方式

各ノードの管理情報を相互に交換する際のトラフィック量が問題になる。つまり、ネットワークを管理するためのモニタリングとして、ネットワーク本来の機能・性能を脅かさないように、そのトラフィック量を最少限におさえることが要求される。

そこで、管理情報の交換を、各ノード間の要求に応じて行うオンデマンド方式を採用する。これにより、ネットワーク上には、各ノードが情報の要求を行った時にのみ、管理情報が流れるので、トラフィック量を最少限に押えることができる。

３－３．情報収集の範囲

モニタリングの基本方式について前述した。

しかし、分散型開発環境はオープンエンディドとなる可能性があるため、ノード数も理論上は上限がない。従って、ネットワーク上の全ノードに管理情報の要求をすることは現実的にはむずかしい。

そこで、情報収集の範囲をドメイン単位とし次に示すように２階層の手順によってモニタリングを行う。

・モニタリングは各ドメイン単位に行う。
・他ドメインの情報を要求する時は、ドメインを対象として情報を要求し、対象ドメイン内で収集した情報をまとめて受け取ることによって他ドメインのモニタリングを行う。

以上のようにモニタリングの範囲をドメイン単位としたことによってオープンエンディドなネットワークへの対応と、管理範囲が明確になるといったメリットが考えられる。

３－４．分散型モニタリング方式

分散型情報管理とオンデマンドにより、ドメイン単位にモニタリングするモニタリング方式を分散型モニタリング方式と呼ぶ。

（１）ドメイン内モニタリング

・要求ノードはドメイン内の全ノードに対して管理情報の要求を放送（ブロードキャスト）する。
・要求を受信した各ノードは、自身の管理情報を収集し、その情報を要求元のノードに返送する。
・要求ノードは返送された情報を整理し、ＣＲＴに表示するかログとして蓄積する。

（２）ドメイン外モニタリング

・要求ノードは対象ドメインの特定ノードに対して管理情報の要求を送信する。
・要求を受信した対象ドメインの特定ノードは（１）の方法に従いドメイン内の情報を収集する。
・収集した情報を要求ノードに返送する。返送された情報は要求ノードにて（１）と同様の処理をされる。

分散型モニタリング方式の特徴は、

・特定ノードを情報収集用として固定しない
・管理情報は各ノードに分散して保有する
・管理情報は必要に応じてネットワークを介して相互にやりとりすることでネットワーク全体の情報を管理する

といった点である。

3－5．プロセス構成

分散型モニタリング方式によるネットワークモニタを考える。ネットワークモニタの構成はマネージャー／エージェントモデルとし、図2．に示す構成から成る。各マネージャー／エージェントを以下に説明する。

（1）Status Requester

ユーザ監視エンティティ，プロセス監視エンティティ，ノード監視エンティティといった監視エンティテ ィをもつマネージャーである。

ユーザーの要求によって動作し、Status Serverに対して管理情報の収集・応答を要求する。ただし、他の ドメインの場合はDomain Serverに要求を行う。

また、Domain Serverの一部としても動作する。

（2）Status Display

Status Requesterに付随して動作し、Status Requesterが収集した情報を表示する。

（3）Status Server

ユーザー情報，プロセス情報，ノード情報を収集・管理するエンティティを持ったエージェントであり、ネットワーク上の各ノード上で動作している。

自ノード上の管理情報の管理を行っており、Status Requesterからの要求により、要求された管理情報を整理し、要求元のStatus Requesterに応答する。

各計算機上でのローカルな管理情報は、各計算機のＯＳの管理情報を参照する。したがって、ＯＳに制御されるひとつのアプリケーションプロセスとして実現する。

（4）Domain Server

１つのドメインネットワークに１つだけ動作しており、他のドメインネットワークのStatus Requesterからの管理情報の要求を待つ。

要求を受信すると、自身のドメインの情報収集を行うために、Status Requesterとなり、ドメイン内のStatus Serverに対して管理情報の要求を行う。これにより収集したドメイン内の管理情報を要求元のStatus Serverに応答する。

（5）Domain DataBase

ドメイン内のネットワークの接続形態・他のドメインのDomain Serverの位置を記述したデータベースであり、Status Requesterが参照する。

図3．にはマネージャーと各エージェントのプロトコルの概要を示す。

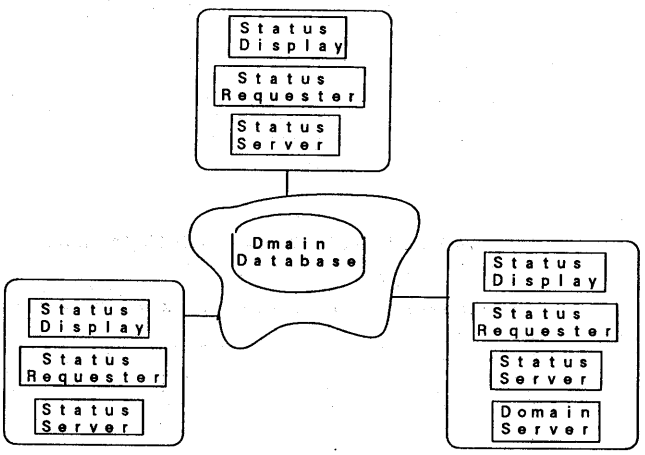

図2．マネージャー／エージェント構成

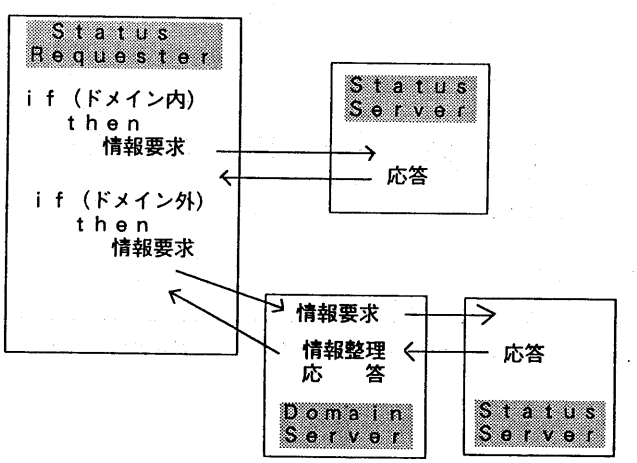

図3．マネージャー／エージェントとプロトコル

4．実現

3．で述べたプロセス構成によるネットワークモニタシステムの実現を次のハードウェア環境で行った。

（1）通信媒体：　Ethernet

（2）計算機　：　AS3000／4000

（3）ネットワーク構成：

Ethernetによりインターネットワーキングされたネットワークであり、中位のプロトコルとしてTCP／IPを用いる。

実現のプロトコル構成を図4．に示す。

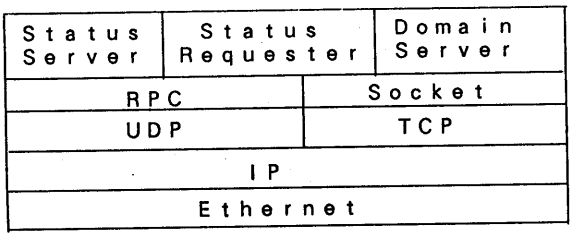

図4．プロトコル構成

図5－1．から図5－3．はStatus Displayによるモニタリング情報の情報表示例である。

図5－1．はネットワークモニタシステムの初期状態での表示であり、ネットワークの接続状態を示す。図5－2．はユーザー監視エンティティにより収集にた各ノードの利用者の一覧を示す。さらに、図5－3．は遠隔からの利用者の一覧を表示している。

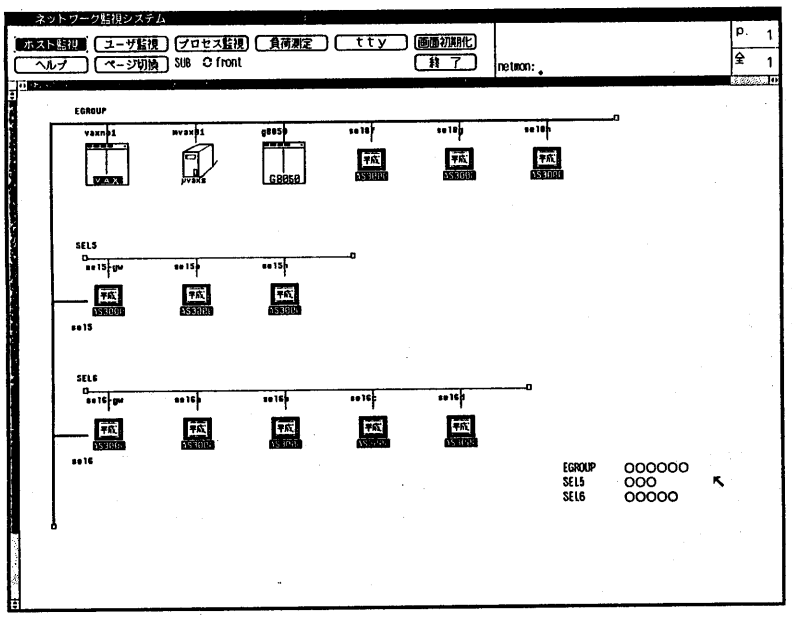

図5－1．ネットワークモニタ（初期画面）

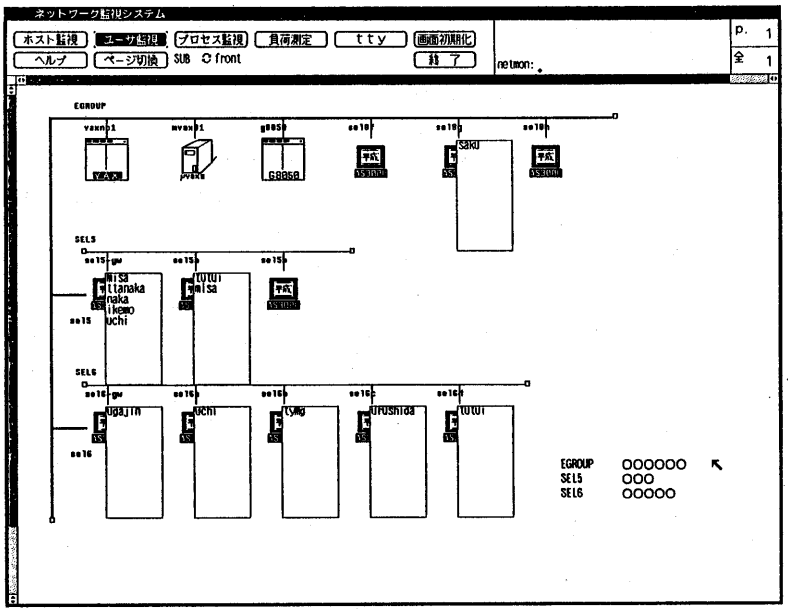

図５－２．ユーザー監視による表示例

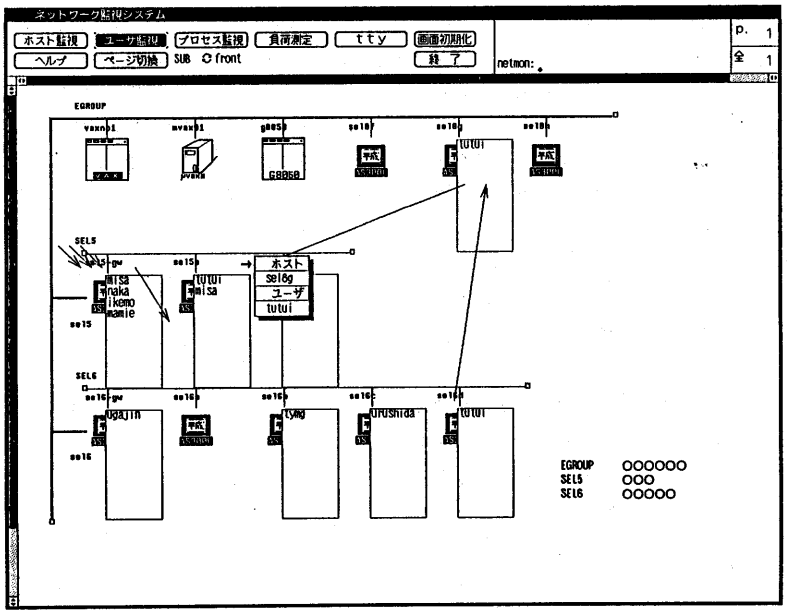

図５－３．遠隔からのユーザー利用の表示例

5．効果

分散型モニタリング方式によるネットワークモニタシステムの効果として以下が挙げられる。

（1）ネットワーク上の任意のノードからのネットワーク利用状況の把握が可能になった。前記した分散型開発環境の特徴を生かしている。

（2）ノードの障害に強いモニタリングシステムが構築できた。

（3）モニタリングの単位をドメイン単位としたことにより管理範囲が明確になった。前記した分散型開発環境の管理上の問題点のひとつを解決した。

6．課題

今後の課題として以下が挙げられる。

・モニタリング情報の解析とフィードバック

・標準化への対応

6－1．情報解析とフィードバック

（1）セキュリティ管理

ユーザ情報、プロセス情報を解析することにより、セキュリティ管理への適用を図る。

現状のモニタリング方式ではユーザーの利用経路（リモートノードからの利用）はひとつ先のノードしかわからない。複数のノードを経由して利用している場合もある。従って、現在の情報を組み合わせることによってユーザーの利用経路をわりだし、不正な資源アクセスの解析や、外部ドメインからの利用に不正がないかといった解析のためのデータとする。

ユーザーの利用経路追跡方式は、ユーザー情報とプロセス情報からわりだすアルゴリズムを考案し試作した。今後評価が課題となる。

（2）構成管理

モニタリング情報からネットワークの利用状況を把握し、効率のよいネットワーク構成のデータとして役立てる。

どのノードにユーザが集中しているか、どれだけ負荷がかかっているか。どのノードにどういうジョブが要求されているか等を解析し、負荷の分散、ユーザの利用効率の向上をもたらす資源構成を考える。

6－2．ネットワーク管理の標準化

図4．に示すように現状のネットワークモニタのプロトコルはＴＣＰ／ＩＰの上にＲＰＣと使って構成されている。

ネットワーク管理プロトコルの標準としてＳＮＭＰ、ＣＭＩＰといったものが挙げられる。ネットワークモニタシステムをこのような標準プロトコルの上に構築し、マルチベンダー対応を考える必要がある。

具体的アプローチとしてＳＮＭＰ上への構築を考える。

7．あとがき

分散型ソフトウェア開発環境におけるモニタリングの一方式として、分散型モニタリング方式を提案し、それに基づくネットワークモニタの実現について報告した。

ソフトウェア開発の現状から、分散型開発環境はソフトウェアの開発環境として定着していくと考えられる。

今後、分散型モニタリング方式の評価・効果の確認を行い、本方式を基盤とした分散型開発環境におけるネットワーク管理の確立をめざす。

＜参考文献＞

[1]G.Skinner "Resource Management In A Distributed Internetwork Environment" ACM comm Apr 1988.

[2]M.Willet "LAN Management in an IBM Framework" IEEE Network Mar.1988.

[3]C.Partridge "The High-Level Entity Management System[HEMS]" IEEE Network Mar.1988.

[4]U.S.Warrier "A Network Management Language for OSI Networks" ACM comm 1988

[5]塚本克治　「分散処理」　昭晃堂

[6]筒井他 "水平分散環境におけるネットワーク管理" 情報処理学会第37回全国大会

[7]筒井他 "分散型ソフトウェア開発環境におけるユーザー利用経路追跡アルゴリズム" 情報処理学会第40回全国大会