

# マシンラーニングから学ぶQoS技法

Shishio Tsuchiya

[shtsuchi@arista.com](mailto:shtsuchi@arista.com)

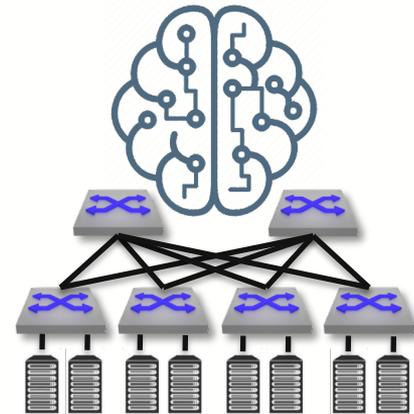
# 自己紹介

- 土屋師子生
  - アリスタネットワークスジャパン合同会社/副技師長
- 経歴
  - 東京勤務/横浜在住
  - 1997-2000 リコーテクノシステム サポートエンジニア
  - 2000-2017 シスコシステムズ コンサルティングシステムズエンジニアでサービスプロバイダーや大規模エンタープライズの案件をサポート
  - 2017- アリスタネットワークスでデータセンターやサービスプロバイダーの案件をサポート



# 応募のきっかけ

- AI/MLのネットワークデザインと今後がとっても楽しい(自分にとって)
- でも…



# APRICOT2024に挑戦

## APRICOT 2024

APNIC

PROGRAM COMMITTEE

HOME REGISTER PROGRAM TRAVEL HEALTH AND SAFETY APNIC REPORT SERVICE



Thank you to all the attendees, sponsors, and everyone else who contribute to APRICOT 2024.

[View Program](#)

### PC Chairs

- Mark Duffell - Telstra
- Mark Tinka - SAFNOG
- Vincent "Achie" Atienza - Globe

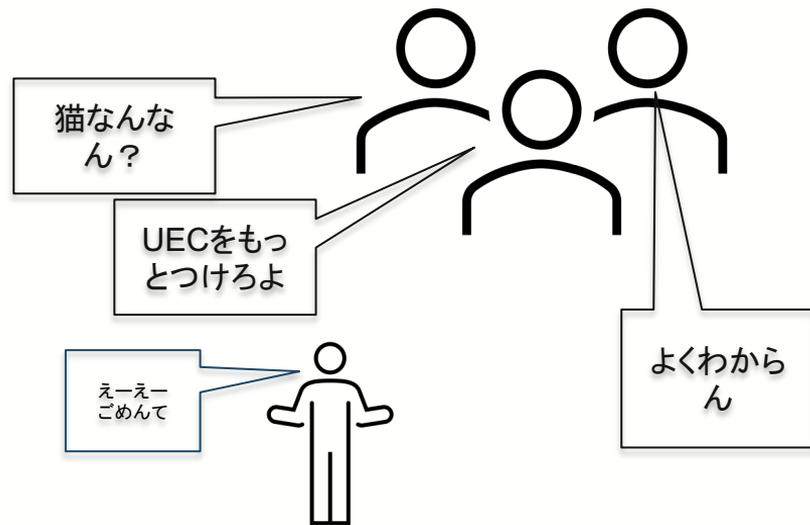
### Committee

- Amante Alvaran - Google
- Dave Phelan - APNIC
- Dhruv Dhody - Huawei
- Fakrul Alam - Cloudflare
- Fraser McGlinn - RISE
- GZ Kabir - BDCOM
- Jose Dante Santiago - University of Guam/MARIIX
- Kabindra Shrestha - PCH
- Kevin Meynell
- Mawatari Masataka - JPIX
- Ooi Cong Jen - MyNOG
- Paresh Khatri - Nokia
- Patrick Okui - Workonline
- Philip Paeps - Alternative Enterprises
- René Fichtmüller - FlexOptix
- Shaila Sharmin - Prime Bank Ltd.
- Sheryl Hерmoso - APNIC
- Shishio Tsuchiya - Arista
- Srimal Rajapaksha - Vuba Technologies Ltd
- Tashi Phuntsho - btNOG
- Ulsbold Enkhtaivan - Mobicom

- 2/21-3/1タイ・バンコクで開催
- プログラム委員としても参加
- プレゼンテーション資料必須
- PCによる採点によりFIFO
- - 5点満点で大体平均4.0がボーダライン
- 都度採点週一でレビュー
- JANOGと同じ資料で応募！！

# 採点一発目

1.8



- 地獄のようなプログラムミーティング…

# JANOG54では

トップページ 参加登録 アンケート NETCON インターネット体験会 野良BoF newsletter プログラム一覧 プログラム紹介 行動規範 (Code of conduct)

若者支援 JANOG Slack 現地情報 ホスト 協賛 協賛のご案内 スタッフリスト

## AI(人工知能)の為のネットワーク

### 概要

AI(人工知能)を学習をさせる為にはいくつかのパラメーターの入力と計算を繰り返す必要がある。GPT-3、Chinchilla、PALMのような大規模言語モデル(LLM)や、DLRMやDHENのようなディープラーニング推測システムはこの計算に関与するパラメーターを共有するいくつかのGPUからなるクラスタ上で構成される。交換されるデータ量は少量のフローで大きなデータがやり取りされる。ネットワークが貧弱であったり、輻輳による速度低下はAIアプリケーションのパフォーマンスに大きな影響を与える。100G/200G/400GのGPUを接続するAIアプリケーションではTCP incastに対処する仕組みとしてECN/PFCまたは輻輳を起こさない為の機器の選択やダイナミックなロードバランスが必要となる。

本セッションでは800G/1.6T/3.2Tの更なる広帯域をサポートする100万を超えるGPUを使う大規模AIネットワークでの要件を定める為立ち上がったUltra Ethernet Consortium (UEC)の情報提供と共にAIネットワークでの必要な要件を運用者の皆さんと議論できればと考えています。

### 場所

QTnet room(306)

### 日時

Day1 2024年1月17日(水) 17:00~18:00(1時間)

### 発表者



土屋 師子生  
アリスタネットワークスジャパン合同会社

### 公開資料

- うーん。反応いまいち

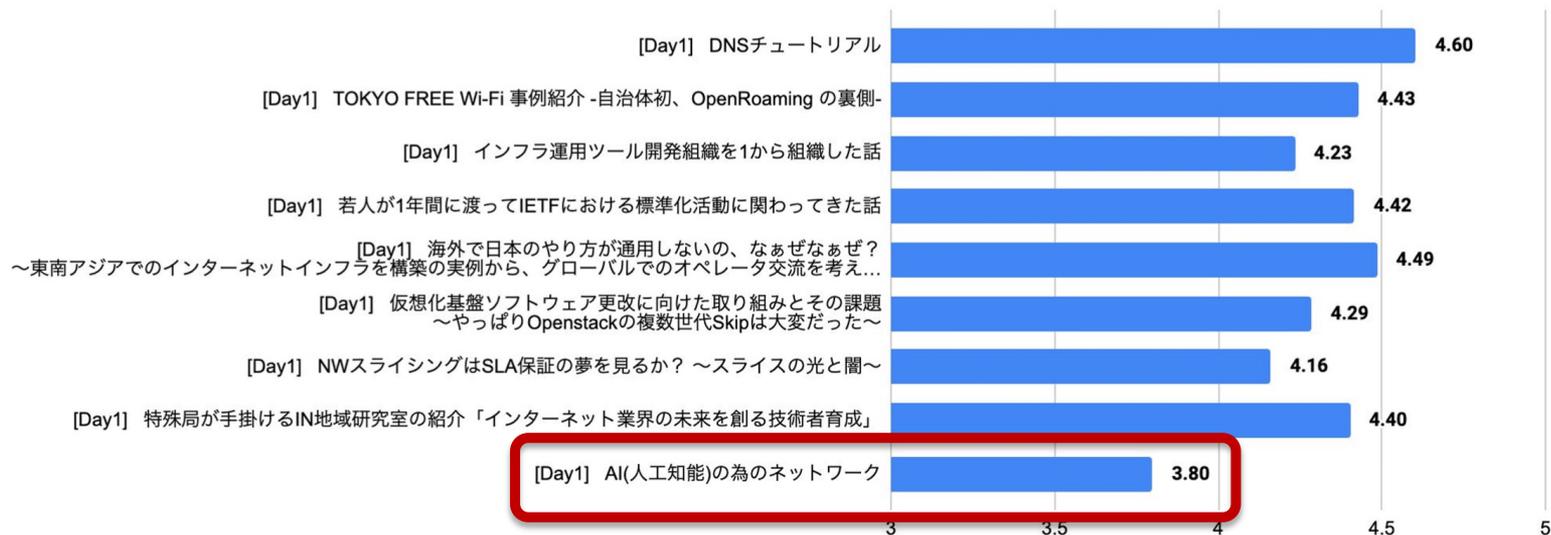
# アンケート結果



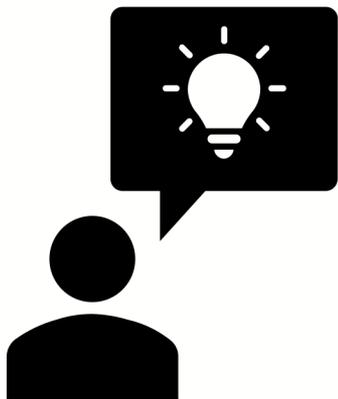
2024.1.17-19 IN HAKATA

通常プログラム平均点 [Day1]

(Day1 - Day3 全プログラム平均点 : 4.33)



# 考えた



- なぜ自分がおもしろいと思ってるのかを説明するプレゼンを作ってみよう！
- 過去から学べる事もあるはずだ

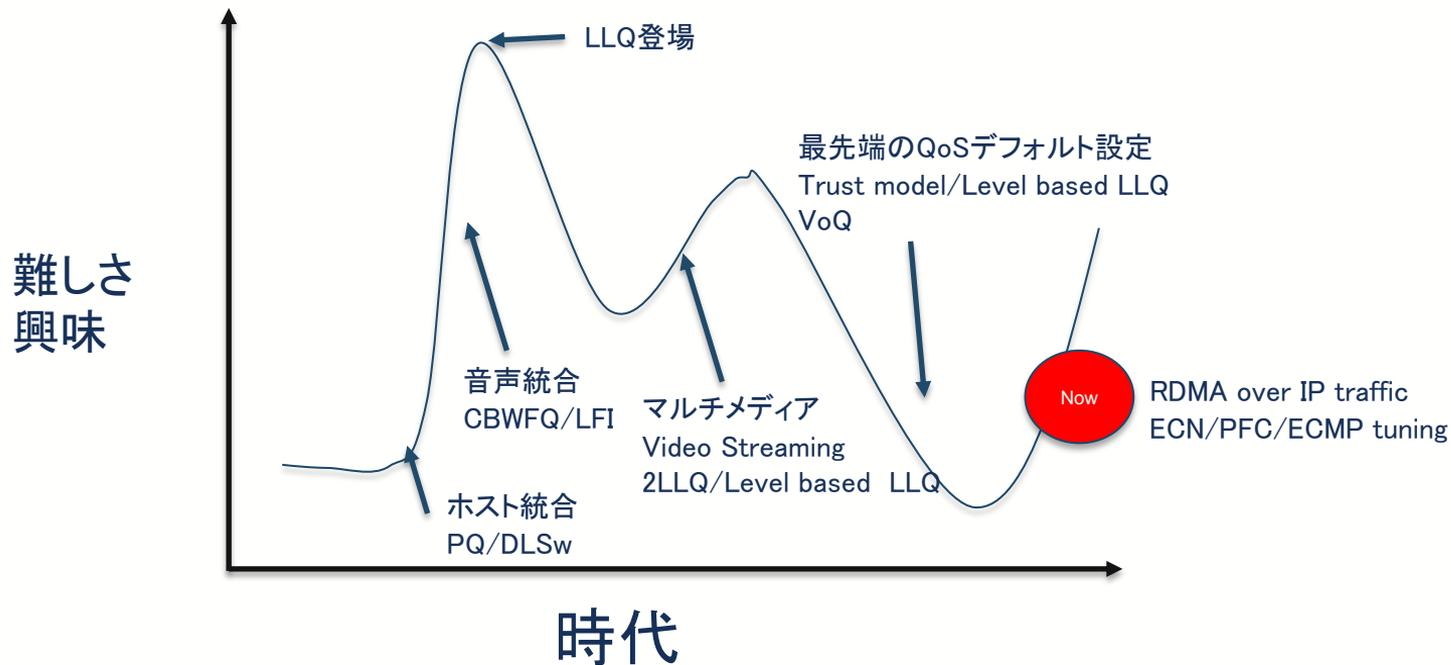
# 温故知新

# 結果

<p><b>Innovation 1</b> </p> <p> Crystal Hall B 11:30 to 13:00 (UTC +07:00)</p> <p><a href="#">Zoom</a> <a href="#">YouTube Live</a></p> <p><b>Chair</b> Mark Tinka</p> <p><b>Speakers</b> Dhruv Dhody - The IETF Roundup Thomas Weible - Coherent optical transce Shishio Tsuchiya - QoS/Traffic Manageme</p> <p><b>SESSION</b></p>	<p><b>AI network infrastructure design</b> </p> <p> Terrace A&amp;B 14:30 to 16:00 (UTC +07:00)</p> <p><a href="#">Zoom</a> <a href="#">YouTube Live</a></p> <p><b>Tutor</b> Shishio Tsuchiya</p> <p><b>TUTORIAL</b></p>
---	--

プログラムと  
チュートリアル  
2セッションゲッ  
ト

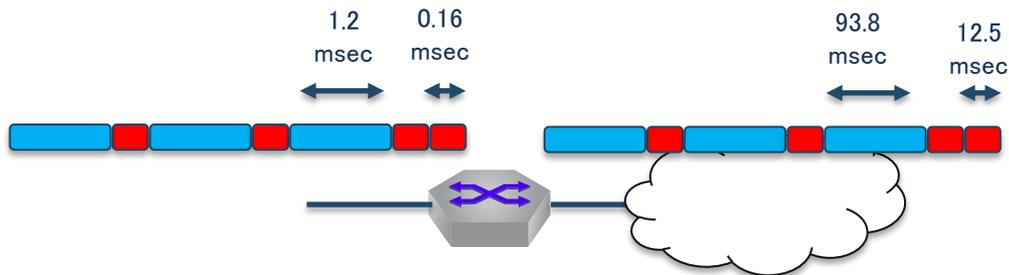
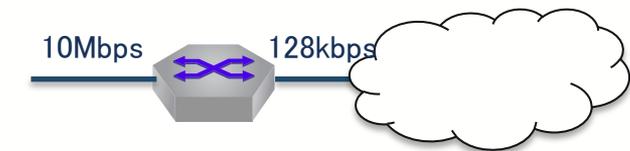
# QoSのライフサイクル



- ざっくりの自身の経験値
- なにか新しいものを統合するときに必要になる
- 今はRoCEv2でのマシーンラーニングトラフィックの最適化QoSを見つけたい時期

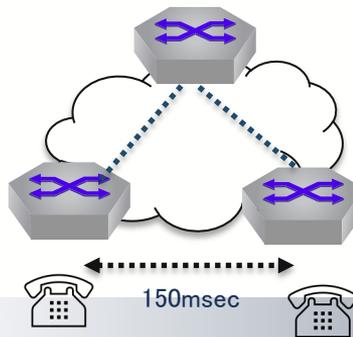
# なんでQoSが必要だったか 2000年以前

- 回線速度の違いが大きい
  - LAN 10Mbps/WAN 128kbps
- そのころのルータのデフォルトのQueuingアルゴリズムはFIFO(First In First Out/先入れ先出し).
  - 上の例だと1500バイトの packets はWAN回線だと93.8msecかかって、LANだと1.2msecかかる
  - 200バイトの packets はWAN回線だと12.5msecかかる、LANだと0.16msec
  - VoIPのエンドツーエンドの遅延の許容範囲は150msec



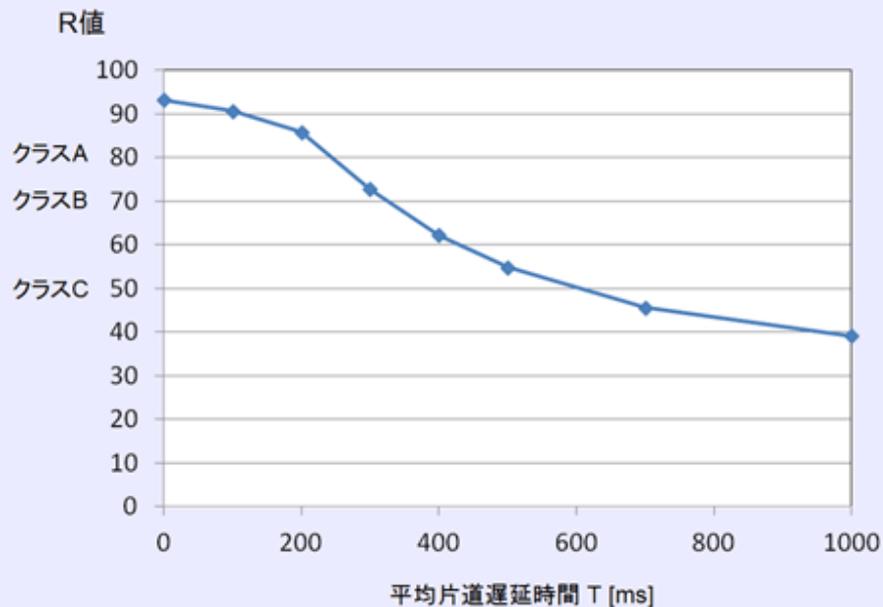
1500byte

200byte



# 音声品質評価特性

[https://www.rd.ntt/ns/qos/technology/sound/05\\_3.html](https://www.rd.ntt/ns/qos/technology/sound/05_3.html)

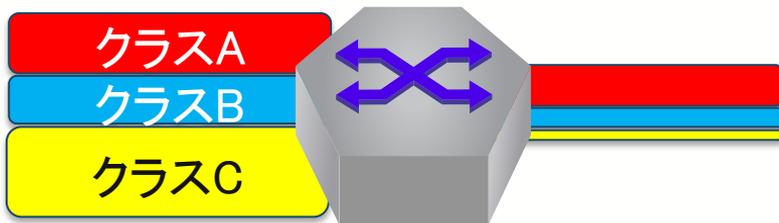


平均片道遅延時間とR値との関係

- クラスA:固定電話レベル
- クラスB:携帯電話レベル
- クラスC:それ以下

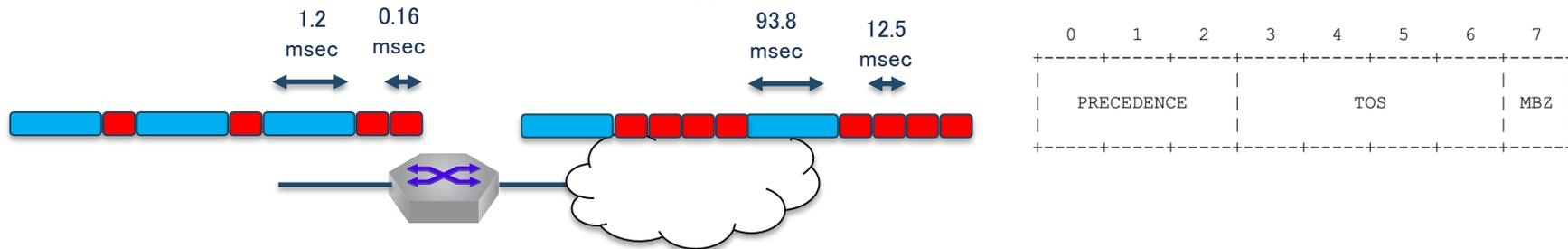
# CBWFQ(Class-Based Weighted Fair Queueing)

[https://www.cisco.com/en/US/docs/ios/12\\_0t/12\\_0t5/feature/guide/cbwfq.html#wp17641](https://www.cisco.com/en/US/docs/ios/12_0t/12_0t5/feature/guide/cbwfq.html#wp17641)



- アプリケーション毎にClassを定義
- それぞれのクラス毎にQueueをアサイン
- 帯域幅やWeightを定義し、最低帯域を保証する

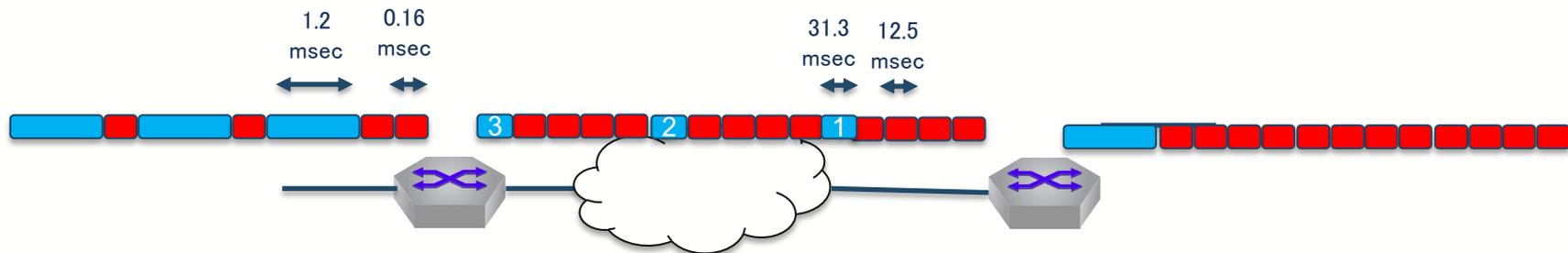
# 2000年以前のQueueing



- 入力インターフェースでACLでアプリケーションを識別、classを定義し、IPのTOSフィールドのPrecedence値をセットする
- ルータはPrecedence値を元にパケットを帯域に基づき送信する
- VoIPのRTPパケットは可能な限り早く送信する必要がある
- ただし、TCPアプリケーションはスループットを得るために帯域を必要とする
- 運用者は各々で検証して、VoIPに耐えうる適切なWeight値を見つけてた
  - NTTコミュニケーションのIP-VPNは90:9:1
    - <https://xtech.nikkei.com/it/article/COLUMN/20060930/249478/>
  - 自分の経験だと500,000,000:2:1のルータもあった
- WAN回線速度が低いと1つの大きなパケットが伝送遅延の為に大きな遅延を生む事になる

# 2000年以前のQueueing- Link Fragmentation Interleaving

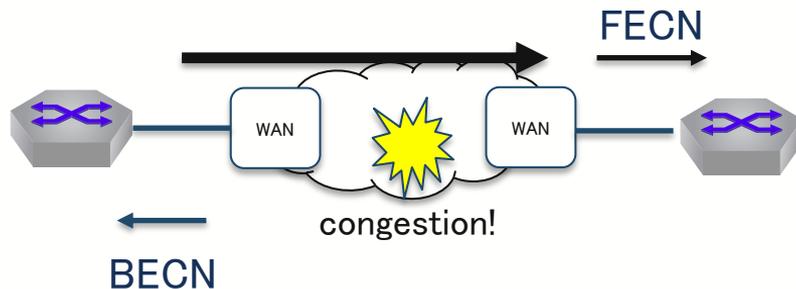
[https://www.cisco.com/c/en/us/support/docs/voice/voice-quality/7111-voip-mlppp.html?dtid=ossdc000283#link\\_frag](https://www.cisco.com/c/en/us/support/docs/voice/voice-quality/7111-voip-mlppp.html?dtid=ossdc000283#link_frag)



- 例えば1500バイトのロングパケットを500バイトで分割する
- フラグメントされたパケットをミッションクリティカルなRTPパケットの中に割り込みさせる
- フラグメントされたパケットは隣接ノードで再組み立てされる
- この技術はWANスピードが発達した事によって必要がなくなった(Ethernet Everywhere)

# Frame Relay Congestion Notification

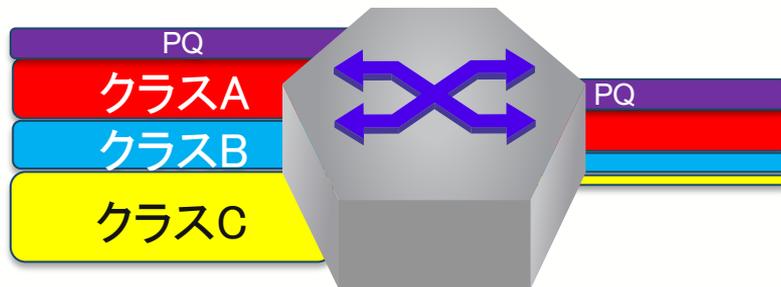
[https://en.wikipedia.org/wiki/Frame\\_Relay](https://en.wikipedia.org/wiki/Frame_Relay)



- フレームリレーサービスは輻輳通知のメカニズムを持っていた
- FECN(Forward Explicit Congestion Notification)とBECN(Backward Explicit Congestion Notification)はネットワークの輻輳通知を加入者にする事が出来る
- 知る限り使ってる人はいなかった。だって使ったもの勝ちだよな？
- TCP/IPはスライドウィンドウと再送制御で最大パフォーマンスと再送をしてくれるので帯域を使い切ったほうが良い

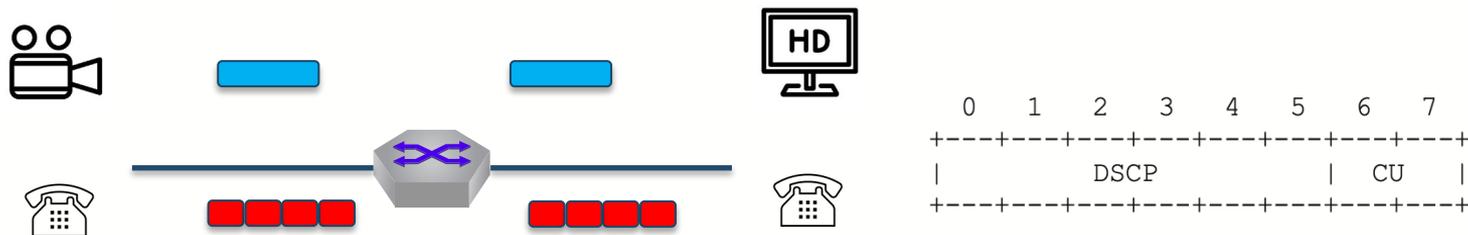
# LLQ(Low Latency Queueing)

[https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos\\_conmgt/configuration/xe-16-12/qos-conmgt-xe-16-12-book/qos-conmgt-llq-pps.html](https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_conmgt/configuration/xe-16-12/qos-conmgt-xe-16-12-book/qos-conmgt-llq-pps.html)



- PQとCBWFQの組み合わせ
- PQはFIFOで遅延する事なく通過する
- CBWFQは従来どおり重みに基づいたQueueing

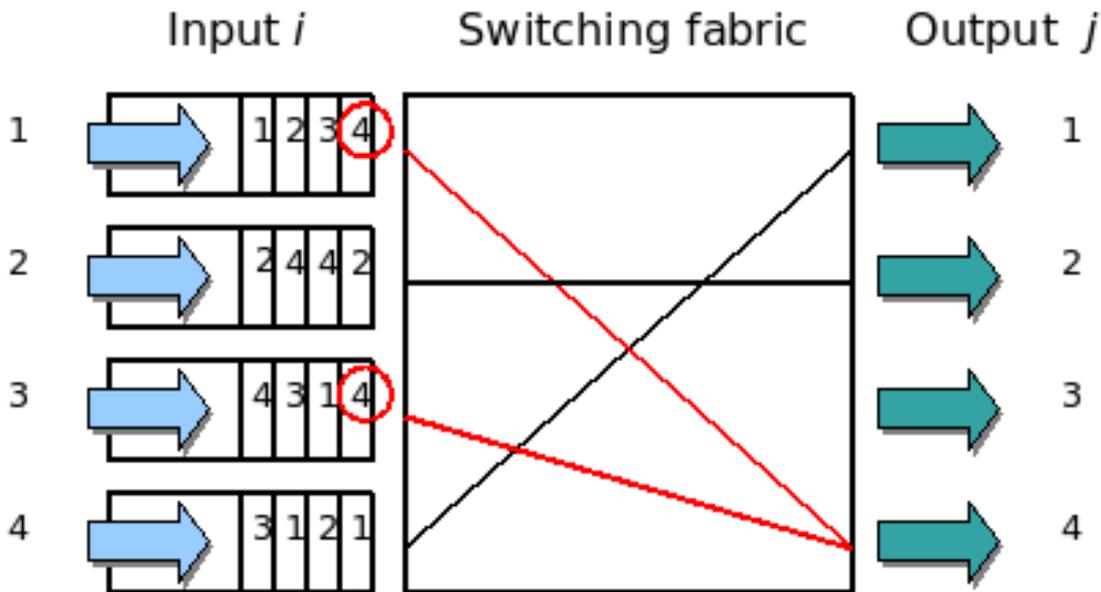
# 2006年頃マルチメディアサービスがスタート



- ビデオという新しいミッションクリティカルなサービスがスタート
- 音声と違ってバースト性も持つ
- ユニキャストだけでなくマルチキャストでのストリーミングもある
- 小さいパケットでバースト性がないVoIPとは違う性質を持つ
- 音声とビデオを違う低遅延のQueueに
  - 2LLQ+CBWFQ
  - マルチキャストQueueとユニキャストQueue
- この頃にはTOSフィールドはマルチサービスの為にDSCPに置き換えられた RFC2474 Differentiated Services Field redefines TOS field by DSCP for multimedia services
- またマルチサービスの為のガイドラインがPublishされたRFC4594 Guidelines for DiffServ Service Classes published as informational

# Head-of-line blocking/HOL

[https://en.wikipedia.org/wiki/Head-of-line\\_blocking](https://en.wikipedia.org/wiki/Head-of-line_blocking)



- 1番目と3番目のフローが同じポートに送信する為に競合してる
- もしファブリックが3番目のフローの送信を割り当てると1番目のフローは待つことになる
- 宛先が複数となるマルチキャストが共存するときこれらの問題が顕著になりユニキャストに影響が

# 映像品質劣化の知覚

[https://www.rd.ntt/ns/qos/technology/visual/01\\_2.html](https://www.rd.ntt/ns/qos/technology/visual/01_2.html)

表1.2.1 代表的な映像品質劣化

歪の種類		劣化の見え方	符号化劣化	伝送劣化
空間的 ひずみ	解像度低下・ぼけ	細かい模様や輪郭がぼけ、精細度が低下	○	
	ブロック歪	モザイク・幾何学パターン歪	○	○
	偽輪郭	明るさや色が緩やかに変化する部分に生じる偽の輪郭	○	
時間的 ひずみ	ジャーキネス	動きの滑らかさが失われ、ギクシャクして見える	○	○
	フリッカ	輝度レベルが変動し、ちらついて見える	○	
	動きぼけ	動いている領域がぼけて見える	○	
	途切れ／フリーズ	再生が途切れる／画面が停止する		○
時空間的 ひずみ	モスキートノイズ	エッジや色の変化の激しい部分で起こるノイズ（蚊が飛び回るように見えるノイズ）	○	
	エッジビジネス	エッジ部分がザラザラして、ちらついて見える	○	
	乱れ(破綻)	画面の一部または全体的に原形を留めない程の歪		○

- デジタル映像では、品質劣化は空間的・時間的に非定常に発生し、一旦乱れ(破綻)、途切れ、停止等の異常が発生すると、極端な劣化が起こるのが特徴

# RFC4594 Guidelines for DiffServ Service Classes

サービスクラス名	トラフィック特性	耐性		
		ロス	遅延	ジッター
Network Control/ネットワーク制御	様々なサイズのパケット。ほとんどは小さいサイズだが、バーストする事もある (BGP)	低い	低い	はい
テレフォニー	固定長の小さいパケット 低いレートフロー	大変低い	大変低い	大変低い
シグナル	可変サイズのパケット、短命のバーストトラフィックもある	低い	低い	はい
マルチメディア会議	可変サイズのパケット一定の送信インターバル、レートは適応型、ロスに対応	低いから中程度	大変低い	低い
リアルタイムインタラクティブ	RTP/UDPストリーム、非弾力的、ほとんどが可変レート	低い	大変低い	低い

# RFC4594 Guidelines for DiffServ Service Classes

サービスクラス名	トラフィック特性	耐性		
		ロス	遅延	ジッター
マルチメディアストリーム	可変サイズパケット、可変レートによる柔軟性	低いから中程度	中程度	はい
ブロードキャストビデオ	一定および可変レート、非弾力的、非バースト性フロー	大変低い	中程度	低い
低遅延データ	可変レート、バースト的で短命な柔軟なフロー	低い	低いから中程度	はい
OAM	可変サイズパケット、柔軟な・非柔軟なフロー	低い	中程度	低い
高スループットデータ	可変レート、バースト的な長寿命柔軟なフロー	低い	中程度から高い	はい
標準	なんでも	特に定義無し		
低優先データ	非リアルタイムで柔軟	高い	高い	はい

# Service Provider Network Configuration Example 1

```

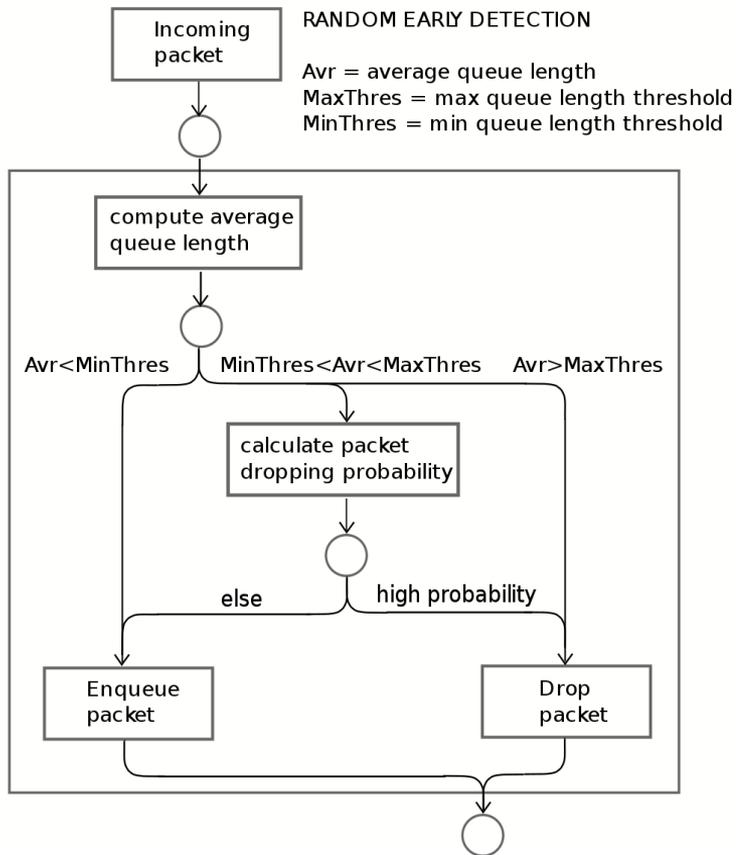
-----
| Service      | DSCP | Conditioning at | PHB | | | |
| Class       |      | DS Edge        | Used | Queuing | AQM |
|=====+=====+=====+=====+=====+=====|
| Network Control | CS6 | See Section 3.1 | RFC2474 | Rate | Yes |
|-----+-----+-----+-----+-----+-----|
| Telephony     | EF  | Police using sr+bs | RFC3246 | Priority | No |
|-----+-----+-----+-----+-----+-----|
| Signaling     | CS5 | Police using sr+bs | RFC2474 | Rate | No |
|-----+-----+-----+-----+-----+-----|
| Low-         | AF21 | Using single-rate, | | | |
| Latency      | AF22 | three-color marker | RFC2597 | Rate | per |
| Data         | AF23 | (such as RFC 2697) | | | DSCP |
|-----+-----+-----+-----+-----+-----|
| OAM          | CS2 | Police using sr+bs | RFC2474 | Rate | Yes |
|-----+-----+-----+-----+-----+-----|
| Standard     | DF(CS0) | Not applicable | RFC2474 | Rate | Yes |
|              | +other | | | | |
|-----+-----+-----+-----+-----+-----|
-----

```

- サンプルのQoS設定例を提示

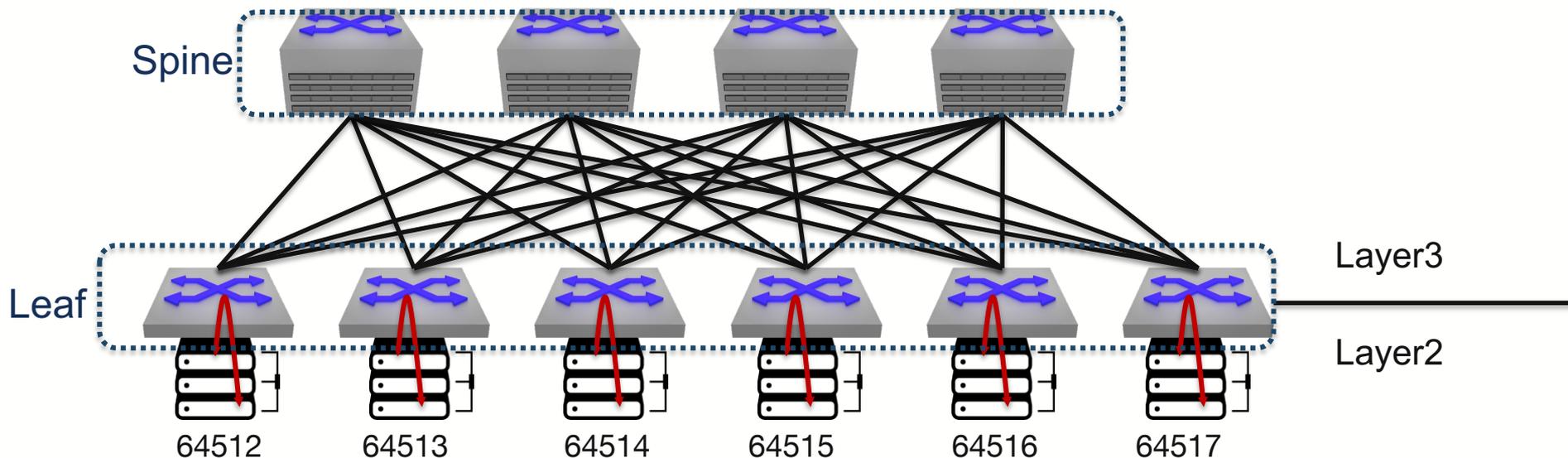
# AQM/Active Queue Management RED

[https://en.wikipedia.org/wiki/Random\\_early\\_detection](https://en.wikipedia.org/wiki/Random_early_detection)



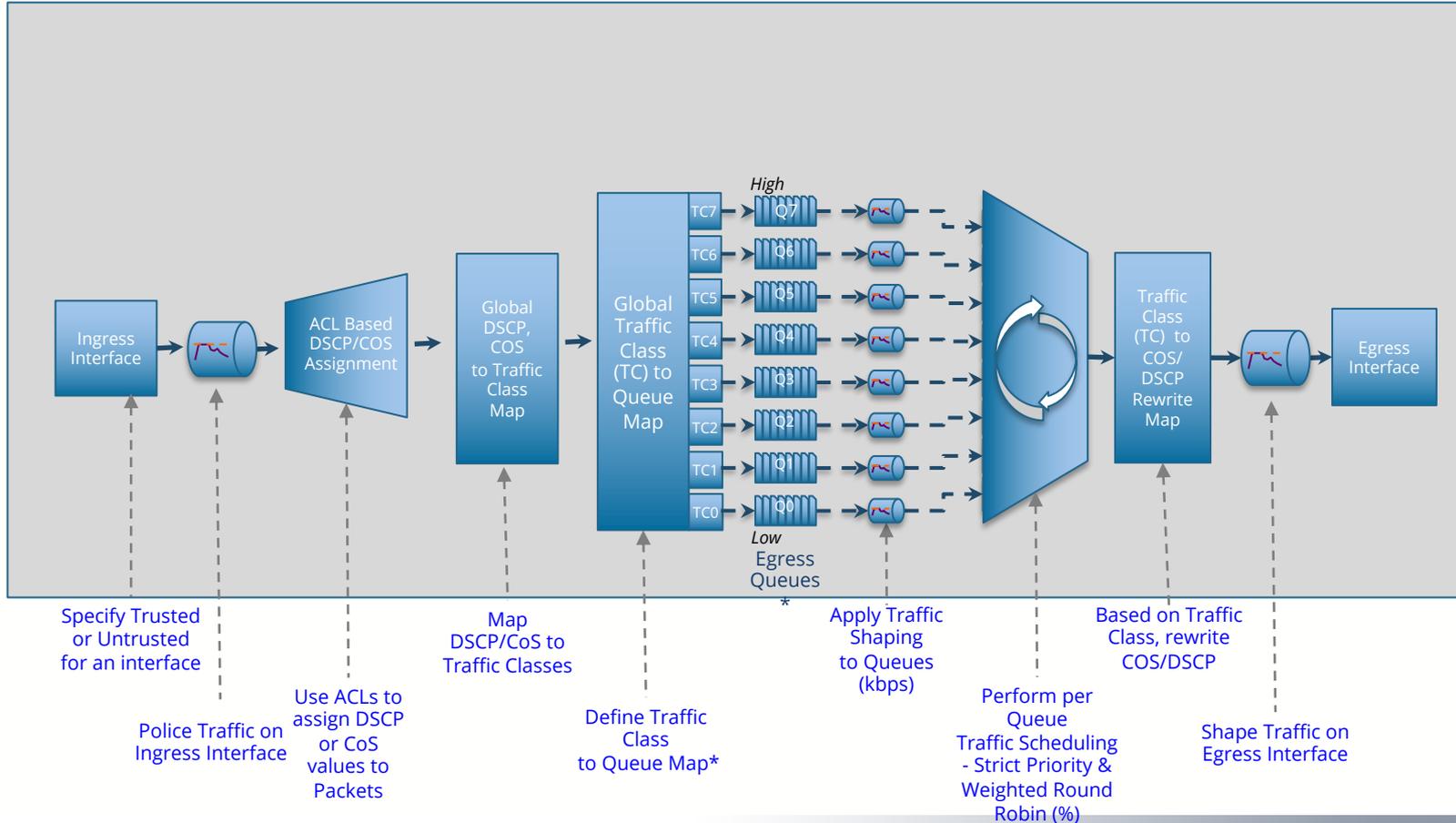
- RED (Random Early Detection) の様なQueueマネジメント手法
- ルータではパケットを可能な限り溜め込み、貯められなかったら捨てていく
- REDではQueueの長さを監視し、確率に基づいて廃棄もしくはマーキングする

# ではデータセンターのQoSはどのようなのか？



- BGPを使ったラージスケールデータセンターデザインはinformational RFCとしてRFC7938で2016年に公開
- 多くのデータセンターでは様々なサービスを展開してる
- データセンタースイッチでQoSを設定して事あるか？って聞くと、ほとんどのケースでノーと答えられる
- これは真実ではあるが、QoSを使っていないわけではないはず

# データセンタースイッチのQoS



# デフォルトのQoS挙動

- 入力識別
  - スイッチングポート(L2)Trust CoS
  - ルーティングポート(L3) Trust DSCP
- トラフィッククラスマッピング
  - 自動的にCoS/DSCP→トラフィッククラスにマッピングされる
  - トラフィッククラスは送信Queueにマッピングする
- レベルに基づいたStrict Priorityで動作する
- トラフィッククラスに基づいた8個ユニキャストQueueと4個のマルチキャストQueueがある

QoSは特別な設定をしなくてもアプリケーションがDSCPなどを意識してくれれば自然に動作する

# Virtual Output Queuingと動的デープバッファ

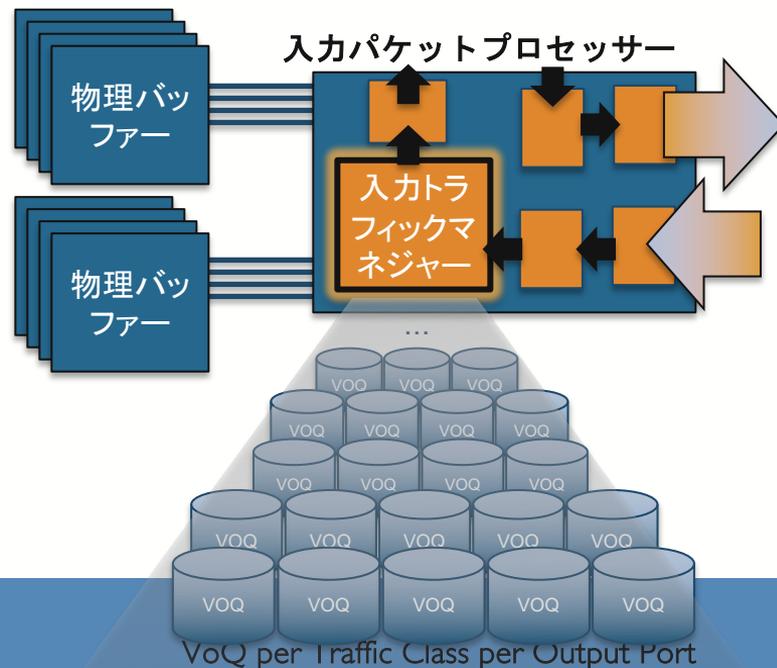
- 入力側で動的デープバッファ – フレームはスロットが予定されるまでキュー
- 入力側でVirtual Output Queues – Nのキューが出力ポートに割当

入力 packets プロセッサ 毎に 4GB のバッファ メモリー (システムで 288GB まで)

- ~30% のバッファ が出力ポート + TC で占有
- ~15% のバッファ が複数宛先のトラフィックに割当
- ~55% の動的バッファ 用に予約

VOQ サブシステム, システムで数百万のキュー

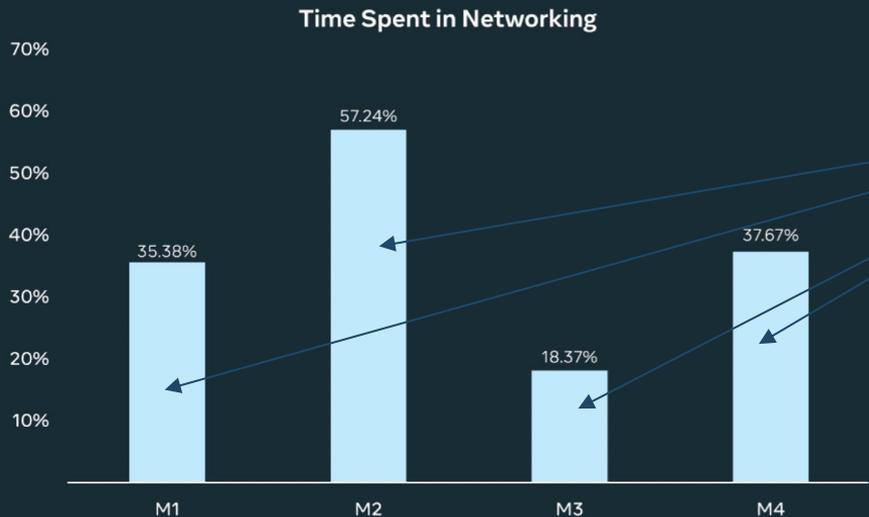
- VoQ 毎にキューを制限 (bytes, frames)
- 常に平等 – QoS ポリシーを常に適応



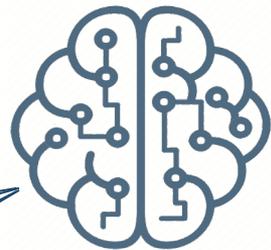
現在、将来の全てのアプリケーションに対応

# AIにおけるネットワークI/Oの重要性

NETWORK I/O IS KEY FOR RECOMMENDATION WORKLOADS



Ranking requires high injection & bisection bandwidth



つまりこの間高価なGPUは何も処理をしない。眠ってる



Infrastructure for Large Scale AI: "Empowering Open" Alexis Bjorlin (Meta)

<https://www.youtube.com/watch?v=miv5PEXXTmc>

[https://drive.google.com/file/d/1qqjo-5JtYAcRIK\\_LWYuQFH-b9MoFOP02/view?usp=sharing](https://drive.google.com/file/d/1qqjo-5JtYAcRIK_LWYuQFH-b9MoFOP02/view?usp=sharing)

# 聞いてみた



You

QoSとはなんですか？



ChatGPT

QoS (Quality of Service) は、ネットワークや通信システムにおけるサービス品質のことを指します。QoSは、通信の信頼性、遅延、帯域幅、ジッター (パケット間の時間の揺れ)、パケットロスなどのパラメータを評価し、これらのパラメータを制御するための手法や技術を包括しています。

QoSの目的は、異なるトラフィックタイプやネットワーク上のアプリケーションに優先順位を与え、ユーザーエクスペリエンスを向上させることです。たとえば、リアルタイムの音声やビデオ通信は、遅延やジッターが少なく、パケットロスが少ないことが求められます。一方、ファイル転送や電子メールなどの非リアルタイムのアプリケーションは、これらの条件に厳密に対処する必要はありません。

QoSを実装するための技術には、帯域幅制御、トラフィック優先度付け、キューイング、トラフィックシェーピング、制御と回復、およびトラフィック工学などが含まれます。これらの技術を使用することで、ネットワーク上のリソースを効果的に管理し、異なるアプリケーションやサービスに適切なサービス品質を提供することが可能になります。



# 聞いてみた



You

マシンラーニングのネットワークでの最適なQoS設定を教えてください



ChatGPT

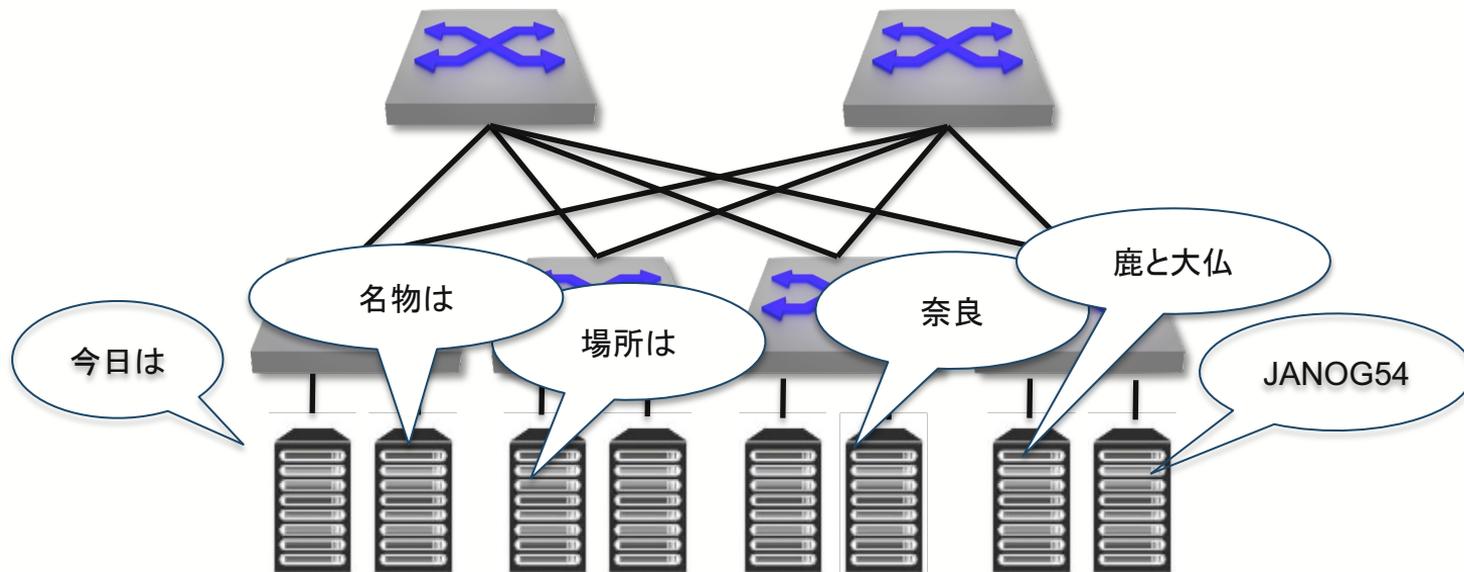
マシンラーニングのネットワークにおける最適なQoS設定は、特定の状況や要件に依存します。ただし、一般的なガイドラインとして以下のポイントが考えられます。

- リアルタイム通信への優先度付け:** マシンラーニングモデルがリアルタイムのデータストリームを処理する場合（例：リアルタイムのセンサーデータ、音声やビデオのストリーミングなど）、これらの通信に高い優先度を与える必要があります。このため、優先度を高く設定し、遅延やジッターを最小限に抑えるQoSポリシーを実装することが重要です。
- バッチ処理や非リアルタイム通信への適切なリソース割り当て:** マシンラーニングモデルのトレーニングやバッチ推論などの非リアルタイムの処理は、リアルタイム通信よりも柔軟なタイムラインで実行されることが多いです。そのため、これらの処理に必要なリソースを確保し、他の通信との競合を最小限に抑えることが重要です。
- トラフィックの分類と優先度付け:** マシンラーニングモデルによって生成されるトラフィックを適切に分類し、それぞれのトラフィックに対して適切な優先度を割り当てることが重要です。例えば、モデルのトレーニングデータや推論結果の転送には高い優先度を与えるべきです。
- ネットワークの帯域幅制御とキューイング:** マシンラーニングワークロードがネットワークを利用する場合、適切な帯域幅制御とキューイングメカニズムを実装することが重要です。これにより、ネットワークの過負荷やボトルネックを防ぎ、サービス品質を維持することができます。

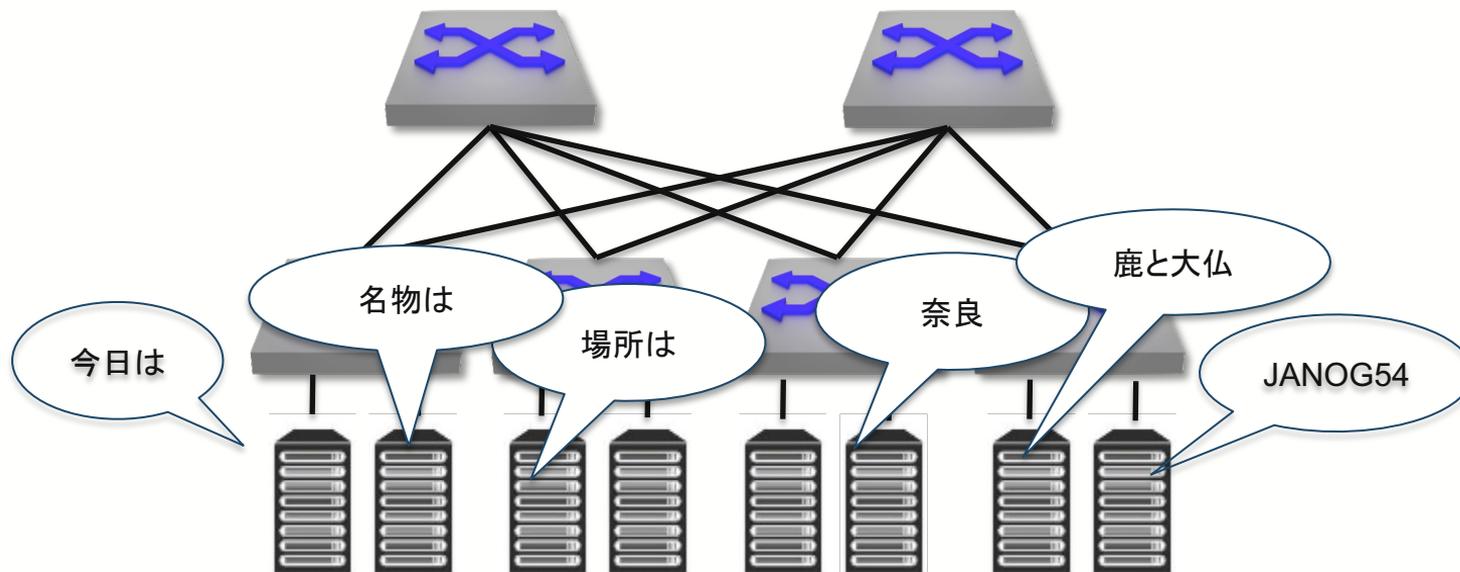
これらのポイントを考慮しながら、マシンラーニングワークロードに最適なQoS設定を行うことが重要です。ただし、環境や要件に応じて個別にカスタマイズする必要があります。



# つまりこういう事

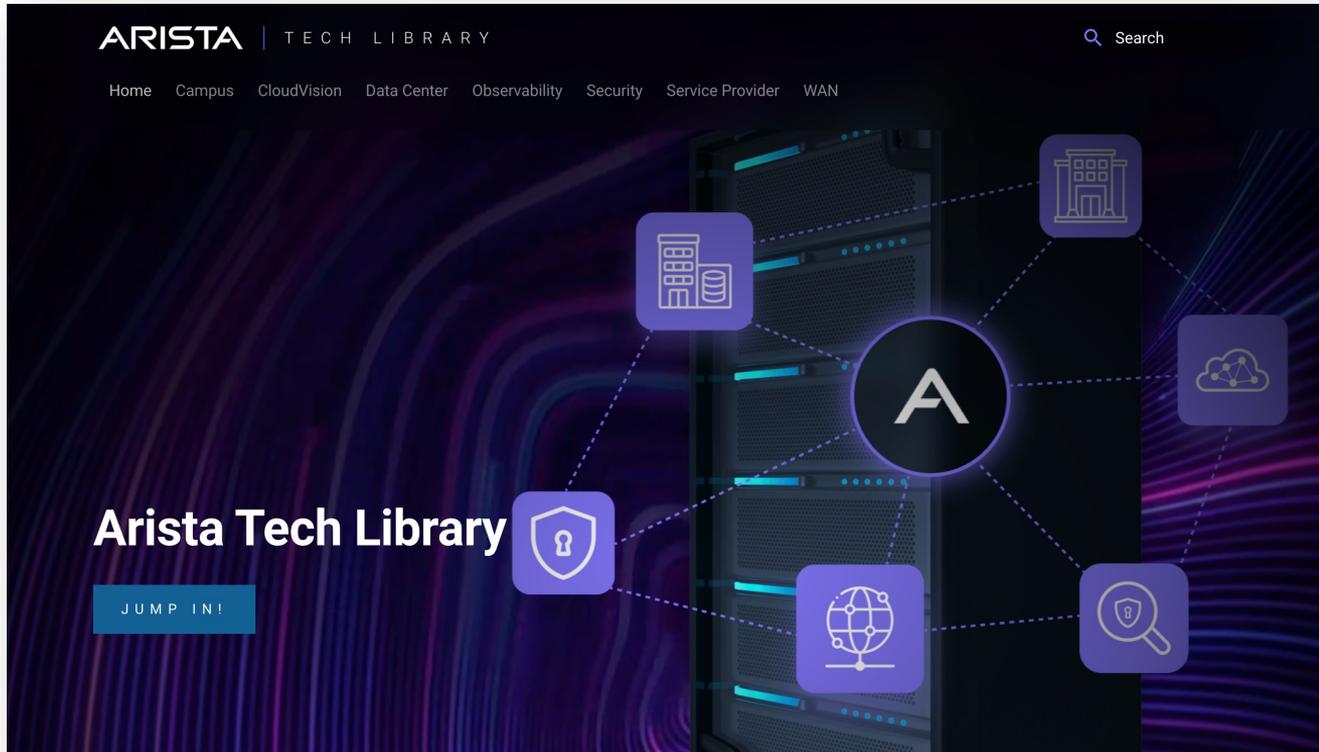


# つまりこういう事

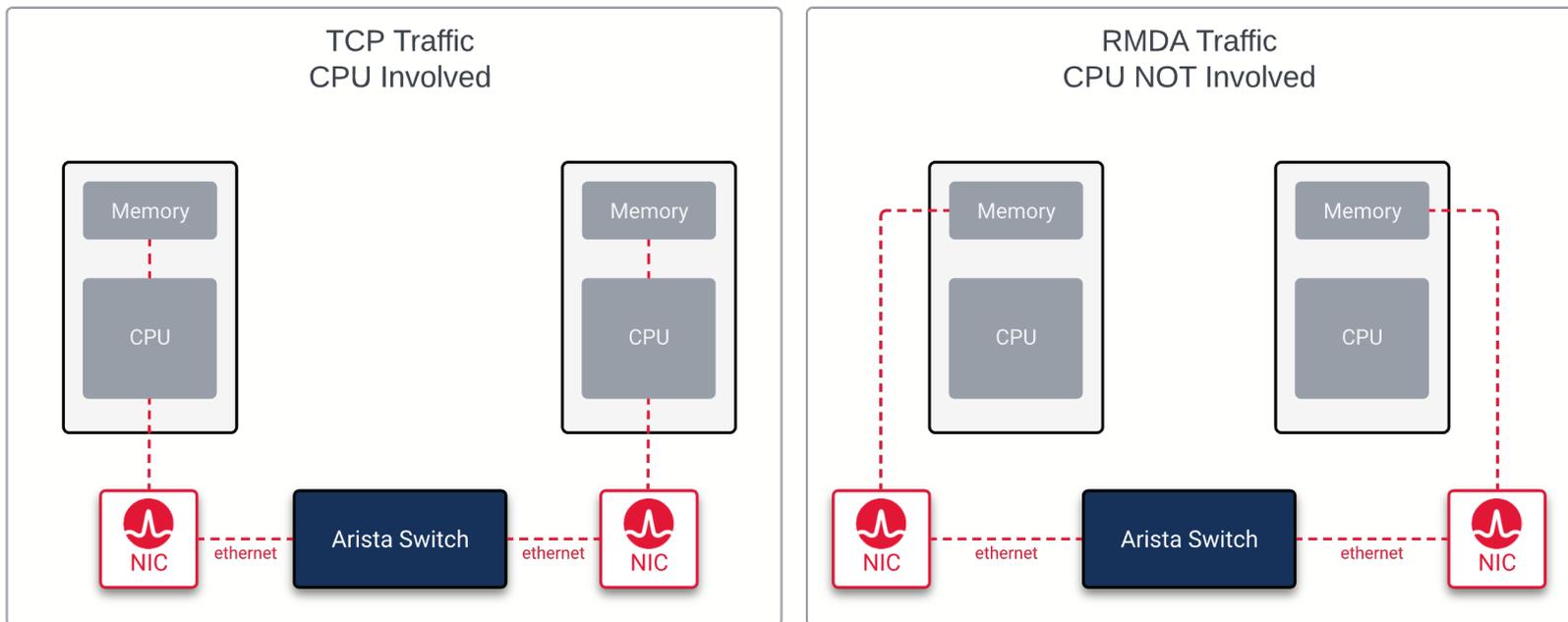


# Arista Tech Library

<https://tech-library.arista.com/>

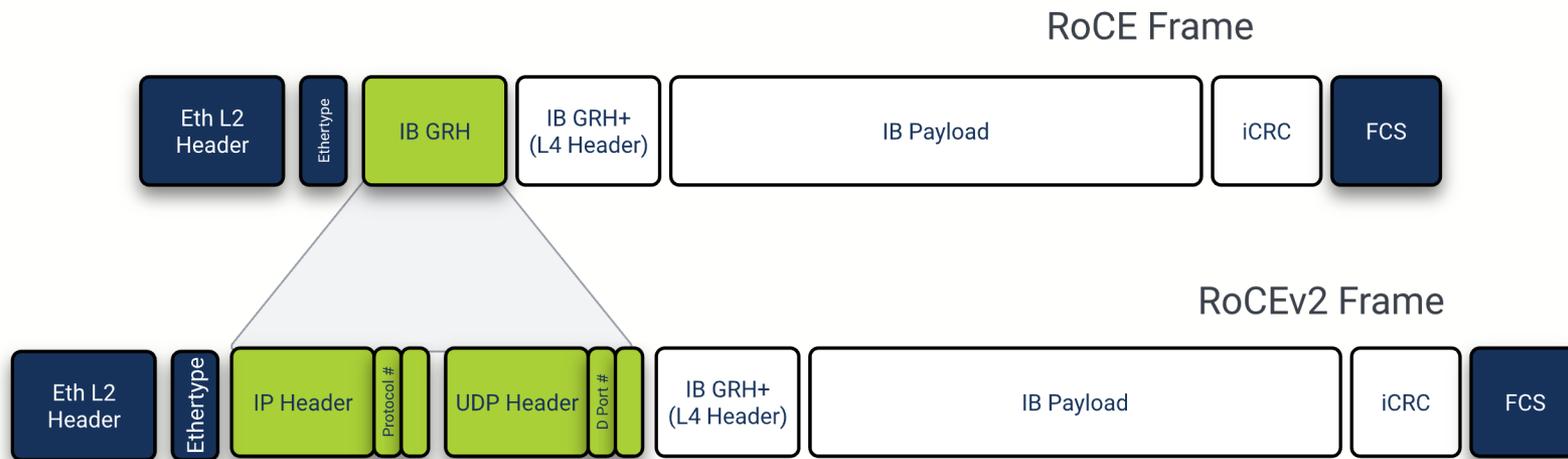


# RDMA



- RDMAではCPUを介せずネットワークを通じて直接やり取りを行う

# RoCEv2



- IPヘッダーを持つ事でルーティング可能
- UDPヘッダーによりECMPをしやすくする
- RoCEv2のデプロイにはPFCとECNが必要

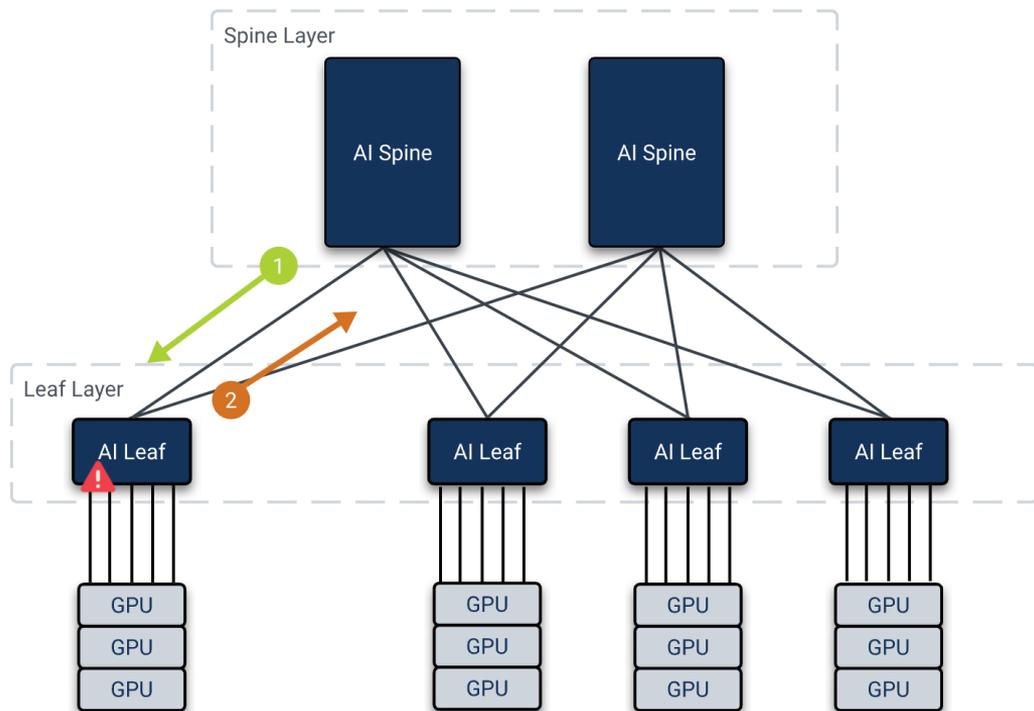
# PFC

Destination MAC	01:80:02:00:00:01	
Source MAC	Station MAC	
Ethertype	0x8808	
OpCode	0x0101	
Class Enable V	0x00 E7 ...E0	- Class-enable vector, 8 bits for each class
Time Class 0	0x0000	
Time Class 1	0x0000	
...		
Time Class 7	0x0000	

- PFCはIEEE 802.1Qbbで定義された輻輳制御のメカニズム
- クラス毎のフロー制御を行う



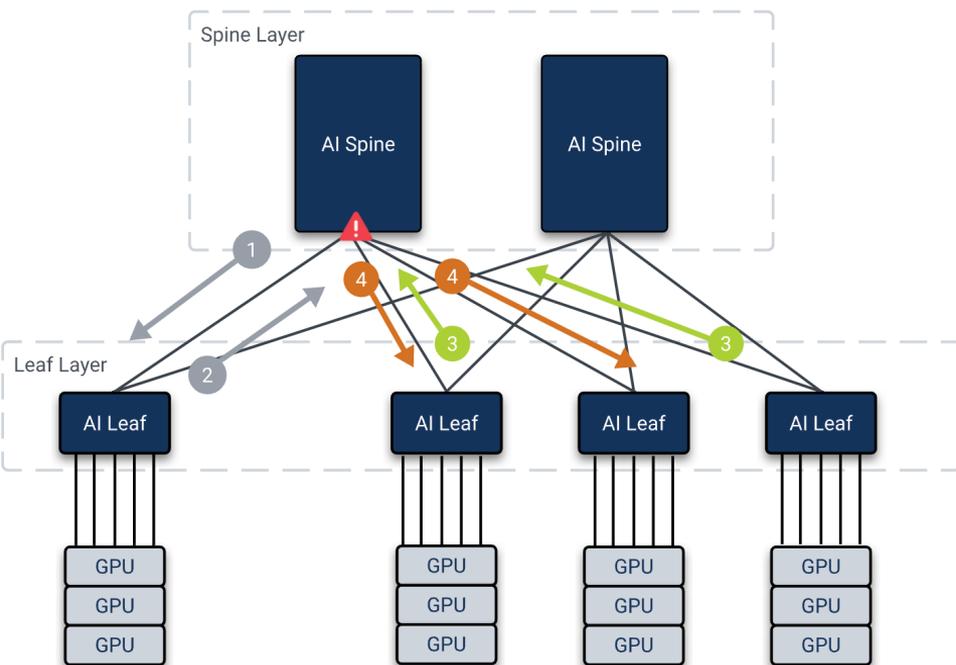
# 輻輳ポイント1 Egress Leaf



▲ Congestion point

- **!** 輻輳ポイント
- SpineからLeafへのトラフィックによりバッファの浅いスイッチでは輻輳が起こり、Queueが溜まりだす
- インターフェーススピードの変化とリモートLeaf→Spine→ローカルLeafのハッシュの偏りが原因

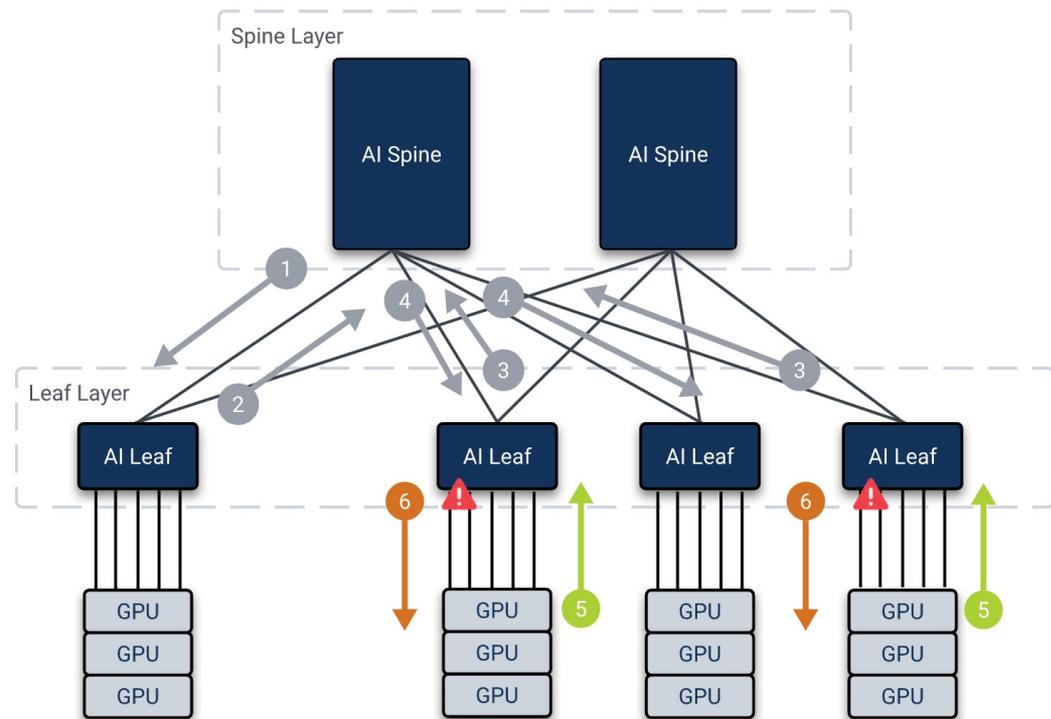
# 輻輳ポイント2 Spine



- **!** 輻輳ポイント
- 複数のLeafからSpineへのトラフィック
- 複数のleafからトラフィックが来ると深いバッファのスイッチであってもQueueを貯めるには十分な量になる

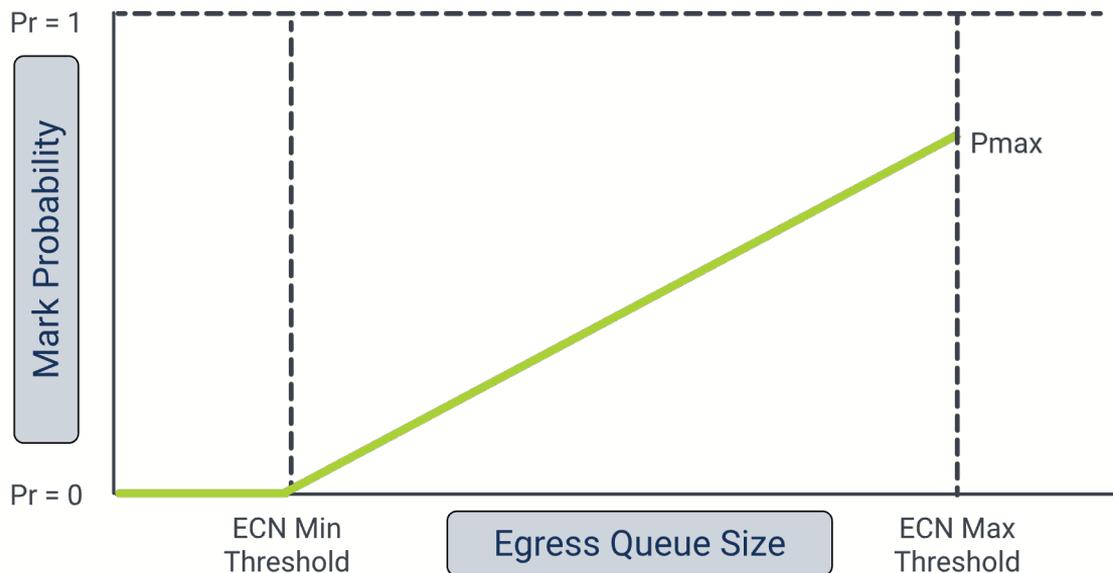
▲ Congestion point

# 輻輳ポイント3 Ingress Leaf



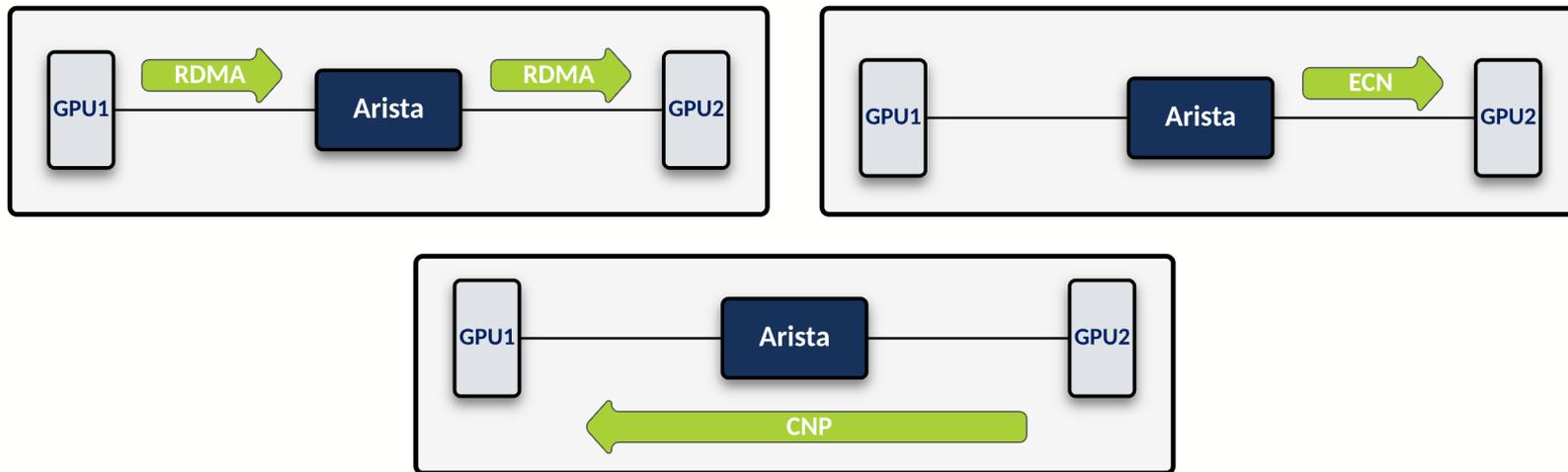
- **!** 輻輳ポイント
- GPUからLeafへのトラフィック
- 浅いバッファのスイッチだとIngress LeafでQueueが貯まる

# ECN/Explicit Congestion Notification



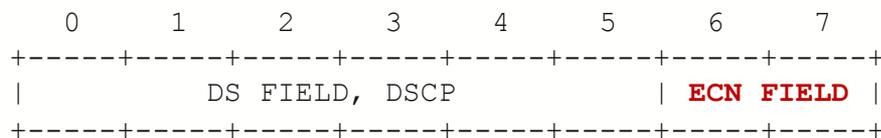
- パケットを破棄すること無くエンドツーエンドでネットワークの輻輳を通知する TCP/IPの拡張
- 平均Queueサイズが最小スレッシュホールドから最大スレッシュホールドまでの間はマーク確率は直線的に上がっていく為、最大スレッシュホールドよりも大きいときにすべてのパケットはECNビットをマークする

# ECN/Explicit Congestion Notification



- 受信側はECNビットのついたパケットを受け取るとCNP(Congestion Notification Packet)を送信する
- CNPにはフローに関する情報 QPが含まれており、送信側はどのフローが輻輳に陥ったのかを知ることが出来る
- 送信側は輻輳が解消されるまでフローのレートを制限し、エンドツーエンドのフロー制御を実現する

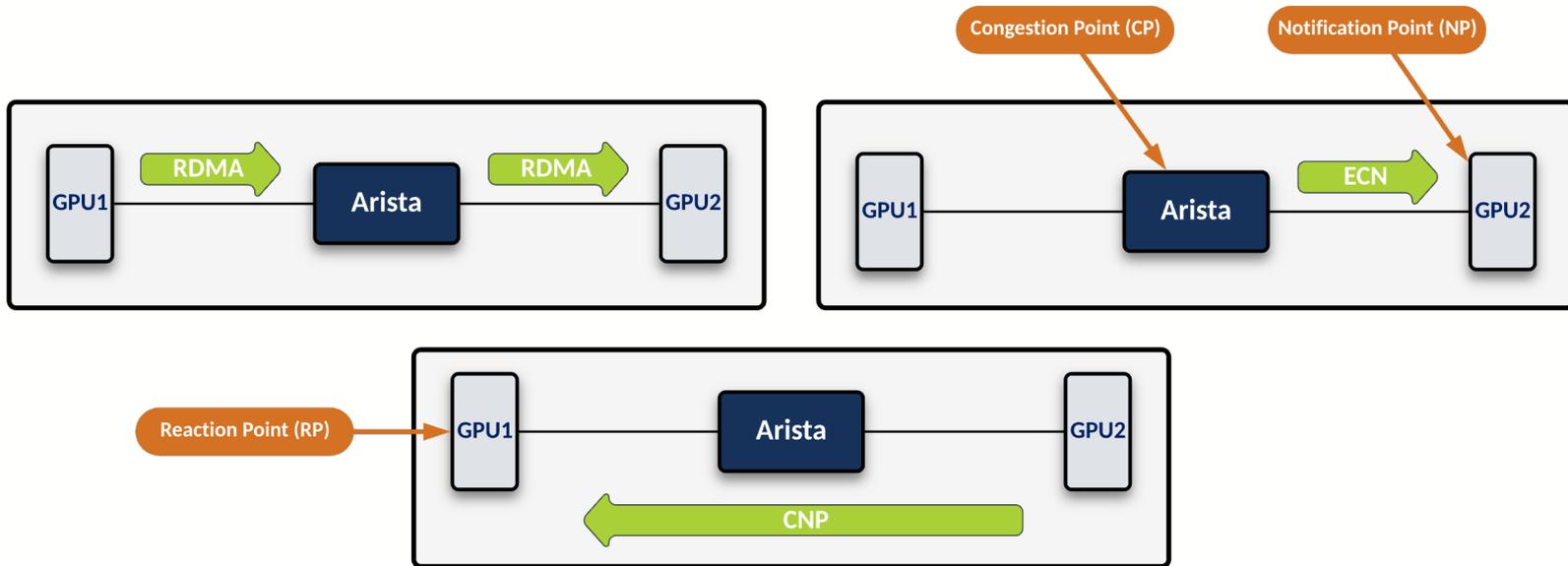
# ECN/Explicit Congestion Notification



Binary	Keyword
00	Not-ECT (Not ECN-Capable Transport)
01	ECT(1) (ECN-Capable Transport(1))
10	ECT(0) (ECN-Capable Transport(0))
<b>11</b>	<b>CE (Congestion Experienced)</b>

- ECPNはIPヘッダーのToSバイトの2ビットを使用する
- Weighted Random Early Detection(WRED)を拡張したもので平均キュー長が特定のしきい値を超えた場合にパケットを破棄する代わりにマークを付けるというものである
- WREDが特定のしきい値を超えたパケットを廃棄する代わりにマークを付ける

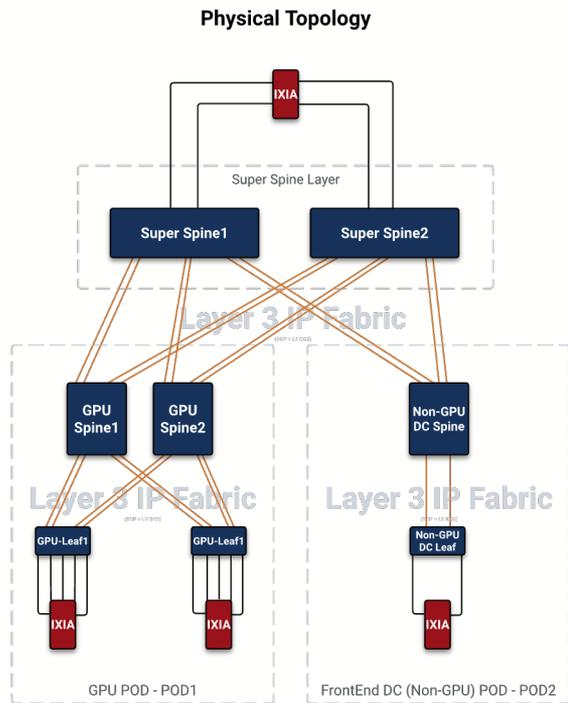
# DCQCN/Data Center Quantized Congestion Notification



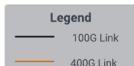
- Congestion Point(CP):ECNをマークしたスイッチ
- Notification Point(NP):ECNをマークしたパケットを受け取って、CNPを送信する受信GPUのNIC
- Reaction Point(RP):NPからCNPを受け取り、バッファオーバーフローさせないようにレートを調整

# RoCEv2 POC

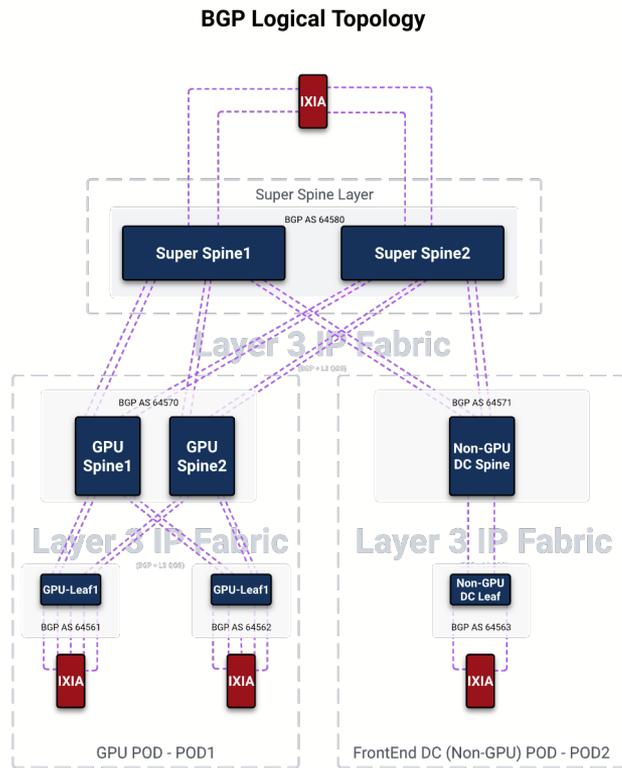
[https://tech-library.arista.com/data\\_center/rocev2/deployment\\_guide/rocev2\\_poc/](https://tech-library.arista.com/data_center/rocev2/deployment_guide/rocev2_poc/)



- POC(Proof Of Concept/コンセプトの実証)
- 物理トポロジー
- GPU PODとNon GPU PODを包括的に構成



# BGP Logical Topology



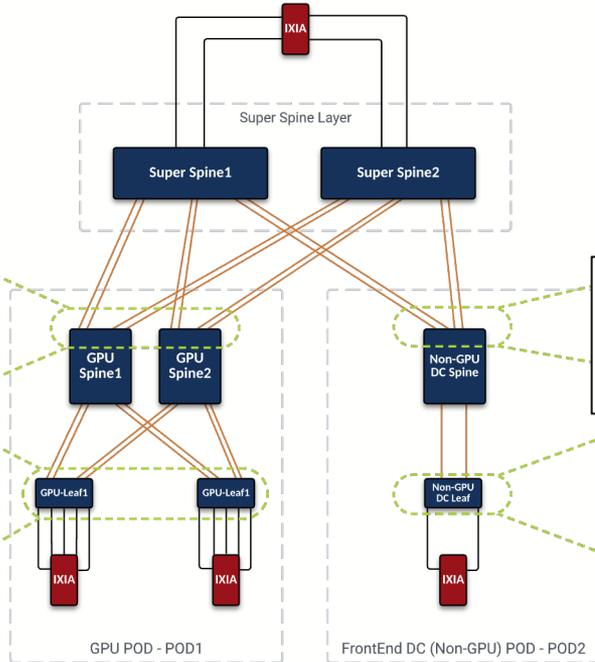
- BGPの論理トポロジー
- 3 Stage IP Closデザイン

Legend

----- eBGP IPv4

# QoS Logical Topology

## QoS Logical Topology



GPU PODから外部NWへのQoS  
Don't trust  
DSCP/TC1をMGMTトラフィック以外にset  
コントロールトラフィックだけ優先

GPU POD ROCEv3 QoSポリシー  
ROCE/MGMT/インターナル/ベストエフォート  
を識別 IngressでユニークなTCへセット  
SP Queue ROCE\_CNPNP WRR 他のトラフィックROCEv2をECN

### DC POD QoS Policy

Ingressにて  
フロントエンド DC NW MGMT/コントロールトラフィック/SDNトラフィックとベストエフォート  
GPU PODから偶発的にこぼれたROCEv2のトラフィックを低優先のTCにセットする

Egressにて  
コントロールトラフィックをSP Q  
他のすべてのトラフィックをWRR  
ROCEv2のトラフィックを低優先にてして帯域保証を無しにする  
DCPODはDCトラフィック以外では結ばない

### Legend

— 100G Link

— 400G Link

### ROCEv2 QoS Policy plan on GPU POD Leafs

Traffic Types	Marking Method	Input Markings	Queueing Behavior	ECN Enabled ?	TC/Queue
Best Effort	Trust Application Marking	CS0	WRR	N	1
Control Traffic	Trust Application Marking	CS7	WRR	N	7
Management Traffic	Trust Application Marking	CS5	WRR	N	4
ROCE CNP	Trust Application Marking	CS2	SP	N	6
ROCEv2	Trust Application Marking	CS3	WRR	Y	2
Virtual Cloud Traffic	Trust Application Marking	CS4	WRR	N	5

### ROCEv2 QoS Policy plan on GPU POD Spines

Traffic Types	Marking Method	Input Markings	Queueing Behavior	ECN Enabled ?	TC/Queue
Best Effort	Trust Application Marking	CS0	WRR	N	1
Management Traffic	Trust Application Marking	CS5	WRR	N	4
ROCE CNP	Trust Application Marking	CS2	SP	N	6
ROCEv2	Trust Application Marking	CS3	WRR	Y	2
Virtual Cloud Traffic	Trust Application Marking	CS4	WRR	N	5
Control Traffic	Trust Application Marking	CS7	WRR	N	7

# Leaf (GPU Pod):

```
class-map type qos match-any Best_Effort
  match dscp cs0
class-map type qos match-any Control_Traffic_cs7
  match dscp cs7
class-map type qos match-any Internal_Management_cs5
  match dscp cs5
class-map type qos match-any ROCE_CNP
  match dscp cs2
class-map type qos match-any ROCEv2
  match dscp cs3
class-map type qos match-any SDN_VCN
  match dscp cs4
!
interface Ethernet1/1
  service-profile GPU-LEAF-SCHEDULER
!
  uc-tx-queue 2
    random-detect ecn count
```

## Class-Map Config/Interface Config

```
policy-map type quality-of-service PM-GPU
  class ROCEv2
    set traffic-class 2
  !
  class SDN_VCN
    set traffic-class 5
  !
  class Best_Effort
    set traffic-class 1
  !
  class ROCE_CNP
    set traffic-class 6
  !
  class Internal_Management_cs5
    set traffic-class 4
  !
  class Control_Traffic_cs7
    set traffic-class 7
  !
  class class-default
    set traffic-class 1
```

## Policy-Map Config

# Leaf (GPU Pod):

## QoS Profile config

```
qos profile GPU-LEAF-SCHEDULER
  service-policy type qos input PM-GPU
  !
  uc-tx-queue 1
    !! Best-Effort
    bandwidth percent 40
  !
  uc-tx-queue 2
    !! ROCEv2
    no priority
    bandwidth percent 54
    random-detect ecn minimum-threshold 256 kbytes maximum-threshold 512 kbytes max-mark-probability 100
weight 0
  !
  uc-tx-queue 4
    !! Internal-Management
    no priority
    bandwidth percent 1
  !
  uc-tx-queue 5
    !! SDN_VCN
    no priority
    bandwidth percent 5
```

# EgressポリシーやSP QueueとBRR Queueとアロケーションの確認

```
EOS# show qos interfaces ethernet 28/1
```

```
Ethernet28/1:
```

```
Trust Mode: DSCP
```

```
Default COS: 0
```

```
Default DSCP: 0
```

```
DSCP rewrite for MPLS encap packets: Disabled
```

```
Port shaping rate: disabled
```

Tx Queue	Bandwidth (percent)	Bandwidth Guaranteed (units)	Shape Rate (units)	Priority Group	Priority	ECN/WRED
UC7	- / -	- / - (-)	- / - (-)	SP / SP	0	D
UC6	- / -	- / - (-)	- / - (-)	SP / SP	0	D
MC1	- / -	- / - (-)	- / - (-)	SP / SP	0	N/A
UC5	5 / -	- / - (-)	- / - (-)	RR / SP	0	D
UC4	1 / -	- / - (-)	- / - (-)	RR / SP	0	D
UC3	- / -	- / - (-)	- / - (-)	RR / SP	0	D
UC2	54 / -	- / - (-)	- / - (-)	RR / SP	0	L
UC1	40 / -	- / - (-)	- / - (-)	RR / SP	0	D
UC0	- / -	- / - (-)	- / - (-)	RR / SP	0	D
MC0	- / -	- / - (-)	- / - (-)	RR / SP	0	N/A

Note: Values are displayed as Operational/Configured

Legend:

RR -> Round Robin

SP -> Strict Priority

- -> Not Applicable / Not Configured

% -> Percentage of reference

ECN/WRED: L -> Queue Length ECN Enabled

W -> WRED Enabled

D -> Disabled

Service-policy PM-GPU input ( programming: Successful )

## Tx Queue毎のカウンターと廃棄統計の確認

```
EOS# show interfaces ethernet 1/1 counters queue detail
```

Port	TxQ	Counter/pkts	Counter/bytes	Drop/pkts	Drop/bytes
Et1/1	UC0	0	0	0	0
Et1/1	UC1	17018275	7370231514	0	0
Et1/1	UC2	17018762	7366037948	0	0
Et1/1	UC3	10372468	4493654012	0	0
Et1/1	UC4	11581601	7369606444	0	0
Et1/1	UC5	17018574	7370116628	0	0
Et1/1	UC6	17012761	7365945326	0	0
Et1/1	UC7	0	0	0	0
Et1/1	UC8	0	0	0	0
Et1/1	MC0	0	0	0	0
Et1/1	MC1	0	0	0	0

## ECNコンフィグの確認

```
EOS# show qos interfaces ethernet 10/1 ecn
```

```
Ethernet10/1:
```

Tx-Queue	Threshold	Unit	ECN Minimum Threshold	ECN Maximum Threshold	ECN Delay Threshold	ECN Mark-Prob
7	-		-	-	-	- / -
6	kbytes		120	150	-	80 / 80
5	-		-	-	-	- / -
4	-		-	-	-	- / -
3	-		-	-	-	- / -
2	-		-	-	-	- / -
1	-		-	-	-	- / -
0	-		-	-	-	- / -

```
Delay Threshold Type: G -> Global      L -> Local
```

```
Note: Values are displayed as Configured/Operational
```

## ECNマーキングされたパケットの確認

```
EOS# show qos interfaces ethernet 32/1 ecn counters queue
```

```
Ethernet32/1:
```

```
Tx-Queue
```

```
Marked Packets
```

-----	-----
0	-
1	-
2	134929208
3	-
4	-
5	-
6	-
7	-

# PFCの設定

```
interface Ethernet1/1
  priority-flow-control mode on
  priority-flow-control priority 2 no-drop
  !
  tx-queue 2
  no priority
  bandwidth percent 40
```

## PFCのモニター

```
EOS# show priority-flow-control interfaces ethernet 3/1/1
```

```
The hardware supports PFC on priorities 0 1 2 3 4 5 6 7
PFC receive processing is enabled on priorities 0 1 2 3 4 5 6 7
The PFC watchdog timeout is 1.0 second(s)
The PFC watchdog recovery-time is 2.0 second(s) (auto)
The PFC watchdog polling-interval is 0.2 second(s)
The PFC watchdog action is drop
The PFC watchdog override action drop is false
Global PFC : Disabled
```

```
E: PFC Enabled, D: PFC Disabled, A: PFC Active, W: PFC Watchdog Active
```

Port	Status	Priorities	Action	Timeout	Recovery Interval/Mode	Polling Config/Oper	Note
Et3/1/1	E - -	01	-	-	- / -	- / -	DCBX disabled

Port	RxPfc	TxPfc
Et3/1/1	0	0

## PFCのモニター

```
EOS# show priority-flow-control counters history
```

```
Last Clear Timestamp: never
```

Interface	Priority	RX PFC Off to On Count	Max RX PFC On Time (ns)	Max RX PFC Timestamp	Total RX PFC On Time (ns)
Et1/1	PFC0	0	0	never	0
Et1/1	PFC1	0	0	never	0
Et1/1	PFC2	0	0	never	0
Et1/1	PFC3	1	272,376,473,453	2023-04-26 11:50:44	272,376,473,452
Et1/1	PFC4	0	0	never	0
Et1/1	PFC5	0	0	never	0
Et1/1	PFC6	0	0	never	0
Et1/1	PFC7	0	0	never	0
Et2/1	PFC0	0	0	never	0
Et2/1	PFC1	0	0	never	0
Et2/1	PFC2	0	0	never	0
Et2/1	PFC3	0	0	never	0
Et2/1	PFC4	0	0	never	0
Et2/1	PFC5	0	0	never	0
Et2/1	PFC6	0	0	never	0
Et2/1	PFC7	0	0	never	0

## PFCのモニター

```
EOS# show priority-flow-control interfaces Et1/1 counters history
```

```
Last Clear Timestamp: never
```

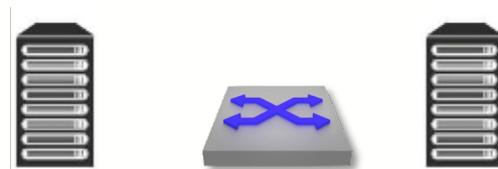
Interface	Priority	RX PFC Off to On Count	Max RX PFC On Time (ns)	Max RX PFC Timestamp	Total RX PFC On Time (ns)
Et1/1	PFC0	0	0	never	0
Et1/1	PFC1	0	0	never	0
Et1/1	PFC2	0	0	never	0
Et1/1	PFC3	1	272,376,473,453	2023-04-26 11:50:44	272,376,473,452
Et1/1	PFC4	0	0	never	0
Et1/1	PFC5	0	0	never	0
Et1/1	PFC6	0	0	never	0
Et1/1	PFC7	0	0	never	0

# まとめ

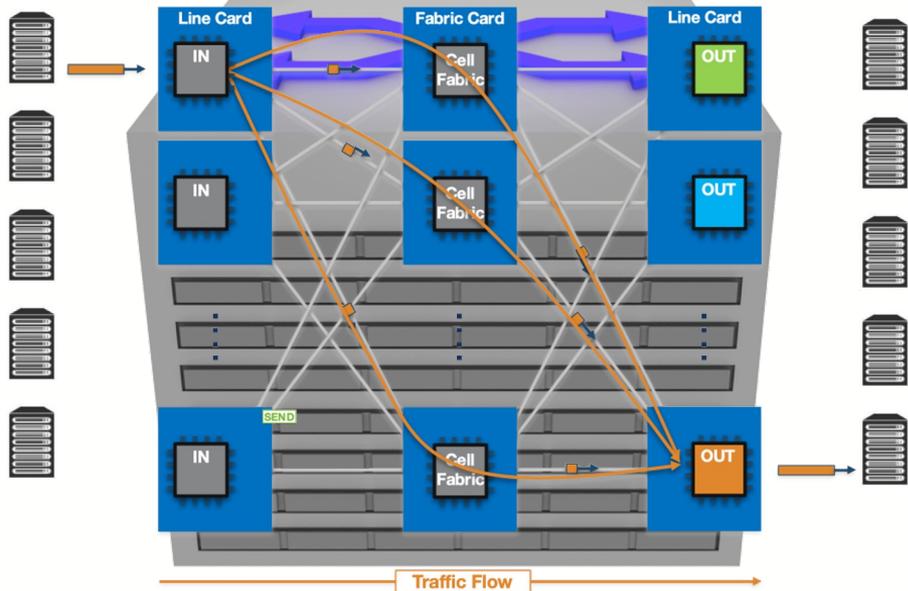
- マシンラーニングのネットワークで個人的に面白いなあってポイントをまとめてみた
  - 音声でも映像でもない新しい頭脳というアプリケーション
  - 品質定義はまだ定められてない
  - そういえば最近のスイッチはデフォルトでQoSが有効になっている
  - フレームリレーでは特に使われなかった(と思う)輻輳通知のメカニズムが人工知能(?RDMA)はダイナミックに使ってくる
- 実証検証が重要になるためヒントとなるRoCEv2のPOC情報を紹介

# 誰がネットワークの影響を感じるか

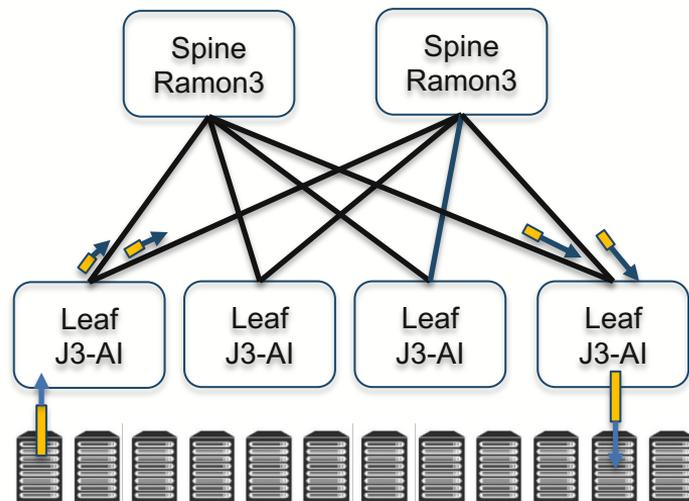
- 音声=>人が耳で確認していた
- 映像=>人が目と耳で確認していた
- AI/ML=>…GPU自身がJCT(Job Complete Time)とCCL(Collective Communication Library)のトラフィックパターンを知ってる



# Jericho3-AI Ethernet Fabric



Jericho2/Ramon2 モジュラーシステムの  
シャーシ内のセルスイッチング



Broadcom Jericho3 AI Ethernet Fabric

<https://youtu.be/fMKq0Y0jPHk?si=lwyoQBTvqTSIXLp6>

- スイッチングチップのJericho3-AIとファブリックチップのイーサネットファブリックの様に使用する
- シャーシ内のアーキテクチャーの様に帯域を有効に使い、数百万台のGPUの展開であってもサポート可能
- ブロードコムが提示したソリューションとして1つの可能性(標準的なアプローチになるのか?)

# 参考URL

- [AI Network WP](#)
- [Congestion Control for Large-Scale RDMA Deployments](#)
- [Arista Boadcom RoCE DSv6](#)
- [EOS DCBX and Flow Control](#)
- [Configure ECN Max Mark Probability and Weight](#)
- [ECN Counters per Tx Queue](#)
- [ECN Counters](#)
- [Priority Flow Control \(PFC\)](#)
- [PFC Global Knob](#)
- [PFC History Counters](#)

# Thank You

[arista.com](https://arista.com)