دفاع پایان‌نامه کارشناسی ارشد

دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

مدیریت ظرفیت جداول جریان در شبکه‌های نرم‌افزار محور

ارائه‌دهنده: محمد صابری

استاد راهنما اول: جناب آقای دکتر موقر

استاد راهنما دوم: جناب آقای دکتر مهدی دولتی

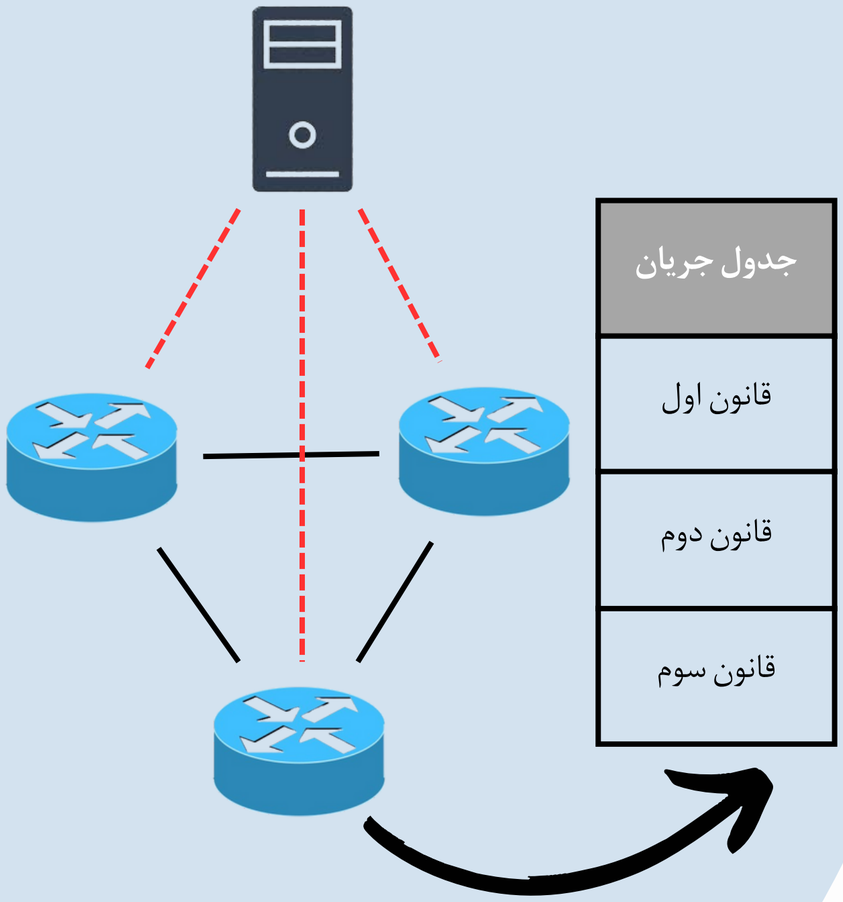
داوران: جنابان آقای دکتر جهانگیر و آقای دکتر حسینی

دی ماه ۱۴۰۳

نمای کلی

- مقدمه
- مدل سیستم و تعریف مسئله
- روش پیشنهادی
- نتایج
- نتیجه‌گیری

مقدمه



قانون:	عمل	اولویت	انطباق

۱ شبکه‌های نرم‌افزار محور

۲ قوانین ارسال

۳ جداول جریان

۴ چالش موجود در شبکه‌های نرم‌افزار محور

مقدمه

مدیریت جداول جریان

تقسیم و توزیع
Cooperative Caching [3]

[3] O. Rottenstreich, A. Kulik et al., “Data plane cooperative caching with dependencies,” IEEE Trans. Netw. Serv. Manag., vol. 19, no. 3, pp. 2092–2106, Sep. 2021.

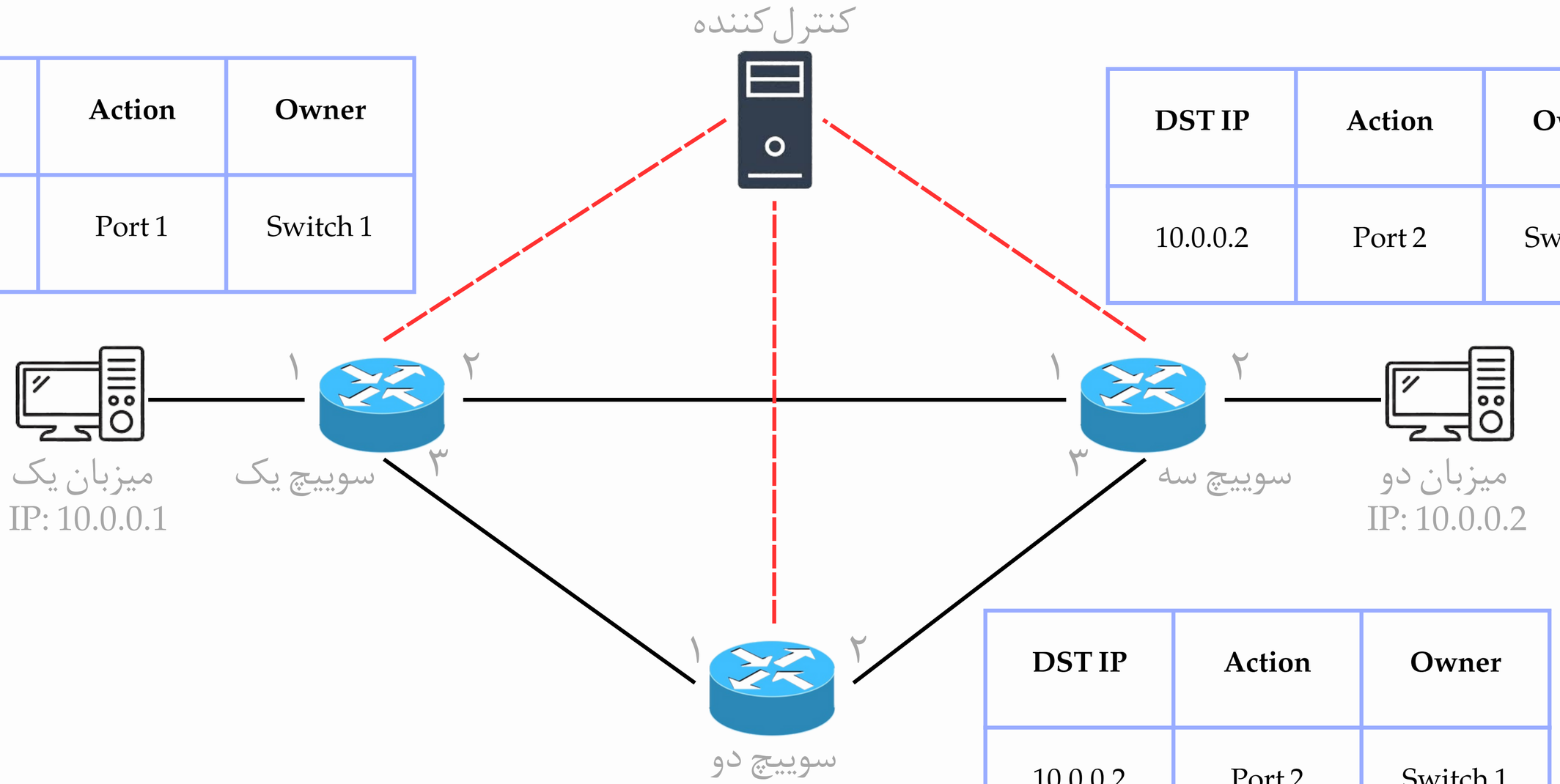
فشرده‌سازی
Wildcard Rules Caching [2]

[2] J.-P. Sheu and Y.-C. Chuo, “Wildcard rules caching and cache replacement algorithms in software-defined networking,” IEEE Trans. Netw. Serv. Manag., vol. 13, no. 1, pp. 19–29, Mar. 2016.

اخراج
Flowmaster [1]

[1] K. Kannan and S. Banerjee, “Flowmaster: Early eviction of dead flow on SDN switches,” in Proc. ICDCN, 2014, pp. 484–498.

مدل سیستم و تعریف مسئله

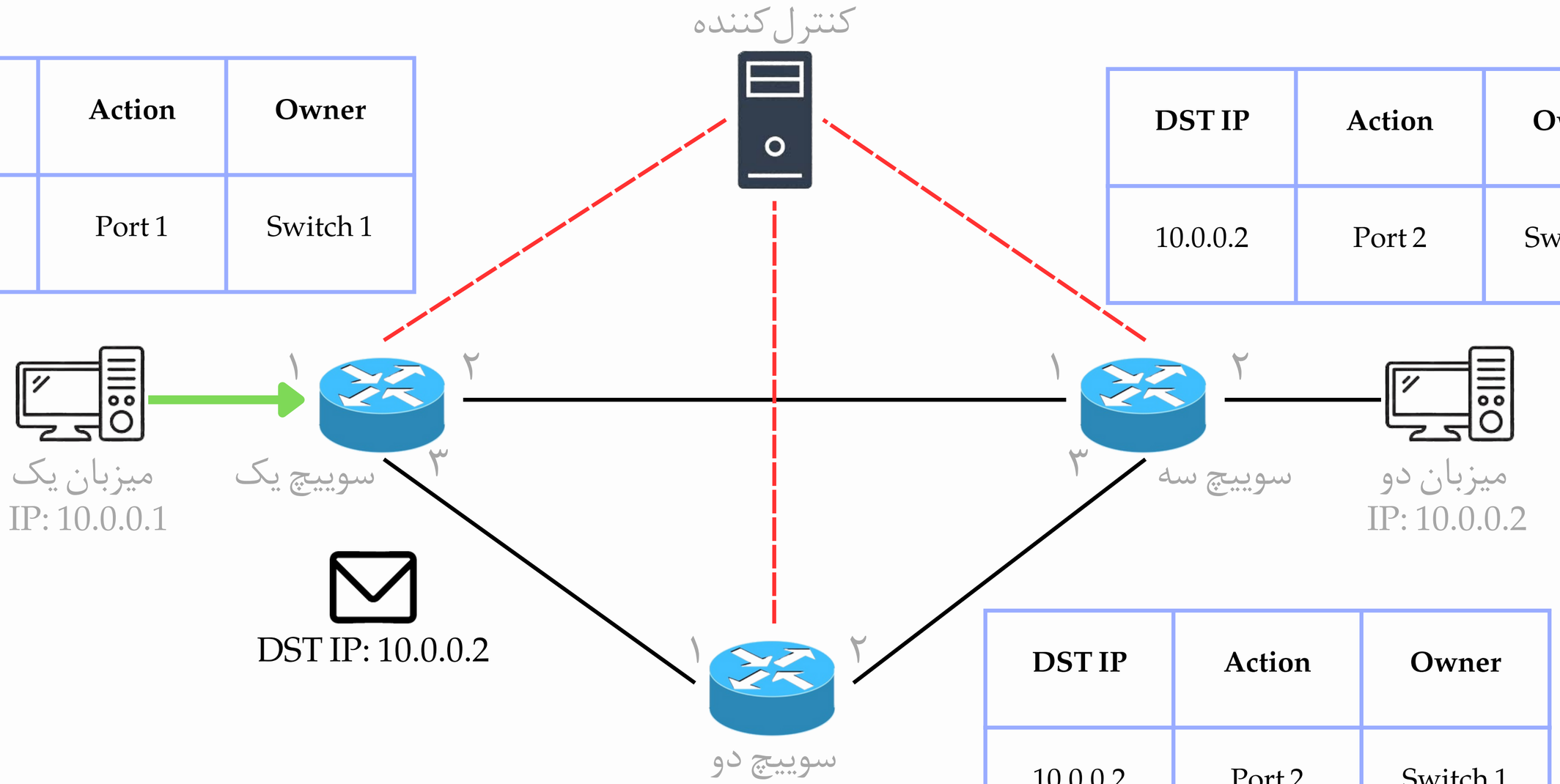


DST IP	Action	Owner
10.0.0.1	Port 1	Switch 1

DST IP	Action	Owner
10.0.0.2	Port 2	Switch 3

DST IP	Action	Owner
10.0.0.2	Port 2	Switch 1

مدل سیستم و تعریف مسئله

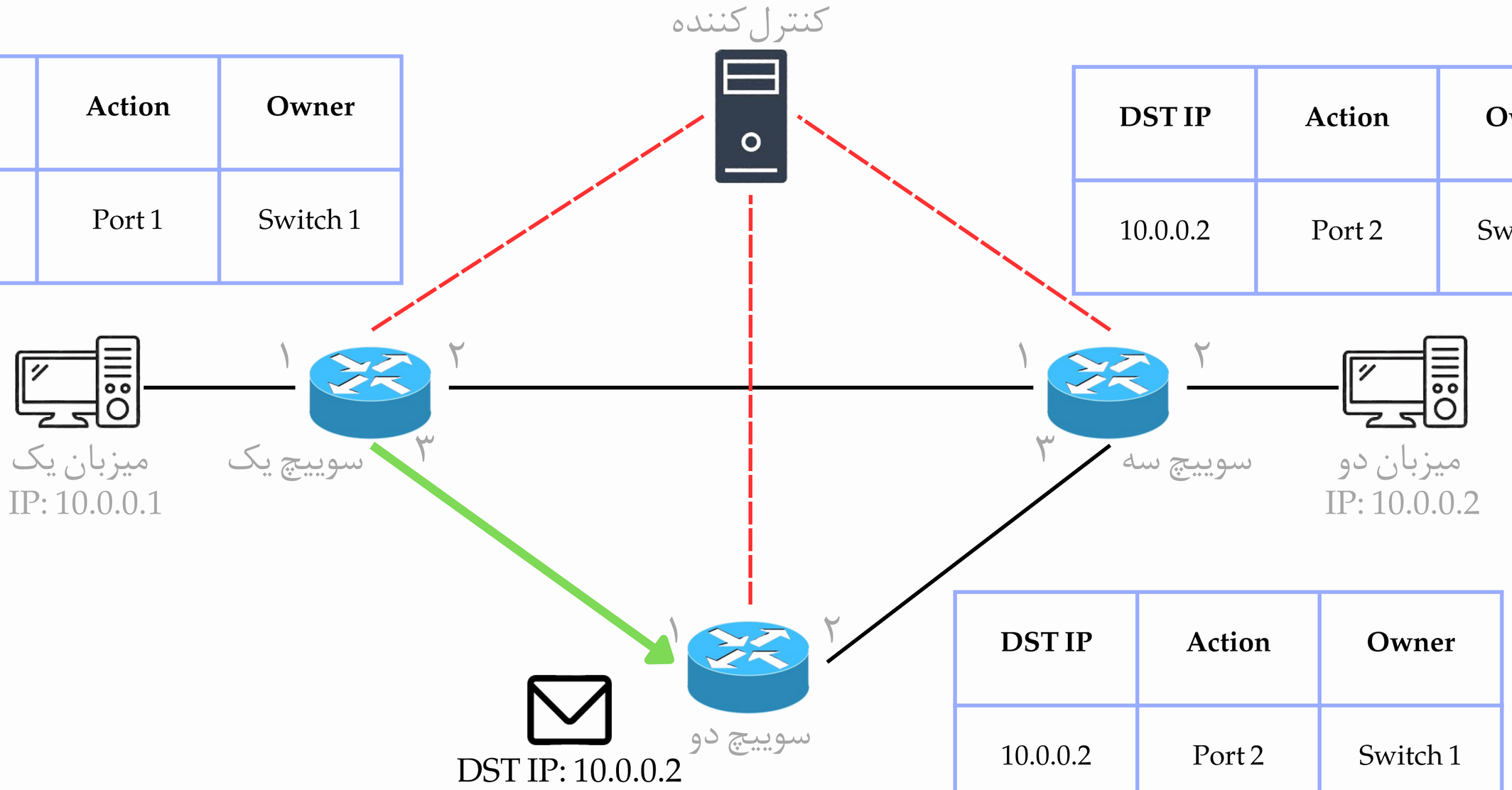


DST IP	Action	Owner
10.0.0.1	Port 1	Switch 1

DST IP	Action	Owner
10.0.0.2	Port 2	Switch 3

DST IP	Action	Owner
10.0.0.2	Port 2	Switch 1

مدل سیستم و تعریف مسئله

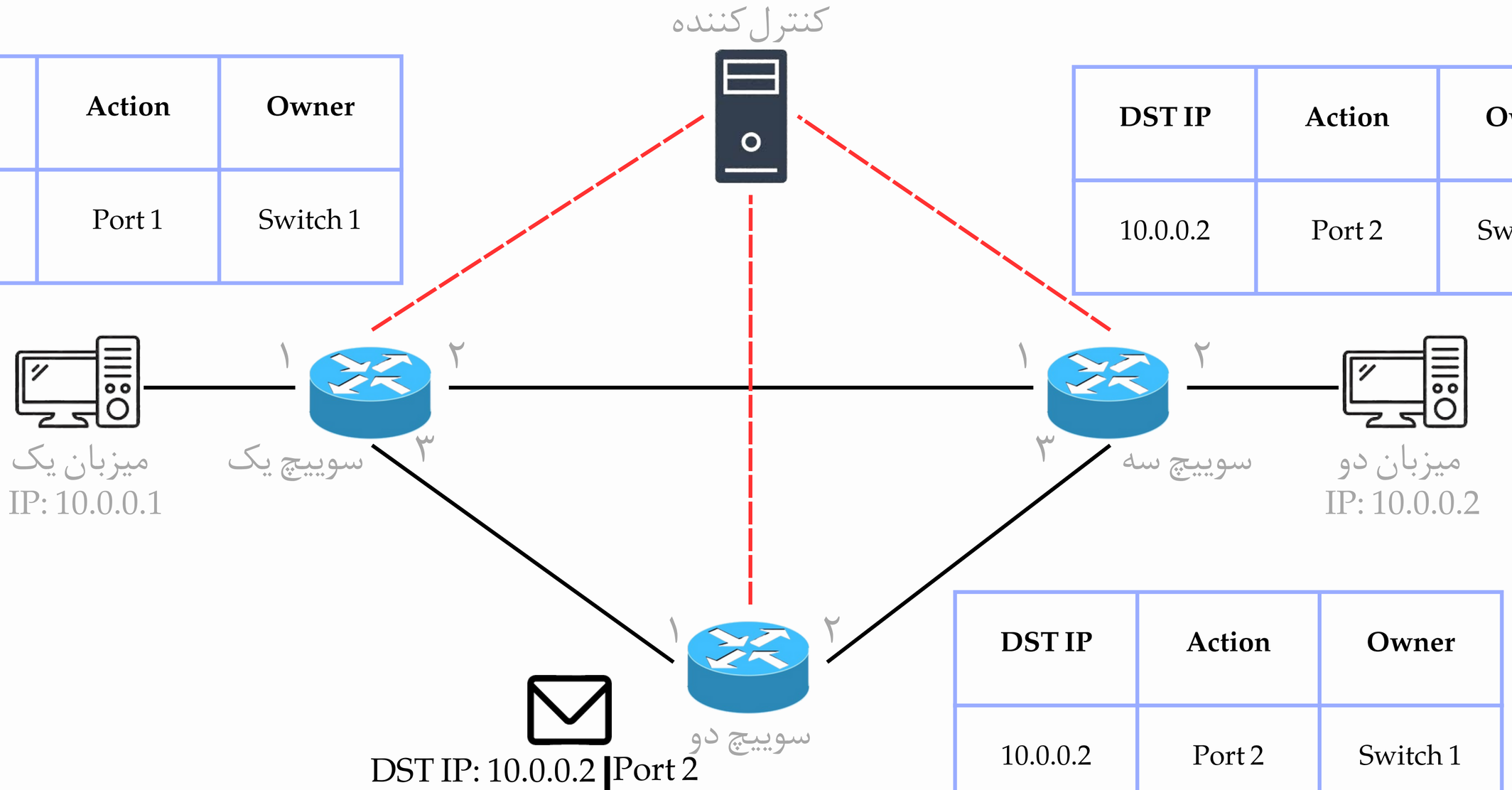


DST IP	Action	Owner
10.0.0.1	Port 1	Switch 1

DST IP	Action	Owner
10.0.0.2	Port 2	Switch 3

DST IP	Action	Owner
10.0.0.2	Port 2	Switch 1

مدل سیستم و تعریف مسئله



DST IP	Action	Owner
10.0.0.1	Port 1	Switch 1

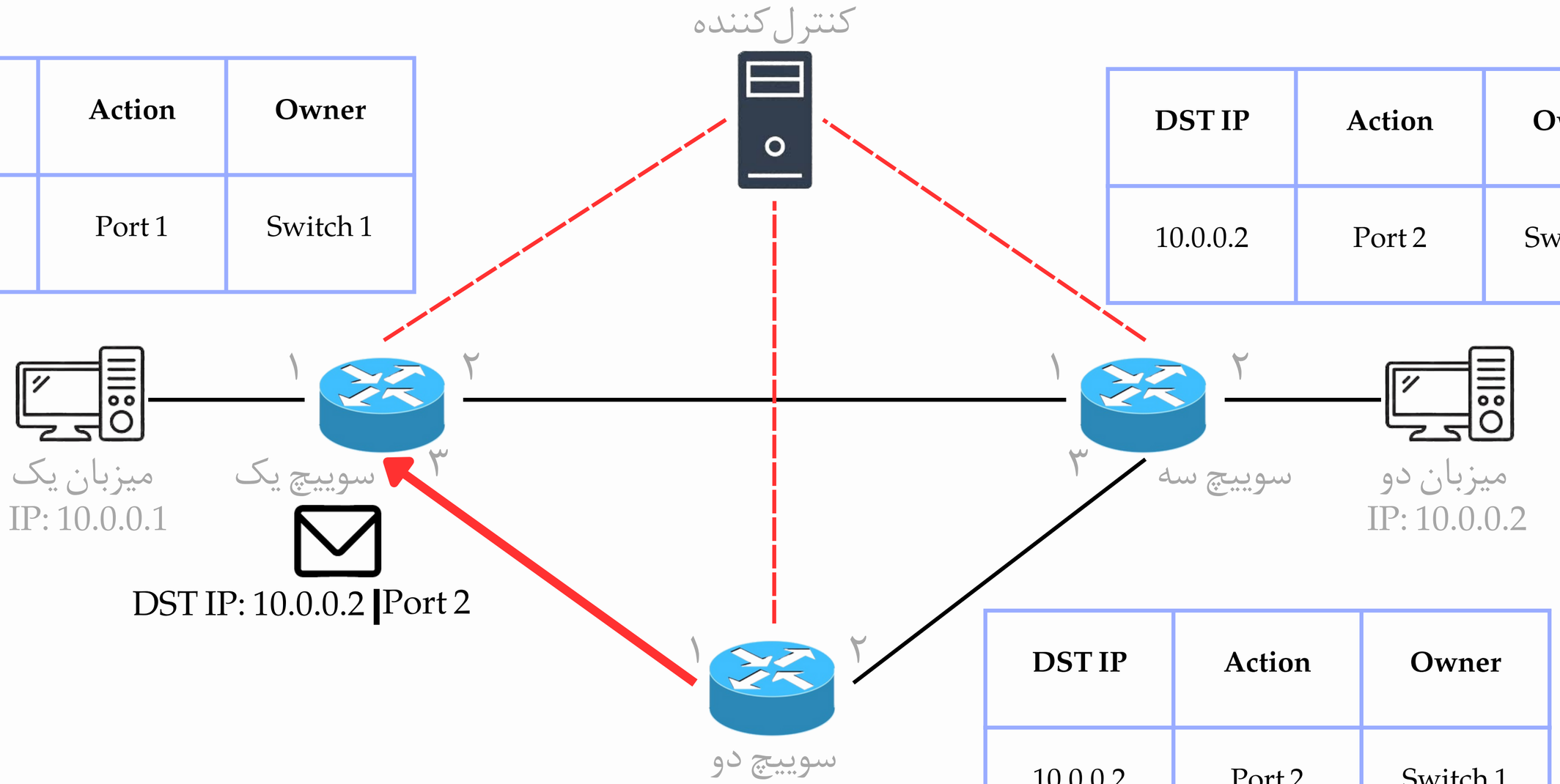
DST IP	Action	Owner
10.0.0.2	Port 2	Switch 3

DST IP	Action	Owner
10.0.0.2	Port 2	Switch 1

مدل سیستم و تعریف مسئله

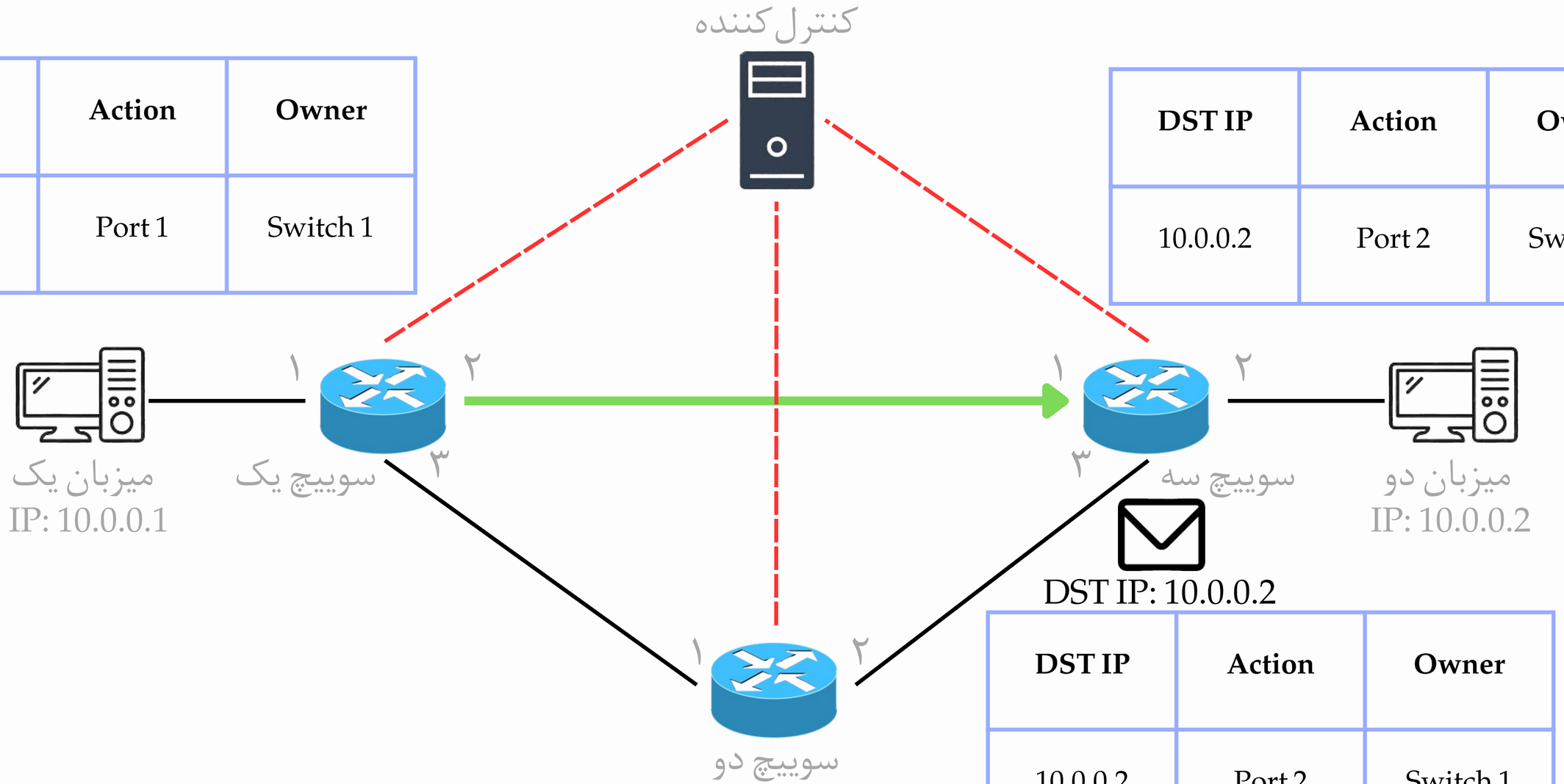
DST IP	Action	Owner
10.0.0.1	Port 1	Switch 1

DST IP	Action	Owner
10.0.0.2	Port 2	Switch 3



DST IP	Action	Owner
10.0.0.2	Port 2	Switch 1

مدل سیستم و تعریف مسئله



DST IP	Action	Owner
10.0.0.1	Port 1	Switch 1

DST IP	Action	Owner
10.0.0.2	Port 2	Switch 3

DST IP	Action	Owner
10.0.0.2	Port 2	Switch 1

مدل سیستم و تعریف مسئله

مجموعه قوانین: $R = \{r_1, r_2, \dots, r_n\}$

مجموعه سوییچ‌ها: $S = \{s_1, s_2, \dots, s_i\}$

ظرفیت جداول جریان سوییچ‌ها: $M = \{m_1, m_2, \dots, m_i\}$

نرخ جریان‌ها: $\rho = \{\rho_1, \rho_2, \dots, \rho_n\}$

سوییچ که از قانون r را استفاده می‌کند: w_r

مدل سیستم و تعریف مسئله

زمان بازیابی قانون:

- زمان بازیابی قانون از حافظه سوییچ اصلی: d_1
- زمان بازیابی قانون از جفت سوییچ: d_2
- زمان بازیابی قانون از کنترل کننده: d_3

متغیرهای تصمیم گیری:

- نشان دادن اینکه آیا قانون r در سوییچ w قرار گرفته است یا خیر: $y_{w,r}$
- نشان می دهد که آیا سوییچ v قوانین را برای سوییچ w ذخیره می کند یا خیر: $z_{w,v}$

مدل سیستم و تعریف مسئله

هدف: به حداقل رساندن متوسط تأخیر بازیابی قانون

$$min \sum_{r \in R} \rho_r . \{d_1 . y_{w_r,r} + \sum_{v \in S - \{w_r\}} d_2 . y_{v,r} + d_3 . (1 - \sum_{v \in S} y_{v,r})\}$$

محدودیت تضمین می کند که هر سوییچ w دقیقاً یک جفت دارد:

$$s.t. \sum_{v \in S_w} z_{w,v} = 1, \; w \in S$$

مدل سیستم و تعریف مسئله

محدودیت تضمین می‌کند که هر قانون r حداکثر در یک سوییچ قرار می‌گیرد:

$$\sum_{w \in S} y_{w,r} \leq 1 , r \in R$$

محدودیت اعمال می‌کند که هر قانون r فقط می‌تواند در w_r یا جفت آن ذخیره شود:

$$y_{v,r} \leq 1_{v=w_r} + z_{w_r,v} , r \in R , v \in S_{w_r}$$

مدل سیستم و تعریف مسئله

محدودیت تضمین می کند که ظرفیت حافظه سوییچ‌ها رعایت می‌شود:

$$\sum_{r \in R} q_r . y_{w,r} \leq M_v, \; v \in S$$

محدودیت تضمین می کند که اگر قانون r در یک سوییچ قرار گیرد، تمام قوانین وابسته آن نیز در آنجا قرار می گیرند:

$$y_{v,r} \leq y_{v,r'}, \; r \in R, \; r' \in R_r, \; v \in S$$

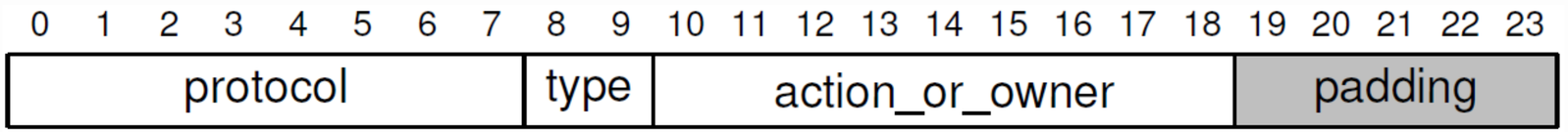
روش پیشنهادی

طراحی پی‌فور: از دو جدول برای ذخیره داده‌ها و سوییچ‌های جفت‌شده استفاده می‌کنیم:

```
1  table cache_t {
2    key = {
3      cache_h.action_or_owner:
               exact;
4      ipv4.dstAddr: lpm;
5    }
6    actions = {
7      get_self_action;
8      get_cached_action;
9      NoAction;
10   }
11 }
```

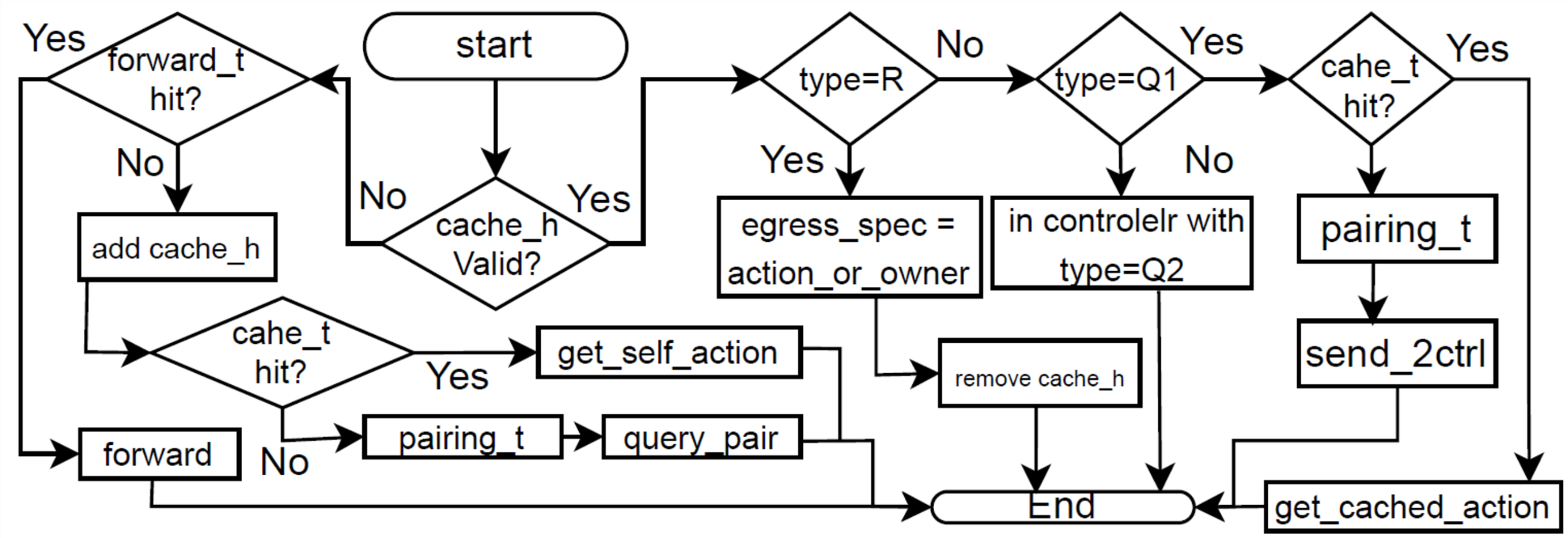
```
12 table pairing_t {
13   key = {
14     cache_h.type: exact;
15   }
16   actions = {
17     query_pair;
18     send_2ctrl;
19     NoAction;
20   }
21 }
```

تعریف سرآیند بسته‌ای به نام cache_h :



روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده



روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

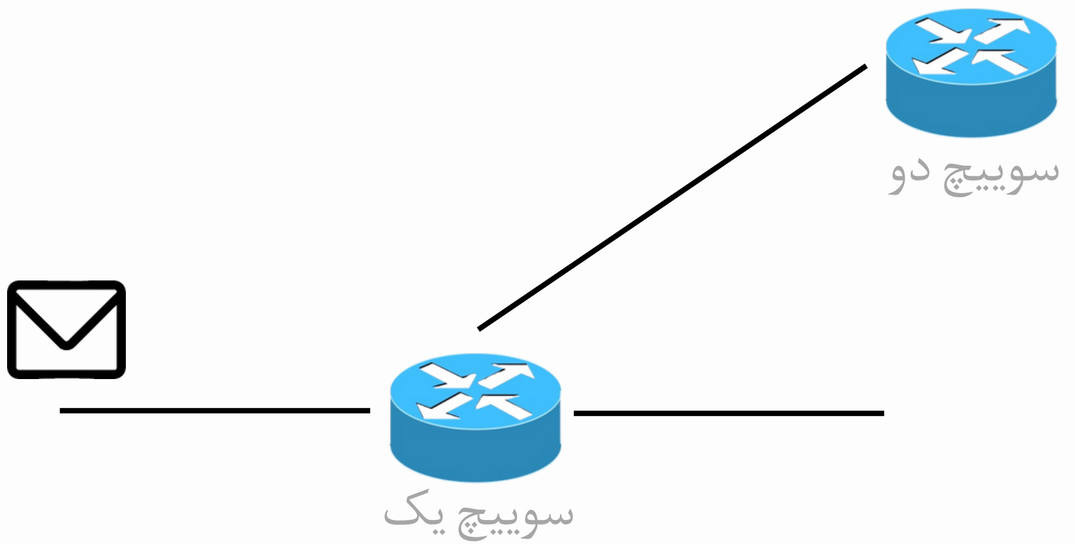


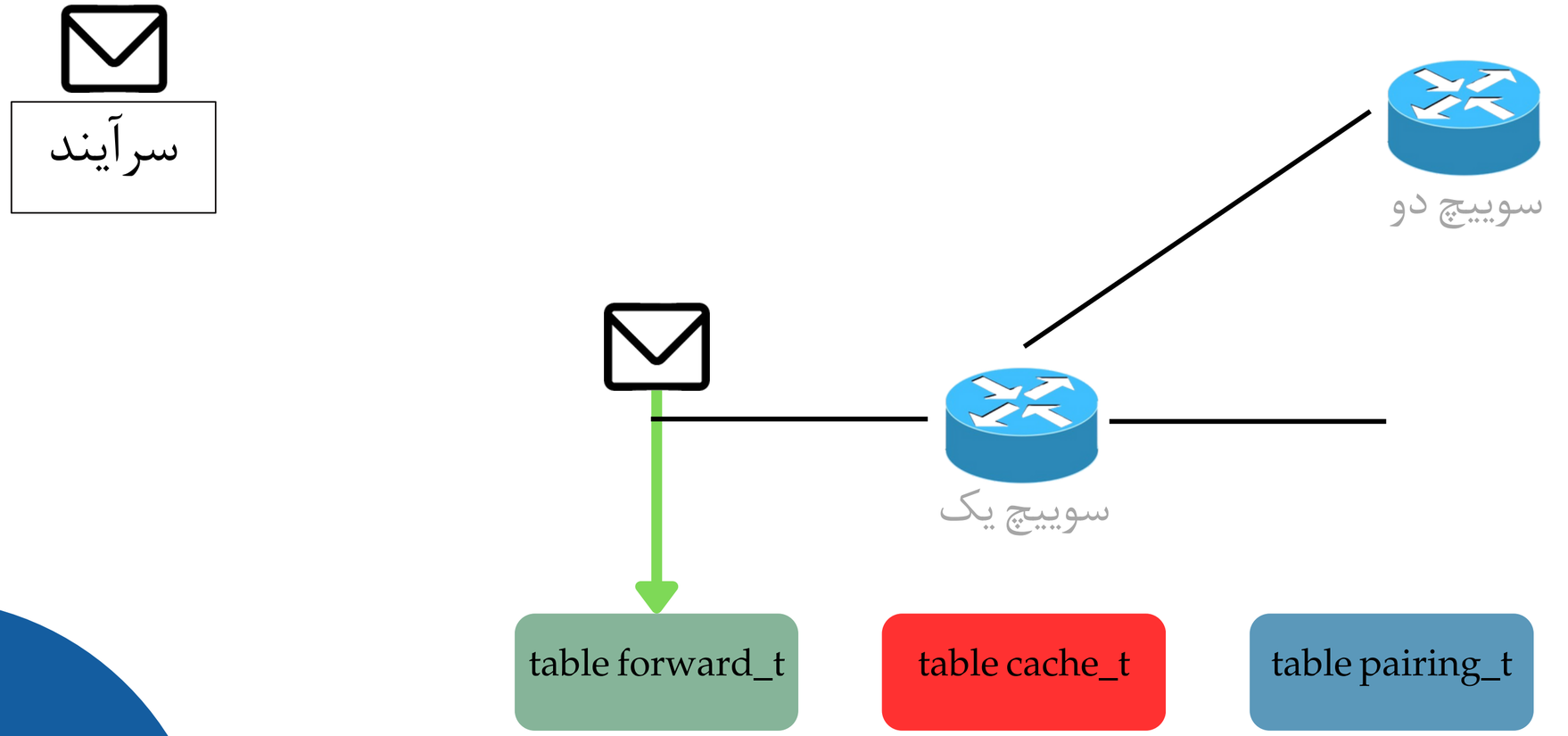
table forward_t

table cache_t

table pairing_t

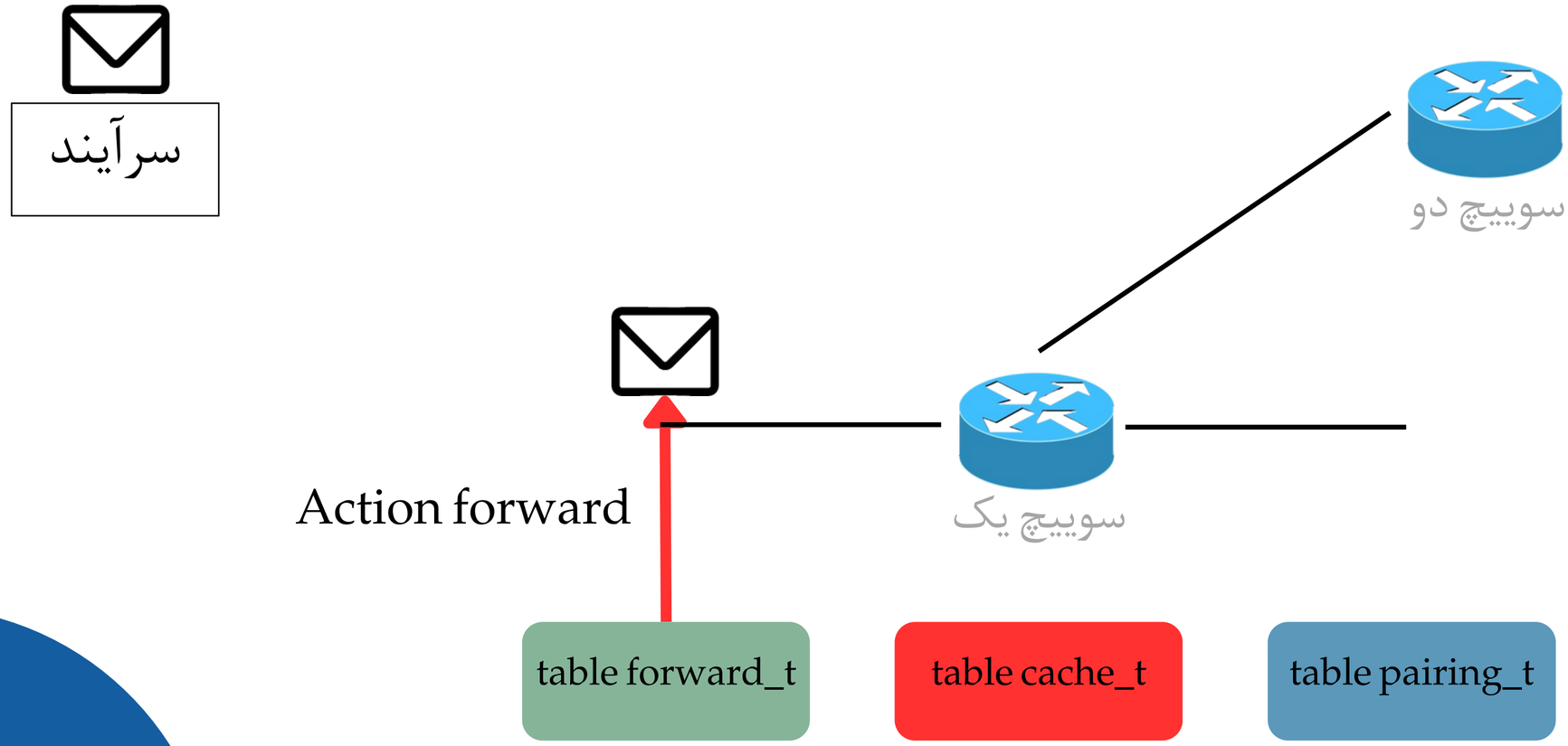
روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده



روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده



روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

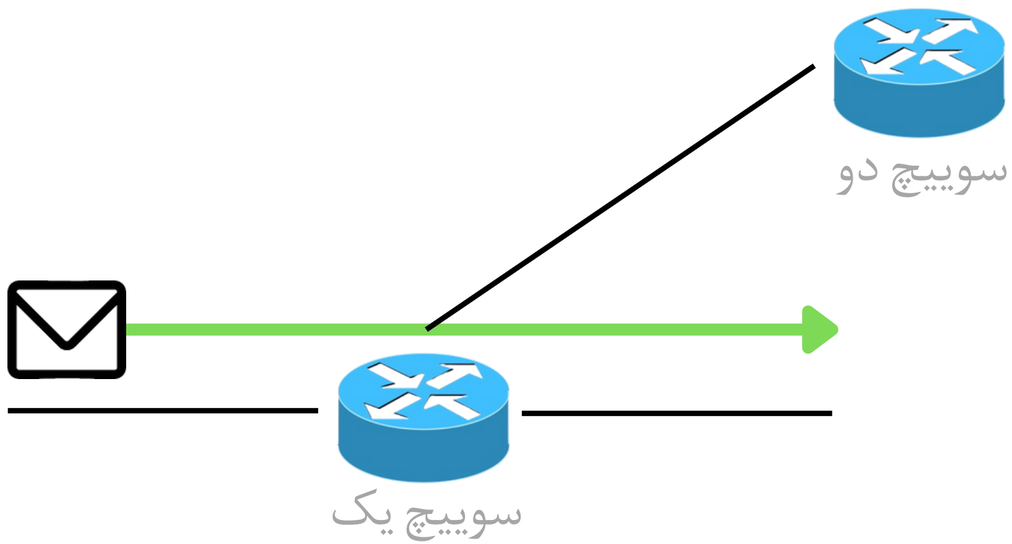


table forward_t

table cache_t

table pairing_t

روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

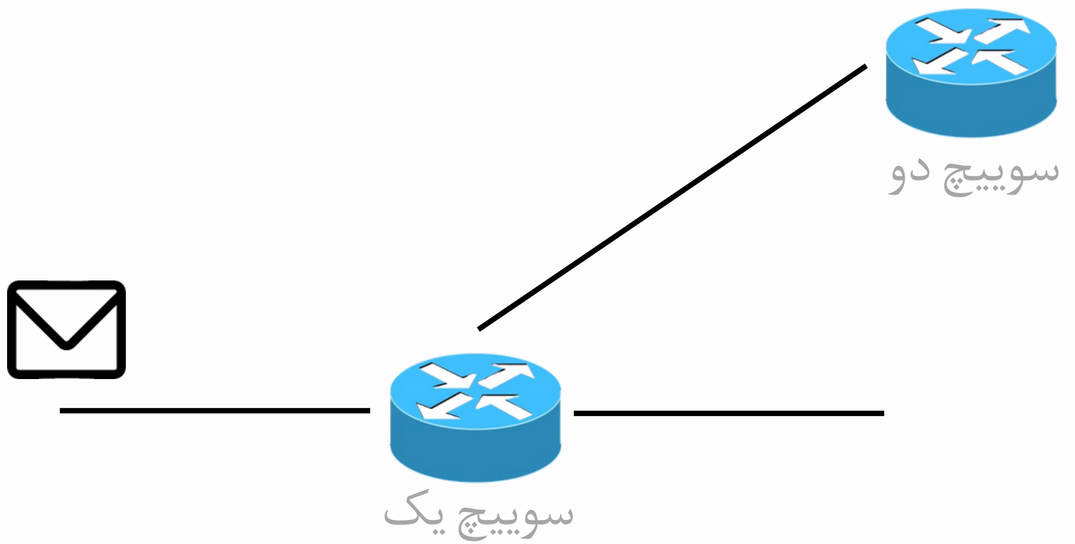


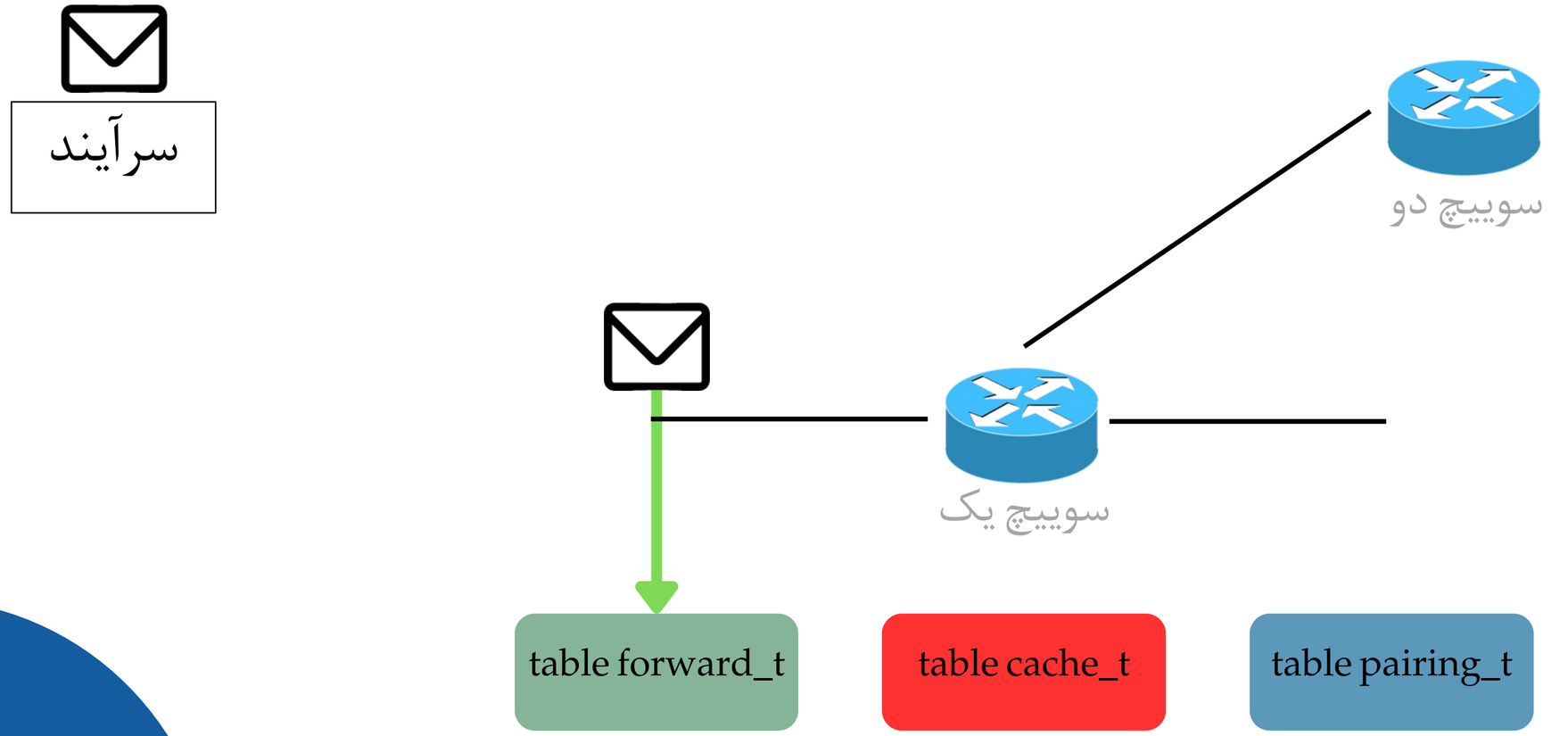
table forward_t

table cache_t

table pairing_t

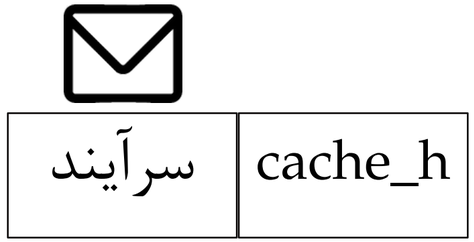
روش پیشنهادی

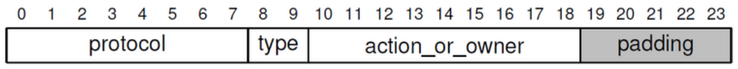
طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

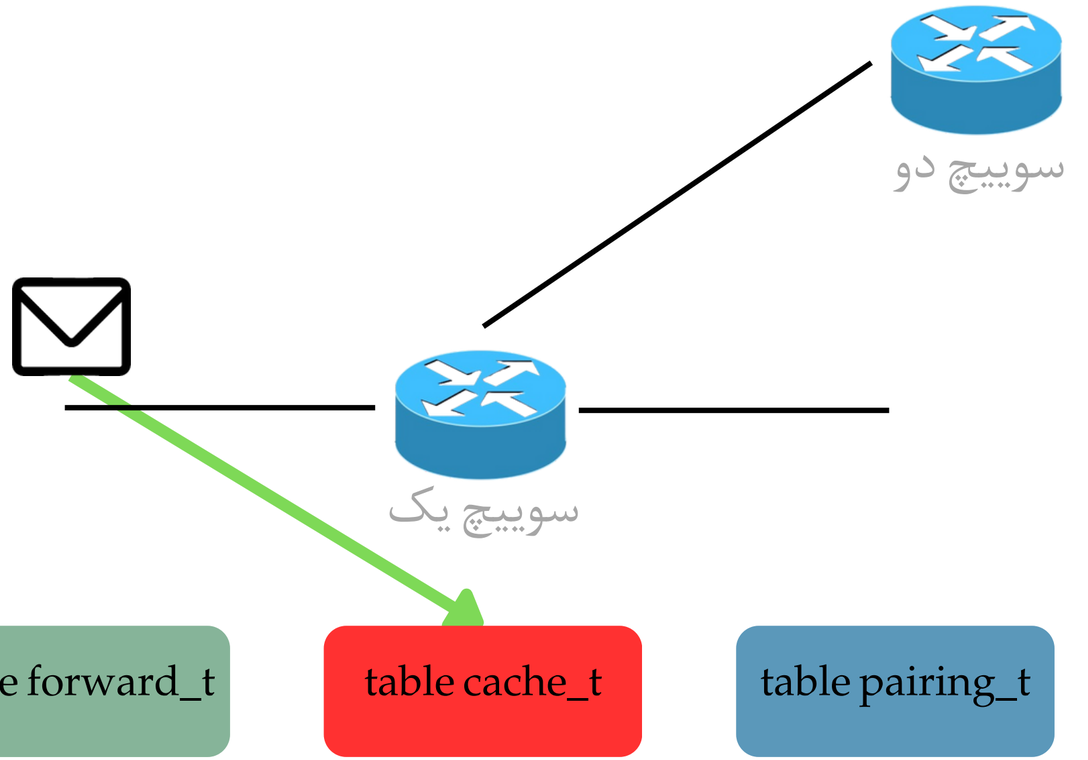


روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

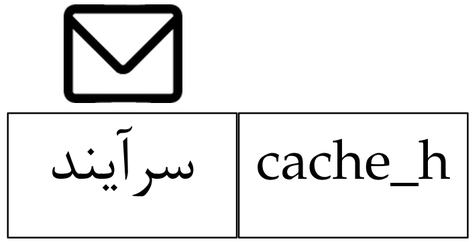


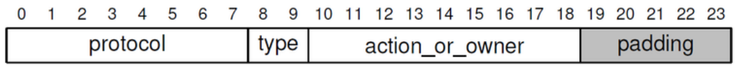


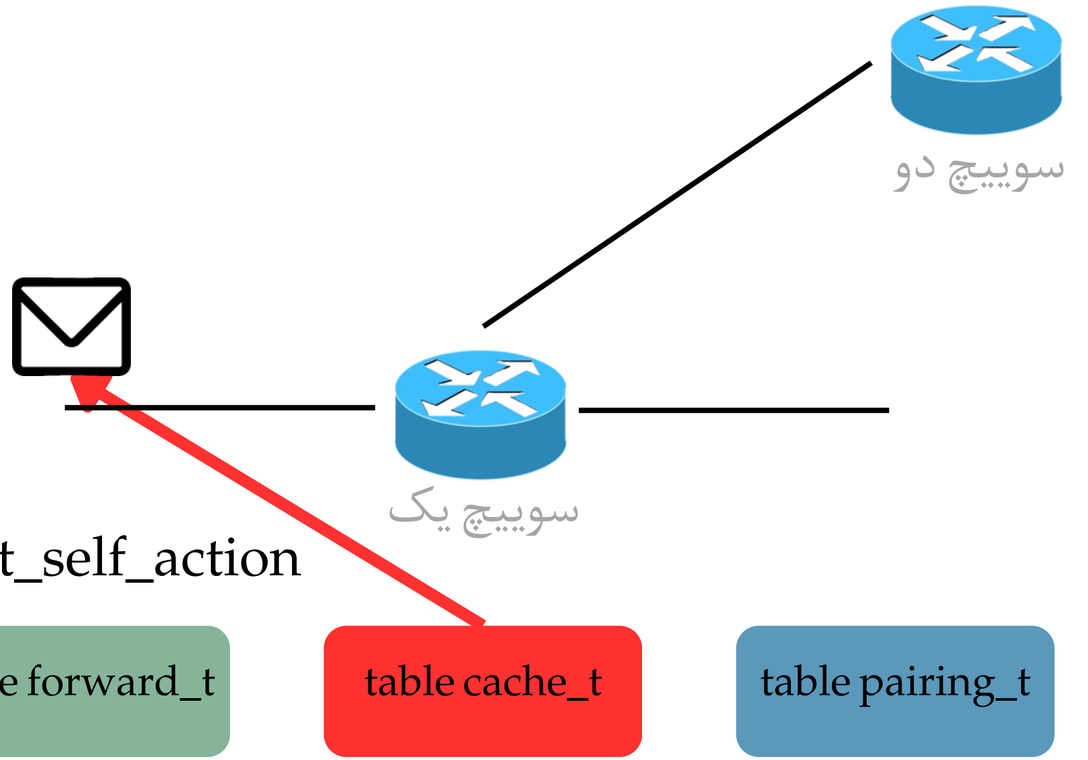


روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

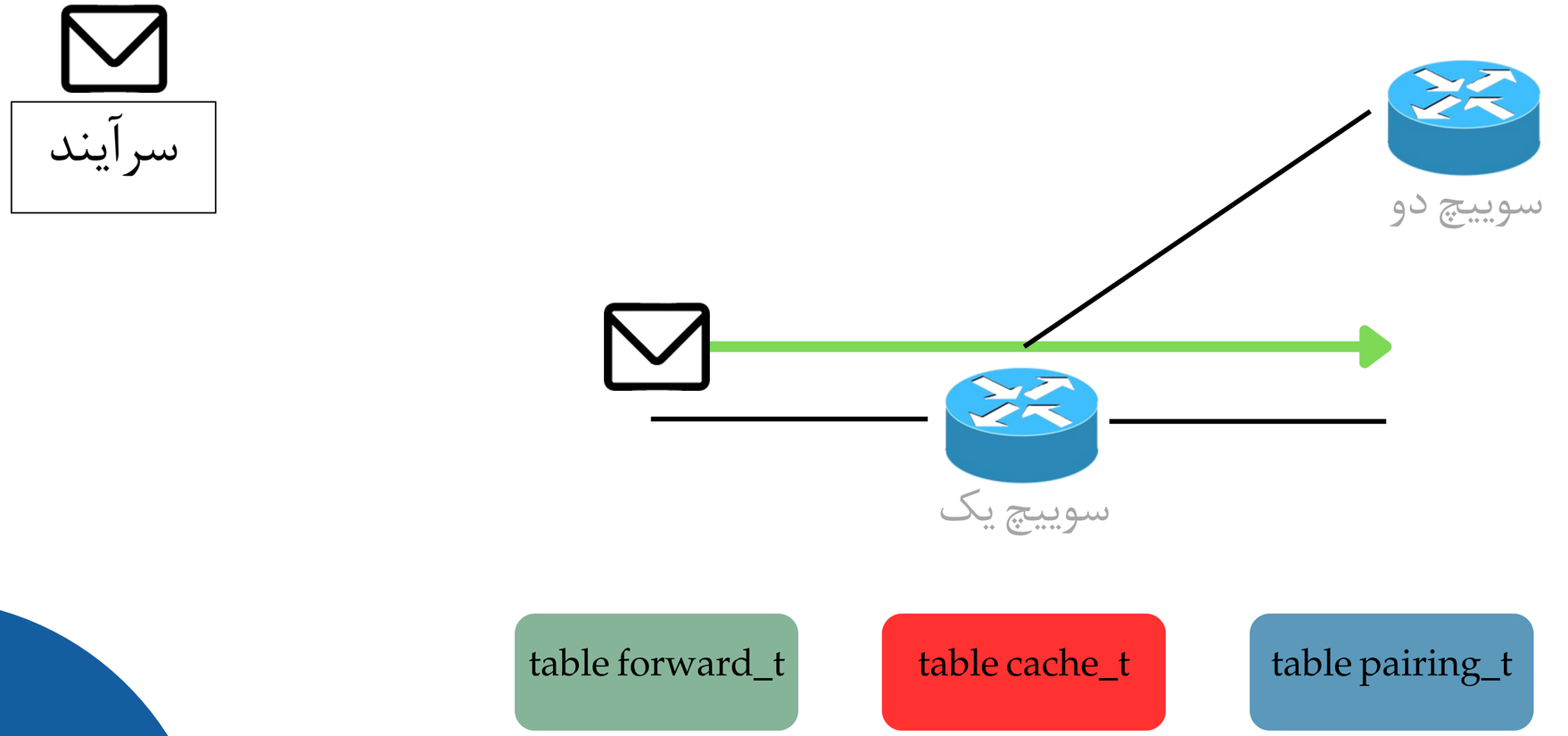






روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده



روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

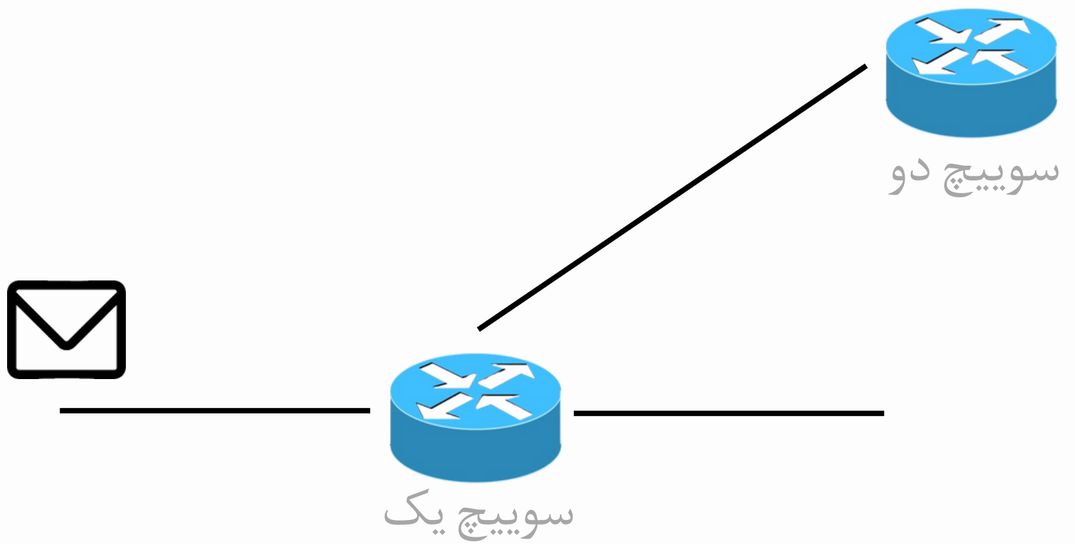


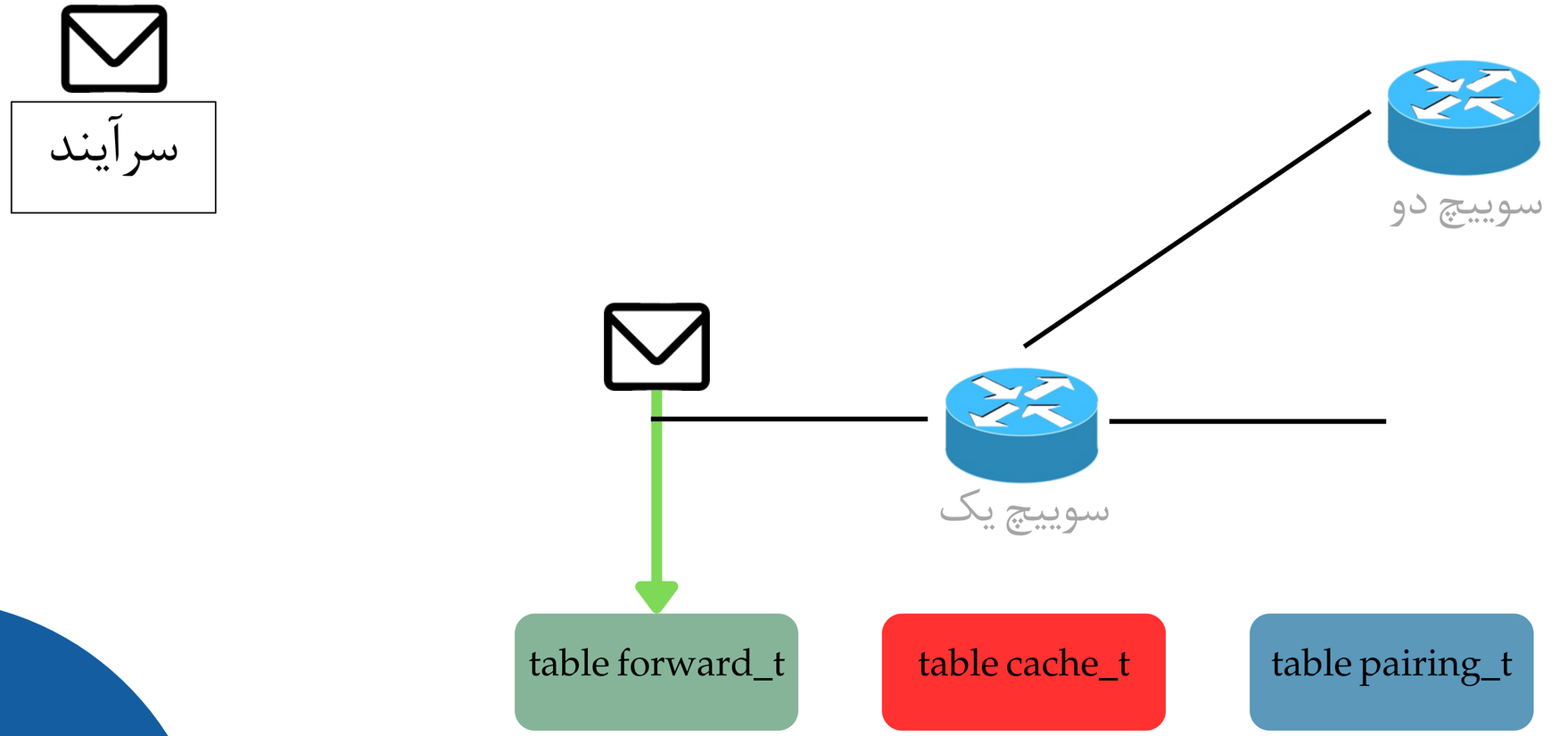
table forward_t

table cache_t

table pairing_t

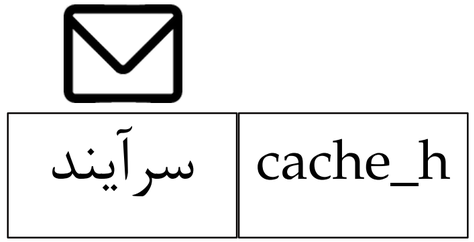
روش پیشنهادی

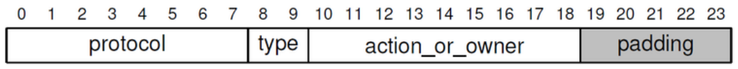
طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

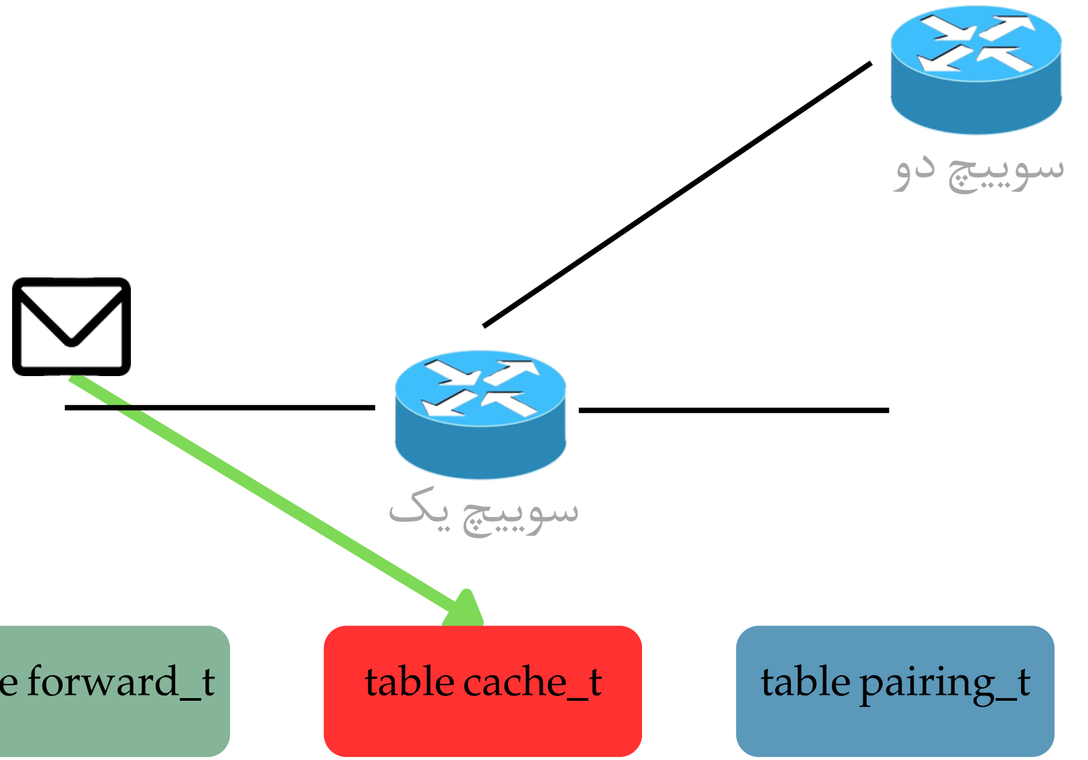


روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

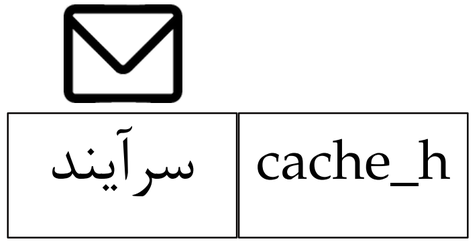


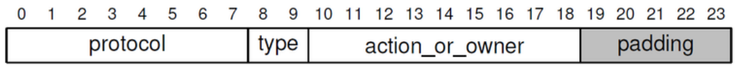


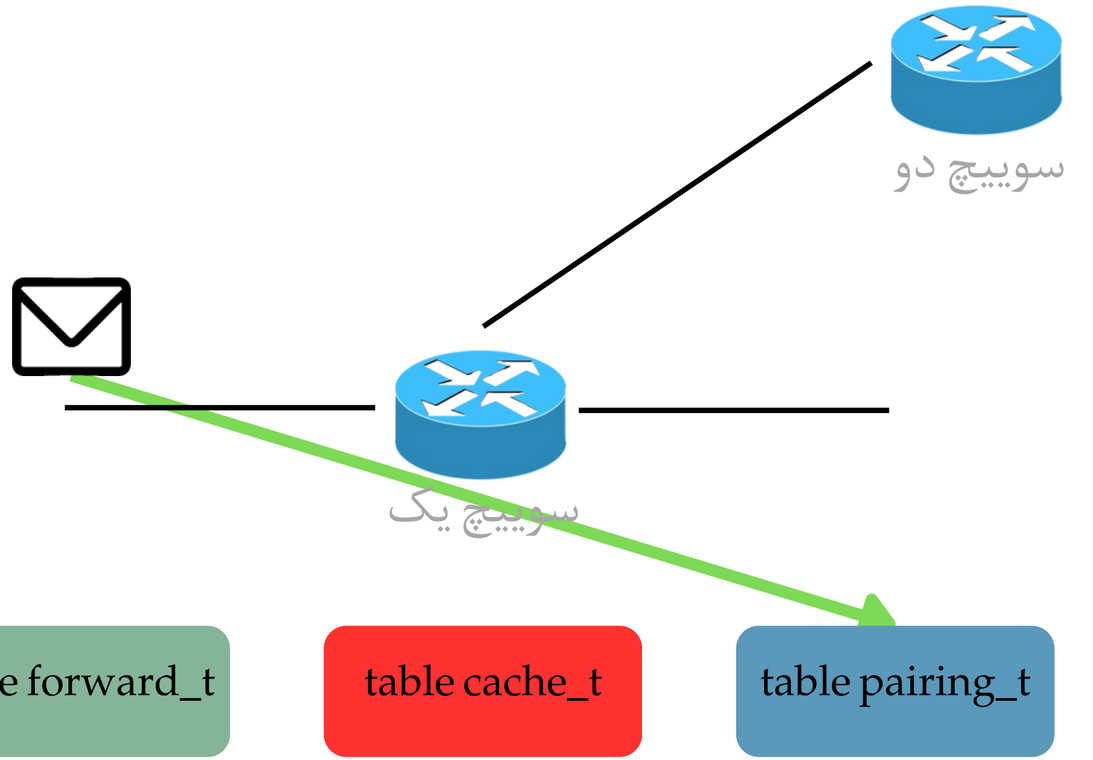


روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

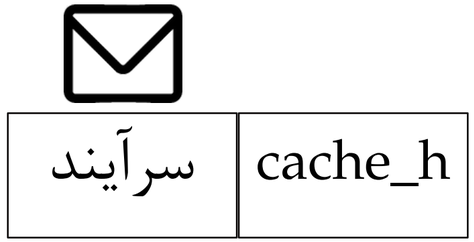


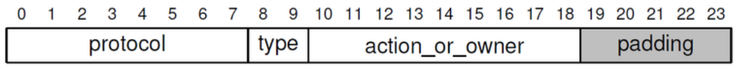


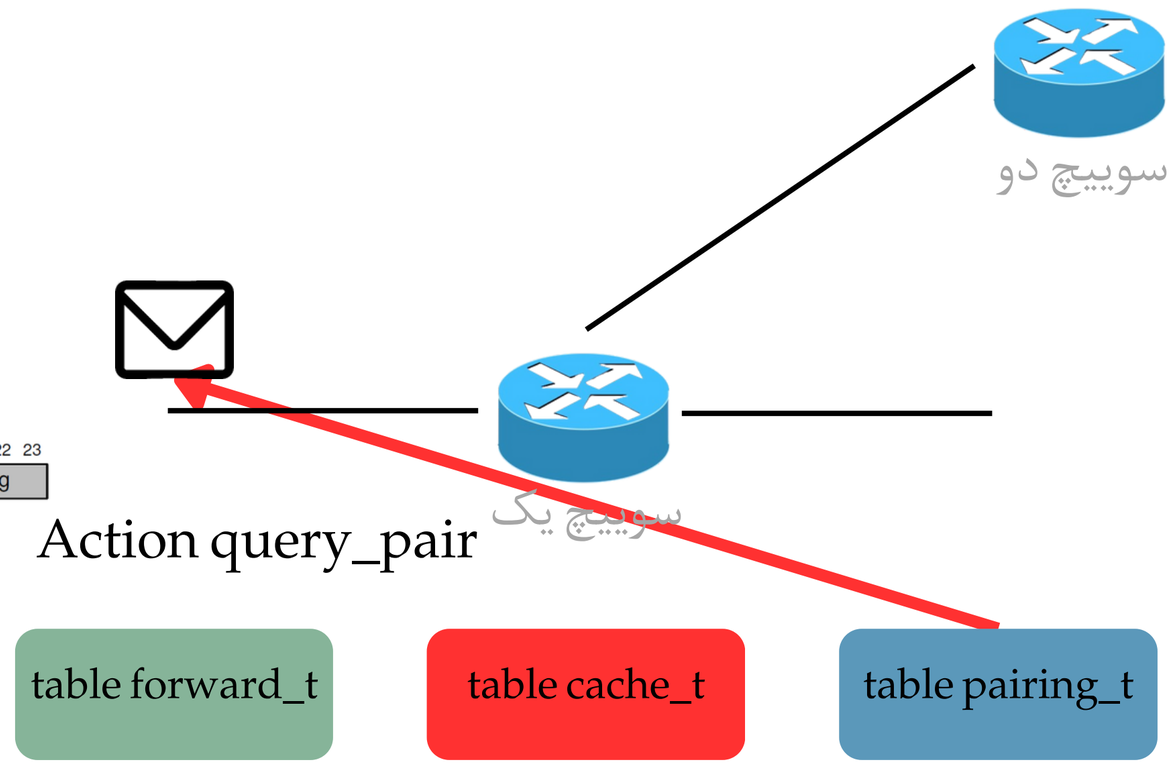


روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

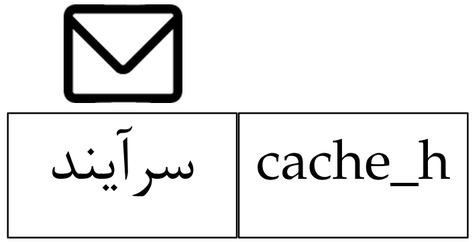


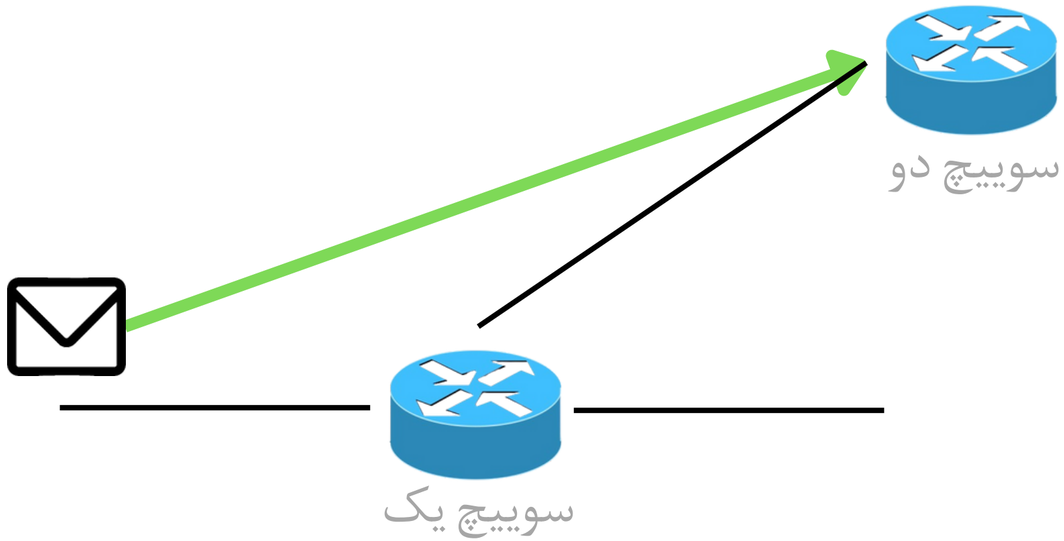




روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده





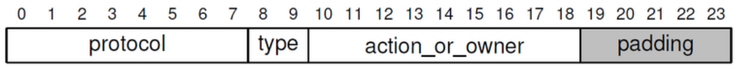


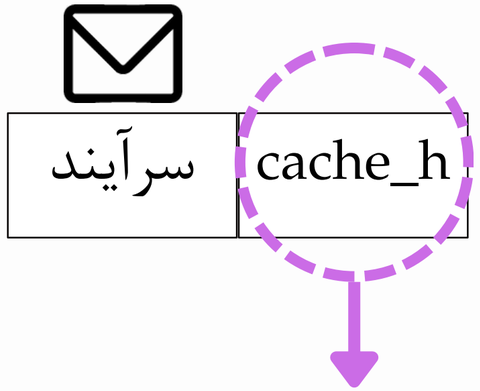
table forward_t

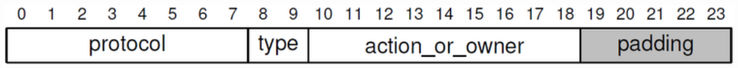
table cache_t

table pairing_t

روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده





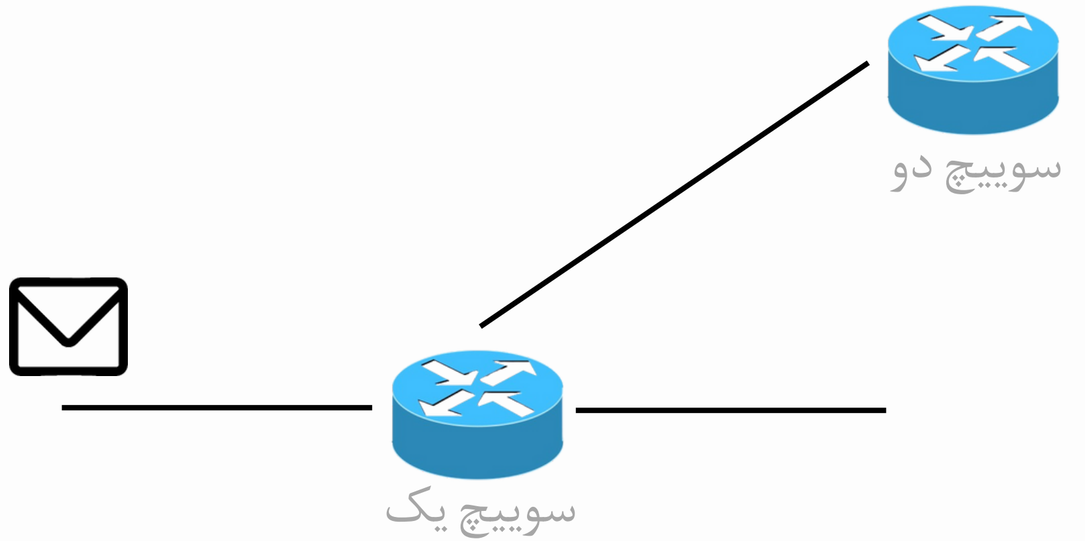


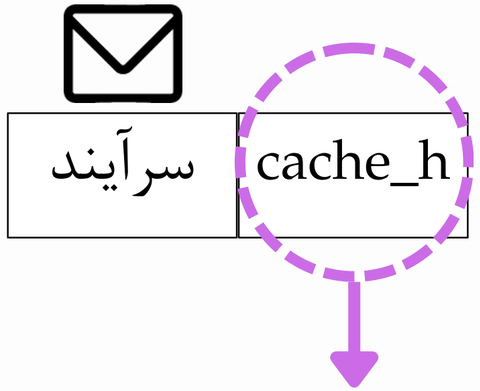
table forward_t

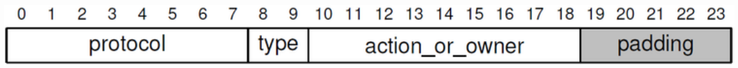
table cache_t

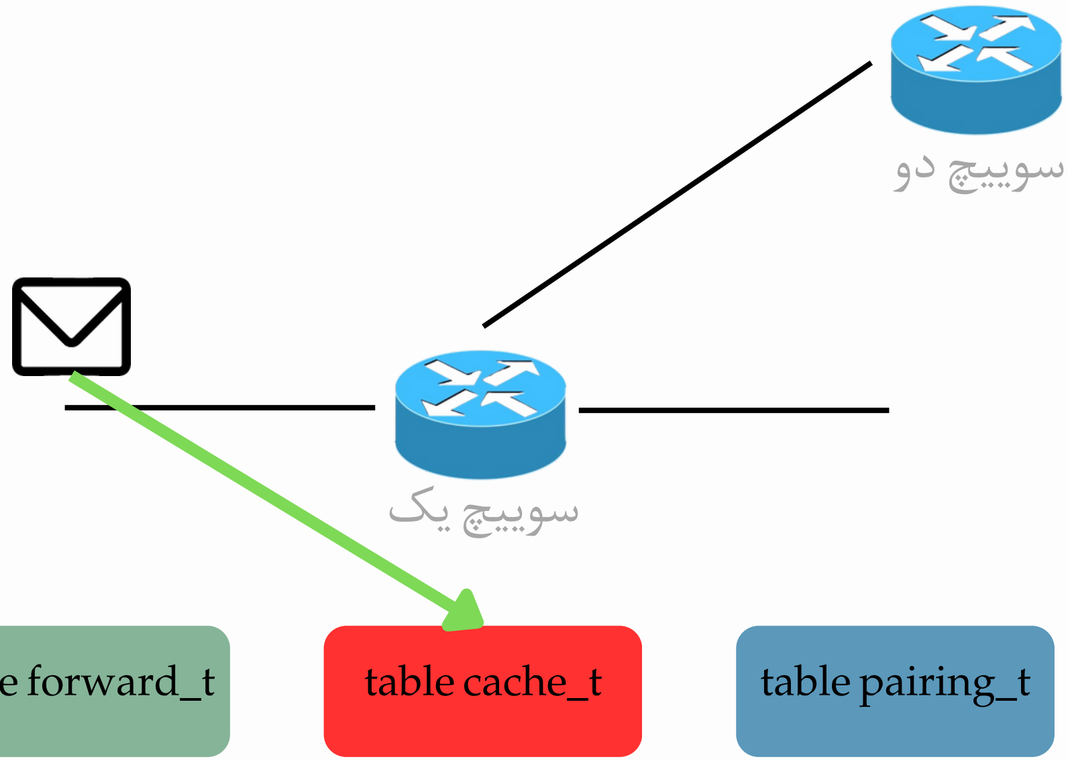
table pairing_t

روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده



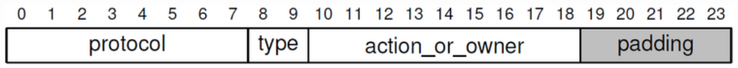


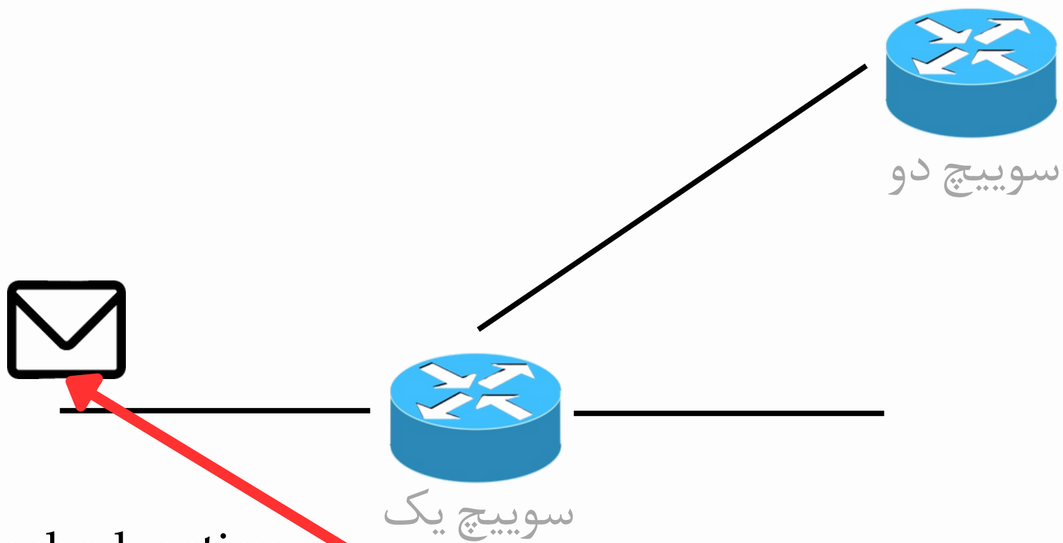


روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

Type = Q1



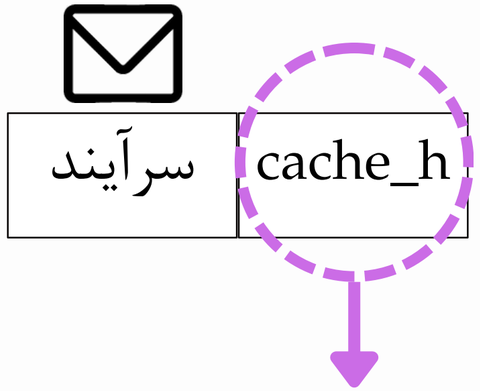


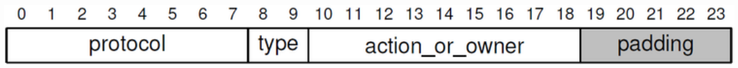
Action get_cached_action

table forward_t | table cache_t | table pairing_t

روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده





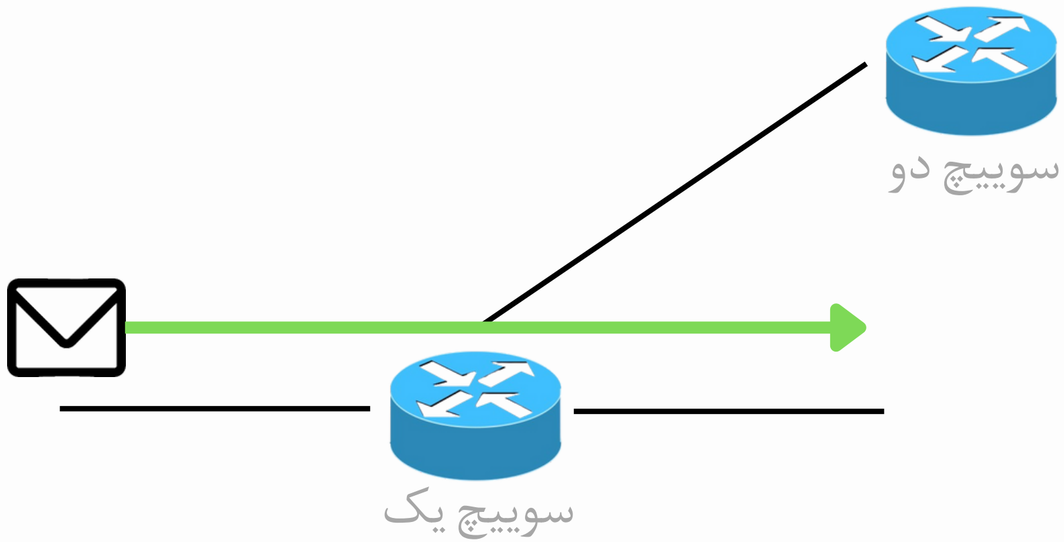


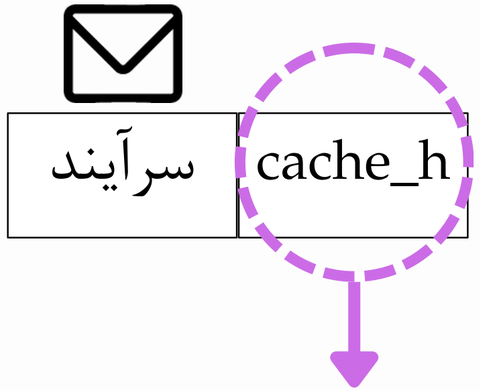
table forward_t

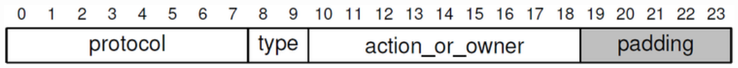
table cache_t

table pairing_t

روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده





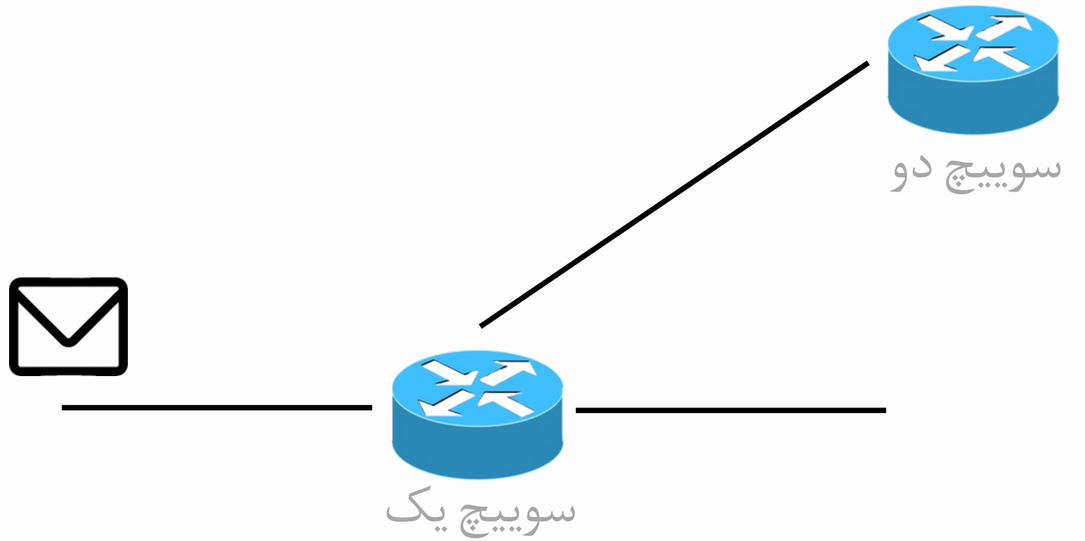


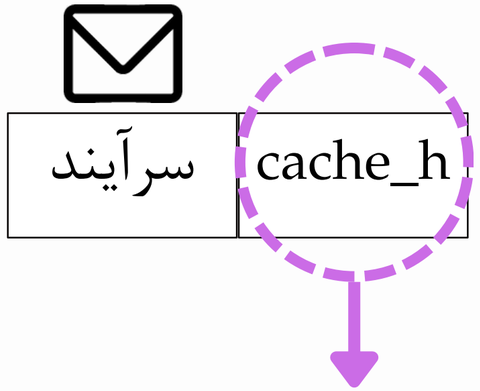
table forward_t

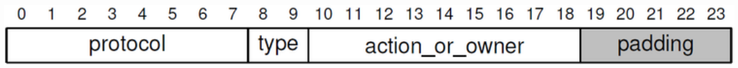
table cache_t

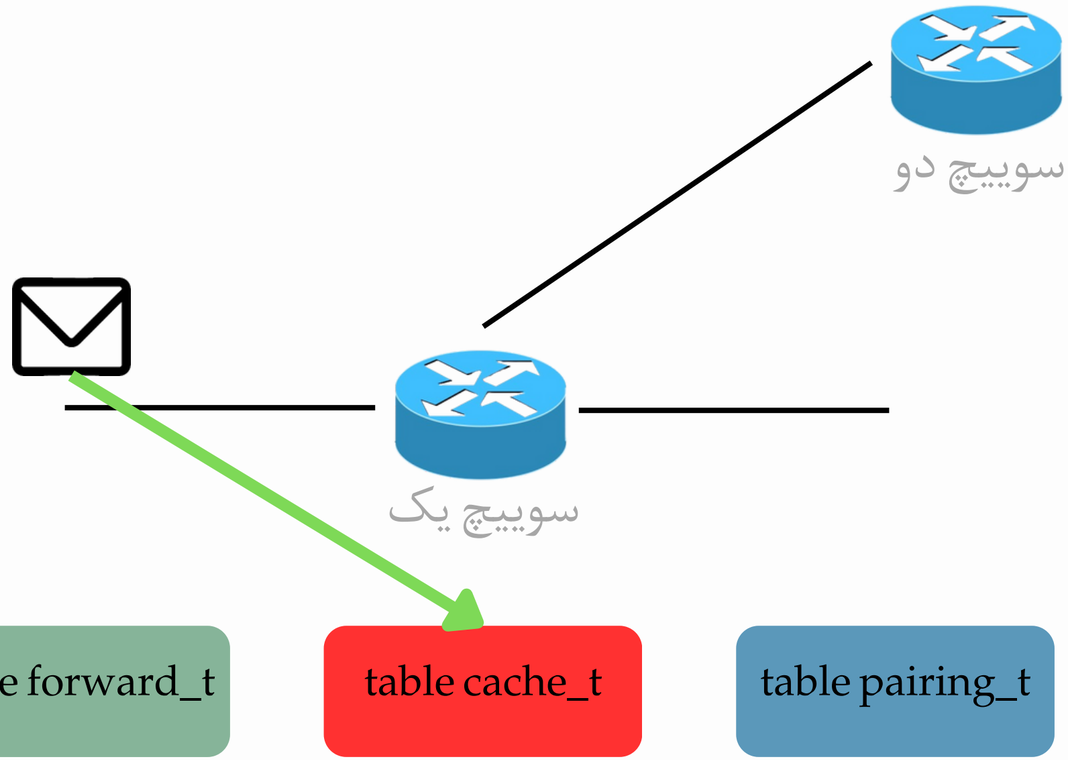
table pairing_t

روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

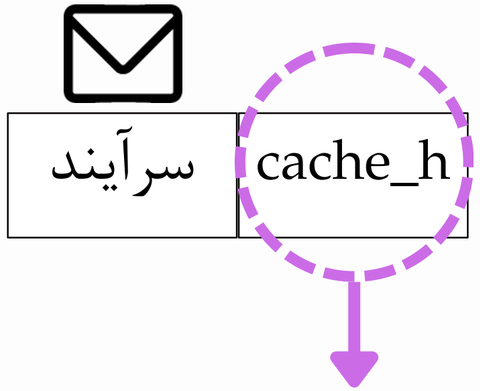


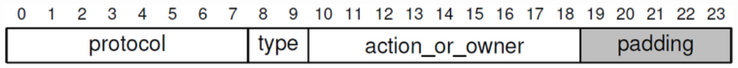


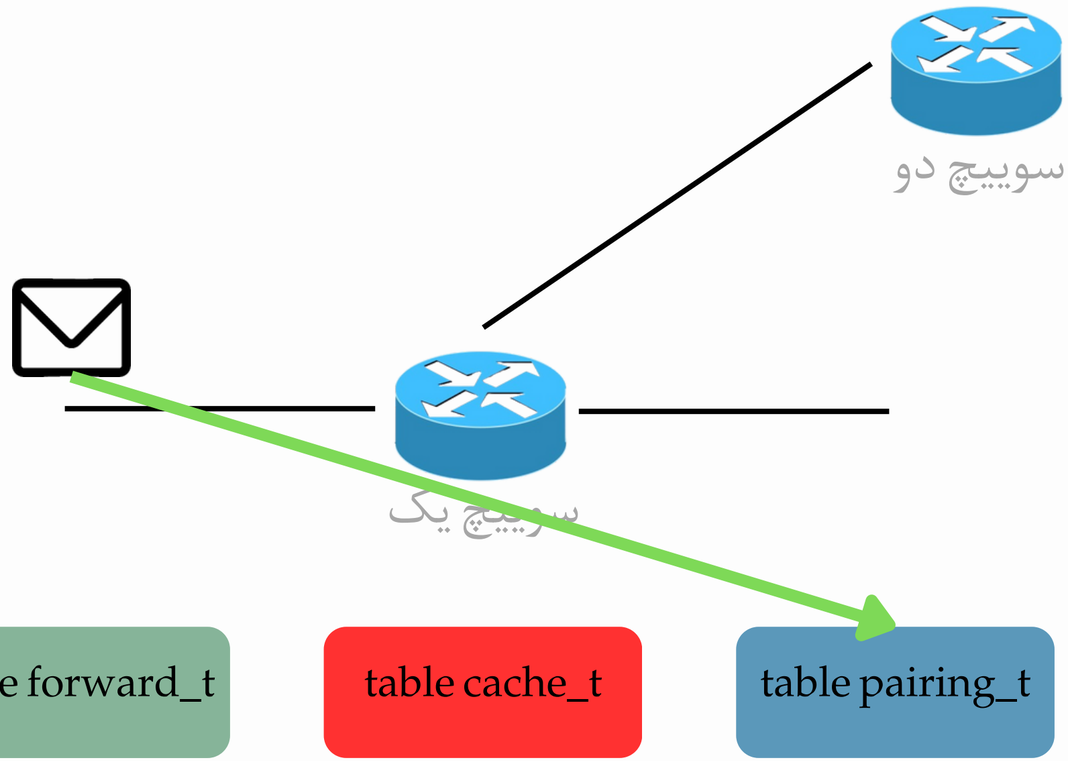


روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

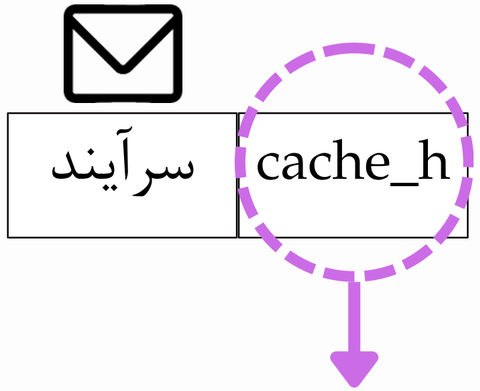


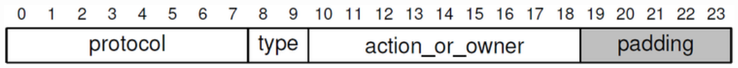


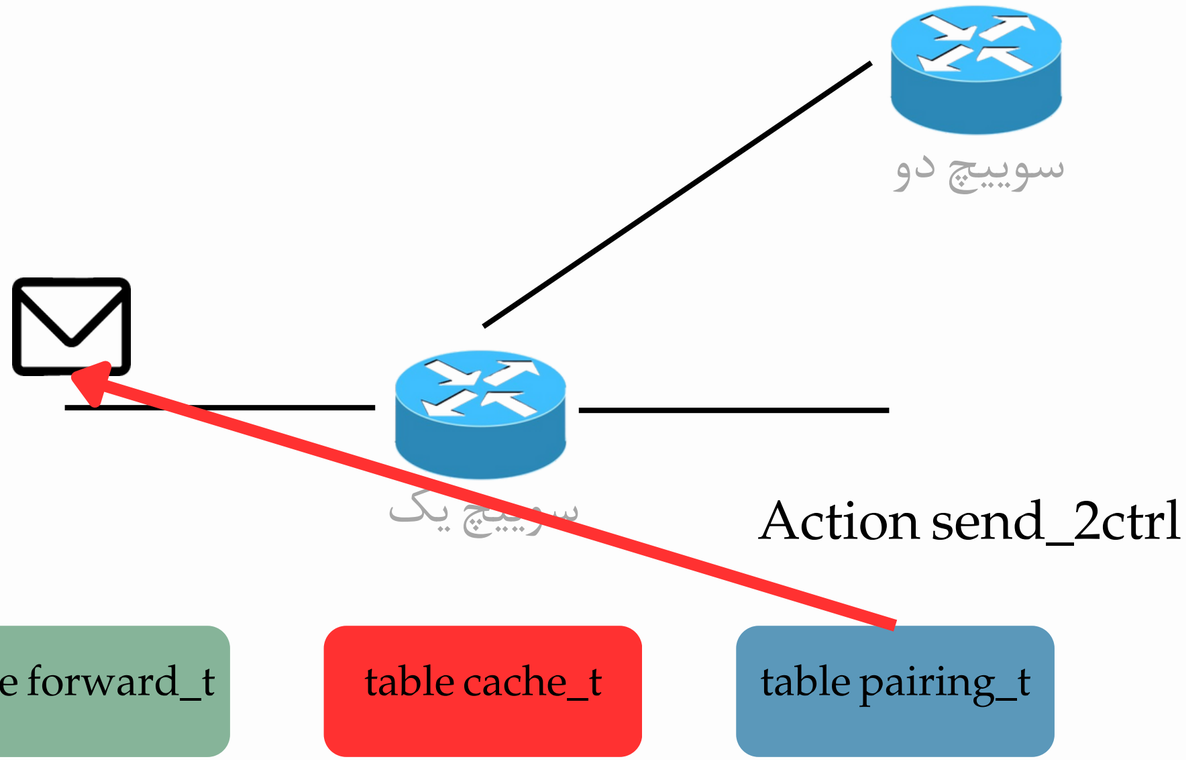


روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

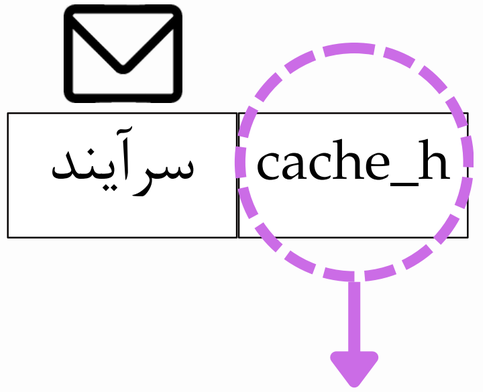


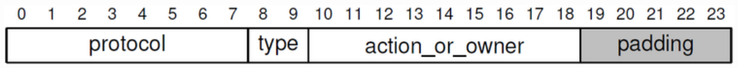




روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده





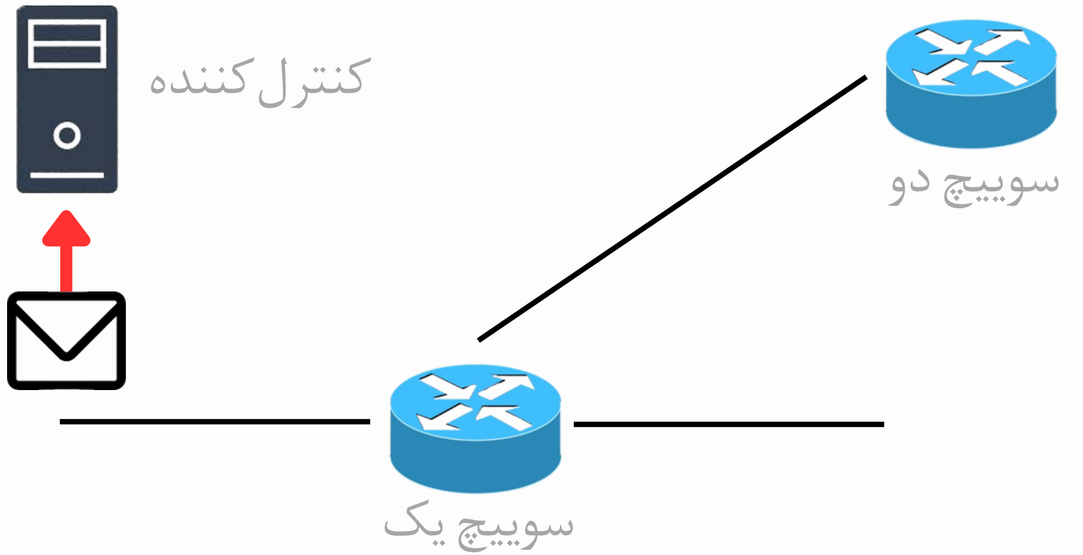


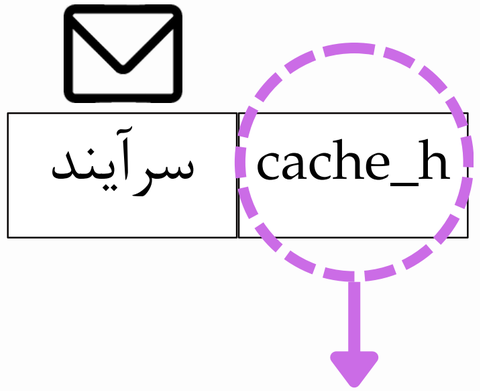
table forward_t

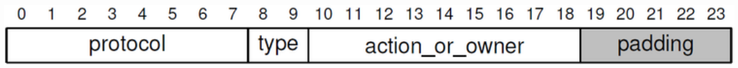
table cache_t

table pairing_t

روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده





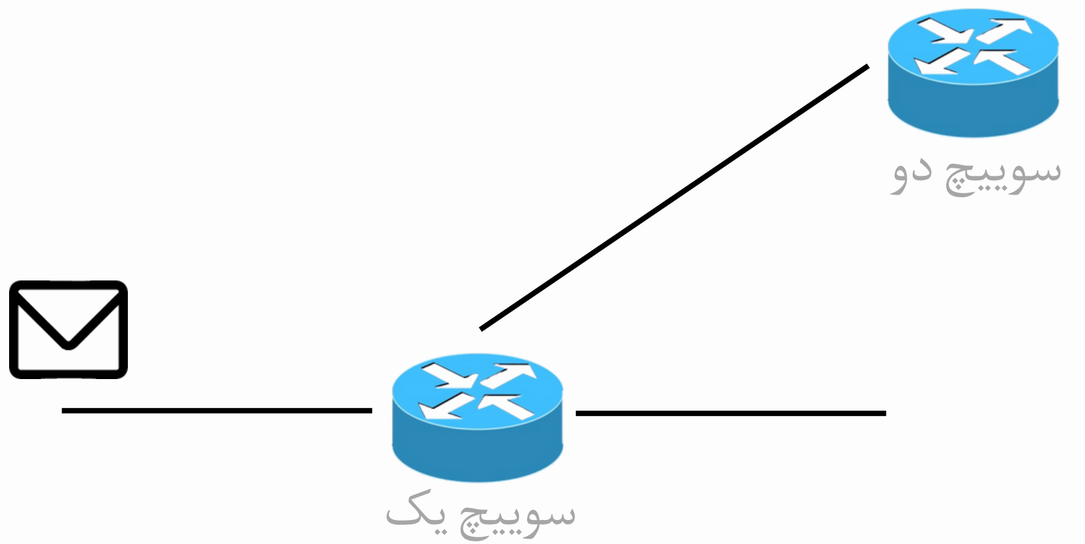


table forward_t

table cache_t

table pairing_t

روش پیشنهادی

طراحی پی‌فور: منطق پردازش ورودی سوییچ طراحی‌شده

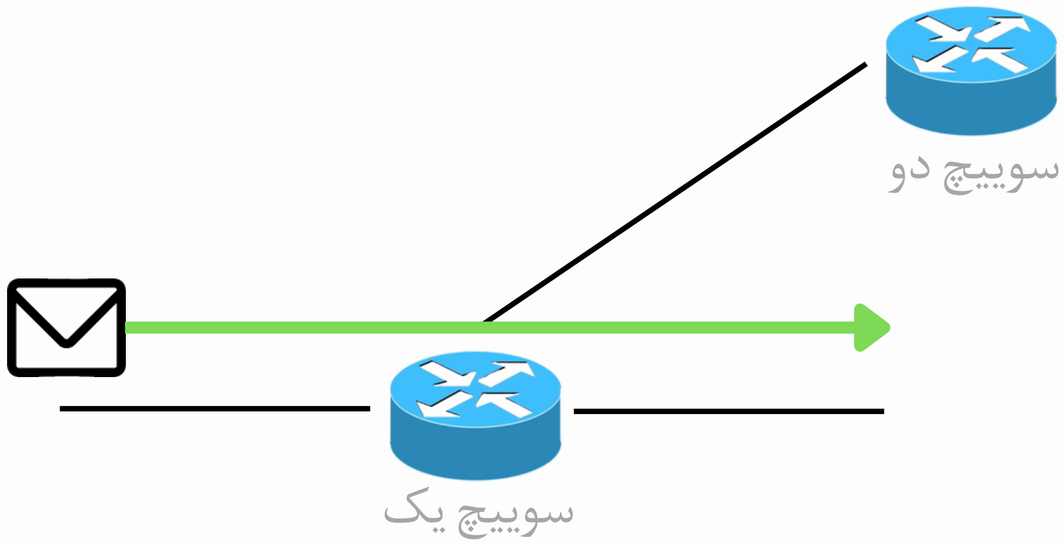


table forward_t

table cache_t

table pairing_t

روش پیشنهادی

راه حل پیشنهادی دارای دو بخش است:

- انتخاب جفت: با استفاده از گردکردن قطعی،
- قراردادن قانون: با استفاده از گردکردن قطعی آگاه از وابستگی قوانین،

$m \leftarrow \text{PaSe}(\mathcal{R}, \mathcal{S}, \{\rho_r\}, \{d_1, d_2, d_3\}, \{M_v\}, \{q_r\})$
$\widetilde{m} \leftarrow \text{relax}(m)$
$\{\widetilde{z}_{w,v}, \widetilde{y}_{w,r}\} \leftarrow \text{solve}(\widetilde{m})$
for $w \in \mathcal{S}$ **do**
    $v \leftarrow \arg\max_{v' \in \mathcal{S}} \widetilde{z}_{w,v'}$
    $\widetilde{m}.\text{add_constraint}(z_{w,v} = 1)$
$\widetilde{m} \leftarrow \text{update}(\widetilde{m}, \{\frac{M_v}{3}\})$
$\{\widetilde{z}_{w,v}, \widetilde{y}_{w,r}\} \leftarrow \text{solve}(\widetilde{m})$

روش پیشنهادی

راه حل پیشنهادی دارای دو بخش است:

- انتخاب جفت: با استفاده از گرد کردن قطعی،
- قراردادن قانون: با استفاده از گردکردن قطعی آگاه از وابستگی قوانین،

$\{L_r\} \leftarrow$ NULL
for $r \in \mathcal{R}$ **do**
  $v \leftarrow \arg\max_{v' \in \mathcal{S}} \widetilde{z}_{w,v'}$
  $\alpha \leftarrow 1 - \widetilde{y}_{w_r,r} - \widetilde{y}_{v,r}$
  $s \leftarrow$ NULL
  if $\widetilde{y}_{w_r,r} \geq \widetilde{y}_{v,r}$ and $\widetilde{y}_{w_r,r} \geq \alpha$ and (L_r is NULL or $L_r = w_r$) **then**
    $s \leftarrow w_r$
  else if $\widetilde{y}_{v,r} \geq \widetilde{y}_{w_r,r}$ and $\widetilde{y}_{v,r} \geq \alpha$ and (L_r is NULL or $L_r = v$) **then**
    $s \leftarrow v$
  else
    $\widetilde{m}$.add_constraints($y_{w_r,r} = 0, y_{v,r} = 0$)

نتایج

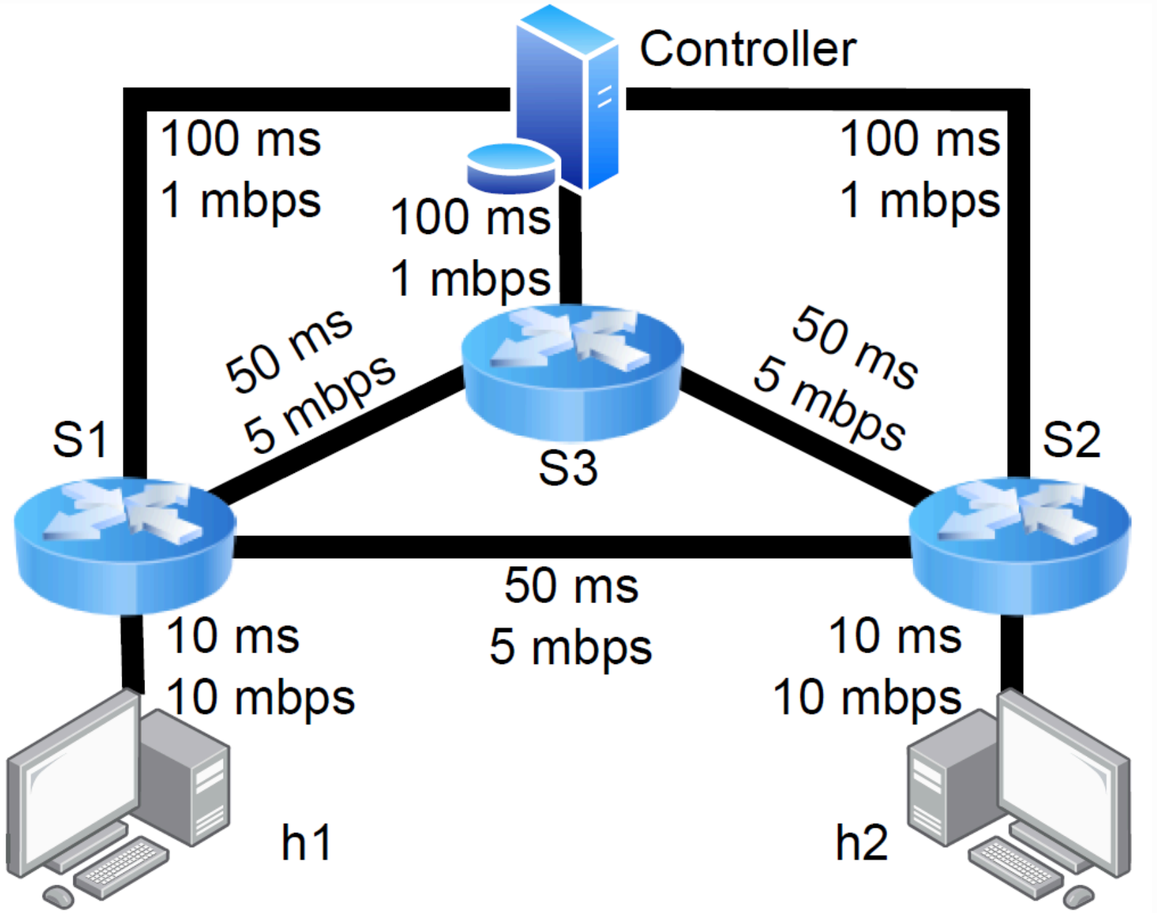
همانندسازی

روش‌های پایه

- بدون ذخیره در جفت سوییچ اصلی
- اولویت میزبان یک بر دو
- اولویت میزبان دو بر یک

ابزار

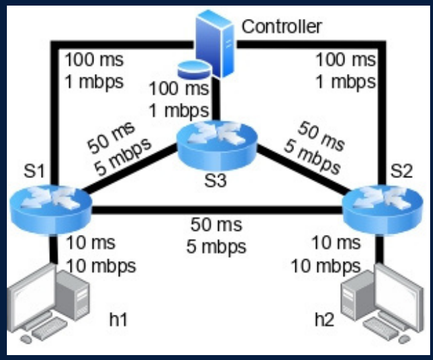
- سوییچ نرم‌افزاری پی‌فور
- مینینت
- آی‌پرف
- پینگ



نتایج

نتایج همانندسازی:

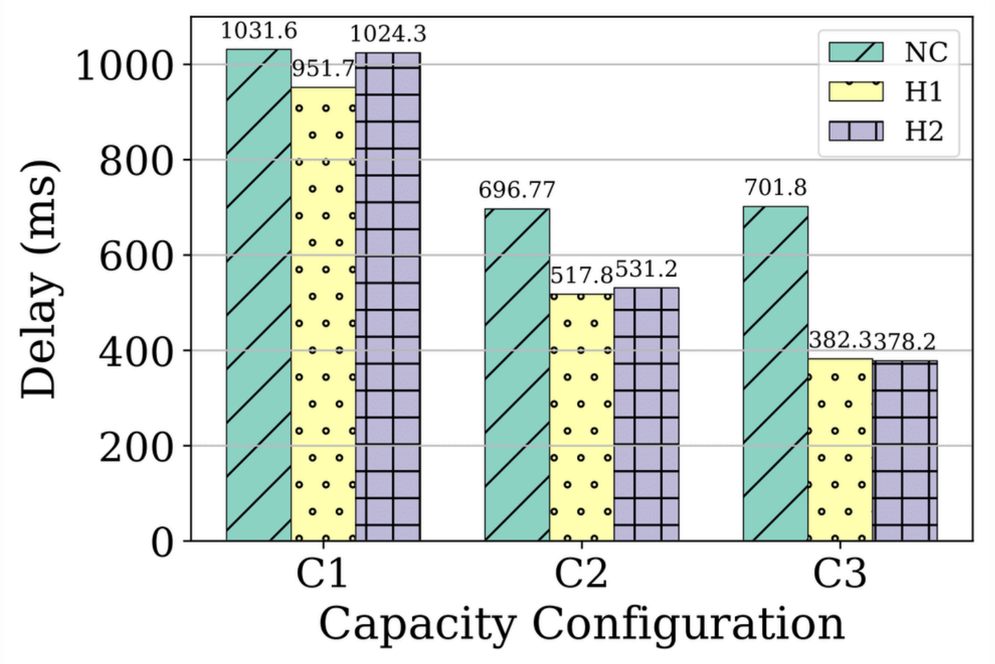
- تحت پیکربندی‌های مختلف حافظه، بهبود قابل‌توجهی در نرخ و تأخیر قابل دستیابی بین میزبان‌ها به‌عنوان نتیجه به‌کارگیری روش ذخیره قوانین پیشنهادی ما مشاهده می‌شود.
- تا ۵ برابر بهبود نرخ و ۷۰٪ کاهش تاخیر

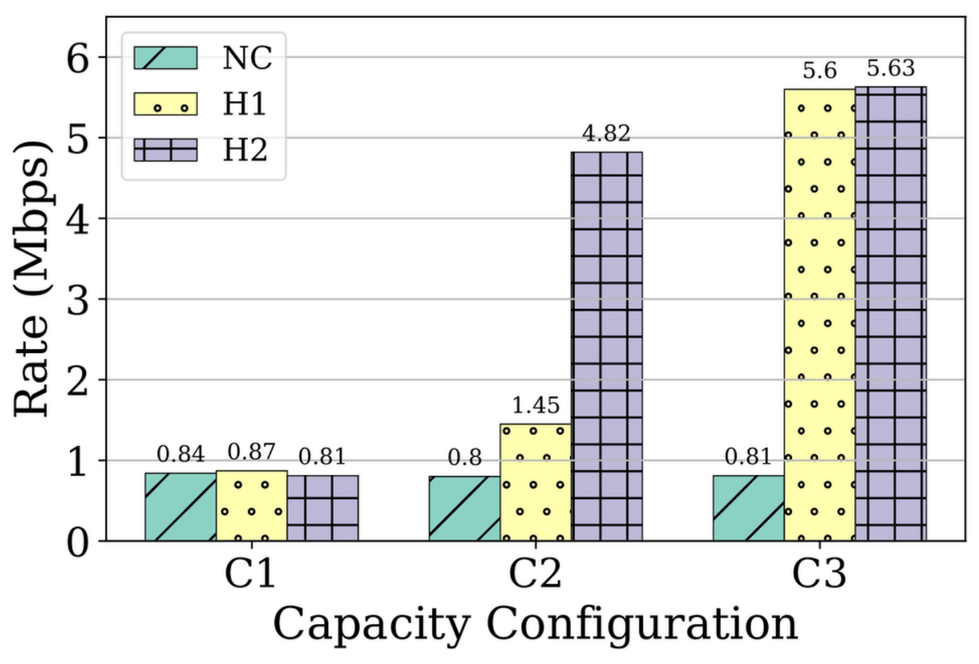


C1: S1 = S2 = S3 = 0

C2: S1 = S2 = S3 = 1

C3: S1 = S2 = 1 , S3 = 2





نتایج

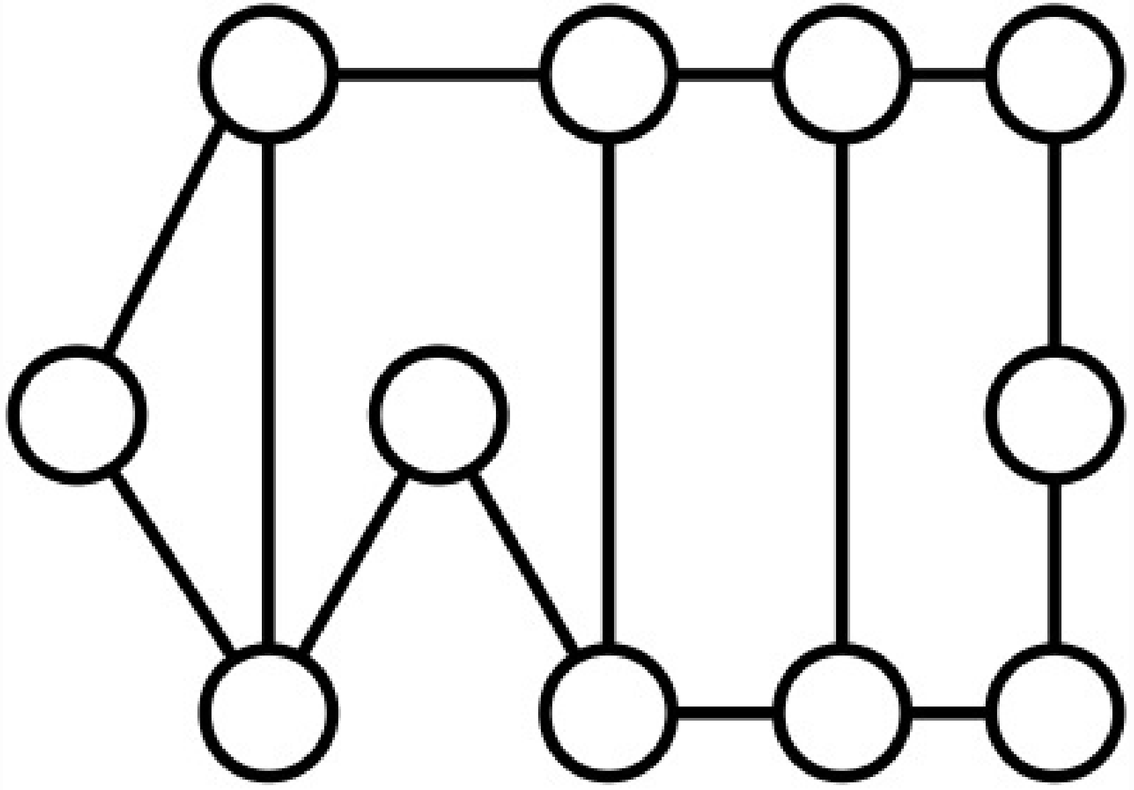
شبیه‌سازی

روش‌های پایه

- بهینه
- جفت‌یابی تصادفی، جایابی بهینه
- جفت‌یابی تصادفی، جایابی حریصانه
- بدون ذخیره در جفت سوییچ اصلی

ابزار

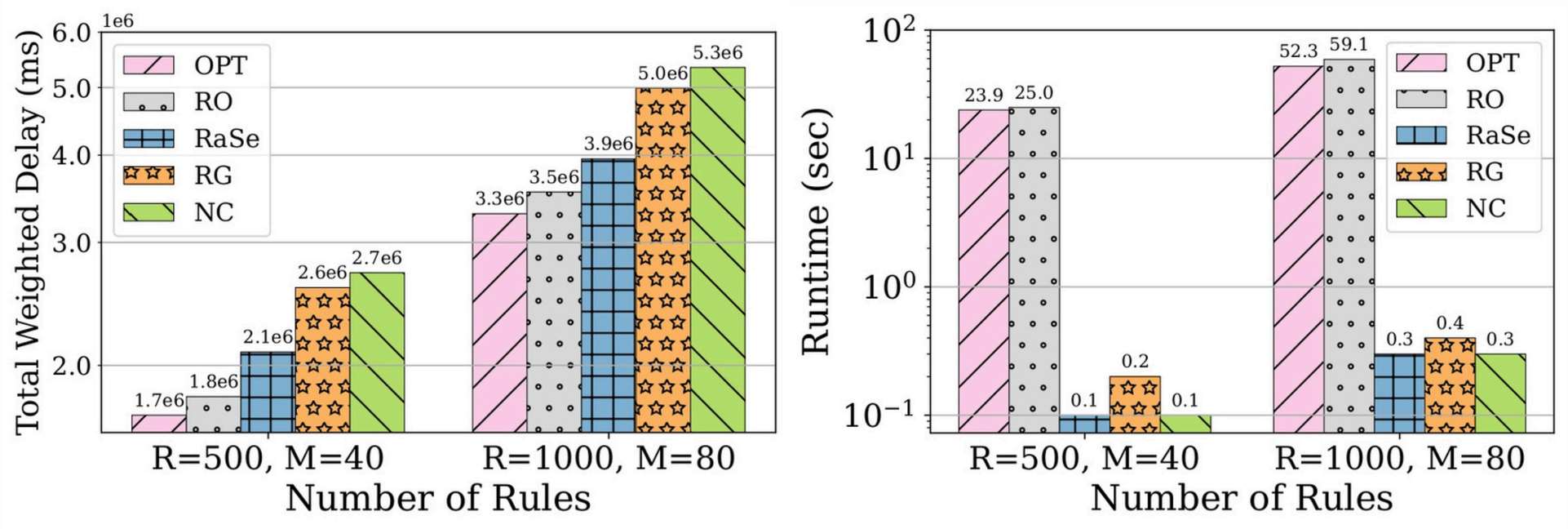
- حل‌کننده سای‌پای
- سی‌وی‌ایکس‌پی‌وای
- زبان برنامه‌نویسی پایتون

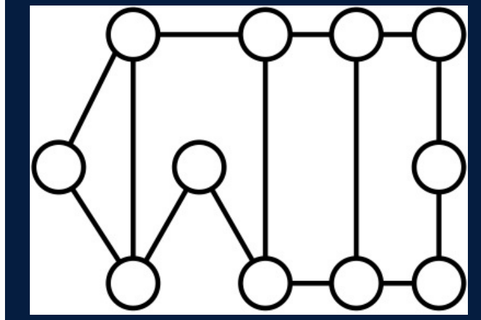


نتایج

نتایج شبیه‌سازی:

- روش پیشنهادی ما در برابر بهینه
- حدود ۱۶ درصد سربار تأخیر برای بیش از ۱۶۰ برابر بهبود زمان اجرا

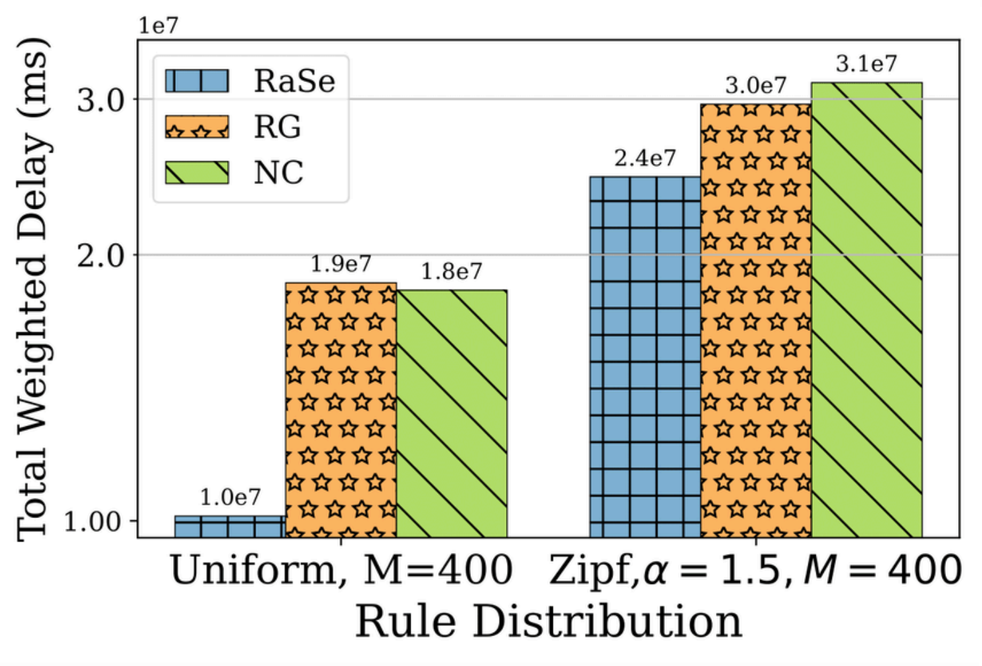


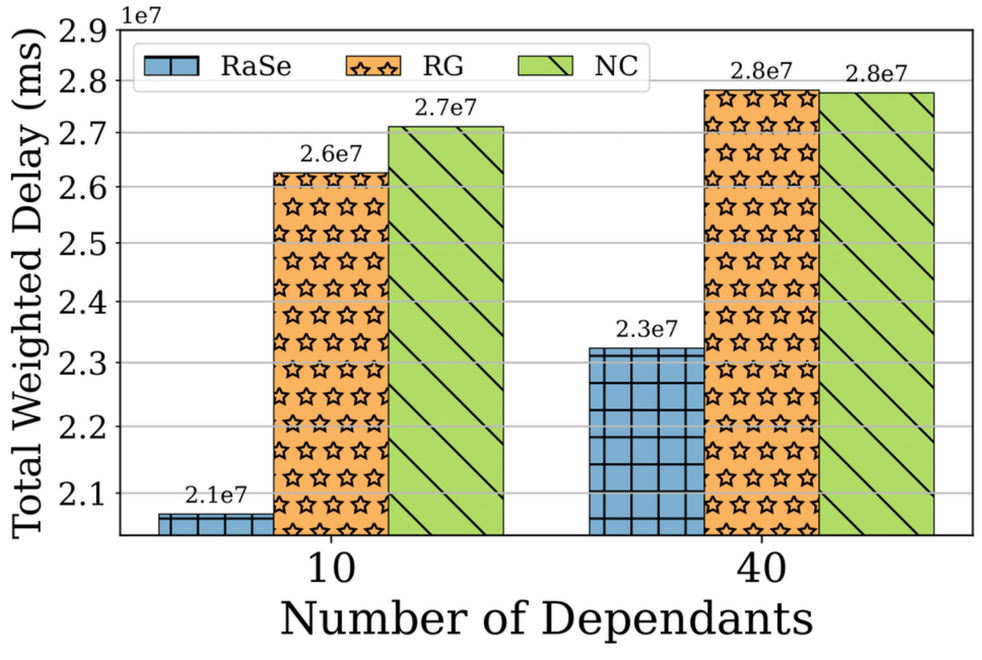


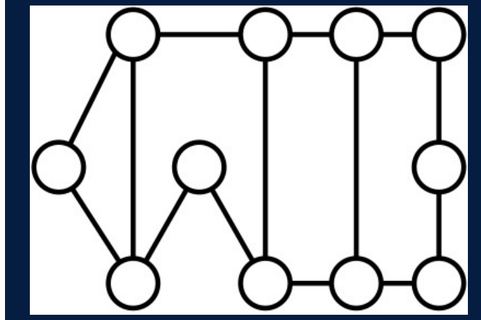
نتایج

نتایج شبیه‌سازی:

- روش ما در مقابل روش‌های پایه
- تحت تنظیمات مختلف بین ۱۵ تا ۲۱ درصد بهبود قابل مشاهده است.



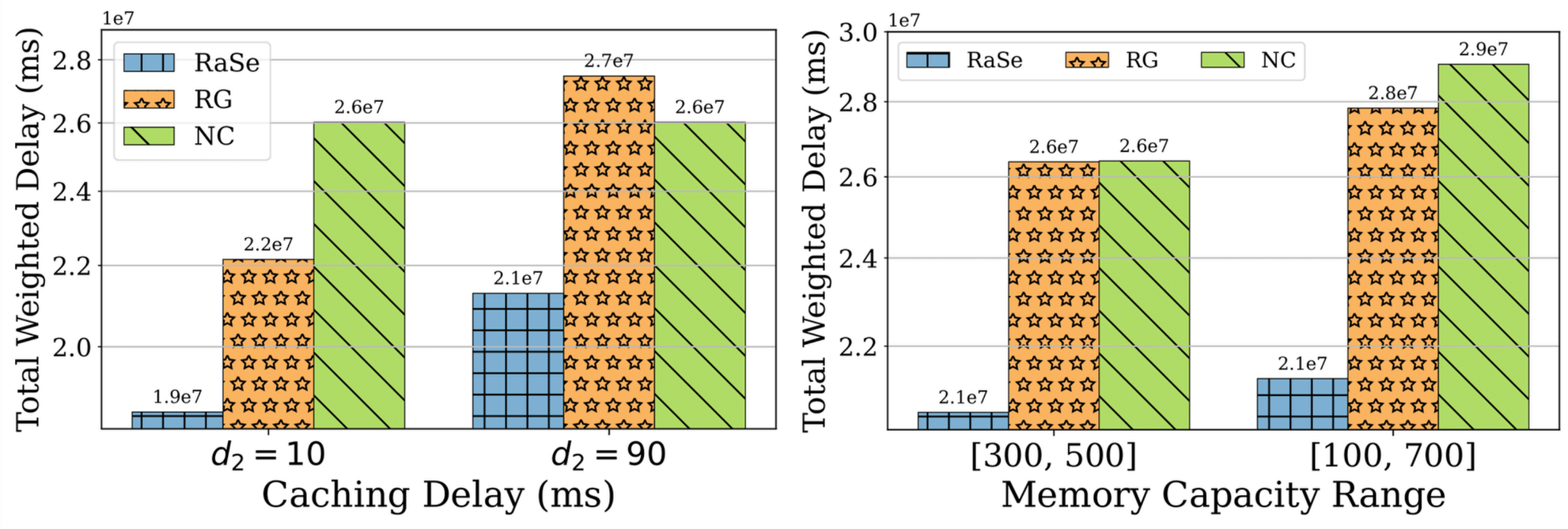


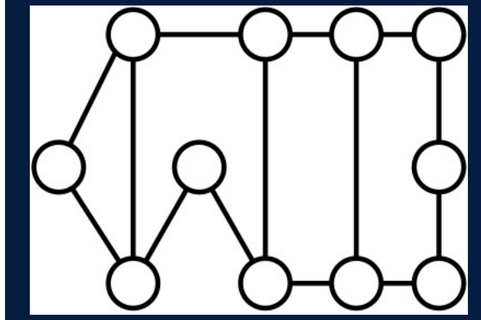


نتایج

نتایج شبیه‌سازی:

- روش ما در مقابل روش‌های پایه
- تحت تنظیمات مختلف بین ۱۵ تا ۲۱ درصد بهبود قابل مشاهده است.





نتیجه‌گیری

- این کار پژوهشی RaSe را معرفی می‌کند، الگوریتمی برای جفت‌کردن سوییچ‌ها برای ذخیره قوانین در حافظه پنهان سوییچ‌های شبکه‌های نرم‌افزار محور دارای پی‌فور و از گردکردن قطعی برنامه‌ریزی خطی عدد صحیح استفاده می‌کند و یک راه‌حل تقریبی ارائه می‌دهد.

- جزئیات طراحی و اجرای پی‌فور یک سوییچ با قابلیت ذخیره‌سازی را نشان می دهد و عملکرد سوییچ را در یک شبیه‌سازی مبتنی بر مینینت ارزیابی می‌کند.

- شبیه‌سازی‌های عددی گسترده‌ای را برای ارزیابی مقیاس‌پذیری و اثربخشی RaSe، به‌ویژه تحت بارهای ناهموار شبکه انجام می‌دهد.

- تحقیقات آینده را برای بررسی سناریوهایی که شامل جفت سوییچ‌های متعدد یا همسایگان غیرمستقیم است، پیشنهاد می‌کند.

پایان‌نامه ارشد بنده در کنفرانس
GLOBECOM 2024 پذیرفته ،
ارائه و منتشر شده است.

تمامی کدهای این کار پژوهشی در
لینک زیر موجود می‌باشد.

https://github.com/mmdsbri98/Efficient-Rule-Caching

با تشکر از توجه شما

لطفا سوالات خود را مطرح بفرمایید

پیوست ۱

محاسبه نرخ قوانین:

- روش‌های تخمین زدن نرخ قوانین (جریان‌ها) در شبکه
  - نمونه‌برداری (Sampling)
  - استفاده از شمارنده‌ها (Counters)
  - بازتاب بسته‌ها (Packet Mirroring)
  - جریان نمونه (Sampled Flow (sFlow))
  - نت‌فلو (NetFlow)
  - قرارداد مدیریت شبکه ساده (Simple Network Management Protocol ((SNMP)
  - استفاده از ساختارداده اسکچ (Sketch)

پیوست ۲

وابستگی بین قوانین:

اولویت	عمل	بخش انطباق	قانون
۰	ارسال به درگاه صفر	(**1*)	R0
۱	ارسال به درگاه یک	(0**1)	R1
۲	ارسال به درگاه دو	(011*)	R2
۳	ارسال به درگاه سه	(0111)	R3

پیوست ۲

وابستگی بین قوانین:

- وابسته مستقیم
  - **R1** مستقیما به **R0** وابسته است
- وابسته غیر مستقیم
  - **R2** غیر مستقیم به **R0** وابسته است
- هنگامی که یک قانون در سوییچ نصب می شود، همه قوانینی که به آن بستگی دارد باید به طور همزمان نصب شوند.