

動的なネットワークにおける 効率的なサービス発見機能と経路構築機能

高杉 耕一[†] 中村 元紀[‡]

日本電信電話(株) NTT 未来ねっと研究所

1. はじめに

我々は動的なネットワークにおいて、ノード間でアプリケーションを変更した場合に、サービスを中断なく連続するシームレスサービスプラットフォーム[1]を提案し、必要な技術について検討してきた。このシームレスサービスプラットフォームはアプリケーションレベルのオーバーレイネットワークで構成される。動的なネットワークでは常にネットワークの状態が変化し、分断、融合する。そのため、サービス中にネットワークが分断された場合においても、動的に代替サービス提供主体を選択し、サービスを継続することで可用性を高める必要がある。これを実現するためにネットワーク上のサービス提供主体は同じサービスの複製を他の複数のノードに分散し保持する。またサービス中にサービスを楽しむ端末を変更したり、通信媒体、ネットワーク上の位置が変化したりすることが考えられる。このような場合には、サービス中であっても、常に適切なサービス提供主体を発見し、それにいたる経路を評価することで、冗長な経路をなくし、ネットワーク的に近傍のサービス提供主体に切り替える必要がある。我々は、このようなプラットフォームについて、小規模なネットワークを使って実証実験を行ってきたが、大規模なネットワークではサービスや経路の広告が膨大になりそのまま適用することはできない。そこで、本稿ではスケーラブルなサービス発見方法、経路構築方法について提案する。

2. 要求される機能

複数の同一サービス提供主体を発見し、それぞれのサービス提供主体に対し経路を評価することで、どのサービス提供主体にどのような経路で接続するかを決定する。要求される機能は以下の2つである。

- (1) サービス提供主体の発見機能
- (2) サービス提供主体の評価機能

これらの機能が大規模なネットワーク上でスケーラブルに動作する必要がある。

3. 提案方式

3.1 シームレスサービスプラットフォーム

シームレスサービスプラットフォーム[1]の構成を図1に示す。各ノードにはシームレスプロキシ(S-proxy)が配置され、従来のトランスポート層でつくられるネットワークの上に仮想(オーバーレイ)ネットワークを構成している。アプリケーション間で行われているデータの送受信は複数のS-proxyによって中継される。各S-proxyはS-proxy IDによって一意に識別される。

3.1.1 動的なサービス情報の蓄積

S-proxy は中継するデータを解析したり、アプリケーションに問い合わせることでアプリケーションの状態やプロトコルの通信状態(動的なサービス情報)を把握する。

Scalable Service Discovery and Routing Method for Dynamic Network

Koichi Takasugi[†], Motonori Nakamura[‡], NTT Network Innovation Laboratories, NTT Corporation

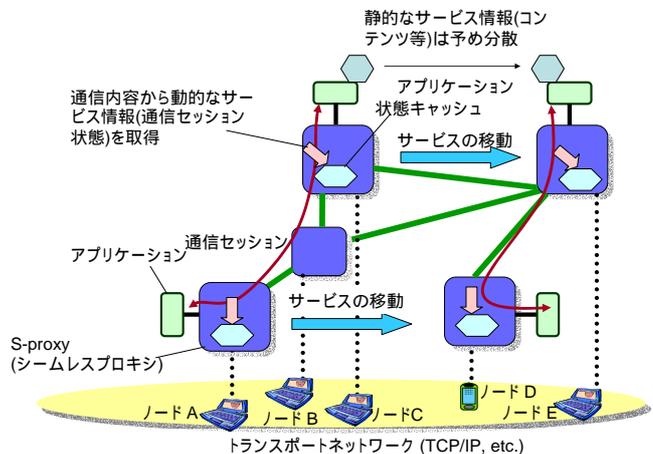


図1 シームレスサービスプラットフォーム

3.1.2 静的なサービス情報の分散

動的なネットワーク環境では、サービスを提供している端末も、ネットワークから切断される可能性がある。また、ネットワークの変化に伴い、サービス提供端末までの経路が長くなり十分な帯域が確保できなくなる可能性もある。

そこで、予め静的なサービス情報を複製として各端末に分散配置する。これは、動画等のコンテンツ配信サービスであれば、コンテンツを複製することに相当する。もしサービス中にサービスを提供している端末を使い続けることが不適切になった場合、サービスの複製を持っている別の端末に変更することで、サービスを継続することができる。静的なサービス情報は、S-proxy が中継している通信データから自律的に蓄積するか、複製元のS-proxyからの指示により作成することができる。

3.1.3 サービス移動

動的なサービス情報を他のS-proxyに転送し、経路を転送先のS-proxyに変更する。このように、アプリケーションを他の端末上のアプリケーションに変更してもサービスを連続的に継続できる。例えば図1ではノードDのS-proxyにノードCのS-proxyの動的サービス情報を転送することにより、ノードAとノードCのアプリケーション間で行われていた通信をノードDとノードCに切り替える。さらに、ノードDの動的サービス情報をノードEに転送することで、ノードDとノードEの通信に切り替える。この際、ユーザから見たサービス(クライアント、サーバサービス)は連続している。

3.2 スケーラブルなサービス発見方法、経路構築方法

3.2.1 サービス提供主体の評価機能

サービス提供主体を選択し、サービス提供主体までの経路を決定するため、それぞれのサービス提供主体に至るまでの経路のコスト値を加算することにより評価する。ここでの経路はS-proxyのつながりであり、コスト値は

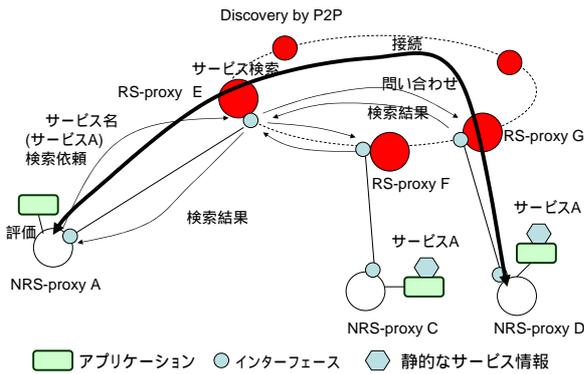


図2 セッション開始

インターフェースや S-proxy 間のリンクに付与される。コスト値は S-proxy の負荷やネットワークへの接続の安定性等を反映して決定する。

しかしながら、フラットな仮想ネットワーク構成では、各コスト値をネットワーク全体に伝播させ、すべての経路の組み合わせに対し、コストを計算する必要がある。したがって、大規模なネットワークにおいても、効率良くコストを計算するためには経路の集約が必須となる。

S-proxy を移動が少なく、端末リソースも多く、比較的安定して動作する Reliable S-proxy (RS-proxy) と移動が多かったり、端末リソースも少なかったりと比較的不安定な Non reliable S-proxy (NRS-proxy) の二種類に分け、RS-proxy に NR-S-proxy の経路を集約させる。NRS-proxy は RS-proxy へサービス名(例 URL)、S-proxy ID とインターフェース情報(例 IP アドレス、ポート番号)の対を送信し、RS-proxy に所属する。NRS-proxy と RS-proxy 間の通信路は定期的に接続確認し、一定時間接続確認がなければ、RS-proxy はその NRS-proxy のサービス名、インターフェース情報を破棄する。所属していた RS-proxy と通信できなくなった NRS-proxy は新たに別の RS-proxy を探す。RS-proxy は NRS-proxy から通知されたサービス名とインターフェース情報を保持する。

3.2.2 接続先サービス発見機能

RS-proxy 間は P2P 技術によって、サービス名からそのサービスに接続できる NRS-proxy が所属している RS-proxy の S-proxy ID とインターフェース情報を得る。

また、サービス広告の管理検索方法としては RS-proxy の数が多い場合はノード数 n に対し $O(\log n)$ の通信複雑度で検索できる分散ハッシュテーブル[2]を使い、RS-proxy の数が少ない場合には粗結合分散ハッシュテーブルで簡略化することができる。

3.2.3 サービス開始

図 2 を例にサービス開始時の手順を示す。各インターフェースの通過コストは 5 とし、NRS-proxy と RS-proxy 間のリンクコストは 0 とする。

(手順 1) NRS-proxy A は所属している RS-proxy E にサービス名をキーにサービス検索を依頼する ()。

(手順 2) RS-proxy E は P2P 技術を利用して、サービスを実現する NRS-proxy が所属している RS-proxy のインターフェース情報を得る ()。

(手順 3) RS-proxy E は手順 2 で得られた RS-proxy に接続し接続先の NRS-proxy のインターフェース情報とコストを得る。RS-proxy E, G 間のコストは 10, RS-proxy E, F 間のコストは 20 であるとする。(RS-proxy 間のコスト

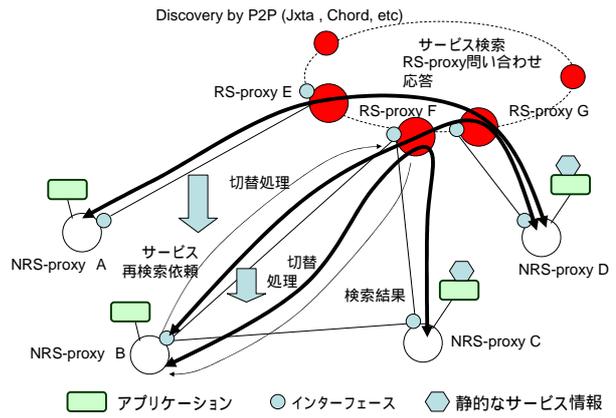


図3 サービス移動

は遅延などを考慮し、RS-proxy E が動的に決定) (~)

(手順 4) 手順 3 で得られた結果(NRS-proxy C, D のインターフェース情報とコスト)を NRS-proxy A に送る ()。

(手順 5) NRS-proxy A は手順 4 で得られた結果(NRS-proxy D へのコストは $5+5+5+10+5+5+5=40$, NRS-proxy C へのコストは $5+5+5+20+5+5+5=50$ を評価し ()、NRS-proxy D を選択し、NRS-proxy D に RS-proxy E, G 経由で接続し、セッションを開始する ()。

3.2.4 サービス移動

図 3 にサービス移動の動作例を示す。NRS-proxy A から NRS-proxy B へ接続元のアプリケーションを変更する ()。その後 NRS-proxy B はサービス開始時と同様にサービス提供主体の再検索を行う (~)。このとき RS-proxy F, G 間のコストが 15 だとすると、NRS-proxy D へのコストは $5+5+5+15+5+5+5=45$ 、NRS-proxy C へのコストは $5+5+5+5=20$ となるため、接続先を NRS-proxy C に切り替える ()。

4. 考察

RS-proxy に関連がある NRS-proxy をグループ化するように配置すると、ある RS-proxy に所属している NRS-proxy のネットワーク構成の変化は RS-proxy の保持しているインターフェース情報が変わるだけでネットワーク全体に波及しない。また、同じ RS-proxy に所属している NRS-proxy 間に閉じたサービスはネットワーク全体に広告する必要がなく、RS-proxy 単位でアクセス制御することが可能である。4 万台の端末で 20 万のサービスを展開する場合でも、各 NRS-proxy に 50 のサービス、各 RS-proxy に 20 の NRS-proxy、RS-proxy が 2000 とすると、1000 個のサービスを持っている 2000 台の端末の P2P ネットワークに還元できる。さらに、NRS-proxy で構成されるネットワークが動的に変化しても、その影響は RS-proxy 内に局所化されネットワーク全体への波及は最小限となる。そのため、大規模なネットワークにも十分適用可能である。

5. まとめ

大規模なネットワーク上でシームレスサービスプラットフォームを構築する際、動的に接続サービスを発見し、発見したサービスを評価する方法を提案した。

参考文献

- [1] 高杉, 中村, "P2P 技術を利用したシームレスサービスプラットフォームの実現," FIT2003, M-099, 2003.
- [2] I. Stoica, R. Morris, etc. "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," SIGCOMM 2001, San Diego, 2001.