



Isfahan University of Technology
CE Department

Network 2: Project 2

MohammadMahdi Ahmadi

9816743

Fall 2023

Dr. Hashemi

Contents

1	2
2	3
2.1	3
2.2	3
2.3	4
2.4	5
2.5	5
2.6	6
3	7
4	11
4.1	11
4.2	13
4.3	14
4.4	14

1

ابزار Wireshark برای شناسایی مشکلات شبکه، بسته‌های شبکه را تجزیه و تحلیل می‌کند. الگوهای ترافیکی، پروتکل‌ها و مشکلات مربوط به performance را به دقت مورد بررسی قرار می‌دهد. با قابلیت‌های filtering نمودارها و آمارها، به تعیین دقیق مشکلات کمک می‌کند. همچنین می‌تواند روندهای شبکه را visualize کند.

در زمینه شناسایی تهدیدهای امنیتی، Wireshark به عنوان یک ابزار قدرتمند برای شناسایی anomalies، تلاش‌های دسترسی غیرمجاز و فعالیت‌های malicious عمل می‌کند. این ابزار قادر به تجزیه و تحلیل محتوای packetهاست، که می‌تواند تهدیدهای امنیتی مثل exploit malware و حملات denial-of-service را آشکار کند. هرچند Wireshark قادر به رمزگشایی ترافیک رمزنگاری شده بدون کلیدهای مناسب نیست، اما هنوز می‌تواند هدرهای غیررمزنگاری شده را تجزیه و تحلیل کند. علاوه بر این، این ابزار در مانیتورینگ ارتباطات دستگاه‌های اینترنت اشیا (IoT) نیز مفید است و به شناسایی مشکلات امنیتی مرتبط با این دستگاه‌ها کمک می‌کند.

2

2.1

از پروتکل TCP و یا OpenFlow استفاده میکنند

2.2

The image shows a Wireshark packet capture from interface lo. The packet list displays several TCP packets (Nos. 4-26) and two OpenFlow packets (Nos. 27-28). The details pane for the selected OpenFlow packet (No. 27) shows it is a HELLO message (Type: OFPT_HELLO) with a Transaction ID of 2252087214. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.194566	127.0.0.1	127.0.0.1	TCP	54	5837 → 58494 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5	0.192997	127.0.0.1	127.0.0.1	TCP	74	58588 → 5837 [SYN] Seq=0 Win=43698 Len=0 MSS=65495 SACK_PERM TSval=383277197 TSecr=0 WS=512
6	0.193813	127.0.0.1	127.0.0.1	TCP	54	5837 → 58588 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7	0.194418	127.0.0.1	127.0.0.1	TCP	74	58588 → 5837 [SYN] Seq=0 Win=43698 Len=0 MSS=65495 SACK_PERM TSval=383277198 TSecr=0 WS=512
8	0.194427	127.0.0.1	127.0.0.1	TCP	54	5837 → 58588 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
9	0.195669	127.0.0.1	127.0.0.1	TCP	74	58528 → 5837 [SYN] Seq=0 Win=43698 Len=0 MSS=65495 SACK_PERM TSval=383277199 TSecr=0 WS=512
10	0.195677	127.0.0.1	127.0.0.1	TCP	54	5837 → 58528 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
11	0.196927	127.0.0.1	127.0.0.1	TCP	74	58528 → 5837 [SYN] Seq=0 Win=43698 Len=0 MSS=65495 SACK_PERM TSval=383277201 TSecr=0 WS=512
12	0.196930	127.0.0.1	127.0.0.1	TCP	54	5837 → 58528 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
13	0.198463	127.0.0.1	127.0.0.1	TCP	74	58542 → 5837 [SYN] Seq=0 Win=43698 Len=0 MSS=65495 SACK_PERM TSval=383277202 TSecr=0 WS=512
14	0.198480	127.0.0.1	127.0.0.1	TCP	54	5837 → 58542 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15	0.199595	127.0.0.1	127.0.0.1	TCP	74	58558 → 5837 [SYN] Seq=0 Win=43698 Len=0 MSS=65495 SACK_PERM TSval=383277204 TSecr=0 WS=512
16	0.199606	127.0.0.1	127.0.0.1	TCP	54	5837 → 58558 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
17	0.200762	127.0.0.1	127.0.0.1	TCP	74	58566 → 5837 [SYN] Seq=0 Win=43698 Len=0 MSS=65495 SACK_PERM TSval=383277204 TSecr=0 WS=512
18	0.200761	127.0.0.1	127.0.0.1	TCP	54	5837 → 58566 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
19	0.203210	127.0.0.1	127.0.0.1	TCP	74	58570 → 5837 [SYN] Seq=0 Win=43698 Len=0 MSS=65495 SACK_PERM TSval=383277206 TSecr=0 WS=512
20	0.201923	127.0.0.1	127.0.0.1	TCP	54	5837 → 58570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
21	0.203819	127.0.0.1	127.0.0.1	TCP	74	58588 → 5837 [SYN] Seq=0 Win=43698 Len=0 MSS=65495 SACK_PERM TSval=383277207 TSecr=0 WS=512
22	0.203827	127.0.0.1	127.0.0.1	TCP	54	5837 → 58588 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
23	0.210771	127.0.0.1	127.0.0.1	TCP	74	38372 → 6653 [SYN] Seq=0 Win=43698 Len=0 MSS=65495 SACK_PERM TSval=383277222 TSecr=0 WS=512
24	0.218795	127.0.0.1	127.0.0.1	TCP	74	6653 → 38372 [SYN, ACK] Seq=0 Ack=1 Win=43698 Len=0 MSS=65495 SACK_PERM TSval=383277223 TSecr=383277222 WS=512
25	0.218813	127.0.0.1	127.0.0.1	TCP	66	38372 → 6653 [ACK] Seq=1 Ack=1 Win=44032 Len=0 TSval=383277223 TSecr=383277223
26	0.219041	127.0.0.1	127.0.0.1	OpenFlow	74	TYPE: OFPT_HELLO
27	0.219057	127.0.0.1	127.0.0.1	TCP	66	38372 → 6653 [ACK] Seq=1 Ack=1 Win=44032 Len=0 TSval=383277223 TSecr=383277223
28	0.219688	127.0.0.1	127.0.0.1	OpenFlow	74	Type: OFPT_HELLO
29	0.219715	127.0.0.1	127.0.0.1	TCP	66	6653 → 38372 [ACK] Seq=0 Ack=1 Win=44032 Len=0 TSval=383277223 TSecr=383277223
30	0.219868	127.0.0.1	127.0.0.1	OpenFlow	74	Type: OFPT_FEATURES_REQUEST
31	0.219874	127.0.0.1	127.0.0.1	OpenFlow	78	Type: OFPT_FEATURES_REPLY

Frame 26: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface lo, id 0
 Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
 Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 Transmission Control Protocol, Src Port: 6653, Dst Port: 38372, Seq: 1, Ack: 1, Len: 8
 OpenFlow 1.0
 .000 0001 - Version: 1.0 (0x01)
 Type: OFPT_HELLO (0)
 Length: 8
 Transaction ID: 2252087214

2.3

وقتی یک سویچ به شبکه SDN اضافه میشود، یک پیام Feature_Request از کنترلر دریافت میکند که به معنای این است که کنترلر میخواهد اطلاعات آن سویچ را دریافت کند. در پاسخ سویچ اطلاعات خود را در قالب پیام Feature_Reply به کنترلر ارسال میکند. این اطلاعات شامل ورژن OpenFlow سویچ، ماکزیمم اندازه ورودی Flow ها، action های پشتیبانی شده و ... است.

The screenshot shows a Wireshark packet capture titled "Capturing from Loopback: lo". The packet list shows several TCP and OpenFlow packets. Two green arrows highlight specific OpenFlow packets:

- Packet 39: OFPT_FEATURES_REQUEST (5) from 127.0.0.1 to 127.0.0.1. Details: OpenFlow 1.0, .000 0001 = Version: 1.0 (0x01), Type: OFPT_FEATURES_REQUEST (5), Length: 8, Transaction ID: 2785423896.
- Packet 38: OFPT_FEATURES_REPLY (5) from 127.0.0.1 to 127.0.0.1. Details: OpenFlow 1.0, .000 0001 = Version: 1.0 (0x01), Type: OFPT_FEATURES_REPLY (5), Length: 8, Transaction ID: 2785423896.

The packet details for the selected packet (No. 39) are shown below:

```

Frame 39: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 6653, Dst Port: 38372, Seq: 9, Ack: 9, Len: 8
OpenFlow 1.0
.000 0001 = Version: 1.0 (0x01)
Type: OFPT_FEATURES_REQUEST (5)
Length: 8
Transaction ID: 2785423896
  
```

2.4

The image shows a Wireshark packet capture interface. The packet list pane displays a series of packets, including TCP SYN, RST, and ACK packets, as well as OpenFlow packets. The packet details pane for packet 35 (OpenFlow 1.0) is expanded, showing the OFPT_PACKET_IN structure. The packet bytes pane shows the raw data of the packet.

2.5

دلیل اول زمانی است که سوییچ نمیداند باید با بسته ورودی چه کند، پس بسته را برای کنترلر میفرستد تا کنترلر تصمیم بگیرد. در نهایت ممکن است کنترلر FlowTable ها را در سوییچ ها آپدیت کند. دلیل دوم زمانی است که کنترلر صراحتاً گفته باشد که بسته های خاصی را برای من بفرست.

2.6

برای بسته ها پروتکل OpenFlow استفاده شده است.

سوییچ نتوانسته است که مسیریابی را به درستی انجام بدهد و table-miss اتفاق افتاده است. پس بسته های icmp در بسته های OpenFlow گنجانده شده اند و به کنترلر فرستاده شدند تا کنترل مسیریابی را انجام بدهد.

The screenshot shows a Wireshark capture on interface 'lo'. The packet list contains three entries:

No.	Time	Source	Destination	Protocol	Length	Info
706	656.8252..	10.0.0.1	10.0.0.2	OpenFlow	182	Type: OFPT_PACKET_IN
710	656.8259..	10.0.0.1	10.0.0.2	OpenFlow	182	Type: OFPT_PACKET_OUT
712	656.8259..	10.0.0.2	10.0.0.1	OpenFlow	188	Type: OFPT_PACKET_OUT

The details pane for the selected packet (No. 712) shows the following structure:

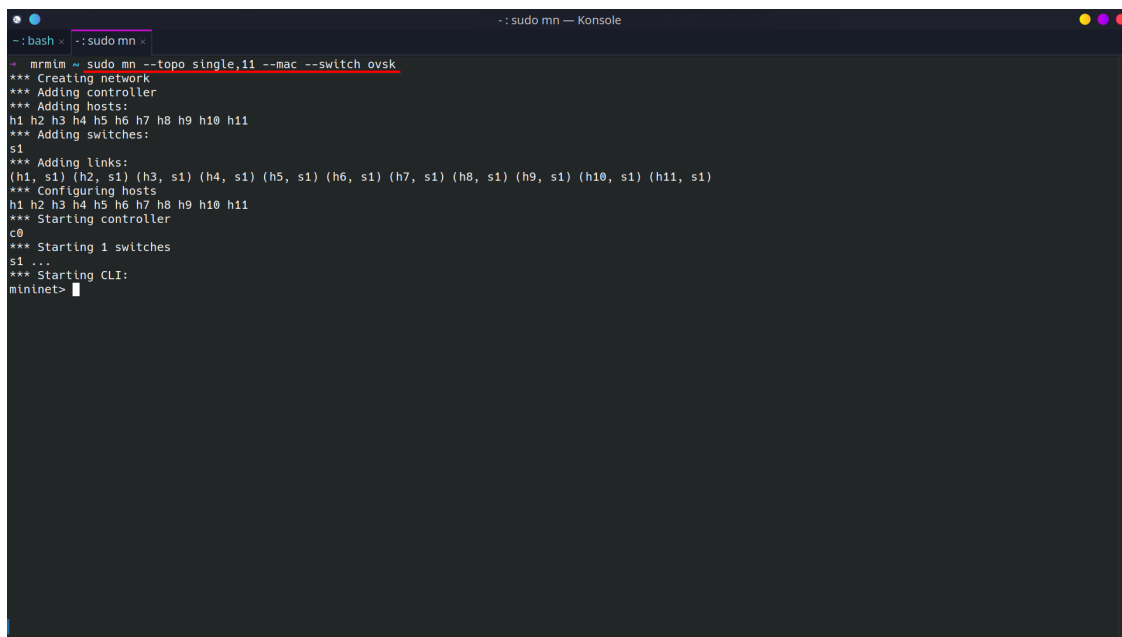
- Frame 708: 188 bytes on wire (1504 bits), 188 bytes captured (1504 bits) on interface lo, capture length 188
- Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
- Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- Transmission Control Protocol, Src Port: 6653, Dst Port: 38972, Seq: 6097, Ack: 6069, Len: 188
- OpenFlow 1.0
 - .000 0001 = Version: 1.0 (0x01)
 - Type: OFPT_PACKET_OUT (13)
 - Length: 122
 - Transaction ID: 0
 - Buffer ID: 0xffffffff
 - In port: 1
 - Actions length: 8
 - Actions type: Output to switch port (0)
 - Action length: 8
 - Output port: 2
 - Max length: 0
 - Ethernet II, Src: 7a:86:07:65:bc:02 (7a:86:07:65:bc:02), Dst: c2:0b:2d:10:58:e0 (c2:0b:2d:10:58:e0)

The packet bytes pane shows the raw data of the packet, including the OpenFlow header and the Ethernet II frame.

3

1. `sudo mn --topo single,11 --mac --switch ovsk`

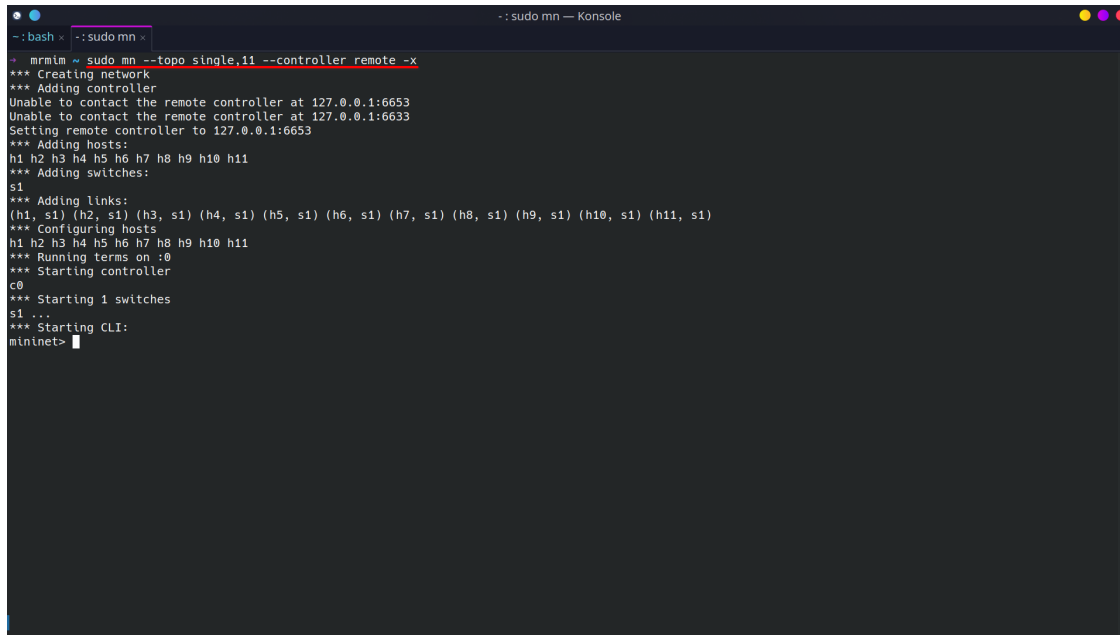
- `--topo single,11`: Creates a single-switch topology with 11 hosts.
- `--mac`: Assigns MAC addresses to hosts.
- `--switch ovsk`: Specifies that Open vSwitch should be used as the switch.



```
--:bash x --:sudo mn x
mrm ~ sudo mn --topo single,11 --mac --switch ovsk
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1) (h7, s1) (h8, s1) (h9, s1) (h10, s1) (h11, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```


2. `sudo mn --topo single,11 --controller remote -x`

- `--topo single,11`: Creates a single-switch topology with 11 hosts.
- `--controller remote`: Configures a remote controller for SDN functionality.
- `-x`: Enables Mininet's CLI (Command Line Interface) after launching.



```
~: bash x  ~: sudo mn x
+ mrmim ~ sudo mn --topo single,11 --controller remote -x
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1) (h7, s1) (h8, s1) (h9, s1) (h10, s1) (h11, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11
*** Running terms on :0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

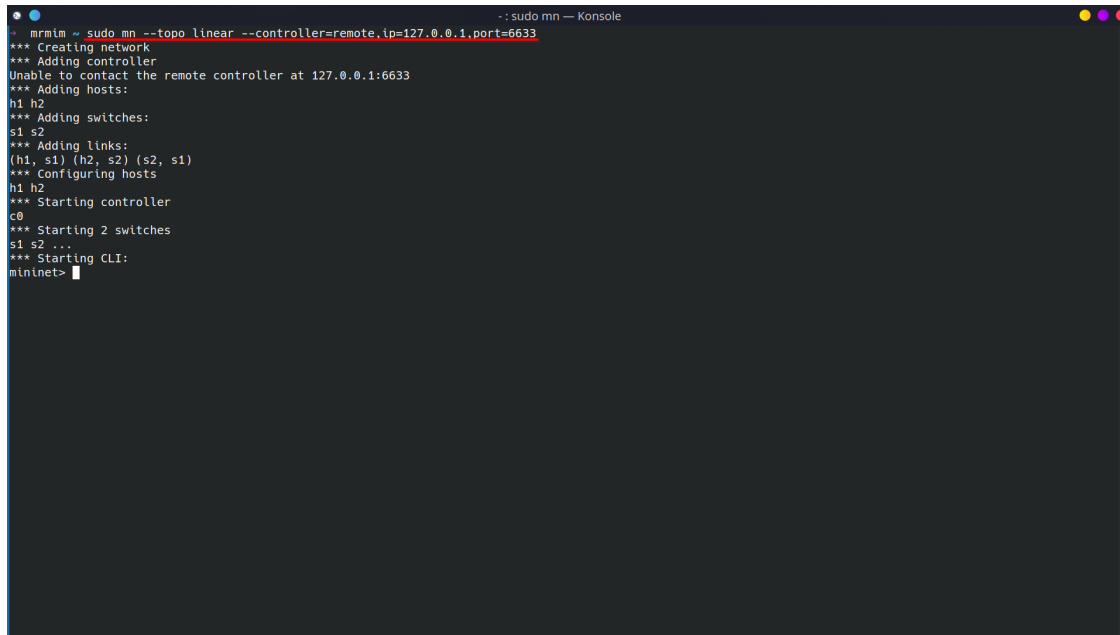
3. `sudo mn -topo tree,11 -mac -arp`

- `--topo tree,11`: Creates a tree topology with 2^{11} hosts and $2^{11} - 1$ switches.
- `--mac`: Assigns MAC addresses to hosts.
- `--arp`: Enables ARP (Address Resolution Protocol) in the network.

```
--: bash — Konsole
+ mrmim ~ sudo mn --topo tree,11 --mac --arp
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42
h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64 h65 h66 h67 h68 h69 h70 h71 h72 h73 h74 h75 h76 h77 h78 h79 h80 h81
h82 h83 h84 h85 h86 h87 h88 h89 h90 h91 h92 h93 h94 h95 h96 h97 h98 h99 h100 h101 h102 h103 h104 h105 h106 h107 h108 h109 h110 h111 h112 h113 h114 h115 h116
h117 h118 h119 h120 h121 h122 h123 h124 h125 h126 h127 h128 h129 h130 h131 h132 h133 h134 h135 h136 h137 h138 h139 h140 h141 h142 h143 h144 h145 h146 h147 h
148 h149 h150 h151 h152 h153 h154 h155 h156 h157 h158 h159 h160 h161 h162 h163 h164 h165 h166 h167 h168 h169 h170 h171 h172 h173 h174 h175 h176 h177 h178 h179
h180 h181 h182 h183 h184 h185 h186 h187 h188 h189 h190 h191 h192 h193 h194 h195 h196 h197 h198 h199 h200 h201 h202 h203 h204 h205 h206 h207 h208 h209 h210
h211 h212 h213 h214 h215 h216 h217 h218 h219 h220 h221 h222 h223 h224 h225 h226 h227 h228 h229 h230 h231 h232 h233 h234 h235 h236 h237 h238 h239 h240 h241 h242
h243 h244 h245 h246 h247 h248 h249 h250 h251 h252 h253 h254 h255 h256 h257 h258 h259 h260 h261 h262 h263 h264 h265 h266 h267 h268 h269 h270 h271 h272 h273
h274 h275 h276 h277 h278 h279 h280 h281 h282 h283 h284 h285 h286 h287 h288 h289 h290 h291 h292 h293 h294 h295 h296 h297 h298 h299 h300 h301 h302 h303 h304 h305
h306 h307 h308 h309 h310 h311 h312 h313 h314 h315 h316 h317 h318 h319 h320 h321 h322 h323 h324 h325 h326 h327 h328 h329 h330 h331 h332 h333 h334 h335 h336
h337 h338 h339 h340 h341 h342 h343 h344 h345 h346 h347 h348 h349 h350 h351 h352 h353 h354 h355 h356 h357 h358 h359 h360 h361 h362 h363 h364 h365 h366 h367
h368 h369 h370 h371 h372 h373 h374 h375 h376 h377 h378 h379 h380 h381 h382 h383 h384 h385 h386 h387 h388 h389 h390 h391 h392 h393 h394 h395 h396 h397 h398 h399
h400 h401 h402 h403 h404 h405 h406 h407 h408 h409 h410 h411 h412 h413 h414 h415 h416 h417 h418 h419 h420 h421 h422 h423 h424 h425 h426 h427 h428 h429 h430
h431 h432 h433 h434 h435 h436 h437 h438 h439 h440 h441 h442 h443 h444 h445 h446 h447 h448 h449 h450 h451 h452 h453 h454 h455 h456 h457 h458 h459 h460 h461 h462
h463 h464 h465 h466 h467 h468 h469 h470 h471 h472 h473 h474 h475 h476 h477 h478 h479 h480 h481 h482 h483 h484 h485 h486 h487 h488 h489 h490 h491 h492 h493
h494 h495 h496 h497 h498 h499 h500 h501 h502 h503 h504 h505 h506 h507 h508 h509 h510 h511 h512 h513 h514 h515 h516 h517 h518 h519 h520 h521 h522 h523 h524
h525 h526 h527 h528 h529 h530 h531 h532 h533 h534 h535 h536 h537 h538 h539 h540 h541 h542 h543 h544 h545 h546 h547 h548 h549 h550 h551 h552 h553 h554 h555 h556
h557 h558 h559 h560 h561 h562 h563 h564 h565 h566 h567 h568 h569 h570 h571 h572 h573 h574 h575 h576 h577 h578 h579 h580 h581 h582 h583 h584 h585 h586 h587 h588
h589 h590 h591 h592 h593 h594 h595 h596 h597 h598 h599 h600 h601 h602 h603 h604 h605 h606 h607 h608 h609 h610 h611 h612 h613 h614 h615 h616 h617 h618 h619
h620 h621 h622 h623 h624 h625 h626 h627 h628 h629 h630 h631 h632 h633 h634 h635 h636 h637 h638 h639 h640 h641 h642 h643 h644 h645 h646 h647 h648 h649 h650
h651 h652 h653 h654 h655 h656 h657 h658 h659 h660 h661 h662 h663 h664 h665 h666 h667 h668 h669 h670 h671 h672 h673 h674 h675 h676 h677 h678 h679 h680 h681
h682 h683 h684 h685 h686 h687 h688 h689 h690 h691 h692 h693 h694 h695 h696 h697 h698 h699 h700 h701 h702 h703 h704 h705 h706 h707 h708 h709 h710 h711 h712 h713
h714 h715 h716 h717 h718 h719 h720 h721 h722 h723 h724 h725 h726 h727 h728 h729 h730 h731 h732 h733 h734 h735 h736 h737 h738 h739 h740 h741 h742 h743 h744
h745 h746 h747 h748 h749 h750 h751 h752 h753 h754 h755 h756 h757 h758 h759 h760 h761 h762 h763 h764 h765 h766 h767 h768 h769 h770 h771 h772 h773 h774 h775 h776
h777 h778 h779 h780 h781 h782 h783 h784 h785 h786 h787 h788 h789 h790 h791 h792 h793 h794 h795 h796 h797 h798 h799 h800 h801 h802 h803 h804 h805 h806 h807 h808
h809 h810 h811 h812 h813 h814 h815 h816 h817 h818 h819 h820 h821 h822 h823 h824 h825 h826 h827 h828 h829 h830 h831 h832 h833 h834 h835 h836 h837 h838 h839
h840 h841 h842 h843 h844 h845 h846 h847 h848 h849 h850 h851 h852 h853 h854 h855 h856 h857 h858 h859 h860 h861 h862 h863 h864 h865 h866 h867 h868 h869 h870 h871
h872 h873 h874 h875 h876 h877 h878 h879 h880 h881 h882 h883 h884 h885 h886 h887 h888 h889 h890 h891 h892 h893 h894 h895 h896 h897 h898 h899 h900 h901 h902 h903
h904 h905 h906 h907 h908 h909 h910 h911 h912 h913 h914 h915 h916 h917 h918 h919 h920 h921 h922 h923 h924 h925 h926 h927 h928 h929 h930 h931 h932 h933 h934
h935 h936 h937 h938 h939 h940 h941 h942 h943 h944 h945 h946 h947 h948 h949 h950 h951 h952 h953 h954 h955 h956 h957 h958 h959 h960 h961 h962 h963 h964 h965
h966 h967 h968 h969 h970 h971 h972 h973 h974 h975 h976 h977 h978 h979 h980 h981 h982 h983 h984 h985 h986 h987 h988 h989 h990 h991 h992 h993 h994 h995
h996 h997 h998 h999 h1000 h1001 h1002 h1003 h1004 h1005 h1006 h1007 h1008 h1009 h1010 h1011 h1012 h1013 h1014 h1015 h1016 h1017 h1018 h1019 h1020 h1021 h1022
h1023 h1024 h1025 h1026 h1027 h1028 h1029 h1030 h1031 h1032 h1033 h1034 h1035 h1036 h1037 h1038 h1039 h1040 h1041 h1042 h1043 h1044 h1045 h1046 h1047 h1048
h1049 h1050 h1051 h1052 h1053 h1054 h1055 h1056 h1057 h1058 h1059 h1060 h1061 h1062 h1063 h1064 h1065 h1066 h1067 h1068 h1069 h1070 h1071 h1072 h1073 h1074 h1075
h1076 h1077 h1078 h1079 h1080 h1081 h1082 h1083 h1084 h1085 h1086 h1087 h1088 h1089 h1090 h1091 h1092 h1093 h1094 h1095 h1096 h1097 h1098 h1099 h1100 h1101
h1102 h1103 h1104 h1105 h1106 h1107 h1108 h1109 h1110 h1111 h1112 h1113 h1114 h1115 h1116 h1117 h1118 h1119 h1120 h1121 h1122 h1123 h1124 h1125 h1126 h1127
h1128 h1129 h1130 h1131 h1132 h1133 h1134 h1135 h1136 h1137 h1138 h1139 h1140 h1141 h1142 h1143 h1144 h1145 h1146 h1147 h1148 h1149 h1150 h1151 h1152 h1153
h1154 h1155 h1156 h1157 h1158 h1159 h1160 h1161 h1162 h1163 h1164 h1165 h1166 h1167 h1168 h1169 h1170 h1171 h1172 h1173 h1174 h1175 h1176 h1177 h1178 h1179
h1180 h1181 h1182 h1183 h1184 h1185 h1186 h1187 h1188 h1189 h1190 h1191 h1192 h1193 h1194 h1195 h1196 h1197 h1198 h1199 h1200 h1201 h1202 h1203 h1204 h1205
```

4. `sudo mn -topo linear --controller=remote,ip=127.0.0.1,port=6633`

- `--topo linear`: Creates a linear topology (a straight line) with a default of 2 switches.
- `--controller=remote,ip=127.0.0.1,port=6633`: Configures a remote controller with the specified IP address (127.0.0.1) and port (6633).



```
- mrm ~ sudo mn --topo linear --controller=remote,ip=127.0.0.1,port=6633
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s2) (s2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet>
```

4

4.1

```

→ mrmim Project2 sudo python3 A/4_1.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7 s8
*** Adding links:
(h1, s1) (h2, s2) (h3, s2) (h4, s3) (h5, s4) (h6, s5) (h7, s5) (h8, s5) (h9, s6) (h10, s7) (h11, s7) (h12, s8) (s1, s2) (s2, s3) (s2, s5) (s4, s5) (s5, s6) (s6, s7) (s7, s8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
*** Starting controller
c0
*** Starting 8 switches
s1 s2 s3 s4 s5 s6 s7 s8 ...
h1 h1-eth0:s1-eth1
h2 h2-eth0:s2-eth1
h3 h3-eth0:s2-eth2
h4 h4-eth0:s3-eth1
h5 h5-eth0:s4-eth1
h6 h6-eth0:s5-eth1
h7 h7-eth0:s5-eth2
h8 h8-eth0:s5-eth3
h9 h9-eth0:s6-eth1
h10 h10-eth0:s7-eth1
h11 h11-eth0:s7-eth2
h12 h12-eth0:s8-eth1
*** Starting controller
c0
*** Starting 8 switches
s1 s2 s3 s4 s5 s6 s7 s8 ...
*** Starting CLI:
mininet>

```

```

1 from mininet.net import Mininet
2 from mininet.topo import Topo
3 from mininet.util import dumpNodeConnections
4 from mininet.log import setLogLevel
5
6
7 class MyTopology(Topo):
8     def __init__(self):
9         Topo.__init__(self)
10
11         # Adding switches
12         s1 = self.addSwitch('s1')
13         s2 = self.addSwitch('s2')
14         s3 = self.addSwitch('s3')
15         s4 = self.addSwitch('s4')
16         s5 = self.addSwitch('s5')
17         s6 = self.addSwitch('s6')
18         s7 = self.addSwitch('s7')
19         s8 = self.addSwitch('s8')
20
21         # Adding hosts
22         h1 = self.addHost('h1')
23         h2 = self.addHost('h2')

```

```
24     h3 = self.addHost('h3')
25     h4 = self.addHost('h4')
26     h5 = self.addHost('h5')
27     h6 = self.addHost('h6')
28     h7 = self.addHost('h7')
29     h8 = self.addHost('h8')
30     h9 = self.addHost('h9')
31     h10 = self.addHost('h10')
32     h11 = self.addHost('h11')
33     h12 = self.addHost('h12')
34
35     # Connecting hosts to switches
36     self.addLink(h1, s1)
37     self.addLink(h2, s2)
38     self.addLink(h3, s2)
39     self.addLink(h4, s3)
40     self.addLink(h5, s4)
41     self.addLink(h6, s5)
42     self.addLink(h7, s5)
43     self.addLink(h8, s5)
44     self.addLink(h9, s6)
45     self.addLink(h10, s7)
46     self.addLink(h11, s7)
47     self.addLink(h12, s8)
48
49     # Connecting switches
50     self.addLink(s1, s2)
51     self.addLink(s2, s3)
52     self.addLink(s2, s5)
53     self.addLink(s4, s5)
54     self.addLink(s5, s6)
55     self.addLink(s6, s7)
56     self.addLink(s7, s8)
57
58
59 def runExperiment():
60     topo = MyTopology()
61     net = Mininet(topo)
62     net.start()
```

```

63
64     dumpNodeConnections(net.hosts)
65     net.interact()
66
67     net.stop()
68
69 if __name__ == '__main__':
70     # Tell mininet to print useful information
71     setLogLevel('info')
72     runExperiment()

```

4.2

```

h2 h2-eth0:s2-eth1
h3 h3-eth0:s2-eth2
h4 h4-eth0:s3-eth1
h5 h5-eth0:s4-eth1
h6 h6-eth0:s5-eth1
h7 h7-eth0:s5-eth2
h8 h8-eth0:s5-eth3
h9 h9-eth0:s6-eth1
h10 h10-eth0:s7-eth1
h11 h11-eth0:s7-eth2
h12 h12-eth0:s8-eth1
*** Starting controller
c0
*** Starting 8 switches
s1 s2 s3 s4 s5 s6 s7 s8 ...
*** Starting CLI:
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.81 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.980 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.144 ms
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.144/3.311/8.809/3.902 ms
mininet> h3 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data:
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.87 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=1.14 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.145 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.138 ms
^C
--- 10.0.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3012ms
rtt min/avg/max/mdev = 0.138/2.572/8.872/3.659 ms
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10 h11 h12
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10 h11 h12
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10 h11 h12
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10 h11 h12
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10 h11 h12
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10 h11 h12
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10 h11 h12
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h11 h12
h11 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h12
h12 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11
*** Results: 0% dropped (132/132 received)
mininet>

```

4.3

علت این است که سوییچ ها در ابتدا باید مسیریابی به مقصد را نمیشناسند و باید با ارتباط گرفتن با کنترلر، FlowTable خود را آپدیت کنند

4.4

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=32542>
<Host h2: h2-eth0:10.0.0.2 pid=32544>
<Host h3: h3-eth0:10.0.0.3 pid=32546>
<Host h4: h4-eth0:10.0.0.4 pid=32548>
<Host h5: h5-eth0:10.0.0.5 pid=32553>
<Host h6: h6-eth0:10.0.0.6 pid=32565>
<Host h7: h7-eth0:10.0.0.7 pid=32567>
<Host h8: h8-eth0:10.0.0.8 pid=32569>
<Host h9: h9-eth0:10.0.0.9 pid=32571>
<Host h10: h10-eth0:10.0.0.10 pid=32573>
<Host h11: h11-eth0:10.0.0.11 pid=32575>
<Host h12: h12-eth0:10.0.0.12 pid=32594>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=32599>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None,s2-eth4:None,s2-eth5:None pid=32602>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=32605>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None pid=32612>
<OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth2:None,s5-eth3:None,s5-eth4:None,s5-eth5:None,s5-eth6:None pid=32615>
<OVSSwitch s6: lo:127.0.0.1,s6-eth1:None,s6-eth2:None,s6-eth3:None pid=32622>
<OVSSwitch s7: lo:127.0.0.1,s7-eth1:None,s7-eth2:None,s7-eth3:None,s7-eth4:None pid=32649>
<OVSSwitch s8: lo:127.0.0.1,s8-eth1:None,s8-eth2:None pid=32657>
<Controller c0: 127.0.0.1:6653 pid=32535>
mininet>
mininet> nodes
available nodes are:
c0 h1 h10 h11 h12 h2 h3 h4 h5 h6 h7 h8 h9 s1 s2 s3 s4 s5 s6 s7 s8
mininet>
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10 h11 h12
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10 h11 h12
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10 h11 h12
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10 h11 h12
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10 h11 h12
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10 h11 h12
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10 h11 h12
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h11 h12
h11 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h12
h12 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11
*** Results: 0% dropped (132/132 received)
mininet>
```