マルチメディア通信と分散処理 94−27
分散システム/インターネット運用技術 14−27
（1999. 7. 16）

分散環境におけるアプリケーション運用支援システム

齋藤 武夫 †, Glenn Mansfield‡, 木下 哲男 ‡, 白鳥 則郎 ‡
† 東北大学大学院 情報科学研究科
‡ 東北大学 電気通信研究所

概要： 分散環境におけるアプリケーションの効率的運用，管理，プランニングのためには，ネットワークの現在の通信品質や輻輳状況，それらの過去の履歴など，高度なネットワーク情報が必要とされる．そこで本稿では，従来のネットワーク管理フレームワークを拡張し，ネットワーク情報を収集，蓄積，分析，加工する機能を持つことで高度なネットワーク情報の提供を行う，アプリケーションの運用支援を行なうためのシステムのコンセプトとそのモデルの提案を行う．

An Application Operation Support System in a Distributed Environment

Takeo Saitoh †, Glenn Mansfield ‡,
Tetsuo Kinoshita ‡, Norio Shiratori ‡
† Graduate School of Information Sciences, Tohoku University
‡ Research Institute of Electrical Communication, Tohoku University

Abstract : For efficient operation, (network) applications should have access to network information. E.g. applications need to know the congestion status, the communication quality, the past performance figures etc. of the network in order to determine the operational parameters that will yield optimal performance. Yet, it is inefficient and impractical for applications to generate and maintain such information. In this work, we introduce the concept of an Application Operation Support System (APOS) that gathers network information, analyses it and provides effective information to the applications for better operations. The system concept as well as its model is presented.

1 はじめに

インターネットの普及とその利用者層の拡大に伴い，ネットワーク上で運用されるアプリケーションは多様化の一途をたどっている．従来の電子メールや WWW (World Wide Web) に加え，音声，動画像を用いた放送やビデオ会議システムなどの利用が増え，エレクトリックコマースや医療分野などでも新たなアプリケーションが開発されつつある．

しかし，現在のインターネット環境ではネットワークの通信品質を保証することが困難であるため，あるアプリケーションがなんらかのサービス品質 (QoS) を満たさなければならない場合，自らネットワークの情報を収集し，状況を予測しながら動的に QoS の制御を行う必要がある．よって，従来のほとんどのアプリケーションはこの様なネットワークの状況に応じた運用は行っておらず，ごく一部のアプリケーションにおいて，通信を開始したのちに RTP[1] プロトコル等を

用いてエンド間の通信品質を測定しながら QoS の制御を行なうにとどまっている．

そこで我々は，アプリケーションが容易にネットワーク情報を得ることが出来る支援システムを構築することにより，アプリケーションの効果的な運用が行える環境の実現を試みている．この環境の下におけるアプリケーションは，支援システムからより高度なネットワーク情報を得ることにより，通信を開始する前にユーザが要求するサービスが実現可能かを予測したり，実現可能な QoS を予測することが出来る．また，サービスが実現不可能な場合，他の時間帯なら可能かどうかを予測し，可能な場合，なんらかのネットワーク資源を予約することが出来る．

我々は現在までに，ネットワーク・トラフィックからネットワークの特徴情報を測定する技術を確立しており，その成果をもとにいくつかのインテリジェントなアプリケーションの提案・構築を行っている [2]．また，分散環境で高精度の情報を収集分析することにより，広域ネットワーク上での輻輳の発見とボトルネックの診断アルゴリズムを確立している [3]．

そこで本稿では，従来のネットワーク管理フレームワークを拡張することで，アプリケーションのより高度で効果的な運用を実現するための運用支援システム（APOS）のコンセプトを示し，そのモデルの提案を行う．

2 アプリケーション運用支援システム (APOS)

2.1 アプリケーションの効率的運用

アプリケーションの効率的な運用・管理のためには，高度なネットワーク情報が要求される．たとえば，ネットワークの通信品質情報はアプリケーションがユーザに提供する QoS と密接な関係があり，アプリケーションが利用する通信路の通信品質がわかれば，ユーザに提供可能な QoS を予測することが可能となる．また，ネットワークの輻輳情報はアプリケーションのネットワークに対する負荷を軽減させるきっかけとなり，QoS の変更や通信の実行を見合わせるなどの判断においてきわめて有用な情報となる．さらに，ネットワークを共有する他のアプリケーションの運用スケジュールを知ることが出来れば，ネットワーク利用の競合による QoS の悪化を避けるために運用時刻を変更することも可能となる．

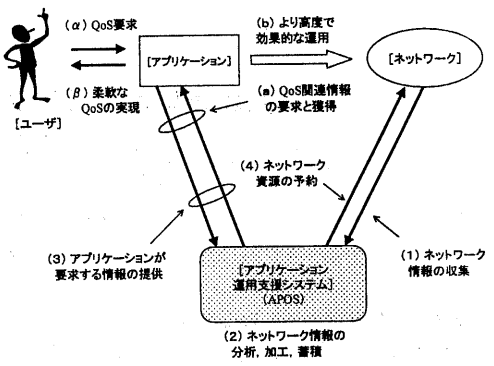

図 1: APOS のシステムモデル

これら高度なネットワーク情報を得るためには，様々なネットワーク情報の収集と分析が必要となる．以下に，アプリケーションの効率的な運用を行うために必要とされるネットワーク情報の例を挙げる．

(1) ノード間のトラフィックフロー情報

(2) ネットワーク構成情報

(3) 輻輳情報

(4) 資源予約情報

(5) アプリケーションのスケジュール予約情報

(6) これらネットワーク情報の履歴

アプリケーションの利用する通信路の通信品質情報を得るためには (2) をもとに (1) の分析を行う必要があり，輻輳情報を得るためには同様に (2) をもとに (3) の分析を行う必要がある．また，ネットワーク資源の利用状況を予測するためには，(2), (4), (5) を基本として，(1), (3) 等の履歴情報を含めて分析を行う必要がある [2]．

2.2 アプリケーション運用支援システム (APOS) の位置付け

しかし，現在のインターネット環境で得られるネットワーク情報はネットワーク管理を目的とした運用情報がほとんどであり，先に挙げたネットワーク情報を得るためには，様々なネットワーク運用情報の収集，分析を行わなければならない．そこで，仮に個々のアプ

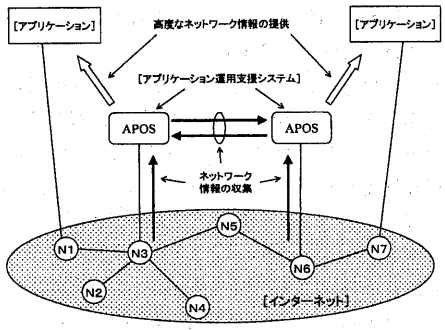

図 2: APOS の位置付け

リケーションがネットワーク情報を収集，分析を行なおうとすると，収集された情報を他のアプリケーションと共有することが困難で，アプリケーションが終了してしまうと情報の履歴も残らない．また，分析に用いる知識の共有も行われず，分析した結果である高度なネットワーク情報の共有も困難であるなど，ネットワーク情報の効率的利用の面で問題が残る．

そこで我々は，アプリケーションが容易にネットワーク情報を得ることが出来る支援システムを構築することによってこれら問題を解決し，アプリケーションの効果的な運用が行える環境の実現を試みている．図 1 に，我々が提案しているアプリケーション運用支援システム (APOS) のモデルを示す．APOS はネットワーク情報の収集，分析，蓄積，加工を行い，アプリケーションの要求に基づき高度なネットワーク情報の提供を行う．また，必要に応じて，アプリケーションのスケジューリングやネットワーク資源の予約等も行う．

インターネット環境における APOS の位置付けを図 2 に示す．APOS はインターネット上に分散されて運用される．個々の APOS は，APOS が運用されるドメインのポリシーの範囲でネットワーク情報の収集を行い，アプリケーションの要求に従って，高度なネットワーク情報の提供を行う．また，他の APOS からネットワーク情報を得ることにより，より精度の高いネットワーク情報をアプリケーションに提供する．

3 アプリケーション運用支援システム (APOS) の設計

本章では，前章で提案したアプリケーション運用支援システムを実現するために，APOS が持つべき機能に関する議論を行う．

3.1 ネットワーク情報の収集

アプリケーションが必要とするネットワークの通信品質情報としては，アプリケーションが動作するホスト間のネットワークの利用可能帯域，ディレイ，ジッタ，輻輳発生状況などが挙げられる．図 2 に示す様なインターネット環境においてこの様な高度なネットワーク情報を得るためには，ネットワークの構成情報をもとにネットワーク N1, N7 間の経路を特定し，その経路上におけるトラフィック状況の分析を行わなければならない．

現在利用可能な，ネットワーク情報を得る方法を以下に挙げる．

(1) SNMP[4] エージェント型: ネットワーク機器上のエージェントから機器の運用情報を収集する

(2) RMON[5] エージェント型: あるネットワークに接続されたノード間を横切るパケットの統計情報や特定のパケット自体を収集する

(3) パケットダンプ分析型: あるネットワーク上を流れるパケットをハードディスク等に一旦ダンプし，後ほど分析を行い情報を生成する

(4) アプリケーションログ分析型: WWW サーバ等アプリケーションによるエンド間の通信品質の測定ログ情報からネットワークの性能情報を得る

本稿ではネットワークから得られる情報に焦点を絞るため，以下，これら方法のうち (1),(2),(3) で得られる情報について議論を行う．

(1) は従来よりネットワーク管理の目的で広く用いられており，ネットワーク機器の様々な統計情報を収集することが出来る．しかし，得られる情報は要求を出した時点での統計情報であるため，トラフィックの傾向などを知るためには，定期的にこれら情報を収集し，履歴情報として蓄積しておく必要がある．

(2) は (1) の方法に比べ，よりトラフィックに関する高度な情報を得ることが出来る．しかし，(1) と同様に傾向を知るためには定期的な情報の収集と蓄積が必要であり，また，機器の資源に限りがあるために，収集できるパケット数に限りがある．

(3) は，分析手法によって容易に高度なネットワーク情報を得ることが可能であるが，一旦ハードディスク等に蓄積してから分析を行うため得られる情報の即時性に問題がある．さらに，ネットワークの広帯域化が進むと，(2) 同様蓄積できるパケットの量，すなわ

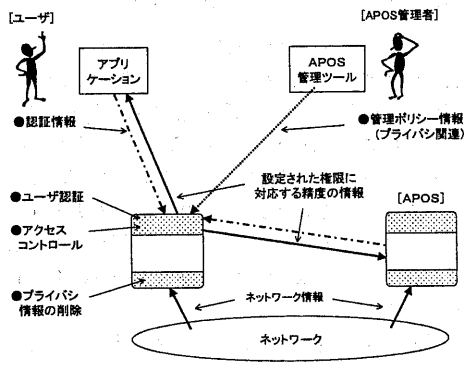

図 3: プライバシ保護のためのアクセス制御

ち観測期間に制限が生じる可能性もある．したがって，パケットの観測地点において高度な分析が行える必要がある．また，パケットをダンプする手法には，本質的にパケットに含まれるプライバシー情報の保護について十分考慮する必要性がある．プライバシーの保護に関する議論は次節であらためて行う．

ネットワーク経路の特定のためは，ネットワーク・マップ・データベース [6] に格納されているネットワーク構成情報，もしくはルーティング・レジストリ [7] に格納されているルーティングポリシー情報を用いる．後者の情報はネットワーク運用者が登録するポリシー情報のため，情報が古かったり誤っている場合もある．これら情報は SNMP を用いて収集することが可能であり，我々は既に CHAIN プロジェクト [8] において，この様なネットワーク構成情報をもとにした情報分析手法を確立している．

以上の議論より，APOS にはネットワーク情報の“収集機能”に加え，パケットダンプを基本とした“トラフィック観測機能”，これら情報にタイムスタンプを付加するなどの基本的な分析を行う“分析機能”，ネットワーク情報を保存する“蓄積機能”，蓄積されたネットワーク情報をもとに，より高度な分析と加工を行う“加工機能”が必要なことがわかる．

3.2 プライバシー情報の保護

前節で述べたように，パケットをダンプし分析する手法を用いる場合，パケットに含まれるプライバシー情報の保護について十分考慮しなければならない．また，APOS により分析提供される高度なネットワーク情報にも，人や組織のアクティビティなどプライバシーに関わる情報が含まれる可能性もある．

我々は，これら問題を解決するためにデータウェアハウスの概念の導入を試みている．機密性の高いデータをあらかじめ生のデータから削除し，残りのデータをデータウェアハウスのシェルで囲むことによって，多様な精度のデータへのアクセスを提供するとともに，プライバシを保証するアクセス制限を行う．図 3 に，そのフレームワークを示す．

APOS を利用するアプリケーションはそのアプリケーションを利用するユーザの権限を持つ．ある APOS を利用するアプリケーションのユーザとしては，APOS 管理者や APOS が運用されているネットワークの管理者，他の APOS の管理者や他のネットワークの管理者，ネットワークの研究者，一般の利用者等が想定される．APOS は APOS の管理者により設定されたユーザとその権限情報により，アプリケーションに提供するネットワーク情報の精度を制御する．以下に IP アドレスに関する精度の制御の例を挙げる．

(1) ノードの IP アドレス（無制限）

(2) ノードのネットワークアドレスのみ

(3) ノードの属する AS 番号

(4) ノードのアドレスをスクランブルする

(5) ノード識別情報を一切与えない

3.3 分散環境における認証

分散環境で運用される APOS において前節で述べた機能を実現するためには，分散環境における安全なユーザ認証および認証システムの運用方法が必要である．APOS では，この認証手段として SNMPv3[4] のセキュリティフレームワークを用いる．SNMPv3 の Userbased Security Model (USM)[9] では，ユーザ認証および認証情報の設定をリモートから安全に行うための，MIB とアルゴリズムが定義されている．

表 1 に USM-MIB を示す．**usmUserName** と **usmUserSecurityName** はユーザを表し，**usmUserEngineID** は，そのユーザの SNMP エンジン [10] にあらかじめ与えられたユニークな ID 番号である．

ユーザは，ユーザが通信を行うリモートの SNMP エンジン (本稿では APOS 内に実装される) と共有す

```
UsmUserEntry ::= SEQUENCE          {
    usmUserEngineID                SnmpEngineID,
    usmUserName                    SnmpAdminString,
    usmUserSecurityName            SnmpAdminString,
    usmUserCloneFrom               RowPointer,
    usmUserAuthProtocol            AutonomousType,
    usmUserAuthKeyChange           KeyChange,
    usmUserOwnAuthKeyChange        KeyChange,
    usmUserPrivProtocol            AutonomousType,
    usmUserPrivKeyChange           KeyChange,
    usmUserOwnPrivKeyChange        KeyChange,
    usmUserPublic                  OCTET STRING,
    usmUserStorageType             StorageType,
    usmUserStatus                  RowStatus
}
```

表 1: USM-MIB

る 2 つの秘密鍵をもつ．それぞれの鍵は，**usmUserAuthProtocol** と **usmUserPrivProtocol** で指定された認証プロトコルと暗号化プロトコルで用いられる．鍵の生成のためには，ユーザを特定するパスワードと **usmUserEngineID** が用いられ，強力なハッシュ関数をもちいて生成される．USM で定義されている鍵の更新アルゴリズムは，新しい鍵とその更新メッセージから古い鍵を特定されない様に設計されており，暗号化されない伝送路での鍵配送を可能としている．

鍵の更新のためには，あらかじめリモートの SNMP エンジンに，ユーザが用いる初期の鍵と初期の暗号化アルゴリズムが設定されている必要がある．そのため，ユーザが鍵の更新を行えるのと同様に，APOS の管理者もユーザの鍵の登録と更新が行えなければならない．この時管理者が用いる認証鍵が **usmUserAuthKeyChange** である．一方，ユーザが認証鍵の更新を行うときには **usmUserOwnAuthKeyChange** を用いる．この MIB に対するオペレーションは，オペレーションの発行者のユーザ名とオペレーション対象の MIB のユーザ名が同一である場合にのみ成功する．これらのフレームワークを用いることにより，管理者はユーザの初期鍵の設定が行え，それによりユーザ自身が鍵の更新を行えるようになる．

4 APOS のアーキテクチャ

前章までの議論に基づいて設計した APOS のアーキテクチャを図 4 に示す．APOS はネットワーク情報

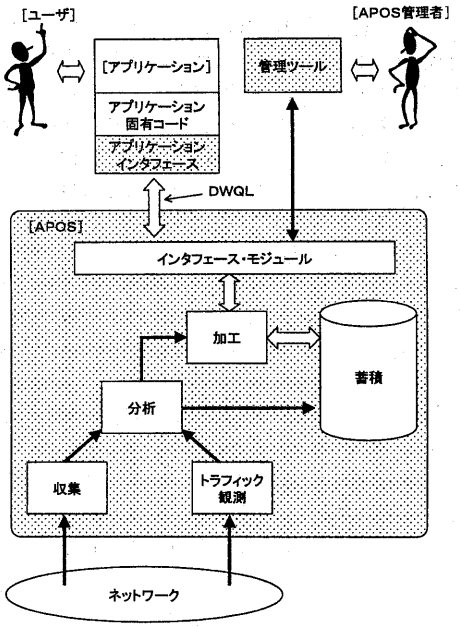

図 4: APOS のアーキテクチャ

の “収集”，“トラフィック観測” 機能をもつ．これら機能で得られたネットワーク情報は “分析” 機能によりプライバシー情報が削除され，時刻情報の付加など基本的な分析が行われたのち，“蓄積” 機能もしくは “加工” 機能に渡される．“加工” 機能は，“分析” 機能や “蓄積” 機能から得たネットワーク情報をもとにより高度な分析を行い，高度なネットワーク情報を生成し，その情報を蓄積したり，ユーザの権限に応じた情報の精度の制御を行いアプリケーションに提供する．

我々は現在，RMON-MIB をベースとしてこの加工機能を制御するためのネットワーク情報データウェアハウジングクエリー言語 (DWQL) の開発を行っている．アプリケーションは，必要な高度なネットワーク情報やその加工方法を DWQL によって記述することで表現する．

また，前章で述べた認証機能は，アプリケーションインタフェース部と APOS のインタフェース・モジュールに実装される．

5 他研究との比較

NetSpec[11] および NIMI[12] はネットワークの性能測定を目的とする．あらかじめ決められた時間に試験トラフィックを流し測定を行うため，ネットワークに負荷を与える．NeTraMet[13] は課金を目的としており，実トラフィックの統計情報を計測している．SPAND[14] は，アプリケーション自体にネットワークの性能情報を測定する機能を付け加え，そこで得られたネットワーク性能情報をパフォーマンス・サーバで管理し一般に情報を提供する．従って，アプリケーションを動作させない限り，ネットワーク情報を得ることは出来ない．堀内らのシステム [15] は，ネットワークの MIB 情報を既存のより高いレイヤの MIB 情報に加工して提供するシステムの設計を行い，MIB 情報の変換加工を記述し変換モジュールを生成するための言語処理系の評価を行っている．しかし，変換モジュールはシステムに静的に組み込まれており，また，ネットワーク情報の履歴を蓄積加工する機能はなない．

6 まとめ

本稿では，分散環境におけるアプリケーションの効率的な運用を支援するためのアプリケーション運用支援システムの提案と設計を行った．現在設計の詳細化と，各機能の実装実験を進めている．

参考文献

[1] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC1889, Jan.1996.

[2] Ahmed Ashir, Glenn Mansfield, Norio Shiratori, "Estimation of Network Characteristics and Its Use in Improving Performance of Network Applications," IEICE Transactions, Vol.E82-D No.4, pp.747-pp.755, 1999.

[3] Takeo Saitoh, Glenn Mansfield, Norio Shiratori, "Network Congestion Monitoring and detection using the IMI infrastructure," Proc of ICPP-99 Aizu, Japan, To be Published.

[4] J. Case, et al. "Introduction to Version 3 of the Internet-standerd Network Management Framework," RFC2570, Apr.1999.

[5] S. Waldbusser, "Remote Network Monitoring Management Information Base Version 2 using SMIv2," RFC2021, Jan.1997.

[6] Glenn Mansfield et.al , "Techniques for Automated Network Map Generation Using SNMP," Infocom96, March 26-28, 1996, San Francisco, USA.

[7] C. Alaettinoglu, T. Bates, E. Gerich, D. Karrenberg, D. Meyer, M. Terpstra, and C. Villamizar, "Routing policy specification language (rpsl)," RFC 2280, Jan.1998.

[8] "Charting the Internet:CHAIN", `http://www.cysols.com/IPAMaps/`

[9] U. Blumenthal, B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)," RFC2574, Apr.1999.

[10] D. Harrington, et al. "An Architecture for Describing SNMP Management Frameworks," RFC2571, May.1999.

[11] R,Jonkman, "NetSpec:Philosophy, Design and Implementation," `http://www.ittc.ukans.edu/Projects/AAI/products/netspec/roel.ps`, 1994.

[12] A.Adams, J.Mahdavi, M.Mathis, V.Paxson, "Creating a ScalableArchitecture for Internet Measurement," Proc.INET'98, Geneva, Jul.1998.

[13] N.Brownlee, "Traffic Flow Measurement: Experiences with NeTraMet," RFC2123, Mar.1997.

[14] Srinivasan Seshan, Mark Stemm, Randy H. Katx, "SPAND: Shared Passive Network Performancs Discovery," `http://www.cs.berkeley.edu:80/~ss/papers/usits97/html/photo.html`

[15] 堀内 浩規, 吉原 貴仁, 杉山 敬三, 小花 貞夫, 鈴木 健二, "ネットワーク管理のための管理情報ベース（MIB）に対する柔軟なビュー提供方式", 情報処理学会論文誌 Vol.39, No.2(1998), pp.367-378.