

Dynamic Network Defense: Automating Traffic Engineering based on ThousandEyes Insights

cisco Live !

Kemal Šanjta
Principal Internet Analyst

Mike Hicks
Principal Solutions Analyst

Cisco Webex App

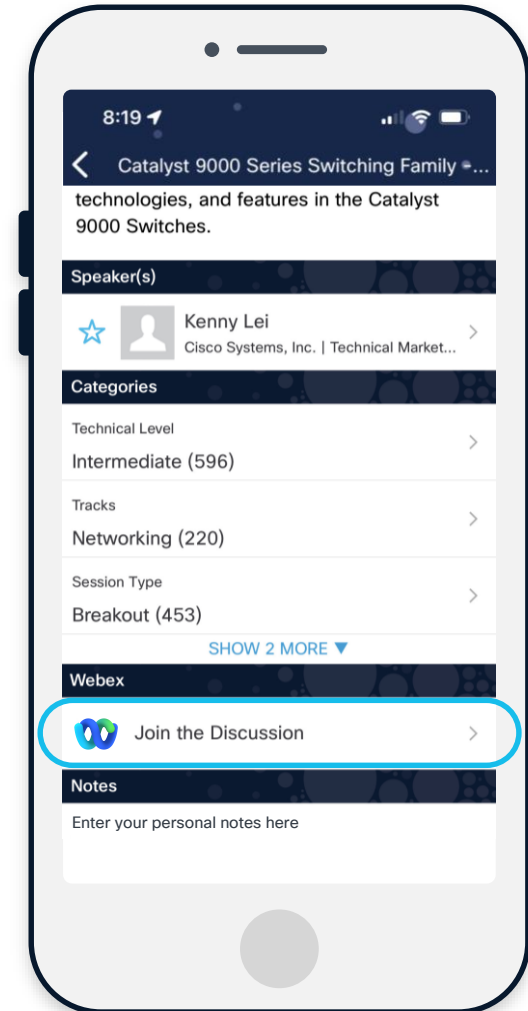
Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 13, 2025.



<https://ciscolive.ciscoevents.com/ciscolivebot/#BRKOPS-1009>

Agenda

- 01 Automation
- 02 Dynamic Network Defense
- 03 Dealing with unexpected
- 04 Automating DDOS Remediation
- 05 Automating BGP Hijack Remediation
- 06 Conclusion

By way of Introduction...

Kemal is a result-oriented engineer focusing on **designing, operating and troubleshooting large-scale networks**. I am a passionate Linux user with a deep understanding of SRE/NRE practices, and have been with Cisco since 2018.

Over the last two decades, I have worked at several large-scale companies applying NRE practices and automating remediation actions.

As a Principal Internet Analyst at ThousandEyes, I **focus on research and providing deep and meaningful insights** into outages through the lenses of ThousandEyes.

Kemal Sanjta
Principal Internet Analyst
kemals@cisco.com



By way of Introduction...

Mike Hicks is a **recognized expert in network and application performance**, with more than 30 years of industry experience **supporting large, complex networks** and working closely with infrastructure vendors on application profiling and management.

He is the author of "**Managing Distributed Applications: Troubleshooting in a Heterogeneous Environment**" (Prentice Hall 2000) and "**Optimising Applications on Cisco Networks**" (Cisco Press 2004).

Mike Hicks

Principal Solutions Analyst
mhicks@cisco.com





The art of possible

Automation

- **Automation** is the use of systems, software, or machines to perform tasks with minimal or no human intervention
- Provides **solution to repetitive tasks**, efficiency, accuracy and scalability
- In context of networking, we are speaking about automating configuration, management, testing, deployment and operations

Why?

- **Boosts efficiency:** Replacing repetitive, manual tasks with faster and consistent processes
- **Reduces errors:** Minimizing human errors, especially in high volume and precision critical tasks
- **Lower operational costs:** Reduced labor and operational costs
- **Scale improvements:** Automating repeatable tasks frees up engineering hours for more important tasks
- **Speed and Responsiveness:** Real time processing and quicker standardized decision making
- **Consistency and Quality improvements:** Automated workflows lead to more predictable outcomes

What to automate?

Routine, repetitive tasks

Automate well understood, routine, repetitive, simple tasks with minimal risk. Building trust in the process is one of the key steps before moving towards more complex and demanding tasks.

Other automation candidates

Device configuration management, backup and restore configurations, code upgrades, new device provisioning, source of truth

What is out there?

Tools

Ansible

- Agentless automation using YAML based templates
- Excellent for network device configuration (all major vendors are supported)
- Large community
- Simple

Python (Netmiko / NAPALM)

- Netmiko is SSH lib tailored for network devices
- NAPALM provides multi vendor abstraction layer for config and state collection
- Flexibility and ease of use with Python

Terraform

- Infrastructure as a Code (IaC)
- Different, major providers are available
- Offers scalability

Salt

- Event driven Automation and config management
- Supports both servers and network devices
- Scalable

Frameworks

Cisco NSO (Network Services Orchestrator)

- Multivendor service orchestration platform
- Uses model driven architecture (YANG)
- Offers deep integration with Cisco and third-party hardware

Nornir

- Python based, plugin driven framework for network automation
- No YAML abstraction
- Offers more flexibility
- Useful for inventory management, data collection, config deployment

Cisco PyATS / Genie

- Automated network testing and validation
- Offers parsing libs, test case generation and rich CLI integration
- Extensible (but vendor specific)
- Useful for network validation, regression testing and automated troubleshooting

Glance / Nautobot

- Source of truth and automation frameworks
- Support for workflows, jobs and webhooks
- Integrates with Nornir and Ansible
- Useful for automation using single source of truth

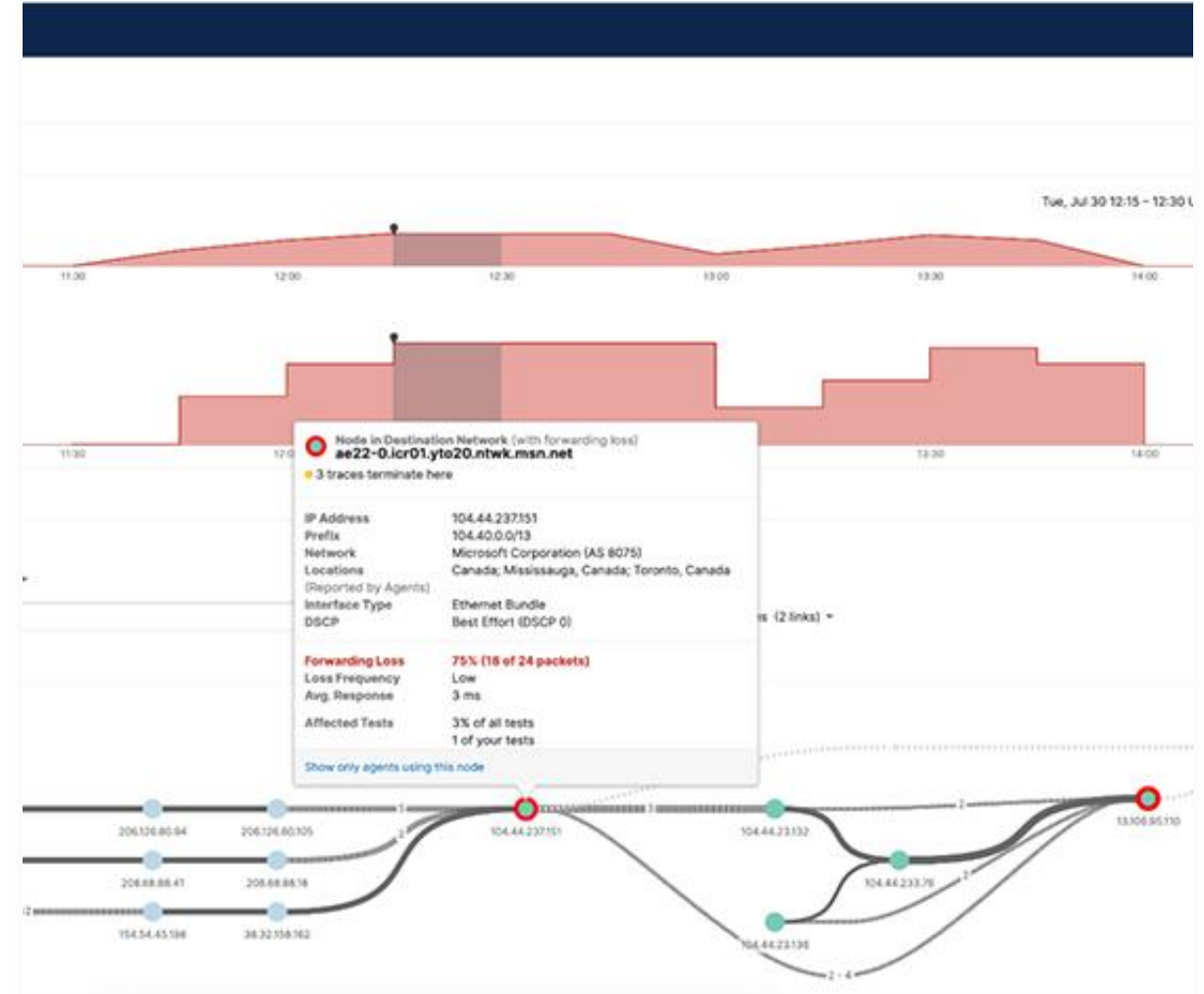


“The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency.”

Bill Gates

When Automation bites back

- Automation triggered by condition
- Multiple disparate overlapping automated processes occurring
- Unexpected signals trigger inappropriate automated response
- Increased user impact





Dynamic Network Defense



- **Automation tool:** Open-source automation tool used for configuration management, app deployment and task automation
- **Agentless architecture:** It doesn't require any agent to be installed on managed nodes (utilizes SSH to execute tasks)
- **Plays and playbooks:** Tasks and configuration are defined in simple YAML files called playbooks which are easy to read and write
- **Idempotent Operations:** It ensures that repeated executions produce the same result without unintended changes
- **Modules and roles:** It uses modules to perform tasks and roles to organize related playbooks, variables and files for reuse and clarity
- **Inventory:** Hosts / groups of hosts are listed in inventory file which can be statically or dynamically generated
- **Security:** Vault ensures that sensitive information such as passwords can be encrypted

Ansible Building Blocks

Inventory

```
% cat hosts
[eastus1]
mkt-aws-cpr-eastus1-az[1:3]

[eastus2]
mkt-aws-cpr-eastus2-az[1:3]

[aws:children]
eastus1
eastus2

[aws:vars]
ansible_become=True
ansible_user=ubuntu
```

Play

```
tasks:
  - name: Install apache2
    package:
      name: apache2
      state: present

  - name: Copy agent installation file
    copy:
      dest: /home/ubuntu/osquery.deb
      src: ../osquery/osquery.deb
```

Playbook

```
---
-
  hosts:
    aws

  vars:

  tasks:
    - name: Install dpkg-sig
      package:
        name: dpkg-sig
        state: present

    - name: Copy agent installation file
      copy:
        dest: /home/ubuntu/osquery.deb
        src: ../osquery/osquery.deb

...
```

Event-Driven Ansible (EDA)

- Framework that allows automation of IT operations in response to the events in real time
- Listens for external events (alerts, log changes, webhook triggers, API calls) and then automatically executes Ansible playbooks or rulebooks based on defined conditions
- Rulebooks define event sources, conditions and actions
- Supports multiple different event sources such as webhooks, Splunk, ServiceNow, Kafka streams, Prometheus alerts, etc.
- Speeds up incident response

Event Driven Ansible Building Blocks

Rulebook

```
---
- name: Receive and print ThousandEyes webhook
  events
    hosts: all
    sources:
      - ansible.eda.webhook:
        host: 0.0.0.0
        port: 8080

    rules:
      - name: Execute traffic engineering to switchover to
        standby ISP
        condition: event.payload.type == "2"
        actions:
          - run_playbook:
            name: ./switchover.yaml
...

```

Playbook 1/2

```
---
-
  hosts: r1

  vars:
    ansible_connection:
    ansible.netcommon.network_cli
    ansible_network_os: cisco.ios.ios
    ansible_become: yes
    ansible_become_method: enable

```

Playbook 2/2

```
tasks:
  - name: Merge provided configuration with
    device configuration
    cisco.ios.ios_bgp_global:
      config:
        as_number: 32100

      neighbor:
        - neighbor_address: 100.100.20.10
          description: Primary transit
          remote_as: 700
          soft_reconfiguration: true
          fall_over:
            bfd:
              set: true
            route_maps:
        - name: FORCE-STANDBY-ISP-STATUS-OUT
          out: true
        - name: FORCE-STANDBY-ISP-STATUS-IN
          in: true
...

```

Demo

CISCO ThousandEyes



- **Network Intelligence Platform:** provides visibility into network performance of owned and unowned networks
- **Synthetic Monitoring:** Utilizes concept of the synthetic traffic that is sharing same data plane production/user traffic is taking
- **End to end monitoring:** Measures digital experience from different vantage points through Enterprise, Cloud and Endpoint agents
- **Path Visualization:** Provides detailed path trace visualization that shows each network hop and helps narrow down where the problems are happening, when they are happening, ultimately shortening Mean Time to Resolution
- **Comprehensive alerting** and dashboarding capabilities

ThousandEyes Test Building Blocks



Test target



Test interval



Testing agents

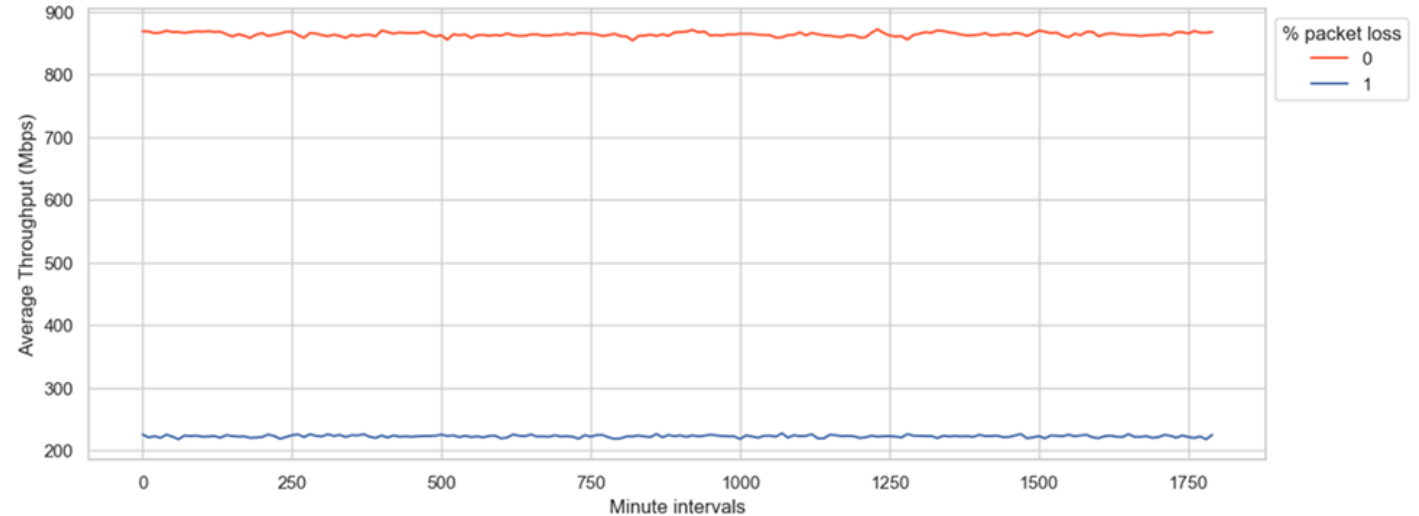
Demo



Dealing with unexpected

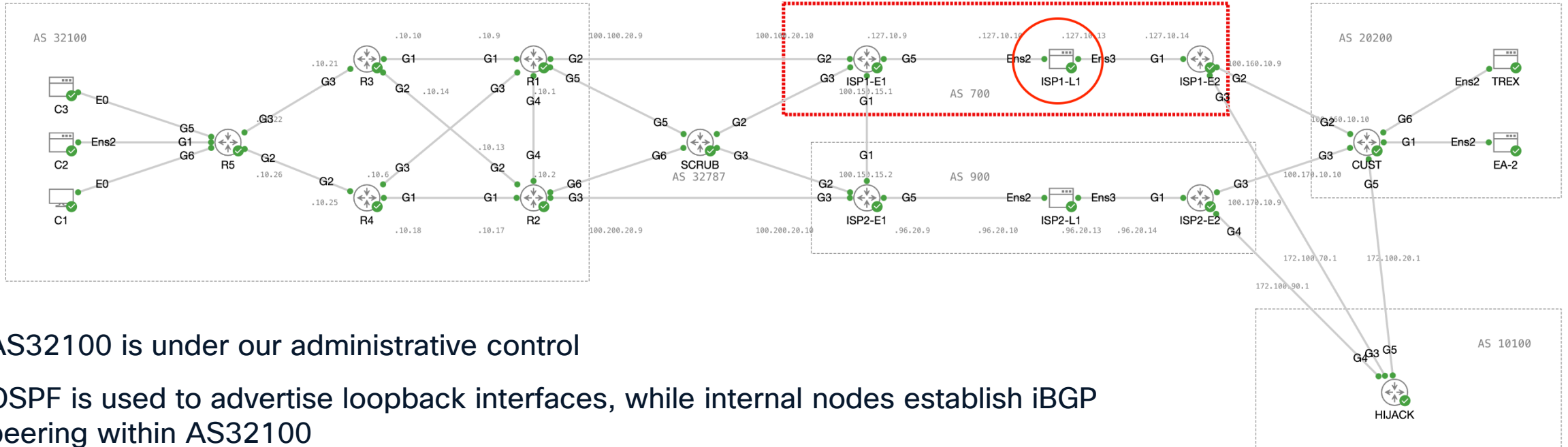
Packet loss?

- Inability of packets traveling across the network to reach their destination
- Root causes often revolve around:
 - Congestion
 - Signal degradation
 - Hardware or software issues
 - Routing related issues
 - Overflows
 - Security misconfigurations
- Small amounts of packet loss often get disregarded



1% of sustained packet loss can cause up to 70% throughput degradation as per ThousandEyes research on the topic of [“The surprising impact of 1% packet loss”](#)

Spicing things up with packet loss



- AS32100 is under our administrative control
- OSPF is used to advertise loopback interfaces, while internal nodes establish iBGP peering within AS32100
- Routers R1 and R2 peer with upstream providers AS700 and AS900, respectively
- AS700 serves as the primary transit provider, while AS900 acts as a backup - configured using LOCAL_PREF for outbound (egress) traffic and AS_PATH prepending for inbound (ingress) traffic
- ISP1-L1 is a Linux node with IP forwarding enabled, acting as a router using the BIRD routing daemon
- ISP1-L1 begins to introduce packet loss using Linux tool traffic control (tc)

Alerting on packet loss

- An alert is configured to trigger upon detecting 10% packet loss between the source and target
- Alerts are associated with a specific tests
- Alert conditions should be carefully tuned to minimize noise and avoid false positives
- When triggered, the alert sends a webhook to Event-Driven Ansible

Alert Type

Network

Agent to Agent

Rule Name

Packet Loss!

Settings

Notifications

Direction

Both Directions

Tests

1 of 2 test(s) selected

Agents

All agents

Severity

Info

Minor

Major

Critical

ALERT CONDITIONS

Alert Detection ?

Adaptive

Manual

Manual alerting lets you configure when to trigger alerts.

All conditions are met by

any of

1

agent

1

of

1

time in a row:

Packet Loss

≥

10

%

+ Add Alert Condition

Integrations? Webhooks?

- ThousandEyes supports multiple integration options, helping streamline alerting and operational workflows
- Webhook is a way for one application to send real-time data to another application as soon as an event occurs
- Instead of relying on continuous polling for updates, a notification is triggered immediately when a change is detected
- In our case, the underlying alert condition changed due to a spike in packet loss
- As a result, a webhook was sent to Event-Driven Ansible



AppDynamics

Test Recommendations

Alert Notifications



Custom Webhook

Alert Notifications



Slack

Alert Notifications



PagerDuty

Alert Notifications



ServiceNow

Alert Notifications

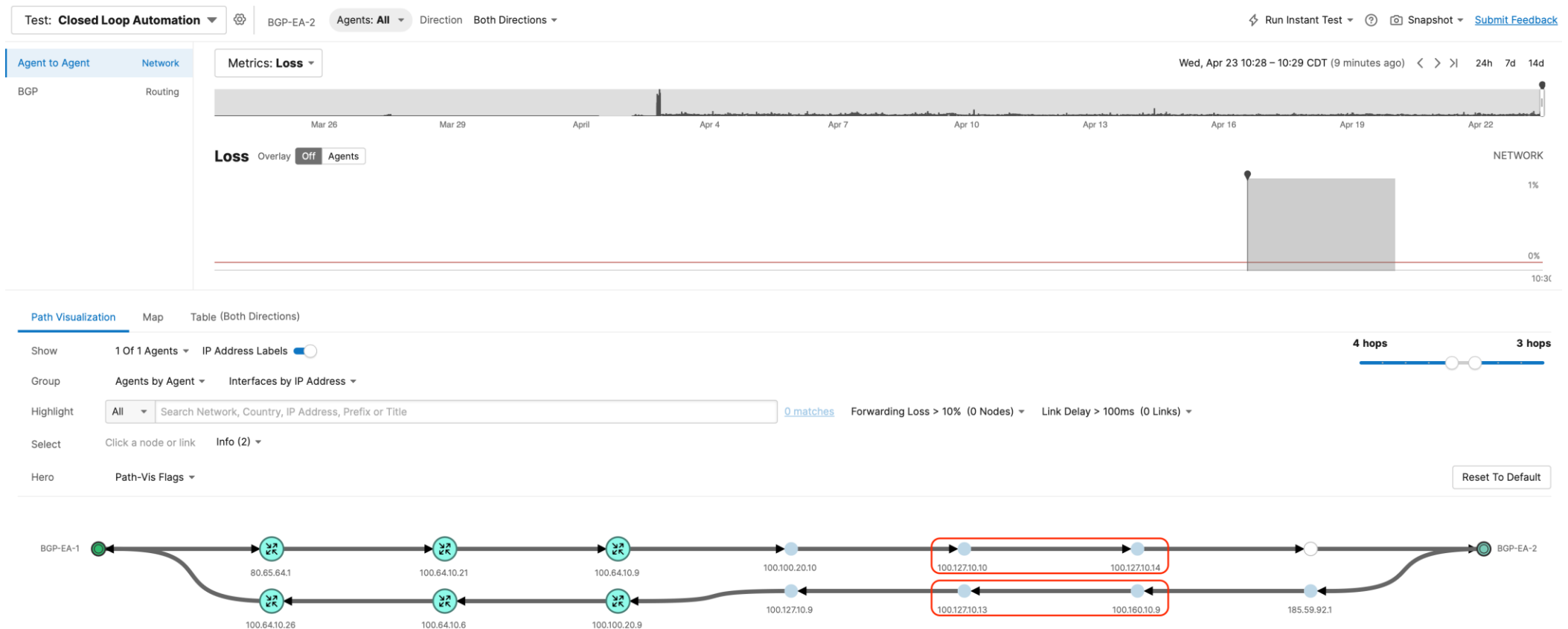
Automated Traffic Engineering

- ThousandEyes detects a spike in packet loss and triggers an alert
- Alert is sent as a webhook to Event-Driven Ansible
- Upon receiving the webhook, Event-Driven Ansible will:
 - SSH to router R1
 - Apply inbound and outbound route maps to the BGP session with AS700, the primary transit provider
- As a result of this automated traffic engineering, traffic is rerouted through the backup transit provider, AS900

Traffic engineering

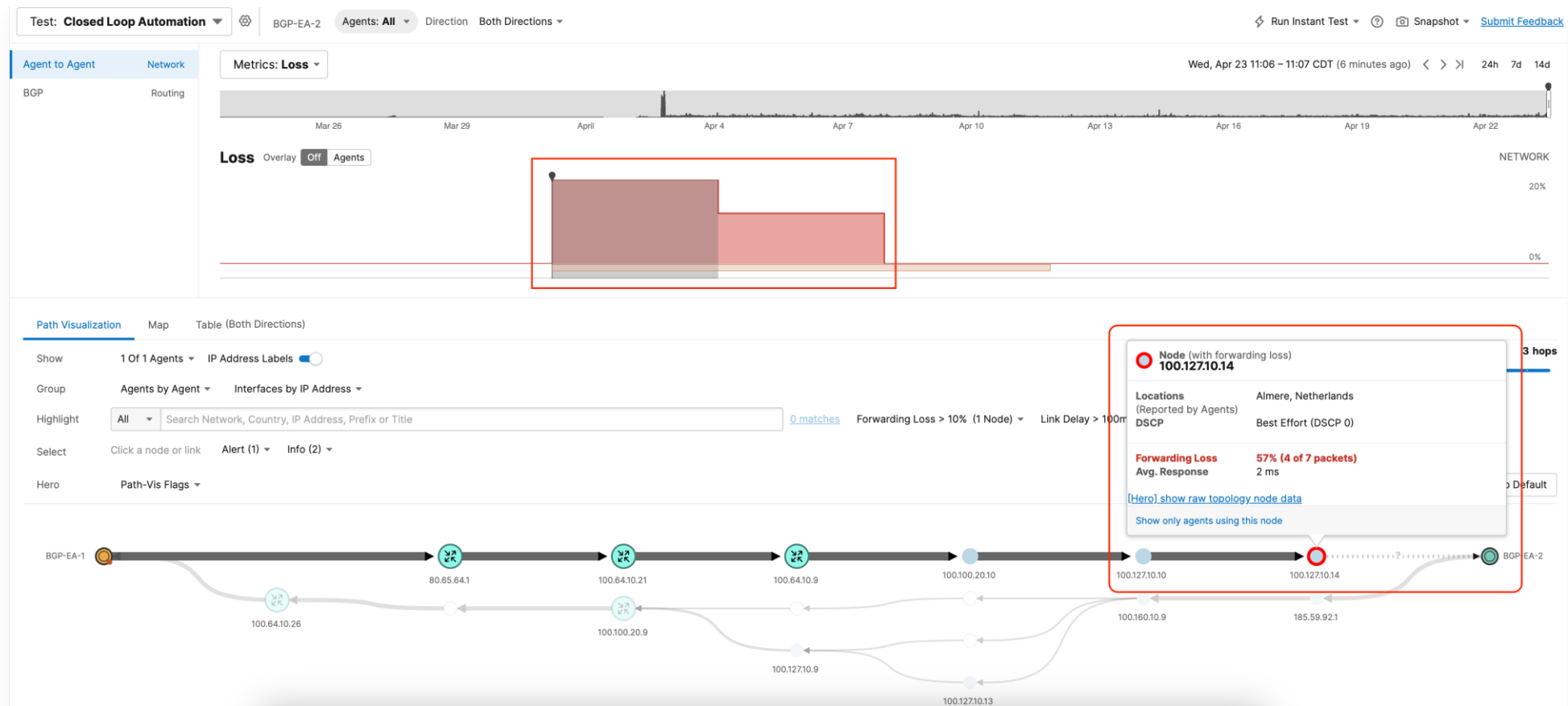
```
!  
route-map FORCE-STANDBY-ISP-  
STATUS-IN permit 10  
  match ip address prefix-list  
  RECEIVED-PREFIX  
  set local-preference 25  
!  
  
!  
route-map FORCE-STANDBY-ISP-  
STATUS-OUT permit 10  
  match ip address prefix-list  
  ADVERTISED-PREFIX  
  set as-path prepend 32100 32100  
!
```

Path Visualization



- Path Visualization shows that there is no packet loss between the source and target
- Traffic in both directions is traversing AS700, as evidenced by the use of IP addresses with octets starting with 100.127.

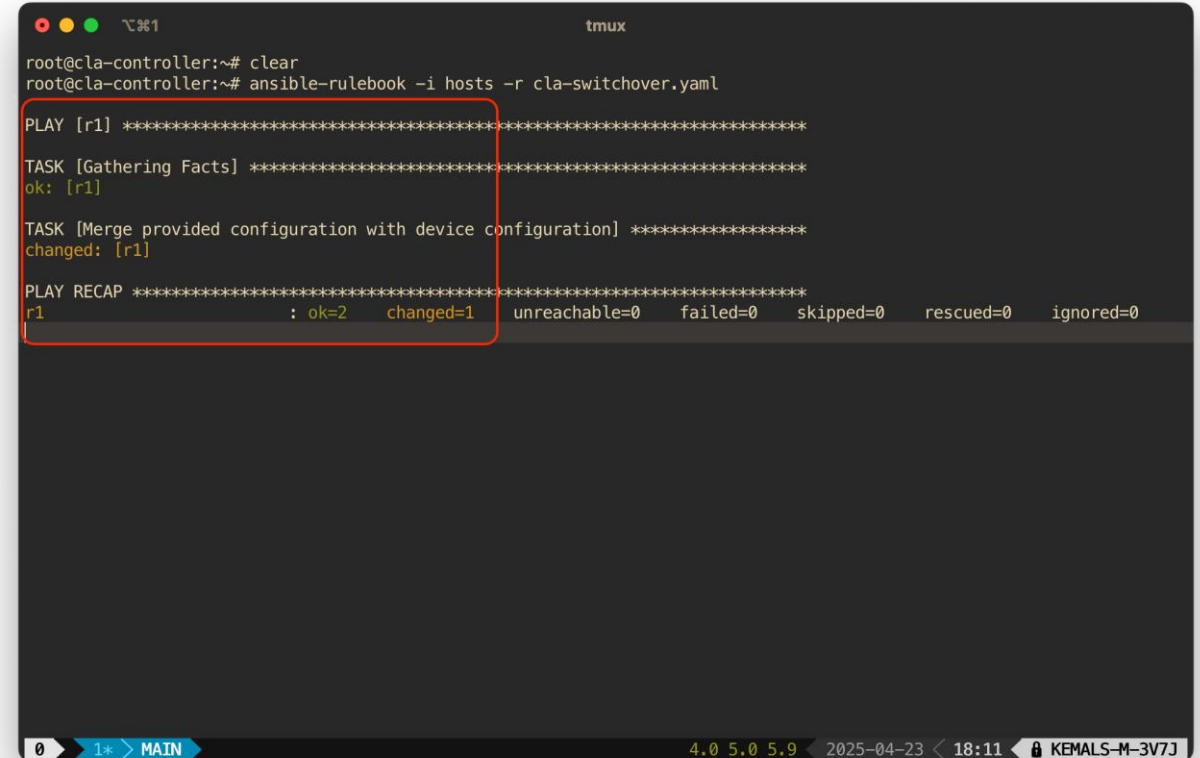
ThousandEyes detects packet loss



- ThousandEyes detects the packet loss previously introduced in AS700 and triggers an alert
- Path Visualization helps pinpoint the exact location of the packet loss!

Automated Traffic Engineering

- Upon receiving the webhook from ThousandEyes, Event-Driven Ansible executes the switchover playbook:
- It applies FORCE-STANDBY-ISP-STATUS-OUT on the peering with AS700, prepending the AS path twice to influence ingress traffic
- It applies FORCE-STANDBY-ISP-STATUS-IN on the peering with AS700, setting LOCAL_PREF to 25, which affects egress traffic



A terminal window titled 'tmux' showing the execution of an Ansible playbook. The user is at the 'root@cla-controller:~#' prompt. They first run 'clear' and then 'ansible-rulebook -i hosts -r cla-switchover.yaml'. The output shows a play named 'r1' with two tasks: 'Gathering Facts' (successful) and 'Merge provided configuration with device configuration' (changed). A 'PLAY RECAP' line at the bottom shows 'ok=2' and 'changed=1'. A red box highlights the task output section.

```
root@cla-controller:~# clear
root@cla-controller:~# ansible-rulebook -i hosts -r cla-switchover.yaml

PLAY [r1] *****

TASK [Gathering Facts] *****
ok: [r1]

TASK [Merge provided configuration with device configuration] *****
changed: [r1]

PLAY RECAP *****
r1 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

Event Driven Ansible

Upstream provider switchover

Rulebook

```
---
- name: Receive and print ThousandEyes webhook
  events
    hosts: all
    sources:
      - ansible.eda.webhook:
        host: 0.0.0.0
        port: 8080

    rules:
      - name: Execute traffic engineering to switchover to
        standby ISP
        condition: event.payload.type == "2"
        actions:
          - run_playbook:
            name: ./switchover.yaml
...

```

Playbook 1/2

```
---
-
  hosts: r1

  vars:
    ansible_connection:
      ansible.netcommon.network_cli
    ansible_network_os: cisco.ios.ios
    ansible_become: yes
    ansible_become_method: enable

```

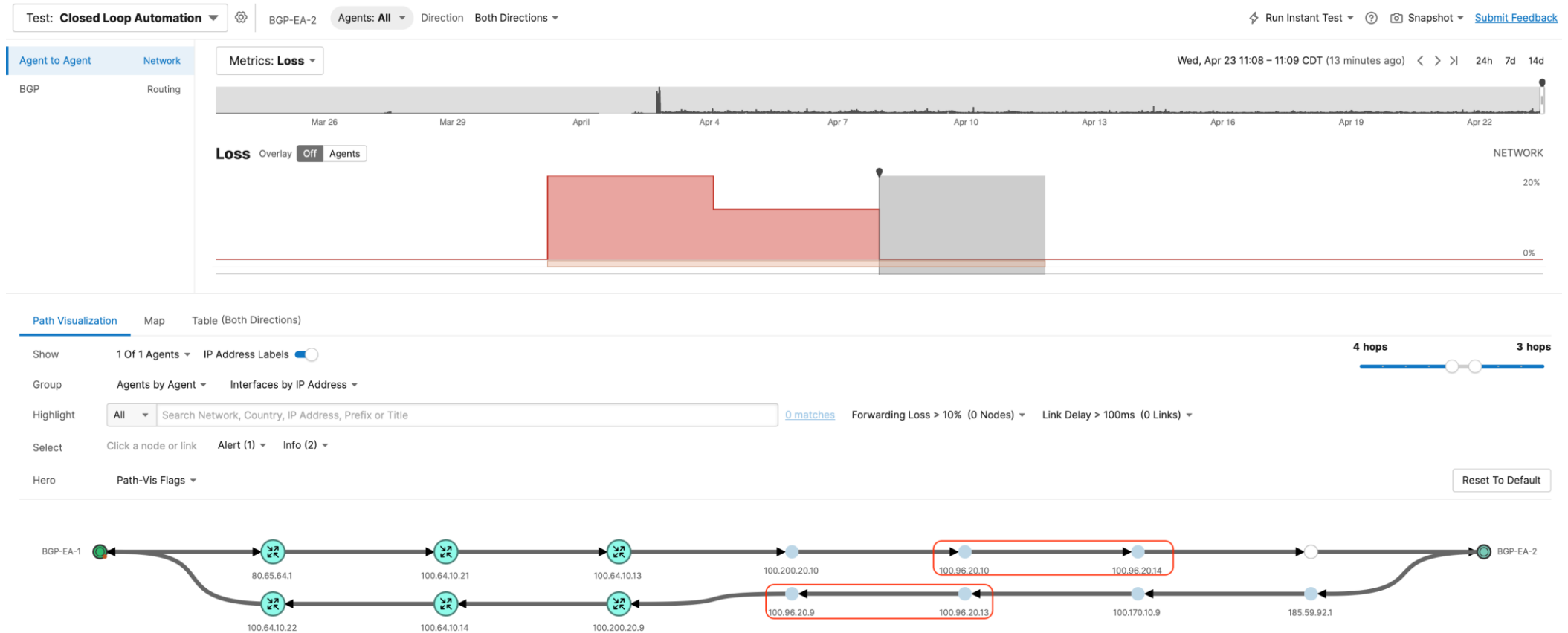
Playbook 2/2

```
tasks:
  - name: Merge provided configuration with
    device configuration
    cisco.ios.ios_bgp_global:
      config:
        as_number: 32100

      neighbor:
        - neighbor_address: 100.100.20.10
          description: Primary transit
          remote_as: 700
          soft_reconfiguration: true
          fall_over:
            bfd:
              set: true
            route_maps:
              - name: FORCE-STANDBY-ISP-STATUS-OUT
                out: true
              - name: FORCE-STANDBY-ISP-STATUS-IN
                in: true
...

```

ThousandEyes visualizes effects of automation



- After automated traffic engineering, traffic begins transiting through AS900, as indicated by the use of IP addresses with octets starting with 100.96.
- ThousandEyes detects the new path and clears the alert, as there is no longer any packet loss

Demo



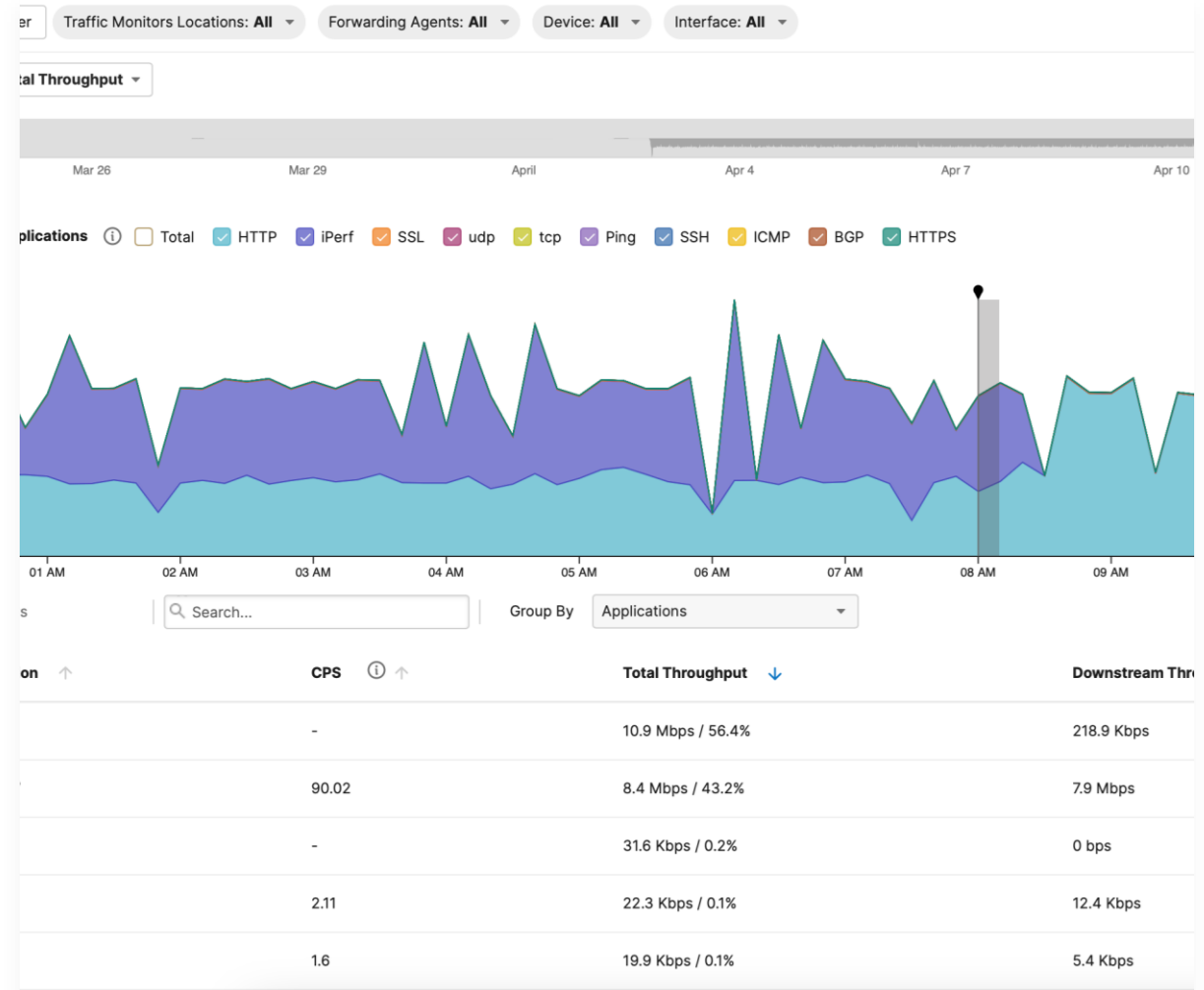
Automating DDOS Remediation

Distributed Denial of Service (DDoS)

- **Distributed Denial of Service (DDoS)** attacks aim to overwhelm a target (website, servers or network) with traffic from multiple sources, rendering it inaccessible
- Different types:
 - **Volumetric Attacks** – Flood the bandwidth (e.g., UDP floods, amplification attacks)
 - **Protocol Attacks** – Exploit weaknesses in Layer 3/4 protocols (SYN floods)
 - **Application Layer Attacks** – Target specific apps or services (HTTP floods)
- **Motivation:** Financial extortion, political or ideological disruption (hactivism), competitive advantage

ThousandEyes Traffic Insights

- Traffic Insights provides network teams with a detailed view of traffic running on the networks they manage
- Collecting and correlating flow (NetFlow and IPFIX) data to generate a contextualized understanding of network performance
- Offers option to view traffic grouped by conversation (5-part tuple), application, or application path to gain macro-level visibility into all the traffic running across the enterprise
- Comprehensive alerting capabilities
- Alerting based on amount of ingress traffic is suitable for detection of DDOS attacks



Traffic Insights Alerting

- Traffic Insights offer comprehensive alerting capabilities, including alerts based on:
 - Devices, applications, subnet tags, and geolocations
 - Traffic direction (ingress/egress)
 - Connections per second
 - Total throughput
- When an alert is triggered due to ingress traffic exceeding the configured threshold, a webhook is sent
- Webhook is received and acted upon by Event-Driven Ansible

The screenshot displays the configuration page for a Traffic Flow alert rule. The 'Alert Type' is set to 'Traffic Flow'. The 'Alert Rule Name' is 'Unexpected Traffic Levels!'. The 'Settings' tab is active, showing 'Scope Types' as 'Devices' with '1 of 2 devices selected'. The 'Severity' is set to 'Critical'. Under 'ALERT CONDITIONS', it specifies 'Any Interface Matching' and 'the following property: Interface Name matches GigabitEthernet2'. It also states 'meets all of the following conditions: Total Throughput ≥ 900 Mbps'. A '+ Add Condition' link is at the bottom.

Alert Type
Traffic Flow

Alert Rule Name
Unexpected Traffic Levels!

Settings Notifications

Scope Types
Devices 1 of 2 devices selected

Severity
Info Minor Major Critical

ALERT CONDITIONS

All conditions are met by at least 5 mins:

Any Interface Matching

the following property:

Interface Name matches GigabitEthernet2 Interfaces...

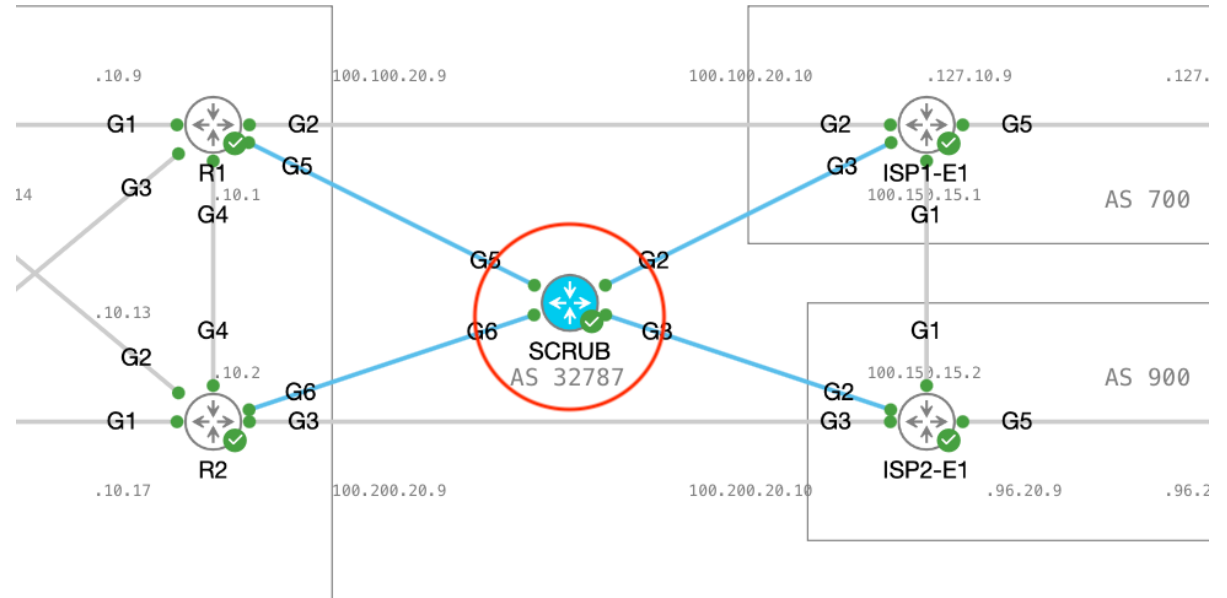
meets all of the following conditions:

Total Throughput ≥ 900 Mbps

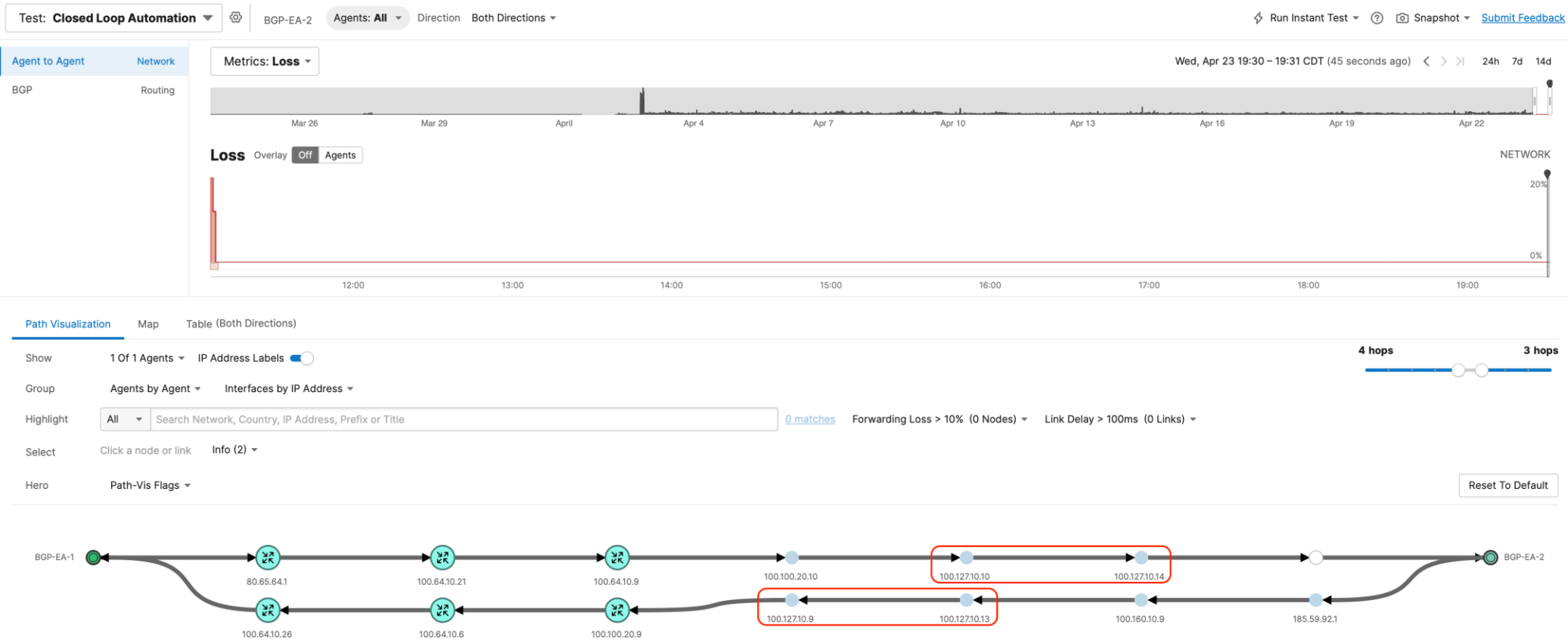
+ Add Condition

Dealing with Distributed Denial of Service (DDOS)

- Few different ways of dealing with DDOS:
 - Traffic filtering and rate limiting
 - Anycast routing
 - Scrubbing (dedicated scrubbing providers or hosted) to absorb and clean the traffic
- On demand and always on scrubbing

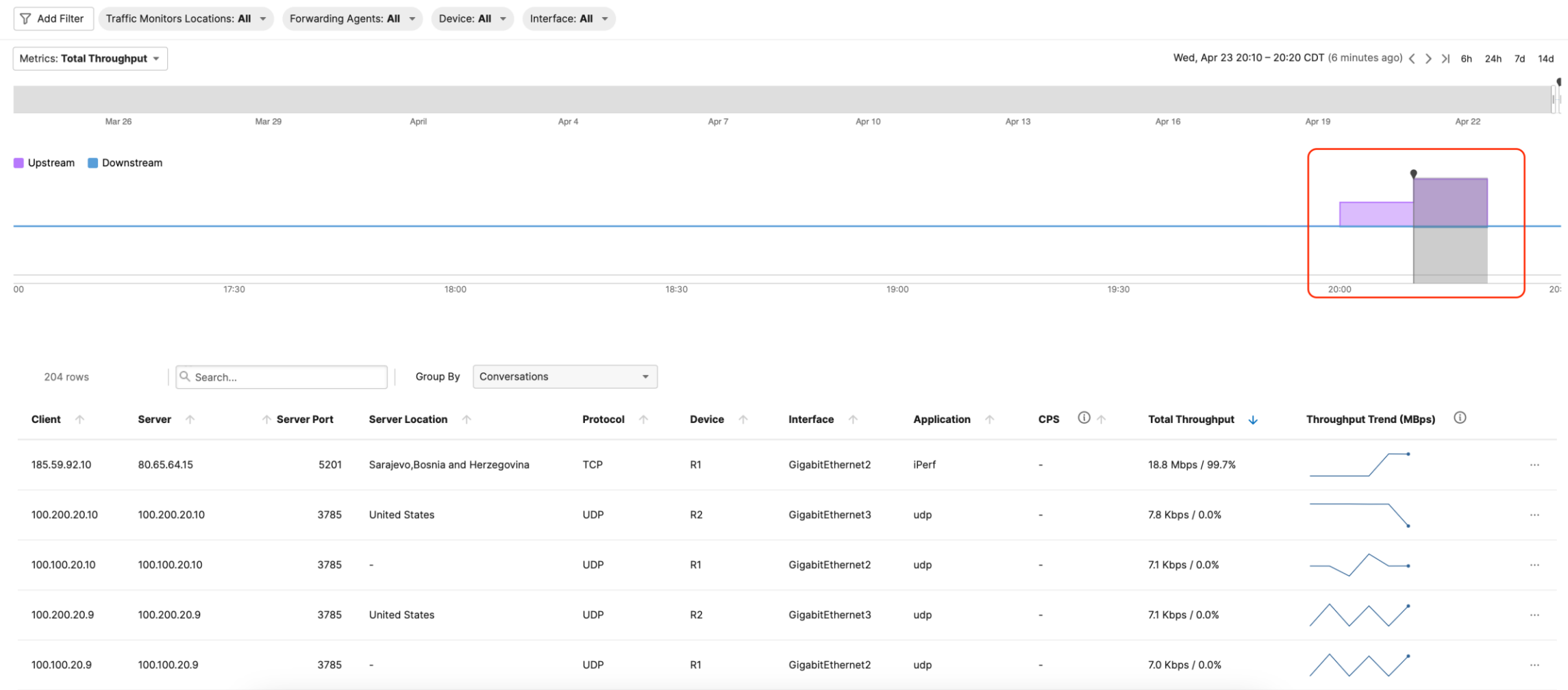


ThousandEyes Path Visualization



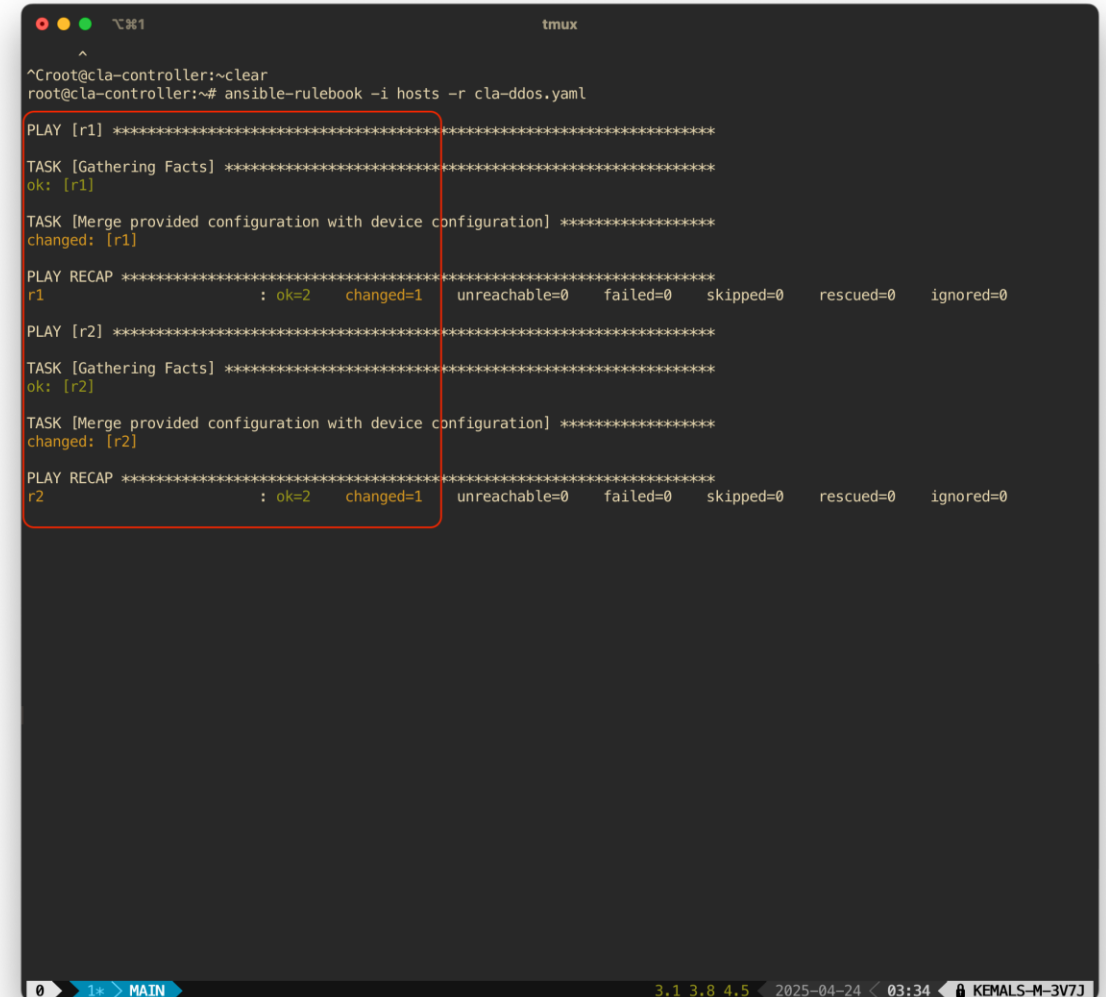
- ThousandEyes indicates that there is no packet loss, which is typically a sign of a DDoS attack
- Traffic in both directions is passing through AS700, as evidenced by the use of IP addresses with octets starting with 100.127.

Traffic Insights observes the spike in Ingress Traffic



Automated Traffic Engineering

- Upon detecting a traffic spike, ThousandEyes triggered a webhook
- Event-Driven Ansible received the webhook and initiated automated traffic engineering
- BGP sessions on routers R1 and R2 with upstream providers AS700 and AS900 had the route-map DENY applied in the outbound direction, preventing prefixes from being advertised to those providers
- Routers R1 and R2 then began advertising prefixes exclusively to the scrubbing provider, AS32787
- The scrubbing provider acted as the sole upstream for AS32100



```
^Croot@cla-controller:~clear
root@cla-controller:~# ansible-rulebook -i hosts -r cla-ddos.yaml

PLAY [r1] *****
TASK [Gathering Facts] *****
ok: [r1]

TASK [Merge provided configuration with device configuration] *****
changed: [r1]

PLAY RECAP *****
r1 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

PLAY [r2] *****
TASK [Gathering Facts] *****
ok: [r2]

TASK [Merge provided configuration with device configuration] *****
changed: [r2]

PLAY RECAP *****
r2 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

0 1* MAIN 3.1 3.8 4.5 2025-04-24 03:34 KEMALS-M-3V7J
```

Event Driven Ansible Automating DDOS Remediation

Rulebook

```
---
- name: Receive and print ThousandEyes webhook
  events
    hosts: all
    sources:
      - ansible.eda.webhook:
        host: 0.0.0.0
        port: 8080

  rules:
    - name: Advertise prefixes to DDOS-Scrubbing
      provider and stop advertising to Transit providers
      condition: event.payload.type == "2"
      actions:
        - run_playbook:
            name: ./ddos-r1.yaml
        - run_playbook:
            name: ./ddos-r2.yaml
```

Playbook (R1)

```
tasks:
  - name: Merge provided configuration with
    device configuration
    cisco.ios.ios_bgp_global:
      config:
        as_number: 32100

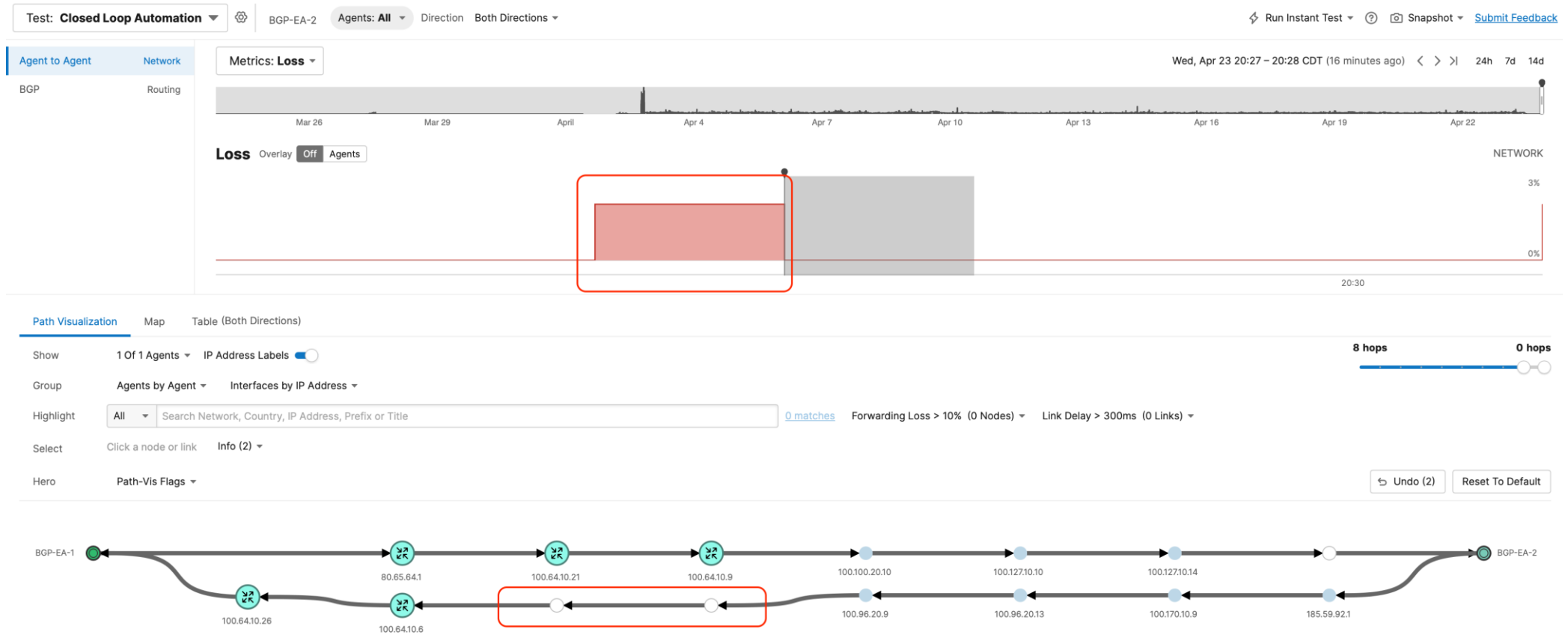
    neighbor:
      - neighbor_address: 100.100.20.10
        description: Primary transit
        remote_as: 700
    route_maps:
      - name: DENY
        out: true
      - neighbor_address: 192.200.100.1
        description: DDOS Scrubbing provider
        remote_as: 32787
        soft_reconfiguration: true
    route_maps:
      - name: PRIMARY-TRANSIT-OUT
        out: true
```

Playbook (R2)

```
tasks:
  - name: Merge provided configuration with
    device configuration
    cisco.ios.ios_bgp_global:
      config:
        as_number: 32100

    neighbor:
      - neighbor_address: 100.200.20.10
        description: Backup transit
        remote_as: 900
    route_maps:
      - name: DENY
        out: true
      - neighbor_address: 192.200.100.5
        description: DDOS Scrubbing provider
        remote_as: 32787
        soft_reconfiguration: true
    route_maps:
      - name: STANDBY-ISP-OUT
        out: true
```

ThousandEyes visualizes remediation



- After automated traffic engineering, ingress traffic begins transiting through AS32787
- Following the traffic engineering, ThousandEyes reports no packet loss, indicating successful reroute to on demand DDOS scrubbing provider

Demo



Automating BGP Hijacks Remediation

BGP Hijacks?

- A **BGP hijack** occurs when a malicious or misconfigured network advertises IP prefixes it doesn't own, diverting or blackholing traffic
- BGP was designed at a time when **security was not a primary concern**, with security often considered an afterthought
- Notable incidents include:
 - Pakistan Telecom hijacking YouTube, causing global outages
 - The Quad9 hijack

Challenges with BGP Hijacks?

- **Detection** focuses on monitoring advertised prefixes
- While **RPKI adoption** is increasing, much more still needs to be done
- Once hijacked prefix propagates on the internet operators are left with **limited options**:
 - Call the upstream provider propagating the hijacked prefix (time-consuming, error-prone, and with a limited success rate)
 - Contact the hijacker's ASN and hope it was an accidental routing mistake
 - Advertise more specific IP address space (if not already done)

Turbocharging BGP Visibility using ThousandEyes

- ThousandEyes offers comprehensive BGP monitoring with near real-time visibility
- It provides an effective way to verify various operational aspects, such as traffic engineering changes, prefix propagation, etc.
- It supports RPKI monitoring and delivers robust alerting capabilities

The screenshot displays the ThousandEyes alert configuration interface. At the top, the 'Alert Type' is set to 'BGP - Routing' and the 'Rule Name' is 'BGP Hijack/Route leak'. Below this, there are two tabs: 'Settings' (active) and 'Notifications'. Under the 'Settings' tab, the 'Tests' dropdown shows '1 of 6 test(s) selected'. The 'Prefix Length' dropdown is set to 'IPv4 /16 - /32 And IPv6 /32 - /128'. The 'Monitors' dropdown is set to 'All monitors'. A checkbox labeled 'Alert on covered prefix data' is checked. The 'Severity' section has four buttons: 'Info' (selected), 'Minor', 'Major', and 'Critical'. Below the settings, the 'ALERT CONDITIONS' section shows 'All conditions are met by' with a dropdown set to 'any of', followed by a value of '1', a dropdown set to 'monitor', and 'within 1 min:'. A condition is listed: 'BGP Origin' meets all of: 'ASN' not in '32100'. There are expand/collapse icons for the condition and the list item. A 'Collapse All' link is at the bottom.

Alert Type BGP - Routing

Rule Name BGP Hijack/Route leak

Settings Notifications

Tests 1 of 6 test(s) selected

Prefix Length IPv4 /16 - /32 And IPv6 /32 - /128

Monitors All monitors

☒ Alert on covered prefix data

Severity Info Minor Major Critical

ALERT CONDITIONS

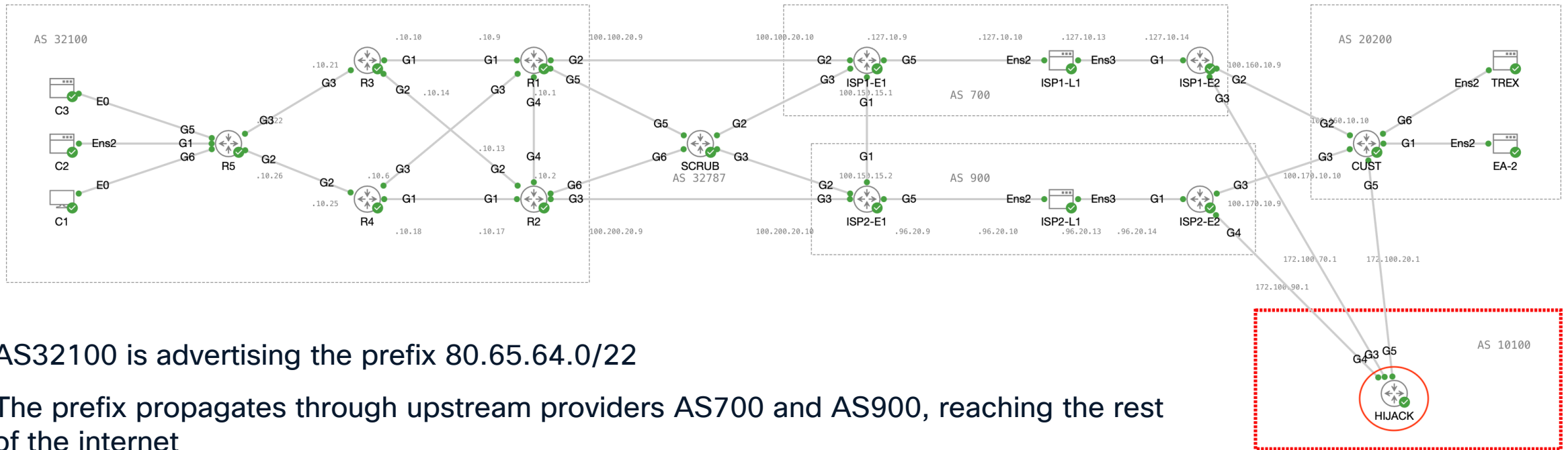
All conditions are met by any of 1 monitor within 1 min:

BGP Origin meets all of:

ASN not in 32100

[Collapse All](#)

Hijacking BGP prefix

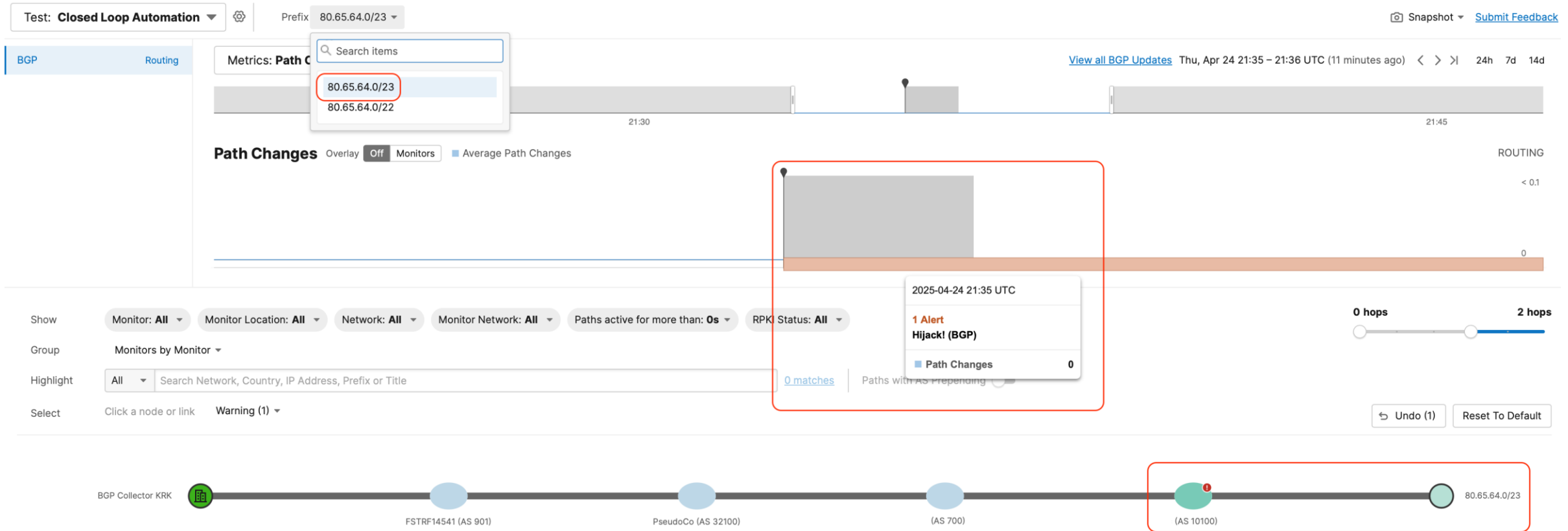


- AS32100 is advertising the prefix 80.65.64.0/22
- The prefix propagates through upstream providers AS700 and AS900, reaching the rest of the internet
- Advertised prefix lacks an RPKI ROA, as AS32100 does not adhere to routing security best practices
- A malicious actor within AS10100 begins advertising the prefix 80.65.64.0/23.
- This unauthorized announcement also propagates through AS700 and AS900, eventually reaching the global internet and resulting in a successful BGP hijack

BGP Hijack Remediation

- From an operational standpoint, the quickest and most effective way to mitigate BGP hijacks is to advertise more specific prefixes, if feasible.
- This leverages a fundamental routing principle: ***"the most specific route always wins"***
- In such cases, the longest prefix match rule applies

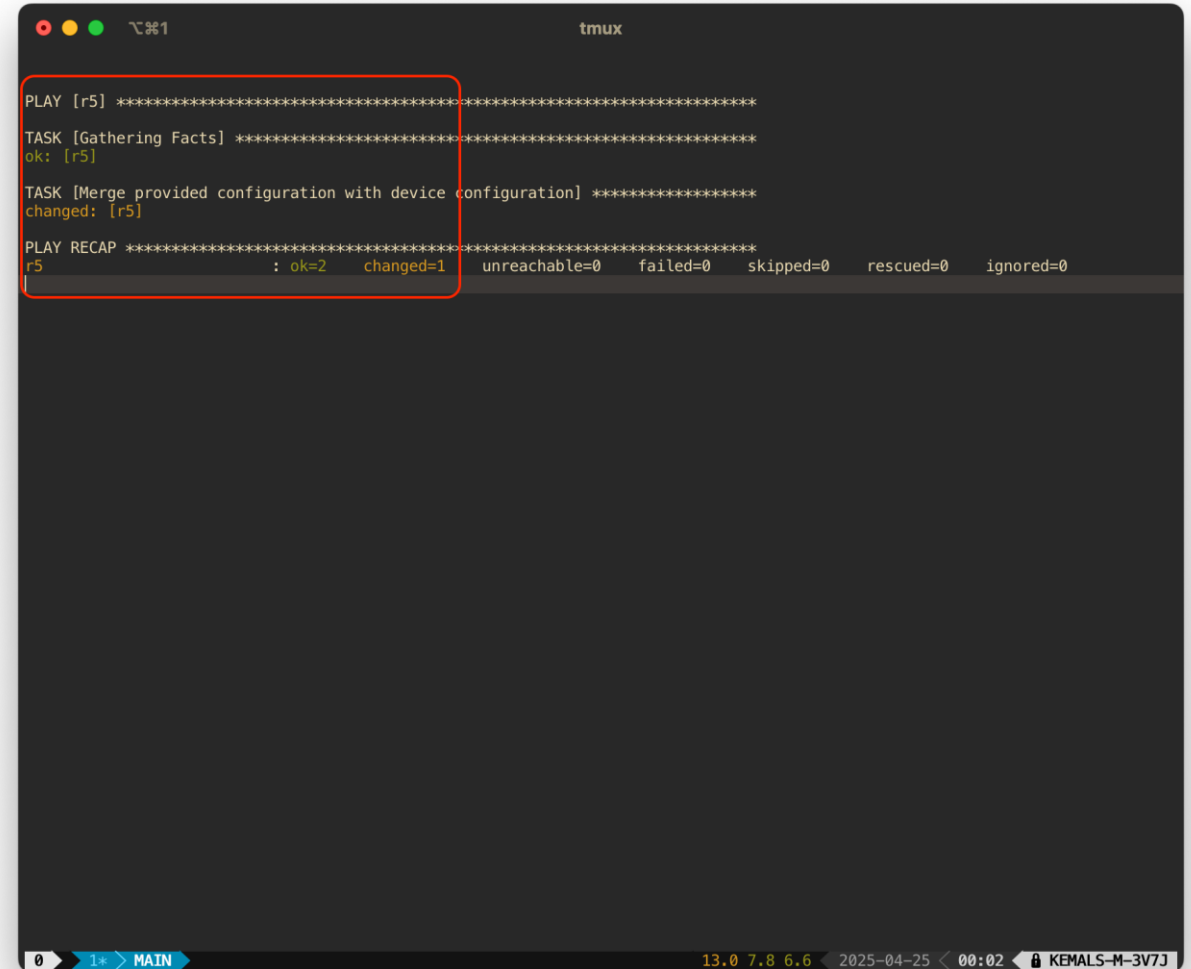
BGP Route Visualization



- ThousandEyes detects the advertisement of the prefix 80.65.64.0/23 and raises a BGP hijack alert
- BGP Route Visualization indicates that AS10100 is originating the announcement of 80.65.64.0/23
- ThousandEyes sends a webhook to Event-Driven Ansible

Automated Traffic Engineering

- Upon receiving the webhook from ThousandEyes, Event-Driven Ansible initiates automated remediation
- The prefix 80.65.64.0/22 is split into four more specific /24 prefixes: 80.65.64.0/24 through 80.65.67.0/24
- Advertising these more specific prefixes in response to the BGP hijack effectively mitigates the issue



```
tmux
PLAY [r5] *****
TASK [Gathering Facts] *****
ok: [r5]
TASK [Merge provided configuration with device configuration] *****
changed: [r5]
PLAY RECAP *****
r5 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

Event Driven Ansible

Automating BGP Hijack Remediation

Rulebook

```
---
- name: Receive and print ThousandEyes webhook
  events
    hosts: all
    sources:
      - ansible.eda.webhook:
        host: 0.0.0.0
        port: 8080

    rules:
      - name: Advertise more specific prefixes to
        remediate hijack
        condition: event.payload.type == "2"
        actions:
          - run_playbook:
            name: ./hijack.yaml
```

Playbook (first part)

```
---
-
  hosts: r5

  vars:
    ansible_connection:
ansible.netcommon.network_cli
    ansible_network_os: cisco.ios.ios
    ansible_become: yes
    ansible_become_method: enable
```

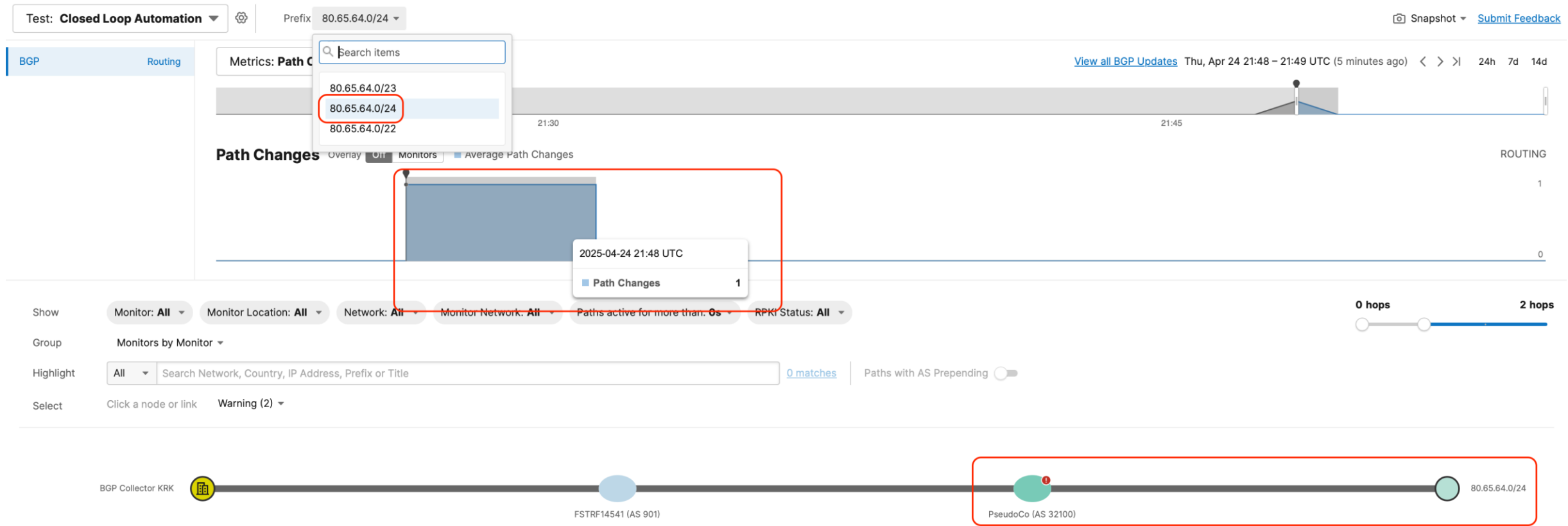
Playbook (second part)

```
tasks:
  - name: Merge provided configuration with
    device configuration
    cisco.ios.ios_bgp_global:
      config:
        as_number: 32100

      neighbor:
        - neighbor_address: 100.64.10.21
          remote_as: 32100
          soft_reconfiguration: true
        - neighbor_address: 100.64.10.25
          remote_as: 32100
      networks:
      - address: 80.65.64.0
        netmask: 255.255.255.0
      state: merged
```

...

BGP Hijack Remediation, visualized



- BGP Route Visualization confirms that the prefix 80.65.64.0/24 is being advertised by AS32100
- The announcement of the more specific prefix successfully mitigates the immediate impact of the BGP hijack

Demo

A person is silhouetted against a vibrant sunset sky, standing on the sharp, rocky edge of a mountain peak. The sun is low on the horizon, casting a warm glow across the sky and illuminating the person's figure. The background features a series of layered, hazy mountain ranges stretching into the distance. The overall mood is one of achievement and contemplation.

Conclusion

Conclusion

- **Start small** with well-understood, repeatable workflows that have minimal impact. This helps build confidence and trust in the automation process
- While automation might seem like a steep hill to climb, **it's well worth the effort**
- **Automation frameworks and tools** have become significantly more powerful and easier to use over the past decade
- Always **include verification steps** in your automated workflows – for example, ensuring an alternate path is healthy, peering sessions are up, and routing changes are safe
- When done right, automation can significantly improve:
 - **Speed and Responsiveness:** Detect and react to incidents in real time – far faster than manual interventions
 - **Scalability:** Handle a large volume of well-understood incidents related to repeatable and well understood workflows without increasing operational load
 - **Consistency and Reliability:** Deliver predictable, repeatable outcomes
 - **Efficiency:** Free up your team to focus on more valuable, strategic work
- **Apply caution and thorough validation:** Poorly implemented automation can lead to outages, reputational damage, and financial loss

Complete your session evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to win 1 of 5 full conference passes to Cisco Live 2026.



Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.



Level up and earn exclusive prizes!



Complete your surveys in the Cisco Live mobile app.

Continue your education



Visit the Cisco Showcase for related demos



Book your one-on-one Meet the Engineer meeting



Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs



Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact us on Webex
Kemal Sanjta | Mike Hicks

Thank you

CISCO Live !

