

論文内容の要旨

論文題目 A Research on Multi-path Communication Protocols

(複数経路通信プロトコルに関する研究)

氏名 ロットウィブンチャイ グンティダー

現在通信ネットワークに利用されているアプリケーションは3つの種類に分けられることができる。1つ目は信頼できるデータ転送（例えば、FTP、WWW。）である。これらのアプリケーションはデータ損失を許されないため、フロー制御や輻輳制御などのサービスが必要である。このようなサービスをアプリケーションに提供するために、TCP（Transmission Control Protocol：伝送制御プロトコル）というトランスポートプロトコルは広範囲に使われている。2つ目はマルチメディアストリーミング（例えば、ビデオ・オン・デマンド、ビデオ会議。）である。これらのアプリケーションはデータ損失を許されるが、厳しいジッター、遅延、スループットを要求する。このようなサービスをアプリケーションに提供するためにUDP（User Datagram Protocol：ユーザー・データグラム・プロトコル）というトランスポートプロトコルは広範囲に使われている。フロー制御と輻輳制御の仕組みはアプリケーション層に任せる。アプリケーション層では、ネットワーク状態を知るためにRTP（Real-time Transport Protocol：リアルタイム・トランスポート・

プロトコル)が使われている。RTPからのネットワークの状態(損失率や遅延など)によってTFRC(TCP-Friendly Rate Control : TCPフレンドリー・レート・コントロール)というプロトコルがストリーミングのレートを調節する。3つ目はトランザクション(例えば、RPC(Remote Procedure Call : 遠隔手続き呼び出し)である。これらのアプリケーションは1つ目の種類と同様にデータ損失を許されない。但し、遅延に大変敏感である。このようなアプリケーションをサポートするためにT/TCP(Transactional TCP : トランザクショナルTCP)は提案される。

上記に述べたようにそれぞれのアプリケーションは要求が異なる。従って、異なったプロトコルは設計され、それぞれのアプリケーションのために使用される。それにもかかわらず、上記に述べた標準プロトコルは複数経路同時通信に対応できるように設計されていない。複数経路同時通信を行ったら、それらの標準プロトコルの性能は大幅に低下してしまう。

そこで、この論文は複数経路通信プロトコルに注目する。一般的には、複数経路通信は単一経路通信より下記の利点を得られる。第一は、複数経路をバックアップとして使い、より良い経路冗長性と障害耐久性が得られる。第二は、複数経路の帯域幅の使用によってより高いスループットが達成できる。第三は、より少ないフローの断続率と高い安定したレートが得られる。特にモバイル無線ネットワークには断続率がもっと少なく、レートがもっと高く安定になる。

この論文では、2つの複数経路通信プロトコルを提案する。1つ目の提案のプロトコルはR-M/TCP(Rate-based Multi-path Transmission Control Protocol : レートに基づいた複数経路通信プロトコル)と呼ばれ、インターネット上に複数経路データ転送を行うために設計された。R-M/TCPのプロトコルの設計は2つの条件を満たす必要がある。1つ目の条件は標準のプロトコルであるTCPとの互換性を持つことである。2つ目の条件はネットワークインフラを変更せずに複数経路ストリーミングを行うことである。1つ目の条件を満たすためには、コネクション確立の時に通信相手が複数経路通信に対応できるかどうかを通信相手と確認する。対応できれば、複数経路通信を行う。対応できなければ、標準のTCPで単一経路通信を行う。2つ目の条件を満たすためには、インターネット・サービス・プロバイダー(ISP)内に変更しないが、エンド・ホスト側に変更が必要である。まず、エンド・ホストには複数ゲートウェイに接続する。それぞれのゲートウェイは違うISPに接続する。これはインターネット上で複数経路を確立するためである。次に、複数経路通信をサポートするためにエンド・ホストのトランスポートレーヤを変更する必要がある。つまり、エンド・ホストにソフトをインストールすることである。R-M/TCPの重要なコンポーネ

ントは損失回避の輻輳制御、帯域幅の推定、キューの長さの推定、レート制御である。既存の TCP のように損失回復の輻輳制御を使用せずに、損失回避の輻輳制御を使用し、ロングファットパイプの環境でプロトコルの性能を向上するためである。

有名な ns2 というネットワークシミュレータを使用し、シミュレーションを行った。シミュレーション結果によると、R-M/TCP は TCP Reno より高いスループットを達成できた。（早い経路を使った TCP Reno に比べると、1.55 倍のスループットが得られた。遅い経路を使った TCP Reno に比べると、2.33 倍のスループットが得られた。）しかし、R-M/TCP の性能は悪い経路に引っ張られた。原因は各経路の特徴を考慮せずに R-M/TCP が簡単にラウンドロビン手法でデータパケットを複数経路に分配するからである。そこで、R-M/TCP の性能を向上するためにパケット・スケジューリング・アルゴリズムを提案する。シミュレーション結果によると、パケット・スケジューリング・アルゴリズムを用いた R-M/TCP はラウンドロビン手法を用いた R-M/TCP より高いスループットを達成できた。

しかし、シミュレーション結果によると、R-M/TCP のスループットはまだ全経路の帯域幅を使い切れていなかった。原因は受信ウィンドウのブロックである。つまり、良い経路の輻輳ウィンドウはデータパケットの送信を許したが、実際に良い経路に何も送っていない時が多かった。今後の課題はこのようなブロック問題を解決する必要がある。

2 つ目の提案のプロトコルは AMTP (Ad hoc Multi-path sTreaming Protocol : アドホック複数経路ストリーミングプロトコル) と呼ばれ、アドホックネットワーク上に複数経路マルチメディアストリーミングを行うために設計された。アドホックなネットワークでは、マルチメディア・ストリーミング・アプリケーションのエンドツーエンド QoS を向上するために、複数経路通信が有効な方法である。インターネットの既存のストリーミングプロトコルは TFRC である。TFRC はパケット損失を使用することによってネットワークのコンジェスションを検出する。次に、検出した結果に応じてストリーミングレートを増やす、または、減らす。TFRC はインターネット上で良い性能が得られるが、アドホックネットワーク上で性能が低下してしまう。それはパケット損失の理由がネットワークコンジェスションだけではないからである。アドホックネットワークでは、パケット損失の理由が 4 つある。1 つ目はコンジェスションである。2 つ目は無線チャンネルエラーである。3 つ目は経路変更である。4 つ目は経路切断である。パケット損失の理由を識別し、その理由に応じて最も適切なエラー・コントロール及びリソース・アロケーショ

ンを行うことは、マルチメディアストリーミングアプリケーションに対してとても重要である。例えば、パケット損失率が増加しているときに、損失の理由はチャンネルエラーだったら、エラー・コントロール・レベルを増やすべきであるが、損失の理由はコンジェスションだとすると、エラー・コントロール・ラベルを増やしても効果がないので、そのときにストリーミングレートを減らすべきである。

マルチメディアストリーミングアプリケーションがエラー・コントロール及びリソース・アロケーションを正しく行うようにするため、AMTP はネットワーク状態を検出することによってパケット損失の原因を区別することができる。そして、検出したネットワーク状態をアプリケーションに報告する。つまり、ネットワーク状態は経路変更だったら、その古い経路が切断される前に新しい経路を見つけ、見つかった経路の状態（帯域幅、エラー・レート）を上位レーヤであるマルチメディアアプリケーションに連絡する。新しい経路の状態を連絡してもらうので、その経路の状態に応じてマルチメディアアプリケーションは最も適切なエラー・コントロール・ラベルとストリーミングレートを決められる。シミュレーション結果によると、複数経路ストリーミング手法は単一経路ストリーミング手法より高いスループットと高い安定したレートが得られた。

（AMTP を用いた複数経路ストリーミング手法は TFRC を用いた単一経路ストリーミングより 1.95 倍のスループットを得られた。）

しかし、ある場合には、複数経路ストリーミングが単一経路ストリーミングより少ないスループットを達成した。原因は、AMTP が要求を満たす複数経路を見つからないときがあることである。見つかったとしても、その経路は近い場所に位置し、多くの相互干渉を持つときがある。それに関しては、今後の課題として、有効な複数経路ルーティングプロトコルを提案する必要がある。そのルーティングプロトコルはノード・ディスジョイントな複数経路を探すだけでなく、最も少ない相互干渉の複数経路を探す必要がある。