系统架构

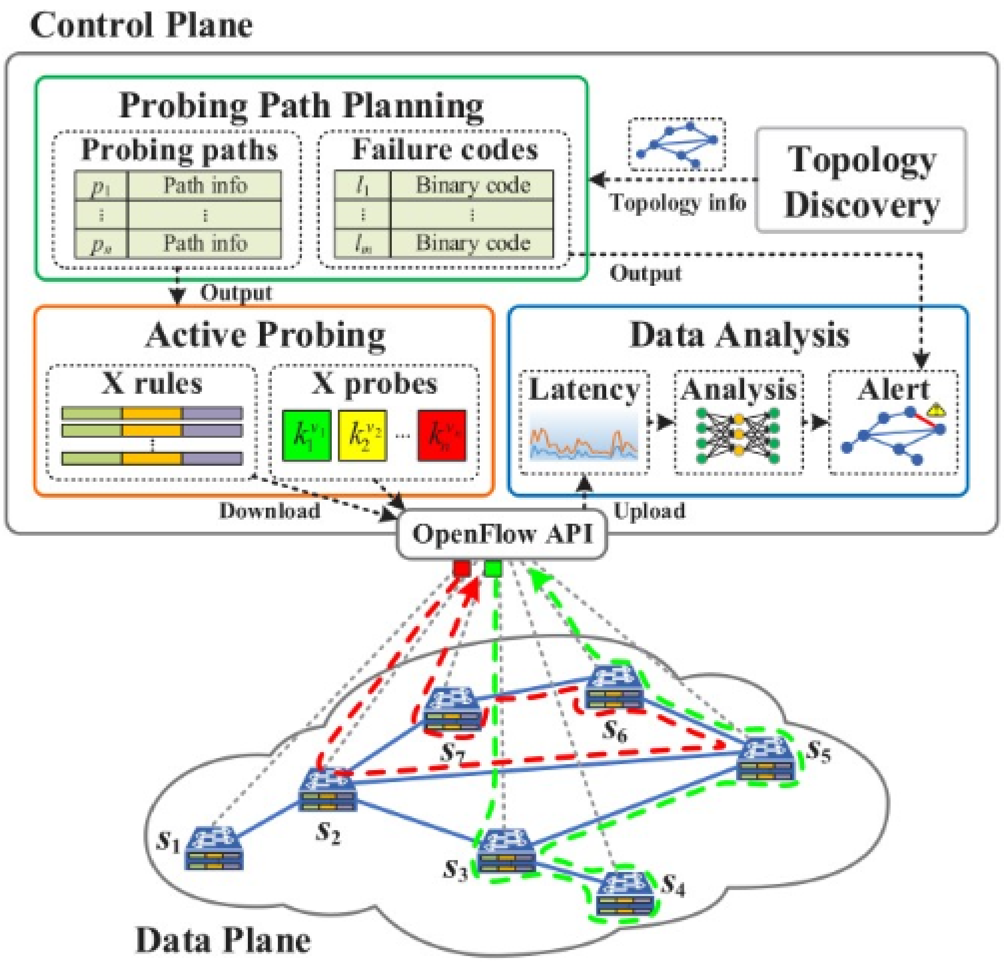


Figure 3: *XShot* architecture. It consists of three components: *probing path planning* designs a set of probing paths that start and end at the controller, *active probing* periodically sends probing packets along the paths, and *data analysis* collects the latency for transmitting probing packets and derives the failure positions.

1. ILP模型规划Cost最小的Probing Cycle, 使用错误码推断fail-stop
2. VAE根据时延推断异常path,进而使用错误码定位partial-failure

符号说明

→ 表示link(x,y)在pathi上出现的次数

表示link(x,y)是否在pathi上面

Ec: 控制器和数据平面之间的链路
Ed: 数据平面中的链路

ILP模型

Each probing cycle requires a probing packet, and they all share the same start and end point (i.e., the controller). Denote by c the centralized controller, then the probing packet cost (c_{pkt}) can be defined as below,

$$c_{pkt} = \sum_{i} \sum_{(c,y) \in E_c} e^{i}_{\overrightarrow{cy}}. \tag{1}$$

For the cost of forwarding rules, as each directed transmission requires a forwarding rule, we define c_{rule} as below,

$$c_{rule} = \sum_{i} \sum_{(x,y) \in E_d} (e^{i}_{\overrightarrow{xy}} + e^{i}_{\overrightarrow{yx}}) + \sum_{i} \sum_{(x,c) \in E_c} e^{i}_{\overrightarrow{xc}}. \tag{2}$$

P1: c->S3->S4->S3->S5->S6->c

Table 2: Forwarding rules for path p_1

Switch	Forwarding Rule	
	Match Fields	*Actions*
S_3	VLAN == $vlan_1$, inport == CONTR[a]	output = $port_3^4$
	VLAN == $vlan_1$, inport == $port_3^4$	output = $port_3^5$
S_4	VLAN == $vlan_1$, inport == $port_4^3$	output = INPORT
S_5	VLAN == $vlan_1$, inport == $port_5^3$	output = $port_5^6$
S_6	VLAN == $vlan_1$, inport == $port_6^5$	output = CONTR

[a]CONTR represents CONTROLLER.

(其中c_rule刚好是path的link数目-1)

Objective:

$$\min \quad \omega \times c_{pkt} + c_{rule} \tag{3}$$

where ω is a weight used to balance the cost c_{pkt} and c_{rule}. In general, $\omega \geq 1$ because the probing packets must be periodically sent to track the network status all the time whereas the forwarding rules are statically installed.

Subject to:

$$e^i_{\overrightarrow{xy}} \leq 1, \quad \forall i, \forall (x,y) \in E \tag{4}$$

$$e^i_{\overrightarrow{yx}} \leq 1, \quad \forall i, \forall (x,y) \in E \tag{5}$$

$$\sum_{(c,y)\in E_c} e^i_{\overrightarrow{cy}} \leq 1, \quad \forall i \tag{6}$$

$$\sum_{(x,c)\in E_c} e^i_{\overrightarrow{xc}} \leq 1, \quad \forall i \tag{7}$$

$$\sum_{(x,y)\in E} (e^i_{\overrightarrow{xy}} - e^i_{\overrightarrow{yx}}) = 0, \quad \forall i, \forall x \in V \tag{8}$$

$$e^i_{\overrightarrow{xy}} \leq q^i_{\overrightarrow{xy}} \leq \varphi \times e^i_{\overrightarrow{xy}}, \quad \forall i, \forall (x,y) \in E \tag{9}$$

$$e^i_{\overrightarrow{yx}} \leq q^i_{\overrightarrow{yx}} \leq \varphi \times e^i_{\overrightarrow{yx}}, \quad \forall i, \forall (x,y) \in E \tag{10}$$

$$\sum_{(x,y)\in E} (q^i_{\overrightarrow{xy}} - q^i_{\overrightarrow{yx}}) \geq \frac{1}{\varphi} \sum_{(x,y)\in E} e^i_{\overrightarrow{xy}}, \quad \forall i, \forall x \in V \tag{11}$$

$$\frac{1}{2}(e^i_{\overrightarrow{xy}} + e^i_{\overrightarrow{yx}}) \leq l^i_{xy} \leq e^i_{\overrightarrow{xy}} + e^i_{\overrightarrow{yx}}, \quad \forall i, \forall (x,y) \in E \tag{12}$$

$$\sum_i l^i_{xy} \geq 1, \quad \forall (x,y) \in E \tag{13}$$

$$\frac{1}{2}(l^i_{xy} + l^i_{ab}) \leq h^i_{xy,ab} \leq l^i_{xy} + l^i_{ab}, \quad \forall i, \forall (x,y) \neq (a,b) \tag{14}$$

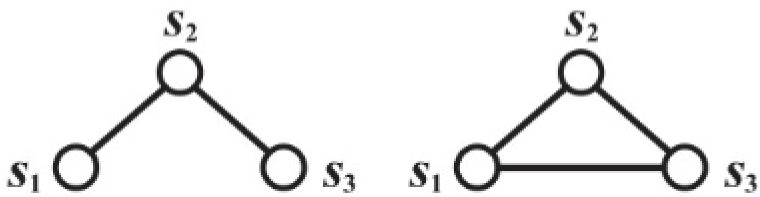
$$\sum_i (2h^i_{xy,ab} - l^i_{xy} - l^i_{ab}) \geq 1, \quad \forall (x,y) \neq (a,b) \tag{15}$$

$$l^i_{xy}, l^i_{ab}, h^i_{xy,ab} \in \{0,1\}, \quad \forall i, \forall (x,y) \neq (a,b) \tag{16}$$

$$e^i_{\overrightarrow{xy}}, e^i_{\overrightarrow{yx}}, q^i_{\overrightarrow{xy}}, q^i_{\overrightarrow{yx}} \in Z^{\geq}, \quad \forall i, \forall (x,y) \in E \tag{17}$$

$$i \in \{1, 2, \ldots, \|E\|\}. \tag{18}$$

现有的方法中的probing cycle不允许穿过一条link多次，最多只能穿过一次。会导致one-cut和two-cut结构出问题



(a) One-cut structure (b) Two-cut structure

Figure 2: One-cut and two-cut network structure.

(4)-(5) link(x,y)在一条path的某个方向上最多出现一次

(6)-(7) link(c,y)在所有path的某个方向上最多出现一次

(8) 保证出入守恒, 从某个交换机出去的流的数目等于进入交换机流的数目

(9)-(11)

(12) 限制l和e之间的关系, lxy是1的时候, exy和eyx至少有一个是1, lxy是0的时候exy和eyx都是0

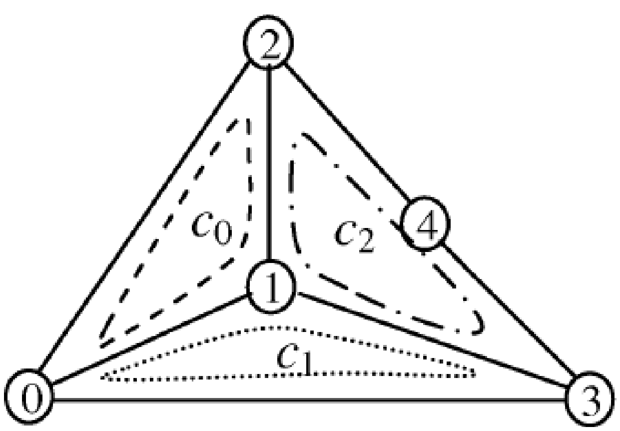
(13) 每个link至少在某条path上出现过一次, 不然就是没有被探测到

(14) 就是表示了h和l之间的关系, 直接用h12 = l1||l2的形式应该也可以

(15) 确保每条链路有唯一的错误码

本质上是任意两个错误码相减的绝对值=1,参考下面的例子

$$\sum_i (2h^i_{xy,ab} - l^i_{xy} - l^i_{ab}) \geq 1, \quad \forall (x,y) \neq (a,b) \qquad (15)$$



(a)

Link	c_2	c_1	c_0	Decimal
(0,1)	0	1	1	3
(0,2)	0	0	1	1
(0,3)	0	1	0	2
(1,2)	1	0	1	5
(1,3)	1	1	0	6
(2,4)	1	0	0	4
(3,4)	1	0	0	4

(b)

(16)-(18)是定义域的限制

mi: 被i条path穿过的控制链路的数目(正向和反向算一个)

1条: m1

2条: m2

n条: mn

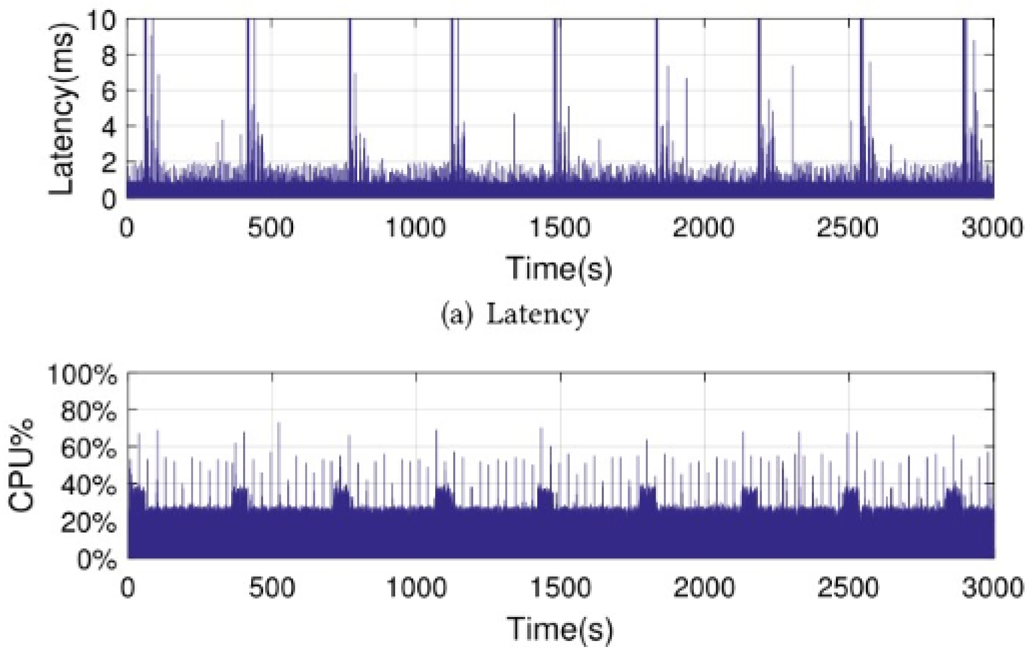
T(mi) 这mi条link被path穿过的次数

结论
n是probing path的数目:

–

流程

1. 构造探测包: 使用vlan-id来区分不同路径上的探测包
2. 定位fail-stop故障, 发包间隔在分钟级别。定位partial-failure故障, 发包间隔在ms级别。
3. 记录延迟, 延迟初始值为0, 收到包之后会更新延迟, 在下一次发包之前如果延迟还是0, 那么说明上一轮探测出现fail-stop故障, 使用错误码定位。
4. Donut VAE模型来识别high-latency导致的故障, 获得异常路径后进而使用错误码定位partial-failure
5. 在实验中发现latency抖动的噪音影响partial-failure, 所以连续两次发探测包的结果一致才会认定为partial-failure.



(a) Latency

(b) CPU usage

Figure 5: Time-varying fluctuations in terms of latency and CPU usage.

场景

单向故障: 比如单向的拥塞, Xshot无法检测到。但是也可以两个方向发双倍的包来进行检测, 但是占用的资源会更多一些
交换机故障判定: 假设link都是正常的, 然后使用0-1来表示交换机
多故障: Xshot无法判定
持续时间很短的故障: Xshot无法判定, 但是也无妨, 因为持续时间太短的故障很快就恢复了