

## シード値を変化させるハッシュ関数を用いたリンクアグリゲーションの負荷分散法の検討

## A study on load balancing method for link aggregation using a hash function with variable seed

○大谷悠晴<sup>1</sup>, 澤邊知子<sup>2</sup>\*Yusei Ohtani<sup>1</sup>, Tomoko Sawabe<sup>2</sup>

**Abstract:** In this paper, we propose a load balancing method for link aggregation. Conventional load balancing methods use hash functions. In the proposed method, a seed value is added to the input of the hash function and this seed value is varied to improve the performance of load balancing.

## 1. まえがき

近年、動画配信やIoT機器などの普及によりネットワークトラフィックが増大している。その結果、冗長性と帯域拡張が同時に確保できるリンクアグリゲーション<sup>[1]</sup>の需要が高まっている。リンクアグリゲーションとは、スイッチングハブなどのネットワーク装置を複数のリンクで接続し、それを束ねて仮想的な1本のリンクとして通信する技術である。各リンクの通信性能を効率よく利用するためには、通信量を各リンクに均等の分散する負荷分散法が不可欠である。本論文では、リンクアグリゲーションの負荷分散で用いるハッシュ関数のシード値を変化させることにより負荷分散の性能を向上させる方式を検討する。

## 2. 従来の負荷分散方式

リンクアグリゲーションの負荷分散方式の1つとして、ハッシュ関数を用いる方式がある<sup>[2]</sup>。この方式を図1に示す。まずは、リンクアグリゲーションで束ねたリンク数をNとする。ハッシュ関数の入力として、通信データの送信元IPアドレス、宛先IPアドレス、送信元ポート番号、宛先ポート番号、プロトコル番号(5-tupleと呼ぶ)などを使用する。求められたハッシュ値のNの剰余数によって通信するリンクを決定する。

## 3. 提案方式

2章で述べた負荷分散方式では、ハッシュ関数の入力である5-tupleにより選択されたリンクが偏る場合もある。そこで、ハッシュ関数の入力にシード値を加え、そのシード値を変化させることによって、均等に負荷分散させる方式を提案する。

提案方式では、通信を一定間隔の時間フレームに分割する。最初の時間フレームでは、シード値を0に設定し負荷分散を行う。また、平行して、複数のシード値

を用いた場合の負荷分散をそれぞれ評価し、一番均等であるシード値を次の時間フレームで採用し、負荷分散を行う。これを繰り返す。負荷分散の評価指数としては、以下の公平性(JFI)を用いる。

$$JFI = \frac{(\sum_{i=1}^N x_i)^2}{N \sum_{i=1}^N x_i^2}$$

ただし、Nはリンク数、 $x_i$ は各リンクの単位時間当たりの通信量である。

## 4. まとめ

本論文では、リンクアグリゲーションの負荷分散方式の提案を行った。負荷分散の方式として、従来からハッシュ関数を用いる方式がある。提案方式では、負荷分散の性能を向上されるために、ハッシュ関数の入力にシード値を追加し、このシード値を変化させる。今後は、提案方式の性能を評価する予定である。

## 参考文献

- [1] IEEE 802.1AX-2008, "IEEE Standard for local and metropolitan area networks--Link Aggregation"  
 [2] Vishal Deep Ajmera, Nitin Katiyar, Pradeep Ventkatesan, and Anju Thomas, "Balance-TCP Bond Mode Performance Improvement",  
<https://www.openvswitch.org/support/ovscon2019/day2/0944-Balance-TCP%20Performance%20Improvement.pdf>,  
 (2025-10-01).

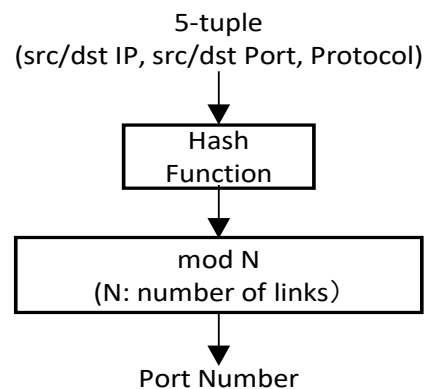


Fig.1 A conventional load balancing method use hash.