

エージェントを用いた仮想回線の QoS 制御の一考察

小林 真也[†] 山田 宏之[†]
藤沢 実^{†,☆} 柏木 紘一[†]

コンピュータネットワークが広がり、利用者が増加するとともに様々な要求を満たす柔軟な通信方式が望まれている。しかし現状のパケット通信方式では、ネットワークの状態や負荷の変化やユーザの求める様々な QoS 等に柔軟に対応することは難しい。本論文では、仮想回線方式において仮想回線をエージェントで管理し、経路を対象とした動的ルーティングを行う方式を提案する。本方式を用いることにより、仮想回線方式における信頼性の高い通信と、スループット重視・遅延重視といったユーザ毎に異なる QoS を提供することが可能となり、そのために必要なエージェントの機能や経路変更時のエージェントの動作について述べる。

Study on QoS control of Virtual Circuit by Multiple Agents

SHIN-YA KOBAYASHI,[†] HIROYUKI YAMADA,[†] MINORU FUJISAWA^{†,☆}
and KOICHI KASHIWAGI[†]

Today, Internet users are on the increase due to the extent of the network and the development of the computer. But, it is difficult to cope with user's various requirements for QoS by the present packet transport service. In this paper, we propose a new virtual circuit protocol. In this new protocol, a virtual circuit is managed by multiple agents. The agents modify the connection route of the virtual circuit due to variation of traffic.

1 はじめに

高度情報化社会の発展に伴い、通信、産業、学問、研究等の分野だけでなく、個人の生活においてインターネットの需要が高まっている。利用者の増加とともに、通信の信頼性や転送速度など、様々な要求を実現する、より柔軟で高品質な通信方式が望まれている。

現在、高品質な通信を実現するパケット通信方式として仮想回線方式 [1] がある。しかし、この方式は経路が固定的であるために、ネットワークの負荷が増加するとデータ伝送の遅延やスループットの低下が生じる。そこで本研究では、仮想回線方式を元に、エージェントを用いて経路を対象とした動的ルーティングを行い、利用者の求める QoS に応じる通信方式を提案する。

2 パケット通信方式の特徴

現在、パケット通信方式には主に、仮想回線方式とデータグラム方式がある。

仮想回線方式では、送信元ホストと宛先ホスト間にコネクションを張り、データ伝送に固定的な経路を用いる。そして、データの順序・輻輳・流量・再送制御を行うことにより信頼性の高い通信を保持している。しかし通信経路が固定的であるため、その経路上のノードの負荷や回線のトラフィックが増加すると、データ伝送の遅延やスループットの低下が生じる。また経路上のノードや回線の障害等により通信が行われなくなった場合、再度両ホスト間のコネクションを確立しなければならぬ。

一方、データグラム方式では、経路は固定されることなくパケット毎に各ノードで独立にルーティングされる。従って、仮想回線方式に比べてノード上で多くの処理をしなければならない。また、各パケットは送信元アドレスと宛先アドレスを完全な形で持たなければならない。データの順序・再送制御等はトランスポート層やその上位層において行う。一方、障害に対してはその発見後、パケットは別の経路でルーティングされるので、例えばノードの障害ではそのときのバッファ内のパケットのみが影響を受けるように、被害は一部

[†] 愛媛大学工学部

Faculty of Engineering, Ehime University

[☆] 現在、松下システムエンジニアリング (株)

Presently with Matsushita Systems Engineering

で抑えられる。また、トラフィックをサブネット全体にわたって平均化させることができる。

このように両方式にはそれぞれ長所と短所があり、現状のバケット通信方式では、仮想回線方式により高品質な通信を、データグラム方式により高速な通信を行っている。

3 エージェントによる動的経路制御

仮想回線方式では経路を固定的にすることが、データグラム方式ではバケット毎にルーティングすることが、それぞれの長所や短所をもたらしている。本研究では、これらの方式の長所を合わせ持つ方法として、経路を対象とした動的ルーティングを提案する。これは通信をする両ホスト間に仮想回線を設定し、ある区間の回線やノードにかかる負荷により、ユーザの求める QoS を満足できない場合には、その区間の経路を動的に変更するものである。

エージェントを用いることにより、仮想回線の持っている信頼性の高い通信を保持し、さらに動的な経路変更が可能となる。

以下、エージェントによる経路制御の概要を述べ、エージェントやノードに必要な機能についてまとめる。

3.1 エージェントによる経路制御の概要

通信要求が発生すると、送信元ホストと宛先ホストの間にコネクションを張る。コネクションを張る際、両ホスト間の仮想回線上のノードにエージェントプログラムを配布する。各ノードでエージェントは起動し仮想回線の管理を行うものとする。

通信を開始すると、ノードはバッファの占有率や自ノードから他ノードへの回線のトラフィックを監視し、エージェントはそれらの値をノードから受け取り、ユーザの求める QoS を満足できない場合、経路変更を行う。そこでエージェントは、障害の原因となっている区間を回避するように経路変更を行う。このため、自ノードの負荷が大きくて自ノードからの経路変更が適切でない判断したエージェントは、仮想回線の1つ上流のエージェントに経路変更を依頼する。また、経路変更後の仮想回線から外れるノード上のエージェントに対してはデータの損失や順序変更がないように処理をしてから削除する。

また経路変更の際、仮想回線にループが発生することがある。ループとなった経路での通信は明らかに無駄となるため、エージェントによりループの検出、削

除を行う。

上記のようにエージェントによる経路制御では、初期設定、経路変更、ループの削除が重要となる。それぞれのアルゴリズムを4章、5章、6章で述べる。

3.2 エージェントに必要な機能

エージェントに必要な機能のうち主なものを次にあげる。これらの動作の詳細は、次章以降で説明する。

- 仮想回線の設定
- ノードの保持するルーティングテーブルを参照して新しい経路を探索
- 新しい経路へ向かってエージェントのコピーの送信
- 他エージェントへの経路変更の依頼
- 経路から外れたエージェントの削除
- ループの検出及び削除

3.3 ノードに必要な機能

本論文ではエージェントを用いた通信方式を提案している。そこで3.1節で述べたエージェントの動作を実現するために、ルータなど従来からあるネットワーク中の各ノードにも新たな機能が必要となる。

- 自身のバッファの占有率や出線のトラフィック等の監視とエージェントへの提示
- エージェントプログラムが動作するためのプラットフォームの提供

4 エージェントを用いた仮想回線の初期設定

仮想回線の初期設定におけるエージェントの配置、動作のアルゴリズムを以下に示す。

- 送信元ホストは、仮想回線確立バケットを宛先ホストに送信する。その際にエージェントプログラムを付加して送信する。
- 宛先ホストに向かう間に通過した各ノードに、エージェントプログラムのコピーを配布する。
- 配布されたエージェントは各ノードで起動し、仮想回線上の1つ上流のエージェントに対して仮想回線設定開始確認応答バケットを送信する。

- 仮想回線設定開始確認応答パケットを受信したエージェントは、それを送信したエージェントにノード間の仮想回線の確立が完了したことを伝える。
- 宛先ノードで起動したエージェントは仮想回線設定完了パケットを送信元ノードに送信する。
- 送信元ノード上のエージェントは宛先ノード上のエージェントからの仮想回線設定完了パケットを受け取ることで仮想回線の確立を知る。

エージェントによる仮想回線の初期設定における時間経過を図1に示す。

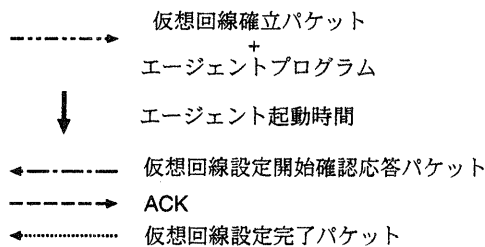
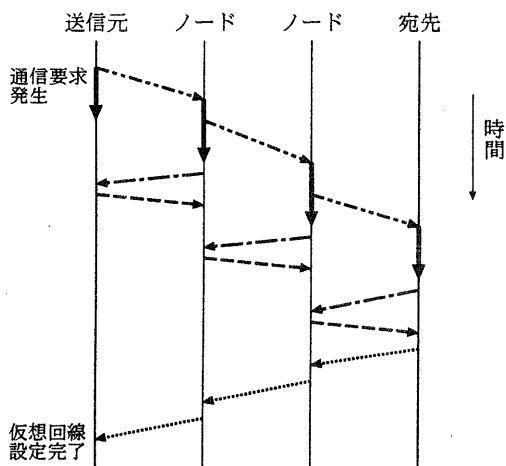


図1: 仮想回線の初期設定における時間経過
Fig.1: Set up of virtual circuit

5 エージェントによる経路変更アルゴリズム

エージェントを用いて仮想回線のコネクションを確立しホスト間で通信を行う場合、ノード間の枝のトラ

フィックやノードの負荷の増加による障害が生じた際、エージェントによる経路変更を行う。

ノード間の枝のトラフィックの増加による障害とは、あるノードに到着するデータパケットの総量はノードの処理能力を越えていないが、出線の1つにおいて通信要求がその能力を越えている場合である。

ノードの負荷の増加による障害とは、あるノードの入線からのメッセージ到着の総和がノードの処理能力を越えている場合である。

5.1 枝における障害発生時

負荷が大きいののは枝の部分であり、その枝に向かってパケットを転送しているノードの負荷は大きくないという状況である。従って、負荷の大きい枝を回避する新しい経路に変更する。以下にそのアルゴリズムを示す。

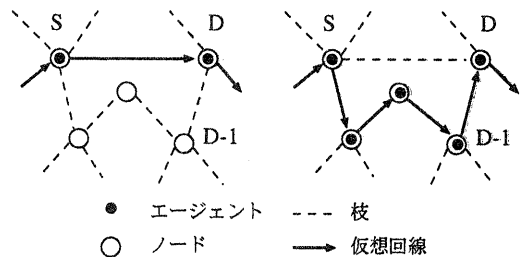


図2: 経路変更の例

Fig.2: Example of changing the virtual circuit

負荷の増加した枝の始点のノード(ノードSと呼ぶ)と枝の終点のノード(ノードDと呼ぶ)の負荷は大きくないため、ノードSから経路変更を行う。

まずノードS上のエージェントは次の転送先であるノードDへのデータパケットの送信を停止させ、それ以降ノードSの上流から送られてくるデータパケットを自ノードのバッファに蓄積させておく。

次にノードS上のエージェントはルーティングテーブルを参照しノードDへ至る別の経路を選択する。ノードSからノードDまでの新しい経路上のノードにはエージェントプログラムを配布していき、新しい経路上のノードで起動したエージェントは4章と同様の処理をし、仮想回線の設定を行う。

ノードSからノードDへのリンクは、ノードSがノードDへ送信したデータパケットの全てが正常に

届いたことをノード S 上のエージェントが確認できた時点で切断される。この切断の確認後、ノード S 上のエージェントはノード S から新しい経路方向へデータパケットの送信を再開させる。但し、新しい経路でノード S からノード D までの仮想回線の設定が完了していない場合、データパケットの新しい経路への送信は、新しい経路上のノードで既に起動している最も下流のエージェントのいるノードまでとし、そのノードでデータパケットの送信を停止させ、バッファに蓄積させる。

また、新しい経路におけるノード D とその1つ上流のノード(ノード $(D-1)$ と呼ぶ)との間の仮想回線の設定については次のようにする。ノード $(D-1)$ 上で起動したエージェントは、ノード D にエージェントプログラムを送らずに、仮想回線確立要求パケットを送る。このパケットを受け取ったノード D 上のエージェントは、ノード $(D-1)$ 上のエージェントに対して仮想回線設定開始確認応答パケットを送る。このパケットを受け取ったノード $(D-1)$ 上のエージェントはノード D 上のエージェントに、ノード $(D-1)$ と D との間の仮想回線の確立が完了したことを伝える。このようにするのは、ノード D 上には既にエージェントが存在するためである。

以上で経路変更は完了する。

5.2 ノードにおける障害発生時

負荷が大きいのはノード自身であり、そのノードからつながっている枝の負荷の大小に関わらずノード自身に障害が生じている。従って、負荷の大きいノードを回避する新しい経路に変更する。以下にそのアルゴリズムを示す。

あるノード N 上のエージェントは、自ノードの負荷の増加によりユーザの求める QoS を満足できないと判断すると、仮想回線上の1つ上流のノード(ノード $(N-1)$ と呼ぶ)上のエージェントに経路変更要請パケットと、ノード N の次の転送先であるノード(ノード $(N+1)$ と呼ぶ)の ID を送り、ノード $(N-1)$ からノード $(N+1)$ への経路変更を依頼する。

ノード $(N-1)$ からの経路変更については 5.1 節と同様であるが、新しい経路を探索する際、ノード N を通る経路は選択しないものとする。

また、新しい経路ではノード N を通過しないため、ノード $(N-1)$ 上のエージェントがノード $(N+1)$ へ

の経路変更を行う前に、エージェント消滅命令パケットをノード N 上のエージェントに送信する。このパケットを受け取ったノード N 上のエージェントは、実行すべき処理を全て行った後、自身を消滅させる。

ここで実行すべき処理とは、経路変更の前にノード $(N-1)$ からノード N へ送信されたデータパケットを全てノード $(N+1)$ へ送信させることと、再送制御を行いそれらが正常にノード $(N+1)$ に届いたことを確認後、それをノード $(N-1)$ 上のエージェントに伝えることである。

以上で経路変更は完了する。

6 ループの発生, 検出及び削除

6.1 ループの発生

エージェントによる経路変更は障害の原因となっている区間を回避するように行われるため、新しい経路にはループが発生する可能性がある。これは仮想回線上の各ノードやエージェントは、仮想回線の設定されている隣接ノード間のリンク情報しか管理していないためである。

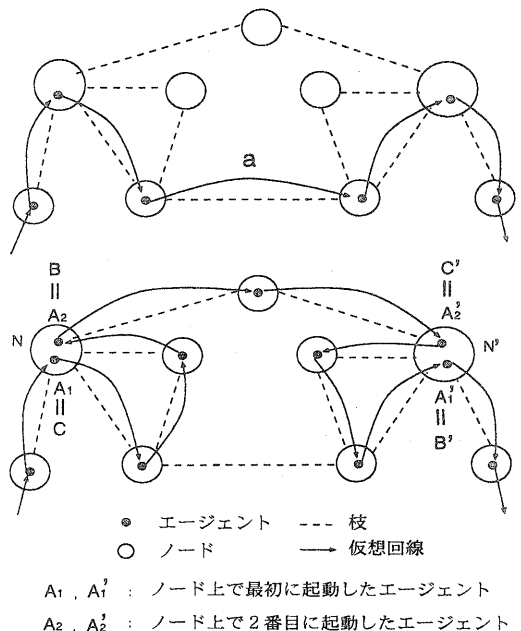


図 3: 経路変更によるループの発生

Fig.3: Loop by changing virtual circuit

例えば、図3の上図において枝 a のトラフィックが増加したため、図3の下図のように経路変更を行った際、2箇所ループが発生している。

6.2 ループの検出と削除

ループが形成されると、あるノードがループの始点と終点となる。仮想回線上のノードにはエージェントが起動されるので、このようなノードには同じ仮想回線を管理するエージェントが2つ起動していることになる。ループの存在しない仮想回線では、その仮想回線を管理するエージェントは経路上の各ノードに唯一つだけ存在する。従って、あるノードでエージェントが起動されたとき、そのノードに同じ仮想回線を管理するエージェントが既に起動していればループが形成されたことになる。

あるノード N で起動したエージェント A_2 は、同じ仮想回線を管理するエージェント A_1 が既に存在している場合、ループの発生を認知する。なお、エージェント A_2 はループを認知した場合でも目的とするノードへの経路探索を継続して行う。

次にエージェント A_2 はループの削除を以下のように行う。

エージェント A_2 はループの発生を知ることはできるが、どこにループが存在しているのか、つまり自ノードの、どの出線と入線がループになっているか分からない。

そこで、まずエージェント A_2 は A_1 にループが発生していることを伝える。そしてエージェント A_2 , A_1 はそれぞれの管理している仮想回線の下流のノードにループ検出パケットを送る。ループ検出パケットにはエージェント A_1 と A_2 のいるノード N の ID と、送出したエージェントの ID が書かれている。

ループ検出パケットを受信したエージェントは、自ノードの ID が書かれていなければループ検出パケットを下流のノードに転送し、自ノードの ID が書かれていれば自ノードの上流のノードがループの一部であることを知り、このパケットを回収する(このループ検出パケットを回収したエージェントを以下エージェント B と呼ぶ)。

また上で回収されたループ検出パケットに書かれているエージェント ID のエージェント(このエージェントを以下エージェント C と呼ぶ)の下流のノードがループを構成しているので、エージェント B はエー

ジェント C にそのことを伝える。

エージェント C は仮想回線上においてエージェント B よりも上流の回線を管理している。そこでエージェント C はループの方向を検出後、ノード N が下流へデータパケットを送信するのを停止させ、それ以降ノード N へ送られてくるデータパケットをバッファに蓄積させる。

次に、エージェント C は下流のノード上のエージェントに、エージェント消滅命令パケットを送る。このパケットを受信したエージェントは自ノードが保持しているデータパケットの全てを1つ下流のノードに送信させた後、エージェント消滅命令パケットを1つ下流のノードに送り、このパケットが正常に受信されたことを確認すると自らを消滅させる。このようにループを形成しているノード上のエージェントを、上流から順に消滅させていく。最後に、エージェント B がエージェント消滅命令パケットを受信すると、自ノードが保持しているデータパケットを全て1つ下流のノードに送信させた後、その送信先のノードの ID をエージェント C に伝え、自らを消滅させる。

エージェント B から得たノード ID のノードをエージェント C はデータパケットの転送先とする。これでループはなくなるので、エージェント C はノード N からのデータパケットの送信を再開させる。

以上でループの削除は完了する。

6.3 ループ削除の例

以下に、図4のようにループが発生する場合のループ削除の具体的な手順を述べる。

ノード N 上で後から起動したエージェント A_2 は同じ仮想回線を管理するエージェント A_1 がすでに起動されているので、仮想回線にループがあることを確認する。一方、エージェント A_2 は次の転送先として選択したノード ($N+3$) と仮想回線の設定を行う。

ループ削除にあたり、まずエージェント A_2 は A_1 にループが発生していることを伝える。そしてエージェント A_2 はノード ($N+3$) 上のエージェントへ、エージェント A_1 はノード ($N+1$) 上のエージェントへそれぞれループ検出パケットを送る。ノード ($N+3$) 上のエージェントに送信されたループ検出パケットは宛先ノードまで送信されて消滅する。一方、ノード ($N+1$) 上のエージェントに送信されたループ検出パケットはいずれノード N に送信される。エージェント A_2 は

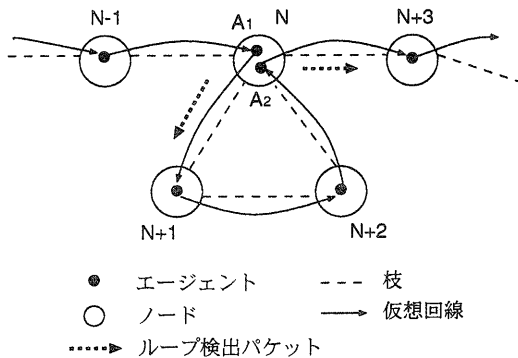


図 4: ループの検出
Fig.4: Loop detection

その packets を回収し、エージェント A_1 の下流からエージェント A_2 の上流でループが発生していることを知り、そのことをエージェント A_1 に伝える。

自らの下流方向にループが発生していることを知ったエージェント A_1 は、ノード N がノード $(N+1)$ へデータパケットを送信するのを停止させ、それ以降送られてくるデータパケットをバッファに蓄積させる。

次に、エージェント A_1 はノード $(N+1)$ の方向にエージェント消滅命令パケットを送信する。このパケットを受信したノード $(N+1)$ 上のエージェントは自ノードが保持しているデータパケットの全てをノード $(N+2)$ に送信させた後、エージェント消滅命令パケットをノード $(N+2)$ 上のエージェントに送信し、このパケットが正常に受信されたことを確認すると、自らを消滅させる。エージェント消滅命令パケットを受信したノード $(N+2)$ 上のエージェントでも同様に処理する。

そして、エージェント A_2 がエージェント消滅命令パケットを受信すると、自ノードの保持するデータパケットの全てをノード $(N+3)$ へ送信させた後、ノード $(N+3)$ の ID をエージェント A_1 に伝え、自らを消滅させる。

エージェント A_1 はデータパケットの転送先をノード $(N+3)$ に変更し、ノード N からのデータパケット送信を再開させる。

7 まとめ

本論文では、仮想回線方式において経路を対象とした動的ルーティングを行う方式を提案した。それを実現させるためにエージェントを利用し、エージェントによる有効な経路変更の方法とそれを行うために必要な機能を示した。

本方式は、エージェントにより通信を行うホスト間の経路を管理し、仮想回線方式の長所である信頼性の高い通信とデータグラム方式の長所である動的ルーティングを合わせ持つ方式として提案されており、また、ループを取り除く機能も持つことから次のような利点がある。

- 仮想回線上の実時間におけるトラフィック増加に対応した効率の良い経路変更が可能である。
- 仮想回線を動的に経路変更させることでデータ伝送の遅延やスループットの低下を改善できる。
- 仮想回線が本来持っている通信の信頼性を失うことなく通信を行うことが可能である。
- エージェントが経路を制御することで、エージェントに経路変更の知識を盛り込むだけでノードの機能を変更せずに様々な経路変更法が実現できるため、コストを抑えることが可能である。
- スループット重視や遅延重視等、通信を行うユーザ側の要求をエージェントに盛り込むことで、それに合わせた経路変更を行い、ユーザの求める QoS に応じた通信が可能である。

今後の課題としては、ノードでエージェントを実行するために必要となるノードの機能や、エージェント記述言語の仕様等の検討を行う必要がある。

謝辞

本研究の一部は、通信・放送機構の“ギガビットネットワーク利活用研究開発制度”(研究開発課題名: 異種超高速ネットワーク間の知的相互運用と tele-care への適用に関する研究開発) から補助を受けている。

参考文献

- [1] Andrew S. Tanenbaum, “Computer Networks Third Edition”, Prentice Hall PTR, (1996).