

リアルタイム通信集約プロトコル向け輻輳制御手法

金井 遵† 大西直哉† 松山拓紀†

本稿では通信集約プロトコル向けの輻輳制御手法を提案する。提案手法ではネットワークの混雑度やサーバの負荷に応じて、パケットの集約周期等の通信集約プロトコル固有のパラメータを制御し、通信可能な実データ量を極力減らさずに輻輳を回避する。実装と評価により、輻輳制御無しの場合や TCP/IP を利用する場合と比較して、ネットワークの混雑時にも通信遅延の増加を防ぐ効果が大きいことを確認した。

A Method of Congestion Control for a Real-time Aggregation Protocol

Jun Kanai†, Naoya Ohnishi† and Hiroki Matsuyama†

In this paper, we describe a congestion control method of a communication aggregation protocol for real-time control systems. This method controls aggregation parameters such as an aggregation interval to avoid decreasing of real traffic capacity. These parameters are controlled by network load and server load. Implementation and evaluation indicate that it suppresses latency in the case of network congestion.

1. はじめに

従来の FA (Factory Automation), 各種プラントなどの制御システムは, 応答時間の制約から各制御対象の近傍に制御装置を設置する構成が主流であった。この構成にはシステム構成の柔軟な変更が困難な問題があり, 近年はクラウド技術を活用し, 柔軟な構成変更に対応可能な高信頼なリアルタイム制御システムが求められている^[1]。クラウド上のサーバと連携してリアルタイムな制御を行うシステムでは, 多数のパケットがサーバやネットワーク機器に集中して通信遅延やパケットロスが発生することが課題となる。制御システムでは, 一般的に一定周期の制御や要求時間内の応答が必要であるため, 通信遅延や応答時間を保証することが重要である。

そこで筆者らは, 制御対象から送信される複数のパケットを集約・分離し, サーバと通信することで通信遅延時間を低減する通信手法(以降, リアルタイム通信集約プロトコルと呼ぶ)を提案している^[2]。本稿ではリアルタイム通信集約プロトコルにおいて, ネットワーク混雑度, サーバ負荷に応じて適切な通信量の制御を行う輻輳制御手法を提案し, 実装と評価を行った結果を述べる。

2. リアルタイム通信集約プロトコルとその課題

2.1. プロトコルの概要

リアルタイム通信集約プロトコルでは, 図1のように制御装置の近傍にゲートウェイを設置し, 複数の制御装置(クライアント)からのパケットをゲートウェイで集約して1つのパケット(集約パケット)にまとめ, サーバに送信する。一方でサーバから送信されるパケットについても, サーバ側で複数のパケットをまとめて集約パケットとしてゲートウェイに送信される。ゲートウェイでは集約パケットを受信すると, 個々のパケットに分離して各制御装置に送信する。これにより, ゲートウェイとサーバ間で通信されるパケットの数を減らすことができる。

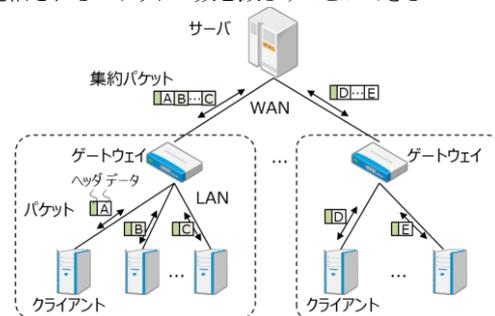


図1 ネットワーク構成

†(株)東芝 Toshiba Corporation

制御装置とゲートウェイ間は TCP/IP で通信を行い、ゲートウェイとサーバ間は遅延が大きい WAN 環境を想定して UDP/IP を用いて通信を行い、遅延を抑える。

ゲートウェイとサーバでは一定周期ごとに集約パケットを生成し、送信する。すなわち、集約パケット送信後に一定時間が経過すると、前回の集約パケット送信後に到着したパケットをまとめて、集約パケットを生成する。このときパケットが大きすぎるとパケットロス発生率が上がるため、集約パケットの最大サイズを決め、その範囲内で集約パケットを生成する。パケット数が多い場合には、最大サイズに収まらないパケットは送信待ちとし、次の集約周期で生成する集約パケットに含める。

また、UDP/IP は再送機能を持たない。そこでリアルタイム通信集約プロトコルでは通信の高信頼化のため、同一の集約パケットを複数回送信することでパケットの到着可能性を高める冗長送信機能を持っている^[3]。

2.2. リアルタイム通信集約プロトコルの利点と課題

制御システムの IoT 化により、多数のセンサからの情報のサーバへの送信など、小さなサイズのパケットが大量に送信されるケースが増えている。この場合、多数のパケットがサーバや経路上のネットワーク装置に到着し、負荷増大を招く。また、通信量全体に占めるヘッダの割合が大きくなり、ネットワーク帯域の利用効率が下がる。リアルタイム通信集約プロトコルでは、複数の小さなパケットを1パケットとすることができ、サーバ等の負荷低減やネットワーク帯域の利用効率改善ができる。

一方でリアルタイム通信集約プロトコルではゲートウェイとサーバ間の通信に UDP/IP を利用するため、輻輳制御機能を持たない。すなわち、ネットワーク混雑時やサーバの負荷増大時に大量にパケットを送信してしまい、輻輳を招く可能性がある。TCP/IP 向けの各アルゴリズム^[4]のようにウィンドウサイズを変動させて輻輳を回避することも考えられるが、時間あたりに通信可能なデータ量が減る。そこで本稿では、リアルタイム通信集約プロトコル固有のパラメータを調整し、実通信量をできるだけ落とさずに輻輳を回避する手法を提案する。

3. リアルタイム通信集約プロトコル向け輻輳制御手法

3.1. パラメータの概要と変更時の影響

リアルタイム通信集約プロトコル固有のパラメータ変更による輻輳制御手法を設計するにあたり、各パラメータ変更が性能に与える影響を分析した。設定できるパラメータには、集約パケットの最大サ

イズ S 、集約間隔 I 、冗長送信回数 N があり、パラメータ変更が性能に与える影響を分析した結果を表 1 に示す。

表 1 パラメータ変更と性能への影響

	変化方向	リアルタイム性	総通信量	サーバ負荷	通信路上でのパケットロス率
最大サイズ S	増	○	×	×	—
	減	×	○	○	—
集約間隔 I	増	×	○	○	—
	減	○	×	×	—
冗長送信回数 N	増	—	×	—	○/×
	減	—	○	—	×/○

表 1 よりパラメータ変更には利点と欠点がある。例えば集約間隔を長くすると、時間あたりに送信される集約パケットの数が減り、総通信量やサーバ負荷を減らせる。一方で通信のリアルタイム性は悪化する。つまりサーバ負荷増大による輻輳を回避するには集約間隔を長くすれば良いが、リアルタイム性は落ちる。また、最大サイズ S を減らすと集約パケットに含まれる最大パケット数が少なくなり、時間あたりに到着するパケットの量が減るため、サーバの負荷が下がる。一方で、送信が先送りになるパケットが増え、リアルタイム性が下がる。

パケットロス率は特殊であり、冗長送信回数を増やすとパケットロス率が上がる場合と、下がる場合の両方がある。パケットロスの原因にはルータ等のネットワーク上の機器の性能限界と、通信路の品質自体の問題がある。前者が原因の場合には冗長送信回数を増やすとさらにネットワーク機器の負荷が増えてパケットロス率が上がる。一方、後者が原因の場合にはパケットの到達可能性が上がるため、パケットロス率が下がると考えられる。

以上のように、輻輳の発生要因に応じて適切にパラメータ制御を行う必要がある。

3.2. 輻輳制御手法の詳細

3.2.1. 基本方針

提案方式では、集約パケットの最大サイズ S 、集約間隔 I 、冗長送信回数 N を制御して輻輳を回避するが、 S 、 I 、 N を制御する際の前提として、時間あたりの通信量がゲートウェイとサーバ間のバンド幅 B を超えない必要がある。時間あたりのパケット送信回数は I の逆数となるため、時間あたりの最大通信量 M は、 S 、 I 、 N を用いて、

$$M = N \times S \times I^{-1} \quad \dots(1)$$

で定義できる。さらに M は、

$$M \leq B \quad \dots(2)$$

を満たす必要がある。(2)式を満たすようにS, I, Nの制御を行う。

3.2.2. バンド幅の予測方法

(2)式を満たすかどうかを確認するには、バンド幅Bを予測する必要がある。Y.Guらは送信側からペアとなる複数のパケットを同時に送信し、受信側でパケットを受け取った時間の時間差からバンド幅を予測する方式を提案している^[5]。パケットが帯域幅の異なる複数の回線を通過する場合にも、通信路上で最も帯域の細い回線のバンド幅を予測できる。帯域の細い回線により、先着パケットと後着パケットの到着時間の差が大きくなるためである。

リアルタイム通信集約プロトコルにおいても冗長送信用パケットが送信側からほぼ同時に送信される特徴を利用し、冗長送信パケット群をペアのパケットとみなし、バンド幅Bを予測する。

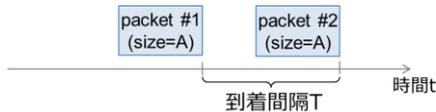


図2 冗長送信パケットの到着間隔の例

例えば、図2のように受信側でサイズAの冗長化された2個のパケットが、Tの時間差で届いたとする。このときバンド幅Bは、サイズAのパケット送信に時間Tを要したと予測されるため、

$$B = A / T \quad \dots(3)$$

でバンド幅を予測する。

3.2.3. 各パラメータの制御方法

輻輳発生要因としては、サーバ負荷によるもの、ネットワーク負荷によるものが考えられる。3.1節で述べたようにサーバ負荷の低減にはIを増加させることが有効であり、ネットワーク負荷の低減にはNを減らすことが有用である。最大サイズSの変更も負荷低減に有効だが、Sを小さくするとパケット送信が次の集約周期に遅延する可能性がある。これはリアルタイム性に与える影響が大きく、まずIとNを制御する。また、サーバ負荷とネットワーク負荷の輻輳の兆候を検出して、パラメータ制御を行う必要があるが、前者はサーバの受信バッファ溢れ回数、後者はパケットロス率により検出することとした。以上より、制御の基本方針は次のようになる。

- (I) サーバ負荷の輻輳（受信バッファ溢れ）が発生した場合にはIを増やす
- (II) ネットワーク負荷の輻輳（パケットロス）が発生した場合にはNを減らす

サーバ負荷の低減にはSの制御も有効であるため、(I)を行っても輻輳が十分に回避できない場合に限り、Sを減らすこととする。

また、パケットロスの発生要因がネットワーク負荷の輻輳でなく、通信路の品質が悪いことによるものだった場合、(II)でNを減らしたことによりパケットロスがさらに増える可能性がある。そこで、パケットロスが発生した場合には、まずNを増やしてみてもパケットロスが増えるようならばNを減らすポリシーとした。輻輳回避からの回復も含め、S, I, Nの制御方針をまとめると表2のようになる。

表2 各パラメータの制御方法

	受信バッファ溢れ		パケットロス率	
	閾値A以上	閾値B以下	閾値C以上	閾値D以下
最大サイズS	②減らす (I=最大値のとき)	①増やす (S>初期値もしくはI=最小値のとき。但しM≤B)	—	—
集約間隔I	①増やす (I<最大値のとき)	②減らす (I<最大値のとき)	—	—
冗長送信回数N	—	—	最初増やし、ロス率が上がれば減らす	元に戻す

表2中の番号は変化の順序を示し、例えばサーバ負荷が閾値A以上の場合、まず集約間隔Iを最大値まで増やし、それでも受信バッファ溢れ回数が閾値A以上の場合には最大サイズSを減らす。

例えば、(a)S=1KB, I=1ms, N=2, (b)S=2KB, I=2ms, N=2, (c)S=4KB, I=2ms, N=1の各パラメータセットで最大通信量Mは、(1)式より2MB/sで同じである。一方、サーバやネットワーク機器にかかる負荷は(a)>(b)>(c)となる。(a)での輻輳発生時には、(b)や(c)のパラメータセットを採用することで、最大通信量を減らさずに輻輳を回避できる。

4. 実装

3章の方式の実現のため、ゲートウェイとサーバ上で動作させる通信集約用ソフトに次の機能を実装した。

(A) バンド幅予測機能

冗長送信されたパケットの到着間隔とサイズの情報からバンド幅を予測する機能。

(B) 受信バッファ溢れ回数計測機能

Linuxのprocファイルシステムより、NICの受信バッファ溢れ回数を取得する機能。

(C) パケットロス回数計測機能

各集約パケットに付与されたシーケンス番号を利用してパケットロス回数を計測する機能。

(D) パラメータ調整機能

3章の方式に従い、(B)と(C)の情報から最大サイズ S 、送信間隔 I 、冗長送信回数 N の値を決定する機能。 I と N については表3のように取りうる値を予め配列で持たせて増加減する。一方で S については表3のように決められた範囲内で AIMD^[6]によって増減する。

表3 パラメータの取りうる値
(下線は輻輳制御無の際のデフォルト値)

集約周期 I	500us, 750us, <u>1ms</u> , 2ms, 4ms
冗長送信回数 N	1, <u>2</u> , 3
最大サイズ S	512byte~8192byte (<u>2048byte</u>)

5. 評価と考察

クライアントPCからサーバに対して毎秒計400件のトランザクションを各秒内でランダム間隔で発生させ、トランザクション開始から終了までの所要時間を計測した。輻輳の再現のため、サーバに対して iperf プログラムで通信負荷をかけられるようにした。その他の評価環境を表4に示し、通信負荷の有無別の測定結果を表5に示す。比較のため、輻輳制御を行わずゲートウェイによる集約を行う通信と、ゲートウェイによる集約を行わないTCP/IPでの通信についても計測した。

表4 評価環境

物理クライアントPC数	2台
物理クライアント1台あたりの 模擬クライアント(プロセス)数	32個
ゲートウェイ台数	2台
トランザクションあたりの通信回数	12回
クライアント-ゲートウェイ間の遅延(往復)	なし
ゲートウェイ-サーバ間の遅延(往復)	1ms
通信路上でのパケットロス率(片道)	1%

表5 通信時間評価結果(ms)

		輻輳制御無	輻輳制御有	TCP/IP
通信 負荷 無	最小	16.2	16.0	7.37
	最大	23.3	22.4	87.3
	平均	19.1	19.0	33.1
通信 負荷 有	最小	16.5	12.7	7.21
	最大	56.5	20.6	618
	平均	19.4	15.2	29.5

輻輳制御無に比べ、輻輳制御有では最小/最大/平均通信時間ともに改善した。特に輻輳制御がない場合には通信負荷の有無で最大通信時間が 23.3ms から 56.5ms まで悪化するが、輻輳制御有では負荷有無の通信時間への影響が少ない。

また TCP/IP では再送時の遅延が大きく、最大の通信時間が 618ms となったが、本方式では最大通信時間が 20.6ms と一桁以上小さい。本方式はリアルタイム性が重視される制御システムへの応用に有用であると言える。

6. まとめ

本稿では通信集約プロトコル向けの輻輳制御手法の設計と実装、評価を行った。一般的な輻輳制御では時間あたりに通信可能なデータ量を増減させるが、本方式では通信集約プロトコルに特化し、通信可能な実データ量を減らさずに輻輳を回避できる。実際に評価を行い、輻輳制御無の場合や TCP/IP と比較してもネットワークの混雑時に通信遅延の発生が抑えられ、優位性があることを確認した。今後の課題として、さらなる評価の追加などが挙げられる。

参考文献

- [1] L. D. Xu et al.: "Internet of Things in Industries: A Survey", IEEE Trans. on Industrial Informatics, Vol. 10, No. 4, pp. 2233-2243 (2014)
- [2] 大西ら: "クラウドサーバ向け通信集約プロトコルの開発", 電気学会通信研究会, CMN-16-032 (2016)
- [3] 松山ら: "クラウドサーバ向けリアルタイム通信集約プロトコルの評価", 電気学会全国大会, Vol. 3, pp.11-12(2017)
- [4] M.Kazmi et al.: "Comparison of TCP Tahoe, Reno, New Reno, Sack and Vegas in IP and MPLS Networks under Constant Bit Rate Traffic", ICACTCM 2014, pp.33-38(2014)
- [5] Y.Gu et al.: "UDT: UDP-based Data Transfer for High-Speed Wide Area Networks", Computer Networks (Elsevier). Vol.51-7 (2007)
- [6] Chiu et al.: "Analysis of increase and decrease algorithms for congestion avoidance in computer networks", Computer Networks and ISDN systems, Vol.17 pp.1-14(1989).