

階層型 P2P ネットワークを用いた インターネット計測基盤の性能評価と 実展開シナリオの考察

益井賢次^{†1} 門林雄基^{†2}

自律性の高いネットワークアプリケーションにとって、インターネットを構成するノードやリンクといった要素の性質を示すネットワーク特性は、自身の将来の動作を決定するための重要な情報である。アプリケーション指向のネットワーク計測基盤 (AOMP) は、このようなアプリケーションに対してネットワーク特性を提供するサービスとして構築されるネットワーク計測基盤である。本論文では AOMP の一設計として、計測ノードの性能に応じて計測エージェントの役割を core と stub の 2 種類に分け、エージェントが構成する階層型 P2P ネットワークを計測ネットワークとして利用する計測基盤 N-TAP を提案する。N-TAP は、サービス応答性と安定性・計測ノード間の負荷分散公平性・収集可能範囲の網羅性・提供される情報の正確性といった AOMP の要件をふまえて設計されており、その計測ネットワークは協調計測手法の実装を容易にする共有記憶領域およびエージェント探索の機能を備える。階層型 P2P ネットワークを用いた AOMP の性能に関する基礎的知見を得るため、PlanetLab や StarBED 上で N-TAP を大規模に展開した実験を行い、エージェントに発生する処理の応答性の特性やエージェント間の処理負荷の分布などについて検証する。また、AOMP の実展開のシナリオを提示し、実展開時の課題を考察する。

An Application-oriented Measurement Platform Built on Role-based P2P Network: Performance Evaluation and Deployment Scenarios

KENJI MASUI^{†1} and YOUKI KADOBAYASHI^{†2}

Autonomic network applications often utilize network characteristics for the decisions on their future behavior. For such applications, an application-oriented measurement platform (AOMP) works as an independent network service that collects and provides network characteristics. In this paper, we propose a novel architecture of AOMP, named “N-TAP.” N-TAP is built on

role-based P2P network, which two types of monitoring agents construct. We designed N-TAP to meet some requirements for AOMPs including service responsiveness and stability, load distribution among monitoring nodes, measurement coverage, and the accuracy of provided network characteristics, and its measurement network provides the features of shared storage and agents search so as to ease the implementation of cooperative measurement methodologies. To acquire the fundamental characteristics of AOMPs built in this manner, we conducted the performance evaluation of our implementation on PlanetLab and StarBED, and explored the tendencies of the responsiveness of respective agents' procedures and the load distribution among agents. Based on the experimental results, we also discuss its deployment scenarios and open issues for the actual deployment of AOMPs.

1. ま え が き

インターネットの利用形態の多様化にともない、インターネットが提供する抽象化されたエンドノード間の通信機能のみでは、自律性の高いネットワークアプリケーションの目的を満足に達成できない状況が見受けられるようになった。インターネットは不均一かつ動的に変化する性質を持ったノードとリンクから構成されており、パケットのルーティングはインターネットを構成するルータなどが処理し、また誤り訂正や流量制御などは各レイヤの通信プロトコルに基づいて行われる。アプリケーションに提供される簡便な通信インタフェースであるソケットは、このようなネットワークの性質やその上で行われる複雑な処理を隠蔽する。従来のアプリケーションはノード間の単純な通信のみで処理が完結していたため、ソケットインタフェースによる通信のみでアプリケーションの目的を達成することができた。一方で昨今、オーバーレイネットワークアプリケーションに代表される、自律性を持つ大規模な分散型ネットワークサービスの展開が進んでいる。ここでの自律性とは、アプリケーションが自身のサービスの規模拡張や維持を目的として、ネットワークの状況の判断とそれに追従した処理を行う性質を指す。ネットワークアプリケーションが自律的な行動を行うための判断材料の 1 つがネットワーク特性である。本論文ではネットワーク特性を、インターネットを構成するノードやリンクといった要素の性質を示す情報であると定義する。ネットワー

^{†1} 東京工業大学
Tokyo Institute of Technology

^{†2} 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

ク特性には、2 ノード間の通信時の往復遅延時間 (Round-Trip Time; RTT) や実効帯域幅などの性能指標、あるいはネットワークの IP トポロジや AS トポロジといった構成情報が含まれる。Peer-to-Peer (P2P) ネットワークの構成手法は、こうしたアプリケーションの自律的な処理の 1 つとして、さかんに研究されてきた。たとえば、分散ハッシュテーブル (Distributed Hash Table; DHT) の一実装である CAN¹⁾ では、オーバーレイネットワーク上でのルーティング遅延を小さくするために、ランドマークノードとの間で計測した RTT を近接性指標として、オーバーレイノードの ID 割当てに反映させている²⁾。Pastry^{3),4)} では、ルーティング時のノード選択において使用できる近接性指標として、対象ノードとの間の IP ホップ数や RTT をあげている。また、こうしたネットワーク特性の利用は、P2P ネットワーク上のコンテンツ配信における低遅延で高速な配信木の構成や冗長経路の確保の場面でも見受けられる⁵⁾。現在のインターネットには、広域にわたるネットワーク特性を提供する標準的な機構が用意されていないため、各アプリケーションが独自に情報収集を行うことになる。

このような流れを受け、ネットワーク特性を収集するための計測処理を 1 つの独立したネットワークサービスとして展開しようとする動きがある⁶⁾⁻⁸⁾。これは、インターネット広域にわたるネットワーク計測基盤を構築し、前述のような大規模アプリケーションを含むさまざまなアプリケーションからネットワーク特性の収集要求を受け付け、収集した情報を提供する、といった形態のサービスである。本論文では、このようにアプリケーションでの利用を前提としたネットワーク特性の収集・提供サービスを行う計測基盤を、アプリケーション指向のネットワーク計測基盤 (Application-Oriented Measurement Platform; AOMP) と呼ぶことにする。これまでの研究で、いくつか存在する AOMP の実装について、主としてその上で用いられる計測手法の性能評価が行われてきた。AOMP の実環境への展開に向け、これに加えてシステムアーキテクチャ面での機能要件や性能指標についての考証が重要となる。

本論文では、計測ノード間で階層構造の P2P ネットワークを形成して、それを計測ネットワークとしてノード間の協調などに用いる計測基盤を、AOMP の要件を満たすシステムのアーキテクチャの一形態として提案する。本方式の特徴として、基盤上でノード間が協調する計測手法を実現しやすいこと、不安定なノードも計測専用ノードとして参加可能であること、アプリケーションが単純なインタフェースを通じて容易に広域のネットワーク特性の収集が可能になることがあげられる。このような計測基盤の展開シナリオを示し、実環境での性能評価実験の結果を通じて、その実現可能性や問題点について論じる。

本論文の構成は、以下のとおりである。2 章では、AOMP の機能要件をまとめる。3 章

では、N-TAP の設計と実装・展開シナリオを示す。4 章では N-TAP を用いた性能評価実験の結果を示し、5 章ではそれに基づき提案システムの実展開の可能性について論じる。最後に、6 章で本論文の成果をまとめる。

2. アプリケーション指向のネットワーク計測基盤 (AOMP) の機能要件

AOMP は、アプリケーションへのネットワーク特性の提供という目的を持ち、これは従来のネットワーク計測基盤の目的とは異なるため、機能要件を新たに検討すべきである。本章ではこの機能要件についてまとめる。

ネットワーク特性の統計解析を目的とした従来の計測基盤は、多くの場合、あらかじめ用意されたネットワーク基盤の上で、固定された計測対象について中長期にわたるネットワーク特性の収集活動を行い、収集した情報を分析することで目的の情報を導出する。一方で AOMP には、このような計測基盤とは異なる以下のような性質が求められる。まず、AOMP を利用するアプリケーションは収集した情報を自身の将来の処理の判断基準として用いることがあるので、その処理を待たせないためにも、AOMP は目的の情報を迅速に提供する必要がある。こうした高い応答性を確保すると同時に、インターネット上の多様な環境に広く展開される大規模サービスとして、負荷分散や耐障害性の向上などについて考慮しなければならない。また、AOMP に要求されるネットワーク特性の内容はアプリケーションに依存するため、計測ノードや計測対象は固定ではなく動的に変化する。このため、AOMP はアプリケーションが計測対象とする範囲を網羅できるよう、アーキテクチャ面で考慮する必要がある。また、極端に誤ったネットワーク特性が提供されることで、提供される情報を判断基準とするアプリケーションが本来目的とした動作を実現できなくなる可能性がある。したがって、特定時点でのネットワーク特性の正確性について、AOMP は十分考慮しなければならない。

これらの用途をふまえた考察から、以下のように AOMP の機能面での要件を定義する。

- 応答性 計測サービスの応答時間は短いほど好ましい。
- 負荷分散公平性 計測サービスの維持に必要な処理が特定ノードに集中すべきではない。
- 安定性 非均一な性質の環境をまたぐサービスが全体として安定動作することに配慮すべきである。
- 収集可能範囲 アプリケーションの必要とするネットワーク特性を可能な限り収集可能なものとして網羅すべきである。
- 正確性 提供される情報が特定の時点でのネットワーク特性を正しく示しているか、考慮す

べきである。

ただし、正確性や収集可能範囲はネットワーク特性の収集方式に強く依存するため、これらの指標が高い方式の実装を支援する機能を計測基盤側で提供することも重要となる。

3. N-TAP の設計と実装

2 章では AOMP の機能要件を定義した。本章では、このような要件を満たす AOMP のアーキテクチャとして、階層型 P2P ネットワークを計測ネットワークとして利用する分散計測基盤 N-TAP を提案し、その設計と実装、そして展開シナリオについて記す。

3.1 設計

N-TAP を構成する要素は、計測ノード上で動作するプログラムである N-TAP エージェントと、エージェント群が協調してネットワーク特性の収集活動を行うために構成される N-TAP ネットワークの 2 つである。以下では、それぞれの設計について述べる。

3.1.1 N-TAP エージェント

N-TAP エージェントは、主に 4 つのコンポーネントから構成される。これらのコンポーネントを列挙すると、通信サーバコンポーネント (Communication Server)、ネットワーク特性収集コンポーネント (Network Characteristics Collector)、N-TAP ネットワーク管理コンポーネント (N-TAP Network Manager)、ネットワーク特性データベース管理コンポーネント (Network Characteristics Database Manager) となる。通信サーバコンポーネントは、N-TAP のサービスを必要とするアプリケーションや他のエージェントとの間で情報を授受するための通信機能を提供する。これは、AOMP の目的であるアプリケーションへのネットワーク特性の提供を実現するために必要な機能である。また、後述する N-TAP エージェントどうしの協調動作においてもエージェント間で情報授受が必要となるため、この機能が必要となる。ネットワーク特性収集コンポーネントは、目的のネットワーク特性を実際に収集するさまざまな計測手法を実装したコンポーネントである。N-TAP ネットワーク管理コンポーネントは、N-TAP ネットワーク (3.1.2 項で詳述) を構成する処理を担当する。ネットワーク特性データベースコンポーネントは、収集したネットワーク特性の記録・読み出し機能を提供し、ネットワーク特性を保持するために計測ノード上に確保された記憶領域を内包する。

各コンポーネントの関係を図 1 に示す。通信サーバコンポーネントは、受け取ったメッセージに応じてタスクを他のコンポーネントに割り振る。アプリケーションが通信サーバコンポーネントに対して送信するメッセージはネットワーク特性収集の要求のみで、このタス

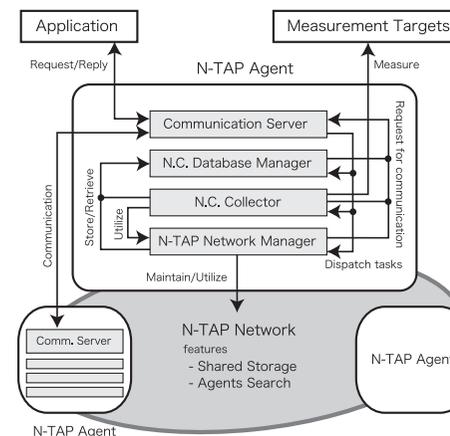


図 1 N-TAP エージェントのアーキテクチャ
Fig. 1 Architecture of the N-TAP agent.

クはネットワーク特性収集コンポーネントに引き継がれる。エージェントが他のエージェントに送信するメッセージは、ネットワーク特性の収集要求・ネットワーク特性データベースの操作要求・N-TAP ネットワークの操作要求の 3 種類で、それぞれネットワーク特性収集コンポーネント・ネットワーク特性データベース管理コンポーネント・N-TAP ネットワーク管理コンポーネントにタスクが割り振られる。また、これら 3 つのコンポーネントは、タスクの結果や前述の種類の要求メッセージの送信を、通信サーバコンポーネントに依頼する。ネットワーク特性収集コンポーネントは、収集したネットワーク特性の記録、あるいは過去に収集したネットワーク特性の参照のため、ネットワーク特性データベース管理コンポーネントを利用する。N-TAP ネットワーク管理コンポーネントは、N-TAP ネットワークが提供する機能を利用するためのインタフェースを備えており、ネットワーク特性収集コンポーネントがそれを利用する。さらに N-TAP ネットワーク管理コンポーネントは、N-TAP ネットワークに関する情報を記録するためにネットワーク特性データベースを利用する。

3.1.2 N-TAP ネットワーク

N-TAP ネットワークは、N-TAP エージェント間の協調を実現するための抽象化された機能を提供する。ここでは協調計測手法に着目し、N-TAP はその実装を支援するような機能を提供するものとする。協調計測手法とは、計測処理が単一ノードのみで完結せず、特定のアルゴリズムに従って他のノードと相互に作用しながら計測処理を行う方法である。協調

計測手法は、多数の計測ノードがネットワーク特性を収集する処理を効率化することから、AOMP が想定するような大規模な計測活動に適するとされている。代表的な協調計測手法には、Doubletree⁹⁾ のような他の計測ノードの収集した情報を参照することで自身の収集活動を効率化するものや、Vivaldi¹⁰⁾ のように特定のネットワーク特性の推測モデルのもとで計測ノード間での情報交換と統計処理をして、一部のネットワーク特性を実際の収集活動を行わず推測可能とするものがあげられる。

2章で述べたように、AOMP は計測基盤として、大規模計測に適した協調計測手法の実装を容易にするような機構を備えることが望まれる。協調計測手法では一般に、計測ノード間の情報共有と、他の計測ノードの発見と通信の機能が要される。したがって、N-TAP ネットワークは具体的に、ネットワーク特性を保持するためのエージェント間の共有記憶領域 (Shared Storage) と、他のエージェントの探索機能 (Agents Search) の 2 つの機能を、計測手法の実装者に対して提供することとする。

3.1.3 エージェントの役割分割と階層型計測ネットワーク

実インターネット環境での展開における安定性の要件を考慮して、N-TAP ではエージェントの役割に core エージェントと stub エージェントの 2 つを用意し、エージェントが動作する計測ノードの性質に従ってその役割を割り当てるものとする。core エージェントは、N-TAP ネットワークの提供する機能の維持を行うエージェントである。core エージェントの構成する N-TAP ネットワークの一部を、core ネットワークと呼ぶことにする。一方で、stub エージェントはこのような処理は行わず、N-TAP ネットワークの機能を core エージェントを経由して利用することとする。すなわち N-TAP ネットワークは、core ネットワークの層と、core エージェントにスター型に接続された stub エージェント群から構成される、階層型の計測ネットワークとなる。なお両方のエージェントとも、計測処理を行ったり、計測手法の実装者に対して N-TAP ネットワークが提供する機能を利用するための同一のインタフェースを提供したりする。

この役割分割を行う理由は、AOMP が想定するインターネット上の計測ノードの非均一性にある。AOMP のような計測基盤上の計測ノードは、必ずしもノード自身あるいは接続リンクの性能が一定であることが保証されない。したがって、すべての計測エージェントが同一の役割で N-TAP ネットワークの維持・管理に関する処理を担当すると、処理性能の遅いノード上で動作するエージェントが N-TAP ネットワークの提供する機能の性能上のボトルネックになったり、頻繁に故障するノードが原因で N-TAP ネットワークの構成を維持する処理の負荷が高まったりすることが考えられる。このような状況は、高速リンクに接続さ

れ安定して動作する処理能力の高い計測ノードでは core エージェントを動作させ、残りのエージェントを stub エージェントとする、といった役割分割を行うことで改善できることになる。

3.2 実装

これまでの要求と設計をふまえ、我々は N-TAP エージェントを実装し、オープンソースソフトウェアとして公開している^{*1}。ここでは、実装に用いた手法などについて述べる。

N-TAP の core ネットワークは、DHT の一実装である Chord¹¹⁾ に基づいて構成される。これは、大規模に展開される AOMP の性質を考慮し、規模拡張性を確保する目的によるものである。各 N-TAP エージェントには 128 ビットのエージェント ID が割り当てられ、これが N-TAP ネットワーク上でのエージェントの一意的識別子となる。エージェント ID には原則としてランダムな値を割り振ることで、core ネットワークの維持処理にともなう負荷が特定のエージェントに集中することを防ぎ、機能要件の「負荷分散公平性」を満たすようにする。core エージェントは core ネットワークの維持・管理処理として、Chord でのルーティングテーブルを管理し、任意の ID に対する担当エージェント (Chord リング上での順方向に、最も近いエージェント ID を持つもの) の発見機能を実現する。N-TAP ネットワークの提供する機能の 1 つ、共有記憶領域は DHT の性質を用いて実装される。収集したネットワーク特性を含むデータ項目には、索引となるデータの種別が定められていて、索引対象のそれぞれのデータに対して 160 ビットの SHA-1 ハッシュ値を計算し、その最初の 128 ビットをとって元のデータ項目のエントリ ID とする。共有データ項目を保存する際には、そのエントリ ID の保持担当である core エージェントを発見し、そのエージェント上のネットワーク特性データベースにデータ項目を記録する。すなわち、各エージェントのネットワーク特性データベースは、共有記憶領域の一部となる。このようにして記録されたデータ項目は、索引となる種類のデータから同様にハッシュ値を計算して担当ノードを発見することで到達可能となり、読み出すことができる。たとえば、RTT に関する項目は (収集時刻, 送信元 IP アドレス s , 宛先 IP アドレス d , 収集方法, RTT, 収集エージェント ID) といった形式をとり、送信元 IP アドレスと宛先 IP アドレスが索引情報として指定されている。この場合、上記のエントリ ID 生成関数を f とすると、 $f(s)$ と $f(d)$ それぞれの担当エージェントに共有データ項目が保存され、それを参照する際はこの ID 値のうちの 1 つを計算したうえで担当エージェントを発見すればよい。また、N-TAP ネット

*1 <http://www.n-tap.net/>

ワークのエージェントの探索機能は、この共有記憶領域機能を利用して実現されている。どの N-TAP エージェントも N-TAP ネットワーク参加時に、自身の IP アドレスや待ち受けポート番号などを索引に持つ共有データ項目を共有記憶領域に保存する。これにより、あるエージェントは他のエージェントをこれらの索引情報から探索し、相互の通信に必要な情報を得られるようになる。他のエージェントを探索する必要があるシナリオの一例として、最も単純な協調計測である「他のエージェントへの traceroute の依頼」があげられる。エージェントがアプリケーションから 2 点間の IP トポロジの収集を依頼されたとき、要求を受け付けたエージェントが指定の 2 点のうちの始点ノード上で動作していなければ、traceroute の手法の制限上、目的の情報は収集できない。そこで、始点ノードの IP アドレスをキーとして共有記憶領域内を検索することで該当ノード上でエージェントが動作しているかを確認し、存在を確認できればそのエージェントの IP アドレス・待ち受けポート番号などの情報を取得して traceroute の処理を依頼することになる。

そのほか、応答性の改善のために、エージェントは“local-first and remote-last”ルールに基づいて収集活動を行う。これは、アプリケーションからの要求を受け付けた段階で、以下の順に収集活動を行うものである。

- (1) 自身のネットワーク特性データベースに要求された情報が存在するか検索し、存在すればそれを利用する。
- (2) 自身で要求された情報を収集可能であれば、収集して利用する。
- (3) 該当情報が他のエージェントによりすでに収集されていないか、共有記憶領域を検索し、存在すればそれを利用する。
- (4) 共有記憶領域に記録されたエージェントリストから要求された情報を収集できるエージェントが存在するか検索し、存在すれば該当エージェントに収集を依頼する。
- (5) 上記のどれにもあてはまらない場合、N-TAP が収集不可能なネットワーク特性であることをアプリケーションに通知する。

このルールは、さまざまなネットワークに分散するエージェント間の通信が処理時間のボトルネックになることを考慮して、可能な限り自ノードで完結できる処理を優先して行うものである。なお、アプリケーションは要求の際、情報鮮度という指標を指定できる。情報鮮度は現在時刻からどれくらい前までの情報を利用可能なものとして許容するかを示す量で、秒単位で指定する。鮮度に大きな値を指定すれば、要求したエージェントのネットワーク特性データベースに該当情報が存在する確率が高くなるため応答性が高まることが期待できるが、鮮度の古い情報は現状のネットワーク特性を反映していない可能性があるというリス

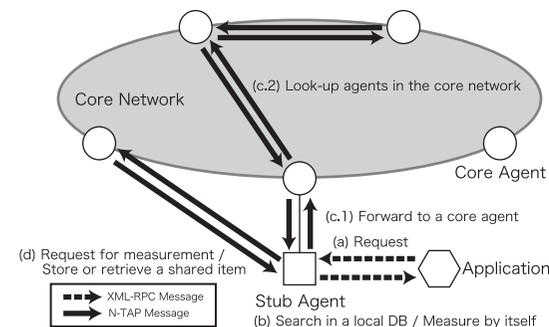


図 2 ネットワーク特性提供までの処理の流れ

Fig. 2 The sequence of procedures for providing requested network characteristics.

クもある。すなわち、AOMP の機能要件としてあげた応答性と正確性についてのトレードオフの関係性を、アプリケーションが選択することになる。ところで、計測処理自体の処理時間が長い場合、(2) の処理で待たされるために、(2) よりも (3) を優先した方が全体の応答性が改善されることも考えられる。現時点では、N-TAP は上記の戦略で固定して要求処理を行っているが、計測手法自体の応答性に応じて各手順の優先順位を変化させたり、2 つ以上の手順を同時に実行したりするなどの拡張を行うことで、さらに応答性の改善が期待できると考えられる。

ここまでのまとめとして、アプリケーションからの要求受付からネットワーク特性を提供するまでの処理の流れを、エージェントやアプリケーションの間で交わされる制御メッセージの流れに基づいて図 2 で説明する。なお、アプリケーションからの要求受付には XML-RPC プロトコルを、エージェント間の情報授受には独自のメッセージングプロトコルを用いる。図中 (a) のようにエージェントがアプリケーションから要求を受け付けると、前述の“local-first and remote-last”ルールに基づき、(b) に示す自身のネットワーク特性データベース内の検索や自身での計測処理が実行される。ここまでの目的のネットワーク特性が得られない場合は、共有データ項目中からのネットワーク特性およびエージェント情報の検索が発生する。要求を受け付けたエージェントが stub である場合は、他の core エージェントの発見のため自身が接続している core エージェントに発見処理を依頼し (図中 c.1)、core ネットワーク内での Chord のルーティング処理により目的のエージェントが発見される (図中 c.2)。要求を受け付けたエージェントが core である場合は、(c.1) の手順は必要ない。そして、他のエージェントに計測処理を依頼したり、共有データ項目の保存・

読み出しを行ったりする場合は、そのための依頼メッセージが発生する(図中 d)。これまでの手順の中で、目的のネットワーク特性を取得できた段階で、アプリケーションにはそれが提供されることになる。

現在、N-TAP エージェントに実装されている計測手法は、RTT 計測のための通常の ping と Vivaldi、IP トポロジ探索のための traceroute と Doubletree である。N-TAP エージェントは C++ で記述され、ネットワーク特性データベースには SQLite を用いている。エージェントの動作は Mac OS X、Linux、FreeBSD で確認している。

3.3 展開シナリオ

N-TAP は、インターネット広域にわたりオーバーレイネットワークを構成するアプリケーションからの利用を想定している。このため、構成エージェント数は数十から数万程度になるものと考えている。本節では、このような多数のエージェントを実インターネット環境へ展開するための、実現可能性が高いと考えられる 2 つのシナリオを示す。

3.3.1 管理ネットワーク内の基盤サービスとしての構成

1 つ目のシナリオでは、ISP や企業・組織といったネットワークの管理単位内で以下のようにエージェントを配置する。

- ネットワーク管理者が管理するノード上で core エージェントを動作させる。
- N-TAP のサービスを利用するその他のノードは、自身のネットワーク内の core エージェントに接続する stub エージェントを動作させる。
- N-TAP を利用するアプリケーションと同一ノードでエージェントを動作させ、アプリケーションはそのエージェントに問合せを行う。

このシナリオでは、core エージェントの動作しているノードがネットワーク管理者の管理により安定して動作することが期待できるため、N-TAP ネットワークは安定し、ネットワーク内の基盤サービスとして実現性が高いといえる。この場合、stub エージェントの動作時には接続する core エージェントの IP アドレスや待ち受けポート番号といった情報が必要となるため、ネットワーク管理者はこれを手動もしくは自動で利用者に知らせる手段を用意しなければならない。自動での情報配布の方法には、DHCP の拡張情報として core エージェントの情報を埋め込むことや、DNS の MX レコードのように該当ドメイン内の core エージェントの情報を示す拡張レコードを用意することが考えられる。一方で、core エージェント自身やその情報の管理コストがネットワーク管理者にとってもなうことは、このシナリオの問題点の 1 つとしてあげられる。なお、アプリケーションが必要とするネットワーク特性は、多くの場合、そのアプリケーションが動作しているノード間のものである。した

がって、上記の規則の 3 つ目にあるように、N-TAP を利用するアプリケーションの動作するノードにエージェントを配置し、関係ノード間のネットワーク特性を収集できる状態とすることで、前述の要件にあった収集可能範囲の網羅性を確保している。

3.3.2 エンドノードの任意参加による構成

特に管理点を設けずに、エンドノードで動作する各エージェントが自律的に core あるいは stub として N-TAP ネットワークに参加してサービスを構成する方法が、2 つ目のシナリオである。なお、前述のシナリオと同様に、アプリケーションは自ノードで動作しているエージェントに問合せを行うものとする。したがって、アプリケーションの関係ノード間でネットワーク特性の収集が可能であり、収集可能範囲の網羅性は同様に確保できる。この場合、公開されている 1 つ以上の core エージェントの情報を何らかの方法で取得しておき、エージェントはそのうちの 1 つの core エージェントに N-TAP ネットワークへの参加申請をすることになる。本シナリオでは管理された core エージェントを用意しないため、1 つ目のシナリオに比べて管理コストが小さくなるのが利点ではあるが、反面任意のノード上のエージェントが任意の役割を担当できるため、大量の不安定なノードが core エージェントとして参加することで、N-TAP ネットワークの性能が低下する懸念がある。現在の実装ではエージェントの役割は手動で指定するが、こうした懸念に対応するためにも、計測ノードの状況に応じた役割の自動選択・変更の戦略を持つことが重要となるであろう。

3.4 利用者にとっての利点

アプリケーションが N-TAP を利用することで、従来不可能もしくは困難であった手順が簡略化されて利用可能となる。本節では、利用者(アプリケーション)が N-TAP を利用することでどのような利点があるかまとめる。

利点の 1 つは、ネットワーク特性の収集範囲が拡大することである。アプリケーション単体、あるいはノード単体では、計測手法によってはネットワーク特性の取得対象の範囲が限定される。たとえば traceroute では探索パケットの送信元ノードが自ノードに限定されるため、自ノードを根とするツリー状のトポロジ情報しか得られない。しかし、N-TAP のように他のエージェントに計測処理の実行を依頼できる環境では、他のエージェントが動作するノードを根とするトポロジ情報も得られ、このようなトポロジ情報を複数組み合わせることで、より広域のトポロジ情報の収集が可能となる。

もう 1 つの利点は、複雑な計測手法をサービスの一機能として利用可能となることである。協調計測手法は他の計測ノードとの通信や状態管理といった、ノード単体で完結する計測処理に比べて複雑な処理が必要となる。このような手法をサービスとして共通化し、利

ユーザーに簡便なインターフェースを提供することで、協調計測手法の有利な性質を利用しやすい環境をつくることできる。また、ネットワーク特性の収集サービスの一元化で、複数のアプリケーション間で収集したネットワーク特性の再利用が可能になる。これにより、ネットワーク特性の収集により生じる資源消費の量が低減することが期待できる。

4. 性能評価

提案する計測基盤の実展開の可能性の検証に必要な基礎的知見を得るため、実際に計測エージェントを動作させてシステムの性能評価実験を行った。本章ではその実験結果を記す。

4.1 実験環境

以下で述べる実験は、PlanetLab¹²⁾ と StarBED¹³⁾ の2つの実験環境を用いて行われた。PlanetLab のノードは世界中に分散しており、ノード間の通信遅延はその間のネットワークの構成や状況に依存してさまざまな大きさになる。また、ノードによっては同時利用者が多く高負荷状態にあるため、ノードの処理速度はいくぶん緩慢かつ不均一で、時間とともに変化しやすい。一方、StarBED のノードは一カ所で管理された実験ノード群であり、ノードの性能および接続リンクの性能は均一であった。実験期間中は、ノードやネットワークの資源を本実験のみで占有していたため、これらの大きな性能の変化はなかったものと考えられる。実験では、どちらの環境においても無作為に選択したノードを用い、各ノードに1個ずつエージェントを動作させた。また、各エージェントの ID は 128 ビット長の乱数値を用い、Chord ネットワークのトポロジはランダムにようにした。

4.2 core ネットワークの応答性

2章で述べたサービスの応答性の高さを確保するために、N-TAP エージェントがネットワーク特性の収集において頻りに利用する機能が、システム規模の拡大に対しても応答性を維持することが望まれる。本節では、アプリケーションからのネットワーク特性収集の要求に対して、core ネットワークで生じる処理ごとに所要時間を調べ、応答性のボトルネックとなる部分を検証する。

実験は、PlanetLab および StarBED で、core エージェントの個数 c が 8, 16, 32, 64, 128 のそれぞれの場合について、core エージェントのみの N-TAP ネットワークを構成して行われた。ここで、ランダムに選択された計測ノードと計測対象ノード (IP アドレスを無作為に選択) の間の RTT 計測処理要求を、PlanetLab では 1,000 回、StarBED では 4,000 回の計測結果が得られるまで発生させ続け、エージェント内での処理の遷移をタイムスタンプとともにログファイルに記録した。なお、さまざまな処理が発生するように、アプリ

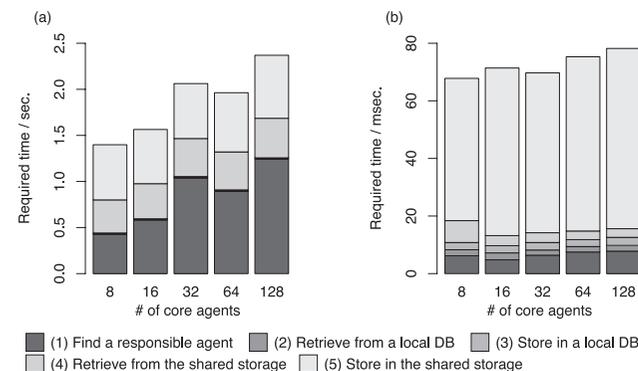


図3 (a) PlanetLab, (b) StarBED での実験におけるエージェントの処理別の所要時間

Fig. 3 Required time for respective agent's tasks in the experiments on (a) PlanetLab and (b) StarBED.

ケーションが許容する情報鮮度には十分大きな値を指定して、要求に対する処理がローカルのネットワーク特性データベースの探索のみで終わらないようにした。収集したログファイルから、発生した主な処理、(1) Chord アルゴリズムに従った N-TAP ネットワーク内の担当エージェントの探索、(2) ローカルのネットワーク特性データベースからの読み出し、(3) ローカルのネットワーク特性データベースへの書き出し、(4) 共有記憶領域からの読み出しのための担当エージェントへのメッセージ送受信、(5) 共有記憶領域への書き込みのための担当エージェントへのメッセージ送受信の5つについて、所要時間の平均を算出した。各処理を図2と対応づけると、(1)は図中の(c.2)のChordネットワーク内のルーティング処理、(4)と(5)は(d)のようなエージェント間の直接の通信処理に対応し、(2)と(3)はエージェント内で完結する処理である。図3に、所要時間の算出結果を示す。

まず、各処理の core エージェント数に対する変化に注目する。(2)から(5)までの処理は core エージェント数の変化に対してその所要時間は大きく変化していないが、(1)については core エージェント数の対数に比例して大きくなる傾向が読み取られる。これは、Chord アルゴリズムの性質から、core エージェント数 c に対して、エージェント間のメッセージ転送回数が $\log c$ 程度になることに由来するものである。特にこの傾向は、ノード間の通信遅延が大きい環境である PlanetLab での実験結果において、顕著に現れている。処理(4)、(5)もエージェント間の通信が生じるが、その処理は1回のメッセージ交換で完結するため、core エージェント数の増加に影響をほとんど受けていないといえる。

次に、各 core エージェント数での処理ごとの所要時間を比較する。PlanetLab での実験結果では、core エージェント数の増加にともなって、処理 (1) が他の処理に比べて所要時間がより長くなっている。前述のとおり、通信遅延の大きい環境では、core ネットワークが大きくなるにつれ、全体の処理の流れの中で処理 (1) の所要時間が全体の所要時間に占める割合は大きくなるといえる。一方で StarBED では、最も処理に時間を費やしたのは処理 (5) である。同様にエージェント間通信をとまなう処理 (1), (4) に比べて、この処理が 1 桁のオーダーで所要時間が長いことには、通信遅延よりもむしろ、メッセージ受信から SQLite データベースへの書き込み準備処理といった、エージェント内での実装依存である処理が影響していると考えられる。

ローカルで完結する処理 (2), (3) 以外の処理では、処理時間のオーダーは PlanetLab と比べて StarBED のものは 1 桁程度小さい。これは、core エージェントが動作するノードすべてが高速なリンクに接続され、処理性能も高いことに由来する。すなわち、core ネットワークの提供する機能の応答性は、core エージェントの動作するノードの性能と core エージェント間の通信遅延に大きく依存することが示されている。

なお、実際には N-TAP エージェントは、core エージェントであれ stub エージェントであれ “local-first and remote-last” ルールに基づいてアプリケーションからの要求を処理する。仮に要求を受け付けたエージェントが自身のネットワーク特性データベース内に指定された鮮度条件を満たすネットワーク特性情報を保持している場合、処理 (2) と同等の時間スケールで情報収集を完了できることになる。共有記憶領域へのアクセスなどネットワークアクセスを含む処理は、図 3 から分かるように処理 (2) に比べて、所要時間が数十から数百倍にもなる。ローカルキャッシュの参照は情報の正確性の面で実計測には劣るが、代償として高いサービス応答性を得られることから、その利用に対して動機付けができると考える。

4.3 ID 割当て方式による負荷分散

前述のとおり、core エージェントには共有記憶領域のデータ項目が保存される。本実験では、エージェント ID には乱数が割り当てられ、データ項目の ID の生成には SHA-1 ハッシュ値を用いているため、core エージェント間で均質に共有データ項目が分散配置されることが期待できる。ここでは、公平性の 1 つとして保持する共有データ項目数が均等であることを用い、core エージェントの共有データ項目の保持数の統計的な分散を基準とした公平性指標 F を以下のように定義する。

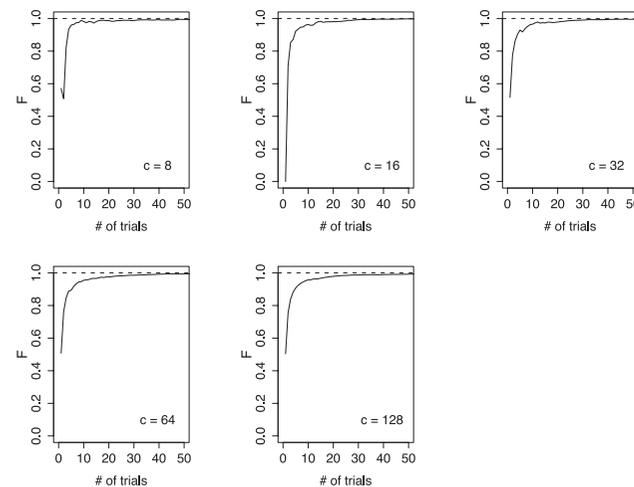


図 4 共有データ項目の分散に関する公平性
Fig. 4 Fairness of the distribution of data entries.

$$F = 1 - \frac{c}{c-1} \sum_{i=1}^c \left(\frac{1}{c} - p_i \right)^2 \quad (1)$$

この式において、 c は core エージェントの個数、 p_i は i 番目 ($i = 1, 2, \dots, c$) の core エージェントが持つ共有データ項目数の全体数に対する割合で、 $0 \leq p_i \leq 1$ および $\sum p_i = 1$ を満たす。 F は正規化されて $[0, 1]$ の値をとり、1 に近いほど公平性が高いと見なすことができる。以下で、本実装での ID 割当て方式が、 F の値を 1 に収束させるものであることを確認する。

さきほどの PlanetLab での実験において、アプリケーションからの要求受付回数にともなう F の値の変化を調べた。図 4 はその結果で、それぞれの c の値について、横軸がアプリケーションの要求受付回数、縦軸が F の値を示している。いずれの c の値の場合も、45 回の要求受け付け以後は F の値が 0.99 以上となっており、1,000 回目のアプリケーションからの試行が終わるまで F の値が 1 に近づく傾向は続くことを確認した。このことから、意図したとおりの共有データ項目の均質な分散配置が行われていることが確認できる。

今回の実験では計測ノードと計測対象ノードが無作為に選択されているためにこのよう

な結果が期待できたが、実運用においては、アプリケーション側にネットワーク特性を収集したい対象に偏りがあることが考えられるので、必ずしもこの傾向には従わず、公平性指標は今回のものより低い値をとると考えられる。N-TAP がインターネット広域にわたり展開され、さまざまなアプリケーションがそれを利用して非常に多くの対象についてネットワーク特性を収集するようになれば、大局的に見てこのような偏りは小さくなり、このエージェント ID および共有データ項目 ID の生成方法は、core エージェント間で共有データ項目を保持する負荷を均等に分散させるには有効であるといえる。

4.4 メッセージ処理数の分布

N-TAP エージェントは、エージェント間で交換するメッセージを受信することで処理が発生するため、メッセージの処理数をエージェント間の処理負荷を比較するための指標として用いることができる。3.1.3 項で述べたように、stub エージェントよりも安定性と処理性能が高い core エージェントで構成される core ネットワーク内に、ネットワーク特性の収集にともなう処理負荷を集中させるため、N-TAP はエージェントの役割を分割している。ここでは、アプリケーションからの要求の処理に際して、各エージェントにどのように処理負荷が分散しているかを観察するため、メッセージ処理数の分布を調べる実験を行った。

実験では StarBED の 128 ノードについて、このうちの core エージェントの個数を 8, 16, 32, 64, 128 のいずれかとし、残りのエージェントを stub とする、計 5 通りの場合を用意した。それぞれの場合で、無作為に選んだ 1 つの core エージェントと 1 つの stub エージェントに対して、無作為に選んだ計測ノード間の RTT の計測要求を発生させる試行を 2,000 回ずつ、計 4,000 回繰り返した。この間、各エージェントでメッセージの送信先をログファイルに記録し、この記録をもとに各エージェントのメッセージの送受信数の合計を算出した。

メッセージの処理数に対する、core/stub 別のエージェント数の割合の累積分布を図 5 に示す。この図では、横軸にメッセージ処理数が対数軸で示されており、縦軸は各役割のエージェント数の割合を累積分布で示している。実線は core エージェント、破線は stub エージェントの結果であり、図中 (e) のグラフは $c = 128$ の場合、すなわちすべてのエージェントの役割が core であったため、stub エージェントについての結果は示されていない。これらの結果から、いずれの場合も stub エージェントのメッセージ処理数は 100 程度であるのに対し、core エージェントについては 1,000 から 100,000 にかけて分布していることが分かる。したがって、stub エージェントは core エージェントに比べて低いメッセージ処理負荷で動作していたことがいえる。また、core エージェントの間ではメッセージ処理数に一定

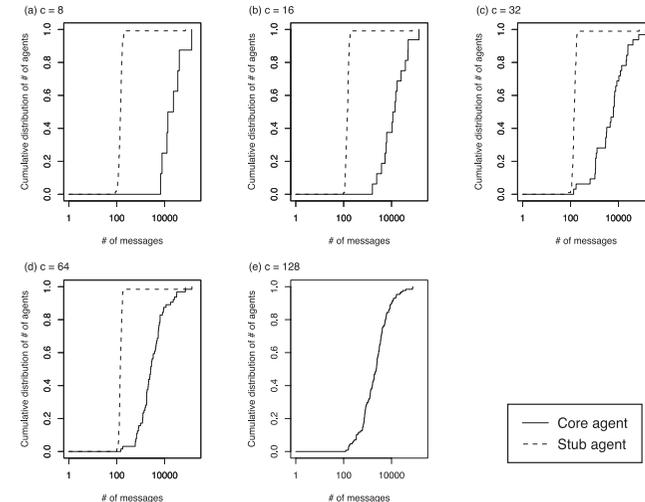


図 5 メッセージの処理数に対する役割別エージェント数の累積分布

Fig. 5 Cumulative distribution of the numbers of agents for the numbers of exchanged messages.

の分布の幅が見られる。これは、収集した RTT 情報の計測ノードと対象ノードの数が限定されていたため、それらの情報から生成される共有データ項目 ID が十分に分散していないことが原因の 1 つである。現実装での ID 割当て方式では、さらに多数の計測対象を持つことで、core エージェント間のメッセージ処理数の偏りは小さくなることが期待できる。なお、いずれの場合も極端にメッセージ処理数の多い core エージェントと stub エージェントが 1 個ずつ存在し、その結果が図中にも現れているが、これらは 2,000 回の計測要求を受け付けたエージェントであるため、多くのメッセージを交換する結果となった。以上により、階層型 P2P ネットワークを用いることで、N-TAP ネットワークにおけるメッセージ処理負荷が core ネットワークに集約することが確認できた。

4.5 core-stub 間通信の応答性への影響

図 2 に示したように、stub エージェントが core ネットワーク上のエージェントの発見機能を利用する際には、接続している core エージェントにその処理を依頼する形態をとるため、core エージェントと stub エージェントの間で 1 回のメッセージの送受信が発生する。したがって、stub エージェントが当該機能を利用する際には、core エージェントの場合に比べて core-stub 間の RTT にほぼ相当する時間分、応答が遅くなるといえる。実際に

PlanetLab および StarBED 上で、32 個の core エージェントと 1 個の stub エージェントからなる N-TAP ネットワークを構成し、無作為に選んだ 1 個の core エージェントと stub エージェントのそれぞれで 4,000 回の core エージェント発見処理を発生させ、処理の完了に必要な時間を記録した。この結果、PlanetLab では平均 1.018 秒、StarBED では 0.003 秒、core エージェントが stub エージェントに比べて早く処理を完了することが確認された。特にノード間通信遅延が大きい PlanetLab では、stub エージェントの応答性の低下の度合いが大きい。PlanetLab ノードの処理負荷が常時高く通信遅延が大きくなる傾向があることを勘案しても、このことは実環境における展開でも処理負荷量の低減とのトレードオフとして再現すると考えられる。

4.6 機能要件を満たすことの確認

N-TAP は、2 章であげられた AOMP のいくつかの機能要件に考慮して設計・実装された。ここでは、これまでに得られた性能評価の結果と N-TAP の想定動作環境の条件をもとに、N-TAP がそれらの要件を満たすよう設計され、また動作していることをまとめて確認する。

まず応答性について、性能評価の結果からサービス全体の応答性は core ネットワークの提供する機能の応答性に強く依存することが分かった。PlanetLab ノードの多くが余剰資源の乏しい状態にあることを考えると、PlanetLab 上での core ネットワークの応答性(図 3(a))は通常よりも悪条件での結果であると考えられる。同じような条件で全エージェント数が数万単位でそのうち core エージェントが 1,000 程度であると想定すると、担当エージェントの探索処理の所要時間は core エージェント数が 128 のときの結果の約 8 倍程度、すなわち 10 秒前後のオーダになるといえる。他の処理に関しては規模拡張に対してもほぼ一定であるといえる。したがって、実際の展開時には探索処理の所要時間が応答時間を大きく占めることになる。通信遅延による応答時間の増加は分散型サービスにおいて避けられないものではあるが、本システムの場合、近接性を考慮した DHT の構成手法などを適用することで一定の改善は可能である。さらに、“local-first and remote-last” ルールにおいてローカルキャッシュを活用できることを勘案すれば、実際はより高いサービス応答性が得られる。

負荷分散公平性については、4.3 節および 4.4 節での実験で実証されたとおり、core エージェント間でほぼ均一に core ネットワークの維持処理にともなう負荷を分散できることが期待される。安定性については、エージェントの役割を core と stub に分割することで、不安定ノードの影響を最小限にとどめるような設計となっている。収集可能範囲については、

いずれのシナリオにおいても N-TAP を利用するアプリケーションが動作するノード上で N-TAP エージェントも動作していることを想定しており、この条件下ではオーバーレイネットワークアプリケーションが必要とするオーバーレイノード間のネットワーク特性の収集は一般的に担保される。正確性については、応答性とのトレードオフとして確保できるように、アプリケーションによる情報鮮度の指定を可能としている。

以上により、N-TAP がそれぞれの要件を充足するための機能を有することが確認された。

5. 展開シナリオの実現性と今後の課題

本章では、前章での性能評価実験の結果をふまえ、3.3 節で示した展開シナリオについて再度考察する。また、N-TAP や AOMP 一般について、実展開に向けた今後の課題についてもまとめる。

まず提示したシナリオのうち、前者の「管理ネットワーク内の基盤サービス」として N-TAP を展開した場合を考える。この場合、利用者が動作させる stub エージェントは各ネットワーク内の core エージェントに接続するため、多くの場合、4.5 節で示した core-stub 間の通信遅延は問題にならないほど大きくないことが想定できる。したがって、利用者は相対的にメッセージの処理負荷が小さい stub エージェントを、core エージェントと比べてほとんど応答性を損なわずに動作させられるため、ネットワーク管理者に一定の基盤管理コストを強い一方で、利用者の利用の障壁は低いことが期待される。ただし 4.6 節で示したように、core ネットワークの応答性の影響は無視できないので、その性能改善が重要となる。

また、もう 1 つのシナリオである「エンドノードの任意参加」で計測基盤を構築する場合は、エージェントは複数のネットワーク管理領域に特に秩序なく分散しているため、core-stub 間の通信遅延は前者のシナリオに比べて大きくなることが予想される。したがって、応答性の向上には複数の core エージェントから RTT の小さいものを選択するなどの配慮が必要となるであろう。この展開方式では基盤の明確な管理点が存在しないため、応答性の改善や安定運用・負荷分散といった N-TAP の設計上の考慮を実現することは、完全な分散システムという性質から 1 つ目のシナリオよりも難しくなる。エージェントの役割の動的変更アルゴリズムや、より耐障害性の高い core ネットワークの構築手法を模索することが、本方式での展開の課題となるであろう。加えて、前者のシナリオと同様に core ネットワークの応答性改善も重要となる。

本論文では、多様なネットワーク計測手法を実装するための汎用的なネットワーク計測基

盤として N-TAP を提案し、計測基盤自体の性質・性能を調査したうえで、それらが基盤上に実装される計測手法に与える影響について考察した。N-TAP と同様にアプリケーションからの利用を意識したネットワーク計測基盤の実装には、 S^3 ⁶⁾、iPlane⁷⁾、pMeasure⁸⁾などがあげられる。これらの既存研究では、提案した計測基盤上に実装された計測手法の性能評価を主としており、計測基盤自体に対する考察は少ない。ここでは、それぞれの計測基盤についてその構造に着目して、本研究との差異を述べる。 S^3 や pMeasure は N-TAP と同様に DHT ベースの情報管理ミドルウェアを用いている。ただし、これらの実装では DHT におけるノード ID の割当てに規則性を持たせたり、DHT の上に抽象機能レイヤを構成したりしている。本論文内では、エージェント ID は無作為に選んだ値を用い、DHT は純粋なハッシュ機能として用いている。したがって、DHT 内の情報の偏在の傾向や取り扱う情報種について差異が生じるため、本論文の知見をこれらの実装に適用するには、それぞれの計測基盤固有の条件を設定したうえで再検討が必要になるであろう。iPlane では、計測結果は中央ノードに集約され、中央ノードが他の計測ノードに計測タスクを配分する形態をとる。一方で N-TAP ネットワークは、実装される計測手法に応じて計測処理手順を決定できるように、他の計測エージェントの発見とエージェント間の共有記憶領域といった、手順の流れに制約を与えない機能のみを提供している。core エージェントが 1 個のみの N-TAP ネットワークを構成したうえで iPlane が定める処理手順を実装すれば、N-TAP は iPlane のような集中管理型の計測基盤として機能することもできる¹⁴⁾ ため、本論文の知見の一部は集中管理型の計測基盤についても適用できる。また、田上らが提案する分散型計測基盤¹⁵⁾ は、N-TAP と同様に Chord を共有記憶領域として用いているが、Chord ネットワーク上でのメッセージ配送機構を用いて Chord ノード間のネットワーク特性の計測と計測結果の記録を同時に行うことで情報収集の効率化を図っている点が特徴的である。これは、計測対象を計測ノードのみに限定したうえで、計測情報の記録という計測基盤の機能と計測手法を直接結び付けたために実現したもので、N-TAP は計測対象を制限せず、かつ計測基盤の機能と計測手法を独立したものとして取り扱っているため、直接このような計測手法を実装できない。計測基盤の構成に強く依存することで計測効率を高める計測手法を、どのようにして汎用的なフレームワークの中で取り扱うかは今後の課題の 1 つである。

その他の関連研究として、階層型 P2P ネットワークの構成手法に注目したい。P2P 構成ノードを安定性の高い super node とそれ以外の ordinary node に分けて 2 層構成の P2P ネットワークを構成する手法は、IP ネットワーク上での音声通話・ビデオ会議を実現する Skype¹⁶⁾ などのアプリケーションでも用いられている。Chord における階層化については、

ノード ID に応じた super node と ordinary node の決定手法¹⁷⁾ が提案されている。この方式と N-TAP における Chord の階層化方式の差異は、stub エージェント (ordinary node) が任意の core エージェント (super node) の配下に位置することができる点である。N-TAP では、より多数の監視ノードを確保するために、所有する資源が乏しく安定性の低いノードでも stub エージェントになれるよう、stub エージェントは Chord ネットワークの維持と DHT におけるデータ保持の役割を分担しなくてもいいように設計した。そのため、stub エージェントは Chord ネットワーク上で直接探索可能である必要がないので、任意の core エージェントの配下となることができる。すなわち、本方式では core ネットワークの構成が変更されても、stub エージェントは配下となるべき core エージェントを切り替える必要はなく、より stub エージェントの処理負荷を軽減し、一方で core エージェントにデータ保持などの負荷を分担させる方式となっている。また、P2P ネットワークの形態に応じた super node の選択方法についても研究されている¹⁸⁾。階層化 P2P ネットワークにおいてサービスの安定性は core ネットワークの安定性に大きく依存するものであるため、N-TAP においても動的な core エージェント選出は将来必要になるものと認識している。

より現実的な AOMP を実現するための今後の課題としては、そのほかの計測ネットワークの構成手法の検討、収集した情報へのアクセス制御、現状のリクエスト・レスポンス型以外の購読出版型のようなアプリケーションとエージェント間の情報授受形式の適用などがあげられる。

6. む す び

本論文では、Chord ベースの P2P ネットワークと計測ノードの役割分割を組み合わせた計測ネットワークを持つ分散型計測基盤について、その設計・実装および展開シナリオを示した。これらは、計測基盤の応答性や安定性・負荷分散公平性・収集可能範囲・正確性といった要件を満たすよう考慮されたものである。また実装の性能評価実験を通じて、core ネットワークの応答性特性、core エージェント間の負荷分散、core ネットワークへのメッセージ処理負荷集中、core-stub 間通信の影響について検証し、提案方式で実現される AOMP の実展開に向けた基礎的知見を示した。

AOMP の実現はこれまで霞んでいたインターネットの内部状態を照らし出し、新興の自律的なアプリケーションにとって意義深いものである。加えて、ネットワーク特性が容易に利用できるようになることで、新しい形態のインターネットアプリケーションが創出されることも、おおいに期待するところである。

謝辞 本研究を行うにあたり、分散計測手法やその実環境への展開について助言をいただいたベルギー Université catholique de Louvain の Benoit Donnet 氏とフランス Université Pierre et Marie Curie の Timur Friedman 准教授に、深く感謝の意を表したい。本研究の一部は、科学技術振興機構戦略的国際科学技術協力推進事業の一環として行った。

参 考 文 献

- 1) Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S.: A Scalable Content-Addressable Network, *Proc. ACM SIGCOMM 2001* (2001).
- 2) Ratnasamy, S., Handley, M., Karp, R. and Shenker, S.: Topologically-Aware Overlay Construction and Server Selection, *Proc. IEEE INFOCOM 2002* (2002).
- 3) Rowstron, A. and Druschel, P.: Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems (2001).
- 4) Castro, M., Druschel, P., Hu, Y.C. and Rowstron, A.: Proximity Neighbor Selection in Tree-Based Structured Peer-to-Peer Overlays, Technical Report MSR-TR-2002-82 (2003).
- 5) To, K. and Lee, J.Y.: Parallel Overlays for High Data-Rate Multicast Data Transfer, *Computer Networks*, Vol.51, No.1, pp.31-42 (2007).
- 6) Yalagandula, P., Sharma, P., Banerjee, S., Basu, S. and Lee, S.-J.: S^3 : A Scalable Sensing Service for Monitoring Large Networked Systems, *Proc. INM '06* (2006).
- 7) Madhyastha, H.V., Isdal, T., Piatek, M., Dixon, C., Anderson, T., Krishnamurthy, A. and Venkataramani, A.: iPlane: An Information Plane for Distributed Services, *Proc. OSDI '06* (2006).
- 8) Liu, W., Boutaba, R. and Hong, J.W.-K.: pMeasure: A Tool for Measuring the Internet, *Proc. E2EMON '04* (2004).
- 9) Donnet, B., Raoult, P., Friedman, T. and Crovella, M.: Efficient Algorithms for Large-Scale Topology Discovery, *Proc. ACM SIGMETRICS 2005* (2005).
- 10) Dabek, F., Cox, R., Kaahoe, F. and Morris, R.: Vivaldi: A Decentralized Network Coordinate System, *Proc. ACM SIGCOMM 2004* (2004).
- 11) Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M.F., Dabek, F. and Balakrishnan, H.: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, *IEEE Trans. Networking (TON)*, Vol.11, No.1, pp.17-32 (2003).
- 12) Princeton University: PlanetLab. <http://www.planet-lab.org/>
- 13) The StarBED team: StarBED Project. <http://www.starbed.org/>

- 14) Masui, K. and Kadobayashi, Y.: A Role-Based Peer-to-Peer Approach to Application-Oriented Measurement Platforms, *Proc. AINTEC 2007* (2007).
- 15) 田上敦士, 長谷川輝之, 阿野茂浩, 長谷川亨: コピキタスネットワークにおける大規模計測システムアーキテクチャ, 電子情報通信学会論文誌, Vol.J89-B, No.10, pp.1885-1893 (2006).
- 16) Skype Limited: Skype. <http://www.skype.com/>
- 17) Singh, K. and Schulzrinne, H.: Peer-to-peer Internet Telephony using SIP, Technical Report CUCS-044-04, Columbia University (2004).
- 18) Lo, V., Zhou, D., Liu, Y., GauthierDickey, C. and Li, J.: Scalable Supernode Selection in Peer-to-Peer Overlay Networks, *Proc. HOT-P2P 2005* (2005).

(平成 20 年 5 月 19 日受付)

(平成 20 年 11 月 5 日採録)



益井 賢次 (正会員)

2006 年奈良先端科学技術大学院大学情報科学研究科情報処理学専攻より修士 (工学) 取得。2008 年奈良先端科学技術大学院大学情報科学研究科博士後期課程中退。同年より東京工業大学学術国際情報センター産学官連携研究員。広域ネットワーク計測基盤, ネットワーク運用技術の研究に従事。



門林 雄基 (正会員)

1996 年大阪大学大学院基礎工学研究科物理系専攻博士後期課程中退。同年大阪大学大型計算機センター助手。1997 年大阪大学大学院基礎工学研究科物理系専攻情報工学分野より博士 (工学) 取得。1999 年大阪大学大型計算機センター講師。2000 年奈良先端科学技術大学院大学情報科学研究科助教授。WIDE プロジェクトボードメンバ。2006 年より情報通信研究機構トレーサブルネットワークグループプロジェクトリーダーを兼任。ネットワークセキュリティ, オーバレイネットワークの研究に従事。著書に岩波講座インターネット第 2 巻「ネットワークの相互接続」。