فرم گزارش کار آزمایشگاه شبکه

نام و نام خانوادگی	حسین تاتار	شماره دانشجویی	40133014	نام و شماره آزمایش	11 - پیکربندی و راه اندازی شبکه SDN

هدف آزمایش	آشنایی با مفاهیم پایه شبکه های نرمافزار و نحوه راه اندازی یک کنترلر مرکزی با استفاده از Ryu در محیط Mininet.در این آزمایش دانشجویان با نحوه نصب کنترلر ، Ryu اجرای کنترلر در کنار Mininet و استفاده از ابزار FlowManager برای مدیریت جریان های ترافیکی آشنا میشود.
ابزارهای مورد نیاز	VMware Workstation نرمافزار فایل ماشین مجازی Mininet pip و Python3 نسخه 2.3.0 کنترلر Ryu
شرح آزمایش	**آمادگی پیش از آزمایش** **سوال 1:** شبکه های SDN چه تفاوتی با شبکه های سنتی دارند؟ **جواب:** • کنترل متمرکز vs غیرمتمرکز: در شبکه های سنتی، هر دستگاه (مانند سوئیچ یا روتر) به طور مستقل تصمیم گیری میکند. (با استفاده از پروتکلهایی مانند OSPF یاSTP). درSDN ، کنترلر مرکزی هوشمند، تمام تصمیم های مسیریابی و مدیریت ترافیک را میگیرد و دستگاهها فقط نقش اجرایی دارند. • انعطاف پذیری: SDN امکان برنامه ریزی پویای شبکه را فراهم میکند، درحالیکه شبکه های سنتی سخت افزارمحور و محدود به تنظیمات دستی هستند. • جداسازیPlane های کنترل و داده: درSDN ، Control Plane(منطق شبکه) از Data Plane(فورواردینگ ترافیک) جدا شده است. **سوال 2:** نقش کنترلر در معماری SDN چیست؟ **جواب:** • مغز متفکر شبکه :کنترلر تصمیم میگیرد که چگونه ترافیک مدیریت شود (مثلاً مسیر جریان ها را تعیین میکند). • ارتباط با دستگاهها: از طریق پروتکلی مانند OpenFlow، کنترلر به سوئیچها میگوید چگونه بسته ها را فوروارد کنند. • مدیریت پویا :امکان اعمال سیاستهای جدید مثلQoS و فایروال بدون نیاز به تنظیم دستی هر دستگاه. **سوال 3:** مزایای استفاده از فریمورک Ryu چیست؟ **جواب:** • سبک و مبتنی بر پایتون :نوشتن برنامه های کنترل شبکه با کدنویسی ساده تر امکانپذیر است. • پشتیبانی ازOpenFlow : سازگاری با آخرین نسخه های پروتکلOpenFlow • ماژولار بودن :امکان توسعه آسان با ماژولهای ازپیش ساخته مثلFirewall ، L2 Switch • جامعه فعال : مستندات و مثالهای فراوان برای یادگیری سریع. **شرح ازمایش**

سوال 4: تب flows و topology را بازکنید و اطلاعات مشاهده شده خود را بیان کنید.
در گام بعد در ماشین مجازی mininet مانند زیر دستور pingall را بزنید و تب های flows و topology را مجدداً بازکنید. نتایج حاصل شده را تشریح کنید.
چند جریان پس از اجرای pingall اضافه شد؟ منبع و مقصد آنها را تحلیل و تشریح کنید.

جواب:

وضعیت اولیه (قبل از pingall) : در تب Flows فقط یک جریان پیش‌فرض وجود دارد که تمام ترافیک را به کنترلر می‌فرستد (OUTPUT:CONTROLLER) برای مثال به صورت زیر:

PRIORITY=0, MATCH=ANY, ACTIONS=OUTPUT:CONTROLLER

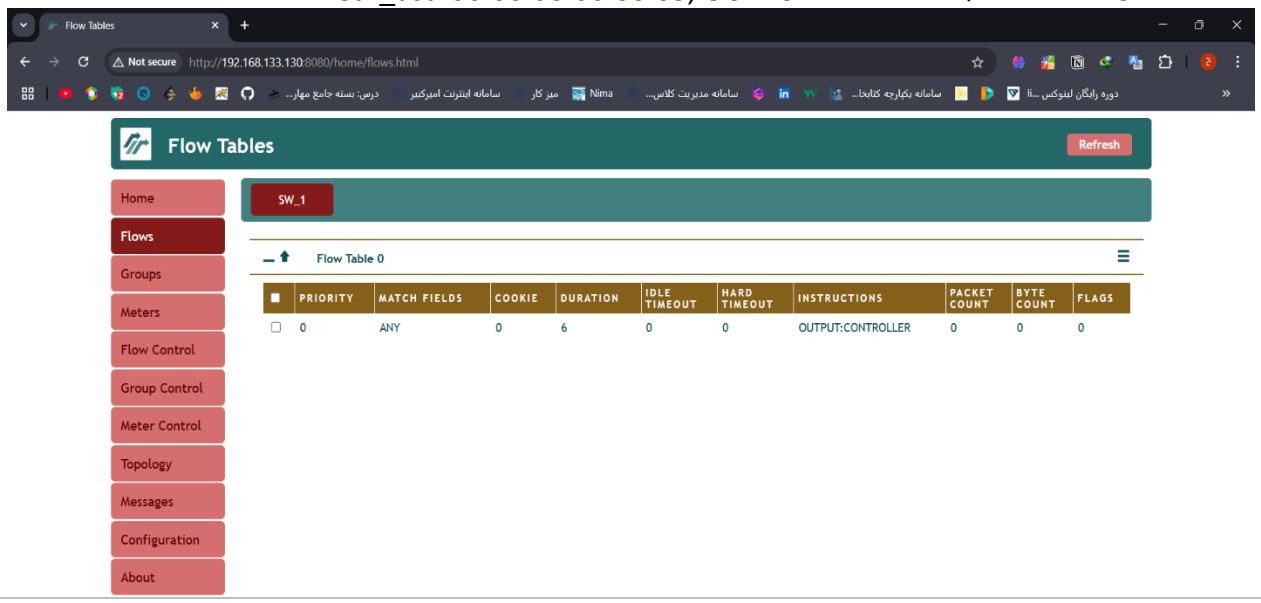
هدف کنترلر اینست که تمام بسته‌ها را بررسی کند تا قوانین مسیریابی را تعیین کند.
در تب Topology فقط دستگاه‌ها (هاست‌ها و سوئیچ) نمایش داده می‌شوند، اما هیچ اتصالی بین آنها دیده نمی‌شود.
پس از اجرای pingall درMininet : این دستور بین تمام هاست‌ها مثلاً h1, h2, h3 پینگ متقابل انجام می‌دهد.
5 جریان جدید ایجاد شده است (جریان‌های با OUTPUT:1, OUTPUT:2, OUTPUT:3).
نمونه جریان‌ها به صورت زیر است:

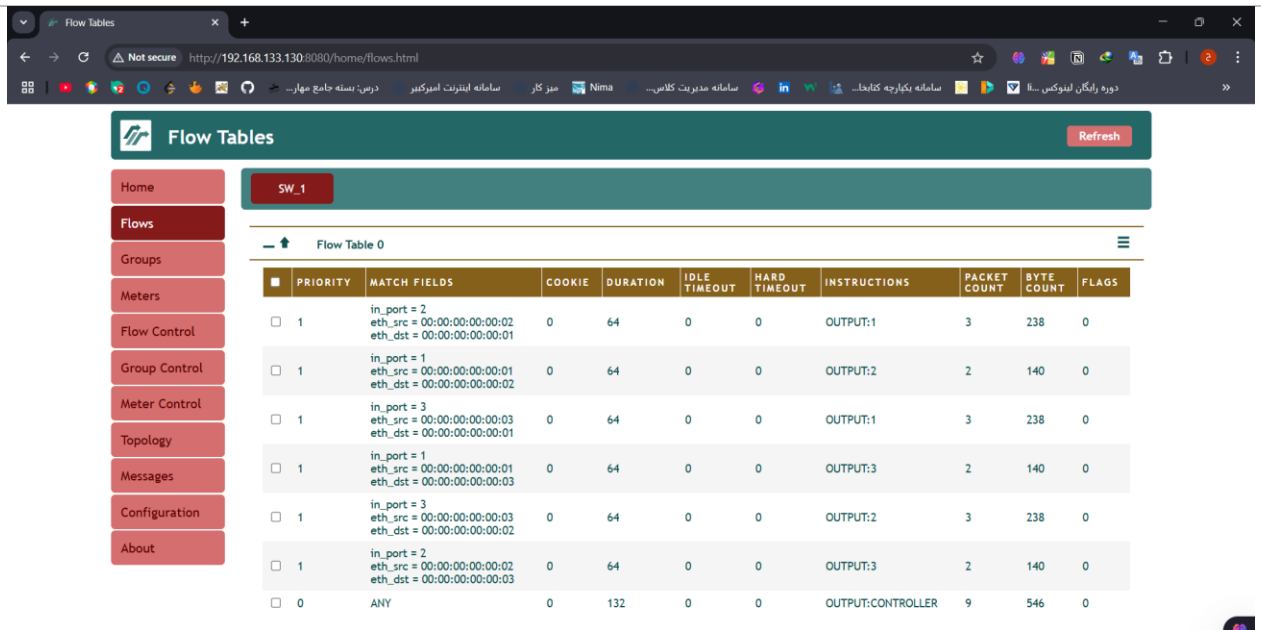
1. eth_dst=00:00:00:00:00:02, OUTPUT:1
   - منبع :هاست با MAC 00:00:00:00:00:01 یا 00:00:00:00:00:03 .
   - مقصد :هاست با MAC 00:00:00:00:00:02 .
2. eth_dst=00:00:00:00:00:03, OUTPUT:2
   - منبع :هاست با MAC 00:00:00:00:00:01 .
   - مقصد :هاست با MAC 00:00:00:00:00:03 .

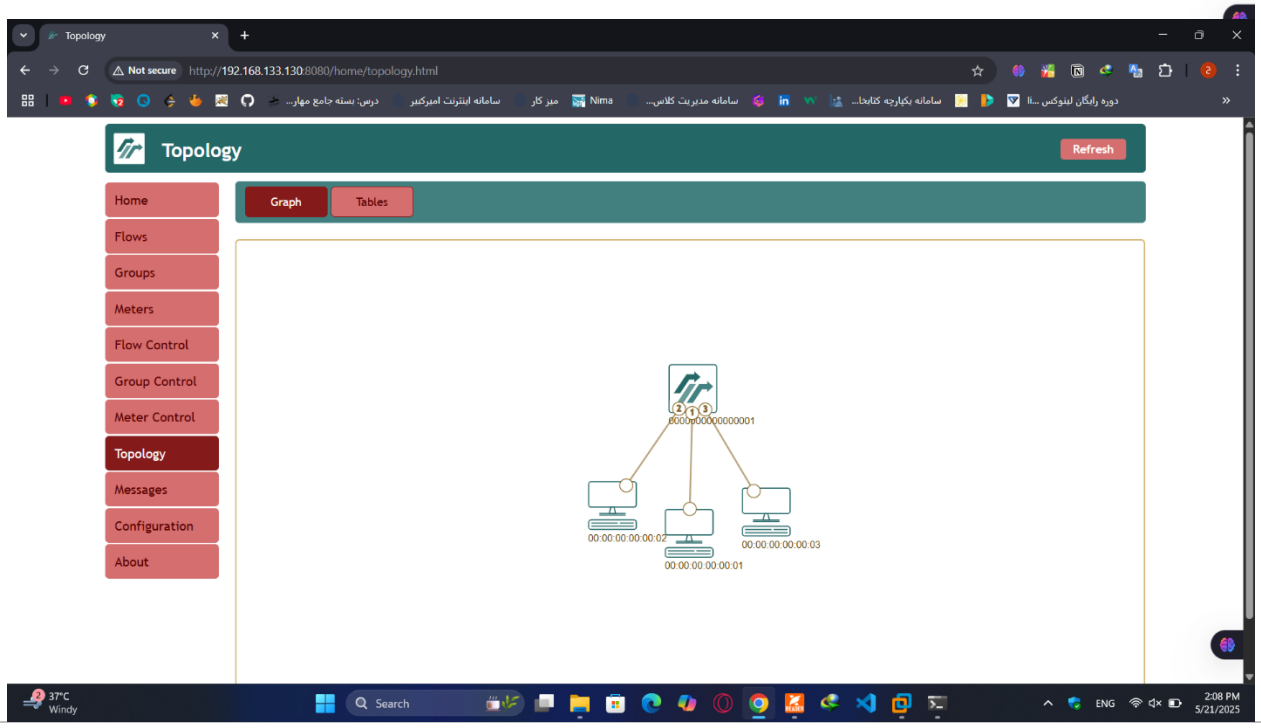
نحوه کار به این صورت است که سوئیچ با یادگیری MAC آدرس‌ها، جریان‌ها را ثبت می‌کند (مثلاً اگر h1 به h2 پینگ بزند، جریان با OUTPUT: پورت مربوط به h2 ایجاد می‌شود).
در تبTopology اتصالات بین هاست‌ها و سوئیچ باید به صورت خطوط نمایش داده شوند . MAC آدرس‌ها باید کامل و مرتبط با پورت‌های سوئیچ باشند.

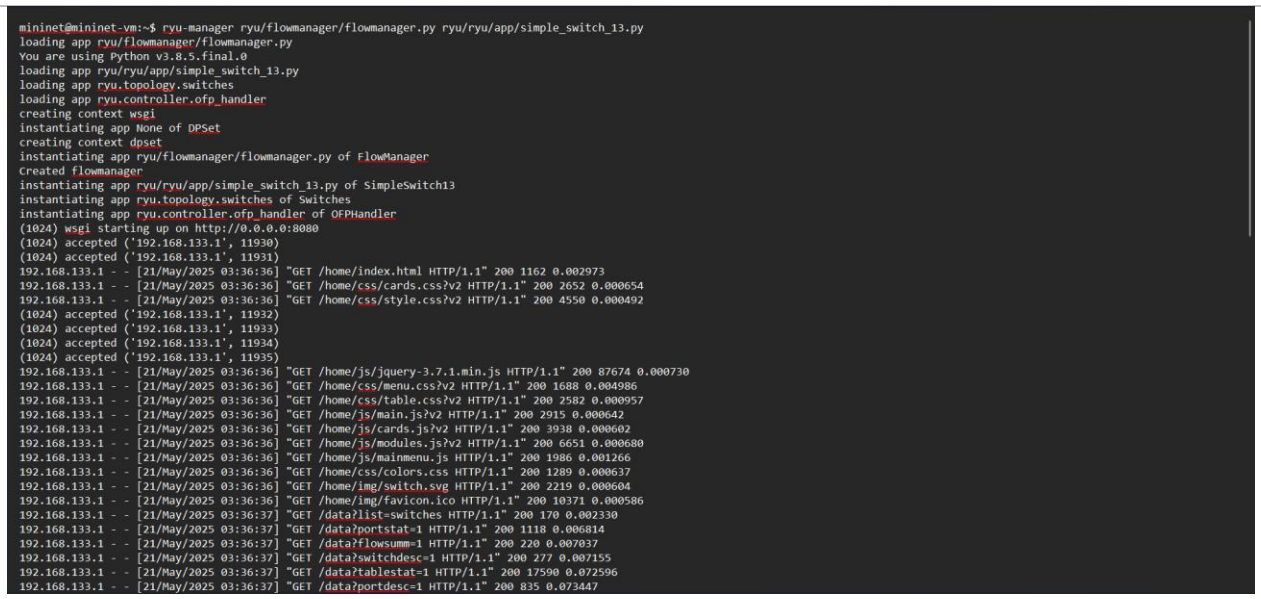
تحلیل منبع و مقصد جریان‌ها: تعداد جریان‌های اضافه شده برابر با 5 مورد است:

1. هاست ۱ → هاست ۲: eth_dst=00:00:00:00:00:02, OUTPUT:1
2. هاست ۲ → هاست ۱: eth_dst=00:00:00:00:00:01, OUTPUT:2
3. هاست ۱ → هاست ۳: eth_dst=00:00:00:00:00:03, OUTPUT:3
4. هاست ۳ → هاست ۱: eth_dst=00:00:00:00:00:01, OUTPUT:1
5. هاست ۲ → هاست ۳: eth_dst=00:00:00:00:00:03, OUTPUT:2




Flow Tables
Not secure
http://192.168.133.130:8080/home/flows.html
Flow Tables
Refresh
Home
Flows
Groups
Meters
Flow Control
Group Control
Meter Control
Topology
Messages
Configuration
About
SW_1
Flow Table 0
PRIORITY
MATCH FIELDS
COOKIE
DURATION
IDLE TIMEOUT
HARD TIMEOUT
INSTRUCTIONS
PACKET COUNT
BYTE COUNT
FLAGS
1 in_port = 2 eth_src = 00:00:00:00:00:02 eth_dst = 00:00:00:00:00:01 0 64 0 0 OUTPUT:1 3 238 0
1 in_port = 1 eth_src = 00:00:00:00:00:01 eth_dst = 00:00:00:00:00:02 0 64 0 0 OUTPUT:2 2 140 0
1 in_port = 3 eth_src = 00:00:00:00:00:03 eth_dst = 00:00:00:00:00:01 0 64 0 0 OUTPUT:1 3 238 0
1 in_port = 1 eth_src = 00:00:00:00:00:01 eth_dst = 00:00:00:00:00:03 0 64 0 0 OUTPUT:3 2 140 0
1 in_port = 3 eth_src = 00:00:00:00:00:03 eth_dst = 00:00:00:00:00:02 0 64 0 0 OUTPUT:2 3 238 0
1 in_port = 2 eth_src = 00:00:00:00:00:02 eth_dst = 00:00:00:00:00:03 0 64 0 0 OUTPUT:3 2 140 0
0 ANY 0 132 0 0 OUTPUT:CONTROLLER 9 546 0


Topology
Not secure
http://192.168.133.130:8080/home/topology.html
Topology
Refresh
Home
Flows
Groups
Meters
Flow Control
Group Control
Meter Control
Topology
Messages
Configuration
About
Graph
Tables
00:00:00:00:00:02
00:00:00:00:00:01
00:00:00:00:00:03
37°C Windy
Search
ENG
2:08 PM 5/21/2025



سؤالات/وظایف

سوال 5: معماری SDN چه مشکلی از شبکه های سنتی را حل میکند؟ با ذکر مثال، توضیح دهید که چگونه جداسازی plane کنترلی و دادهای در SDN باعث افزایش انعطاف پذیری شبکه میشود؟

جواب:

مشکلات اصلی شبکه‌های سنتی:

- مدیریت پیچیده :در شبکه‌های سنتی، هر دستگاه (سوئیچ/روتر) به‌صورت مستقل و با پیکربندی دستی تنظیم می‌شود.
  مثال : برای اعمال یک سیاست QoS جدید، باید تک‌تک دستگاه‌ها را تنظیم کرد.
- عدم انعطاف‌پذیری :تغییرات شبکه (مثل اضافه کردن سرویس جدید) زمان‌بر و پرخطاست.
  *مثال :*اضافه کردن یک فایروال مرکزی نیاز به تنظیم دستی ACL روی تمام روترها دارد.
- وابستگی به سخت‌افزار :شبکه‌های سنتی به‌سختی با نیازهای جدید مثل virtual networking سازگار می‌شوند.

راه‌حلSDN : جداسازی Control Plane و Data Plane

- Control Plane منطق شبکه: درSDN ، کنترلر مرکزی مثل OpenDaylight یا Ryu تصمیم‌گیری می‌کند (مثلاً مسیر جریان‌ها را تعیین می‌کند).
- Data Plane انتقال داده: سوئیچ OpenFlow فقط نقش فورواردینگ بسته‌ها را براساس قوانین کنترلر اجرا می‌کنند.

مزایای کلیدی جداسازی Plane ها:

1. مدیریت متمرکز: کنترلر دید کلی از شبکه دارد و می‌تواند سیاست‌های یکپارچه اعمال کند.
2. برنامه‌پذیری: توسعه‌دهندگان می‌توانند باAPI های کنترلر، اپلیکیشن‌های شبکه بنویسند.
3. سازگاری با مجازی‌سازی: ایجاد شبکه‌های مجازی (VXLAN, VLAN) بدون نیاز به تنظیم دستی سوئیچ‌ها.

مثال واقعی Google B4 :

- مشکل :گوگل نیاز داشت ترافیک بین دیتاسنترهایش را بهینه کند.
- راه‌حلSDN : با استفاده از کنترلر مرکزی، ترافیک را براساس شرایط لینک‌ها (مثل تاخیر یا پهنای باند) به‌صورت دینامیک مسیریابی کرد. نتیجه ان اینست که ۳۰٪ بهبود در استفاده از پهنای باند است.

سوال 6: با توجه به ساختار متمرکز کنترلر در ، SDNچه مزایا و چالشهایی برای مدیریت شبکه به وجود میآید؟ این مزایا و چالشها را در مقایسه با معماری توزیعشده شبکههای سنتی تحلیل کنید

جواب: مزایای کنترلر متمرکز در SDN

الف) مدیریت یکپارچه و ساده‌تر : درSDN ، کنترلر مرکزی به عنوان مغز شبکه عمل می‌کند و دید جامعی از کل شبکه دارد. این ویژگی امکان اعمال سیاست‌های یکپارچه را فراهم می‌سازد.

- مثال : اگر بخواهید یک سیاست کیفیت خدمات (QoS) برای ترافیک ویدیویی در کل شبکه اعمال کنید، کافی است این سیاست را یک بار در کنترلر تعریف کنید. کنترلر به طور خودکار آن را به تمام سوئیچ‌های تحت مدیریتش منتقل می‌کند.
- مقایسه با شبکه سنتی :در شبکه‌های سنتی، باید این سیاست را روی هر روتر یا سوئیچ به صورت جداگانه پیکربندی کنید که بسیار زمان‌بر و مستعد خطای انسانی است.

ب) انعطاف‌پذیری و چابکی بالا : SDN امکان اعمال تغییرات شبکه را به صورت پویا و از طریق نرم‌افزار فراهم می‌کند.

- مثال : اگر یکی از لینک‌های شبکه از کار بیفتد، کنترلر می‌تواند در کمتر از یک ثانیه مسیرهای جایگزین را محاسبه و به سوئیچ‌ها اعلام کند.
- مقایسه با شبکه سنتی :در شبکه‌های سنتی، پروتکل‌های مسیریابی توزیع‌شده مانند OSPF یا BGP ممکن است چندین ثانیه یا حتی دقیقه زمان ببرند تا همگرا شوند و مسیرهای جدید را پیدا کنند.

ج) برنامه‌پذیری و امکان توسعه : کنترلرهای SDN معمولاً API های قدرتمندی ارائه می‌دهند که امکان توسعه اپلیکیشن‌های سفارشی را فراهم می‌کنند.

- مثال : می‌توانید یک برنامه امنیتی بنویسید که ترافیک مشکوک به DDoS را شناسایی و بلافاصله مسدود کند.
- مقایسه با شبکه سنتی :در شبکه‌های سنتی، چنین قابلیت‌هایی معمولاً وابسته به سخت‌افزارهای خاص (مانند فایروال‌های اختصاصی) هستند و انعطاف‌پذیری کمتری دارند.

د) کاهش هزینه‌های عملیاتی : با خودکارسازی بسیاری از وظایف مدیریتی، SDN نیاز به مداخله دستی را کاهش می‌دهد.

- مثال : ایجاد شبکه‌های مجازی (VLAN) برای بخش‌های مختلف یک سازمان را می‌توان با چند خط کد انجام داد.
- مقایسه با شبکه سنتی :در شبکه‌های سنتی، این کار نیاز به پیکربندی دستی هر سوئیچ دارد که بسیار زمان‌بر است.

چالش‌های کنترلر متمرکز درSDN

الف) نقطه شکست واحد (Single Point of Failure) : کنترلر مرکزی به عنوان قلب شبکه SDN عمل می‌کند و خرابی آن می‌تواند کل شبکه را تحت تأثیر قرار دهد.

- مثال : اگر کنترلر Ryu از کار بیفتد، سوئیچ‌های OpenFlow ممکن است نتوانند جریان‌های جدیدی ایجاد کنند و ترافیک شبکه مختل شود.
- مقایسه با شبکه سنتی :در شبکه‌های سنتی، هر دستگاه به صورت مستقل عمل می‌کند و خرابی یک روتر معمولاً فقط بخشی از شبکه را تحت تأثیر قرار می‌دهد.

ب) چالش‌های مقیاس‌پذیری : کنترلر مرکزی ممکن است در شبکه‌های بسیار بزرگ به گلوگاه تبدیل شود.

- مثال : در یک شبکه با هزاران سوئیچ، کنترلر ممکن است نتواند به موقع به تمام درخواست‌ها پاسخ دهد.
- مقایسه با شبکه سنتی :در شبکه‌های سنتی، تصمیم‌گیری توزیع‌شده مثلاً در پروتکل BGP بار پردازشی را بین دستگاه‌ها توزیع می‌کند.

ج) مسائل امنیتی : کنترلر مرکزی می‌تواند به هدفی جذاب برای حمله‌کنندگان تبدیل شود.

- مثال :یک حمله DDoS به کنترلر می‌تواند کل شبکه را فلج کند.
- مقایسه با شبکه سنتی :در شبکه‌های سنتی، حمله به یک روتر معمولاً تأثیر محدودتری دارد.

د) پیچیدگی انتقال به SDN : مهاجرت از شبکه سنتی بهSD ن نیاز به تغییرات اساسی در زیرساخت دارد.

- مثال : ممکن است نیاز به جایگزینی سوئیچ‌های قدیمی با سوئیچ‌های OpenFlow باشد.
- مقایسه با شبکه سنتی : شبکه‌های سنتی با پروتکل‌های موجود مانند VLAN و STP سازگار هستند و نیاز به تغییرات گسترده ندارند.