n チャンネル通信のための経路制御

木下研究室　　小川　真人 (200703006)

1　はじめに

従来の公開鍵暗号方式では公開鍵の正当性を証明するために認証局のような信頼できる第三者機関が必要であった (図 1)。それに対して n チャネルメッセージ伝送方式では事前の鍵が不要なため第三者機関も必要ない。そこで本研究では n チャネルメッセージ伝送方式に着目した。今回はｎチャンネル通信の要となる、ネットワーク層でのプログラムの実装を目的とした。

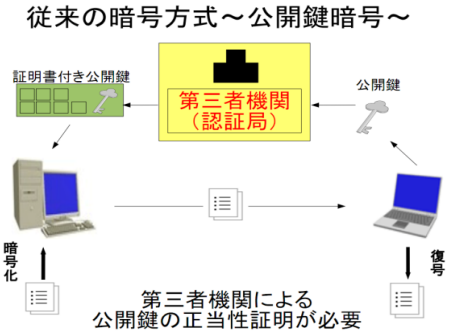

図 1: 公開鍵暗号方式

2　n チャネルメッセージ伝送方式

n チャネルメッセージ伝送方式は文書を n 本の通信路を使用してファイルを分散させて通信を行う方式である。分散されたファイルが、すべて違う経路を通り相手に届くことが理想である。しかし、現在のネットワークシステムでは、送信先しか指定できないので、ｎ本の経路を用意するのは不可能である。ｎ本の経路を用意するために、本研究では、経路制御（ソースルーティング）に着目した。

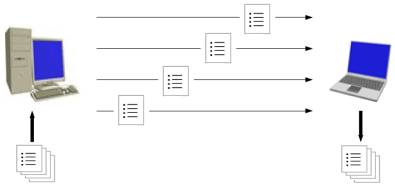

図 2: n チャネルメッセージ伝送方式

3　ルーティング

インターネットのように大規模なネットワークでは、あて先ネットワークまでデータを送信したい場合、複数の経路があることが大半である。そこで、パケットの中継地点となるルータが、適切だと思われる経路にパケットを送り出し、あて先ネットワークまでデータを届けるのである。このように、通信経路から最適な経路を選択・制御する仕組みのことをルーティングと呼ぶ。

4　経路制御（ソースルーティング）

先に説明したように、TCP/IP のルーティング経路は途中のルータが自動的に指定するようになっているが、ネットワーク層、TCP/IP ネットワークのルーティングテーブルに、パケットの通過経路を送信者が指定するのが、ソースルーティングである。途中のすべての経路を指定する「ストリクトソースルーティング」(strict source routing) と、いくつかの経路を通過することを指定して、それ以外は途中のルータに任せる「ルーズソースルーティング」(loose source routing) の 2 種類がある。図 3 のように、最短経路ではなく迂回させた経路を辿らせることによって、ｎ本の経路を確保する。赤の線が最適化された経路であり、緑や青は迂回させたルートとなっている。

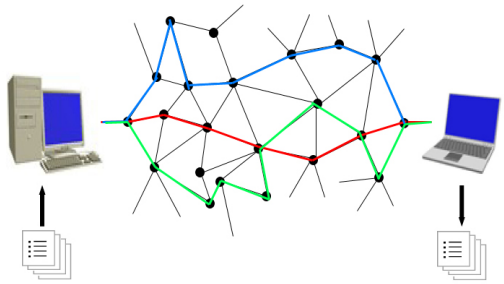

図 3: 経路制御のイメージ

図 4 のような情報をパケットの先頭につけることで、経路制御は実現できる。

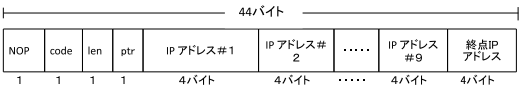

図 4: 始点経路制御指定の形式

本実験は Fedora14 にて行った。実際に Fedora14 の入ったパソコンを複数台用意し、経路の途中にあるパソコンで、tcpdump を用いてパケットの流れを確認するといった方法にて実験した。tcpdump とは、ネットワーク上に流れるパケットをモニタリングするもので、オプションとして条件式を指定すれば、取得したい情報にフィルタリングしてパケットを取得することが可能である。本実験で作っているプログラムは UDP で通信をしている。TCP を選択しなかったのは、パケットシーケンスチェックによる欠損パケット再送などのエラー訂正機能などを持っているため、複雑になる為である。

5　おわりに

ｎチャンネル通信を実現するために、パケットの先頭にルーティングテーブルを書き込むプログラムを作成した。また、プログラムの作成にあたり、さまざまな準備を行った。