サーバー間連携の稼働実態解析技術

高良 真穂

Technique of Analyzing a Server Relationship State

Maho Takara

企業情報システムの可用性や信頼性の向上のためには，システムを構成するサーバー群の連携状態を把握することが重要である．本論文では，既存サーバーとネットワーク機器への影響を与えずにパケットを収集してサーバーの連携状態を解析する方式を提案し，それを一般的ハードウェア装置で実装して実システム環境に適用して有効性を確認したので報告する．

To improve the reliability and availability of a corporate information system, it is important to understand the cooperative state of the server group that makes up the system. In this paper, I propose a system which collects packets without any influence on the existing server and network apparatus and analyzes a server's cooperative condition, and report that when it was mounted with general hardware equipment and applied to real system environments, its validity was confirmed.

Key Words & Phrases：サーバー間連携，ネットワーク監視，トラフィック分析，解析ツール，IP ネットワーク
Linkage between servers, Network monitoring, Traffic analysis, Analysis tool, IP network

1. はじめに

現在の企業情報システムは，多数のサーバーの複雑な連携として構築されていることが多く，ハードウェアやネットワークなどの障害時には，その影響範囲の把握と対処が容易ではない．

企業は，事業の継続性を損なわないために，サーバーの障害やデータ・センターの被災を想定して，影響を受ける事業所，利用者数，サーバー数などを把握し，影響の大きさに応じて，信頼性確保にかける費用や対策を判断することが必要とされている．

企業の情報システムでは，利用者の PC が数万台に達し，サーバー数が数千台に達することも多い．このような大規模のシステム環境では，高速なネットワーク上に，膨大な量の情報が行き交っており，サーバー間の連携状態を把握することは容易でない［1］．

この多数のサーバーの複雑な連携を把握するための技術として，サーバーに情報収集のエージェントを導入する方式がある．しかし，企業情報システムのサーバー群は，一般に多数のベンダーにより提供され，アプライアンス製品からメインフレームに至るまで多種多彩なハードウェアと OS で構成されている．そのため対応できる範囲が制限される．この方式では，サーバーの CPU 負荷増大によるレスポンスなどへの悪影響が考えられるため，既存環境に適用して影響範囲を評価するツールとしては不十分であった．

一方，ネットワーク機器の機能を活用した方式もあるが，既存環境への適用を考慮した場合，これら機器のアップグレードや CPU 負荷増大による処理能力への悪影響が懸念されるため，適用が容易ではない．

本論文では，企業の大規模なシステム環境を想定し，既存のサーバーとネットワーク機器へ影響を与えない方式でパケットを収集し，そのパケット中の特徴的な情報に着目し，サーバー間の複雑な連携状態を解析するツールを提供することを目的とする．具体的には，2 章で先行技術の課題を整理し，3 章でその課題を解決するシステムについて提案する．4 章でその実装上の考慮点を述べ，大規模システム環境へ対応する実現方法を示す．5 章で適用事例を紹介し有効性を確認する．最後に 6 章で，まとめとして有用性と今後の課題を示す．

提出日:2008年9月9日　再提出日:2008年12月15日

2. 先行技術と課題

サーバー間の連携状態を把握する従来技術として，ネットワーク監視装置［2］とトラフィック分類装置［3］があるが，以下の課題がある．

(1) プロトコル仕様が非公開なアプリケーションは解析できない．このため，特定アプリケーションに限定された解析となる．
(2) 非常に多くの種類があるアプリケーション層プロトコルの解析に時間がかかってしまう．
(3) 未知のアプリケーションが利用するポート番号やアプリケーション層のプロトコルをすべて把握し，解析対象を設定することは現実的ではない．
(4) ポート番号以外の情報を追跡しないため，DoS 攻撃手段として，大量の SYN パケットを特定のサーバーへ集中的に送りつけるなどの不正アクセスの検知に応用できない．

また，大規模なネットワーク環境に対応した技術として，ルーターやスイッチなどのネットワーク機器の機能により，通信状況を収集する方式［4］，［5］，［6］があるが，以下のような課題がある．

(5) 既存のネットワーク機器に対して，変更を加え，新たなモニタリング・プロセスを稼働させる［7］．これにより機器の CPU 負荷が増加するのでリスクを伴う．場合によりメモリーの追加，または機器更新を伴う．

これまでにも小規模な解析対象であれば，パケット・スニッファリングのソフトウェアを用いて，通信パケットをディスクに保存し，サーバーの連携関係の解析が可能である．

ところが，本論文が取り組む企業情報システム全体に及ぶ範囲を対象とすると，既存のソフトウェアを用いる方法では，下記のような課題が生じる．

(6) 高速なネットワークのパケットを収集，解析する際，トラフィック量に CPU 処理能力が追いつかず，取りこぼしなどが発生する［8］．
(7) 高速なネットワークでは，伝送されるデータ量が膨大である．そのため，収集したパケットをログとしてディスクへ保存するには，膨大なディスク容量が必要となる．

3. 提案システムの構成と処理

3.1. システム構成

サーバー間の連携状態を解析する提案システムの概要を図 1 に示す．本システムは，解析対象の中にあるネットワーク機器からパケットを収集して，連携関係を解析することで，レポートを利用者へ提供する．

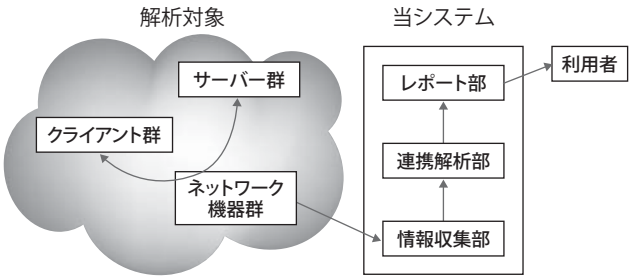

図1. 提案システムの概要

3.2. 情報収集部

情報収集の方式として，スイッチング・ハブからパケットのコピーを取り出す方式［9］がある．これはネットワーク機器に内蔵された物理層の LSI で処理するため，ネットワーク機器の CPU 負荷増大を考慮する必要がない．この機能は企業用スイッチング・ハブに一般的に装備されている機能であり，外部の連携状態の解析システムとインターフェースする上で有用な機能となる．

そこで先行技術の課題（5）を解決するパケット・キャプチャー方式の情報収集部のブロック・ダイヤグラムを図 2 に示す．

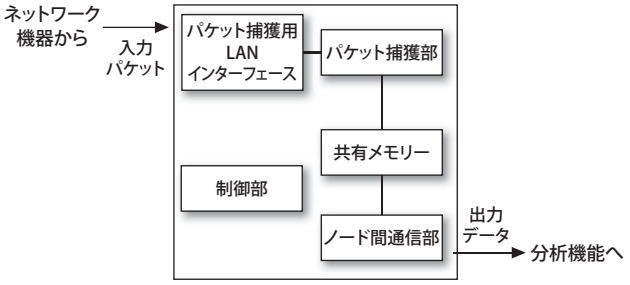

図2. 情報収集部のブロック・ダイヤグラム

図 2 中のパケット捕獲部は，先行技術の課題（7）の解決のため，受信したパケットの必要な部分だけを選別して共有メモリーへ保存する．

ノード間通信部は，先行技術の課題（6）の CPU 負荷によるパケット取りこぼしを防止するために，解析部以降の処理を外部のハードウェアで実行可能なように，データ転送機能などを提供する．

入力パケットの例

イーサネット・ヘッダー	IPv4 ヘッダー	TCP ヘッダー	データ	FCS
(22byte)	(20～28byte)	(20～28byte)	(～1,500byte)	(4byte)

イーサネット・ヘッダー	IPv4 ヘッダー	UDP ヘッダー	データ	FCS
(22byte)	(20～28byte)	(8byte)	(～1,500byte)	(4byte)

出力データの例（1レコード）

あて先IPアドレス	あて先ポート番号	送信元IPアドレス	送信元ポート番号	プロトコル番号	TCPフラグ	パケット長
(4byte)	(2byte)	(4byte)	(2byte)	(2byte)	(1byte)	(2byte)

図3. 情報収集部の入出力

情報収集部の入力パケットと出力データの概念を図3に示す．この出力データは，共有メモリーに蓄積された複数レコード分を一つの伝送単位として連携解析部へ提供する．

3.3. 連携解析部

一般にサーバー上で稼働するソフトウェアとしてのサービスは，対応するTCPポートなどがアサインされている．例えば，Webサーバーは TCP ポート 80 番，ssh サーバーは，TCP ポート 22 番のように，サービスごとに受付窓口を分けて，クライアントからの要求を待っている．要求を受けると処理をして，クライアントへ結果を返信するといった具合にサービスを提供する．

サーバーの中で稼働するソフトウェアとしてのサービスは，クライアントとしてほかのサーバーへ処理を依頼することもある．例えば，Web アプリケーション・サーバーは，データの取り出しと保存のためにデータベース・サーバーのサービスを利用する．このようにサービスを提供するサーバーが，他サーバーのサービスをクライアントとして依頼することもある．

先行技術の課題（1）～（4）を解決し，クライアントとサーバーの連携関係を求める処理概要について図4に示す．

情報収集部から受けたデータは，図4（1）でセッション表を作成する．セッション表とは，図5に示すキーとデータからなる表である．ここで，対向する通信相手の一端をA，もう一端をBとして表記している．これにより情報収集部から送られた個々のパケットの情報は，セッションを成すAとB間のAからBへ送信量，BからAへ送信の集計値とする．

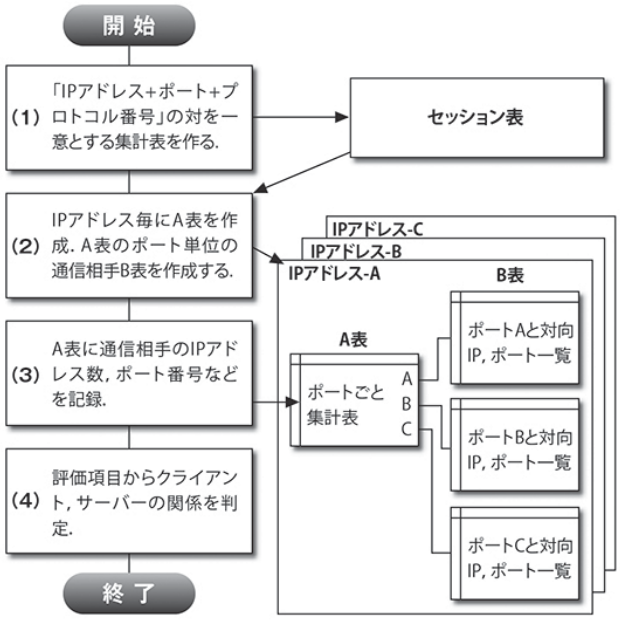

図4. 連携解析部の処理概要

ここで，IPプロトコルのヘッダーには，送信先IP，送信元IPの順で配置されている．そのため，IPアドレスAとIPアドレスB間の通信パケットを収集すると，そのログには，IPアドレスAとIPアドレスBが時系列に交互に現れるように見える．

このログから一意にセッションを識別できるキーを作るために次の処理を行う．

IPv4のIPアドレスは，4オクテットからなり，32ビットの処理系では，符号なし整数型として扱われる．これを利用して大小値の比較を行い，IPアドレスが小，大の順に並ぶようにキーを作成する．これに伴って，データ部分には通信方向のAからBへ，または，BからAへ

キー	A-IPアドレス	A-ポート番号	B-IPアドレス	B-ポート番号	プロトコル番号

データ	A→B パケット数	A→B バイト数	A←B パケット数	A←B バイト数	A→B フラグ計数値	A←B フラグ計数値

図5. セッション・データ

の送信パケット数，バイト数のカウンターを設ける．送信の向きに応じたカウント処理を行う．

図５のデータ部分の「A → B フラグ計数値」「A ← B フラグ計数値」は，TCP ヘッダーの SYN，ACK，FIN，RST の TCP フラグごとに設けたカウンターである．そのフラグのビットが 1 の場合，増加する．

図４（2）では，セッション表を読み取って現れた IP アドレスごとに，プロトコル番号とポートの一覧からなる A 表を作成する．さらに，通信相手の IP アドレスとポート番号の一覧である B 表を作成する．

次の図４（3）で，B 表を集計することで対向する IP アドレスの出現数，ポート番号の出現数を求め A 表へ記録する．

最後に図４（4）では，図４（3）で記録された結果に基づき，A 表に記録されたポート番号とプロトコル番号が，後述する特徴から，サーバーかクライアントであるかの判定を行う．

この特徴の基礎となるクライアント・サーバー・モデルは，コンピューター・ネットワークにおけるソフトウェアのモデルで,サーバー（サービス提供者）とクライアント（サービス要求者）が対となって機能する．このモデルの基礎となる TCP 通信では，このソフトウェアのクライアントとサーバーは，IP アドレスとポート番号のペアにより，対を識別して相手と交信する．

このソフトウェアのサーバー（サービス提供者）は，一定のポート番号と結び付いて，クライアントからの要求を受け付ける．一方，クライアントはサーバーと接続する際に，空いているポート番号から任意に割り当てられるために，クライアントのポート番号は一定しない．

TCP 通信では，サーバーとクライアントが，対の関係を確立する際に，パケットに特徴的なフラグを付加する．一方，UDP 通信では特徴的なフラグを付加しない．しかし，一般にサーバーは，数多くのクライアントからの要求を受ける．

このような特性をどの程度持っているのかの評価を行い，どちらの特徴が強いかでサーバーであるか，クライアントであるか関係を識別する．

以下に，サーバーの通信の特徴例を挙げる．

(1) TCP プロトコルで，SYN，ACK を送信するポートは，サーバー・ポートである．
(2) 多数の IP アドレスと通信するポートは，サーバー・ポートである．
(3) 相手ポート番号が複数である場合，そのポートはサーバー・ポートである可能性が高い．

同様に，クライアント・ポートの特徴例も挙げる．

(1) クライアント・ポートは，SYN，ACK フラグの付加された TCP パケットを受信する．
(2) 相手の IP アドレスは 1 個であることが多い．
(3) 相手のポート数は 1 個である．

3.4. レポート部

クライアントとサーバーの連携の実態を本システムでは，**表 1** の形式で Web ページのレポートとして提供する．

表1. 連携実態レポートの例

IP アドレス	192.168.1.11	
ホスト名	CommonWebService1	
外部へ提供するサービス（サーバー）一覧		
プロトコル	クライアント数	通信量
TCP/ssh	12	10M
TCP/http	4,500	12.5G
TCP/50020	301	30M
外部へ要求するサービス（クライアント）一覧		
プロトコル	サーバー IP	通信量
TCP/jdbc	192.168.1.13	975M
TCP/mq	192.168.1.200	200M

上記例では，IP アドレス（ホスト）ごとに，外部へ提供するサービス（サーバー）の一覧と外部へ要求するサービス（クライアント）の一覧を提供する．ホスト名と IP アドレス，ポート番号とプロトコル名の対応表を保有し，変換表示する機能があると，より分かりやすいレポートとなる．

4. 提案システムの実装

図１に示したシステム概要図をデータ量と計算量の観点から概念図としたものが図６となる．網掛け矢印の大小はデータ量を表し，四角の大きさは，計算量の多さを表した．

図６に表した計算量とパケット量の関係および下記に挙げる理由から，情報収集部は連携解析部やレポート部から独立した専用ハードウェアとして構成できることが求められる．

(1) パケットの取りこぼしが発生しないように，連携解析部とハードウェアを分離するなどの CPU 使用率を低く抑える考慮が求められる．

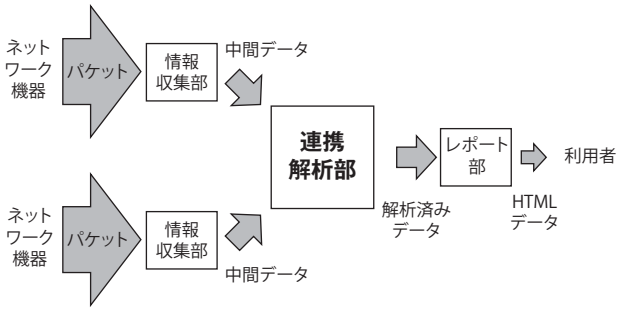

図6. データ量と計算量

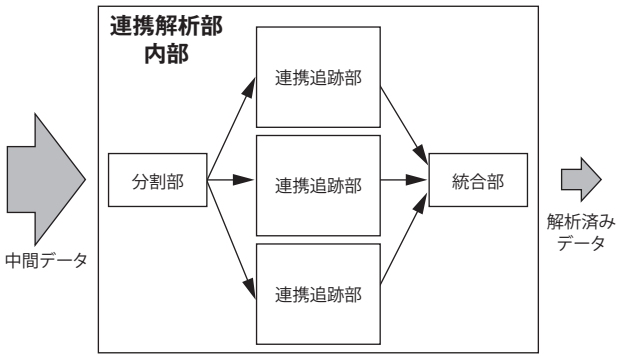

図7. 連携解析部の内部構造

(2) 基幹のネットワーク機器はハードウェア障害に備えて二重化することから，2 つ以上の収集部を構成することが求められる．
(3) ネットワーク機器からのケーブル長の範囲に設置することが求められる．

連携解析部の実装では，3.3 章で詳述した処理を実行することから高い処理能力を要求する．特に DDoS（Distributed Denial of Service: 分散型サービス停止）攻撃の検知などの用途では，リアルタイム性を要求されることから，処理能力は重要となる．

連携解析部の処理は，IP アドレスの範囲で分担して並列処理が可能である．そこで，図 7 に示すように，データ分割部とデータ統合部を設け，特定の IP アドレス範囲を連携解析する部分を連携追跡部として，サンドイッチする構造とする．

分割部は，アドレス範囲で分割したデータを連携追跡部へ供給する．統合部は，分割して求めた結果を再統合し，レポート部へ供給する．

この構造により連携解析部のハードウェアが，マルチ CPU 方式などの場合には，連携追跡部のプロセスを複数稼働させ，処理能力向上を図る．さらに，ハードウェア 1 台の処理能力を超える連携解析処理が必要な場合は，複数のハードウェアで分担して処理することを可能とする．

一方，監視対象規模が小さい場合は，本システムを持ち歩いて対象機器に接続して，その場でレポートを取得できることが求められる．

これら相反する要件を満たすために，図 8 に示す基本ユニットを考案した．

この基本ユニットは，すべての機能を搭載し，パケット

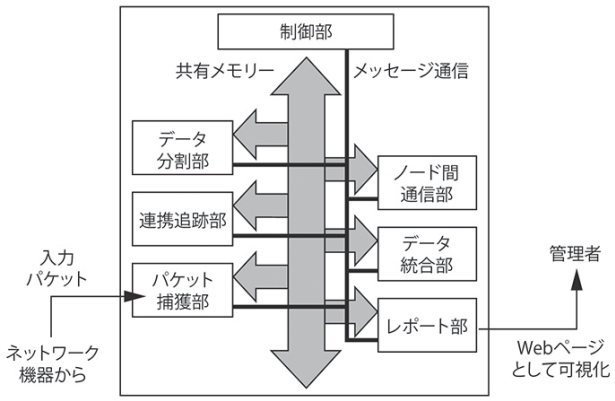

図8. 本システムの基本ユニット

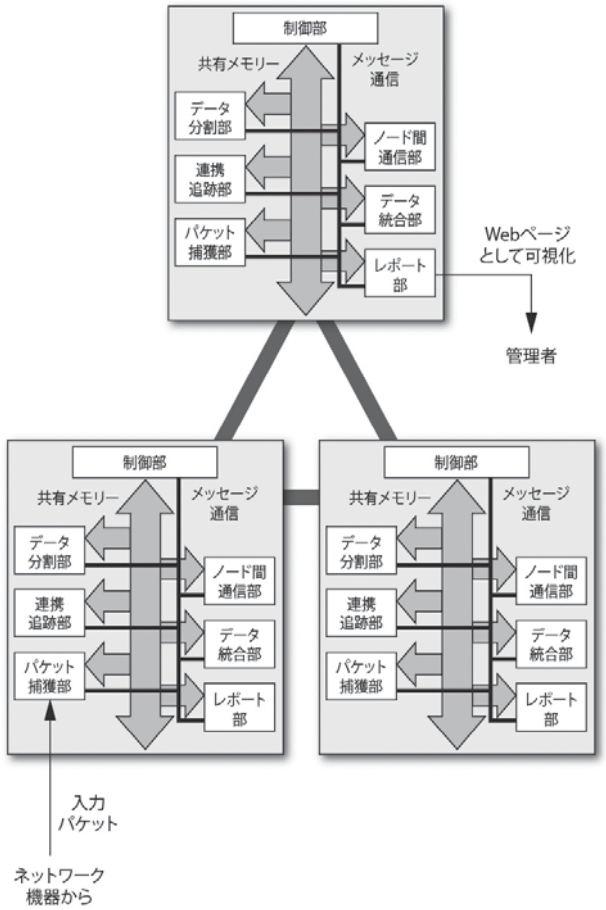

図9. 基本ユニットを組み合わせ例

の収集からレポート出力までを実行できる．制御部とノード間通信部の機能により，複数の基本ユニットで，処理を分担することも可能である．**図９**に 3 個の基本ユニットで，収集，解析，レポーティングを分担する例を示す．

レポート部は，実際に実装した Web 画面と遷移図を**図 10** に示す．

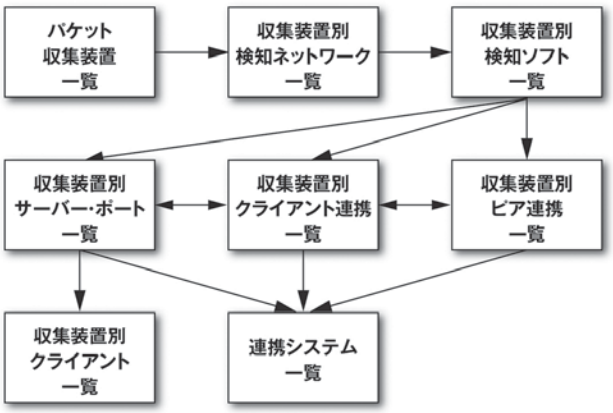

図10. レポート部の画面遷移図

5. 適用事例

表２に示す機器構成で，プロトタイプ・システムを開発し，実システムの環境に適用した．

表2. システム実装概要

情報収集部 PC	7 台　Pentium4 3GHz メモリー 2GB Linux
連携解析部 PC	5 台　Pentium4 2.4GHz メモリー 1GB Linux
レポート部 PC	1 台　Pentium4 3GHz x2 メモリー 1GB Linux
開発言語	Perl 5.8, gcc
データベース	sdbm（the clone Berkeley ndbm）
メッセージ通信	Socket, SystemV IPC
パケット収集	Libpcap-0.9.1

適用環境は，**図 11** のように全国の販売会社の販売業務サーバー群が集約されている部分と基幹ネットワークを接続する部分に対して，ポート・アナライザー機能［8］を設定して，本システムを接続し実測を行った．

約 1 時間の測定で**表３**に示すように，全国約 16,000 の PC クライアントおよび生産業務や会計業務などのほか，サーバー 400 台弱の連携関係を解析することが可能となった．わずか 1 時間の測定であったため，その間に通信を発生しなかったものは含まれていない．連続的に計測できる時間が長くなれば解析結果の

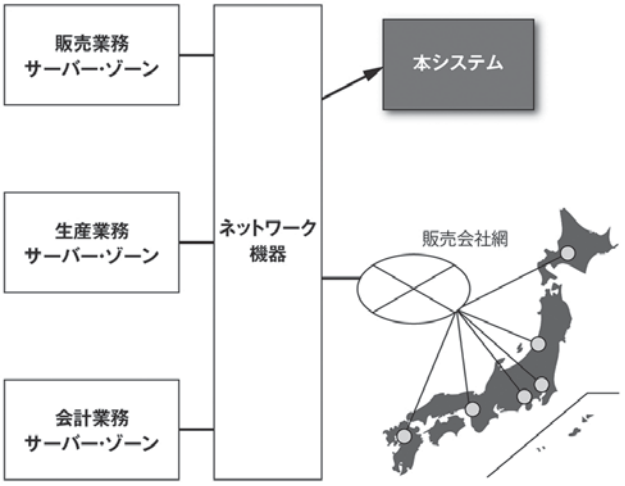

図11. 適用環境の概要

表3. 測定結果サマリー

測定時間	1 時間 1 分
検知した総データ通信量	26 GB バイト
連携を検知したネットワーク数	285 ネット・アドレス
稼動を検知した PC 数	16,075 台
稼動を検知したサーバー数	382 台

表4. 長時間稼働

測定時間	857.7 時間（約 36 日間）
検知した総データ通信量	4.7TB バイト
蓄積された解析データ量	2.1G バイト

正確度が向上する．

長時間測定した場合，解析結果を蓄えるデータ量の増大が懸念されるが，**表４**のように解析データ量が，PC のディスク・ドライブの容量以下に収まることが確認された．

本検証で得た結果レポートは，セキュリティー上の観点から記載を控えるが，**表１**を詳細化した内容となっている．以上，この方式が短時間測定での解析範囲と長時間測定での解析データ量の収束において，極めて有望であることが確認された．

6. おわりに

既存のサーバーとネットワーク機器へ影響を与えない方式で，パケットを収集し，そのパケット中の特徴的な情報に着目し，サーバー間の連携状態を解析するツールを提案した．本手法を実装したプロトタイプ・システムを

開発し，大規模な実システム環境の課題解決に当たった結果，システム間連携の稼働実態を容易に把握できることが確認された．また，長期間の計測においても蓄積される解析データ量の収束が確認され，長期間の測定が可能なことが確認された．

本手法は，サーバー障害やデータ・センターなどの拠点の被災を想定した影響度分析ツールや，不正パケットによりサービスを妨害する種のコンピューター・ウイルスの検知ツールなどへの応用が可能である．

業務システムは，経営環境の変化に迅速に対応できるように，SOA（Service Oriented Architecture: サービス指向アーキテクチャー）の概念を取り入れた形態へ進化を続けている［10］．このようなサーバー間の連携の高度化に伴い可用性や信頼性をサポートするツールとしても期待される．

今後の発展として，蓄積した解析データをさまざまな視点から分析するために，データベースの活用を検討したい．

謝辞

本システムの実システム適用にあたり，環境を提供いただきました大手自動車株式会社のご担当者様，また，計測作業に協力くださいましたソフトバンクテレコム株式会社，アイ・ネット・リリー・コーポレーション株式会社のご担当者様に，あらためて深謝いたします．

参考文献

［1］（独）情報処理推進機構 : “高トラフィック観測・分析法に関する技術調査 ,” http://www.ipa.go.jp/security/fy15/reports/traffic_mon/index.html（2004）．
［2］横山 峰明 : ネットワーク監視装置 , 特開平 7-321783（1995）．
［3］丹野 秀和 : トラフィック分類装置およびトラフィック分類方法 , 特開 2002-261799（2002）．
［4］P. Phaal,S. Panchen, and N McKee:InMon Corporation’s sFlow: A Method for Monitoring Traffic in Switched and Routed Networks, RFC 3176（2001）．
［5］B. Claise: Cisco Systems NetFlow Services Export Version 9, RFC 3954（2004）．
［6］K. Claffy, G. Polyzos, and H.-W. Braun: “Application of Sampling Methodologies to Network Traffic Characterisation,” Proc. ACM SIGCOMM'93, pp.194-203（1993）．
［7］Cisco Systems Inc.: “NetFlow によるパフォーマンス分析 ,” ホワイトペーパー（2004）．
［8］Cisco Systems Inc.:” Configuring the Catalyst Switched Port Analyzer（SPAN）Feature” http://www.cisco.com/warp/customer/473/41.html
［9］Luca Deri:” Improving Passive Packet Capture: Beyond Device Polling,” http://luca.ntop.org/Ring.pdf（2004）．
［10］田中秀一郎 , 西澤茂隆 , 田中章弘 , 中村匡秀 , 松本健一 : “SOA システム構築のための既存システムの再利用性評価 ,” 電子情報通信学会技術研究報告 , Vol.106, No.578, pp.471-476（2007）．

日本アイ・ビー・エム株式会社
シニア ITアーキテクト

高良 真穂 Maho Takara

［プロフィール］
2002 年日本 IBM 入社．システム・エンジニアとして，多くのお客様のプロジェクトに参画．主に，テクニカル・リーダーまたは，プロジェクト・マネージャーとして，お客様プロジェクトに従事．