

ポリシング

学習内容

1 ポリシングとシェーピングの概要と目的

2 両者の違いとグラフイメージ

3 トラフィック制御の根幹「トーカンバケット」

4 ポリシングの具体的な実装とアルゴリズム

01

1. ポリシングとシェーピングの概要

トラフィック制御の二大柱

速度制御を行うための「ポリシング」と「シェーピング」

ポリシング (Policing)

超過パケットを即座に破棄、または優先度を書き換える。破棄による帯域制限が主目的。

シェーピング (Shaping)

超過パケットを一旦バッファに格納し、後で送信。トラフィックの平滑化が主目的。

実装コマンド

ポリシングは`police`または`rate-limit`、シェーピングは`shape`で実装されることが一般的。

ポリシングとシェーピングの決定的な違い

超過トラフィックの扱いに伴う動作と適用場所の対比

ポリシング

超過パケットを即座に破棄（または優先度変更）

遅延が発生しない

トラフィックはノコギリ状になる

着信I/Fにも発信I/Fにも適用可能

メモリ消費は少ない

シェーピング

超過パケットをバッファに格納し後で送信

バッファ遅延が発生する

トラフィックは平滑化される

発信I/Fにのみ適用可能

メモリ消費は比較的多い

通信グラフのイメージ

制限値超過時のトラフィックの波形

ポリシング: 超過分を即時破棄するため、レートグラフは**ギザギザのノコギリ状**になる

シェーピング: バッファで平滑化するため、レートグラフは**平滑化された波状**になる

視覚的な違いを理解することで、それぞれの動作のイメージを掴める

02

2. トークンバケットアルゴリズム

トークンバケットの基本構造

パケット送信の「権利」を管理する仕組み



トークンバケットを構成する3つの要素

平均レート、バーストサイズ、時間間隔の関係

Mean Rate (平均レート)

CIR (bps)

補充速度

Burst Size (バーストサイズ)

Bc (bit)

補充量

Time Interval (時間間隔)

Tc (sec)

補充周期

03

3. ポリシングのより詳細な理解

ポリシングの二大実装方式

Class-Based PolicingとCARの違い

Class-Based Policing

MQCでClassに適用

police

conform / exceed / violation
の3段階

シングル/デュアルトークン

設定方法

コマンド

処理段階

アルゴリズム

CAR

インターフェースに直接設定

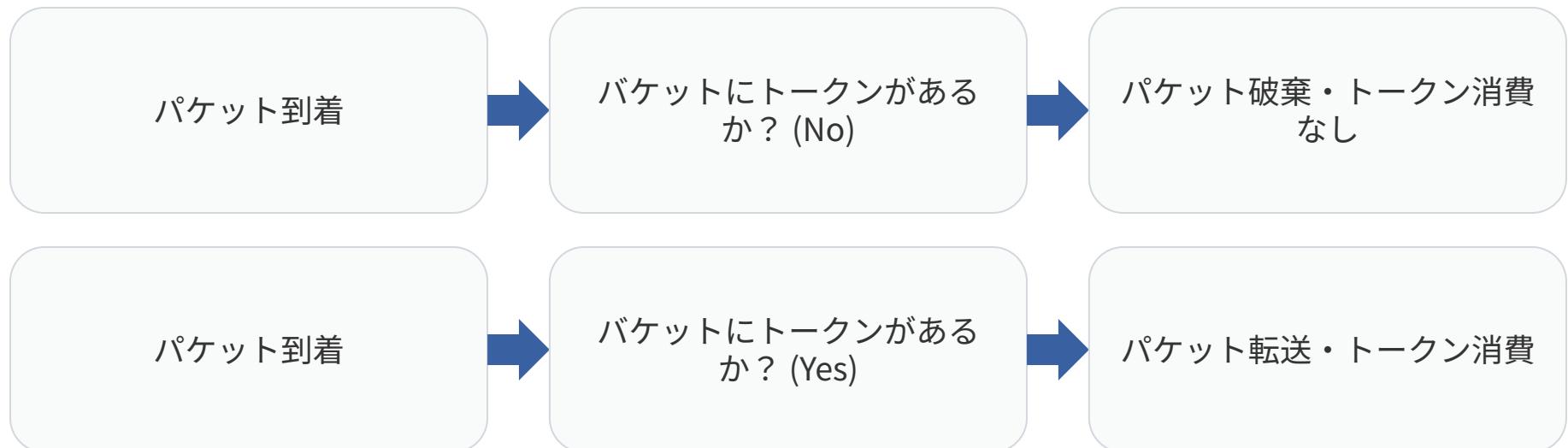
rate-limit

conform / exceed の2段階

シングルトークン

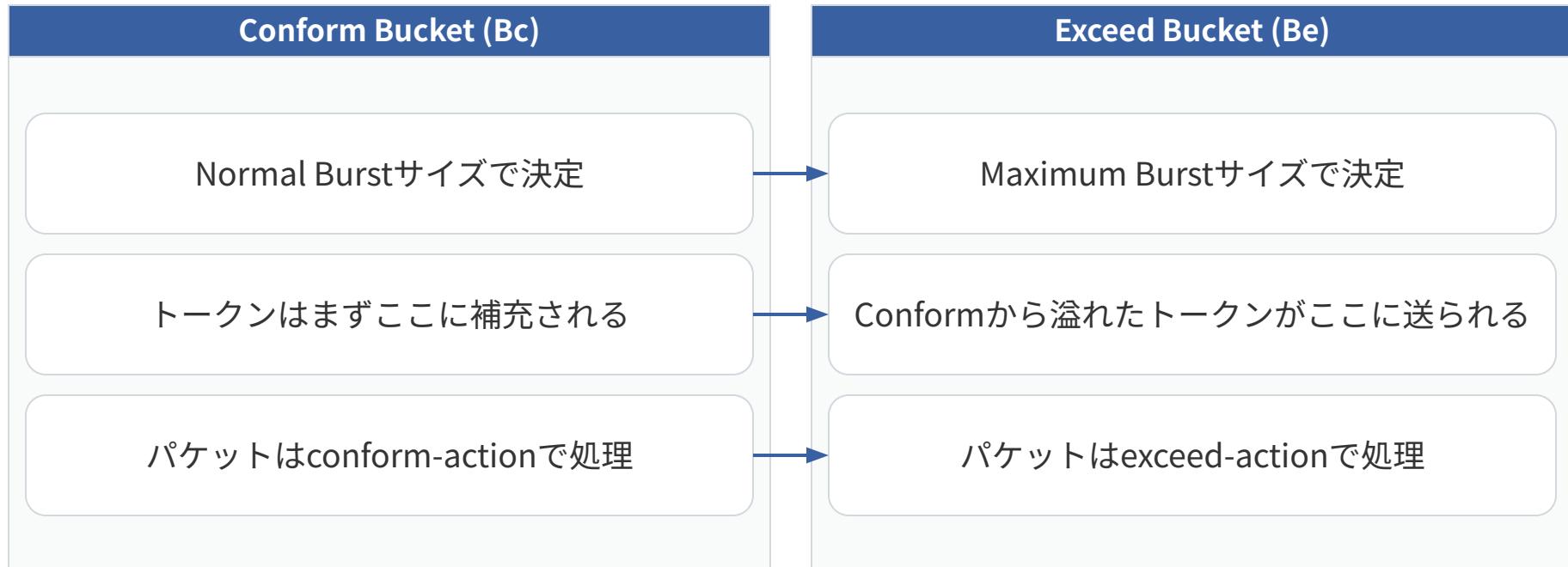
シングルトークンバケットの動作 (violate-actionなし)

トークン不足時はパケットを**破棄**する動作



デュアルトークンバケットの構成 (violate-actionあり)

コンフォームとエクシードの2つのバケットを持つ



Class-Based Policingの設定構文

シングルレート・デュアルトークン設定の例

MQC (Modular QoS CLI) の`policy-map`内で`police`コマンドを設定

`police bps burst-normal burst-max conform-action action exceed-action action violation-action
action`

Bc値の算出: $Bc = CIR \times 1/8 \times 1.5$

Be値の算出: $Be = Bc \times 2$

BcとBeはバースト許容量を決定する重要なパラメータ