

一対多通信におけるネットワーク障害物対応方法選択プロトコルの設計

日山雅之† 河野英太郎‡ 前田香織‡

(†広島市立大学情報科学部, ‡広島市立大学大学院情報科学研究科)

1 はじめに

近年、ネットワークの急速な普及により、動画の配信サービスが増えている。しかし、多くの家庭、教育機関、企業で用いられるインターネット接続にはNAT/NAPTやファイアウォールなど、ネットワーク障害物[1]と呼ばれる機器が介在する 경우가多い。今後、さらにデータ配布等の多様なサービスの提供が想定され、ネットワーク障害物の配下ではこれらの配信サーバとの通信が阻害される場合が増えると想定される。

本稿では、今後、増大すると予想されるストリーム配信などの一対多通信において、通信経路上にネットワーク障害物が介在する場合でも、できるだけ多くのユーザがサービスを楽しむためにこれらの障害物に対応する方法を選択するプロトコルを提案する。

2 ネットワーク障害物

ネットワーク障害物と呼ばれる NAT/NAPT はネットワーク内のノードにプライベートアドレスを割り当てるため、限りあるグローバルアドレスを効率よく使うことが可能になる。しかしNAT内部のノード同士の通信においては、相手のアドレスを一意に特定できず、直接通信ができない。それを可能にする対応方法としてUDP Hole Punching[2]やRelaying[3]などのNAT越え(NAT Traversal[2])技術がある。また、自ネットワークで使用されている NAT/NAPT の種類を調べるプロトコルとしてSTUN[4]がある。

3 提案プロトコルの概要

提案プロトコルのネットワークモデルを図1に示す。提案プロトコルで対象とするネットワークモデルは Sender, Receiver, Broker, STUN Server の4つのノードから構成される。Sender は、1つ以上の Receiver へデータを送信し、複数存在しない。また、Sender と Receiver は STUN Server と通信することにより自ネットワークで使用されている NAT/NAPT の種類の情報を得る。Broker は Sender と Receiver のネットワーク環境に応じて、NAT の対応方法を決定し、Relaying の際の中継サーバとしての役割も果たす。

NAT 対応技術を選択するとき、Relaying は Sender, Receiver のネットワーク環境によらず対応できるが、Broker に負荷が集

中するため、なるべく選択すべきでない。UDP Hole Punching はどちらか一方が Symmetric NAT 内部にあると対応できない。逆にこれらの技術を使わなくても、一方が NAT の外部である場合、NAT 内部ノードから NAT 外部ノードへ通信を開始することで通信が確立できる性質を利用できる。

これらのことから、ネットワーク障害物に対応する方法を選択するプロトコルをまとめたものを図2に示す。

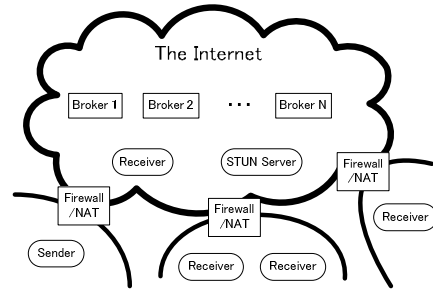


図1:提案プロトコルのネットワークモデル

4 おわりに

現在、提案プロトコルを用いるプロトタイプシステムを Windows と Linux を用いて開発している。今後は、動作確認や選択方法の合理性や一対多通信を用いる様々なデータ配信サービスへの応用について評価する。また、Broker を利用する際の認証方法についても検討する。

参考文献

- [1] B.Carpenter, "Internet Transparency", RFC2775, IETF, (2000).
- [2] B. Ford, P. Srisuresh, and D. Kegel, "Peer-to-peer communications across network address translations", In the 2005 USENIX Annual Technical Conference, USENIX, pp. 179-191 (2005).
- [3] J. Rosenberg, R. Mahy, and C. Huitema, "Traversal using relay NAT (TURN)", Internet-Draft, IETF (2005).
- [4] J. Rosenberg, J. Weinberger, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC3489, IETF, (2003)

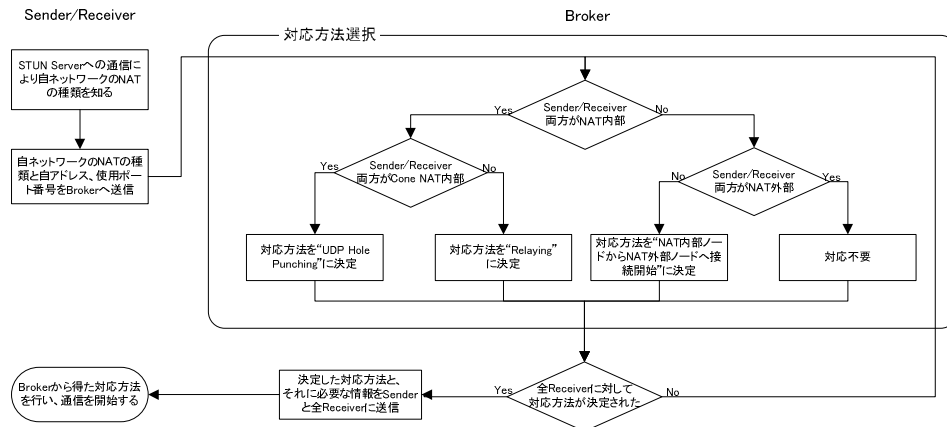


図2:ネットワーク障害物対応プロトコル