社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2003－DPS－115　(10)
2003／11／14

自律的なグループ形成機構を用いた分散型ネットワークモニタの実装とその評価

内山 彰[†] 梅津 高朗[†] 安本 慶一[‡] 東野 輝夫[†]

[†] 大阪大学大学院情報科学研究科

[‡] 奈良先端科学技術大学院大学情報科学研究科

本稿では，多数のネットワークセグメントからなる広域ネットワークにおいて，各ノード（ネットワーク構成機器）やトラフィックの異常を効率よく検出するための分散ネットワークモニタシステムを提案し，その性能評価を行う．従来のネットワーク管理システムでは，監視対象ネットワークセグメント数が増加すると，管理者計算機の近辺に制御用トラフィックが集中する，あるいは，異常検知のための時間が長いなどの問題があった．提案システムでは，複数の監視対象ネットワークセグメントで情報収集のためのエージェントを実行し，それらエージェント間で，指定した条件が成立するときに自律的にグループを形成し，グループ間通信を行わせることで，管理者計算機に効率よく情報を収集することができる．Java によるプロトタイプシステムを試作し，広域ネットワークで実験を行った結果，異常検知のための時間が十分に短く，監視対象セグメント数が増加しても，管理者計算機付近の制御トラフィックを実用的な範囲に抑えることができることを確認した．

Implementation and Evaluation of A Distributed Network Monitor with Autonomous Grouping Mechanism

Akira Uchiyama[†], Umedu Takaaki[†], Keiichi Yasumoto[‡] and Teruo Higashino[†]

[†] Graduate School of Info. Sci. & Tech., Osaka Univ., Japan

[‡] Graduate School of Info. Sci., Nara Institute of Sci. & Tech., Japan

In this paper, we propose a new network monitoring system where our system can efficiently detect failure of each node and traffic overload at each network segment in a large and complex network consisting of multiple network segments. Existing network monitoring systems where a single manager retrieves all information from all nodes in a network, are not scalable in the sense that the traffic received at the manager increases in proportion to the number of nodes and network segments to be monitored. In the proposed system, we execute agents on respective network segments and let those agents form a group including a manager if necessary according to the condition which the manager specifies. Once a group is formed, the information is transmitted to the manager with group communication facility such as multicast. We have implemented a prototype system of our monitoring system in Java, and carried out several experiments on a wide area network. As a result, we have confirmed that our system can detect failure and traffic overload in real-time, and the traffic received at the manager is scalable w.r.t. the number of network nodes/segments to be monitored.

1　はじめに

近年のインターネット利用者の急増に伴い，インターネットサービスプロバイダ (ISP) や各自律システム (AS) を構成するネットワークが肥大化・複雑化してきている．また，これらのネットワークの構成ノードにおける OS の脆弱性などをつく，ウィルスやサービス否定攻撃 (DoS 攻撃) [1, 2, 3] が後を絶たない．以上の背景から，ネットワークを広域に監視し，ネットワーク構成ノードにおける障害や異常なトラフィックを検知し，さらには，ウィルス，DoS 攻撃などを早期発見・予防できるための分散ネットワーク管理手法の確立が求められている．

現在までにさまざまなネットワーク管理手法および監視ツールが提案されている．MRTG [4] は，SNMP[5] によりルータと通信しルータ上のトラフィックカウンターを読み取り集計することにより，ネットワークの負荷をグラフ表示するツールである．一方，Snort[6] では，ウィルス検出のための規則をあらかじめ設定しておき，ネットワークを流れるパケットがこれらの規則に一致すると warning を出力することで，ウィルスや DoS 攻撃の検出を可能としている．

しかし，幾つものネットワークセグメントからな

る大規模なネットワークにおいて，これらのツールを使用する際には，遠隔の機器から SNMP などのプロトコルにより定期的に管理者側の計算機に情報を収集する必要があり，監視対象機器が増えるに従い，管理者サイドでの制御トラフィックが増大する上に，膨大な情報の中から必要な情報を把握するのが困難になる．文献 [7] では，モバイルエージェント技術を用いて，エージェントに監視対象 LAN セグメントを巡回させ，遠隔で取得した情報を管理者側に持ち帰ることにより，制御トラフィックの増大を防いでいるが，巡回方法によっては，機器の異常を検知するための時間が遅くなるといった問題点があった．

以上の問題に対し，我々の研究グループは，文献 [8] において，複数の監視対象ネットワークセグメントで情報収集のためのエージェントを実行し，それらエージェント間で，指定した条件が成立するときに自律的にグループを形成し，グループ間通信を行わせることで，管理者が効率よく情報を収集する分散ネットワークモニタ方式を提案した．本稿では，この分散型ネットワークモニタを実装し，その評価実験を行った結果を報告する．

提案方式では，分散ネットワークモニタを，ネットワークの各セグメントに配置実行される**トラフィック収集用エージェント**と，あるノード (複数ノードでの実行も可能) で実行され，収集した情報を GUI を介して管理者に提示する**管理用エージェント**とで構成する．トラフィック収集用エージェントは，SNMP を用いて，監視対象セグメントのルータからトラフィックの情報を取得し，プロトコルごとの単位時間あたりパケット数などの情報を集計するよう実装を行った．管理用エージェントが，多数のトラフィック収集エージェントのうち，問題が発生したエージェントとのみ通信を行えるようにするため，我々のグループ通信ミドルウェア [9] を使用し，指定した条件が成立する時にのみ，管理用エージェントを含むエージェントのグループを形成し，マルチキャストなどによるグループ通信を行わせるよう実装を行った．

提案方式の有用性を評価するため，プロトタイプシステムを Java により実装し，広域ネットワークで実験を行った．実験の結果，異常発生後のグループ形成時間が，数秒程度と十分実用的であること，また，提案方式により管理用エージェント付近の制御トラフィックを大幅に削減できることなどを確認した．

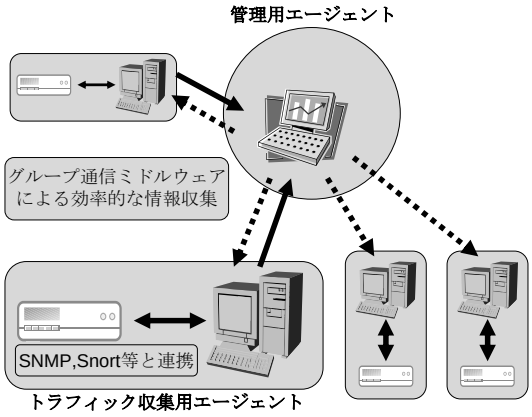


図 1: ネットワークモニタシステムの構成

2 自律的なグループ形成機構を用いた分散型ネットワークモニタ構成法

本システムは各管理対象用ネットワーク (以下，セグメントと呼ぶ) に配置・実行され，必要な情報を収集する「トラフィック収集用エージェント」と，ある管理者ノードで実行され様々なセグメントから収集した情報を実時間表示する「管理用エージェント」が協調動作することでネットワークモニタ機能を提供する (図 1)．本システムは 2 層で構成され，上位層は，指定した条件を満たす時のみ監視対象グループを形成し，グループ通信を行う．これにより，効率的な情報の収集が可能となる．下位層は，各セグメントのトラフィック情報などを監視して，統計情報として保持する．各セグメントの監視方法は，様々な既存のツールが利用可能であるので，監視目的に応じて柔軟に対応でき，かつシステムの実装が容易となる．以下では，各層についてそれぞれ説明する．

2.1 上位層：グループ通信ミドルウェア

上位層では，我々のグループ通信ミドルウェア [9] を利用する．我々のグループ通信ミドルウェアでは，通信チャネルの作成のための 2 つのメソッド (**Advertise** と **Participate**) を提供する．Advertise/Participate メソッドはそれぞれ，グループメンバの募集/グループへの参加要求を行うメソッドであり，同期チャネル (ゲート) 名のリスト，値リスト，チャネル生成条件を指定して呼び出す．Advertise は任意のアドレスのリストを指定して実行することができるので，ブロードキャストなども可能である．従って，状況に応じて近隣の全ての

エージェントに対してグループ参加募集メッセージを送信したり，特定のエージェントだけをグループに参加させるなどの動作が可能である．グループ参加募集メッセージを受信したエージェントがParticipate メソッドを実行し，与えられた値リストがチャネル生成条件を満たした場合にはチャネルが生成される．その際，Advertise を行っても条件を満たす Participate がない場合などには，Advertise/Participate は失敗となる．これらのメソッドを用いることにより，値リストに通信量を設定し，チャネル生成条件として，閾値を指定することで特に通信量の多いエージェントと結合する，といったようなプログラムが簡潔に記述できる．それらを呼び出したエージェントは互いに指定されたチャネル名リストに含まれるチャネル上での**同期関係**により結合され，仕様記述言語 LOTOS[10] のセマンティクスに従い動作する．結合されたエージェント群がさらに他のエージェント (群) と結合することで，階層的に多数のノード間における同期関係を構成できる．同期関係により結合されたエージェント群は，それらのチャネル上の入出力アクションを同期的に実行 (マルチランデブ) することで互いに通信を行う．また，生成されたチャネルは生成時に得られるチャネルの ID によって区別でき，ID を指定し切断メソッド (**Disc**) を呼び出すことで破棄できる．

このグループ通信ミドルウェアを用いることで，指定した条件を満たした場合のみ，グループを形成して情報を集めることができ，効率的な情報の収集が可能となる．

2.2 下位層：既存の監視ツールを利用した情報取得

下位層では SNMP，tcpdump，Snort などの様々な監視ツールを利用することができる．例えば，SNMP を用いた場合はトラフィック情報を各トラフィック収集用エージェントが取得し，統計情報として保持する．また，Snort を用いた場合は，DoS 攻撃などの検知が可能となる．このように，下位層で利用するツールを変更することで，監視目的に応じて柔軟な対応ができる．また，既存のツールを利用することができるので，システムの実装が容易となる．

3 分散型ネットワークモニタの実装

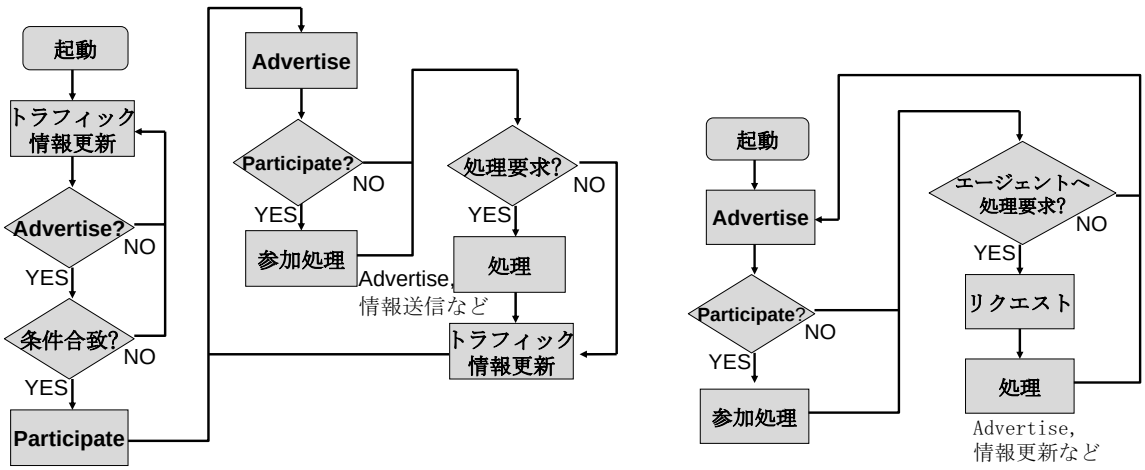
2 章で述べた「トラフィック収集用エージェント」と「管理用エージェント」の実装に関して説明する．各エージェントは Java を用いて実装を行った．なお，下位層としては，SNMP，tcpdump，Snort などの様々なツールと連携可能であるが，ここでは SNMP を用いた実装について述べる．

3.1 トラフィック収集用エージェントの実装

トラフィック収集用エージェントは以下の機能を持つ．

（1） 配置されているセグメント (ネットワークインターフェイス) を通過するパケット量を SNMP によって取得し，統計情報として記録する
（2） 管理用エージェントの指示に従い，トラフィック情報を送信する
（3） 管理用エージェントによって指定された参加条件を満たす場合，監視対象グループへの参加を行う
（4） 監視対象グループへの参加後，自律的に周辺のトラフィック収集用エージェントに対して同様の参加条件で参加募集を行う
（5） 管理用エージェントの指示に従い，周辺のトラフィック収集用エージェントに条件を指定して参加募集を行う

トラフィック収集用エージェントは，図 2(a) のフローチャートのように実装できる．まず，トラフィック収集用エージェントは，IP，ICMP，UDP プロトコルそれぞれの，パケット量を SNMP によって定期的に取得し，統計情報として保持する．管理用エージェント，もしくは周辺のトラフィック収集用エージェントから，Advertise によるグループ参加募集があった場合，それらの情報が与えられる条件を満たした際には，管理用エージェントに対してグループへの参加要求を行う．これにより管理用エージェントは異常などの発生を知ることができる．そしてグループ参加後，周辺のトラフィック収集用エージェントに対して同様の条件でグループ参加募集を行い，自律的に監視対象グループを拡大する．また，管理用エージェントからのリクエストに従い，定期的に，収集したトラフィック情報を送信することで実時間性を有したネットワーク管理が行える．さらに，トラフィック収集用エージェントは，管理用エージェントからのリクエストに



(a) トラフィック収集用エージェント　　(b) 管理用エージェント

図 2: エージェントの実装 (フローチャート)

よっても，監視対象グループの拡大を行う．具体的には，周囲のトラフィック収集用エージェントに対して，管理用エージェントから与えられた条件で参加募集を行い，グループを拡張する．これによって，管理用エージェントは監視対象グループを条件を変えながら拡大することができ，問題の原因特定に役立つと考える．

トラフィック収集用エージェントの実行時には，管理用エージェントおよびトラフィックを収集するセグメントのアドレスが与えられる．パケット量の取得は SNMP を用いるので，常に情報取得のためのトラフィックは一定で，メッセージサイズも十分小さい．また，情報取得は分散した各セグメントごとに行われるので，その際の処理やトラフィックは問題とならない．

3.2 管理用エージェントの実装

管理用エージェントは，以下の機能を持つ．

（1） トラフィック収集用エージェントに対して条件を指定してグループ参加募集を行う
（2） 監視対象グループの各セグメント情報を GUI を介して表示する
（3） 監視対象グループの指定したセグメントに対してトラフィック情報の要求，指定した条件でのグループ参加募集などのリクエストを行う

管理用エージェントは図 2(b) のフローチャートのように実装できる．まず，管理用エージェントは周囲のトラフィック収集用エージェントに対して条件を指定して Advertise を行う．トラフィック収集用エージェントは受信した Advertise メッセージを実際に利用されるまで保持しておくため，Participate によるグループ参加要求や，指定条件の変更などがない限り，再度 Advertise を行う必要はない．ただし，Advertise メッセージは UDP パケットのブロードキャストによって送信されるため，パケットが失われてしまう可能性も考慮し，グループ参加要求がない場合でも，一定時間経過後再び Advertise を行う．収集用エージェントからの Participate による参加要求があれば，その参加処理を行い，参加したセグメントの情報を GUI を介して表示する．監視対象グループに参加しているセグメントに対しては，トラフィック情報の送信，セグメント周辺への Advertise，などのリクエストを行うことができる．モニタリングを実施しているセグメントの監視を終了したい場合には，そのセグメントのトラフィック収集用エージェントとの間の通信チャネルに対して Disc メソッドを実行する．それにより監視対象グループからトラフィック収集用エージェントが離脱し，モニタリングが終了する．

実行に際しては，管理可能なネットワーク (監視するトラフィック収集用エージェントが配置されたネットワーク) のアドレスを登録する．グループ形成時には実行時に与えられたネットワークに存在する，全てのトラフィック収集用エージェントに対して定期的にマルチキャストを行うが，その頻度を調節することにより，制御用メッセージによるトラフィックの増加は十分抑えることができると考えられる．

3.3 グループ形成例

ここでは，Snort との連携により DDoS 攻撃を受けているネットワークセグメントの範囲を検出する例を示す．Snort には設定したルールに基づいて，DDoS 攻撃を検出する機能がある．このルールに基づき，あるノードで Snort が DDoS 攻撃を検出したとする (図 3(a) の E，F)．あらかじめ管理用エージェントは「DDoS 攻撃を検出した」という条件でグループ参加募集メッセージを周辺のトラフィック収集用エージェントに定期的に送っている．(図 3(a) の A) 従って，DDoS 攻撃を検出したノードは管理用エージェントに対して参加要求メッセージを送る．これにより，監視対象グループが形成される．さらに，グループに参加したノードは周囲のエージェントに対して同じ条件でグループ参加募集メッセージを送る．(図 3(b) の E，F) Snort が DDoS 攻撃を検出したノードはこのメッセージを受け取ると，参加要求メッセージを送る．(図 3(b) の G，H) このようにして監視対象グループが自律的に形成され，DDoS 攻撃を受けているネットワークセグメントの範囲を検出することができる．

4 実験と評価

本方式の性能を確かめるため，以下の項目についての実験と評価を行った．

- 異常事態と見なすべき条件が満たされた後，その情報が管理用エージェントに伝わるまでの時間 (マルチランデブ 1 回にかかる時間)
- Advertise を実行したとき，遠隔地にあるエージェントがグループに参加するまでの時間

監視対象グループに対して，トラフィック情報送信条件を，「ICMP パケットが 1000$packet/sec$ 以上の頻度で到着している」という条件の Advertise を行う，その状態で，実際にあるセグメントに対して ICMP パケットを 1000$packet/sec$ 以上の頻度で送り始めてから，管理用エージェントにその情報が届くまでの時間を計測した．

ノード間の遅延がほとんどない場合の理想的な環境での計測結果は，ノード数 50 で約 1 秒，100 で約 3 秒，150 で約 5 秒，200 で約 8 秒となり，エージェント数に対して線形的にマルチランデブ 1 回にかかる時間も増加していることが分かった．

次に，実際にそれぞれ異なるネットワークに存在する，PC(A)[1] を含む 3 台の PC(B，C) を用意し，実験を行った．A-B 間の遅延時間は約 26 ミリ秒，A-C 間の遅延時間は約 900 ミリ秒である．PC(A) で管理用エージェント，残りの 2 台でトラフィック収集用エージェントを 1 つずつ起動して，グループ形成を行い，その後指定した条件を満たしてから管理用エージェントにその情報が届くまでの時間を計測した．この時間は，前述の通りマルチランデブ 1 回にかかる時間とほぼ等しくなる．この計測結果から，実際に異なるネットワークにエージェントが分散していても，2 ノードのときは 1 秒程度でマルチランデブ 1 回が行えることが分かった．

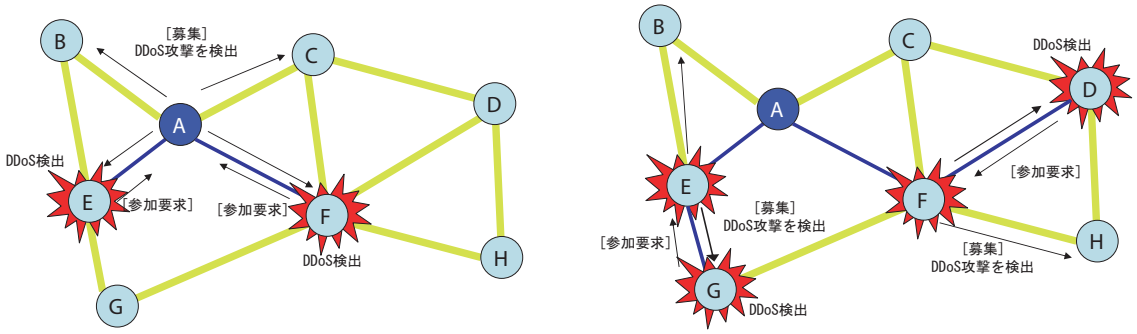
さらに，実際に異なるネットワークに存在する PC(A) を含む 2 台の PC を用意し，PC(A) で管理用エージェント，もう 1 台でトラフィック収集用エージェントを起動して，Advertise を行ってからトラフィック収集用エージェントがグループに参加するまでの時間を計測したところ，約 1.5 秒かかった．ping を用いてこの PC 間の通信遅延時間を計測すると約 0.9 秒であり，同一 LAN 内にエージェントがある場合，1 エージェントの参加処理に要する時間を測定したところ，約 0.5 秒であった．従ってこの計測結果は，通信遅延がない場合の 1 エージェントの参加処理時間に通信遅延時間を加えた理論値とほぼ等しい．ここで，エージェント数が増えた場合でも，1 エージェントの参加時にやりとりされる情報量は一定であるので，管理用エージェント付近の回線容量が十分あるとして，参加処理に要する時間に最大通信遅延時間を加えれば，ほぼ実測値を算出できることが分かる．

5 まとめ

本研究では，自律的なグループ形成機構に基づく，分散型ネットワークモニタの実装とその評価を行った．

本手法では，監視したいネットワークの各セグメントに「トラフィック収集用エージェント」を配置，実行することで，そのセグメントのトラフィック情報を一定時間ごとに収集，統計情報として保持させる．これらの情報は，本研究グループで提案しているグループ通信ミドルウェアが提供する動的マルチランデブ機構を用いて「監視対象グルー

[1] CPU:Pentium III 1GHz(Dual Processor), Memory:768MB, OS:Debian GNU/Linux 3.0, Java 2 SDK, Standard Edition, version 1.4



(a) 監視対象グループの作成　　(b) 監視対象グループの拡大

図 3: 管理範囲の拡大

プ」を形成し,「管理用エージェント」に送られる．管理用エージェントでは監視したいセグメントや，そのセグメントでモニタする項目の指定を行い，トラフィック収集用エージェントから送られる情報を元にトラフィック状況などを表示する．

今後の課題として，広範囲のネットワークに適用した場合の通信遅延時間を考慮してグループ形成の順序調整を行うことにより，問題発生後その状況が終了するまでに適切なグループ形成を行う手法や，Snort などと連携することによるネットワーク状態の監視，DoS 攻撃の検出などの際にパケットキャプチャによりさらに情報を解析する機能の追加などが考えられる．また，ネットワークのオーバーヘッドをさらに抑えるため，トラフィック収集用エージェントから参加募集メッセージを，管理用エージェントに送るという手法が考えられる．この場合，各トラフィック収集用エージェントに対してあらかじめグループ形成条件を指定しておく必要があるが，定期的に参加募集メッセージを送る必要がないため，大幅なトラフィック量の削減が可能と思われる．各トラフィック収集用エージェントに条件を指定するため，あらかじめデフォルト値を設定しておく，起動時に管理用エージェントが条件を各トラフィック収集用エージェントに配布するなどの方法がある．また，管理用エージェントから動的に条件を変更することができるようにし，よりネットワークの状況に応じた柔軟な対応が可能となるようにする必要がある．さらにトラフィック情報の可視化を行うことで，直感的な理解を可能にすることも計画している．

参考文献

[1] Garber, L. : “Denial-of-Service Attacks Rip the Internet”, *IEEE Computer*, pp. 12 – 17 (2000).

[2] Moore, D., Voelker, G. M. and Savage, S. : “Inferring Internet Denial-of-Service Activity”, *USENIX Security Symposium* (2001).

[3] Schuba, C., Krsul, I., Kuhn, M., Spafford, E., Sundaram, A. and Zamboni, D. : “Analysis of a Denial of Service Attack on TCP”, *Proc. of the 1997 IEEE Symposium on Security and Privacy (S&P1997)*, pp. 208 – 223 (1997).

[4] MRTG : The Multi Router Traffic Grapher, http://www.mrtg.org/

[5] Case, J., Fedor, M., Schoffstall, M. and Davin, J. : “A Simple Network Management Protocol(SNMP)”, *IETF RFC1157* (1990).

[6] Snort.org , http://www.snort.org/

[7] Gavalas, D., Greenwood, D., Ghanbari, M. and O’Mahony, O.: Hierarchical network management: a scalable and dynamic mobile agent-based approach, Computer Networks, Vol. 38, No. 6, pp. 693–711 (2002).

[8] 梅津高朗，賀紋孝夫，安本慶一，東野輝夫 ：“セグメント間での自律的なグループ形成機構を用いた分散型ネットワークモニタの実現” , DICOMO 2003, Vol. 2003, No. 9, pp. 625 – 628 (2003).

[9] Umedu, T., Yaumoto, K., Nakata, A., Yamaguchi, H. and Higashino, T. : “Middleware for Synchronous Group Communication in Wireless Ad Hoc Networks”, *Proc. of Communications and Computer Networks*, pp. 48 – 53 (2002).

[10] ISO: “Information Processing System - Open Systems Interconnection - LOTOS - A Formal Description Technique based on the Temporal Ordering of Observational Behavioer”, *ISO 8807* (1989).

[11] 安本慶一，中田明夫，寺島芳樹，梅津高朗，東野輝夫，谷口健一 ：“マルチランデブチャネルの動的確立機構を持つモバイルアプリケーション記述言語の提案”, コンピュータソフトウェア, Vol. 19, No. 2, pp. 35 – 46 (2002).