

データ転送プロトコルに関する性能要求からの プロトコルパラメータの決定

山口成基¹ 山口弘純¹ 谷口健一¹

¹ 大阪大学 大学院基礎工学研究科 情報数理系専攻

{s-yamagu,h-yamagu,taniguchi}@ics.es.osaka-u.ac.jp

本稿では、エンドノード間でデータ転送を行うプロトコルの設計を支援する一手法を提案する。提案手法では、あらかじめカラーベトリネット¹で記述したデータ転送制御機能(再送制御機能, フロー/レート制御機能)のモジュール群を用意しておき、設計者はそれらの機能をプロトコルに付加するか否かを指定する。また、設計対象とする層の上位層から与えられる転送データの特性及び、下位層におけるデータ転送遅延特性が与えられたとき、提案手法では対象層が実現するデータ転送率と、データ転送遅延特性を、対象層内部のプロトコルパラメータの関数として自動導出する。設計者はそれらの値に対する要求値を指定することでプロトコルパラメータ値が決定される。プロトコル仕様は機能モジュールを組み合わせ、決定されたパラメータ値を埋め込むことで得られる。最終的には得られたプロトコル仕様記述からネットワークシミュレータ^{ns}の実装コードへ自動で変換し、得られたプロトコルのシミュレーションによる性能評価を容易に行えるようにすることを目標とする。

Deriving Parameters of Data Transfer Protocols from Performance Requirements

Shigeki Yamaguchi¹, Hirozumi Yamaguchi¹ and Kenichi Taniguchi¹

¹ Graduate School of Engineering Science, Osaka University

In this paper, we propose a technique for designing data transfer protocols. In the proposed technique, we describe the functionalities of data transfer control mechanism (retransmission control mechanism and flow/rate control mechanism) as a set of Coloured Petri Net's modules. Then protocol designers specify some of those functionalities that they would like to add the protocol to be designed. Also, given the characteristics of data from the upper layer protocol and the delay distribution in the lower layer protocol, we express the data transfer ratio and the delay bound of the target protocol as a function of the protocol's parameters. The protocol parameters are determined satisfying the requirements of designers to the data transfer ratio and delay bound. Now we are trying to formalize a technique which automatically derives an executable code in network simulator from a derived CPN protocol specification. This technique would help designers to evaluate the performance of the designed protocols.

1 はじめに

近年、無線通信、衛星通信など様々なネットワークインフラストラクチャの整備発展に伴い、それらのネットワーク特性や送信データの特性に応じた既存のデータ転送プロトコルの性能調整やアプリケーションレベルでの新しいプロトコルの構築が必要とされる機会が多い。例えば、映像ビデオ会議のようなインタラクティブな実時間動画像配信をエラー率の高い無線通信網を介して行う場合は、パケット再送による遅延は許容できないため、画像フレーム間の依存関係に基づきアプリケーション層で設定されたパケット優先度に基づき、送信側でパケット棄却を行うようなレート制御が望まれる。また、ストリーミング配信などでは多少の遅延は受信ノードのバッファリングで許容できるため、被依存度の高い重要なフレームのパケットのみを再送しながら実効転送率の向上を図るプロトコルが望まれる。

一般に、そのようなプロトコルの設計の際には、要求記述からのプロトコル仕様記述(protocol specification)、仕様の正当性検証(validation)、仕様からのシミュレータコードの実装(implementation)、シミュレータによるシミュレーション実験(performance evaluation)といった設計段階を経て、かつ性能向上のために各段階への適当なフィードバックを繰り返す。しかし、一般にそのプロトコルが利用されるネットワーク特性をプロトコル仕様レベルで考慮しながら記述することは単純ではない。さらに、シミュレータコードの実装などには多くの手間が必要となる。従って、ラビッドプロトタイプングのためなどには設計者は遅延特性や転送率に関する要求及び必要な制御機能のみを指定するだけで、プロトコルの仕様記述及び実装がなるべく機

械的に行えることが望ましい。

これまでに、SDL やベトリネットなどの形式言語でプロトコル仕様を記述し、その上で直接性能評価を行うことで設計から性能評価までを一貫して支援する試みがいくつかなされている[1, 2, 3, 4]。しかしそれらは形式言語で記述された仕様に通信路をモデル化したモジュールを組み合わせ、性能評価をいかに形式モデル上で行うかのみに着目している。しかし、実際にはプロトコル仕様記述自身はそのプロトコルに対する性能要求を元に設計者が自身の知識と経験に基づき行わなければならないこと、また、実際の性能評価はマルチホップ環境など現実のネットワークをより忠実に再現可能なシミュレータで行うことがより望ましいこと、などの問題がある。

本稿では2ノード間のデータ転送を制御するプロトコルの仕様記述から性能評価までを支援する一設計法を提案する。提案手法では、データ転送プロトコルの代表的な機能(再送制御機能, フロー/レート制御機能)をあらかじめカラーベトリネット¹で機能モジュール群として記述しておき、設計者はそれらの機能を組み入れるか否かを指定する。また、設計対象とする層の下位層のネットワーク特性がパケットの遅延分布として与えられ、さらにその上位層からの転送すべきデータパケットの(パケット間の)優先度関係や送信間隔などが与えられるとし、そのもとで実現可能なデータ転送率(信頼性)及び、パケットの遅延分布が、対象層で利用するプロトコルパラメータの関数として自動で提示できる。設計者はその関数式を元に、最大遅延、平均遅延、デー

タ転送率に関する要求を指定することでそれらプロトコルパラメータの値を決めることができる。プロトコル仕様は指定されたモジュール群を組み合わせ、決定されたプロトコルパラメータ値を埋め込むことで得られる。本手法により、設計者はあるネットワーク環境のもとで要求をなるべく満足するプロトコル仕様記述を容易に得ることができる。プロトコル仕様記述にはカラーペトリネットを用いる。カラーペトリネットではパケットレベルでのモデル化が容易であり、デッドロック検出や時間シミュレーションなどを行う機能を持つ design/CPN[8] などの処理系が利用できるという利点がある。さらに、ネットワークシミュレータを用いた性能評価も容易に行えるよう、得られたプロトコル仕様記述からネットワークシミュレータ NS の実装コードへ自動で変換する手法についての検討を行っている。

2 プロトコルの機能決定方法

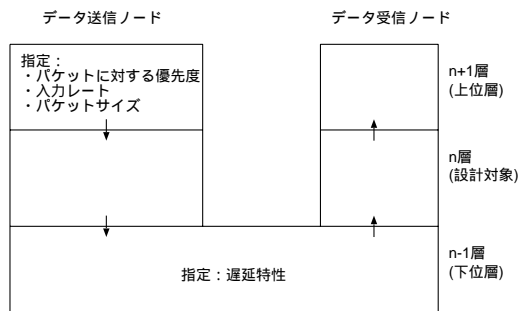


図 1: プロトコル階層構造

提案方式では、設計対象とするデータ転送プロトコル層（以下対象層と呼ぶ）を第 n 層とした場合に、上位層（第 $n+1$ 層）の送信ノードから与えられるデータパケットについて、パケットシーケンス番号に対する優先度指定、入力レート、及びパケットサイズ（これを以下配送データ特性とよぶ）が指定され、下位層（第 $n-1$ 層）については配送するパケットが n 層の送信ノードに入力されてから n 層の受信ノードに届くまでにかかる遅延時間の分布特性（これを以下遅延分布とよぶ）が指定されるとする（図 1）。

n 層においてデータ転送の代表的な制御機能（再送制御機能、フロー制御機能、およびレート制御機能）を用いる場合に、 n 層が入力データに対し実現可能な遅延特性とデータ転送率をそれらの制御機能が利用するパラメータの関係式として表現する。設計者はどの制御機能を利用するか、及びその際どのような性能を n 層に要求するかも指定することで、その指定をなるべく満足する各制御機能のパラメータ値を決定できる。

一般には、 n 層はトランスポート層、 $n-1$ 層はネットワーク層、 $n+1$ 層はアプリケーション層と考えられるが、提案手法ではそれらの層に基づく特定のプロトコルを設計対象としているわけではないため、例えば $n-1$ 層に UDP を仮定し、アプリケーション層に再送制御機能を持つプロトコルを設計したい場合などにも適用できる。

2.1 上位層からの転送データの情報

上位層である $n+1$ 層から対象層である n 層に渡される転送データに関する情報として、以下が与えられているとする。

- パケットに対する優先度
 $priori(rx+i) = j$
 $(0 \leq i \leq r-1, 0 \leq j \leq p_{max})$
 但し、 $rx+i$ はパケットシーケンス番号
- 入力レート In_Rate (bps)
- パケットサイズ Pkt_Size (kB)

優先度はパケットシーケンス番号の関数（パケット r 個毎のパターンを表す関数）として 0 から p_{max} の値で指定する。入力レートは $n+1$ 層から n 層へのデータ入力レートであり、 Pkt_Size はその各パケットの大きさを表す。いずれもある定数値とする。

2.2 下位層の遅延特性

- 遅延平均 D_{μ}
- 遅延分散 D_{σ}

通信層の特性は簡単の為、遅延平均と遅延分散を与えることで対数正規分布により近似する [10]。これにより、遅延時間の幅の大きな通信路から、遅延時間がほぼ一定であるような通信路まである程度近似することができる。また、パケットロスは無限大の遅延として表現できる。

2.3 プロトコルに付加可能な制御機能

次に、対象層のプロトコルに付加可能な制御機能と、それらの制御機能がどのようなパラメータを使用するかについて述べる。

1. 再送制御：再送に関しては以下の 2 つの方針を検討した。
 - (a) 許容遅延時間内は再送を繰り返す再送制御
 例えば、ストリーミングなど、最大遅延まではバッファリングなどにより許容されるが、それを過ぎた場合はパケットが有効でなくなる場合は、許容遅延時間 MT を上限として再送を行なう方針の制御が有効であると考えられる。この制御機能はタイムアウト時間 T_{out} 、許容遅延時間 MT をパラメータとして利用する。
 - (b) 選択的再送制御
 差分データを用いる (MPEG など) 場合、メインとなるピクチャ (I ピクチャ) が損失するとその後の差分ピクチャ (P, B ピクチャ) が再生できないことになる。逆に B ピクチャが損失しても他ピクチャの再生には影響しない。このような場合などには、入力されるパケットに優先度を付け、優先度ごとに再送回数を変え、実効データ転送率を向上させる方針の制御が有効であると考えられる。この制御機

能は受信ノード側のタイムアウト時間 T_{out} , 優先度毎の許容遅延時間 MT_j (j は優先度) 及び再送回数 N をパラメータとして利用する .

2. フロー制御

受信ホストの受信バッファサイズ以上のパケットが連続して伝送されると、受信処理が間に合わない場合にバッファ溢れを起こし、パケットが廃棄される。このような状況为了避免するため、フロー制御(ウィンドウ制御)機能では受信者は送信者に連続して伝送可能なデータ量(バーストサイズ、またはウィンドウサイズ)を明示的に知らせる。提案方式におけるフロー制御機能はバーストサイズ BS , バーストサイズ分のデータ送信間隔 BS_INT をパラメータとして利用する。

3. レート制御

受信ホストの処理能力と比較して、ネットワークの回線速度が低く、しかも十分大きな受信バッファサイズがある場合にフロー制御を行うと、ウィンドウサイズが大きくなりネットワークが輻輳する恐れがある。そこで、レート制御機能では輻輳解消の為に送信レートそのものを制御する。受信者はデータの受信レートを監視し、この情報を一定時間毎に送信者に通知する。送信者は通知されたレートに送信レートを適応させることで、ネットワークに過大な負荷がかかるのを避ける。提案方式におけるレート制御機能は送信レート S_Rate と、フロー制御においても使用されるバーストサイズ BS , バーストサイズ送信間隔 BS_INT をパラメータとして利用する。

以上の各制御機能で利用されるパラメータを以下にまとめる。

- タイムアウト時間 T_{out} (s)
- 許容遅延時間 MT または MT_j (s)
- バーストサイズ BS (kB)
- バースト送信間隔 BS_INT (s)
- 送信レート S_Rate (bps)

タイムアウト時間はパケットを送信してからその受信応答(ACK)を受信するまでの待ち時間である。この時間以内にACKを受信できない場合はパケットが消失したとみなし、再送を行う条件が満たされていれば再送を行う。許容遅延時間は、パケットを送信後この時間が経過するまでに発生したタイムアウトに対しては、再送を行うことを表すパラメータである。選択的再送制御を行う場合はパケット優先度 j に対する許容遅延時間 MT_j をパラメータとして利用する。バーストサイズは送信時に複数のパケットをまとめて送る場合の合計パケットサイズの上限を表す。バースト間隔はバーストサイズ分のパケットを送信する間隔である。送信レートは単位時間に送信するデータ量を表す。

2.4 制御パラメータによるプロトコルの特性の導出

以下では、前節の各制御機能を選択した場合に、 n 層で実現される遅延特性及びデータ転送率を、それらの

制御機能のパラメータの関係式として表す。提案手法では、その関係式をもとに、設計者がどの制御機能を組み入れるか、また n 層に要求する遅延分布の最大値、平均値及びデータ転送率を指定する。それらの指定により、その要求をなるべく満足する適切なパラメータ値を決定する。

1. 再送制御を組み入れた場合

- (a) 許容時間内は再送を繰り返す制御を組み入れる場合

提案方式では遅延時間の分布を対数正規分布で近似しているが、その確率密度関数は以下のようになる。

$$\varphi(x) = \frac{1}{\sqrt{2\pi}D_\sigma x} \exp\left(-\frac{(\log x - D_\mu)^2}{2D_\sigma^2}\right)$$

遅延 T 以下でパケットが受信ノードへ到着する率 $P(\%)$ は、

$$\begin{aligned} P(-\infty < X < T) &= P(-\infty < X^* < \frac{T - D_\mu}{D_\sigma}) \\ &= \varphi\left(\frac{T - D_\mu}{D_\sigma}\right) \end{aligned}$$

さらに許容遅延時間を MT とすると、 N 回目の再送を行うかどうかは $MT \geq T_{out} \times N + T$ が $N-1$ 回目のタイムアウト発生時に成り立つか否かで判定する(成り立つ場合は再送を行う)。ここで N 回目の再送までにパケットが受信ノードに到着する確率を P_N とすると、 P_N は

$$\begin{aligned} P_N &= P_{N-1} + (1 - P_{N-1})P_0 \\ &= (1 - P_0)P_{N-1} + P_0 \\ &= (1 - P_0)^{N-1}P_0 + \frac{P_0((1 - P_0)^N - 1)}{(1 - P_0) - 1} \\ &= (1 - P_0)^{N-1}P_0 - (1 - P_0)^N P_0 + 1 \end{aligned}$$

で得られる。但し

$$P_0 = \varphi\left(\frac{T - D_\mu}{D_\sigma}\right)$$

であり、 N は MT と T_{out} で決定される媒介変数である。この P_N が n 層のデータ転送率を表す。

- (b) 選択的再送制御を組み入れる場合

優先度が j のパケット数が R 個中に r_j 個であり、それらのパケットを最大で N_j ($0 \leq j \leq P_{max}$) 回再送とする。この場合、優先度 j のパケットは全体の $\frac{r_j}{R}$ を占める。選択的再送制御を組み入れたときのパケット到達率は

$$\sum_{j=0}^{P_{max}} \frac{r_j}{R} \times P_{N_j}$$

となる。この値が n 層のデータ転送率を表す。

2. 送信者がネットワーク状況に応じて優先度の低いパケットを棄却する機能を組み入れた場合

レート制御を行っているとき、 $n+1$ 層からのデータ入力レート In_Rate より $n-1$ 層へのデータ送出レート S_Rate の方が小さくなると

$$DR = \frac{Pkt_Size}{In_Rate - S_Rate}$$

毎にパケットを1つ落とすとすする。ただし、落とすパケットの優先度が高ければ優先度の低いパケットが出てくるまで待ち、それを棄却する。 $n+1$ 層からの入力パケット全体に対する、棄却するパケットの割合 (%) は

$$S_Drop = \frac{Rkt_Size}{DR \times In_Rate}$$

となる。ここで、優先度 K 以下のパケットを棄却対象としたときに、パケット到達率は

$$\begin{aligned} P_0 & \quad \text{優先度 } K+1 \text{ 以上} \\ (1 - S_Drop) \times P_0 & \quad \text{優先度 } K \text{ 以下} \end{aligned}$$

従って n 層のデータ転送率は

$$\sum_{j=0}^K \frac{r_j}{R} \times (1 - S_Drop) \times P_0 + \sum_{j=K+1}^{P_{max}} \frac{r_j}{R} \times P_0$$

となる。

2.5 性能要求に基づく制御パラメータ値の決定

まず、プロトコルに組み入れたい制御機能を選択し、それに関連するパラメータの値を決定していく。

- 制御機能を組み入れない場合

送信者が一定時間毎にパケットを送出するだけのプロトコルとなり、プロトコル設計者は送信者が連続して送り出すパケットのサイズ (BS) とその送信間隔 (BS_INT) を決めればよい。

- 1. (a) または (b) を組み入れる場合

これらの機能のいずれかを選択すると、再送までの ACK 待ちタイムアウト時間 (T_{out})、再送回数 (N)、データ転送率 (P_N) の間の関係式が示される。

$$\begin{aligned} P_N &= (1 - P_0)^{N-1} P_0 - (1 - P_0)^N P_0 + 1 \\ P_0 &= \varphi \left(\frac{T - D_\mu}{D_\sigma} \right) \end{aligned}$$

また、MT は $MT \geq T_{out} \times N + T$ を満足する必要がある。設計者は P_N と MT に関して、例えば許容遅延を長くしてリアルタイム性を下げてもパケット到着率を上げたい、またはリアルタイム性を重視して許容遅延を短くしたいといった状況に応じて適切な値を指定することで、パラメータ値を決定できる。

- フロー制御を組み入れる場合

BS は受信者からのフィードバックにより決定されるため、バーストサイズ送信間隔 BS_INT のみを指定する。BS_INT=0 なら受信者のバッファが空いている限りパケットを送り続けることとなる。このように、増えたバッファスペースを使ってデータセグメントを大量に送信すると、ネットワークリソースの利用効率が悪くなる。これは無能ウィンドウ症候群と呼ばれる [9]。これを回避する方法が TCP 等により確立されているが、その方法を取るとリアルタイム性の低下、少量のデータの転送を必要とするアプリケーションに向かない等の問題もある。ここでは、リアルタイム性を重視するか、ネットワークリソースの有効利用を取るかにより BS_INT の値を適切に決定する。なお、一般にフロー制御はネットワークの動的状況に依存するため様々なポリシーが存在し、それらを考慮してデータ転送を表すことを現在検討中である。

- レート制御を組み入れる場合

受信者から受信レートが送信者に報告される。送信レート S_Rate を (報告された) レートに近づけるため、 $S_Rate = BS / BS_INT$ となるように BS、BS_INT の値を決定する。しかし、BS、BS_INT 共に大きな値を取ると送信レートの変動が大きくなり、実現したい S_Rate から離れてしまうことになる。例えばフロー制御と組合せて行っていた場合、バッファサイズの空きが大きい送信レートは低くしなければならない時にこの問題が起こる。これを解決するために、受信者が連続して受け取れた最大データサイズ credit (受信ノードから送信ノードへ通知される)、バーストサイズ更新間隔 BS_REFLESH という値を設定する方法がある。データは BS_INT 間隔で credit サイズずつ送信し、送信する度に BS から credit を引く。BS が 0 以下となると送信は停止する。BS_REFLESH 間隔で BS は ACK によって通知される値に更新される。なお、データ転送率及び MT に関する値を指定することで、パラメータ S_Rate を決定することもできる (なお、再送は行わないので $MT = T$)。

3 プロトコル仕様の導出

制御機能とそれらのパラメータ値が決定されると、提案手法では次にカラーペトリネットにて記述されたプロトコル仕様を導出する。

3.1 ペトリネットとカラーペトリネット

ペトリネットはプレースとトランジションという 2 種類の節点をもつ有向 2 部グラフであり、通常、図 2 のような図表現が可能である。

各プレースには非負整数が割り当てられ、これをマーキングと呼ぶ。ペトリネットではマーキングによりシステムの状態を表す。トークンの移動はトランジションの発火によって起こる。トランジション t は各入力プレース p 上にその間のアークの重み M 以上のトークンが存在するとき発火可能となる。図 2 のマーキングでは $T1$ 、 $T2$ が発火可能である。

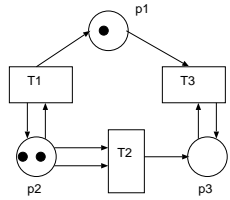


図 2: ペトリネットの図表現

カラーペトリネット [8] はトークンに指定した型 (カラー) の要素を割り当てることができるペトリネットである。また、トランジション毎にそのカラーを用いたガード条件 (発火条件) を指定することができ、トランジションの発火は各入力プレースにアークの重み分のトークンが揃うだけでなく、ガード条件も満たされた時に可能となる。

3.2 機能モジュールとパラメータ値のマッピング

図 3 はいずれの制御機能も付加しない場合のカラーペトリネットで記述されたプロトコル仕様である (これを基本モジュールと呼ぶ)。 $n+1$ 層からのデータ送信コマンド output を受けたら、受信者とのコネクションを確立しデータを単純に送信する機能を表す。TCB (Transmission Control Block) はデータ転送時の制御情報を保持するプレースであり、1 つのコネクションに対しコネクション ID を保持する 1 つのトークンが生成される。送信パケットバッファは $n+1$ 層から送られてきたデータがパケットとして保存される FIFO のバッファである。このプレースのトークンはその値としてパケットのリストを保持する。

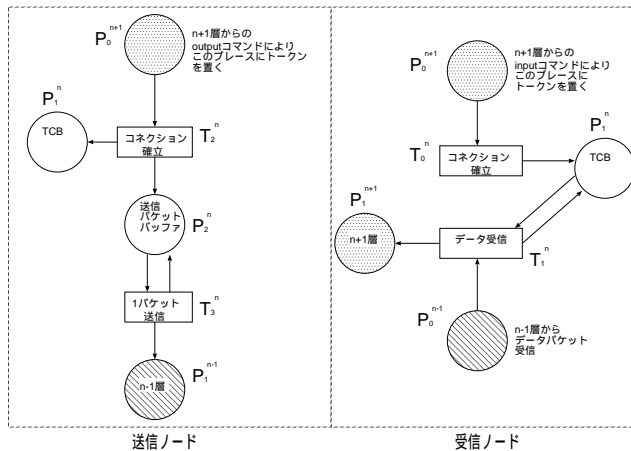


図 3: 基本モジュール (送信ノード及び受信ノード)

図 4 はフロー制御機能、またはレート制御機能を付加する場合に図 3 の送信者基本モジュールに追加するモジュールである。図における太線は追加する部分を表す。このモジュールにより図 3 の送信パケットバッファプレースと 1 パケット送信トランジションの間に送信待ちプレース、タイマプレース、次送信バースト取り出しトランジションが追加される。まず、タイマプレ-

スでは入力されるトークンの時間カラーを BS_INT 時間進めることで次送信バースト取り出しトランジションが次に発火するまでの時間を制御する。送信待ちプレースではトークンのカラーとして現在送信可能であるパケットのリストを保持している。次送信バースト取り出しトランジションは TCB 内の BS 値 (受信者からフィードバックされる) 分のデータを送信可能とするために、送信パケットバッファプレースのリストから適切な量のパケットを消去し、送信待ちプレースのリストに繋ぐ。

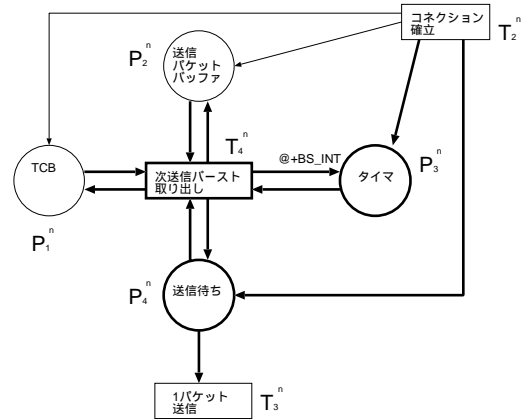


図 4: フロー制御 / レート制御モジュール (送信ノード)

図 5 は再送制御を追加するためのモジュールである。図における太線は追加する部分を表す。フロー制御、及びレート制御モジュールとは独立しており、いずれか一方を組み込むことも両方組み込むことも可能である。追加されるのは、待ち状態プレース、排他制御用プレース、パケット再送トランジション、パケット棄却トランジションである。待ち状態プレースにはパケットが 1 つ送信される毎に 1 つのトークンが生成される。このプレース内におけるトークンはその値としてタイムスタンプ、再送回数 N 、パケットシーケンス番号 seq を保持する。タイムスタンプはトークン生成時に現在時間 $+T_{out}$ にセットされ、時間 T_{out} の間は発火しない。これにより、タイムアウト時間 T_{out} だけ待つことを表す。送信回数 n は再送を含めこのパケットを送信した回数が保持され、この値を用いて次の再送を行うかどうかの判定が行われる ($MT \geq T_{out} \times n + T$ を満たせば再送)。このパケットに対する ACK を受け取っていればパケット棄却トランジションによりパケットが棄却される。そうでなければパケット再送トランジションが発火し、そのパケットが再送される。プレース H は送信排他制御用プレースである。優先度により再送回数を変更する場合はトークンに優先度が値として保持し、トランジション T_6^n において優先度毎の許容遅延時間 MT_m が参照される。

図 6 はフィードバックを必要とするフロー制御、レート制御または、再送制御のモジュールを組み込むときに同時に組み込まれるモジュールである。受信者には受け取ったパケット毎に ACK を返す機能、送信者には受信者から返された ACK を受け取り TCB 内の情報を更新する機能を付け加える。

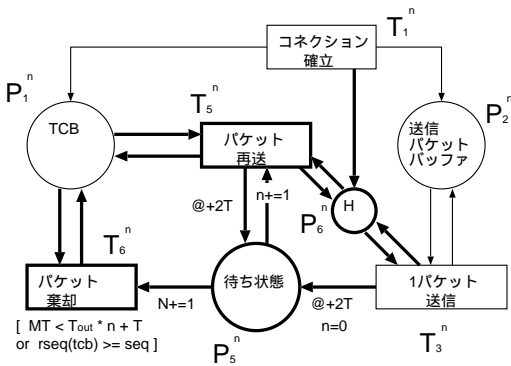


図 5: 再送制御モジュール (送信ノード)

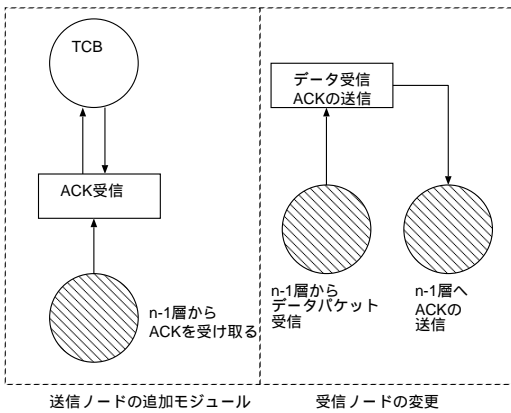


図 6: フィードバックモジュール (送信ノード及び受信ノード)

4 おわりに

本稿では、エンドノード間でデータ転送を行うプロトコル層の設計を支援する一手法の提案及び検討を行った。本研究では、あらかじめカラーペトリネットで記述した機能モジュール群を用意し、その内部パラメータを決めるためのプロトコルパラメータの関係式を導出できる。本手法により、プロトコルの設計者はどの制御機能を利用するか、及びその際どのような性能を要求するかを指定することで、その指定をなるべく満足する各制御機能を備えたプロトコル仕様を得ることができる。なお現在、カラーペトリネットで記述されたプロトコル仕様を、ネットワークシミュレータ NS にモジュールとして組み込める C++ のコードへ自動変換する方法について検討している。基本的にはプロトコル仕様の各モジュールを C++ のあるクラスとして実装することで NS へ組み込むことができると考えられる。また、今後の課題としては、フロー制御機能、レート制御機能の内部パラメータであるバーストサイズ BS、バーストサイズ送信間隔 BS.INT を決めるときに、その値がデータ転送にどのような影響を与えるかを設計者に対して示すことのできる指標の検討、また、本手法に基づきデータ転送プロトコルの設計からシミュレーションまでを一貫して支援する支援系の作成などが挙げられる。

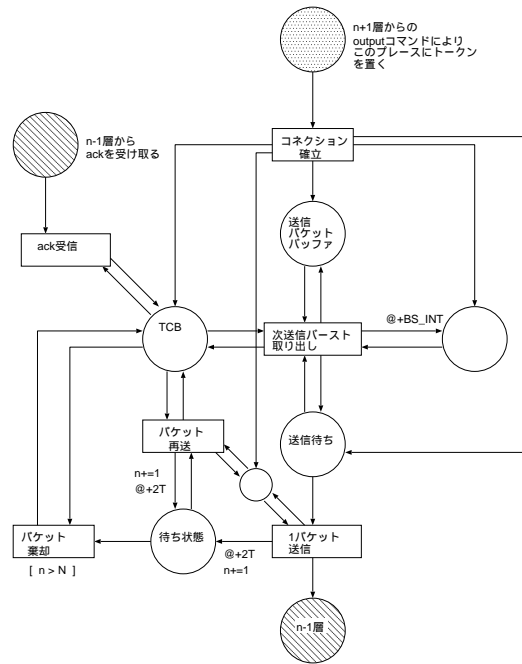


図 7: 再送制御機能とフロー制御機能を付加した場合のプロトコル仕様 (送信ノード)

参考文献

- [1] B. Geppert, A. Kühlmeyer, F. Röbler, and M. Schneider, "SDL-Pattern Based Development of a Communication Subsystem for CAN," Proc. of FORTE XI/PSTV XVIII'98, pp.197-212, 1998
- [2] J. Templemore-Finlayson and E. Borcoci, "Simulating Multicast Transport Protocols in Estelle," Proc. of FORTE XIII/PSTV XX, 2000.
- [3] M. H. Erdogmus and R. Johnston, "On the Specification and Synthesis of Communicating Processes," IEEE Transactions on Software Engineering, Vol. 16, No. 12, pp.1412-1425, 1990.
- [4] J. C. A. de Figueiredo and L. M. Kristensen, "Using Coloured Petri Nets to Investigate Behavioural and Performance Issues of TCP Protocols," Proc. of 2nd Int. Workshop and Tutorial on Practical Use of Coloured Petri Nets and Design/CPN, 1999.
- [5] D. Y. Chao and D. T. Wang, "An Interactive Tool for Design, Simulation, Verification, and Synthesis of Protocols," Software-Practice and Experience, Vol. 24, No. 8, pp. 747-783, 1994.
- [6] A. Khoumsi and K. Saleh, "Two Formal Methods for the Synthesis of Discrete Event Systems," Computer Networks and ISDN Systems, Vol. 29, pp. 759-780, 1997.
- [7] V. P. Sliva, T. Murata, and S. M. Sharz, "Protocol Specification Design Using an Object-Based Petri Net Formalism," Int. Journal of Software Engineering and Knowledge Engineering, Vol. 9, No. 1, pp. 97-125, 1999.
- [8] K. Jensen, "Coloured Petri Nets Basic Concepts, Analysis Methods and Practical Use," ()Vol. 1-3), Springer, 1997.
- [9] P. Miller 著, 荻田幸雄 監訳, "マスタリング TCP/IP 応用編", オーム社, 1998.
- [10] 藤本, "実測に基づいたインターネットにおけるパケット転送遅延のモデル化とプレイアウト制御への適用", 大阪大学特別研究報告, 2000.