4ZB-07

Network Calculus による最悪遅延解析を用いた遅延ボトルネック検出法

松田 脩佑 † 廣森 聡仁 †† 山口 弘純 †† 陶山 洋次郎 ‡ 泉 達也 ‡ 浦山 博史 ‡‡
小林 史歩 ‡‡‡ 梅原 茂樹 ‡‡‡ 谷 英哲 ‡‡‡

† 大阪大学 基礎工学部情報科学科 †† 大阪大学 大学院情報科学研究科 ‡ 住友電気工業株式会社
‡‡ 株式会社オートネットワーク技術研究所 ‡‡‡ 住友電工システムソリューション株式会社

1 はじめに

車両や航空機などにおいても，コンピュータネットワークは欠かせないものとなっており，機器の制御に関わる情報を扱うだけでなく，音楽や動画の配信など，様々なアプリケーションが稼働することが想定される．一方で，ネットワークの資源は限られており，個々のアプリケーションから生じるメッセージは互いに競合するため，個々のアプリケーションで想定するネットワーク性能が保たれるか不明確であり，ネットワーク全体としての性能を適切に把握することが求められる．ネットワーク性能の重要な要素の一つである最悪の通信遅延を評価する手法の一つとして，Network Calculus [1] が挙げられる．Network Calculus は，スイッチを介して接続されるある二点間に生じる通信遅延を評価する手法であり，一方のノード上のアプリケーションの送信トラフィックを Arrival Curve として，また，スイッチが有する転送能力を Service Curve として数式で表現し，これらの組み合わせを計算することで，他方のノードが受信するまでの最悪の通信遅延を数学的に導出する．このように，ネットワークの様々な挙動を数式で表現するため，必ずしも正確に通信遅延を表現できるものではないが，ネットワークシミュレーションと比較し，少ない計算時間や計算資源で導出できるため，様々な状況におけるネットワーク性能を評価することができる．本取組では，車載ネットワークを対象に，与えられた QoS 制御とアプリケーションにおける最悪遅延を Network Calculus によりモデル化し，ネットワークのボトルネックを効率的に発見する方法を考案する．

2 通信遅延のモデル化

本取組では，QoS 制御の一手法である Strict Priority 及び Credit-Based Shaper の規格を利用した際，ある二点間の通信におけるパケット単位の最悪遅延を Network Calculus によりモデル化する．このモデルを利用し，QoS 制御により生じる最悪遅延を効率的に導出し，ネットワーク全体におけるボトルネックを明らかにする．

まず，データ送信元の送信パターンが周期的であると仮定することで，メッセージサイズ m，送信周期 CMI，伝送レート C とすると，トラフィックの入力を表す Arrival Curve は，以下の式にて表現することができる [2].

$$\alpha(t) = \min\{Ct, b + rt\} \tag{1}$$

$$r = \frac{m}{CMI}, b = m(1 - \frac{r}{C})$$

与えられた優先度順にパケットを転送する Strict Priority に対する Service Curve は，優先度の高いクラスのパケットがノードに到着し続ける限り，優先度の低いクラスのパケットは送出を待ち続けるため，その挙動は下記の数式で表現することができる [1].

$$\beta_H(t) = [Ct - l^L_{MAX}]^+ \tag{2}$$

$$\beta_L(t) = [Ct - \alpha_H(t)]^+ \tag{3}$$

上の二式はそれぞれ高優先，低優先の Service Curve を表しており，l^L_{MAX} は低優先の最大パケット長, α_H は高優先の Arrival Curve, C は伝送レートを示している．

優先度毎に定められたパケット送信頻度によりパケット送信割合を抑える Credit-Based Shaper (CBS) において，ある CBS クラスの Service Curve は，以下の式で表現することができる [2].

$$\beta_{CBS}(t) = idSl[Ct - \frac{c^{MAX}_{M_i}}{idSl}]^+ \tag{4}$$

$$c^{MAX}_{M_i} = \frac{max\{l^i_{MAX}\}}{C}\sum_{j=1}^{i} idSl_i - \sum_{j=1}^{i-1} c^{Min}_{M_i} \tag{5}$$

$$c^{Min}_{M_i} = \frac{l^i_{MAX}}{C} sdSl_i \tag{6}$$

クラス i における最大パケット長 l^i_{MAX}．クレジットの最大値，最小値 $c^{MAX}_{M_i}, c^{Min}_{M_i}$ をそれそれ示す．但し，伝送レート C とシェーピングの割合 $isSl$ は，$sdSl = C - idSl$ とする．

これらの単一ノードに対する数式を組み合わせることで，各スイッチの Service Curve を Min-Plus 代数系における Convolution の組み合わせ ($\beta = \beta_1 \otimes \beta_2 \otimes \beta_3 \cdots$) により，多段のスイッチを介した最悪遅延として導出する [3]．また，同じ出力キューを利用する別のトラフィックが存在する場合，各スイッチにおいて，その別トラフィックの Arrival Curve を考慮する必要が

Shusuke Matsuda†, Akihito Hiromori††, Hirozumi Yamaguchi††, Yojiro Suyama Suyama‡, Tatsuya Izumi‡, Hirofumi Urayama‡‡, Shiho Kobayashi ‡‡‡, Shigeki Umehara‡‡‡, Hideaki Tani‡‡‡
†School of Engineering Science, Osaka Univ., Osaka Japan
††Graduate School of Information Science and Technology, Osaka Univ., Osaka Japan
‡Sumitomo Electric Industries, Ltd.
‡‡AutoNetworks Technologies, Ltd.
‡‡‡Sumitomo Electric System Solutions Co., Ltd.

Copyright ©2022 Information Processing Society of Japan.
All Rights Reserved.

ある．各スイッチにおける Arrival Curve は，前段の Arrival Curve と Service Curve の Min-Plus 代数系における Deconvolution ($\beta = \beta_1 \otimes [\beta_2 - \alpha']^+ \otimes [\beta_3 - (\alpha' \oslash \beta_2)] \cdots$) とする [4].

3 最悪遅延の導出

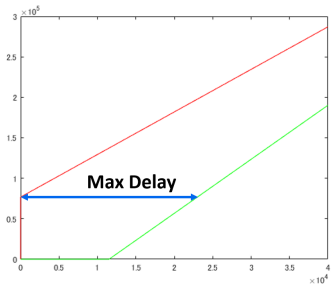

図 1 Arrival Curve と Service Curve

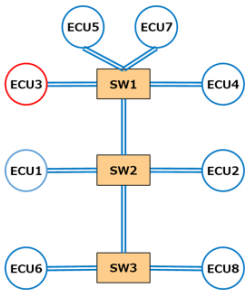

図 2 対象とするネットワークトポロジ

Arrival Curve と Service Curve は，トラフィックの入力と，それに対するスイッチの処理能力を時間当たりの累積データ量で表しており，図 1 に示すように，赤色の Arrival Curve と 緑色の Service Curve の水平最大距離を最悪の遅延時間と捉えることができる．また，これらの数式を，ある二点間における経路上の各スイッチに適用し，それらを組み合わせることで，その経路全体の最悪遅延を把握することができる．

本取組で，対象とするネットワークはパケットの送受信を行う 8 つの ECU 及びパケットを転送する 3 つのスイッチ (SW1〜3) から構成されており，図 2 に示すようなネットワークトポロジとなっている．また，各 ECU からは表 1 のようなメッセージが送信され，各スイッチにおいては，優先度の高いメッセージが優先されるものとする．この状況において，各スイッチにおける QoS 制御として，Strict Priority を用いている際には，メッセージ F に関してパケットロスが発生する．以下，Network Calculus を活用することで，このパケットロスに関わる要因を特定し，改善を図る．

まず，シミュレータと Network Calculus により導出した各メッセージの最悪遅延を表 2 に示す．その際，Scenargie によりシミュレーションを実行し，MATLAB 上で稼働する RTCtoolbox [5] を利用し，Network Calculus を実行している．Network Calculus による結果を NC，シミュレータによる結果を Sim として記載している．この表からわかるように，Network Calculus による最悪遅延は，シミュレータよるものより大きくなっており，各メッセージの最悪遅延を導出できている．同様に，Network Calculus により導出した，各スイッチにおけるメッセージの遅延を表 3 に示す．空欄となっている箇所は，そのメッセージが該当のスイッチを通過していないことを示す．この結果から，他のメッセージと比較し，優先度 2 のメッセージ E と F が SW1 を通過する際の遅延が特に大きいことがわかることから，ここがボトルネックであることが推測される．そこで，SW1 において，メッセージ F より高優先のトラフィックに対し，CBS を適応することで，より細やかにパケット転送を制御したところ，パケットロスを解消できることがわかった．

これらの評価に際し，表 1 に示す経路全体の最悪遅延の導出に要する計算時間は，Network Calculus では 15 [ms] 以下であったのに対し，シミュレータは約 50 [ms] であった．また，表 3 に示す個々のスイッチの最悪遅延の導出する際の全体の計算時間は，20[ms] であった．Network Calculus は，算出する最悪遅延の対象毎に計算が必要となるため，メッセージ毎の計算となる経路全体の最悪遅延の算出より，メッセージとスイッチの組み合わせ毎に対する計算となるスイッチの最悪遅延の算出の場合が，対象数が多くなるため，計算時間が長くなる．しかしながら，個々の計算時間は短いため，対象数が多い場合でも全体の計算時間は，シミュレータよりも短くなる場合が多く，効率的な評価が期待される．

表 1 ECU から送信されるメッセージ

メッセージ名	送信元→宛先	優先度	サイズ [Byte]	送信周期 [ms]
A	2 → 3	6	50000	350
B	8 → 3	4	50000	300
C	4 → 3	4	50000	450
D	1 → 3	3	100000	80
E	5 → 3	2	100000	100
F	6 → 3	2	100000	100

表 2 各メッセージにおける最悪遅延 [μs]

	A	B	C	D	E	F
NC	610	12905	8632	21336	39484	59055
Sim	615	8745	4559	12940	21197	29578

表 3 各スイッチの最悪遅延 [μs]

	A	B	C	D	E	F
SW1	236	12289	8382	20760	39234	46420
SW2	118	4164	-	8427	-	22650
SW3	-	118	-	-	-	4227

4 おわりに

本取組では，Network Calculus により，比較的少ない計算時間や資源で大まかな最悪遅延を導出することで，ネットワークにおけるボトルネックを検出した．今後は，Network Calculus の少ない計算時間を生かし，組み合わせテスト手法を活用することにより，状況毎で最適な QoS 設定を導出する手法について検討したい．

参考文献

[1] Le Boudec, J.-Y. and Thiran, P.: *Network calculus: a theory of deterministic queuing systems for the internet*, Vol. 2050, Springer Science & Business Media (2001).

[2] Daigmorte, H., Boyer, M. and Zhao, L.: Modelling in network calculus a TSN architecture mixing Time-Triggered, Credit Based Shaper and Best-Effort queues (2018).

[3] Zhou, B., Howenstine, I., Limprapaipong, S. and Cheng, L.: A Survey on Network Calculus Tools for Network Infrastructure in Real-Time Systems, *IEEE Access*, Vol. 8, pp. 223588–223605 (2020).

[4] Bondorf, S.: Worst-Case Performance Analysis of Feed-Forward Networks–An Efficient and Accurate Network Calculus (2016).

[5] Wandeler, B. and Thiele, L.: Real-Time Calculus (RTC) Toolbox. Accessed: Oct.2021. [Online]. Available: http://www.mpa.ethz. ch/Rtctoolbox (2006).

Copyright ©2022 Information Processing Society of Japan.
All Rights Reserved.