

# بسم الله الرحمن الرحيم

ارزیابی مبتنی بر شبیه سازی عملکرد TCP چندمسیری در پشتیبانی از انتقال  
زیر جریان های ترافیکی

محمد مهدی سوری

حمیدرضا آذرباد

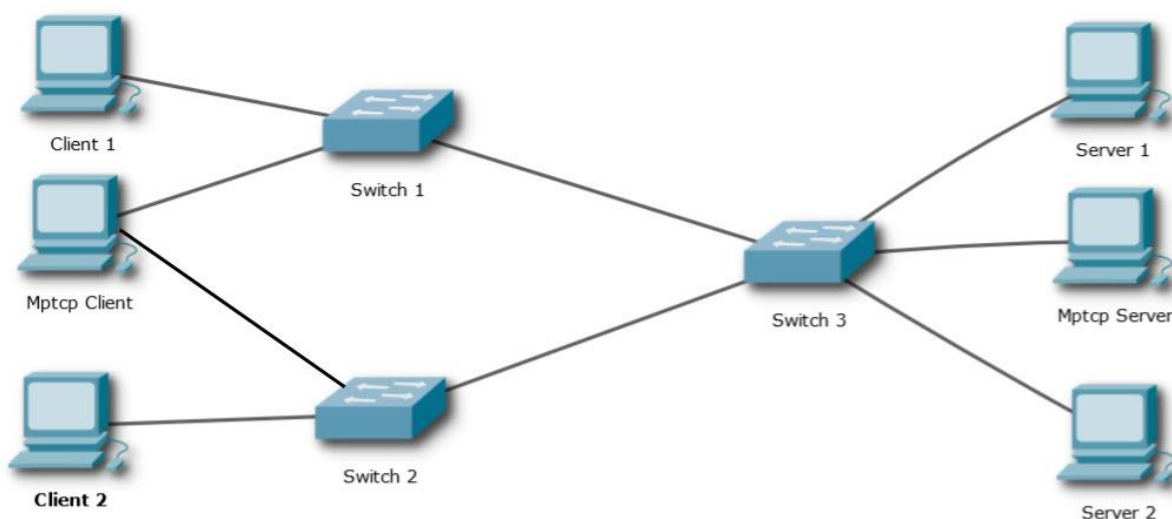
## مقدمه

در این پروژه قصد داریم که با شبیه سازی TCP چند مسیره (MPTCP) درون محیط شبیه ساز Mininet، و با استفاده از پوسته Minitopo عملکرد این پروتکل را در حالت های مختلفی که برای کنترل ازدحام (Congestion Control)، زمانبند (Scheduler) و مدیریت مسیر (Path Manager) تحت یک توپولوژی و سناریو خاص بررسی کرده و نتایج را گزارش کنیم.

ابتدا توپولوژی و سناریو را شرح می دهیم سپس الگوریتم ها را معرفی کرده و نتایج را گزارش می کنیم.

## توپولوژی

ابتدا سه هاست را ایجاد می کنیم به عنوان Client که دو تا از آن ها به صورت TCP معمولی عمل می کنند و سومی به صورت MPTCP عمل می کند. کلاینتی که از MPTCP استفاده می کند به دو سوئیچ متصل است و کلاینت های دیگر هر کدام به یک سوئیچ متصل می شوند. این دو سوئیچ به سوئیچ سوم که مخصوص هاست های سرور ها است متصل می شوند و این دو لینک، لینک های bottleneck مان هستند. سه سرور هم ایجاد می کنیم که یکی از آن ها مخصوص MPTCP و دو تای دیگر برای اتصالات TCP معمولی هستند. شکل زیر چگونگی اتصالات این توپولوژی را نمایش می دهد.



سرعت و تاخیر لینک های bottleneck به ترتیب 20Mbps و 20ms است. سرعت و تاخیر مابقی لینک ها به ترتیب 100Mbps و 0.1ms است.

## سناریو

ابتدا از Mptcp client به Mptcp server به مدت 50 ثانیه داده ارسال می کنیم. در همین حین پس از 10 ثانیه از client 1 به server 1 نیز داده را به مدت 20 ثانیه به صورت TCP معمولی ارسال می کنیم. در زمان 20 ثانیه پس از شروع از client 2 نیز به server 2 داده را به مدت 20 ثانیه به صورت TCP معمولی ارسال می کنیم. سپس پهنای باند و میزان ترافیک انتقال یافته را یافته و نتایج را گزارش می کنیم.

## الگوریتم

### • کنترل ازدحام (Congestion Control)

به دلیل آنکه فرستادن صرفاً افزایشی تعداد بسته ها موجب ازدحام شبکه و از دست رفتن بسته ها می شود لذا الگوریتم هایی پیاده سازی شده است که با توجه به وضعیت شبکه تعداد بسته های ارسالی را کم و زیاد می کند به این الگوریتم ها الگوریتم های کنترل ازدحام می گوئیم.

تفاوت الگوریتم های کنترل ازدحام به دو بخش بستگی دارد یک اینکه با چه فرمولی افزایش پنجره ازدحام انجام می شود و دو اینکه زمانی که بسته ای از دست می رود که به صورت ضمنی نشاندهنده ازدحام در شبکه است پنجره ازدحام به چه صورتی کاهش می یابد. در شکل زیر این فرمول ها برای الگوریتم های wVegas, BALIA, OLIA, LIA, آمده است.

TABLE I  
SUMMARY OF MULTIPATH-COUPLED CONGESTION CONTROLS DEPLOYED IN THE MPTCP LINUX KERNEL IMPLEMENTATIONS.

	Alpha Parameter	Congestion Window Increase	Congestion Window Decrease
<b>LIA:</b>	$\alpha = w_{\text{total}} \frac{\max(w_r / \tau_r^2)}{[\sum_{k \in \mathcal{R}} (w_k / \tau_k)]^2}$	For each ACK received on subflow $r$ , $w_r = w_r + \min\left(\frac{\alpha}{w_{\text{total}}}, \frac{1}{w_r}\right)$	For each packet loss on subflow $r$ , $w_r = w_r - \frac{w_r}{2},$ as in standard TCP [18]
<b>OLIA:</b>	$\alpha_r = \begin{cases} \frac{1/ \mathcal{P} }{ C }, & \text{if } r \in C, \\ \frac{1/ \mathcal{P} }{ W }, & \text{if } r \in W,  C  > 0, \\ 0, & \text{otherwise} \end{cases}$	For each ACK received on subflow $r$ , $w_r = w_r + \left( \frac{w_r / \tau_r^2}{[\sum_{k \in \mathcal{R}} (w_k / \tau_k)]^2} + \frac{\alpha_r}{w_r} \right)$	For each packet loss on subflow $r$ , $w_r = w_r - \frac{w_r}{2},$ as in standard TCP [18]
<b>BALIA:</b>	$\alpha_r = \frac{\max\{x\}}{x_r}, \text{ where } x_r = \frac{w_r}{\tau_r},$ and $x = \left[ \sum_{k \in \mathcal{P}} (w_k / \tau_k) \right]^2$	For each ACK received on subflow $r$ , $w_r = w_r + \left[ \frac{x_r}{\tau_r} \frac{(1 + \alpha_r)}{2} \right] \left( \frac{4 + \alpha_r}{5} \right)$	For each packet loss on subflow $r$ , $w_r = w_r - \left[ \frac{w_r}{2} \min\left(\alpha_r, \frac{3}{2}\right) \right]$
<b>wVegas:</b>	if $(\delta_r > \alpha_r)$ , then $\alpha_r = \omega_r \alpha_{\text{total}}$ , where $\delta_r = \left( \frac{w_r}{\hat{\tau}_r} - \frac{w_r}{\bar{\tau}_r} \right) \hat{\tau}_r$ , $\omega_r = \frac{x_r}{\sum_{k \in \mathcal{R}} x_k}$ , and $x_i = \frac{w_i}{\bar{\tau}_i}$	For each transmission round on subflow $r$ , $w_r = \begin{cases} w_r - 1, & \text{if } \delta_r > \alpha_r, \\ w_r + 1, & \text{if } \delta_r < \alpha_r, \end{cases}$ if $(q_r > \bar{\tau}_r - \hat{\tau}_r)$ , then $q_r = \bar{\tau}_r - \hat{\tau}_r$ if $(\bar{\tau}_r - \hat{\tau}_r \geq 2q_r)$ , then $w_r = w_r \frac{\hat{\tau}_r}{2\bar{\tau}_r}$	For each packet loss on subflow $r$ , $w_r = w_r - \frac{w_r}{2},$ as in standard TCP [18]

## • زمانبند (Scheduler)

در حالت MPTCP هنگامی که می خواهیم بسته ای را ارسال کنیم باید انتخاب کنیم که از کدام زیرجریان ترافیکی آن را می خواهیم ارسال کنیم در نتیجه باید به زمانبند داشته باشیم که طبق الگوریتم خاصی آن بسته را به زیر جریان مربوطه ارسال کند.

### 1. الگوریتم LR (Lowest RTT first)

در MPTCP الگوریتم پیشفرض است و به این صورت است که سهم بیشتر تعداد بسته های ارسالی جدید را به زیرجریانی که RTT یک بسته در آن کمتر از RTT بسته ای در زیرجریان دیگر باشد می دهد.

### 2. الگوریتم Round Robin

این الگوریتم به این صورت کار می کند که سهم تعداد بسته های جدید در هر زیرجریان برابر است.

## • مدیریت مسیر (Path Manager)

مدیریت مسیر نیز در MPTCP فقط معنا پیدا می کند به این معنی که چه زمانی باید زیرجریان ترافیکی جدیدی اضافه کرد یا آن را حذف کرد. این کار و نحوه پیاده سازی آن توسط الگوریتم های مدیریت مسیر انجام می شود.

### **1. الگوریتم Fullmesh**

در این الگوریتم به ازای هر زوج آدرس IP کلاینت و سرور یک زیر جریان ترافیکی ایجاد می شود.

### **2. الگوریتم ndiffports**

در این الگوریتم به ازای یک زوج آدرس IP کلاینت و سرور بیش از یک زیر جریان ترافیکی (به صورت پیشفرض دو) ایجاد می شود. روش این کار این است که شماره پورت source را متفاوت قرار می دهد.

## نحوه اجرای پروژه

ابتدا نرم افزار های [Vagrant](#) و [VirtualBox](#) را دانلود و نصب کنید. سپس با دستور زیر پروژه sigcomm20\_mptp\_tutorial را clone کنید.

```
git clone https://github.com/qdeconinck/sigcomm20_mptp_tutorial.git
```

حال در ترمینال به پوشه sigcomm20\_mptp\_tutorial رفته و در آنجا دستور زیر را اجرا می کنیم.

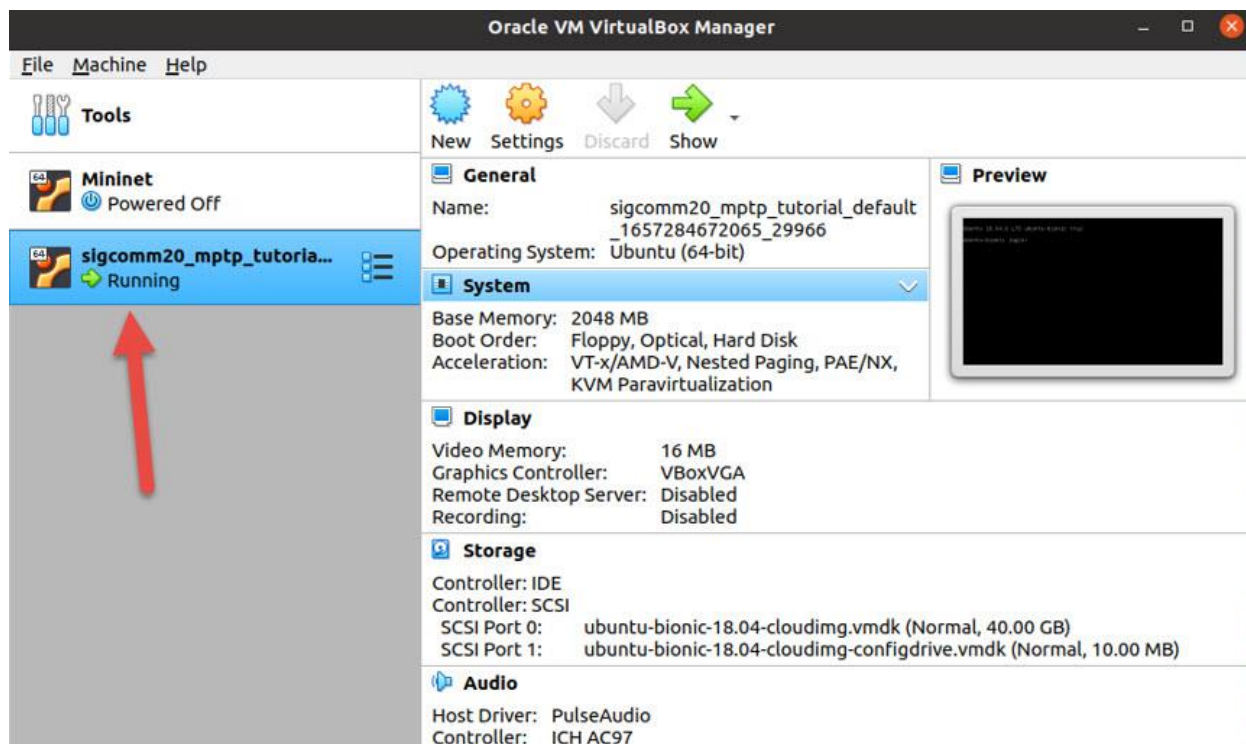
```
vagrant up
```

اگر این دستور کار نکرد می توانیم فایل اجرایی vagrant را در همان پوشه sigcomm20\_mptp\_tutorial قرار دهیم.

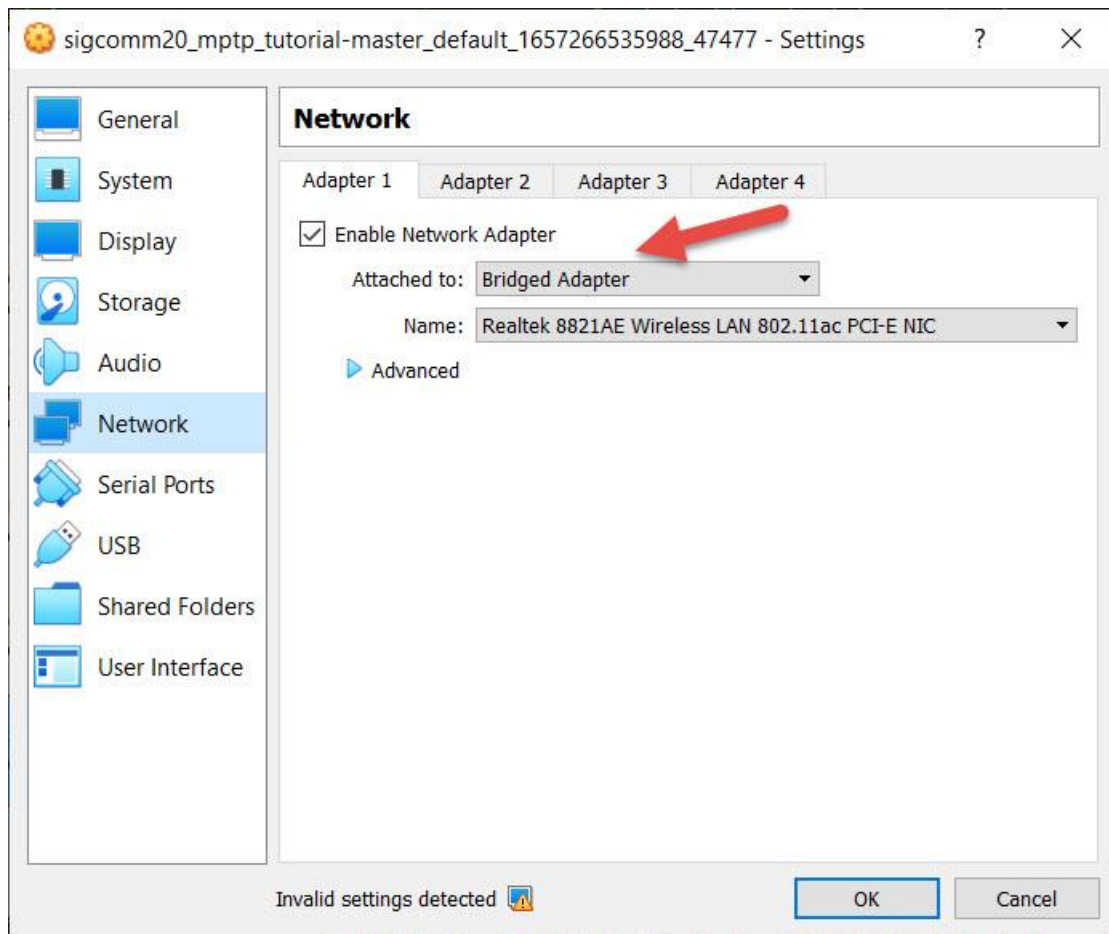
اگر سیستم عامل ویندوز را دارید قبل از دستور قبل باید دستور زیر را وارد کنید.

```
chcp 1252
```

در حالت ایده آل دستور vagrant up باید ابتدا سیستم عامل مجازی لینوکس با کرنل مجهز به MPTCP را ایجاد کرده و سپس Mininet، Minitopo، openflow و ... را ایجاد کرده و آماده به کار شود. اما در تجربه شخصی خود مشاهده کردیم که به ایراداتی بر می خورد لذا زمانی که مانند شکل زیر سیستم عامل مجازی در virtualBox ایجاد شد و دستور vagrant up تمام شد دیگر با vagrant کار نمی کنیم و از virtualBox سیستم عامل مجازی خود را کنترل می کنیم.



برای آنکه اتصال به اینترنت ماشین مجازی فعال شود باید ماشین مجازی خود را خاموش کرده و به قسمت تنظیمات آن رفته و در بخش network، آداپتور NAT را به Bridged Adaptor تغییر دهید.



سپس سیستم عامل مجازی خود را روشن می کنیم. نام کاربری و گذرواژه **vagrant** می باشد. سپس باید دستور زیر را بنویسیم.

**curl** [www.multipath-tcp.org](http://www.multipath-tcp.org)

اگر جواب **yes** را دریافت کردیم به این معنی است که ارتباط به صورت **mptcp** برقرار شده است و مشکلی در کرنل لینوکس مجهز به **mptcp** وجود نداشته است.

```
vagrant@ubuntu-bionic:~$ curl www.multipath-tcp.org
Yay, you are MPTCP-capable! You can now rest in peace.
```

اما اگر به جواب **no** برخوردیم باید یا پروسه را از اول شروع کرده یا از روش های دیگر کرنل لینوکس مجهز به **mptcp** را نصب کنیم. این روش ها در سایت [multipath-tcp.org](http://multipath-tcp.org) آمده است.

```
vagrant@ubuntu-bionic:~$ curl www.multipath-tcp.org
Nay, Nay, Nay, you have an old computer that does not speak MPTCP. Shame on you
```



می توانیم دستورات زیر را درون سیستم عامل مجازی اجرا کرده و کرنل لینوکس مجهز به mptcp را نصب کنیم.

```
wget https://github.com/multipath-tcp/mptcp/releases/download/v0.94.7/linux-headers-4.14.146.mptcp_20190924124242_amd64.deb
wget https://github.com/multipath-tcp/mptcp/releases/download/v0.94.7/linux-image-4.14.146.mptcp_20190924124242_amd64.deb
wget https://github.com/multipath-tcp/mptcp/releases/download/v0.94.7/linux-libc-dev_20190924124242_amd64.deb
wget https://github.com/multipath-tcp/mptcp/releases/download/v0.94.7/linux-mptcp-4.14_v0.94.7_20190924124242_all.deb
sudo dpkg -i linux-*.deb
# The following runs the MPTCP kernel version 4.14.146 as the default one
sudo cat /etc/default/grub | sed -e "s/GRUB_DEFAULT=0/GRUB_DEFAULT='Advanced options for Ubuntu>Ubuntu, with Linux 4.14.146.mptcp'/" > tmp_grub
sudo mv tmp_grub /etc/default/grub
sudo update-grub
# Finally ask for MPTCP module loading at the loadtime
echo "
# Load MPTCP modules
sudo modprobe mptcp_olia
sudo modprobe mptcp_coupled
sudo modprobe mptcp_balia
sudo modprobe mptcp_wvegas

# Schedulers
sudo modprobe mptcp_rr
sudo modprobe mptcp_redundant
# The following line will likely not work with versions of MPTCP < 0.95
sudo modprobe mptcp_blest

# Path managers
sudo modprobe mptcp_ndiffports
sudo modprobe mptcp_binder" | sudo tee -a /etc/bash.bashrc
```

حال باید mininet و minitopo را نصب کنیم. برای اینکار ابتدا source code ها را از git گرفته و هر کدام را نصب می کنیم.

```
git clone https://github.com/mininet/mininet
```

سپس دستورات زیر را وارد می کنیم.

```
cd mininet
git tag # list available versions
git checkout -b mininet-2.3.0 2.3.0 # or whatever version you wish to
install
```

سپس دستور زیر را وارد می کنیم تا mininet نصب شود.

```
util/install.sh -a
```

تعداد پکیج هایی که دستور فوق نصب می کند زیاد است لذا مدت زمان قابل توجهی طول می کشد تا نصب شود.

اگر هنگام نصب به مشکلی خوردید چندبار دیگر هم دستور بالا را اجرا کنید. اگر هنگام clone کردن openflow به مشکل برخوردیم باید در فایل util/install.sh تغییراتی را ایجاد کنیم. در این فایل باید همه git:// ها را به https:// تغییر دهیم (5 مورد). برای مثال مانند زیر خط اول را به خط دوم تبدیل می کنیم.

```
git clone git://github.com/mininet/openflow
git clone https://github.com/mininet/openflow
```

سپس دستور util/install.sh را اجرا می کنیم.

حال دستور زیر را می نویسیم تا مطمئن شویم که mininet درست نصب شده است.

```
sudo mn
```

سپس exit را می نویسیم تا از محیط CLI میننت خارج شویم.

حال به مسیر home/vagrant رفته و دستور زیر را اجرا می کنیم تا کد minitopo را دریافت کنیم.

```
git clone https://github.com/qdeconinck/minitopo.git
```

سپس دستور زیر را می نویسیم تا کد های این پروژه را نیز داشته باشیم.

```
git clone https://github.com/hamidreza7799/mptcp.git
```

برای تعیین حالت های مختلف الگوریتم های کنترل ازدحام و مدیریت مسیر و کنترل ازدحام در minitop لازم است که در فایل minitopo/experiments/iperf\_scenario.py از چهار دستور زیر استفاده کنیم که در حالت های مختلف بعضی از آن ها کامنت و بعضی از آن ها از کامنت خارج می شوند.

```
self.topo.command_global("modprobe mptcp_rr && sysctl -w
net.mptcp.mptcp_scheduler=roundrobin")
self.topo.command_global("modprobe mptcp_ndiffports && sysctl -w
net.mptcp.mptcp_path_manager=ndiffports ")
```

```
self.topo.command_global("echo 2 | sudo tee  
/sys/module/mptcp_ndiffports/parameters/num_subflows ")
```

برای مثال می خواهیم حالت LIA، Fullmesh، LR را اجرا کنیم. به پوشه lia\_fullmesh\_rtt می رویم و دستور های زیر را اجرا می کنیم.

```
python3 configuration.py -m 1  
python3 configuration.py
```

دستور اول این خطوط را اضافه کرده و دستور دوم با توجه به پوشه ای که در آن اجرا می شود خطوط را کامنت می کند یا از کامنت خارج می سازد. (برای پوشه های بعدی فقط دستور دوم استفاده می شود)

حال دستور زیر را وارد می کنیم تا به اجرای minitopo ایرادی نگیرد.

```
git config --global -add safe.directory /home/vagrant/minitopo
```

برای اینکه بتوانیم هر یک از حالت ها را 50 بار آزمایش کنیم باید در هر پوشه دستور زیر را اجرا کنیم.

```
./run.sh
```

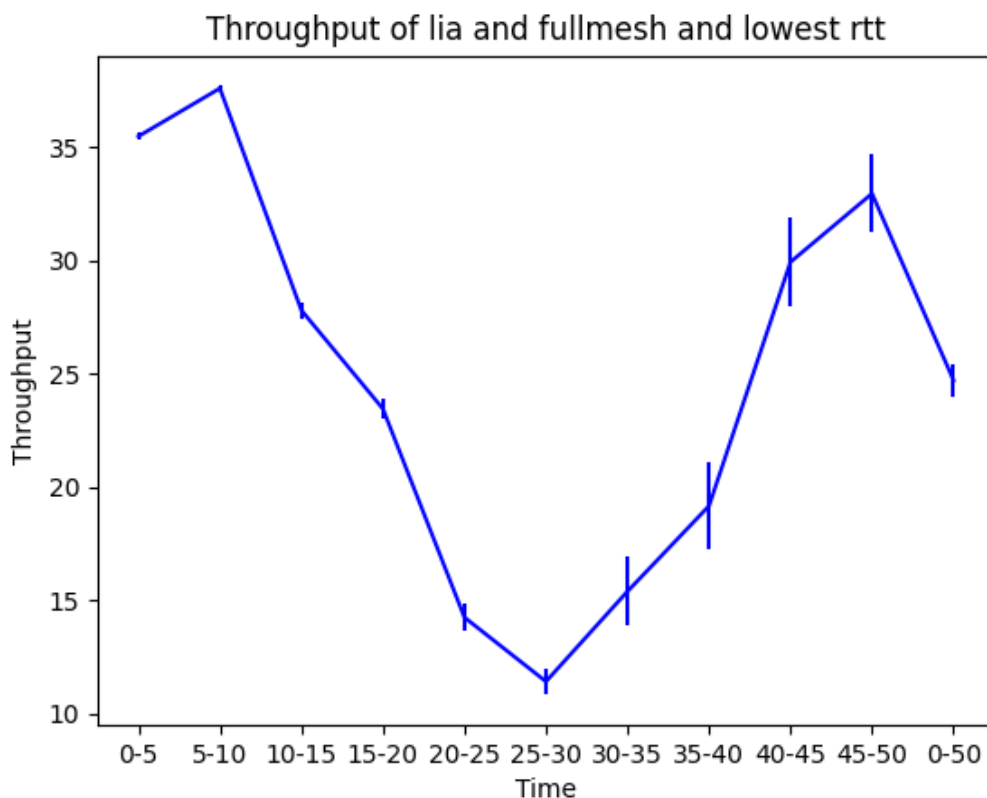
این دستور باعث می شود که iperf کلاینت mptcp در هر بار اجرا ذخیره گردد. در run.sh از minitopo برای شبیه سازی استفاده می شود که فایلی را به عنوان توپولوژی و فایل دیگری را به عنوان experiment دریافت می کند. فایل experiment نیز به iperf\_scenario.py اشاره می کند و برای تغییر در توپولوژی و سناریو باید این فایل را تغییر بدهیم.

## آزمایش

حالت 1 : کنترل ازدحام : LIA، مدیریت مسیر : Fullmesh، زمانبند : LR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

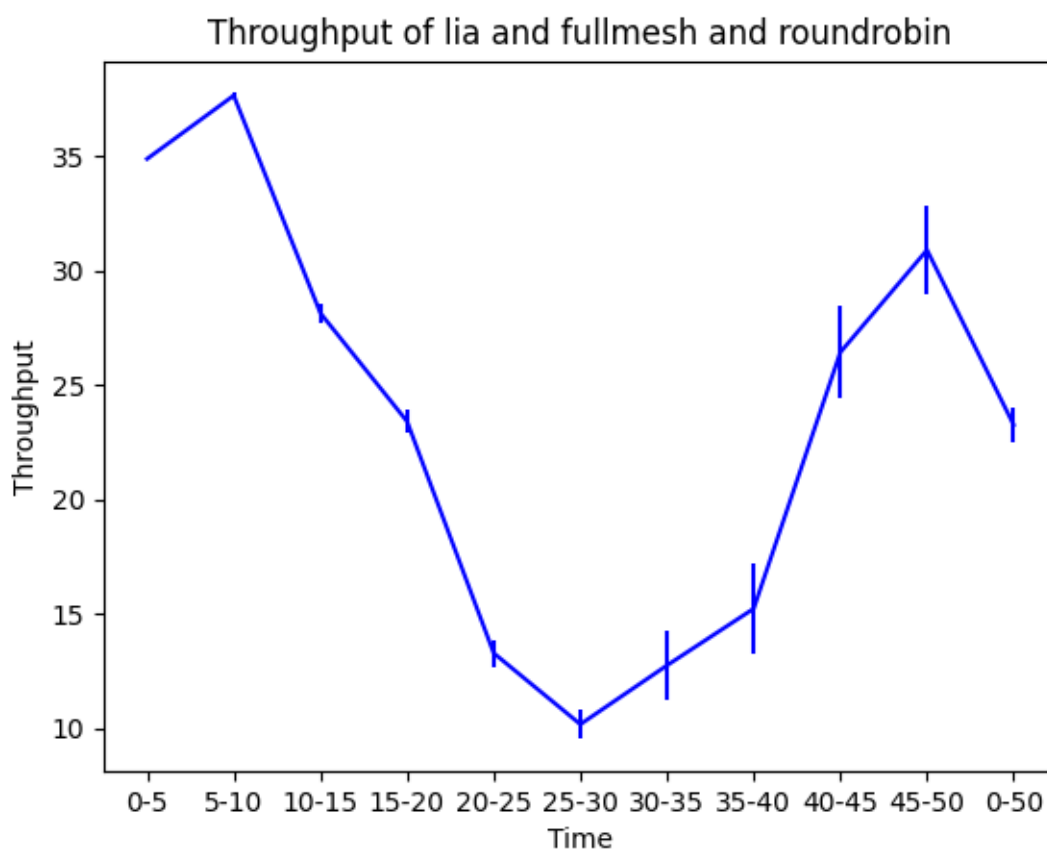
```
-----  
Client connecting to 10.1.0.1, TCP port 5001  
TCP window size: 86.2 KByte (default)  
-----  
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 3]  0.0- 5.0 sec  21.2 MBytes   35.5 Mbits/sec  
[ 3]  5.0-10.0 sec  22.4 MBytes   37.6 Mbits/sec  
[ 3] 10.0-15.0 sec  16.6 MBytes   27.8 Mbits/sec  
[ 3] 15.0-20.0 sec  14.0 MBytes   23.4 Mbits/sec  
[ 3] 20.0-25.0 sec   8.48 MBytes   14.2 Mbits/sec  
[ 3] 25.0-30.0 sec   6.8 MBytes   11.4 Mbits/sec  
[ 3] 30.0-35.0 sec   9.17 MBytes   15.4 Mbits/sec  
[ 3] 35.0-40.0 sec  11.4 MBytes   19.2 Mbits/sec  
[ 3] 40.0-45.0 sec  17.8 MBytes   29.9 Mbits/sec  
[ 3] 45.0-50.0 sec  19.6 MBytes   32.9 Mbits/sec  
[ 3]  0.0-50.0 sec  147 MBytes    24.7 Mbits/sec
```



حالت 2 : کنترل ازدحام : LIA، مدیریت مسیر : Fullmesh، زمانبند : RR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

```
-----  
Client connecting to 10.1.0.1, TCP port 5001  
TCP window size: 86.2 KByte (default)  
-----  
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 3]  0.0- 5.0 sec   20.8 MBytes   34.9 Mbits/sec  
[ 3]  5.0-10.0 sec   22.4 MBytes   37.7 Mbits/sec  
[ 3] 10.0-15.0 sec   16.8 MBytes   28.2 Mbits/sec  
[ 3] 15.0-20.0 sec   14.0 MBytes   23.4 Mbits/sec  
[ 3] 20.0-25.0 sec    7.92 MBytes   13.3 Mbits/sec  
[ 3] 25.0-30.0 sec    6.07 MBytes   10.2 Mbits/sec  
[ 3] 30.0-35.0 sec    7.6 MBytes    12.8 Mbits/sec  
[ 3] 35.0-40.0 sec    9.08 MBytes   15.2 Mbits/sec  
[ 3] 40.0-45.0 sec   15.8 MBytes   26.4 Mbits/sec  
[ 3] 45.0-50.0 sec   18.4 MBytes   30.9 Mbits/sec  
[ 3]  0.0-50.0 sec   139 MBytes    23.3 Mbits/sec
```

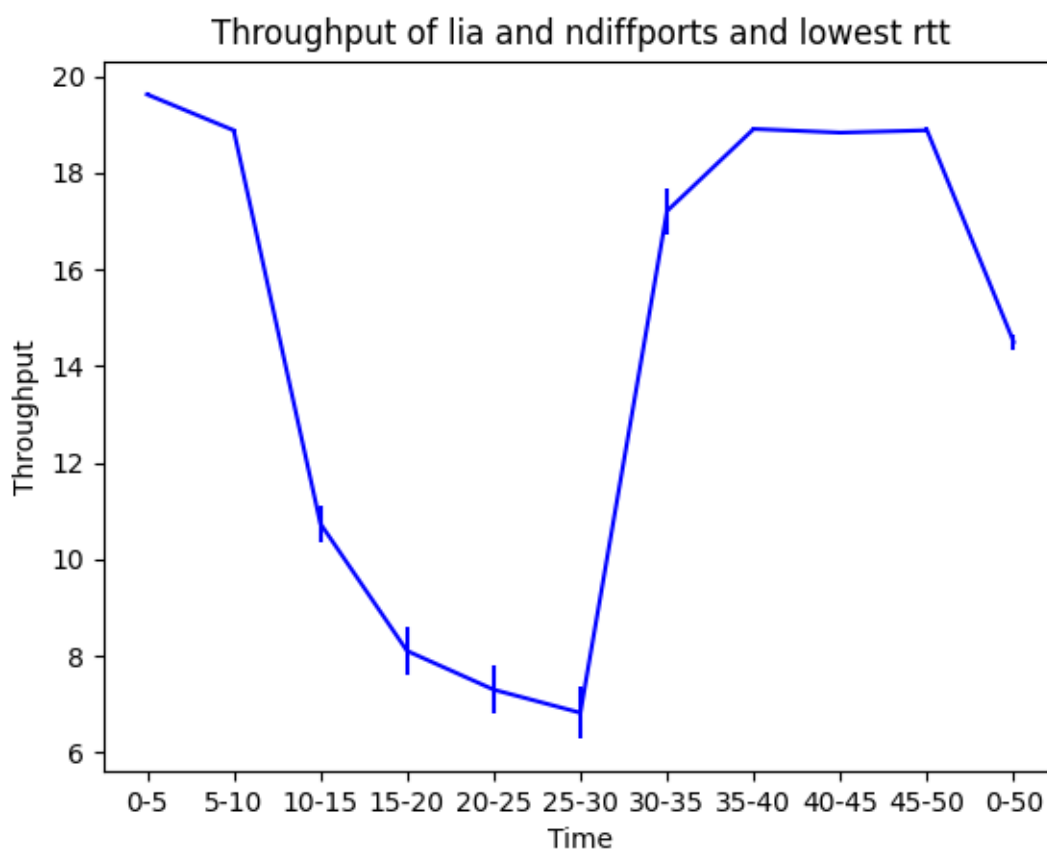


حالت 3 : کنترل ازدحام : LIA، مدیریت مسیر : ndiffports، زمانبند : LR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

```
-----
Client connecting to 10.1.0.1, TCP port 5001
TCP window size: 86.2 KByte (default)
-----

[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0- 5.0 sec  11.7 MBytes 19.6 Mbits/sec
[ 3]  5.0-10.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3] 10.0-15.0 sec   6.4 MBytes 10.7 Mbits/sec
[ 3] 15.0-20.0 sec   4.83 MBytes 8.1 Mbits/sec
[ 3] 20.0-25.0 sec   4.35 MBytes 7.3 Mbits/sec
[ 3] 25.0-30.0 sec   4.07 MBytes 6.82 Mbits/sec
[ 3] 30.0-35.0 sec  10.3 MBytes 17.2 Mbits/sec
[ 3] 35.0-40.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3] 40.0-45.0 sec  11.2 MBytes 18.8 Mbits/sec
[ 3] 45.0-50.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3]  0.0-50.0 sec  86.5 MBytes 14.5 Mbits/sec
```

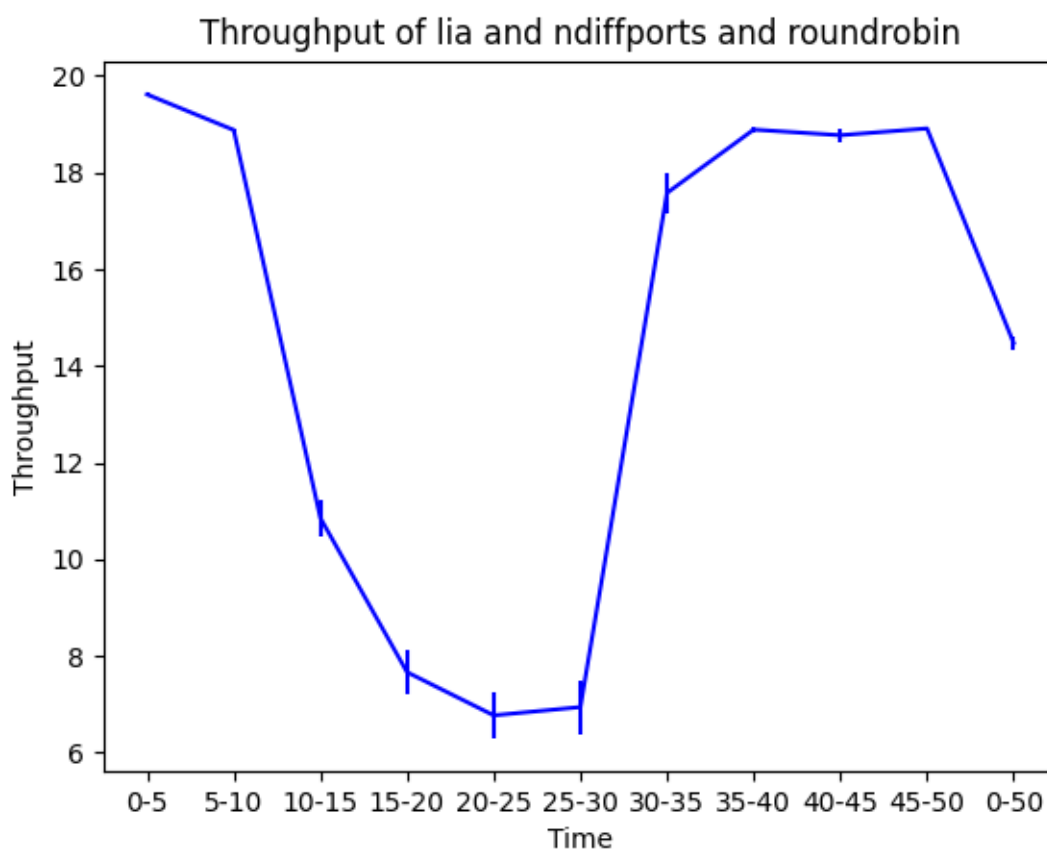


حالت 4 : کنترل ازدحام : LIA، مدیریت مسیر : ndiffports، زمانبند : RR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

```
-----
Client connecting to 10.1.0.1, TCP port 5001
TCP window size: 86.2 KByte (default)
-----

[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3]  0.0- 5.0 sec  11.7 MBytes   19.6 Mbits/sec
[ 3]  5.0-10.0 sec  11.2 MBytes   18.9 Mbits/sec
[ 3] 10.0-15.0 sec   6.46 MBytes   10.9 Mbits/sec
[ 3] 15.0-20.0 sec   4.57 MBytes    7.66 Mbits/sec
[ 3] 20.0-25.0 sec   4.04 MBytes    6.77 Mbits/sec
[ 3] 25.0-30.0 sec   4.14 MBytes    6.94 Mbits/sec
[ 3] 30.0-35.0 sec  10.5 MBytes   17.6 Mbits/sec
[ 3] 35.0-40.0 sec  11.2 MBytes   18.9 Mbits/sec
[ 3] 40.0-45.0 sec  11.2 MBytes   18.8 Mbits/sec
[ 3] 45.0-50.0 sec  11.2 MBytes   18.9 Mbits/sec
[ 3]  0.0-50.0 sec  86.3 MBytes   14.5 Mbits/sec
```





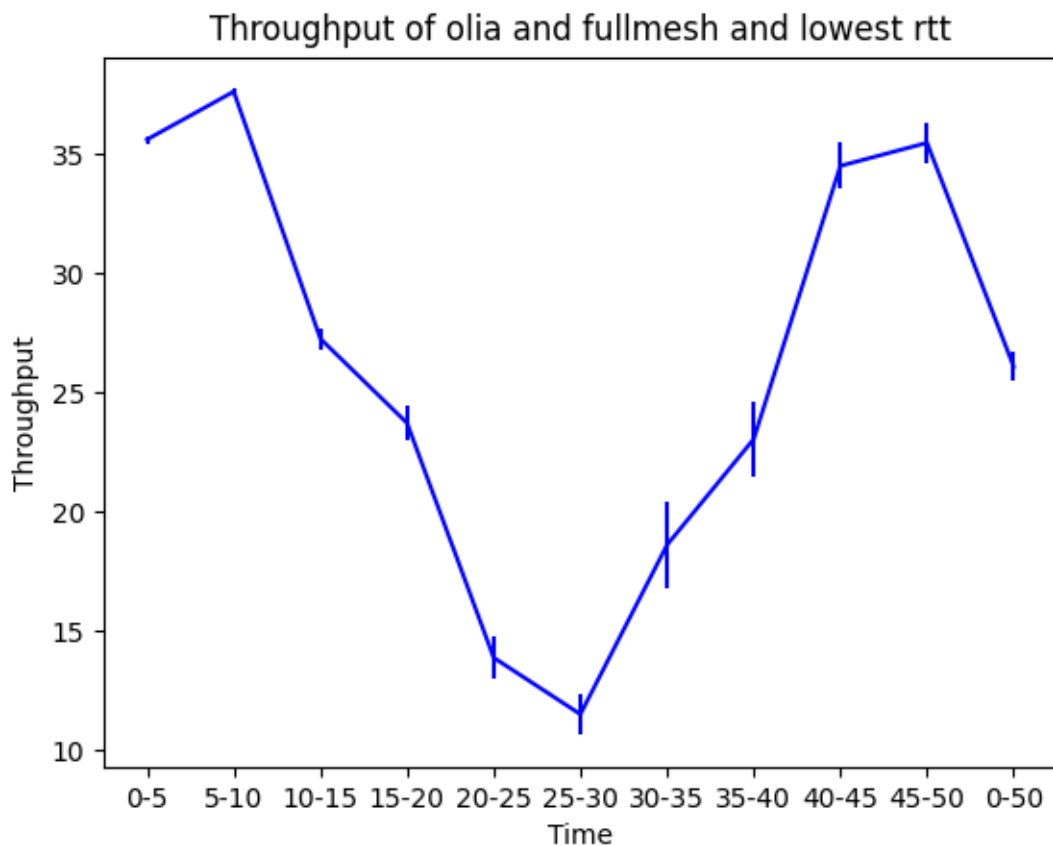
حالت 5 : کنترل ازدحام : OLIA ، مدیریت مسیر : Fullmesh ، زمانبند : LR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

Client connecting to 10.1.0.1, TCP port 5001

TCP window size: 86.2 KByte (default)

```
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3]  0.0- 5.0 sec  21.2 MBytes   35.6 Mbits/sec
[ 3]  5.0-10.0 sec  22.4 MBytes   37.6 Mbits/sec
[ 3] 10.0-15.0 sec  16.2 MBytes   27.2 Mbits/sec
[ 3] 15.0-20.0 sec  14.1 MBytes   23.7 Mbits/sec
[ 3] 20.0-25.0 sec   8.26 MBytes   13.9 Mbits/sec
[ 3] 25.0-30.0 sec   6.85 MBytes   11.5 Mbits/sec
[ 3] 30.0-35.0 sec  11.1 MBytes   18.6 Mbits/sec
[ 3] 35.0-40.0 sec  13.7 MBytes   23.0 Mbits/sec
[ 3] 40.0-45.0 sec  20.6 MBytes   34.5 Mbits/sec
[ 3] 45.0-50.0 sec  21.1 MBytes   35.5 Mbits/sec
[ 3]  0.0-50.0 sec  156 MBytes    26.1 Mbits/sec
```



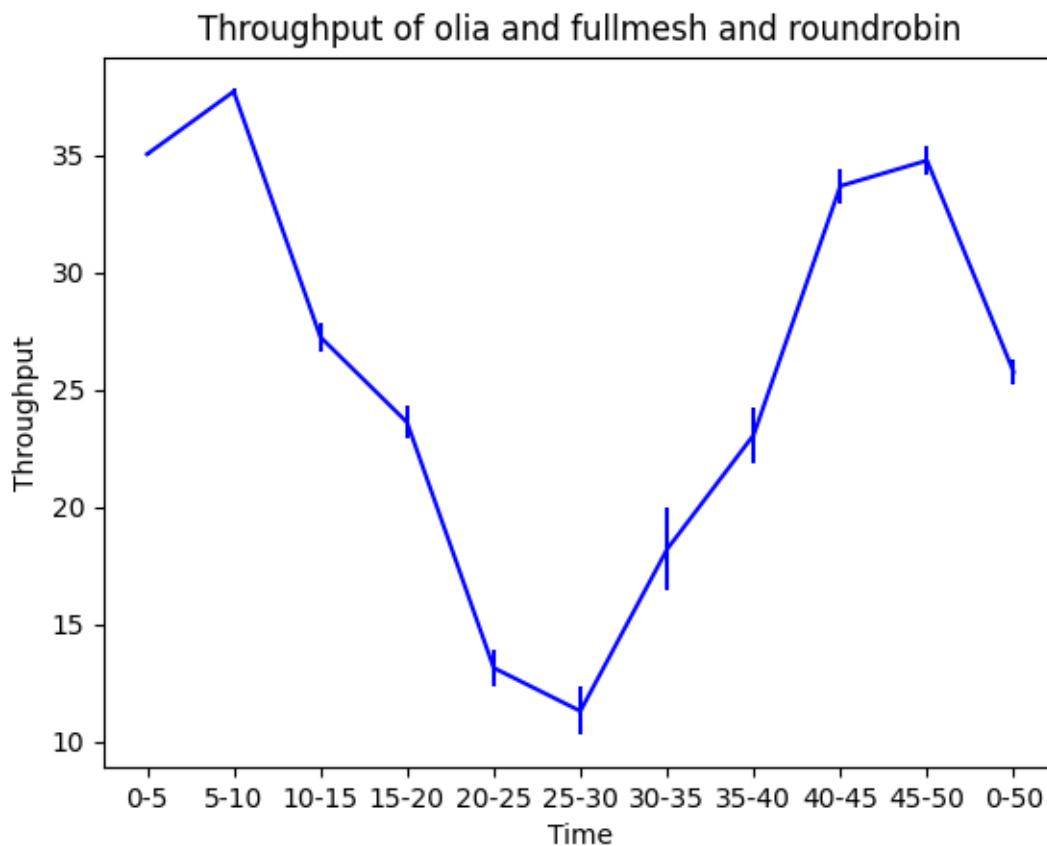


حالت 6 : کنترل ازدحام : OLIA ، مدیریت مسیر : Fullmesh ، زمانبند : RR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

```
-----|
Client connecting to 10.1.0.1, TCP port 5001
TCP window size: 86.2 KByte (default)
-----|

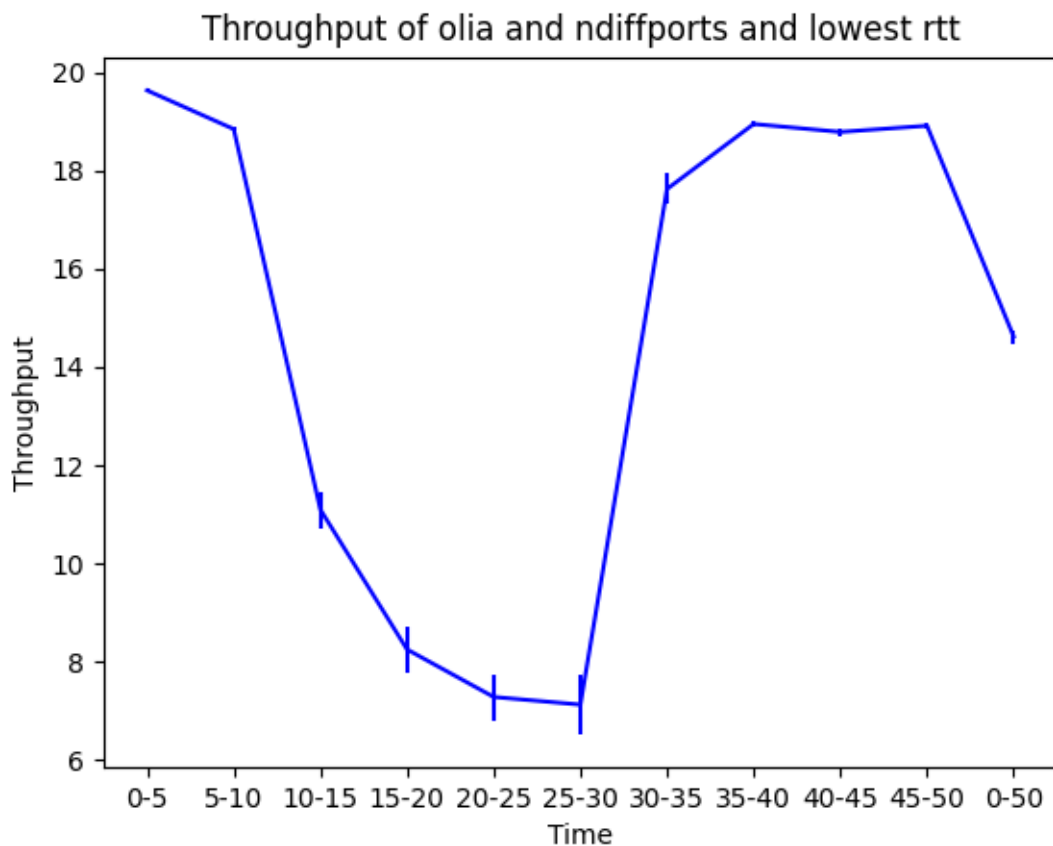
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3]  0.0- 5.0 sec   20.9 MBytes   35.0 Mbits/sec
[ 3]  5.0-10.0 sec   22.4 MBytes   37.7 Mbits/sec
[ 3] 10.0-15.0 sec   16.2 MBytes   27.2 Mbits/sec
[ 3] 15.0-20.0 sec   14.1 MBytes   23.6 Mbits/sec
[ 3] 20.0-25.0 sec    7.82 MBytes   13.1 Mbits/sec
[ 3] 25.0-30.0 sec    6.72 MBytes   11.3 Mbits/sec
[ 3] 30.0-35.0 sec   10.8 MBytes   18.2 Mbits/sec
[ 3] 35.0-40.0 sec   13.8 MBytes   23.1 Mbits/sec
[ 3] 40.0-45.0 sec   20.1 MBytes   33.7 Mbits/sec
[ 3] 45.0-50.0 sec   20.7 MBytes   34.8 Mbits/sec
[ 3]  0.0-50.0 sec   154 MBytes    25.7 Mbits/sec
```



حالت 7 : کنترل ازدحام : OLIA، مدیریت مسیر : ndiffports، زمانبند : LR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

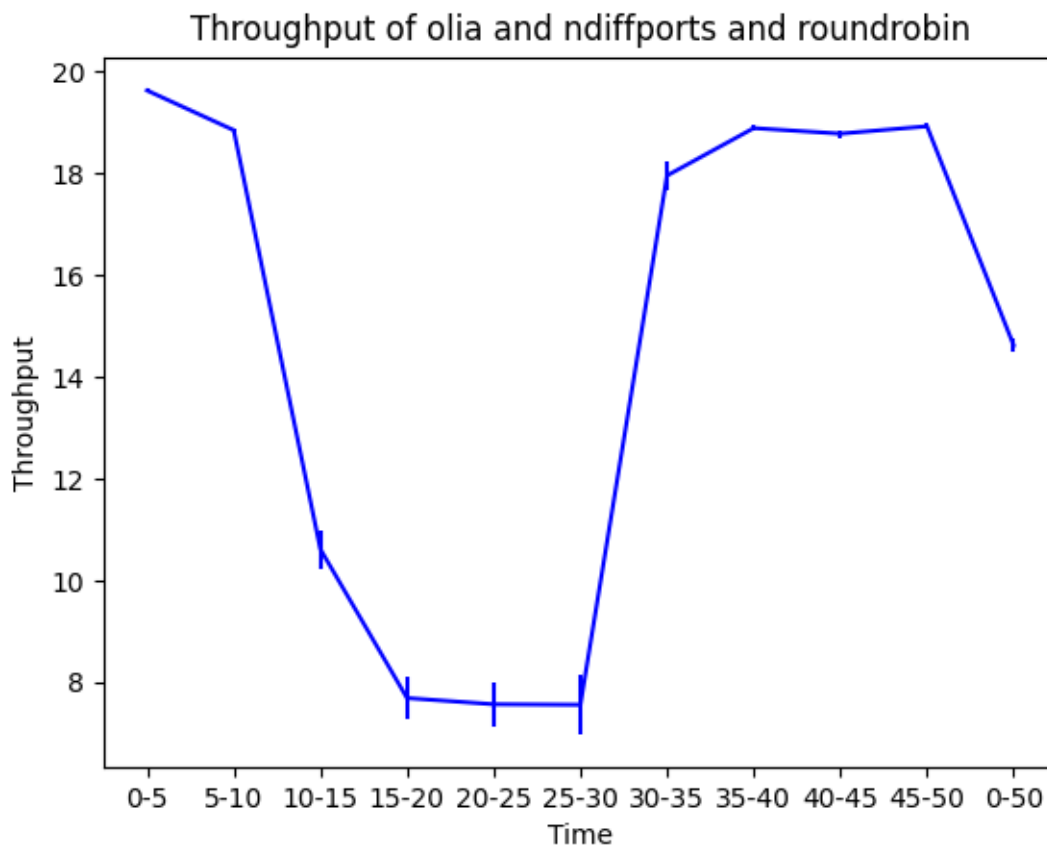
```
-----  
Client connecting to 10.1.0.1, TCP port 5001  
TCP window size: 86.2 KByte (default)  
-----  
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 3]  0.0- 5.0 sec  11.7 MBytes  19.6 Mbits/sec  
[ 3]  5.0-10.0 sec  11.2 MBytes  18.8 Mbits/sec  
[ 3] 10.0-15.0 sec   6.61 MBytes  11.1 Mbits/sec  
[ 3] 15.0-20.0 sec   4.91 MBytes   8.25 Mbits/sec  
[ 3] 20.0-25.0 sec   4.34 MBytes   7.28 Mbits/sec  
[ 3] 25.0-30.0 sec   4.25 MBytes   7.13 Mbits/sec  
[ 3] 30.0-35.0 sec  10.5 MBytes  17.6 Mbits/sec  
[ 3] 35.0-40.0 sec  11.3 MBytes  18.9 Mbits/sec  
[ 3] 40.0-45.0 sec  11.2 MBytes  18.8 Mbits/sec  
[ 3] 45.0-50.0 sec  11.2 MBytes  18.9 Mbits/sec  
[ 3]  0.0-50.0 sec  87.3 MBytes  14.6 Mbits/sec
```



حالت 8 : کنترل ازدحام : OLIA، مدیریت مسیر : ndiffports، زمانبند : RR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

```
-----  
Client connecting to 10.1.0.1, TCP port 5001  
TCP window size: 86.2 KByte (default)  
-----  
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 3]  0.0- 5.0 sec  11.7 MBytes  19.6 Mbits/sec  
[ 3]  5.0-10.0 sec  11.2 MBytes  18.8 Mbits/sec  
[ 3] 10.0-15.0 sec   6.33 MBytes  10.6 Mbits/sec  
[ 3] 15.0-20.0 sec   4.59 MBytes   7.69 Mbits/sec  
[ 3] 20.0-25.0 sec   4.51 MBytes   7.57 Mbits/sec  
[ 3] 25.0-30.0 sec   4.51 MBytes   7.56 Mbits/sec  
[ 3] 30.0-35.0 sec  10.7 MBytes  18.0 Mbits/sec  
[ 3] 35.0-40.0 sec  11.2 MBytes  18.9 Mbits/sec  
[ 3] 40.0-45.0 sec  11.1 MBytes  18.8 Mbits/sec  
[ 3] 45.0-50.0 sec  11.2 MBytes  18.9 Mbits/sec  
[ 3]  0.0-50.0 sec  87.2 MBytes  14.6 Mbits/sec
```



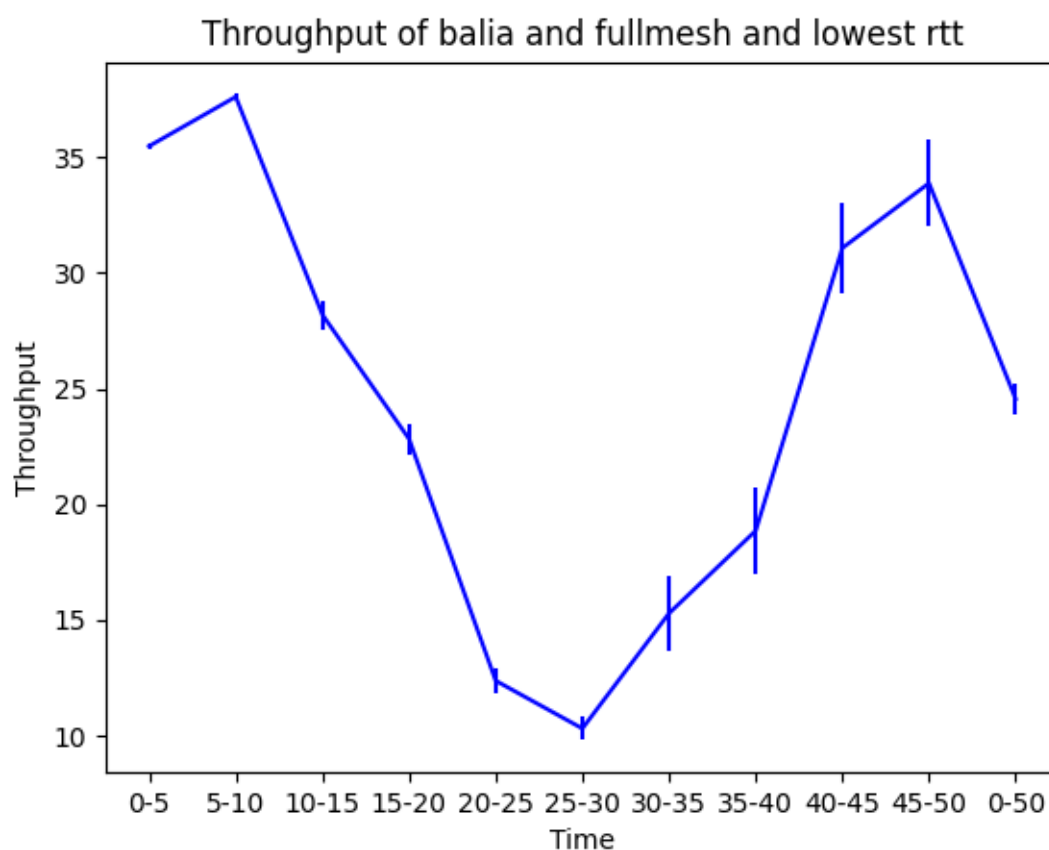
حالت 9 : کنترل ازدحام : BALIA ، مدیریت مسیر : Fullmesh ، زمانبند : LR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

Client connecting to 10.1.0.1, TCP port 5001

TCP window size: 86.2 KByte (default)

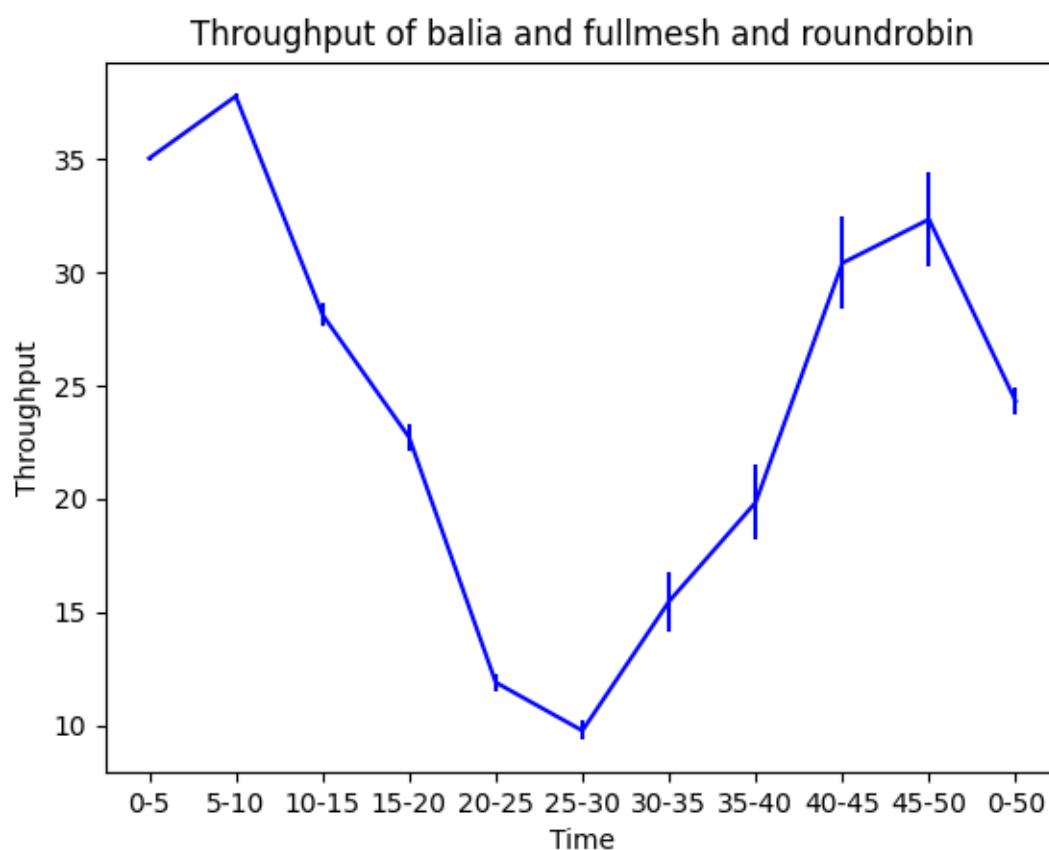
```
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3]  0.0- 5.0 sec  21.2 MBytes   35.5 Mbits/sec
[ 3]  5.0-10.0 sec  22.4 MBytes   37.6 Mbits/sec
[ 3] 10.0-15.0 sec  16.8 MBytes   28.2 Mbits/sec
[ 3] 15.0-20.0 sec  13.6 MBytes   22.8 Mbits/sec
[ 3] 20.0-25.0 sec   7.38 MBytes   12.4 Mbits/sec
[ 3] 25.0-30.0 sec   6.14 MBytes   10.3 Mbits/sec
[ 3] 30.0-35.0 sec   9.11 MBytes   15.3 Mbits/sec
[ 3] 35.0-40.0 sec  11.2 MBytes   18.9 Mbits/sec
[ 3] 40.0-45.0 sec  18.5 MBytes   31.1 Mbits/sec
[ 3] 45.0-50.0 sec  20.2 MBytes   33.9 Mbits/sec
[ 3]  0.0-50.0 sec  146 MBytes    24.6 Mbits/sec
```



حالت 10 : کنترل ازدحام : BALIA، مدیریت مسیر : Fullmesh، زمانبند : RR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

```
-----
Client connecting to 10.1.0.1, TCP port 5001
TCP window size: 86.2 KByte (default)
-----
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0- 5.0 sec   20.9 MBytes 35.0 Mbits/sec
[ 3]  5.0-10.0 sec   22.5 MBytes 37.8 Mbits/sec
[ 3] 10.0-15.0 sec   16.8 MBytes 28.1 Mbits/sec
[ 3] 15.0-20.0 sec   13.6 MBytes 22.7 Mbits/sec
[ 3] 20.0-25.0 sec    7.09 MBytes 11.9 Mbits/sec
[ 3] 25.0-30.0 sec    5.82 MBytes  9.77 Mbits/sec
[ 3] 30.0-35.0 sec    9.22 MBytes 15.5 Mbits/sec
[ 3] 35.0-40.0 sec   11.8 MBytes 19.8 Mbits/sec
[ 3] 40.0-45.0 sec   18.1 MBytes 30.4 Mbits/sec
[ 3] 45.0-50.0 sec   19.3 MBytes 32.3 Mbits/sec
[ 3]  0.0-50.0 sec   145 MBytes 24.3 Mbits/sec
```

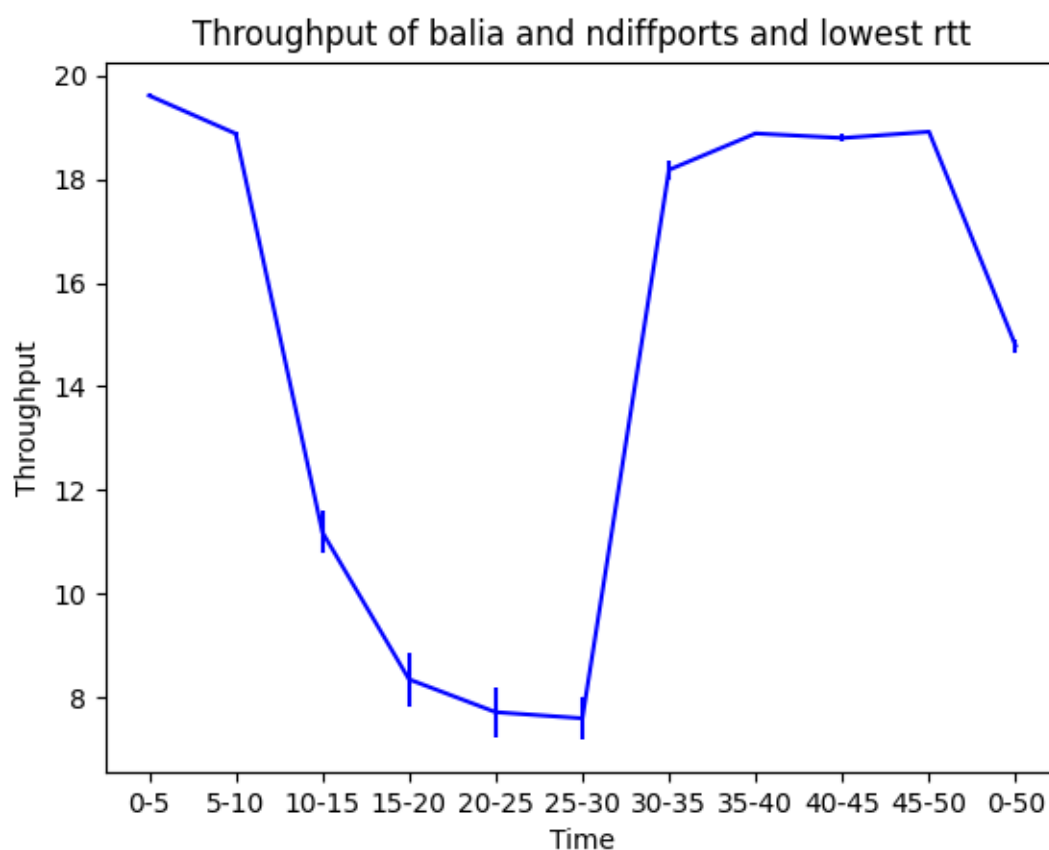




حالت 11 : کنترل ازدحام : BALIA، مدیریت مسیر : ndiffports، زمانبند : LR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

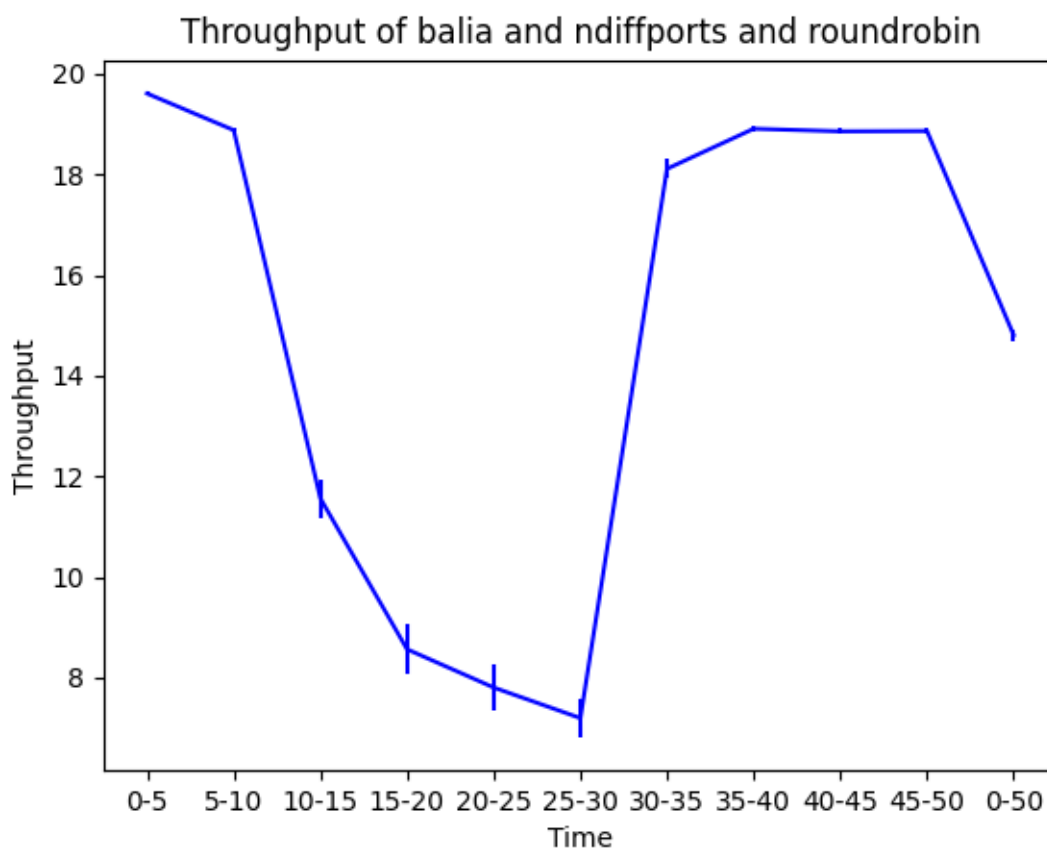
```
-----
Client connecting to 10.1.0.1, TCP port 5001
TCP window size: 86.2 KByte (default)
-----
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0- 5.0 sec  11.7 MBytes 19.6 Mbits/sec
[ 3]  5.0-10.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3] 10.0-15.0 sec   6.67 MBytes 11.2 Mbits/sec
[ 3] 15.0-20.0 sec   4.97 MBytes  8.34 Mbits/sec
[ 3] 20.0-25.0 sec   4.59 MBytes  7.71 Mbits/sec
[ 3] 25.0-30.0 sec   4.53 MBytes  7.59 Mbits/sec
[ 3] 30.0-35.0 sec  10.8 MBytes 18.2 Mbits/sec
[ 3] 35.0-40.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3] 40.0-45.0 sec  11.2 MBytes 18.8 Mbits/sec
[ 3] 45.0-50.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3]  0.0-50.0 sec  88.2 MBytes 14.8 Mbits/sec
```



حالت 12 : کنترل ازدحام : BALIA، مدیریت مسیر : ndiffports، زمانبند : RR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

```
-----
Client connecting to 10.1.0.1, TCP port 5001
TCP window size: 86.2 KByte (default)
-----
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0- 5.0 sec  11.7 MBytes 19.6 Mbits/sec
[ 3]  5.0-10.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3] 10.0-15.0 sec   6.89 MBytes 11.6 Mbits/sec
[ 3] 15.0-20.0 sec   5.1 MBytes  8.56 Mbits/sec
[ 3] 20.0-25.0 sec   4.65 MBytes  7.8 Mbits/sec
[ 3] 25.0-30.0 sec   4.29 MBytes  7.2 Mbits/sec
[ 3] 30.0-35.0 sec  10.8 MBytes 18.1 Mbits/sec
[ 3] 35.0-40.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3] 40.0-45.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3] 45.0-50.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3]  0.0-50.0 sec  88.4 MBytes 14.8 Mbits/sec
```

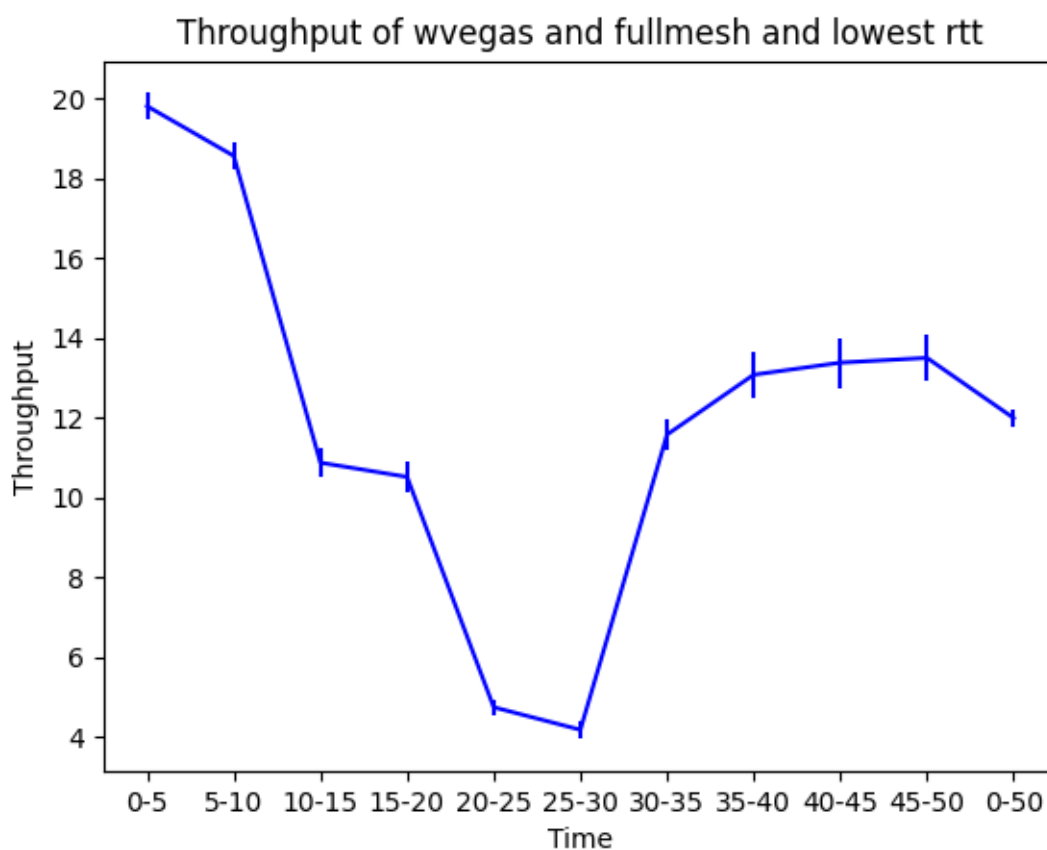


حالت 13 : کنترل ازدحام : WVEGAS، مدیریت مسیر : Fullmesh، زمانبند : LR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

```
-----
Client connecting to 10.1.0.1, TCP port 5001
TCP window size: 86.2 KByte (default)
-----

[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0- 5.0 sec   11.8 MBytes 19.8 Mbits/sec
[ 3]  5.0-10.0 sec   11.1 MBytes 18.6 Mbits/sec
[ 3] 10.0-15.0 sec    6.49 MBytes 10.9 Mbits/sec
[ 3] 15.0-20.0 sec    6.28 MBytes 10.5 Mbits/sec
[ 3] 20.0-25.0 sec    2.83 MBytes  4.75 Mbits/sec
[ 3] 25.0-30.0 sec    2.49 MBytes  4.18 Mbits/sec
[ 3] 30.0-35.0 sec     6.9 MBytes 11.6 Mbits/sec
[ 3] 35.0-40.0 sec    7.79 MBytes 13.1 Mbits/sec
[ 3] 40.0-45.0 sec    7.97 MBytes 13.4 Mbits/sec
[ 3] 45.0-50.0 sec    8.04 MBytes 13.5 Mbits/sec
[ 3]  0.0-50.0 sec   71.7 MBytes 12.0 Mbits/sec
```

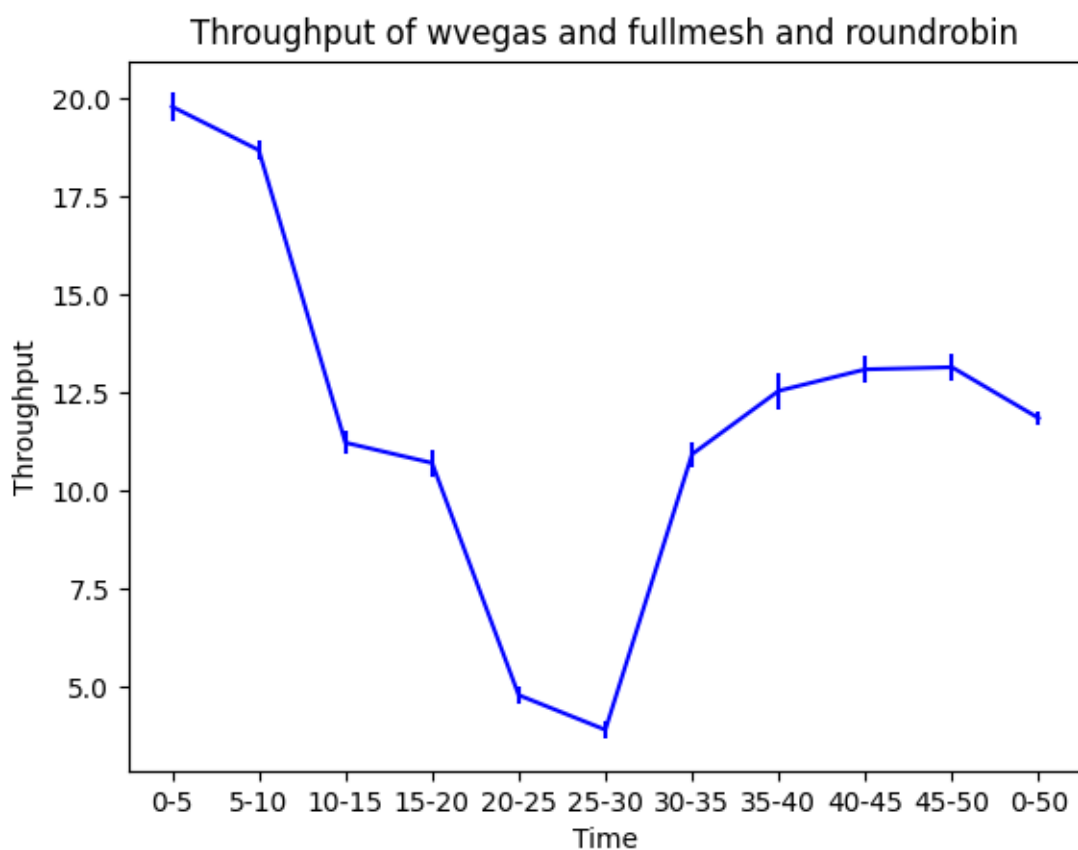




حالت 14 : کنترل ازدحام : WVEGAS، مدیریت مسیر : Fullmesh، زمانبند : RR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

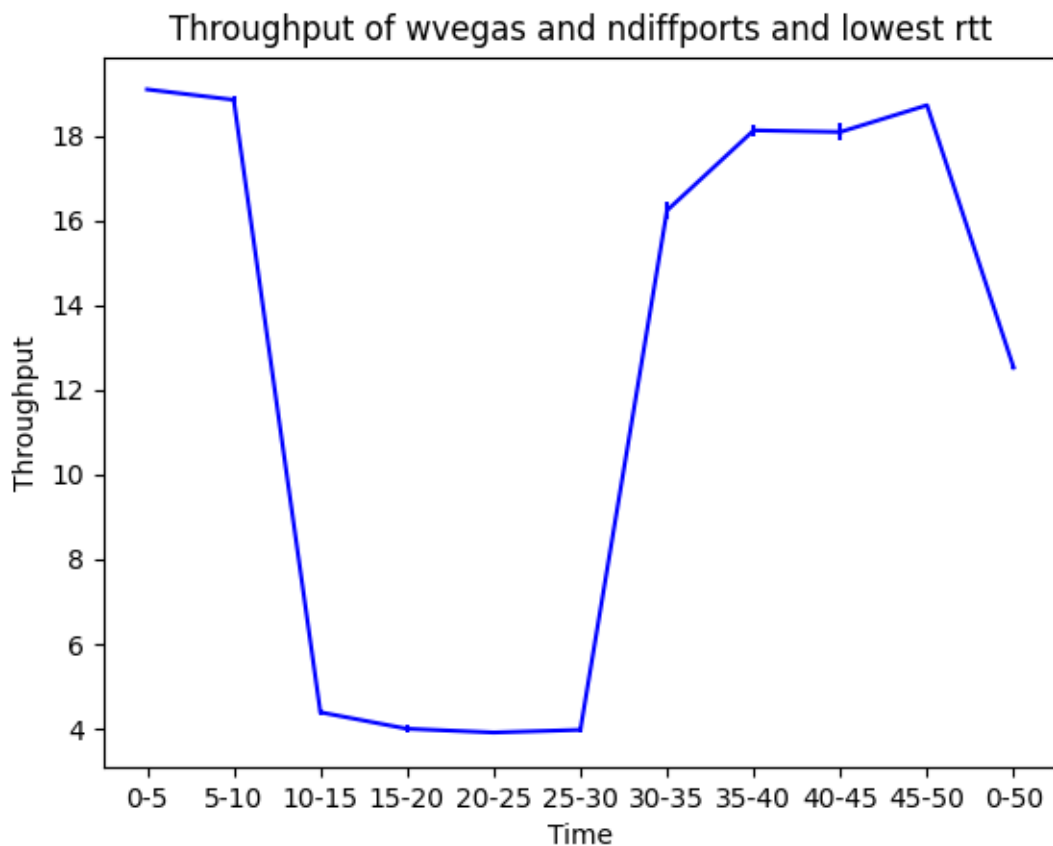
```
-----  
Client connecting to 10.1.0.1, TCP port 5001  
TCP window size: 86.2 KByte (default)  
-----  
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 3]  0.0- 5.0 sec   11.8 MBytes   19.8 Mbits/sec  
[ 3]  5.0-10.0 sec   11.1 MBytes   18.7 Mbits/sec  
[ 3] 10.0-15.0 sec    6.7 MBytes   11.2 Mbits/sec  
[ 3] 15.0-20.0 sec    6.39 MBytes  10.7 Mbits/sec  
[ 3] 20.0-25.0 sec    2.86 MBytes   4.8 Mbits/sec  
[ 3] 25.0-30.0 sec    2.34 MBytes   3.92 Mbits/sec  
[ 3] 30.0-35.0 sec    6.52 MBytes   10.9 Mbits/sec  
[ 3] 35.0-40.0 sec    7.48 MBytes   12.6 Mbits/sec  
[ 3] 40.0-45.0 sec    7.8 MBytes    13.1 Mbits/sec  
[ 3] 45.0-50.0 sec    7.84 MBytes   13.2 Mbits/sec  
[ 3]  0.0-50.0 sec   70.9 MBytes   11.9 Mbits/sec
```



حالت 15 : کنترل ازدحام : WVEGAS، مدیریت مسیر : ndiffports، زمانبند : LR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

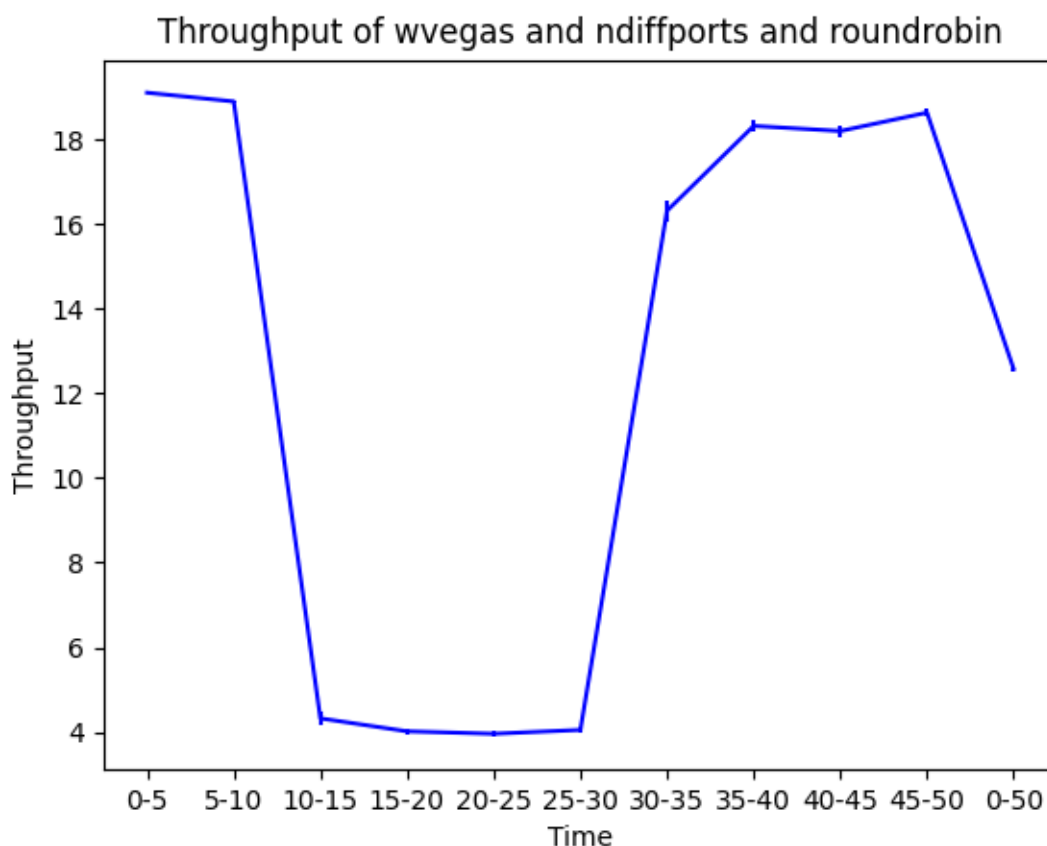
```
-----  
Client connecting to 10.1.0.1, TCP port 5001  
TCP window size: 86.2 KByte (default)  
-----  
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 3]  0.0- 5.0 sec   11.4 MBytes   19.1 Mbits/sec  
[ 3]  5.0-10.0 sec   11.2 MBytes   18.9 Mbits/sec  
[ 3] 10.0-15.0 sec    2.62 MBytes    4.39 Mbits/sec  
[ 3] 15.0-20.0 sec    2.39 MBytes    4.0 Mbits/sec  
[ 3] 20.0-25.0 sec    2.33 MBytes    3.91 Mbits/sec  
[ 3] 25.0-30.0 sec    2.37 MBytes    3.97 Mbits/sec  
[ 3] 30.0-35.0 sec    9.68 MBytes   16.2 Mbits/sec  
[ 3] 35.0-40.0 sec   10.8 MBytes   18.1 Mbits/sec  
[ 3] 40.0-45.0 sec   10.8 MBytes   18.1 Mbits/sec  
[ 3] 45.0-50.0 sec   11.1 MBytes   18.7 Mbits/sec  
[ 3]  0.0-50.0 sec   74.7 MBytes   12.5 Mbits/sec
```



حالت 16 : کنترل ازدحام : WVEGAS، مدیریت مسیر : ndiffports، زمانبند : RR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

```
-----
Client connecting to 10.1.0.1, TCP port 5001
TCP window size: 86.2 KByte (default)
-----
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0- 5.0 sec   11.4 MBytes 19.1 Mbits/sec
[ 3]  5.0-10.0 sec   11.2 MBytes 18.9 Mbits/sec
[ 3] 10.0-15.0 sec    2.58 MBytes  4.32 Mbits/sec
[ 3] 15.0-20.0 sec    2.4 MBytes  4.01 Mbits/sec
[ 3] 20.0-25.0 sec    2.36 MBytes  3.96 Mbits/sec
[ 3] 25.0-30.0 sec    2.42 MBytes  4.05 Mbits/sec
[ 3] 30.0-35.0 sec    9.72 MBytes 16.3 Mbits/sec
[ 3] 35.0-40.0 sec   10.9 MBytes 18.3 Mbits/sec
[ 3] 40.0-45.0 sec   10.9 MBytes 18.2 Mbits/sec
[ 3] 45.0-50.0 sec   11.1 MBytes 18.6 Mbits/sec
[ 3]  0.0-50.0 sec   75.0 MBytes 12.6 Mbits/sec
```



حالت 17 : کنترل ازدحام : RENO، مدیریت مسیر : Fullmesh، زمانبند : LR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

Client connecting to 10.1.0.1, TCP port 5001

TCP window size: 86.2 KByte (default)

[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001

[ ID]	Interval	Transfer	Bandwidth
-------	----------	----------	-----------

[ 3]	0.0- 5.0 sec	21.2 MBytes	35.5 Mbits/sec
------	--------------	-------------	----------------

[ 3]	5.0-10.0 sec	22.4 MBytes	37.5 Mbits/sec
------	--------------	-------------	----------------

[ 3]	10.0-15.0 sec	18.2 MBytes	30.5 Mbits/sec
------	---------------	-------------	----------------

[ 3]	15.0-20.0 sec	17.0 MBytes	28.6 Mbits/sec
------	---------------	-------------	----------------

[ 3]	20.0-25.0 sec	13.9 MBytes	23.4 Mbits/sec
------	---------------	-------------	----------------

[ 3]	25.0-30.0 sec	12.8 MBytes	21.5 Mbits/sec
------	---------------	-------------	----------------

[ 3]	30.0-35.0 sec	17.1 MBytes	28.7 Mbits/sec
------	---------------	-------------	----------------

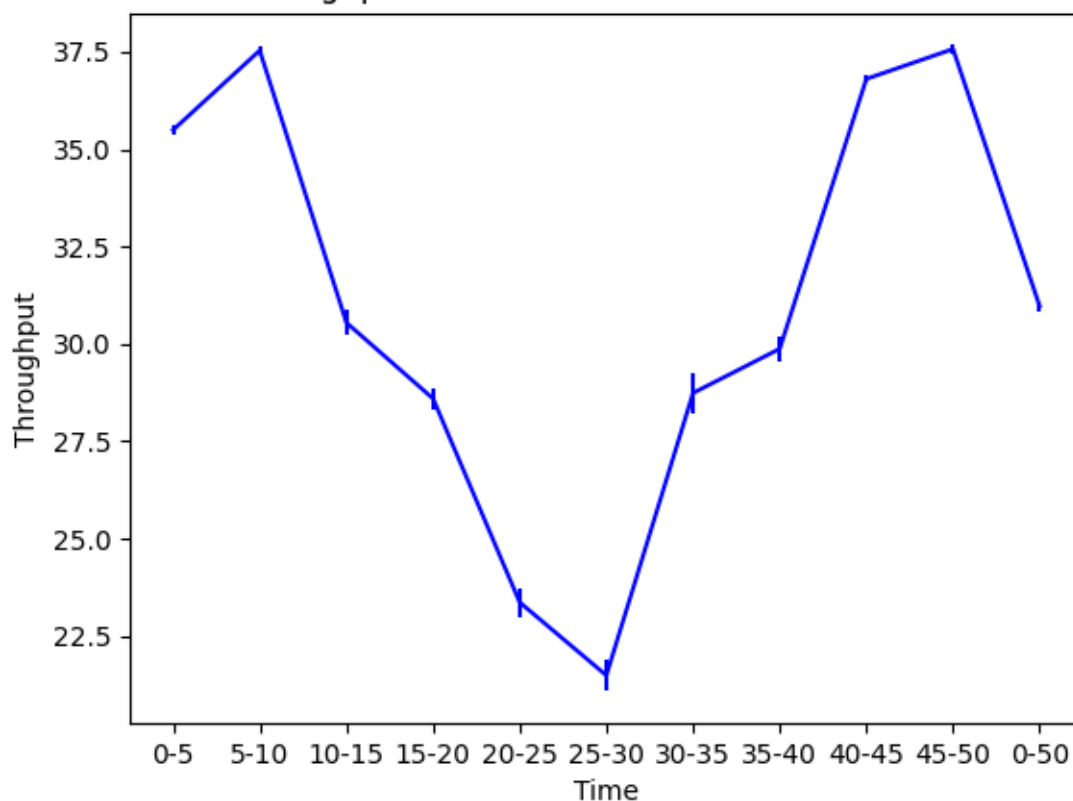
[ 3]	35.0-40.0 sec	17.8 MBytes	29.9 Mbits/sec
------	---------------	-------------	----------------

[ 3]	40.0-45.0 sec	21.9 MBytes	36.8 Mbits/sec
------	---------------	-------------	----------------

[ 3]	45.0-50.0 sec	22.4 MBytes	37.6 Mbits/sec
------	---------------	-------------	----------------

[ 3]	0.0-50.0 sec	185 MBytes	31.0 Mbits/sec
------	--------------	------------	----------------

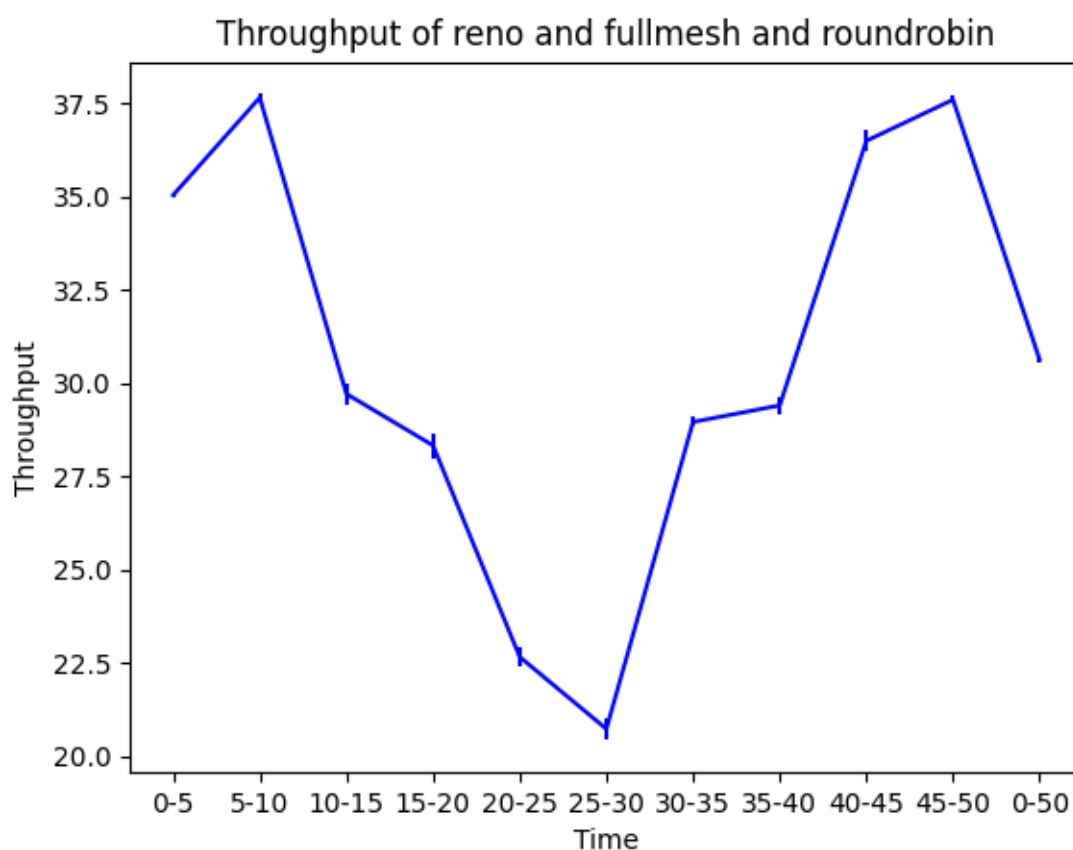
Throughput of reno and fullmesh and lowest rtt



حالت 18 : کنترل ازدحام : RENO، مدیریت مسیر : Fullmesh، زمانبند : RR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

```
-----
Client connecting to 10.1.0.1, TCP port 5001
TCP window size: 86.2 KByte (default)
-----
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0- 5.0 sec  20.9 MBytes 35.0 Mbits/sec
[ 3]  5.0-10.0 sec  22.4 MBytes 37.6 Mbits/sec
[ 3] 10.0-15.0 sec  17.7 MBytes 29.7 Mbits/sec
[ 3] 15.0-20.0 sec  16.9 MBytes 28.3 Mbits/sec
[ 3] 20.0-25.0 sec  13.5 MBytes 22.7 Mbits/sec
[ 3] 25.0-30.0 sec  12.3 MBytes 20.7 Mbits/sec
[ 3] 30.0-35.0 sec  17.2 MBytes 28.9 Mbits/sec
[ 3] 35.0-40.0 sec  17.5 MBytes 29.4 Mbits/sec
[ 3] 40.0-45.0 sec  21.8 MBytes 36.5 Mbits/sec
[ 3] 45.0-50.0 sec  22.4 MBytes 37.6 Mbits/sec
[ 3]  0.0-50.0 sec  183 MBytes 30.6 Mbits/sec
```

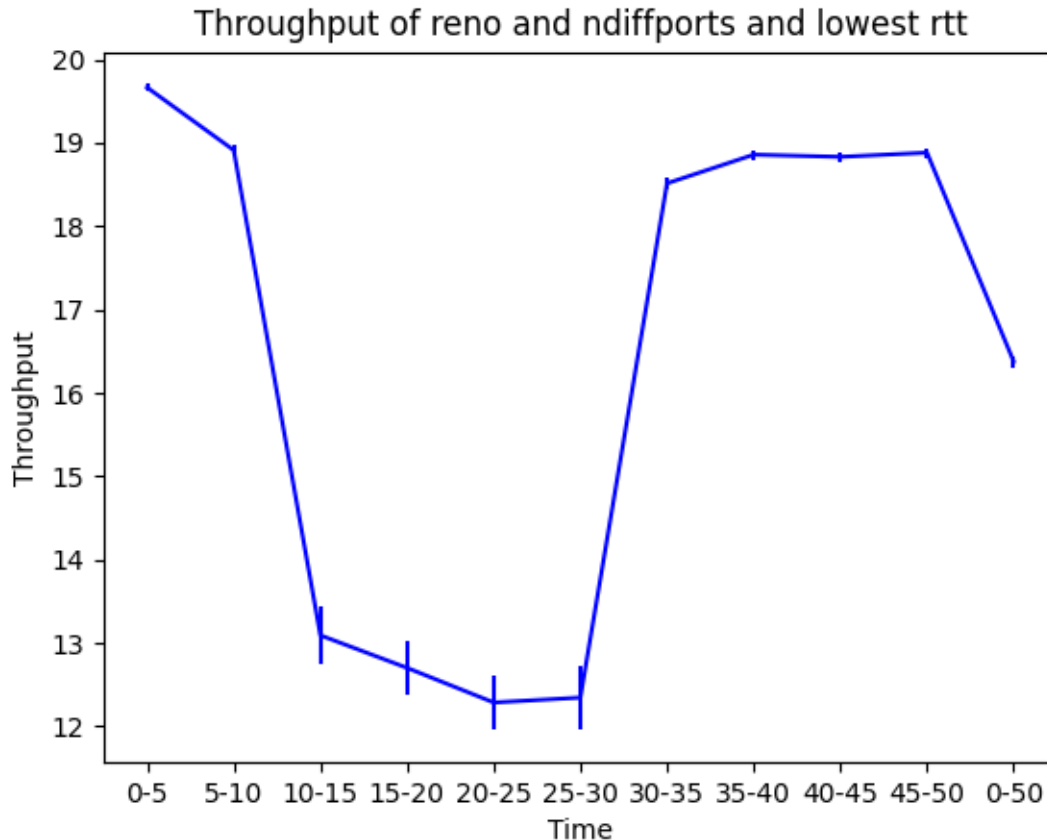




حالت 19 : کنترل ازدحام : RENO، مدیریت مسیر : ndiffports، زمانبند : LR

اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

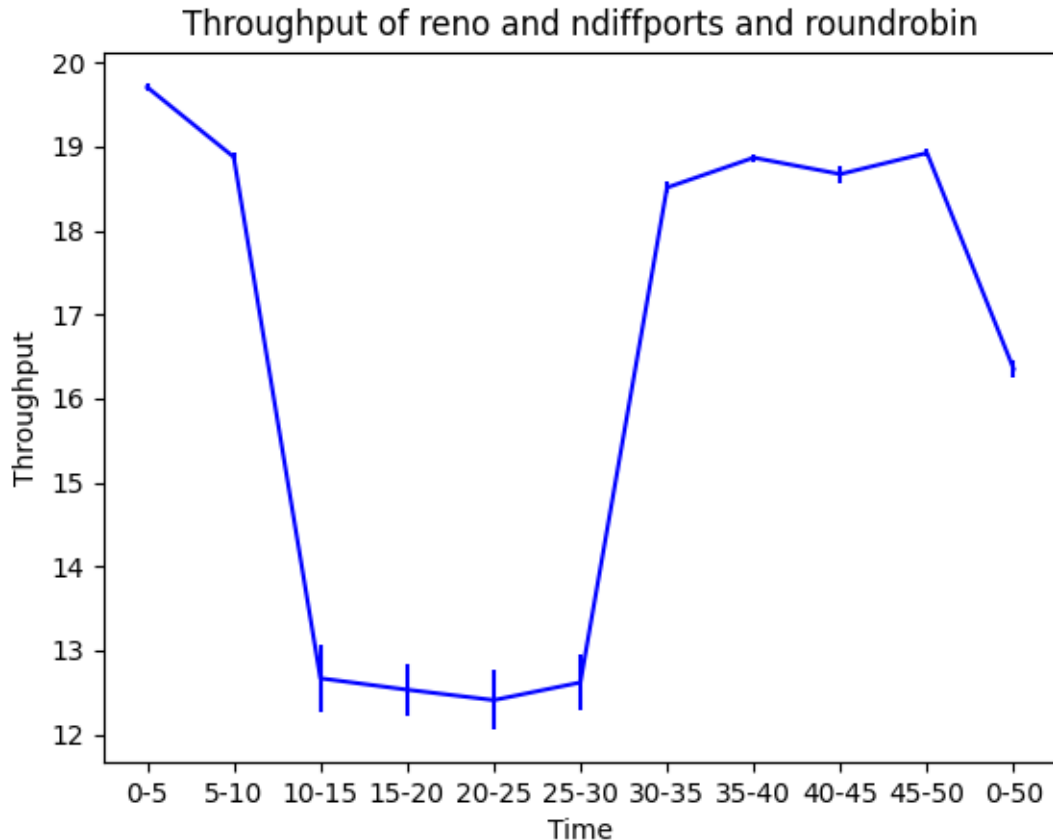
```
-----
Client connecting to 10.1.0.1, TCP port 5001
TCP window size: 86.2 KByte (default)
-----
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0- 5.0 sec  11.8 MBytes 19.7 Mbits/sec
[ 3]  5.0-10.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3] 10.0-15.0 sec   7.8 MBytes 13.1 Mbits/sec
[ 3] 15.0-20.0 sec   7.57 MBytes 12.7 Mbits/sec
[ 3] 20.0-25.0 sec   7.32 MBytes 12.3 Mbits/sec
[ 3] 25.0-30.0 sec   7.36 MBytes 12.3 Mbits/sec
[ 3] 30.0-35.0 sec  11.0 MBytes 18.5 Mbits/sec
[ 3] 35.0-40.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3] 40.0-45.0 sec  11.2 MBytes 18.8 Mbits/sec
[ 3] 45.0-50.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3]  0.0-50.0 sec  97.7 MBytes 16.4 Mbits/sec
```



حالت 20 : کنترل ازدحام : RENO، مدیریت مسیر : ndiffports، زمانبند : RR

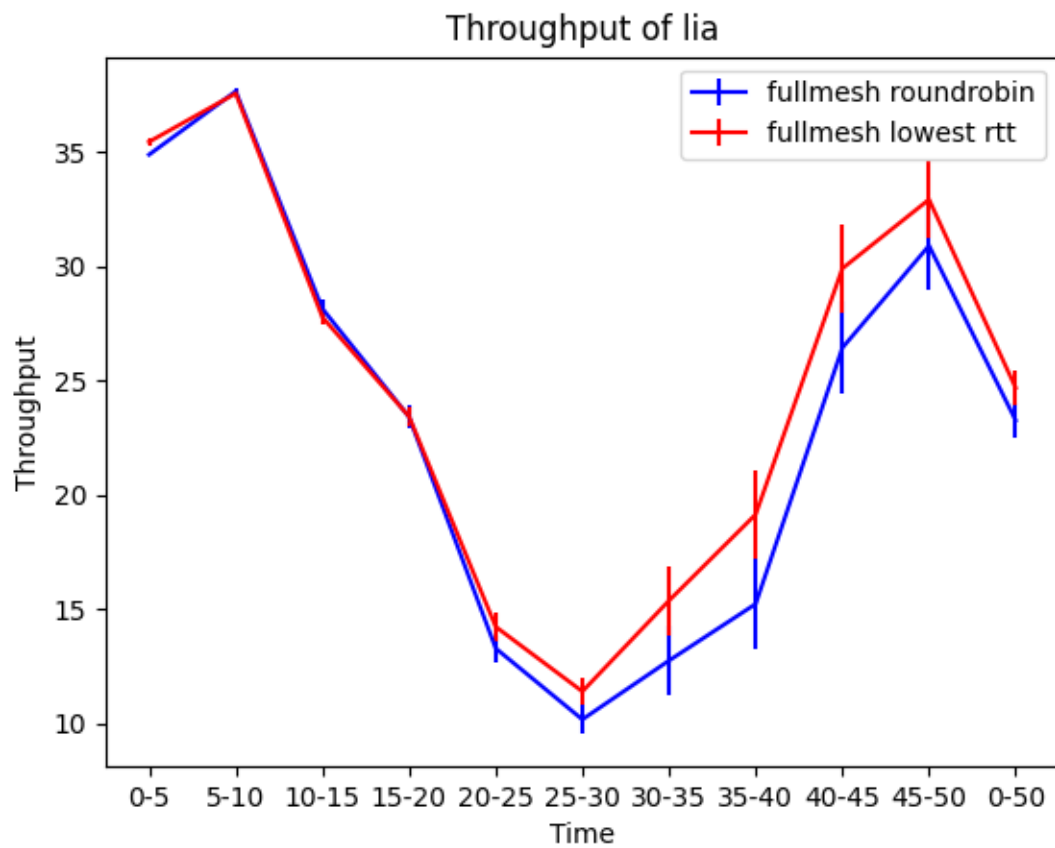
اعداد تصویر زیر میانگین 50 بار تکرار آزمایش را نشان می دهند

```
-----
Client connecting to 10.1.0.1, TCP port 5001
TCP window size: 86.2 KByte (default)
-----
[ 3] local 10.0.0.1 port 54996 connected with 10.1.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0- 5.0 sec  11.8 MBytes 19.7 Mbits/sec
[ 3]  5.0-10.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3] 10.0-15.0 sec   7.55 MBytes 12.7 Mbits/sec
[ 3] 15.0-20.0 sec   7.47 MBytes 12.5 Mbits/sec
[ 3] 20.0-25.0 sec   7.4 MBytes 12.4 Mbits/sec
[ 3] 25.0-30.0 sec   7.52 MBytes 12.6 Mbits/sec
[ 3] 30.0-35.0 sec  11.0 MBytes 18.5 Mbits/sec
[ 3] 35.0-40.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3] 40.0-45.0 sec  11.1 MBytes 18.7 Mbits/sec
[ 3] 45.0-50.0 sec  11.2 MBytes 18.9 Mbits/sec
[ 3]  0.0-50.0 sec  97.5 MBytes 16.4 Mbits/sec
```



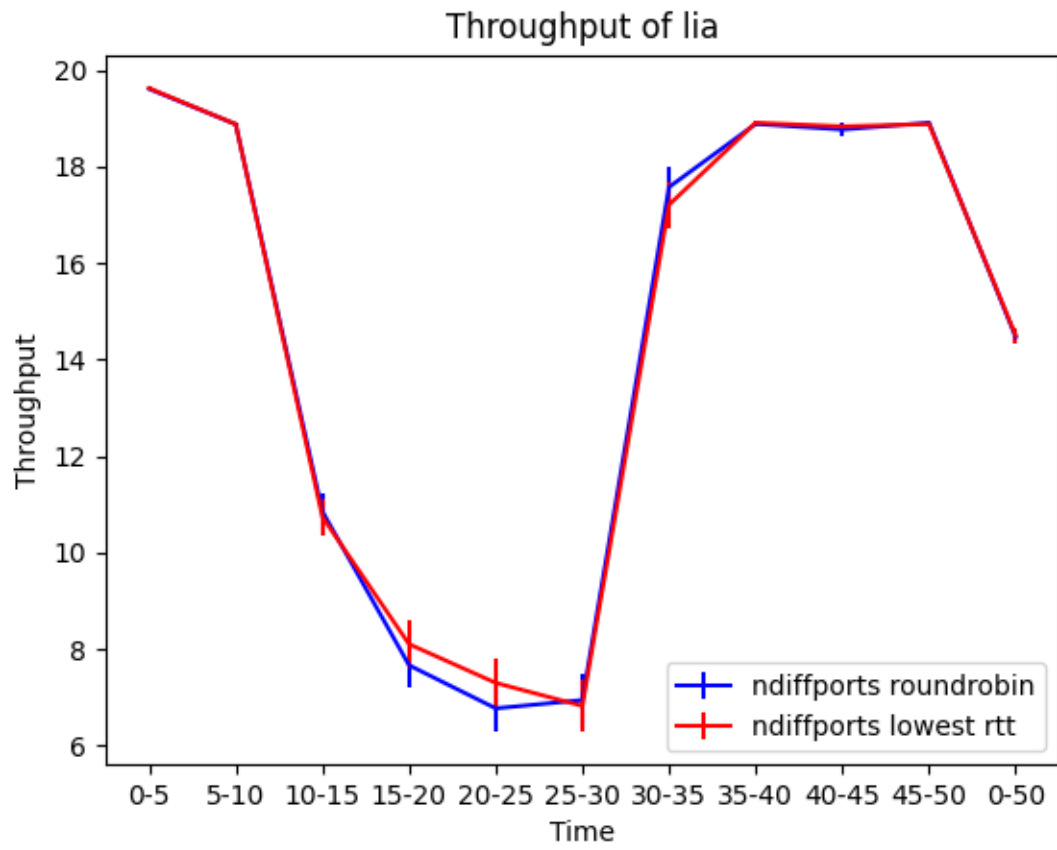
حال نمودار های حالت های قبلی را با هم ترکیب کرده تا تفاوت الگوریتم ها با یکدیگر مشخص شود.

### الگوریتم LIA و Fullmesh

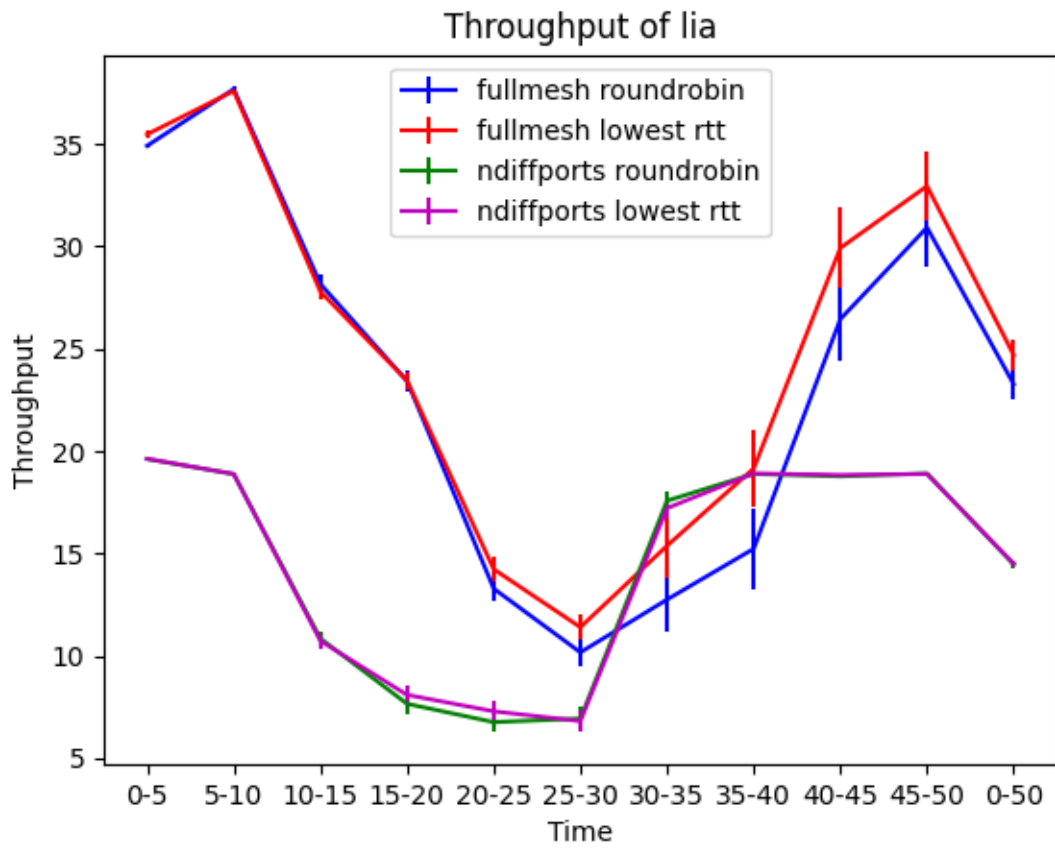




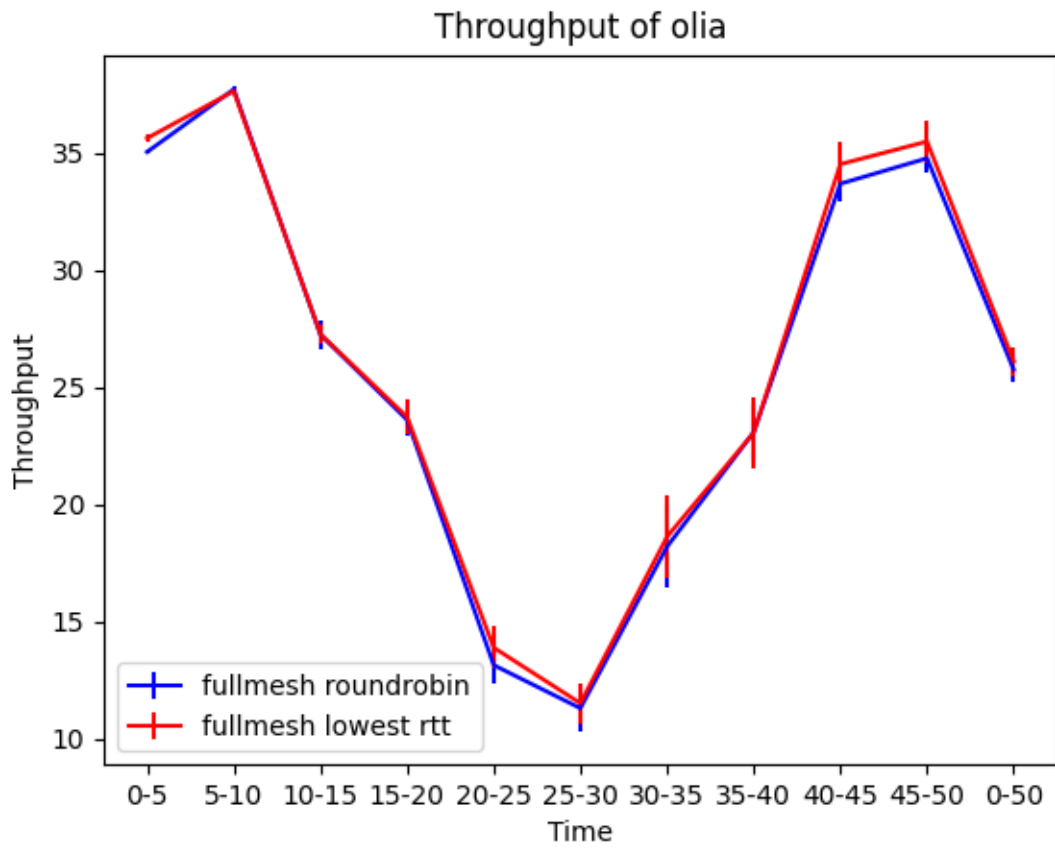
## الگوریتم LIA و ndiffports



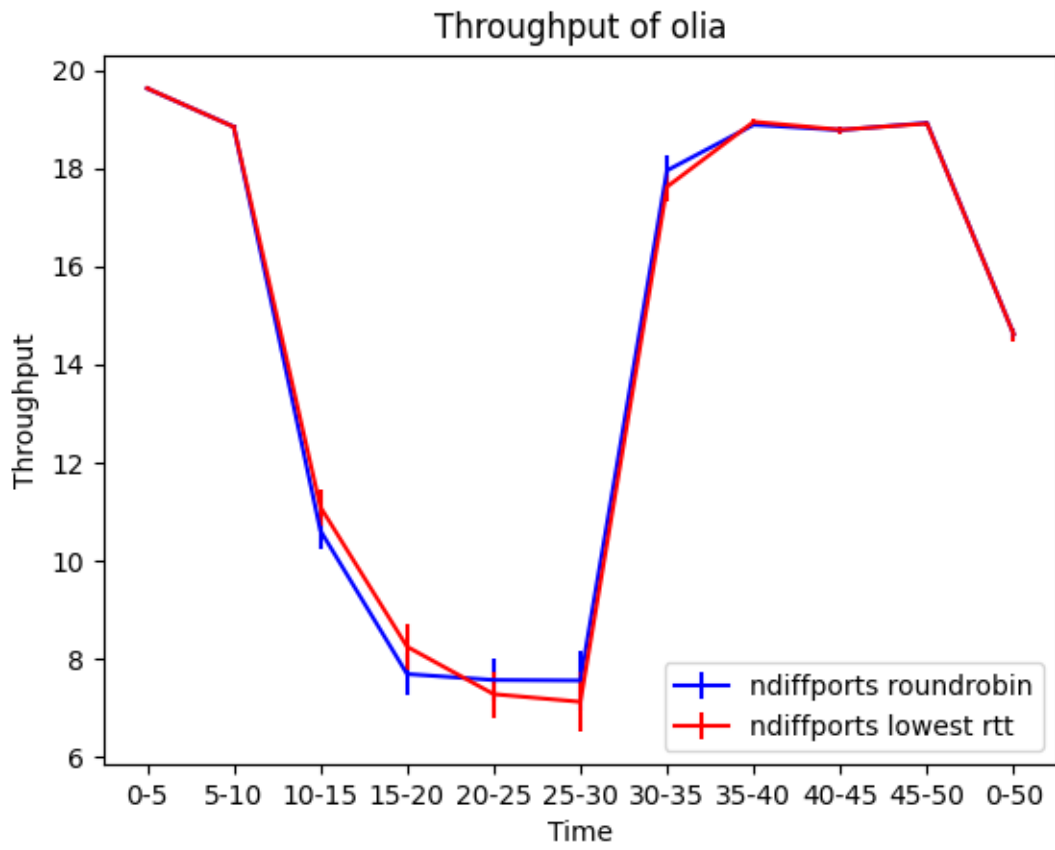
## الگوریتم LIA



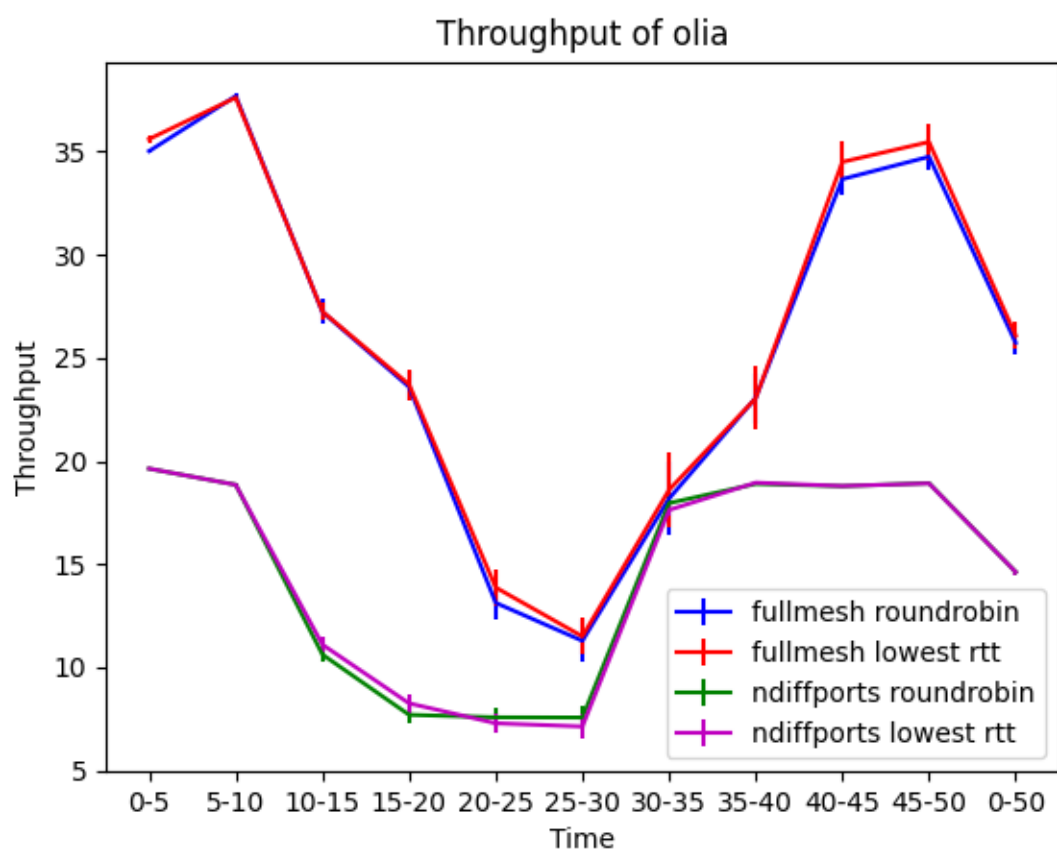
## الگوریتم Fullmesh و OLIA



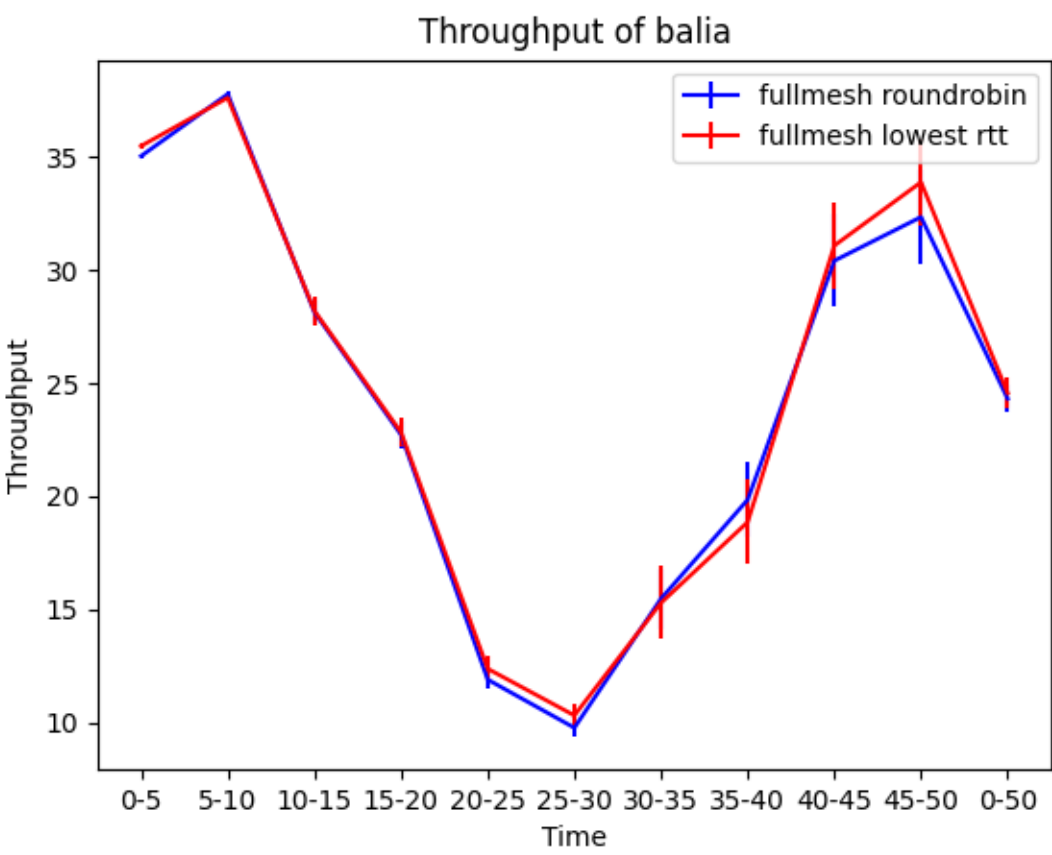
## الگوریتم OLIA و ndiffports



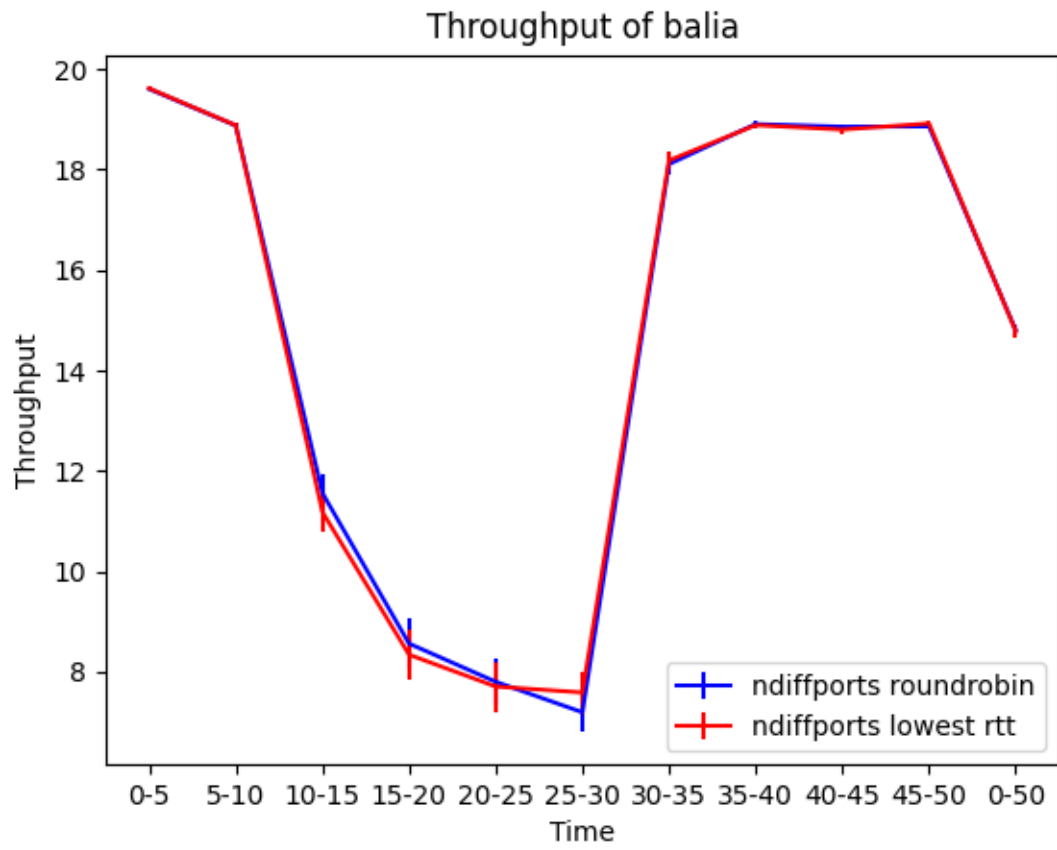
## الگوریتم OLIA



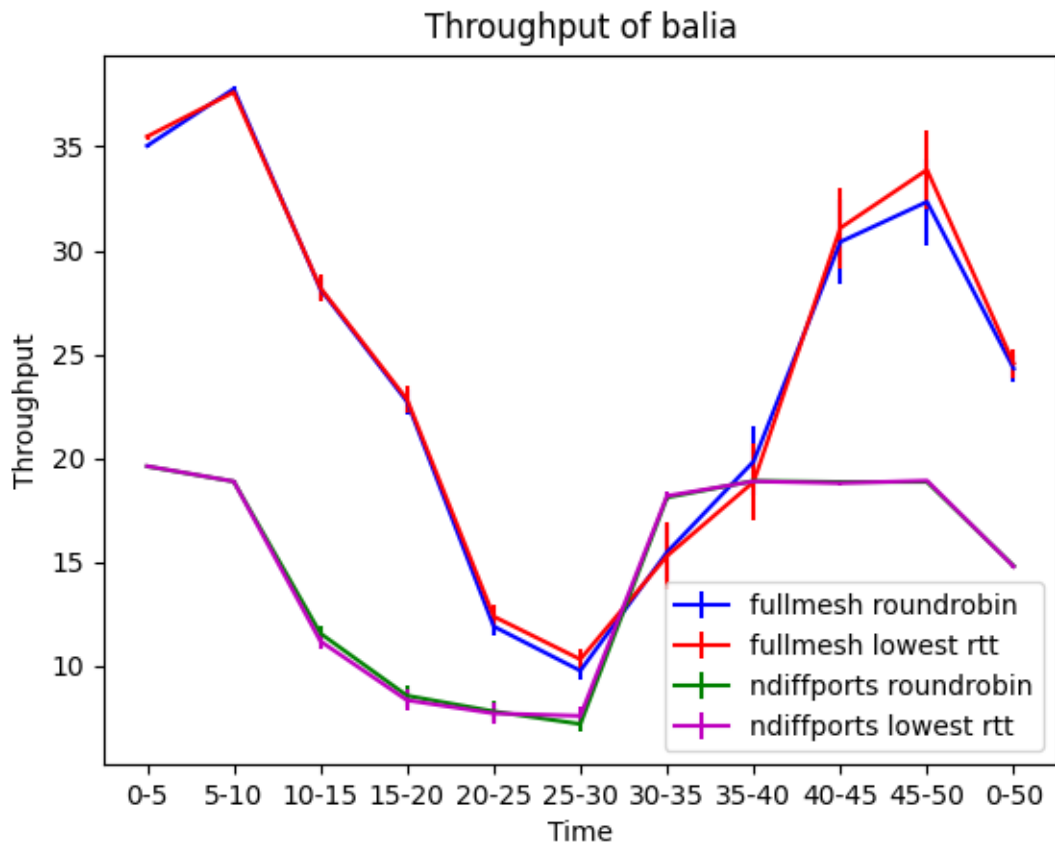
الگوریتم Fullmesh و BALIA



## الگوریتم BALIA و ndiffports

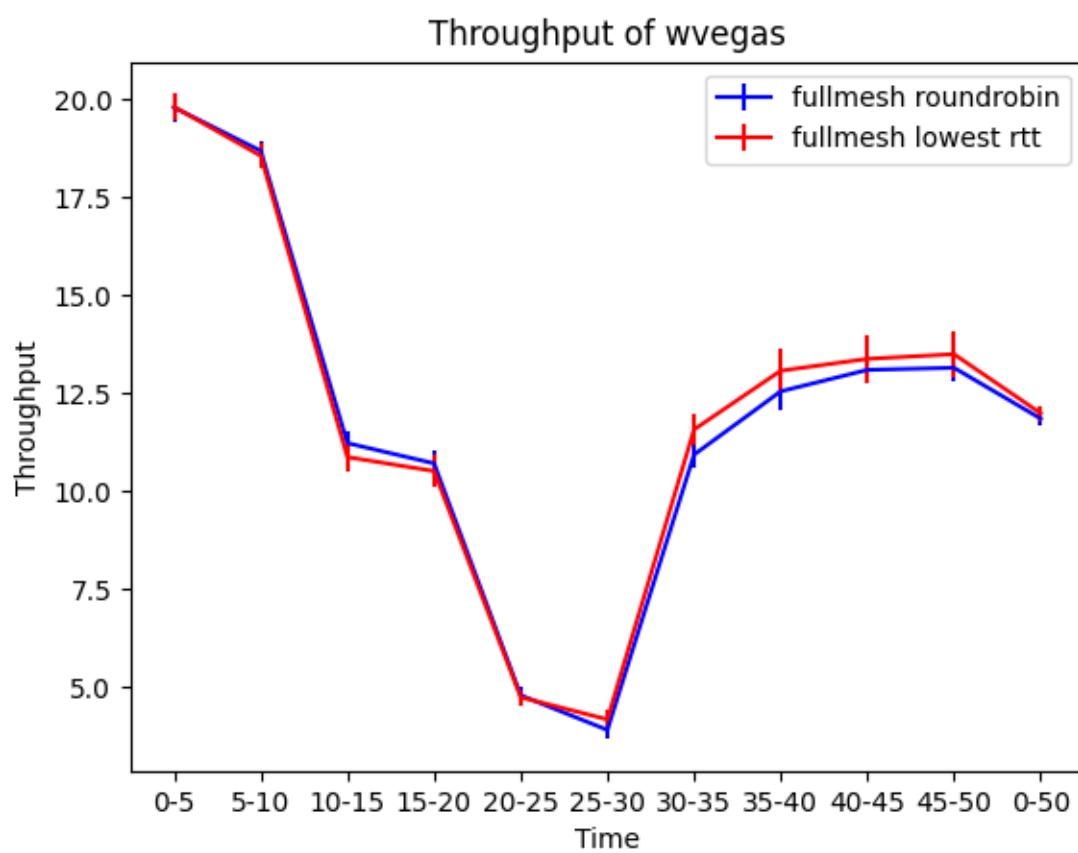


## BALIA الگوریتم

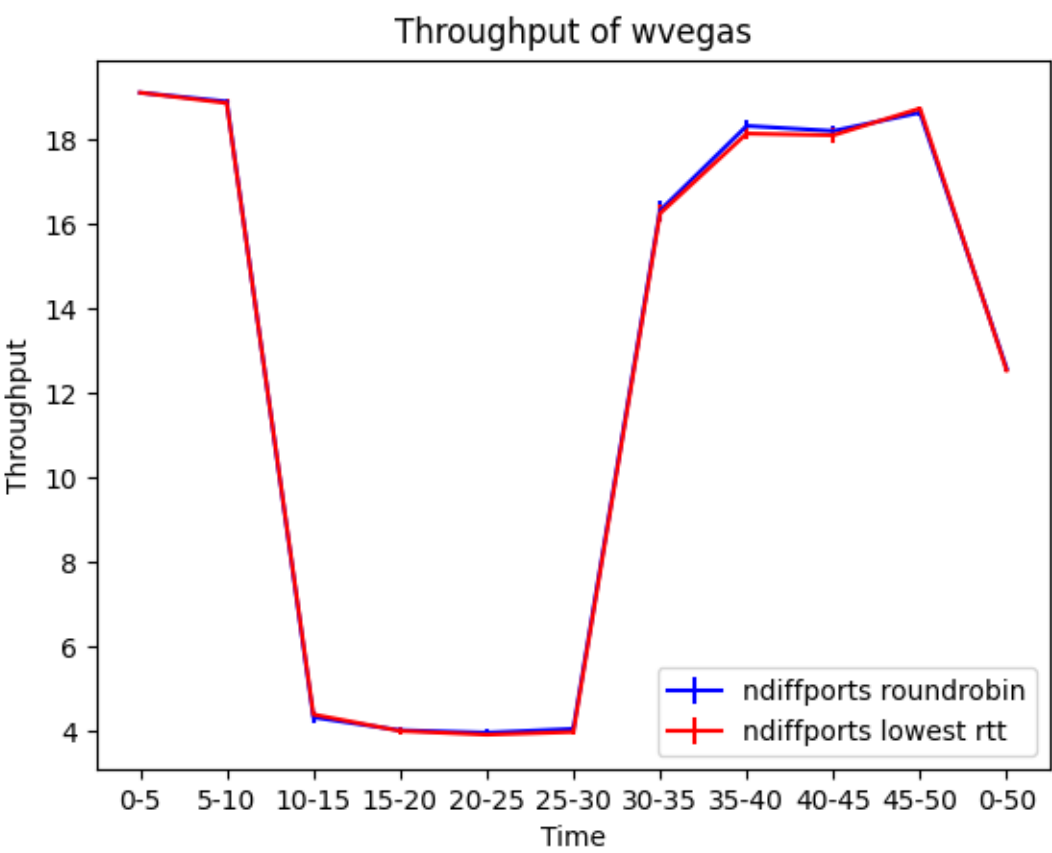




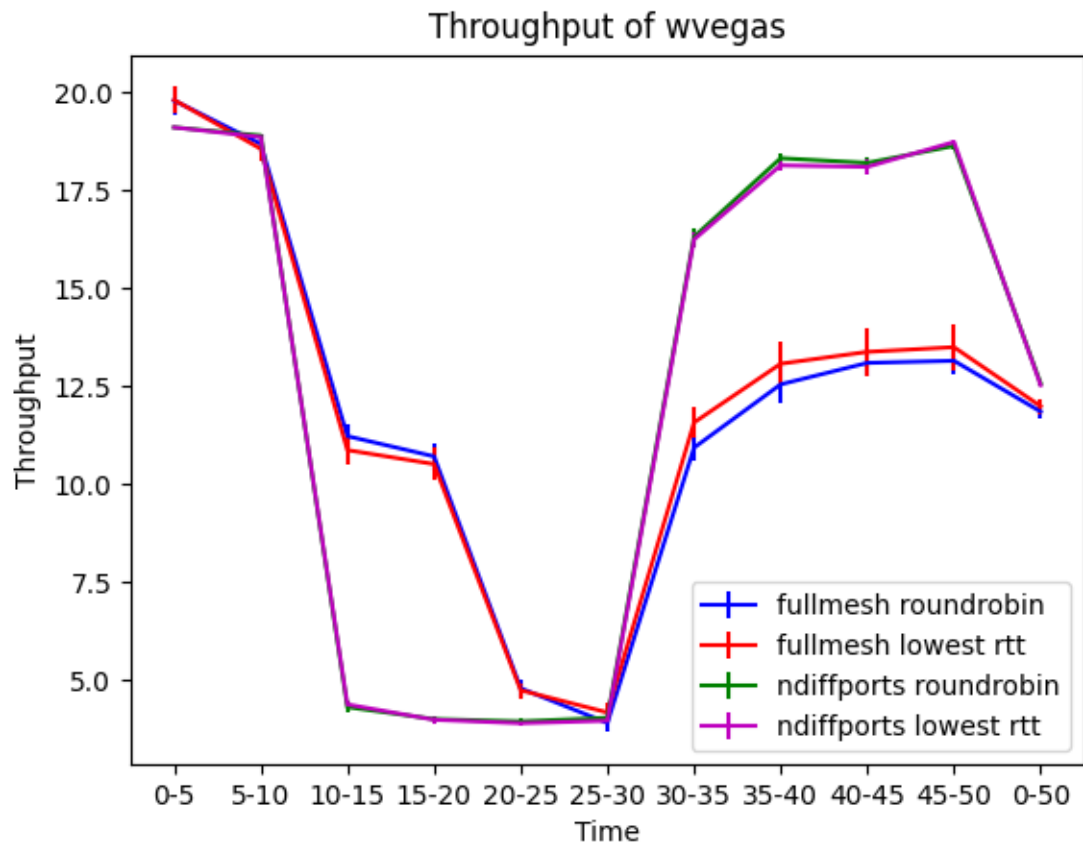
## الگوریتم WVEGAS و Fullmesh



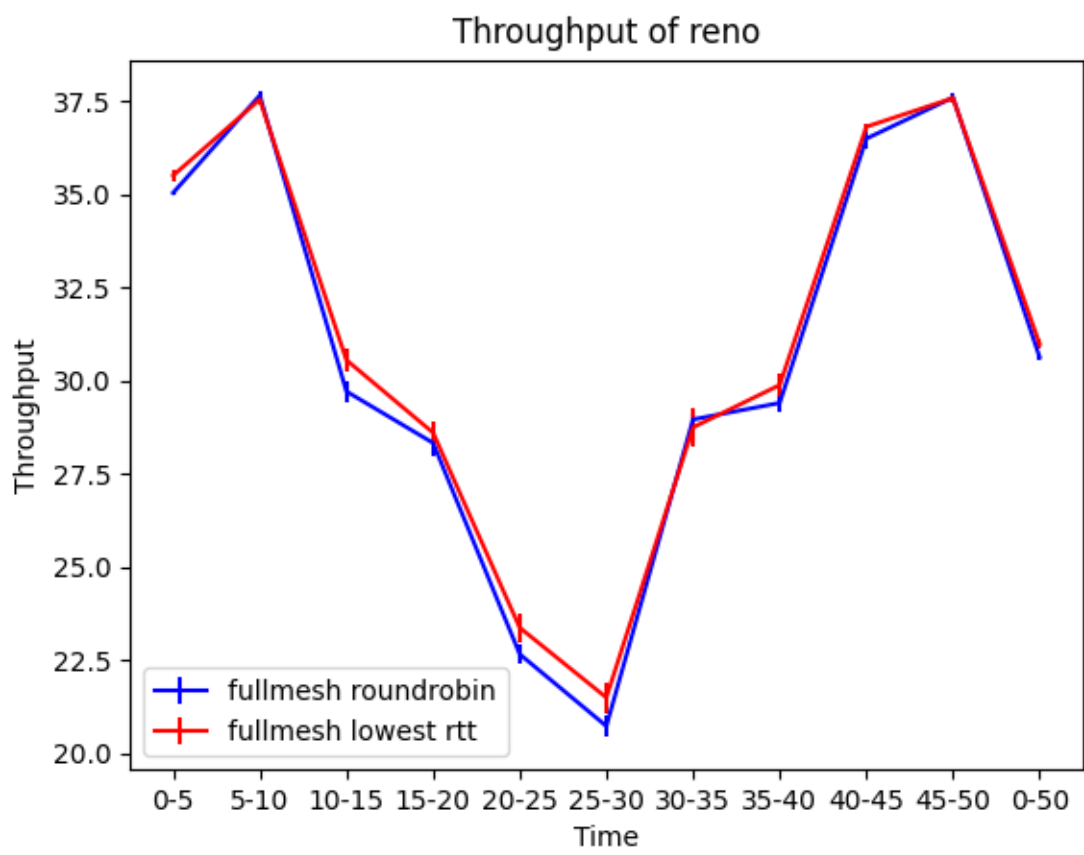
الگوریتم WVEGAS و ndiffports



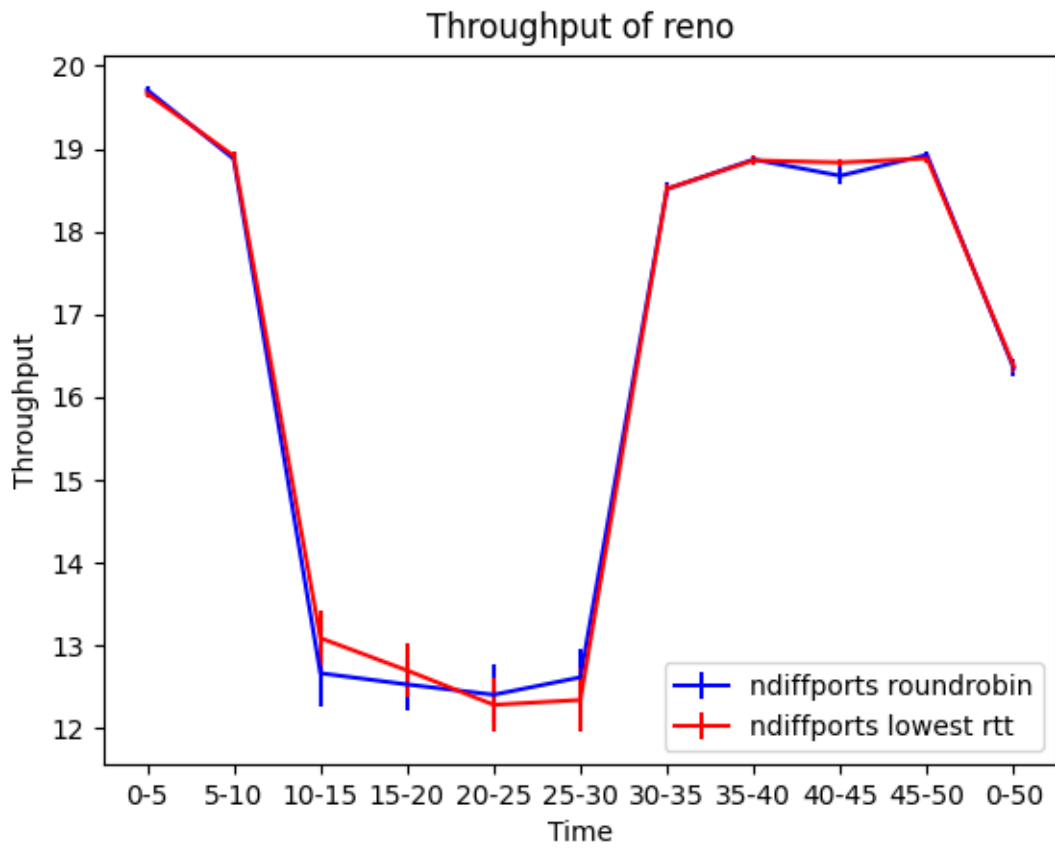
## WVEGAS الگوریتم



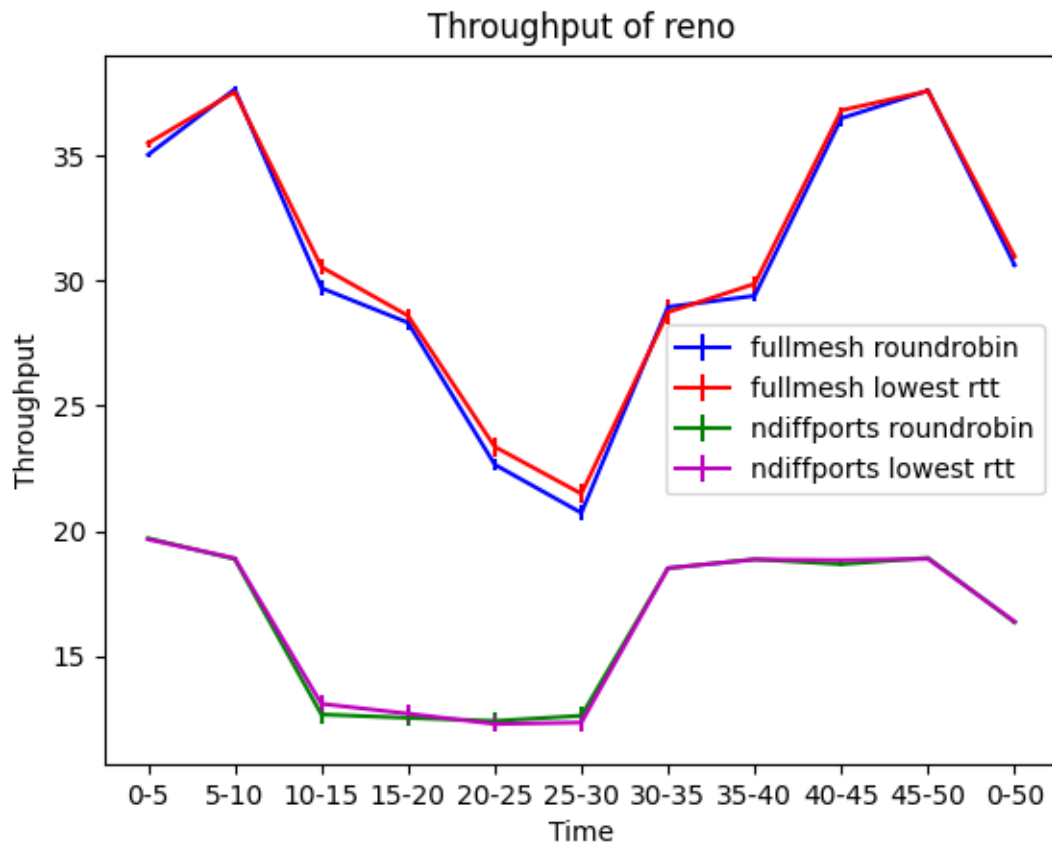
الگوریتم RENO و Fullmesh



## الگوریتم RENO و ndiffports

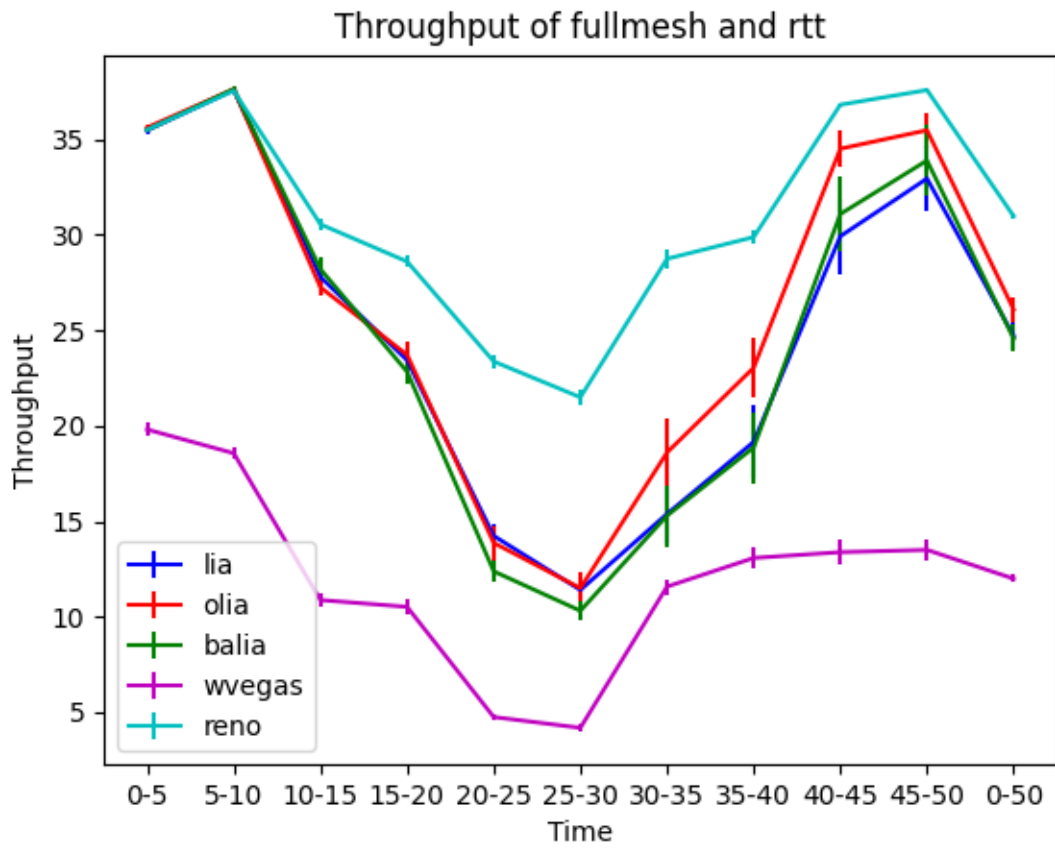


## الگوریتم RENO

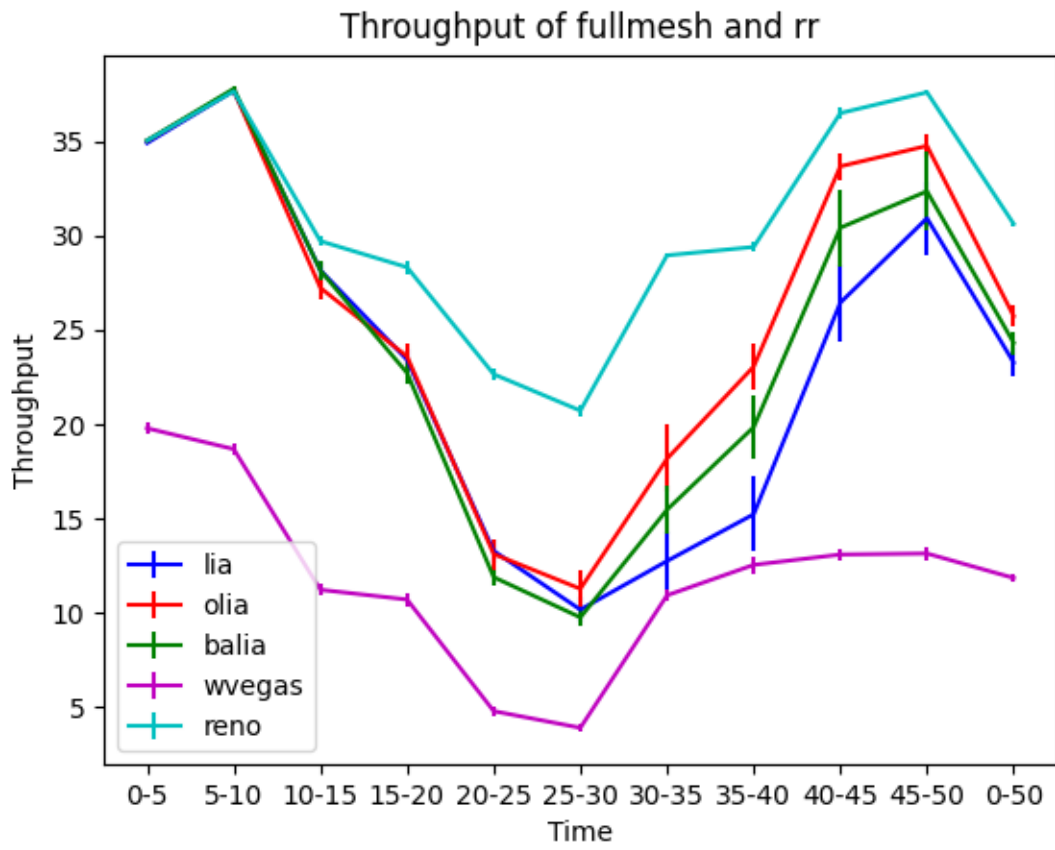




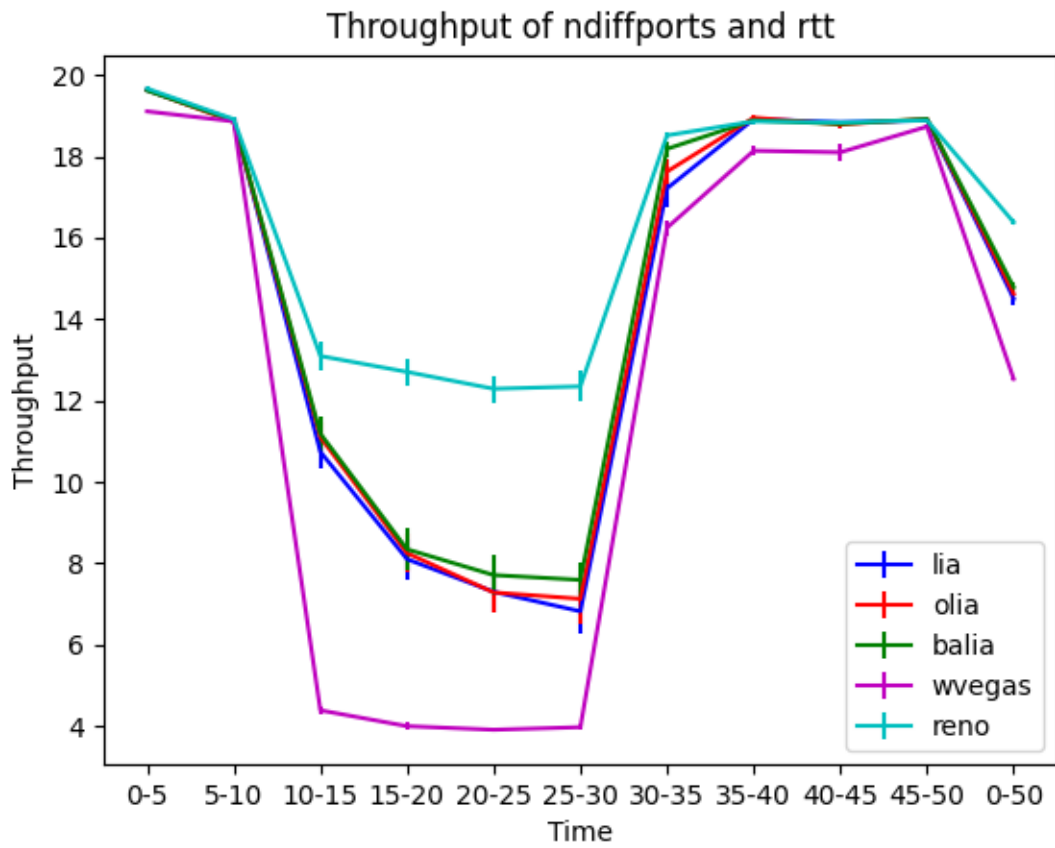
## الگوریتم LR و Fullmesh



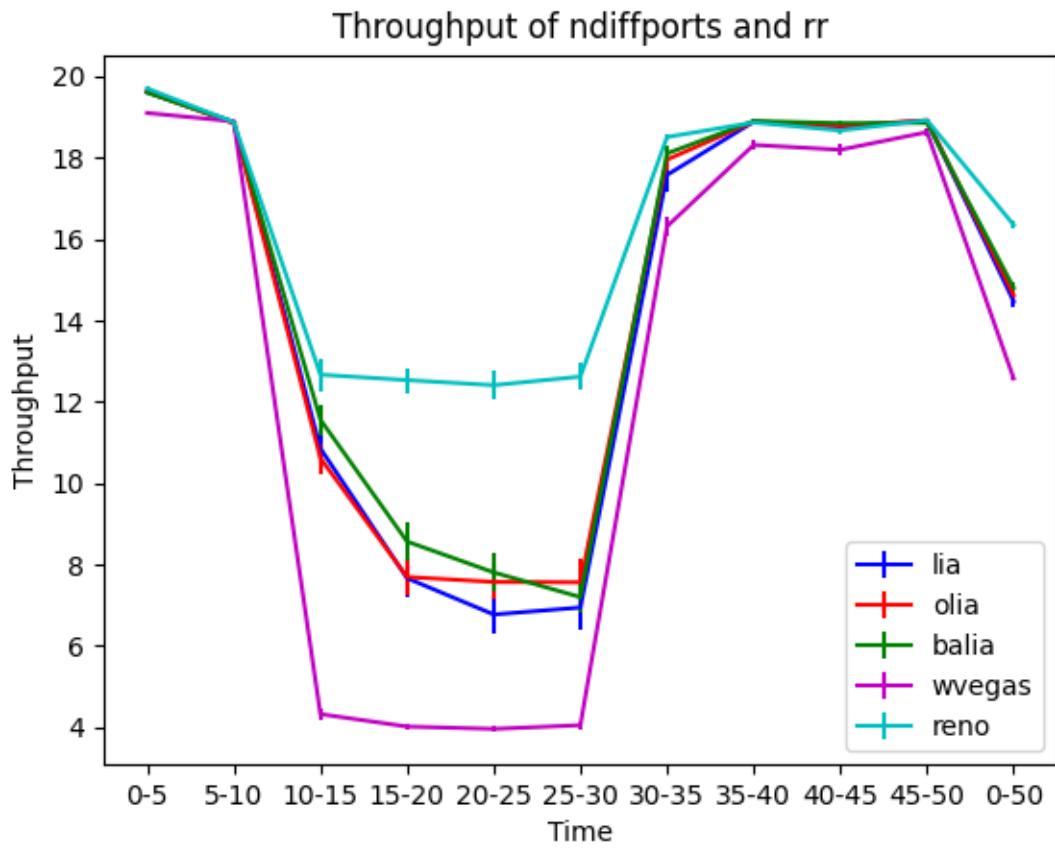
## الگوریتم RR و Fullmesh



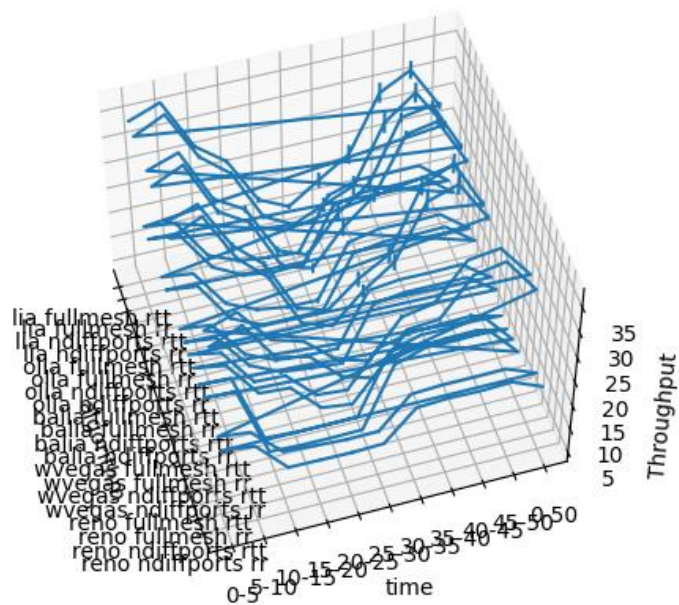
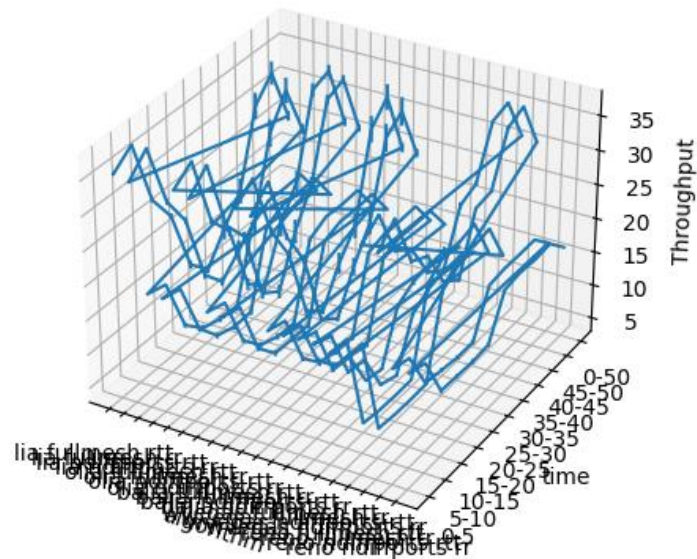
## الگوریتم LR و ndiffports



## الگوریتم RR و ndiffports



در پایان اگر همه نمودارها را بخواهیم به صورت سه بعدی نشان دهیم شکل های زیر ایجاد می شود.



## نتیجه گیری

- در حالتی که الگوریتم LIA را ثابت می کنیم و الگوریتم های زمانبند و مدیریت مسیر را تغییر می دهیم نتیجه می گیریم که به طور کلی الگوریتم Fullmesh بهتر از ndiffports عمل کرده است زمانی که از Fullmesh استفاده می کنیم از دو لینک بالا و پایین در mptcp استفاده می شود و زمانی که هر دو client ها از لینکی استفاده نمی کنند حداکثر پهنای باند می تواند تا 40mbps برسد و زمانی که یکی از client ها فعال است حداکثر پهنای باند می تواند تا 30mbps برسد و زمانی که هر دو فعال باشند تا 20mbps می رسد. در ndiffports فقط از لینک بالا استفاده می شود لذا بیشترین پهنای باندی که می تواند حاصل شود 20mbps است که در حالتی که 1 client شروع به ارسال داده می کند حداکثر پهنای باند به 10mbps می رسد.
- در حالتی که الگوریتم LIA را ثابت کرده و الگوریتم Fullmesh را نیز ثابت می کنیم از مقایسه الگوریتم های زمانبند نتیجه می گیریم که الگوریتم LR و RR از شروع فرآیند تا ثانیه 25 یکسان عمل کرده اند و از ثانیه 30 تا 40 با شیب تندتری الگوریتم LR نسبت به RR رشد کرده است. در مابقی حالات نیز شیب افزایش و کاهش یکسان است.
- در حالتی که الگوریتم OLIA و ndiffports را ثابت کنیم می توانیم نتیجه بگیریم که هنگام کاهش پهنای باند دریافتی الگوریتم RR سعی کرده است که از یک مینیمم پایین تر نیاید و از خود مقاومت نشان می دهد اما الگوریتم LR کاملاً به صورت نزولی کاهش یافته است.
- در حالتی که فقط الگوریتم OLIA را ثابت فرض کنیم می توانیم مشاهده کنیم که از ثانیه 30 تا 35 هر دو الگوریتم های Fullmesh و ndiffports برخلاف حالت LIA به پهنای باند نزدیک به همی رسیده اند.
- هنگامی که الگوریتم Fullmesh و LR را ثابت فرض کنیم می بینیم که الگوریتم RENO از همه بهتر کار می کند چون پهنای باند لینک به صورت مساوی بین client ها تقسیم نمی شود و mptcp client سهم بیشتری را می برد. الگوریتم wvegas از همه الگوریتم

های موجود بدتر عمل می کند چرا که به صورت محتاطانه ای پنجره ازدحام را افزایش می دهد و به نظر می آید که اصلاً از لینک پایینی استفاده نمی کند. بین الگوریتم های coupled در این حالت OLIA بهتر عمل کرده چرا که فرمول افزایش پنجره ازدحام آن سریعتر افزایش می یابد. الگوریتم های LIA و BALIA عملکرد یکسانی دارند.

- در حالتی که الگوریتم های Fullmesh و RR را ثابت فرض کنیم مانند قبل است به جز اینکه الگوریتم BALIA بهتر از LIA عمل می کند.

- در حالتی که الگوریتم های ndiffports و LR را ثابت می کنیم می بینیم که RENO از همه بهتر و wvegas از همه بدتر است. الگوریتم های coupled عملکرد تقریباً یکسانی را دارند. در این حالت چون الگوریتم ndiffports استفاده می شود الگوریتم های کنترل ازدحام به غیر از ثانیه 10 تا 30 عملکرد تقریباً یکسانی دارند چون از یک لینک استفاده می کنند.

- در حالتی که الگوریتم های ndiffports و RR را ثابت می کنیم نتایج مانند حالت قبلی است.