WANネットワークにおける監視の高度化手法: アクティブ監視、テレメトリー、AI技術の活用

ジュニパーネットワークス株式会社
山岸　祐大

Juniper Networks | Driven by Experience™

本日の話

ネットワークインフラの重要性が増す今、ネットワークはどのように監視するべきかを議論

通信インフラ障害による影響は甚大

“配送センターでの障害1時間が$35Mの損失に”
Amazon

“オリンピックの配信が10秒止まると
1億人の視聴者が100M走を見られない”
British Telecom

“学校のネットワークが止まることで
13,400人の学生がテストを受けられない”
Coppell Independent School District

“30秒のネットワーク停止中に
患者が病院から抜け出す可能性が”
Veterans Administration

通信インフラをどう守るか？

今の最適な監視の手段を議論

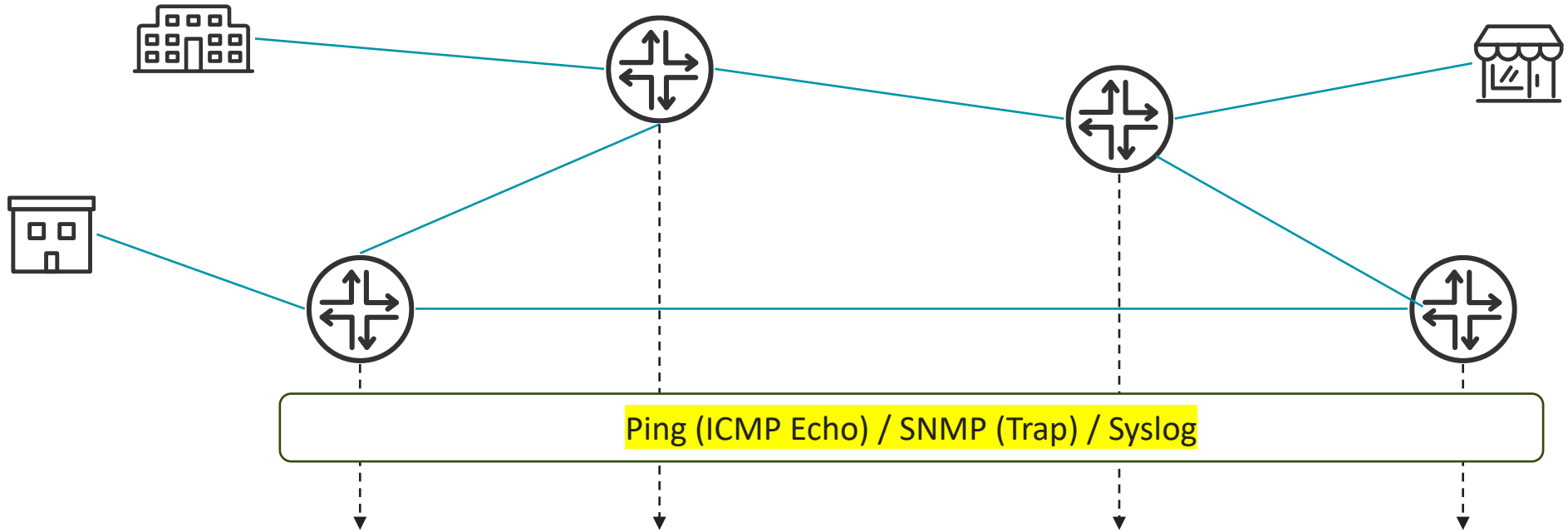
- PING/SNMP/SYSLOG？
- Telemetry？
- アクティブ監視？
- AI？

© 2025 Juniper Networks

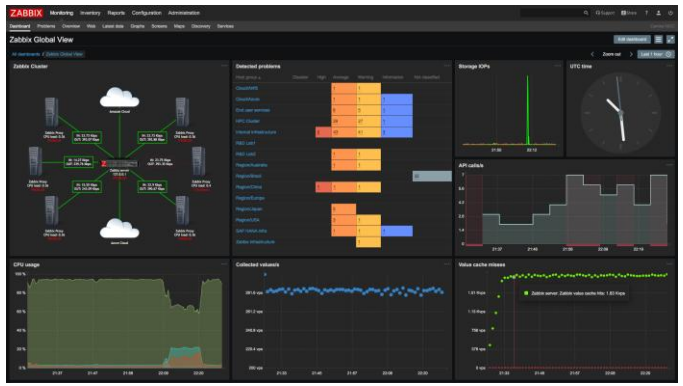
Juniper Business Use Only

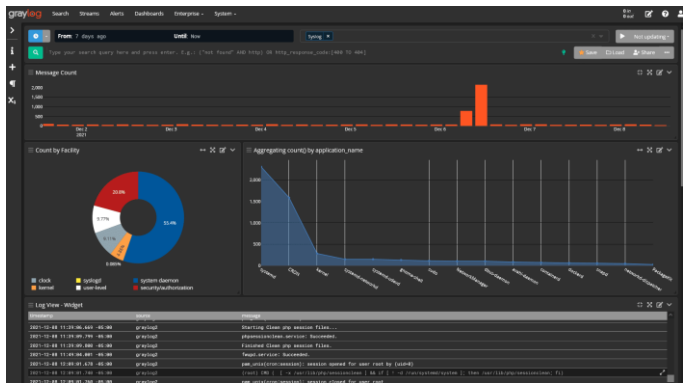
一般的なネットワーク監視

一般的に広く使われているネットワークの監視手段としてPingやSNMP、SYSLOGが挙げられる



監視ソフトウェア
(Ex. Zabbix, GrayLog, Grafana)





警報通知

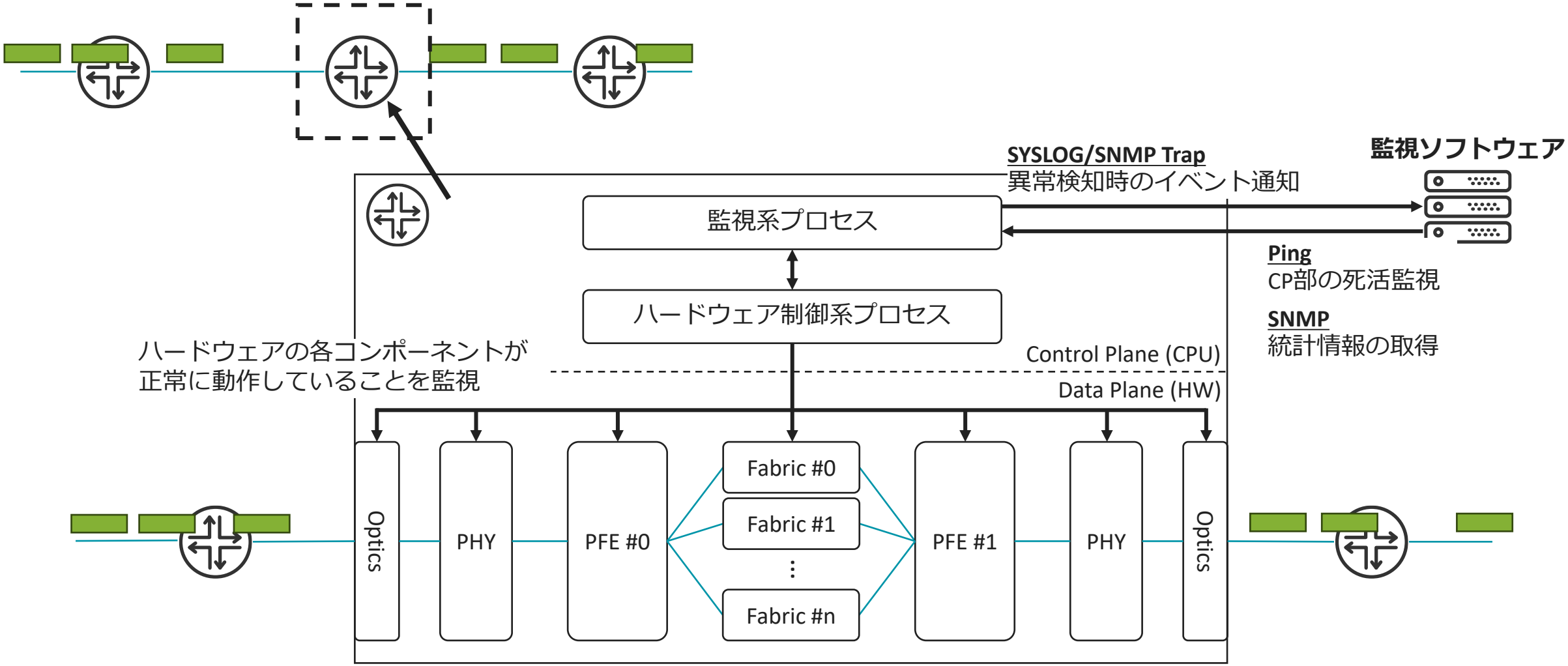
NOCオペレータ

© 2025 Juniper Networks

Juniper Business Use Only

一般的な監視手法で見てるもの

ネットワーク (面)を構成する各ルータ機器が、装置 (点)として正常に動作していることを監視

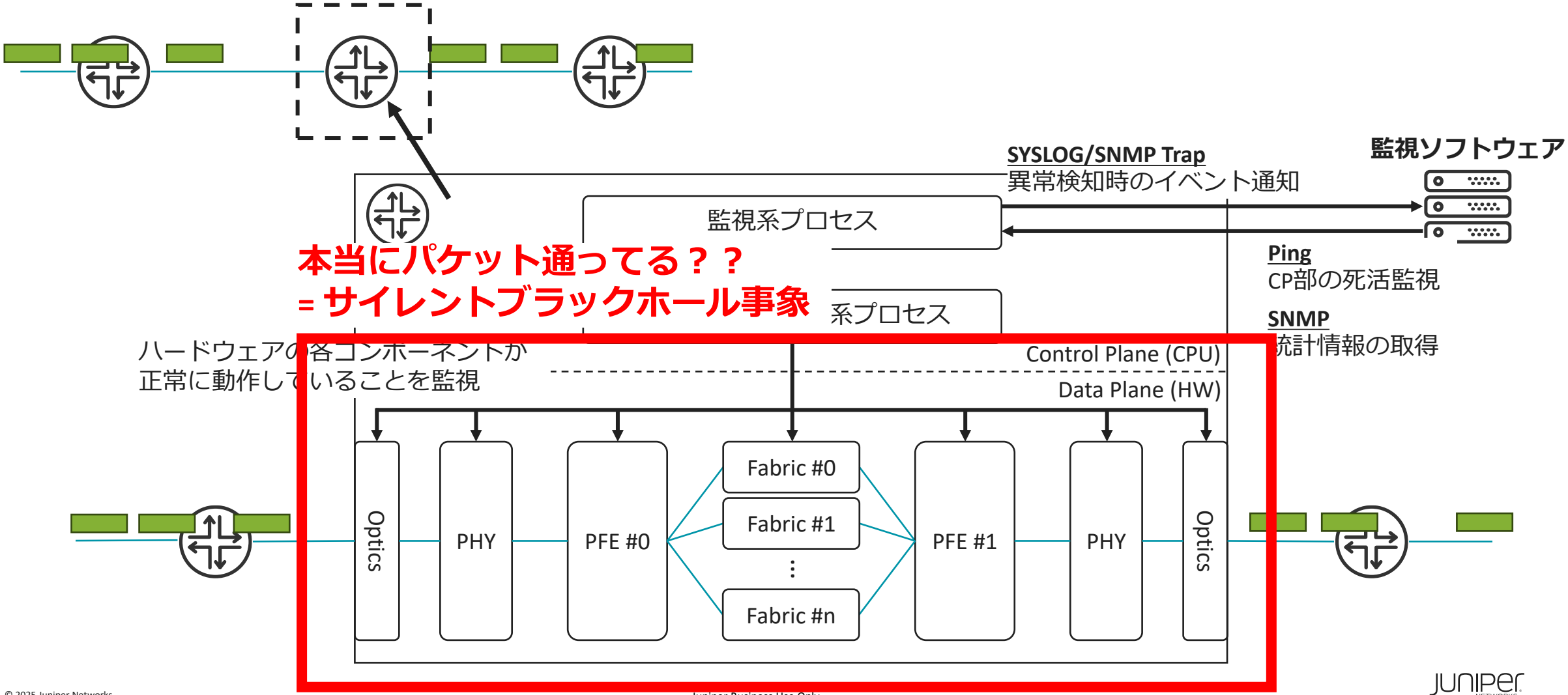


© 2025 Juniper Networks

Juniper Business Use Only

一般的な監視手法で見てるもの

ネットワーク (面)を構成する各ルータ機器が、装置 (点)として正常に動作していることを監視



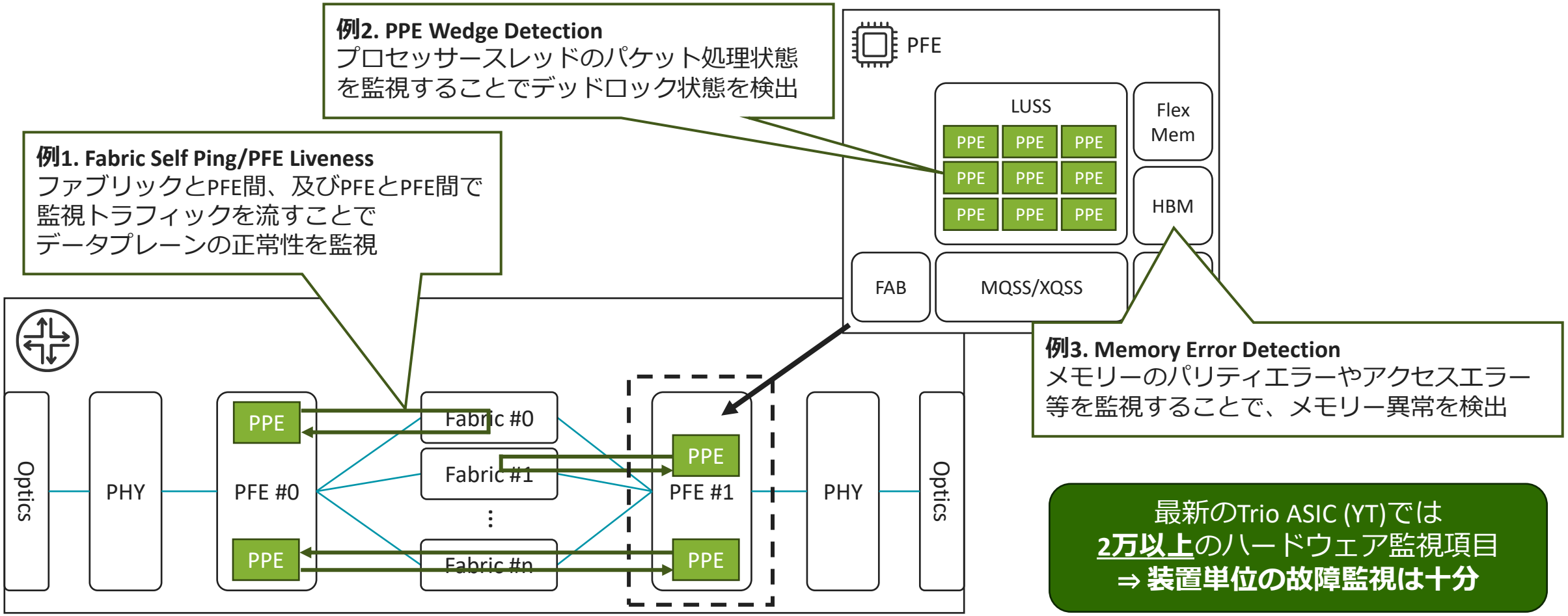
© 2025 Juniper Networks

Juniper Business Use Only

近年の高度な部品の正常性監視

故障時のブラックホール事象等を防止するためにベンダー製品は常に様々な改良が行われている

Juniper MXシリーズの例



最新のTrio ASIC (YT)では
2万以上のハードウェア監視項目
⇒ 装置単位の故障監視は十分

© 2025 Juniper Networks

面監視の必要性

近年のネットワーク障害の多くは機器故障以外が根本原因であるケースが多いと予想される

2024年 北米モバイルキャリア障害事例

- アメリカ全土でモバイルサービス全断
  1億以上のモバイル端末が影響
- 誤りのある設定をネットワークに展開
  ⇒ 数分後にネットワークが全断
- 復旧までに12時間近くかかった

2022年 北米大手通信キャリア障害事例

- コアネットワークにおいて障害が発生し、
  1200以上のモバイルや固定回線のサービスに影響
- ACLの設定更新により多量の経路更新情報が
  バックボーン上に伝搬し、機器が高負荷状態に
- ACLの設定をロールバックしたが機器が
  クラッシュしてバックボーンが全断

2021年 コンテンツプロバイダー障害事例

- オペレータが作業時に投入したコマンドにより
  不本意にインターネット接続断が発生
- マネジメントにも接続できなくなり
  オンサイト駆け付けが必要に
- 6時間近くサービスが使えない状態に

2020年 CDN事業者障害事例

- 設定変更が原因で27分のサービス影響が発生影響、
  約50%のトラフィックが影響
- ルーティングポリシーの設定ミスにより
  意図しない形でトラフィックがルーティングされるように
- 通信が一部のサーバに集中しサービス影響が発生

個々の機器は正常に動作している
しかし、ネットワークの面としては正常に動作していない (壊れた状態)

© 2025 Juniper Networks
Juniper Business Use Only

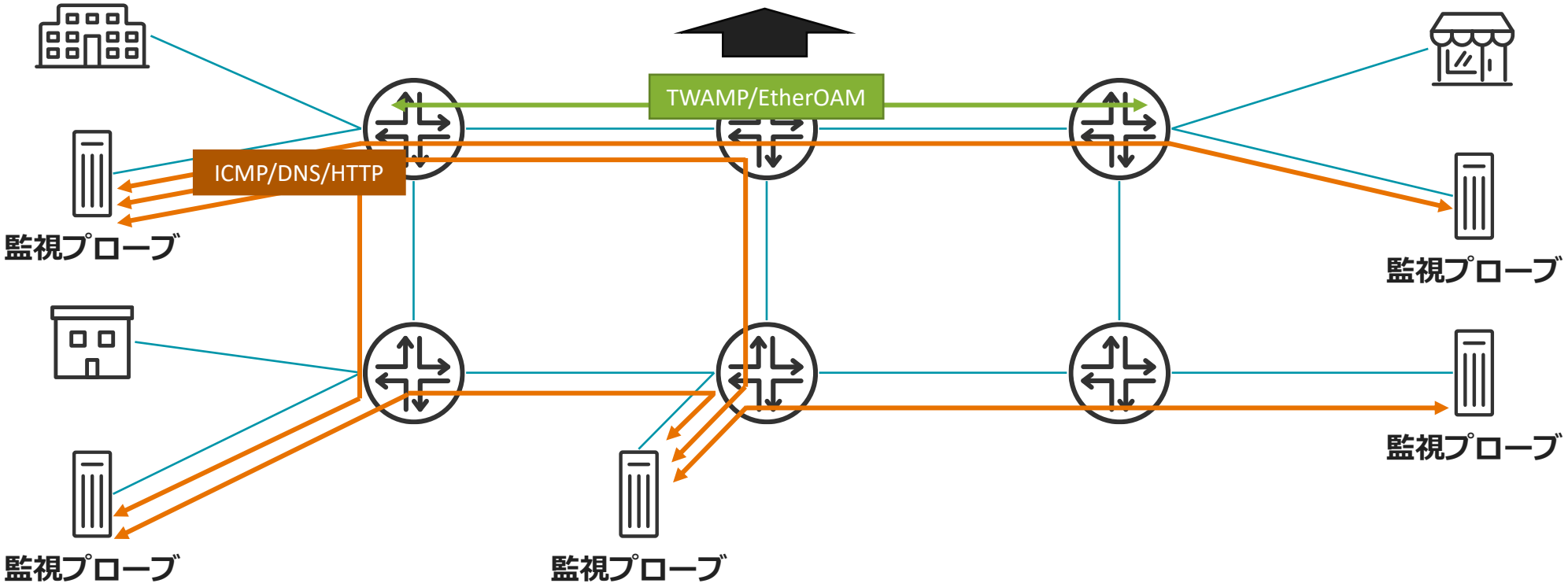
面の監視: アクティブ監視

ネットワーク上でプローブやルータが監視トラフィックを流すことで正常性を監視する手法

監視ソフトウェア
(Ex. Juniper Routing Director,
Cisco Thousand Eyes,
Cisco Provider Connectivity Assurance)

SLA違反通知
疎通性の問題通知

NOCオペレータ

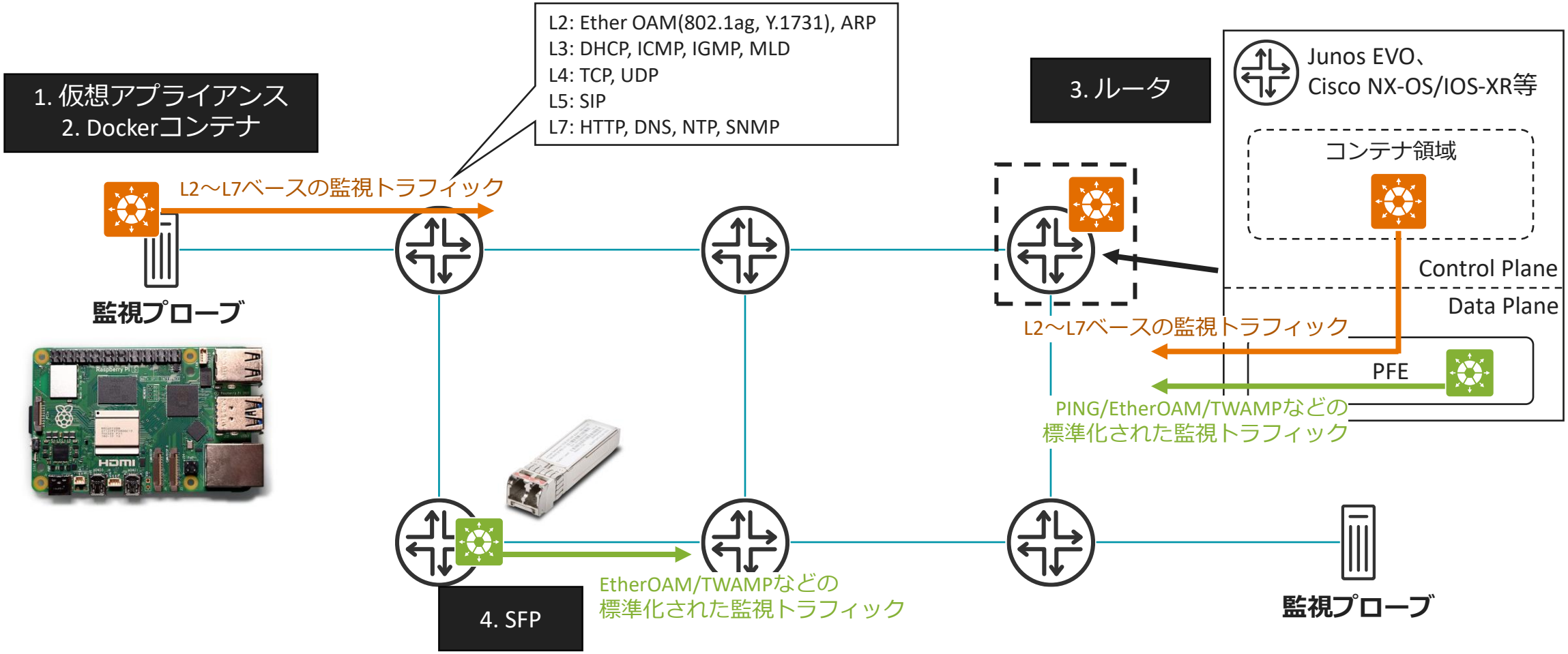


© 2025 Juniper Networks

Juniper Business Use Only

監視プローブの種類

x86上で動作するものからSFP形状のものやルータ上で動作するものなど様々な選択肢がある

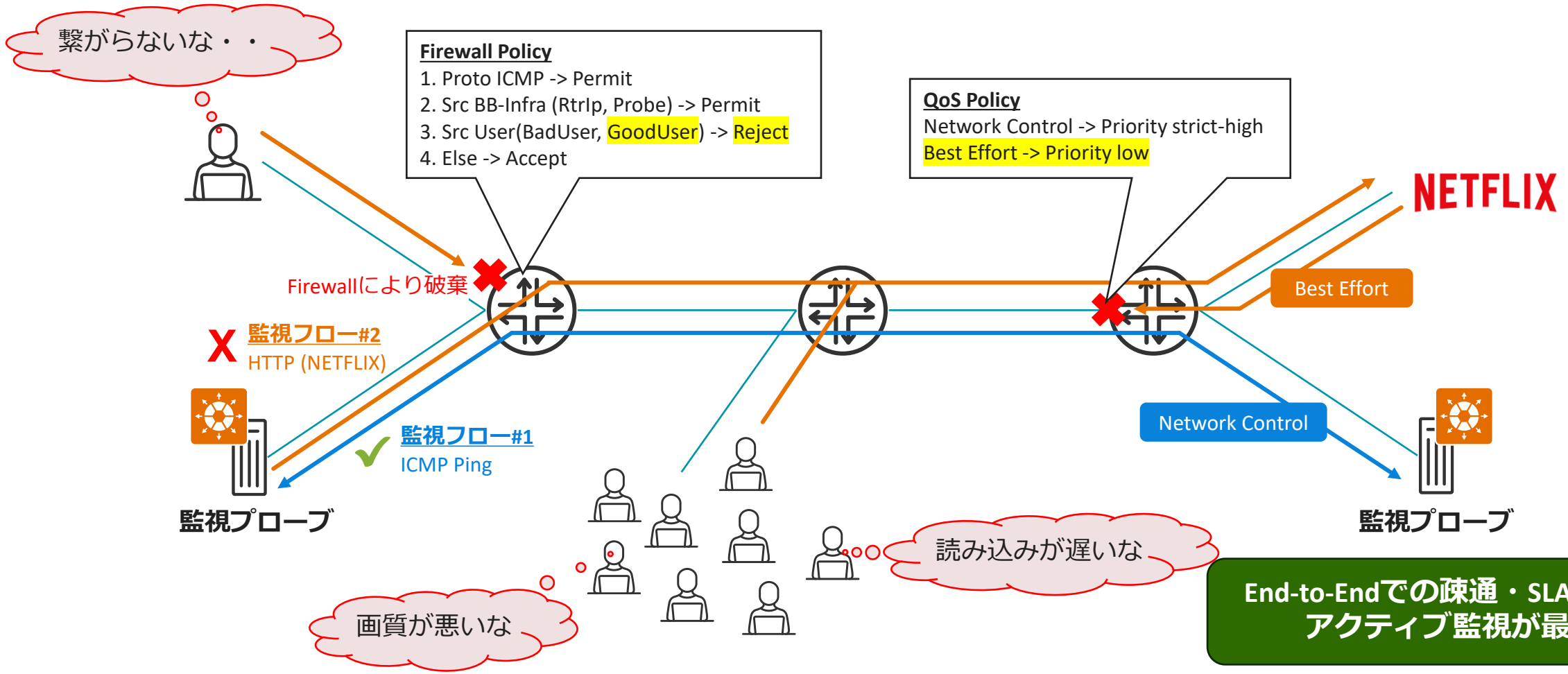


© 2025 Juniper Networks

Juniper Business Use Only

実ユーザに近いトラフィックで監視

ICMPなどは特別扱いされているケースが多く、ユーザ視点での問題に気づけない可能性がある



End-to-Endでの疎通・SLA監視は
アクティブ監視が最適

© 2025 Juniper Networks

Juniper Business Use Only

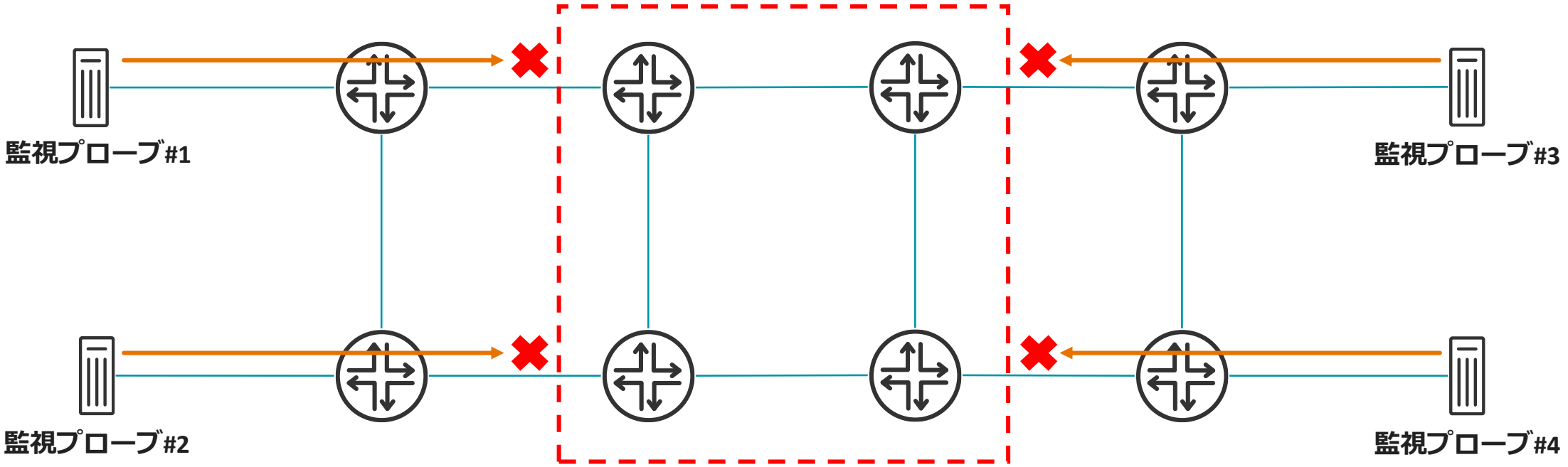
問題個所を特定する方法

アクティブ監視で疎通性やSLAの問題は検知できるものの、問題個所の特定までは中々難しい

何が起きてるんだ？？？

アラート発報
Probe #1 -> #3, Probe #3 -> #2,
Probe #2 -> #4, Probe #4 -> #1,
…

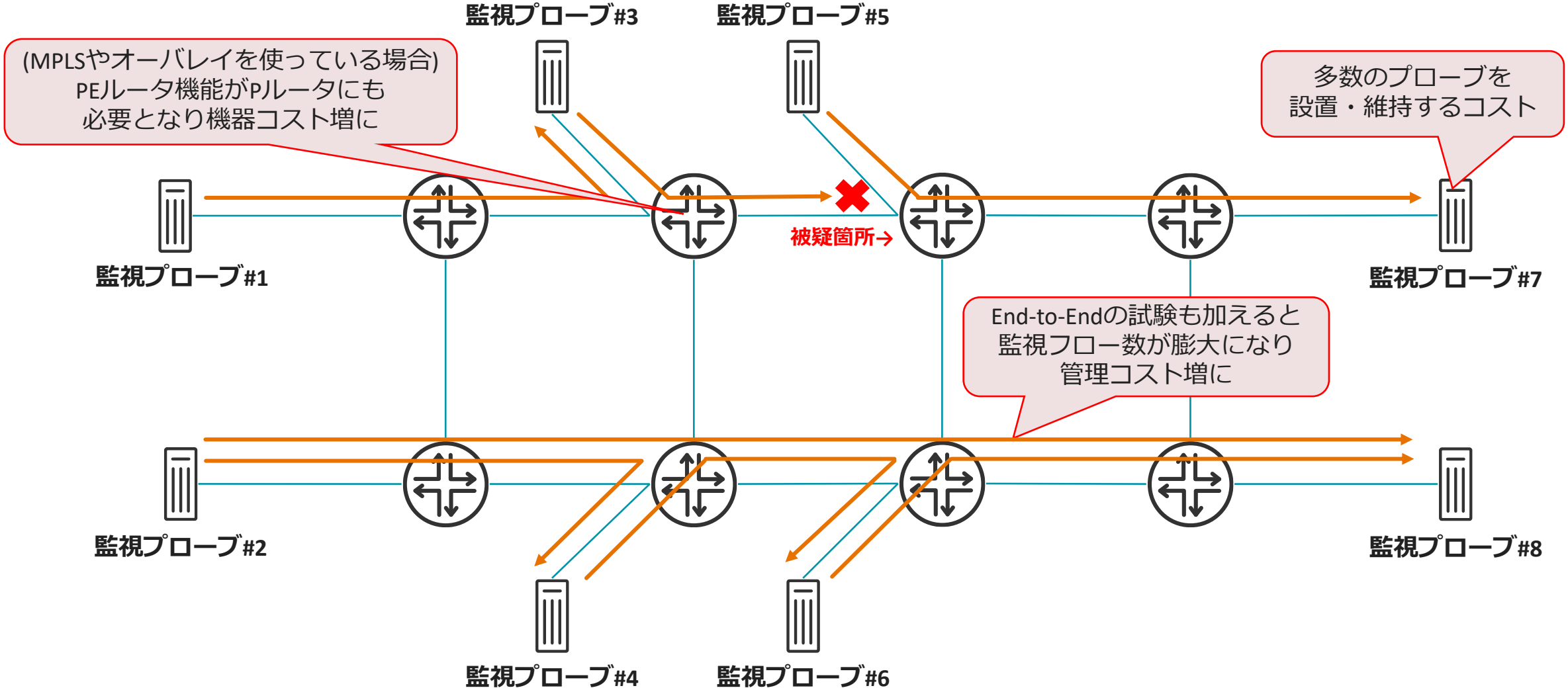
監視ソフトウェア



© 2025 Juniper Networks
Juniper Business Use Only

プローブをばら撒くにも限界が

ネットワークを監視するためのコストに跳ね返るため、全パターンを網羅するには限界がある



© 2025 Juniper Networks

Juniper Business Use Only

機器は破棄した理由を知っている

近年のルータは細かいパケット破棄カウンターや破棄情報を出力する機能を持つものが多い

監視ソフトウェア

警報通知

NOCオペレータ

テレメトリーでの統計値出力

dstip 192.0.2.1

Exception Counter
Discard Route Pkts = 10
TTL Expire Pkts = 12

パケット破棄
Discard Route

L3ループ

パケット破棄
TTL Expire

1412B
Length=14122

パケット破棄
Bad Packet Length

パケット破棄
Unknown Family

Exception Counter
Bad Length Pkts = 21
Unknown Family Pkts = 26

dstip 2001:db8::1

© 2025 Juniper Networks

Juniper Business Use Only

破棄カウンターの例

以下のようにパケットを落とした際の理由と落としたパケットの合計数が記録されている

■ Cisco ASR

```
RP/0/RSP0/CPU0:rasr9000-2w-b#show controllers NP counters np0 location 0/0/CPU0
Tue Dec 10 14:18:39.195 EST
                Node: 0/0/CPU0:
----------------------------------------------------------------

Show global stats counters for NP0, revision v2

Read 59 non-zero NP counters:
Offset  Counter                                        FrameValue   Rate (pps)
--------------------------------------------------------------------------------
  16  MDF_TX_LC_CPU                                      11004363            9
  17  MDF_TX_WIRE                                     8712222364719     29761820
  21  MDF_TX_FABRIC                                  11063035007386     27714366
  29  PARSE_FAB_RECEIVE_CNT                           8712222113330     29761820
  33  PARSE_INTR_RECEIVE_CNT                             9401470            9
  37  PARSE_INJ_RECEIVE_CNT                               832185            1
  41  PARSE_ENET_RECEIVE_CNT                         11070653296959     27714366
  45  PARSE_TM_LOOP_RECEIVE_CNT                          8437075            5
 359  PARSE_MAC_NOTIFY_RCVD                                  183            0
 367  PARSE_FAST_DISCARD_LOW_PRIORITY_DROP_0         106211394050       883832
 368  PARSE_FAST_DISCARD_LOW_PRIORITY_DROP_1         106210662138       883856
 369  PARSE_FAST_DISCARD_LOW_PRIORITY_DROP_2         106211061617       883943
 370  PARSE_FAST_DISCARD_LOW_PRIORITY_DROP_3         106211474043       883922
 373  DBG_RSV_EP_L_RSV_ING_L3_IFIB                     3707021673            0
 830  PUNT_NO_MATCH                                       4746            0
 831  PUNT_NO_MATCH_EXCD                                464963896            0
 849  PUNT_ADJ_EXCD                                       273406            0
 852  PUNT_ACL_DENY                                      1479378            0
 853  PUNT_ACL_DENY_EXCD                              1163570900            0
```

出展: https://www.ciscolive.com/c/dam/r/ciscolive/global-event/docs/2022/pdf/BRKARC-2000.pdf

■ Juniper MX (Juniper Resiliency Interface)

```
SMPC0(mx204 vty)# show jnh 0 exceptions
PFE State Invalid
----------------------
sw error                          DISC( 64)          0          0
invalid fabric token              DISC( 75)          0          0
unknown family                    DISC( 73)   70270275 3169441368
unknown vrf                       DISC( 77)          0          0
iif down                          DISC( 87)       3709     194438
unknown iif                       DISC(  1)   21285498 3156927930
..

Packet Exceptions
----------------------
bad ipv4 hdr checksum             DISC(  2)          0          0
non-IPv4 layer3 tunnel            DISC(  4)          0          0
GRE unsupported flags             DISC(  5)          0          0
tunnel pkt too short              DISC(  6)          0          0
tunnel format error               DISC(  7)          0          0
bad IPv6 options pkt              DISC(  9)        198     129072
bad IPv4 hdr                      DISC( 11)         59       6528
bad IPv6 hdr                      DISC(110)          0          0
bad IPv4 pkt len                  DISC( 12)          2        114
bad IPv6 pkt len                  DISC(111)          2        132
..
```

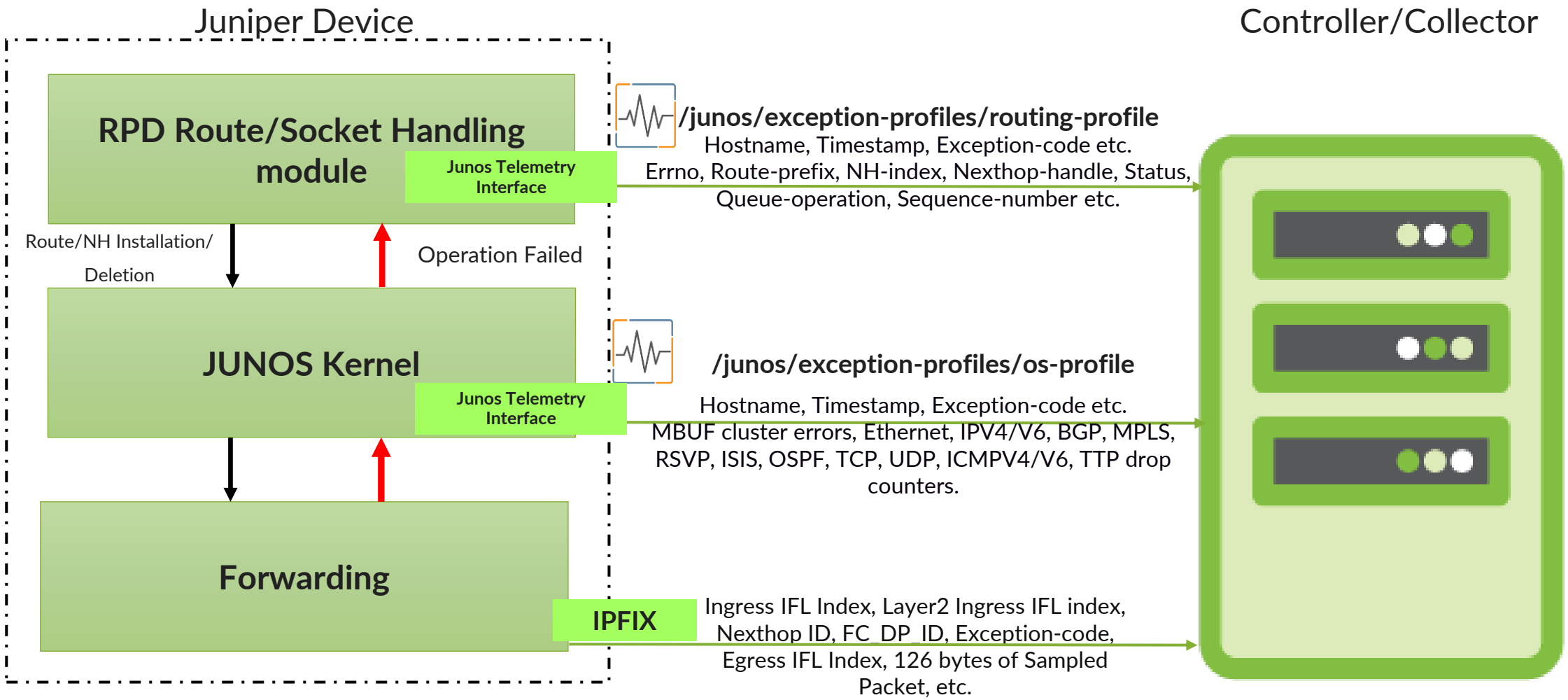
* MX304など新しいソフトウェアモデルを搭載したラインカードではコマンドが異なります
show jnh exceptions level detail inst X <XはPFE ID>

© 2025 Juniper Networks

Juniper Business Use Only

Juniperルータから情報を取得する場合

JuniperルータではTelemetryとIPFIXを使って破棄カウンタの情報を取得することが可能

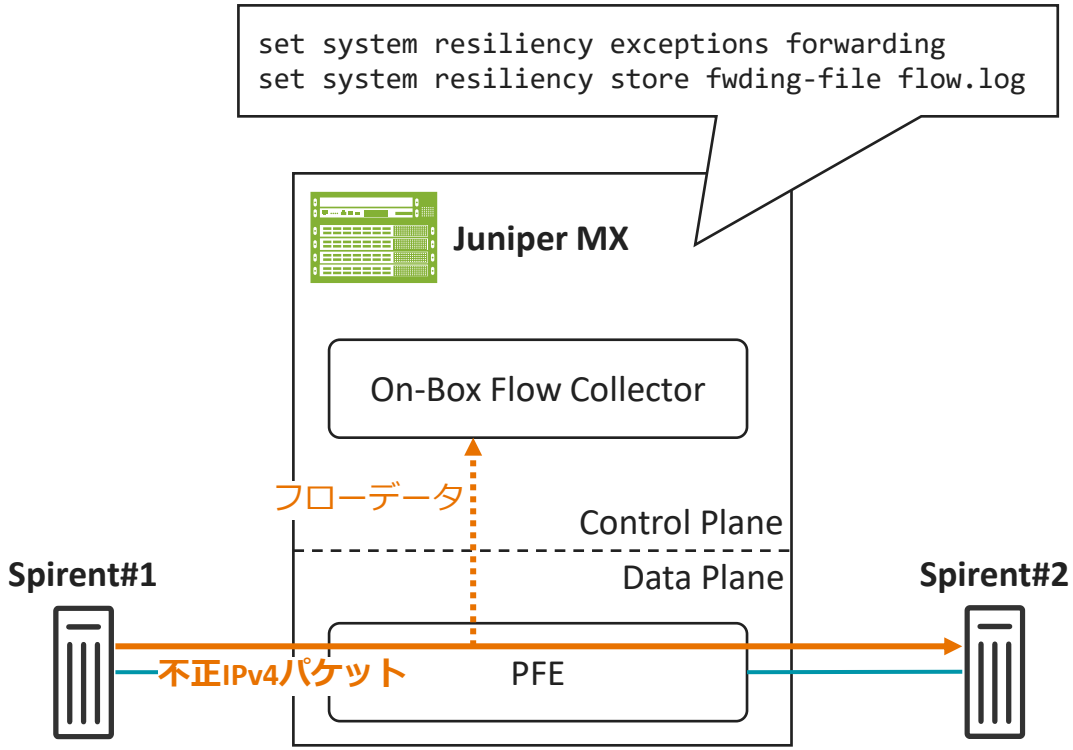


© 2025 Juniper Networks

Juniper Business Use Only

IPFIXのフローデータ取得例

破棄されたパケット数だけでなく、破棄されたパケットのヘッダ情報を取得することが可能




```
lab@mx10004-hatsudai> show log flow.log
***** Netflow/IPFIX ****
Version: 10
Length: 180
Export time: 1753672680
Seq no: 852521
ObservationDomainID: 33619977
setID: 384
setLength: 164
field_type: forwardingClass, field_length: 4
value: 0
field_type: exceptionCode, field_length: 2
value: Bad IPV4 Pkt Len                                  ← 破棄理由
...
field_type: dataLinkFrameSize, field_length: 2
value: 1020
field_type: dataLinkFrameSection, field_length: 126
value: 60c78dd2 19a30010 94000001 8100e064
080045c0 a6b03fbf 0000ff11 5268c0a8                      ← 破棄されたパケット
0002c0a8 01020400 040003d6 b1ee0000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 0000
```

* 設定は以下のコミュニティ記事を参考に
(Junosバージョンによっては FPC再起動が走るため要注意)
https://community.juniper.net/blogs/anton-elita/2024/01/08/packets-lost-in-transit

© 2025 Juniper Networks

フローデータのパケットデータ解読

dataLinkFrameSectionフィールドの中身はEthernetパケットとして解読することができる

■ https://hpd.gasmi.netで解読した結果

- Frame 1: 126 bytes on wire (1008 bits)
- Ethernet II
- 802.1Q Virtual LAN
- Internet Protocol Version 4
  - 0100 .... = Version: 4
  - .... 0101 = Header Length: 20 bytes (5)
  - Differentiated Services Field: 0xc0 (DSCP
  - Total Length: 42672
  - Identification: 0x3fbf (16319)
  - 000. .... = Flags: 0x0
  - ...0 0000 0000 0000 = Fragment Offset: 0
  - Time to Live: 255
  - Protocol: UDP (17)
  - Header Checksum: 0x5268
  - Header checksum status: Unverified
  - Source Address: 192.168.0.2
  - Destination Address: 192.168.1.2
  - Stream index: 0
- User Datagram Protocol
  - Source Port: 1024
  - Destination Port: 1024
  - Length: 982 (bogus, payload length 88)
  - Checksum: 0xb1ee
  - Checksum Status: Unverified
  - Stream index: 0
  - Stream Packet Number: 1
  - Timestamps
  - UDP payload (80 bytes)

IPv4 Length 42672B
=> Bad Length

UDP 982B + IPv4 20B + VLAN 4B + Ether 4B
= Total 1020B

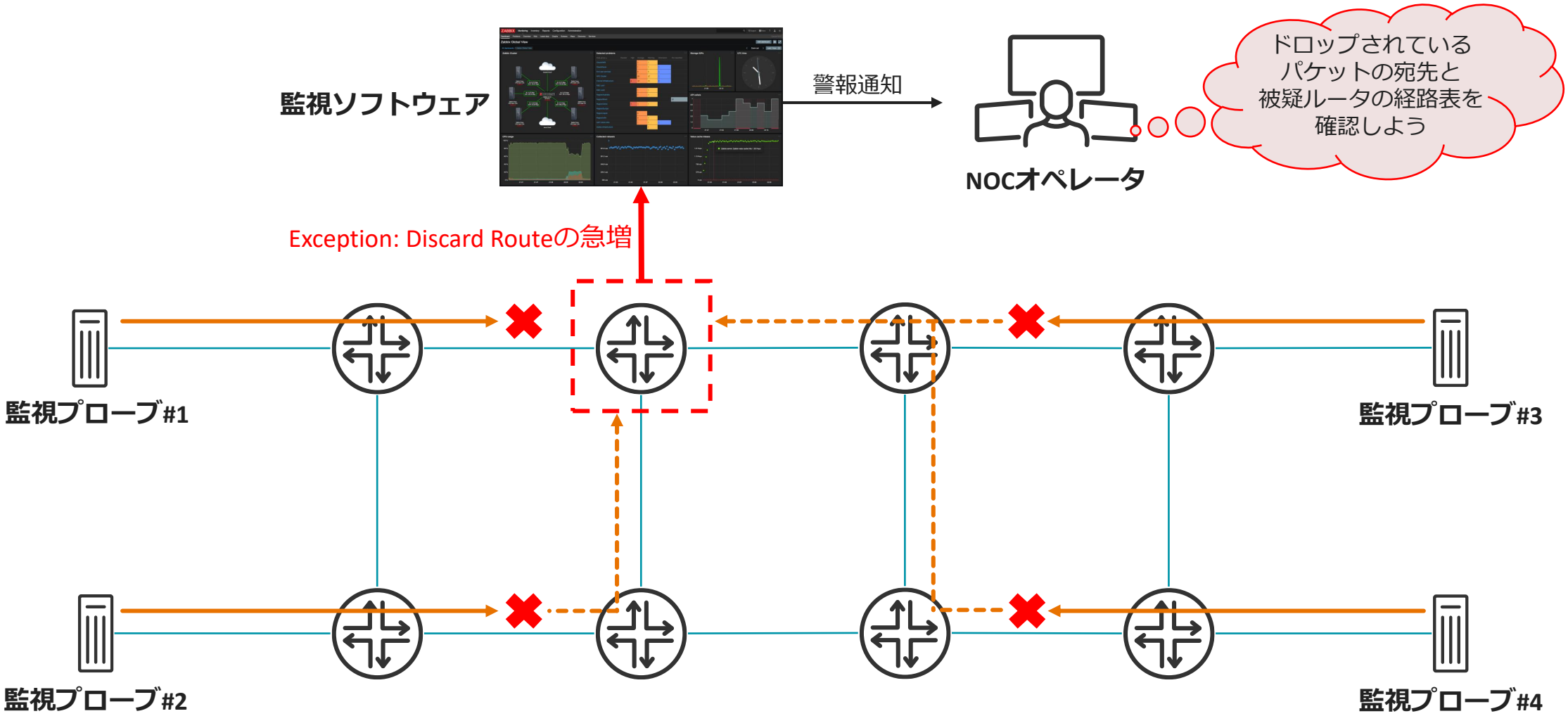
■ IPFIXレコード

```
***** Netflow/IPFIX ****
Version: 10
Length: 180
Export time: 1753672680
Seq no: 852521
ObservationDomainID: 33619977
setID: 384
setLength: 164
field_type: forwardingClass, field_length: 4
value: 0
field_type: exceptionCode, field_length: 2
value: Bad IPV4 Pkt Len
...
field_type: dataLinkFrameSize, field_length: 2
value: 1020
field_type: dataLinkFrameSection, field_length: 126
value: 60c78dd2 19a30010 94000001 8100e064
                                  5268c0a8
                                  01ee0000
                                  00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 0000
```

© 2025 Juniper Networks

破棄カウンタを使った原因箇所特定

破棄カウンタの遷移を監視することで異常個所の特定を多くのケースで迅速化できると考える



© 2025 Juniper Networks

Juniper Business Use Only

破棄カウンタの監視方法

特にインターネットは不正パケットが多く、静的な閾値ベースで監視することは難しいと予想

IX接続ルータにおけるカウンタ例 (一部)

Uptime 約838日
データ取得時の平均流量 約160Mbps

Category	Exception Code	Packets (t1=0s)	Packets (t2=3600s)	Delta
PFE State Invalid	sw error	0	0	0
	unknown family	70,270,275	70,279,564	+9,289
	unknown iif	21,285,498	21,286,860	+1,362
Packet Exceptions	ttl expired	212,676,102	212,683,639	+7,537
	my-mac check failed	23,826,730,473	23,831,520,396	+4,789,923
	my-mac check failed IPv6	304,876,371	304,885,172	+8,801
	DDOS policer violation notifs	37,508	37,510	+2
Firewall	firewall discard	207,854,262	207,877,520	+23,258
	firewall discard V6	59,729,957	59,733,285	+3,328
Routing	discard route	2,309,816,908	2,310,117,787	+300,879
	discard route IPv6	13,632,774,345	13,634,180,110	+1,405,765
	enumcheck mismatch	1,042,873	1,042,882	+9
	reject route	648,395	648,398	+3

ソフトウェアエラーによる破棄
⇒ 0以外は場合は問題あり

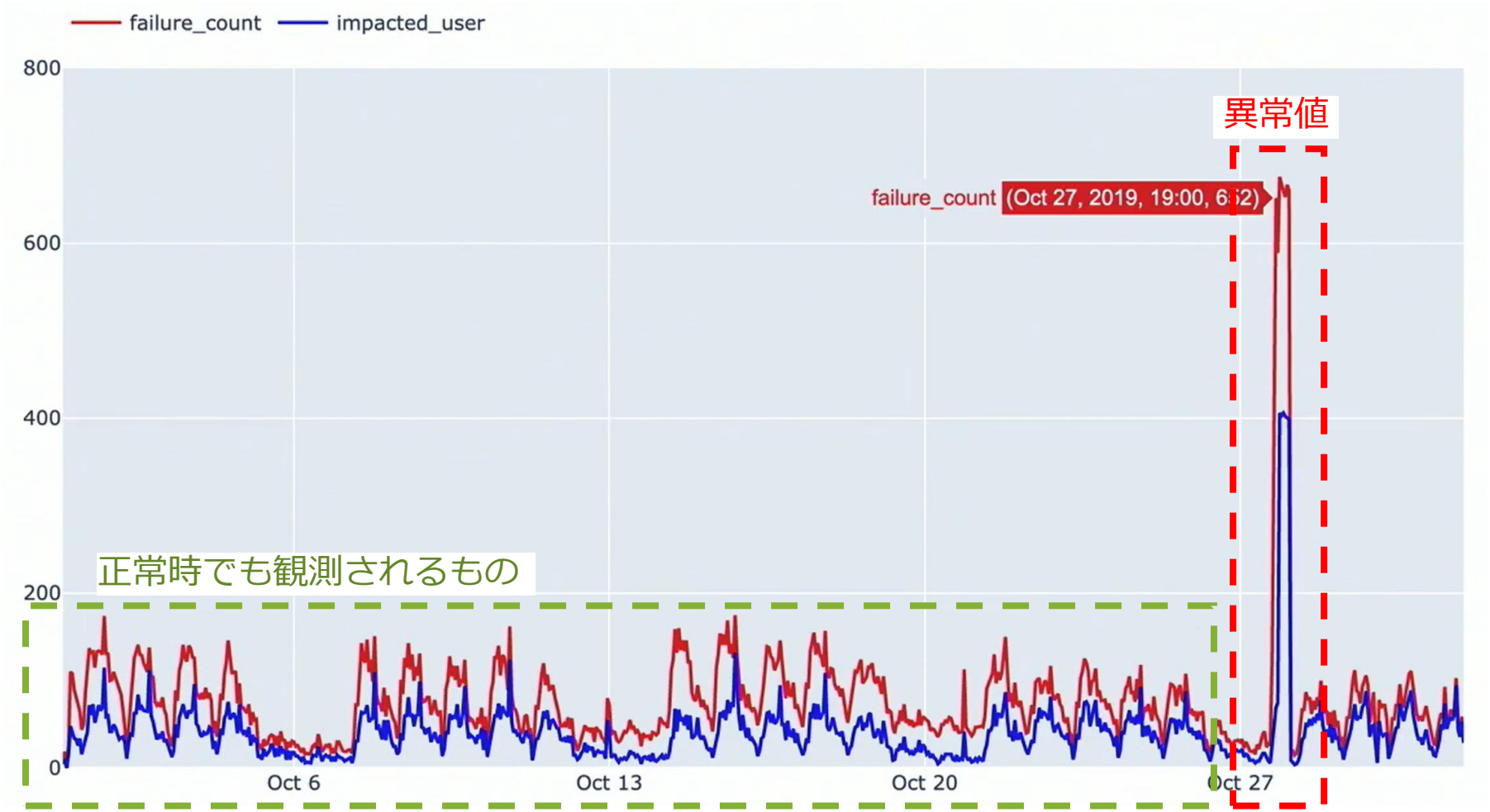
不正パケットなどによる破棄
⇒ 定常的に増えても問題ない可能性が高い

© 2025 Juniper Networks

Juniper Business Use Only

肝心なのはアノマリー

パケット破棄量が周期から外れるようなイベントが発生した場合にアラートを上げる必要がある



© 2025 Juniper Networks

Juniper Business Use Only

AI/ML技術の出番

決められた手順を動かしたり、自ら規則を探して結果を得るのはコンピュータの得意分野

1. 決められた手順で結果を得るもの

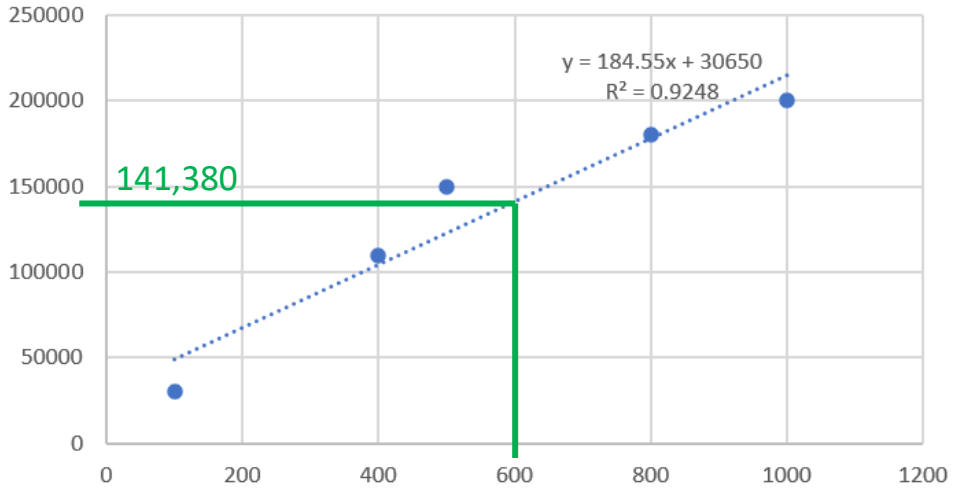
- 人間の手によって結果の求め方を事前に定義
- コンピューターを利用して、
  入力データから決められた手順で結果を算出

例. 来客者に応じた売上高の予測

■ データセット

来客者	売上高
100	30000
500	150000
400	110000
1000	200000
800	180000
600	??

回帰分析

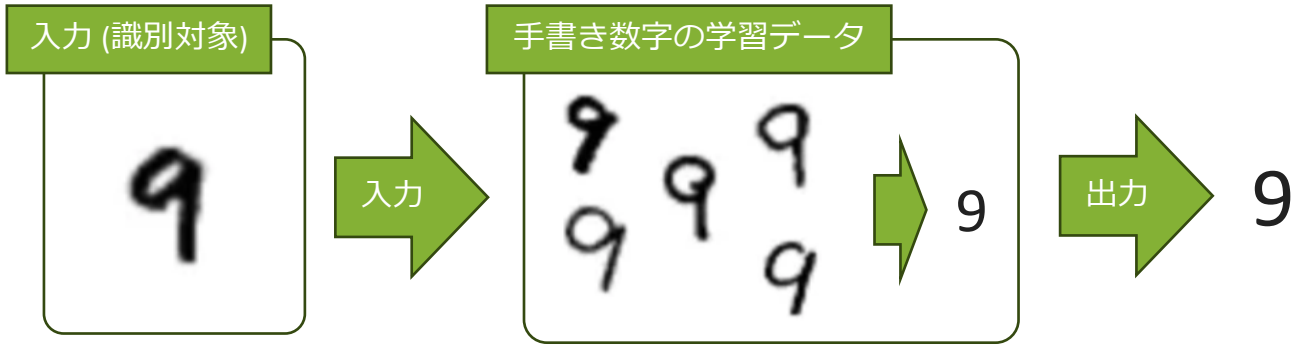


2. 規則性を探し出して結果を得るもの

- 人間の手で結果の求め方を定義できないもの
- 入力データに対する正解データから
  コンピューターに規則性を学習させる
- 学習データした規則性から
  入力データに対する結果を得る

ネットワークの統計データに適用したい

例. 手書きの数字を判別



© 2025 Juniper Networks

Juniper Business Use Only

でも、難しいんでしょ？

便利なライブラリーとクラウドを使うことでNWエンジニアでもAI技術が開発できる時代に

ソフトウェアライブラリー

TensorFlow

Keras

PyTorch

NumPy

クラウド

Google Cloud

aws Azure

Google colab

⇒ GPUリソースも使える
無償のPython開発環境

Google CloudでのGPU時間貸金額

Model	GPUs	GPU memory	GPU price (USD)
NVIDIA T4	1 GPU	16 GB GDDR6	$0.35 per GPU
	2 GPUs	32 GB	

1時間あたり約50円

*2024/09/08時点

Kerasを使ったRNNの学習例

```
from keras.models import Sequential
from keras.optimizers import Adam
from keras.layers import Dense, Activation, SimpleRNN
import numpy as np

x = np.random.randint(0, 2, (128, 5)).astype('float32')
y = np.sum(x, axis=1)

model = Sequential()
model.add(SimpleRNN(16, input_shape=(5, 1)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(x, y, batch_size=10, epochs=10, validation_split=0.1)
```

⇒ 10行程度でRNNの学習が可能に

Untitled0.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
from keras.models import Sequential
from keras.optimizers import Adam
from keras.layers import Dense, Activation, SimpleRNN
import numpy as np

x = np.random.randint(0, 2, (128, 5)).astype('float32')
y = np.sum(x, axis=1)

model = Sequential()
model.add(SimpleRNN(16, input_shape=(5, 1)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(x, y, batch_size=10, epochs=10, validation_split=0.1, verbose=0)

test = np.array([0, 1, 1, 0, 1]).astype('float32').reshape((1, 5, 1))
print(model.predict(test))
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 224ms/step
[[2.506259]]
```

© 2025 Juniper Networks

JUNIPER NETWORKS

最近は監視ツールにもAI機能が

一般的にNOCで使われているツール群にもAnomaly Detection機能が搭載し始めている

Triggers	Key	Interval	History	Trends	Type	Status
Triggers 2	system.cpu.util		7d	365d	Dependent item	Enabled

Severity	Name	Expression	Status
Warning	High CPU utilization (over {$CPU.UTIL.CRIT}% for 5m)	min(/Linux by Zabbix agent active/system.cpu.util,5m)>{$CPU.UTIL.CRIT}	Enabled
Average	CPU usage 2x bigger than in the last 5 seasons(d) (same last 4h)	baselinewma(/Linux by Zabbix agent active/system.cpu.util,4h:now/h,"d",5)*2 < trendavg(/Linux by Zabbix agent active/system.cpu.util,4h:now/h)	Enabled

ZABBIX

Zabbix 6.0.0でアノマリー検知機能を搭載
(2022/02/14)
https://www.zabbix.com/documentation/6.0/jp/manual/introduction/whatsnew600

Grafana

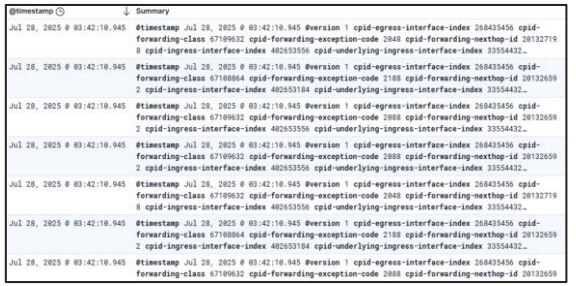
Grafana Machine Learning Toolkitで
外れ値検知 (Outlier Detection)が利用可能に
(2022/12/22)
https://grafana.com/blog/2022/12/22/introducing-outlier-detection-in-grafana-machine-learning-for-grafana-cloud/

© 2025 Juniper Networks

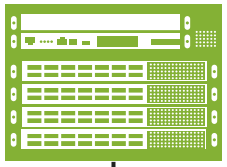
JUNIPER NETWORKS

フローデータのアノマリー検知を行う手法

IPFIXレコードをElasticSearchのAnomaly Detection機能で解析させる



ルータ



1. フローデータ
(IPFIX/UDP)

フローコレクタ　2. JSON変換

GoFlow 2

3. JSON送信

ログプロセッサ

logstash

4. フィールド変換

5. ログ格納

分散データストア・
分析エンジン

elasticsearch

6. データ分析

7. データ取得

フロントエンド

kibana

8. アラート送信

```
{
  "time_flow_start_ns": 1753626294000000000,
  "src_addr": "192.168.0.2",
  "dst_addr": "192.168.1.2",
  ..
  "juniper_properties": [
    67108864,
    2188,
    201326592,
    268435456,
    335544320,
    402653184
  ]
}
```

4. ベンダ固有
フィールドの変換

```
{
  "time_flow_start_ns": 1753626294000000000,
  "src_addr": "192.168.0.2",
  "dst_addr": "192.168.1.2",
  ..
  "cpid-forwarding-class": 67108864,
  "cpid-forwarding-exception-code": 2188,
  "cpid-forwarding-nexthop-id": 201326592,
  "cpid-egress-interface-index": 268435456,
  "cpid-underlying-ingress-interface-index": 335544320,
  "cpid-ingress-interface-index": 402653184
}
```

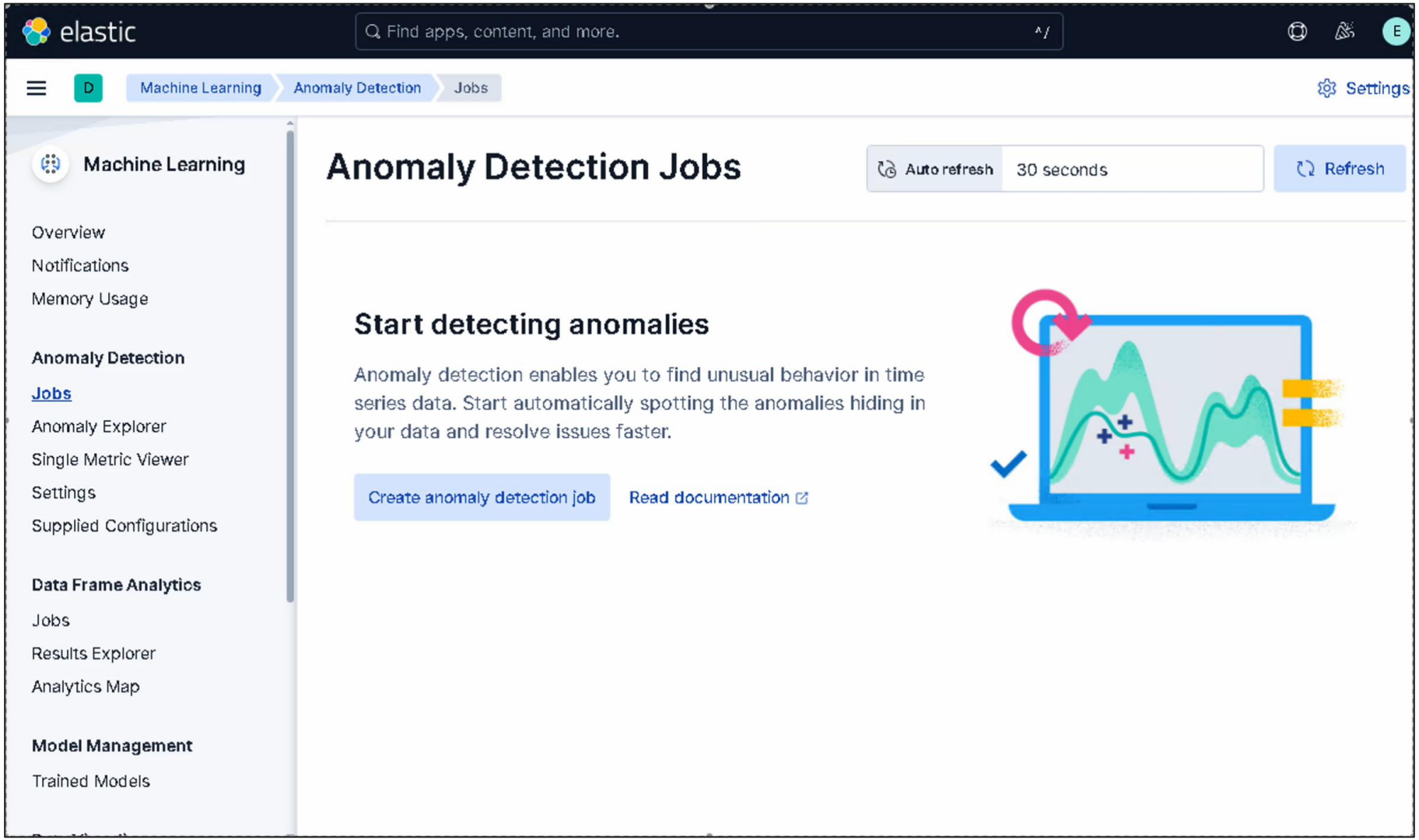
© 2025 Juniper Networks

Juniper Business Use Only

JUNIPER NETWORKS

アノマリー検知の設定

Kibanaのダッシュボードからウィザード形式でパケット破棄傾向のアノマリー検知を設定する

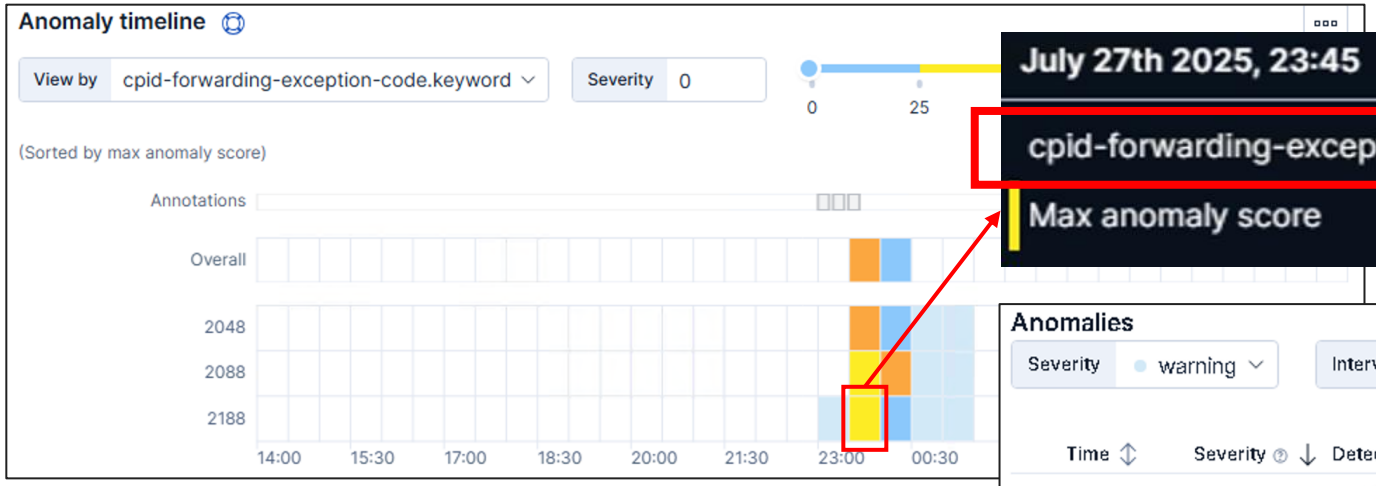


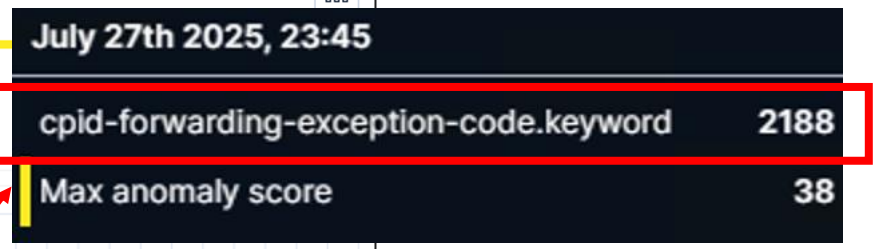
© 2025 Juniper Networks

Juniper Business Use Only

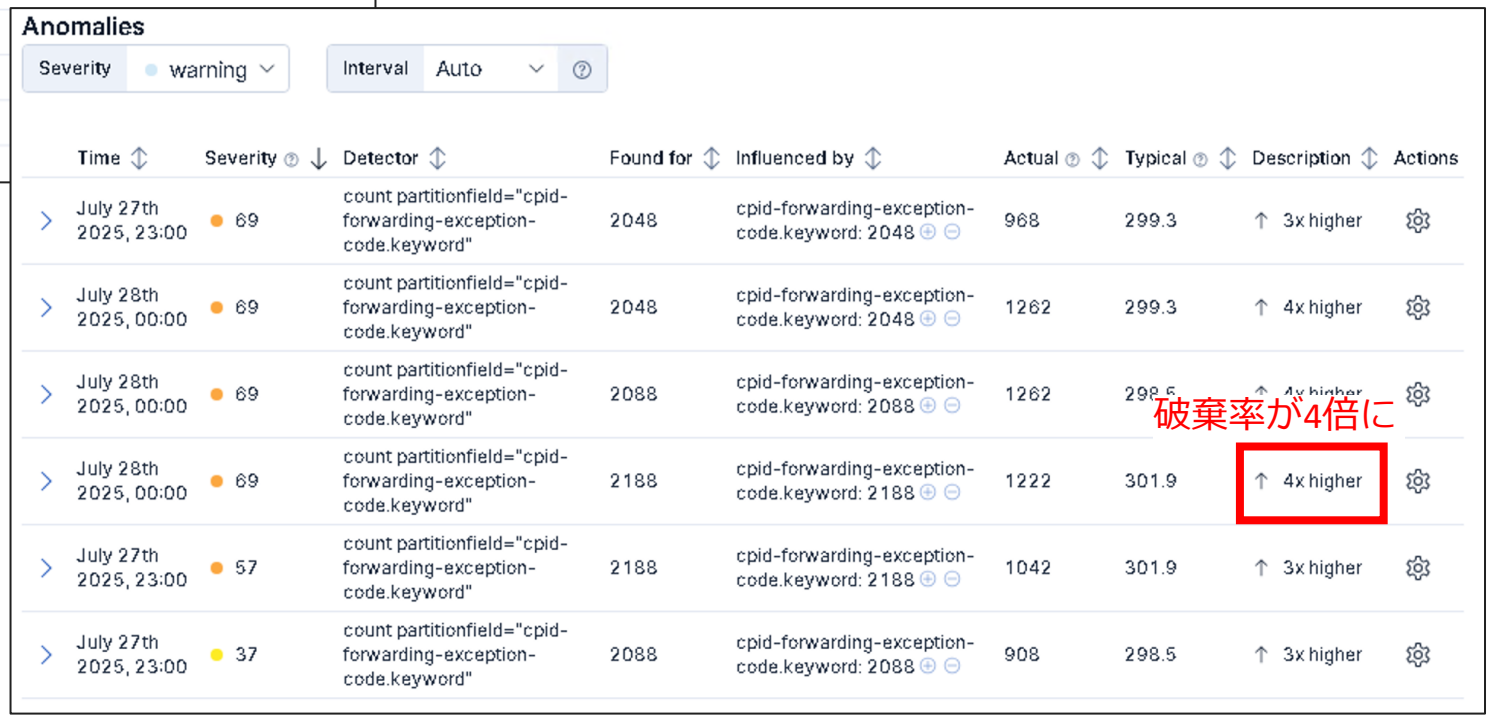
アノマリーの見え方 #1

各エラーコード単位で過去の傾向性から外れたタイミングと判断理由を確認できる





Bad IPv4 Length
(2188 – 2048*1 – 128*2 = Error Code 12)



破棄率が4倍に

*1 2048はCPID Typeフィールドのオフセット処理を省略したため

```
Observation Cloud Level Juniper Common Properties: Forwarding Exception Details
    0000 10.. .... .... = Juniper CPID Type: Forwarding Exception Details (0x02)
    .... ..00 1000 1100 = Juniper CPID Value: 140
```

*2 一部のエラーコードは128を引く必要がる

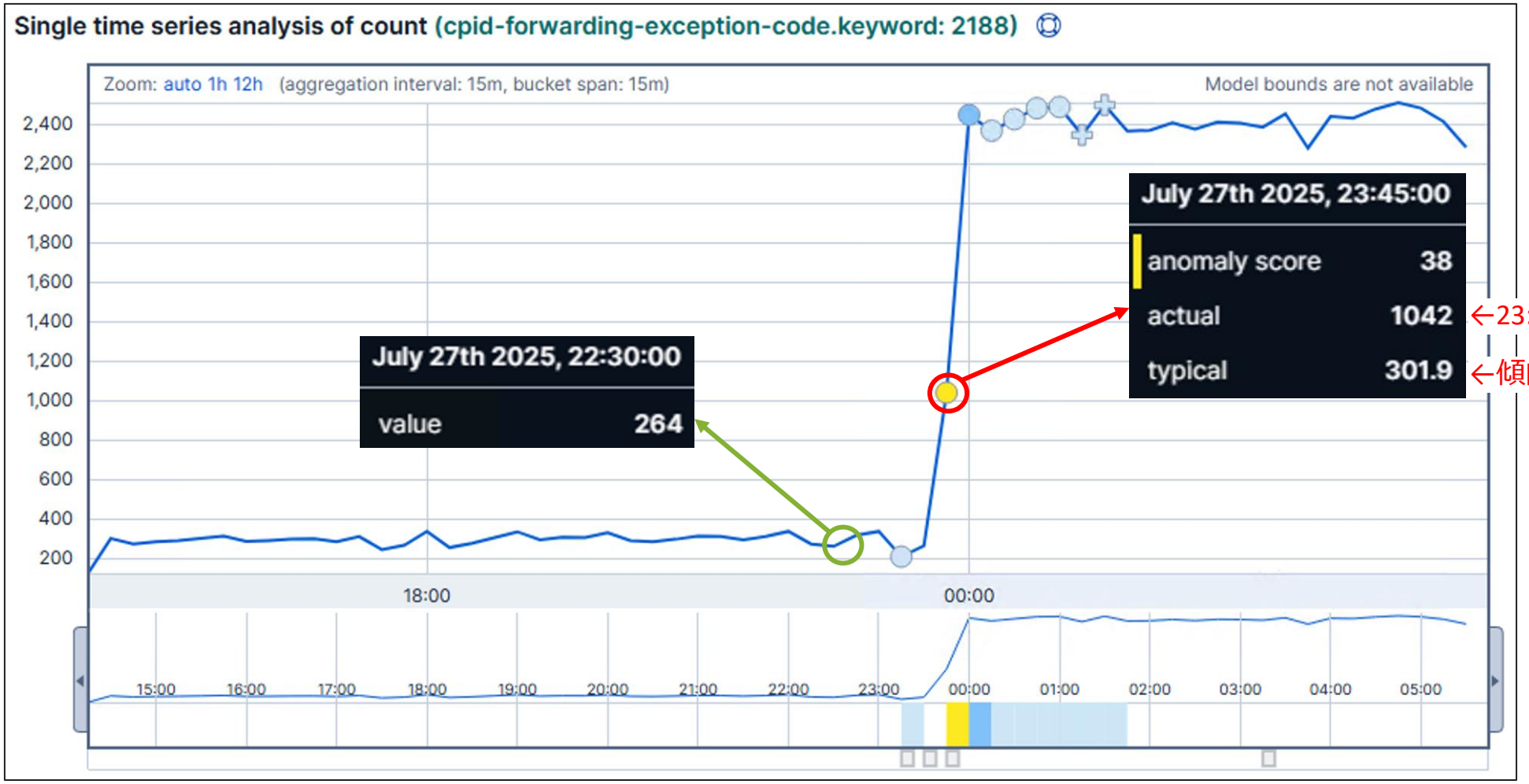
“From an external viewpoint, when the exception is reported, the actual code sent is 128 + *value*. ”
https://www.juniper.net/documentation/us/en/software/junos/flow-monitoring/topics/topic-map/resiliency-exception-reporting.html

© 2025 Juniper Networks

Juniper Business Use Only

アノマリーの見え方 #2

アノマリーが検知されているエラーコードに対してドリルダウンすることで傾向を確認できる



←23:45時点で1042

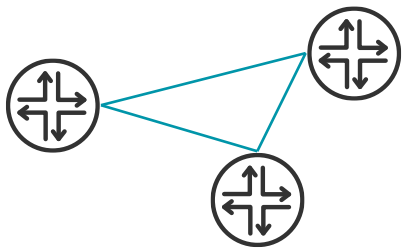
←傾向では301.9前後

© 2025 Juniper Networks

Juniper Business Use Only

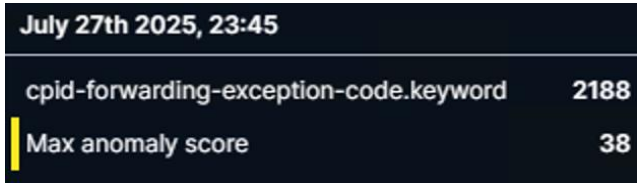
根本原因の特定

パケット破棄情報と他のイベント情報の相関関係分析を行うことで根本原因の特定に近づく



SYSLOG, SNMP Trap,
BGP Monitoring Protocol
(BMP)

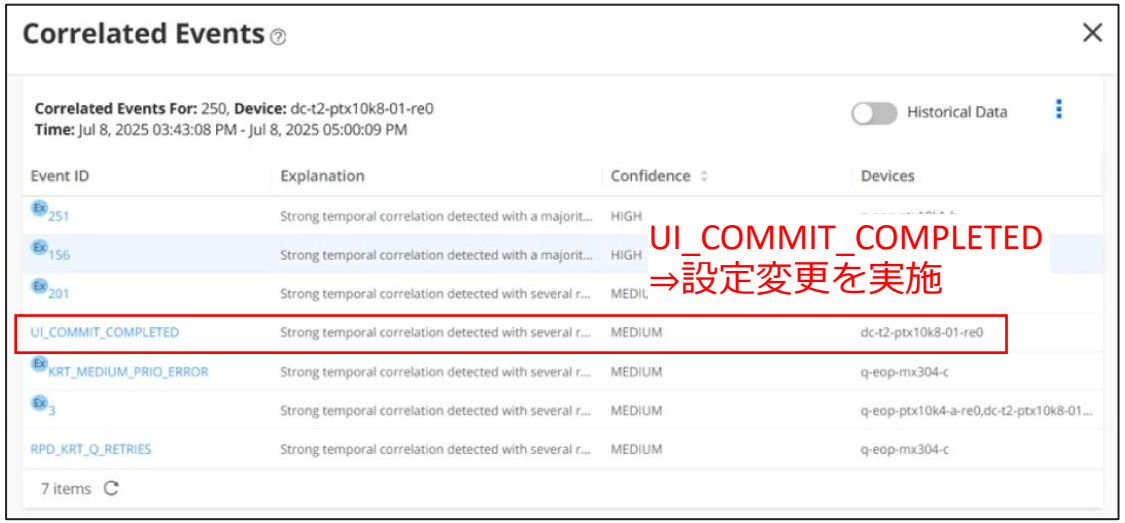
傾向性変動のアラート発報



ナレッジベース

- SYSLOGメッセージ
- 経路アップデート
- SNMPトラップ
- ⋮

発生時刻の近いイベントとの相関関係分析



UI_COMMIT_COMPLETED
⇒設定変更を実施

- ナレッジベースの在り方は？
- イベントと事象に対する関連性の紐づけは？
- データ量をどう集めるか？

© 2025 Juniper Networks

Juniper Business Use Only

LLMの活用は？

LLMへの期待が高まる一方、ネットワーク運用へのLLM適用には出力の信憑性に課題がある

1. ハルシネーション *1, 2

- 事実の不整合や捏造 (Factuality Hallucination)、指示や文脈、論理的な不整合 (Faithfulness Hallucination)がLLMでは多く発生
- 特定の技術分野に関する深い知識や最新の情報が欠落していることから出力が汚染される
- LLMは自身の知識境界を認識できないため、誤った情報を生成してしまうことがある

2. バイアス *3

- 多く学習された内容に沿った回答を出力を好む傾向がある

3. 一貫性の無さ *3

- プロンプトの表現によって全く異なる回答が出てくることもある

- 既存の手法と比較して高い精度でのデータ分析をLLMでは実現できない
- LLMの出力が正しいことを確認する手段が必要

[参考文献]
*1: Huang, Lei, et al. "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions." arXiv preprint arXiv:2311.05232 (2023).
*2: Wayne Xin Zhao, et al. “A Survey of Large Language Models” arXiv preprint arXiv:2303.18223 (2023).
*3: Rickard Stureborg, et al. “Large Language Models are Inconsistent and Biased Evaluators” arXiv:2405.01724 (2024).

© 2025 Juniper Networks

信憑性を向上させるためには

Fine TuningやRAGなどの手法でLLMの知識を強化することで信憑性は向上すると考えられる

Table 1. Results for the MMLU datasets described in Section 4.1 in terms of log-likelihood accuracy (Equation (4)).

Task	Model	Base model	Base model + RAG	Fine-tuned	Fine-tuned + RAG
Anatomy (0-shot)	Mistral 7B	0.556	**0.681**	0.570	0.659
	Llama2 7B	0.393	**0.489**	0.430	**0.489**
	Orca2 7B	0.607	**0.637**	0.600	**0.637**
Anatomy (5-shot)	Mistral 7B	0.600	**0.681**	0.622	0.674
	Llama2 7B	0.467	**0.563**	0.496	0.548
	Orca2 7B	0.570	0.659	0.593	**0.674**
Astronomy (0-shot)	Mistral 7B	0.625	0.678	0.651	**0.697**
	Llama2 7B	0.401	0.467	0.487	**0.520**
	Orca2 7B	0.645	**0.750**	0.651	**0.750**
Astronomy (5-shot)	Mistral 7B	0.658	**0.724**	0.651	0.697
	Llama2 7B	0.401	0.474	0.447	**0.520**
	Orca2 7B	0.664	**0.763**	0.664	0.743
College biology (0-shot)	Mistral 7B	0.681	0.757	0.701	**0.764**
	Llama2 7B	0.438	**0.493**	0.458	0.465
	Orca2 7B	0.583	**0.639**	0.604	0.632
College biology (5-shot)	Mistral 7B	0.722	**0.778**	0.736	0.771
	Llama2 7B	0.451	**0.521**	0.424	0.479
	Orca2 7B	0.604	**0.660**	0.625	0.653
College chemistry (0-shot)	Mistral 7B	0.470	**0.500**	0.490	**0.500**
	Llama2 7B	0.310	0.380	0.390	**0.390**
	Orca2 7B	0.370	**0.440**	0.370	0.390
College chemistry (5-shot)	Mistral 7B	0.470	**0.540**	0.500	0.500
	Llama2 7B	0.370	0.380	0.360	**0.390**
	Orca2 7B	0.430	**0.470**	0.370	0.380
Prehistory (0-shot)	Mistral 7B	0.713	**0.750**	0.719	0.731
	Llama2 7B	0.448	**0.481**	0.457	0.478
	Orca2 7B	0.642	**0.679**	0.673	0.673
Prehistory (5-shot)	Mistral 7B	0.722	**0.762**	0.725	**0.762**
	Llama2 7B	0.515	0.531	0.503	**0.537**
	Orca2 7B	0.664	**0.698**	0.667	0.694

RAGやFine Tuningより
出力の精度が向上

本手法によるネットワーク運用向けの汎用LLM開発も進んでいる

Zou, Hang, et al. "TelecomGPT: A framework to build telecom-specfic large language models." *arXiv preprint arXiv:2407.09424* (2024).

Kan, Khen Bo, et al. "Mobile-LLaMA: Instruction Fine-Tuning Open-Source LLM for Network Analysis in 5G Networks." *IEEE Network* (2024).

しかし、それでも道のりは長い・・

1. 学習データをどのように入手するか?

機器固有の情報 (出てくる統計値やログ、設定例など) 、トラブルシューティングのノウハウ (分析手順や復旧方法など)はどこから入手する?

2. 汎用LLMを運用できるか?

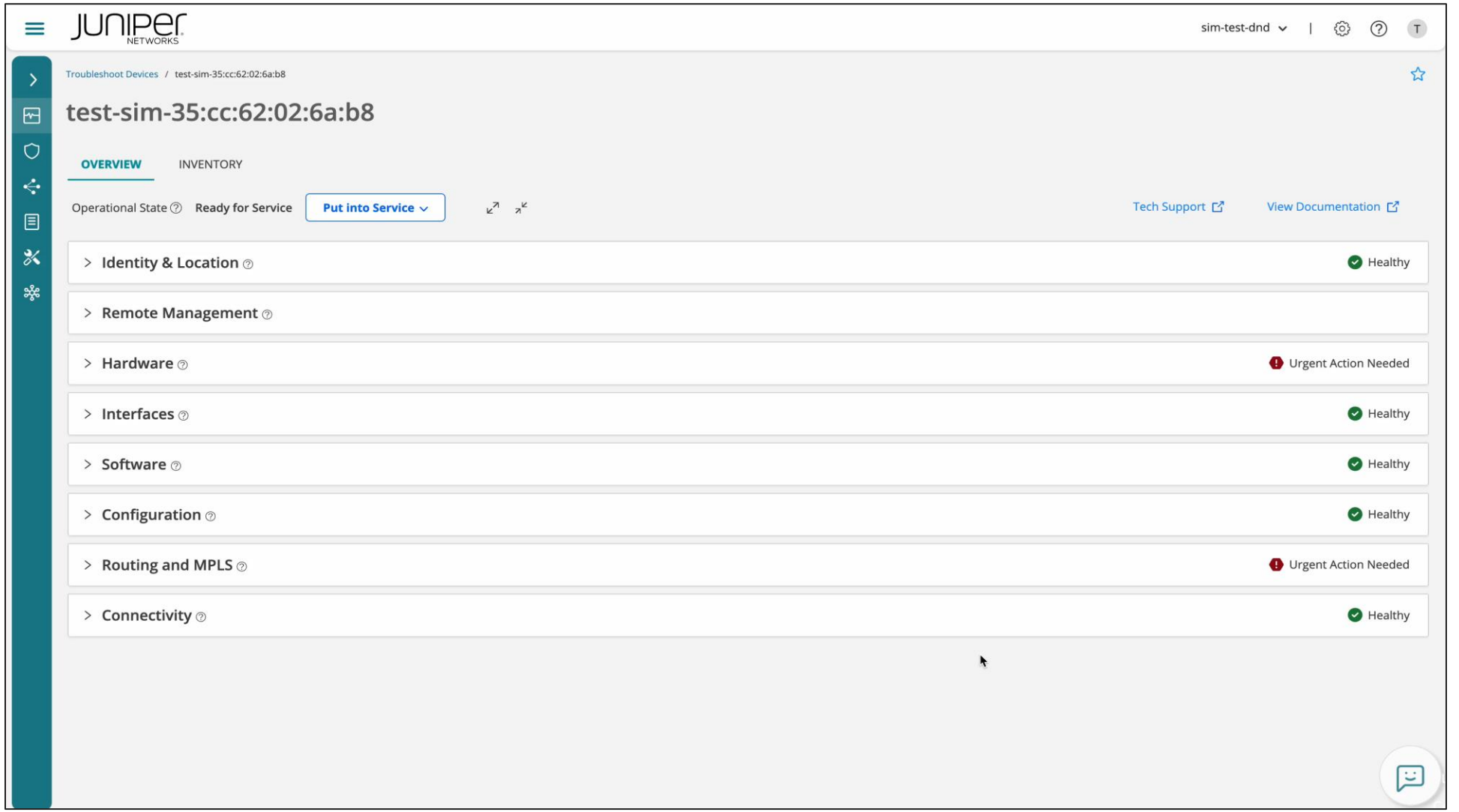
Self TuningやRAGを行うためのプログラム開発は誰がやるのか?
LLMの推論処理はどこでやるのか? コストに見合うのか?
過学習 *1が発生しないようにするためにはどうすれば良いのか?

出展: Ovadia, Oded, et al. “Fine-tuning or retrieval? comparing knowledge injection in llms.” *arXiv preprint arXiv:2312.05934* (2023).
参考文献: *1 Gekhman, Zorik, et al. "Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations?." arXiv preprint arXiv:2405.05904 (2024).

© 2025 Juniper Networks
Juniper Business Use Only

アシスタントとしての導入は始まっている

補助ツールとしてLLMの利用は十分可能であり、既にパッケージ製品では導入が始まっている



© 2025 Juniper Networks

Juniper Business Use Only

まとめ

- ネットワークの監視には「点の監視」と「面の監視」が必要
- 一般的なSNMPやPingなどを使った監視手法に加えて、今では様々な監視手法が登場している
  - アクティブ監視、テレメトリー (JRI)など
- AI/ML技術が使いやすくなりネットワークの現場でも活躍する場面が増えると予想される
- 今後は根本原因の特定自動化やLLM活用が期待される

ディスカッショントピック

- ネットワークの面としての監視はやってますか？どうやってますか？
- ネットワーク運用 (監視)へのAI/ML技術適用は検討されてますか？

© 2025 Juniper Networks
Juniper Business Use Only

Thank you

JUNIPER NETWORKS