INT 技术白皮书

文档版本 V1.0

发布日期 2022-12-16

商标说明

Inspur 浪潮、Inspur、浪潮、Inspur NOS 是浪潮集团有限公司的注册商标。

本手册中提及的其他所有商标或注册商标，由各自的所有人拥有。

技术支持

技术服务电话：400-860-0011

地　　址：中国济南市浪潮路 1036 号

浪潮电子信息产业股份有限公司

邮　　箱：lckf@inspur.com

邮　　编：250101

前言

变更记录

版本	时间	变更内容
V1.0	2022-12-16	首版发布

目 录

1 概述

1.1 背景

在传统的企业网或数据中心内，随着网络规模的不断扩大，对网络监控的需求也在不断增加，特别是在对于网络可靠性要求越来越高的情况下。如何在网络发生状况时快速发现问题以及预测网络即将出现的故障点，是一个业界亟欲解决的问题。

当前业界常用的网络遥测方式有以下几种：

1. 传统网络测量

1.1. 主动测量

主动测量通过向网络中主动传送探测分组，并根据探测分组受网络影响而发生的特性变化来分析网络行为。被测量的网络效能指针通常是丢包率、延迟、抖动、TTL 和带宽等。常见的主动测量协议包括 PING、Traceroute、IP 测量协议(IP Measurement Protocol，IPMP)、单向主动测量协议(One-Way Active Measurement Protocol，OWAMP)、双向主动测量协议(Two-Way Active Measurement Protocol，TWAMP)、MPLS 丢包/延迟测量协议(MPLS L/DM Protocol)。

1.2. 被动测量

被动测量通过捕获流经测量点的分组来测量网络状态、流量特征和效能自变量。被动测量使用控制平面讯息即可监测网络流量状态效能，被监测的效能指针通常是包/字节统计值、协议型别、队列长度和延迟统计信息。常见的被动测量协议有网络数据流统计协议(Cisco Netflow)、sFlow、IP 流量信息输出协议(IPFIX)、数据报取样协议(PSAMP)。

2. 带内网络遥测(INT)

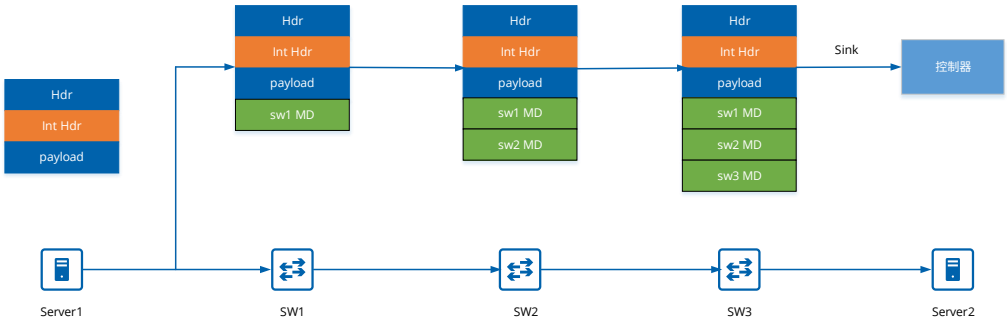
带内测量是近几年兴起的一种混合测量方法，它通过路径中间交换节点对数据报依次插入元数据(Metadata)的方式完成网络状态采集。相较于传统网络测量方案，带内测量能够对网络拓扑、网络效能和网络流量实现更细粒度的测量。

1.2 定义

INT(Inband network telemetry)，是一种将特定的 metadata 插入网络流，然后将该流的网络运行情况返回控制器分析的方法，主要用于确定端到端数据流的网络状态，以及查找网络延迟大的节点，如下图所示，Server1 将封包送到 SW1，SW1 将网络信息附加到该封包

后送到 SW2，SW2 重复同样工作并传给 SW3，SW3 作为最终的配置节点会在此时将该封包传送给控制器。

图 1-1 方案拓扑



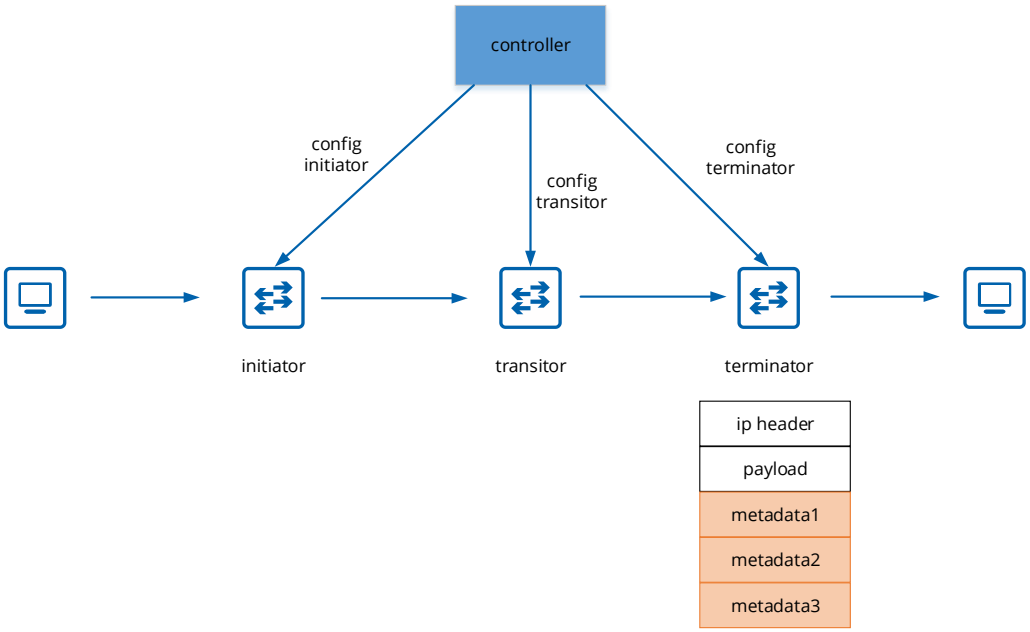
1.3 优点

INT 作为一种网络探测的技术，具有如下优点：

- 记录的是带内的网络等待时间，相较于带外查询的数据更精准
- 减少封包探测次数，一次探测就能回收整条流的网络信息

2 缩写和术语

图 2-1 INT



如上图所示：

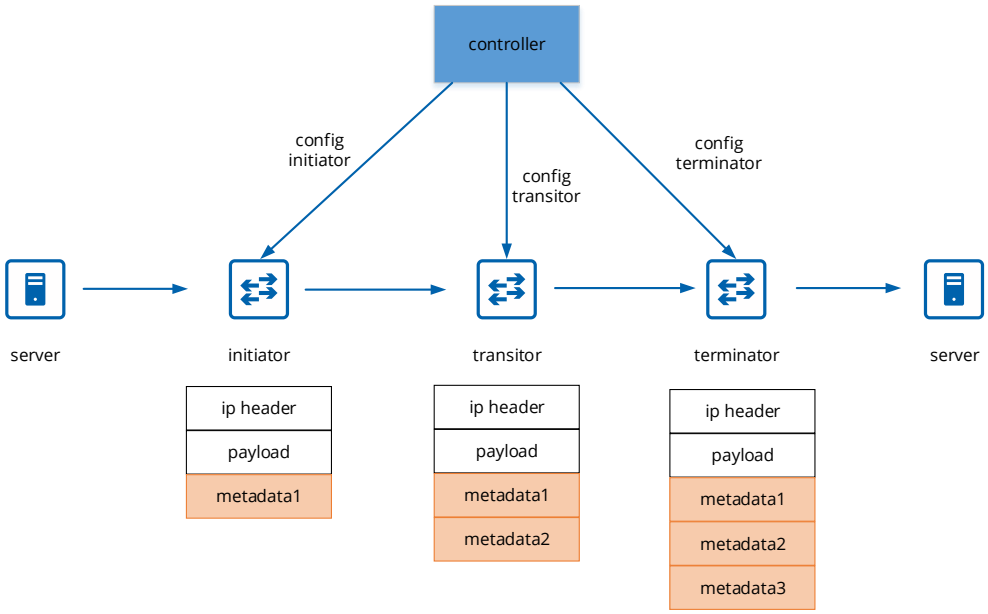
缩写和术语	解释
INT	Inband Network Telemetry，带内网络遥测
Controller	控制交换机配置的控制器，也负责采集及分析回传数据
Initiator	路径上第一台交换机，传统上负责建立INT header及插入网络状态到封包中，在DSCP based INT框架中，只插入网络状态到封包中
Transit	路经中间的n台交换机，负责插入网络状态到封包中
Terminator	路径上最后一台交换机，负责回传遥测数据给controller
MD	Metadata，交换机附加到封包中的网络状态
DSCP	Differentiated Services Code Point，在ip header中用来做为封包分类的用途

3 技术介绍

本章主要介绍 INT 的芯片支持和软件支持的技术特点，包括应用场景、三种角色介绍及新增的 metadata 字段内容。

3.1 应用场景

图 3-1 应用场景



通常情况下，当数据中心需要测量端到端之间的网络状态时，会使用 INT，此时会在封包通过的交换机上进行配置，这些交换机被分为三种角色，initiator、transitor 和 terminator，当数据封包到达一个交换机时，识别 INT 报文表头，交换机会在封包后插入 MD，以此类推，直到整个遥测系统的最后一跳，再通过 gRPC 或是 ERSPAN 的方式回传给控制器。

3.2 芯片支持 INT

交换芯片需支持 INT 表头，包含封装和解析报文表头。对于首节点的镜像报文来说，需要由 INT 交换芯片对其添加 INT 头，生成 INT 报文；对于尾节点来说，INT 交换芯片将 INT 报文中监测信息的封装格式做一致性检查，然后对 INT 报文封装外层表头回传给控制器。

3.3 软件支持 INT

INT 不需要芯片支持也可以实现。当芯片不具有看懂 INT header 的能力，不能够通过解析其内容来判断要添加什么网络状态到封包中的时候，可以通过 DSCP based INT，在匹配封

包时，不需寻找 INT header，只需保留特定的 DSCP 作为 INT 的识别符号，这样不须芯片支持也可实现软件 INT。

3.4 角色

3.4.1 Initiator

遥测系统的第一跳。芯片支持 INT 中这个角色要负责插入 INT header，供 Transit 判断是否要插入 MD，将报文发送给中间节点。软件支持 INT 的框架中，使用的是控制器发出的特定 DSCP 的探测封包，不是业务封包，所以 Initiator 的任务和 Transit 节点相同，都是配置一个 ACL 去拦截特定 DSCP 的封包后插入 MD。

3.4.2 Transit

遥测系统的中间传递者。芯片支持 INT 中这个角色要判读 INT header，并解析内容再对封包插入需要的 MD。在软件支持 INT 框架中，只需拦截特定 DSCP 封包后插入 MD，再将报文发送给下游节点。

3.4.3 Terminator

遥测系统的最后一跳。芯片支持 INT 中这个角色负责提取全部的 MD 信息，根据用户配置的报文封装参数，对监测信息进行 UDP 头及 IP 头封装，转发到控制器。在软件支持 INT 框架中，使用的是芯片 ERSPAN 的能力，将符合该条件的封包转发到控制器。

3.5 Metadata

为了满足设备维护和网络状态监控等多样需求，现行 INT 的 MD 信息可以说是越来越多样化，下面这张图表展示的是通常 INT 可以收集的统计数据，但是要强调的是，收集的资料越多，代表附加的 MD 会越多，而且每一跳都会增加相对应的 MD，这导致该探测封包有可能在中途因为达到 MTU 的上限值而被丢弃，所以慎选对分析有帮助的 MD 数据上传才是正确的做法。在之前软件支持 INT 框架中，针对客户希望分析的问题——网络等待时间分析，我们挑选了“数据包入端口号”“数据包出端口号”“数据包入端口时间戳”“数据包出端口时间戳”及“交换机编号”这五项指标做为 MD，通过这五项指标，控制器就已经能清楚分析出延迟的网络段范围。

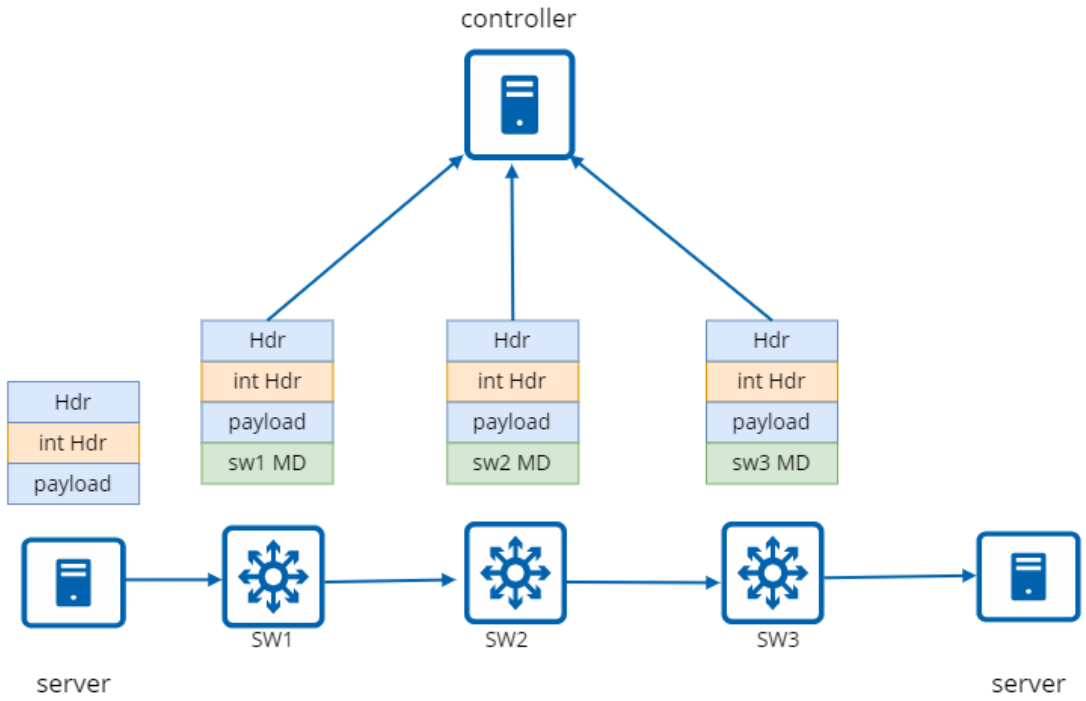
分类方式	统计信息	可读/可写
交换机级状态信息	交换机编号(Switch ID)	Y/N
	L2/L3层流表计数(L2 or L3 flow table count)	Y/N
	流表版本号(Flow table version)	Y/N
	时间戳(Timestamp)	Y/N

	接收数据包计数(Received packets)	Y/N
	接收字节计数(Received bytes)	Y/N
端口级状态信息	端口号(Port ID)	Y/N
	数据包入端口号(Ingress port ID)	Y/N
	数据包出端口号(Egress port ID)	Y/N
	入队列字节数(Bytes enqueued)	Y/N
	链路利用率(Link utilization)	Y/N
	接收字节计数(Bytes received)	Y/N
	传偷字节计数(Bytes transmitted)	Y/N
	丢弃字节计数(Bytes dropped)	Y/N
	接收数据包计数(Packet received count)	Y/N
	传输数据包计数(Packet transmitted count)	Y/N
	丢弃数据包计数(Packet dropped count)	Y/N
	接受错误计数(Receive error count)	Y/N
	传输错误计数(Transmit error count)	Y/N
	接收溢出错误(Receive overrun error count)	Y/N
	接收帧对齐错误(Receive frame alignment error count)	Y/N
	接收CRC校验错误(Receive CRC Error count)	Y/N
	数据包入端口时间戳(Ingress timestamp)	Y/N
	数据包出端口时间戳(Egress timestamp)	Y/N
队列级状态信息	队列ID(Queue ID)	Y/N
	入队列字节数(Bytes enqueued)	Y/N
	丢弃字节数(Bytes dropped)	Y/N
	接收溢出错误计数(Receive overrun error count)	Y/N
数据包级状态信息	数据包入交换机端口(Packet 's input port)	Y/N
	数据包出交换机端口(Packet 's output port)	Y/Y
	数据包计数(Packet number count)	Y/Y
流表级状态信息	数据包查找计数(Packet lookup count)	Y/N
	数据包匹配计数(Packet match count)	Y/N
流级状态信息	流计数(Flow count)	Y/N

3.6 Postcard 模式

除了一般模式以外，INT还有Postcard模式，不再是基于Path进行监控，而是各个节点单独发送INT metadata给采集器，每个INT节点都具备网络事件检测能力，业务数据包在网络的传输过程中不会被插入Metadata。

图 3-2 Postcard 模式



4 主要特性

- 芯片支持 INT，使用芯片能力来处理或封装 INT 报文表头。
- 软件支持 INT，使用 DSCP 作为判断是否为 INT 封包的依据，而非 INT header。

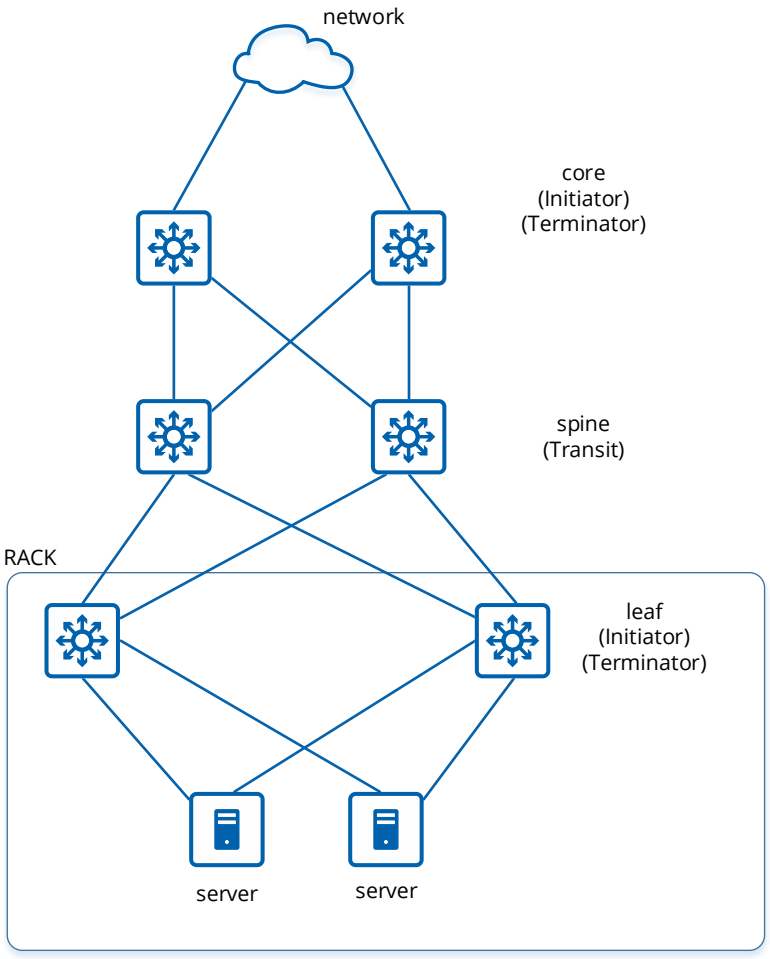
5 典型应用指南

5.1 典型组网方案

下图是数据中心常用的典型拓扑。

- 机柜中的 server 会连接到机柜内的 leaf 交换机，两台 leaf 交换机互为备援和负载平衡。
- 每个机柜中的 leaf 交换机上行口会汇聚到 spine 层的交换机，提供更高的带宽。
- 最后这些 spine 层的交换机会通过 core 层交换机离开数据中心网络系统，core 层交换机通常做为数据中心的网关节点。
- 在 INT 配置中，leaf 和 core 层级的交换机作为遥测网络的边界，通常会同时具备 Initiator 和 Terminator 的身份，spine 层级的交换机则通常配置为 Transit。

图 5-1 典型拓扑



5.2 具体配置

1. 配置 Initiator

1.1. 配置 device id，用来表示回传交换机的标识符

```
Switch1# configure terminal
sonic(config)# tam dev-id set 100
```

1.2. 配置一个 INT 的实体，设定角色为 initiator

```
sonic(config)# tam int int1
sonic(config-tam-int-int1)# role initiator
sonic(config-tam-int-int1)# exit
```

1.3. 配置一个 ACL table 和 rule 并设定 dscp=5 作为 INT 封包的识别符号

```
sonic(config)# access-list mirror-dscp t1 in
sonic(config-acl-mirror-dscp-t1)# access-rule r1 tam-action int1 dscp 5 63
sonic(config-acl-mirror-dscp-t1)# bind ethernet 45
sonic(config-acl-mirror-dscp-t1)# commit
```

2. 配置 Transit

2.1. 配置 device id，用来表示回传交换机的标识符

```
Switch1# configure terminal
sonic(config)# tam dev-id set 200
```

2.2. 配置一个 INT 的实体，设定角色为 Transit

```
sonic(config)# tam int int1
sonic(config-tam-int-int1)# role transit
sonic(config-tam-int-int1)# exit
```

2.3. 配置一个 ACL table 和 rule 并设定 dscp=5 作为 INT 封包的识别符号

```
sonic(config)# access-list mirror-dscp t1 in
sonic(config-acl-mirror-dscp-t1)# access-rule r1 tam-action int1 dscp 5 63
sonic(config-acl-mirror-dscp-t1)# bind ethernet 45
sonic(config-acl-mirror-dscp-t1)# commit
```

3. 配置 Terminator

3.1. 配置 device id，用来表示回传交换机的标识符

```
Switch1# configure terminal
sonic(config)# tam dev-id set 300
```

3.2. 配置一个 collector 的实体，指定控制器的 ip 和本地端口的 ip

```
sonic(config)# tam collector collector1
sonic(config-tam-collector-collector1)# dst-ip 11.1.3.3
sonic(config-tam-collector-collector1)# src-ip 3.3.3.3
sonic(config-tam-collector-collector1)# mode none
sonic(config-tam-collector-collector1)# exit
```

3.3. 配置一个 erspan 的 session，注意 destination ip 要和控制器 ip 一致，source ip 要和本地端口 ip 一致，dscp 为 1，ttl 为 254

```
sonic(config)# monitor erspan mirror1 destination 11.1.3.3 source 3.3.3.3 dscp 1 ttl 254
```

3.4. 配置一个 INT 的实体，设定角色为 Terminator，绑定 collector 和 erspan session

```
sonic(config)# tam int int1
sonic(config-tam-int-int1)# role terminator
sonic(config-tam-int-int1)# collector collector1
sonic(config-tam-int-int1)# bind erspan mirror1
sonic(config-tam-int-int1)# exit
```

3.5. 配置一个 ACL table 和 rule 并设定 dscp=5 作为 INT 封包的识别符号

```
sonic(config)# access-list mirror-dscp t1 in
sonic(config-acl-mirror-dscp-t1)# access-rule r1 tam-action int1 dscp 5 63
sonic(config-acl-mirror-dscp-t1)# bind ethernet 45
sonic(config-acl-mirror-dscp-t1)# commit
```

6 维护

下面主要介绍如何监控 INT 模块运行状态及进行相关的故障排查。

1. 查看 INT 相关配置

在 Switch 上查看：

```
sonic (config) # do show tam
Device ID:100
Collector:
 Collector     | SRC IP (only for INT) | DST IP    | DST PORT (only for MOD) | MODE
-----------------  ----------------------------  -----------  ----------------------------------  ----------
 Collector1    | 3.3.3.3                    | 11.1.3.3 |                                | NONE
INT:
 INT    | Role           | Collector
---------  ---------- ------  --------------
 int2   |
 int1   | TERMINATOR | collector1
MOD:
 State    | Collector    | Refresh Time(min) | Leave Time(min)
-----------  ----------------  --------------------------  ----------------------------
          |                |                    1 |                        10
sonic(config)#
```

Show tam 指令可以查询 INT 配置，检查下列事项

- INT role 配置是否正确
- 检查 collector IP 配置是否正确
- 检查 INT 和 collector 的绑定是否正确

2. 查看 ACL 相关配置

在 Switch 上查看：

```
sonic # show access-list
Table        Type                 Status      Binding           Stage      Rule
```

```
------------  ------------------------  ------------  ----------------------  ------------  -------------
t1            MIRROR_DSCP     Up          Ethernet45          ingress    r1
t1(TEMP)    MIRROR_DSCP     Down      Ethernet45          ingress    r1
sonic # show ac
access-list          access-rule
sonic # show access-rule
Rule        Table          Status       Priority     Action                    Match
------------  -------------  -----------  -------------  -------------------------  --------------------
r1            t1             Up                  10    TAM_ACTION: int1    DSCP: 5/63
r1            t1(TEMP)    Down               10    TAM_ACTION: int1    DSCP: 5/63
sonic #
```

Show access-list 和 show access-rule 指令可以查询 ACL 的配置，检查下列事项

- ACL stage 必须是 ingress，INT 才能正常运作
- Status 必须为 up，若不是应检查是否有绑上端口
- Action 必须是 TAM_ACTION 并绑上正确的 INT 实体

3. 查看 ERSPAN 配置

```
sonic (config) # do show monitor session
ERSPAN    Sessions
Name     Status    SRC IP DST IP GRE DSCP TTL Policer Monitor Port SRC Port Direction
----------  ---------  --------- ---------- ----- -------- ---- ----------- ------------------ ------------ -------------
mirror1    active 3.3.3.3 11.1.3.3             1 254
sonic (config) #
```

在 Switch 上查看：

Show monitor session 可以用来查询 ERSPAN 的状态，检查下列事项

- Status 必须是 active，若不是可尝试由控制器 ping Switch 来检查两者之间的联机
- DST IP 是否为控制器的 IP
- SRC IP 是否为交换机上的端口 IP