بسم الله الرحمن الرحیم

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

محمد مهدی سوری

حمیدرضا آذرباد

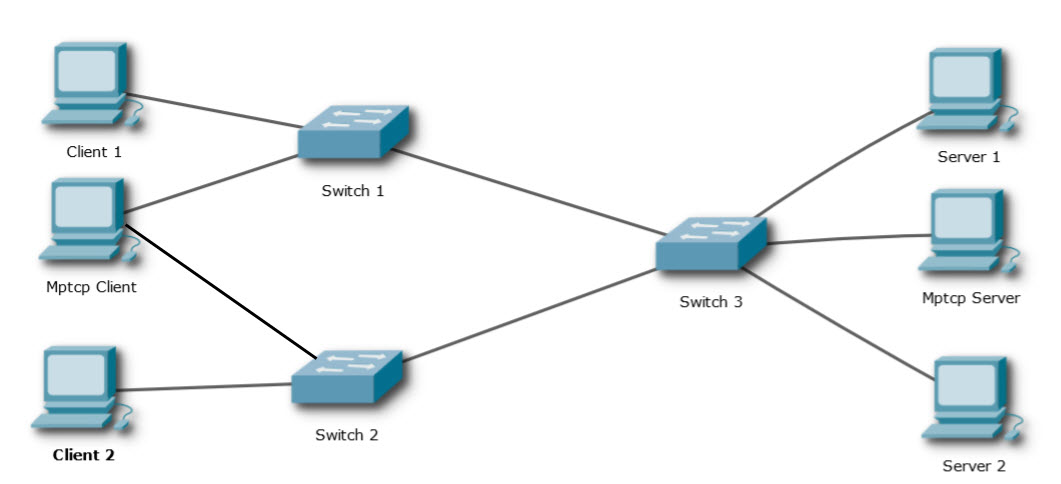
**مقدمه**

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 اتصلات این توپولوژی را نمایش می دهد.



سرعت و تاخیر لینک های bottleneck به ترتیب 20Mbps و 20ms است.

**سناریو**

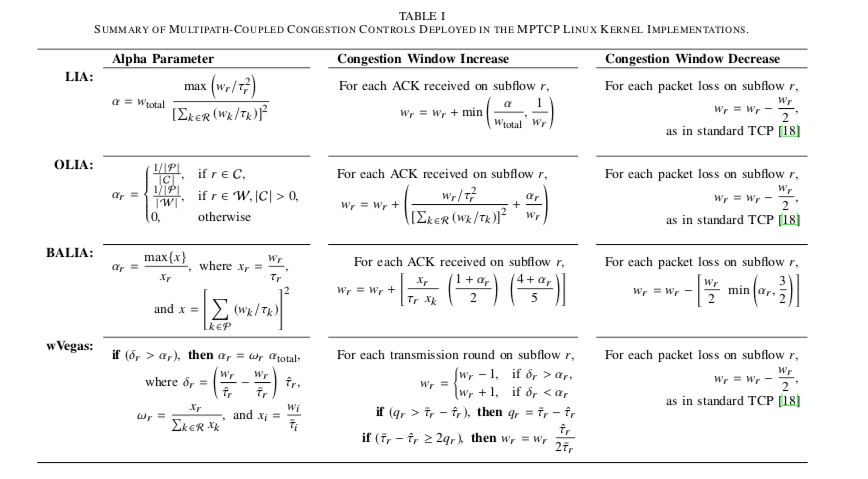
ابتدا از Mptcp client به Mptcp server به مدت 50 ثانیه داده ارسال می کنیم. در همین حین پس از 10 ثانیه از client 1 به server 1 نیز داده را به مدت 20 ثانیه به صورت TCP معمولی ارسال می کنیم. در زمان 20 ثانیه پس از شروع از client 2 نیز به server 2 داده را به مدت 20 ثانیه به صورت TCP معمولی ارسال می کنیم. 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 صرفا 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 گوییم.

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 LIA, OLIA, BALIA, wVegas, آمده است.



* **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 یه 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.

1. **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**

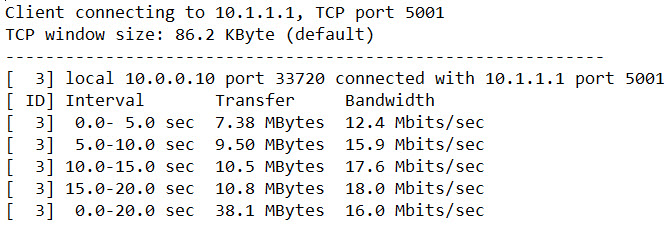
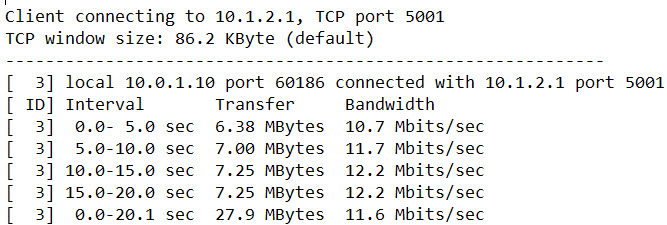
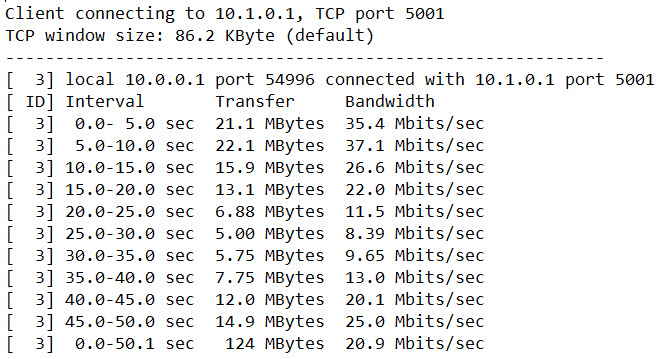
در این الگوریتم به ازای هر زوج آدرس IP کلاینت و سرور یک زیر جریان ترافیکی ایجاد می شود.

1. **الگوریتم ndiffports**

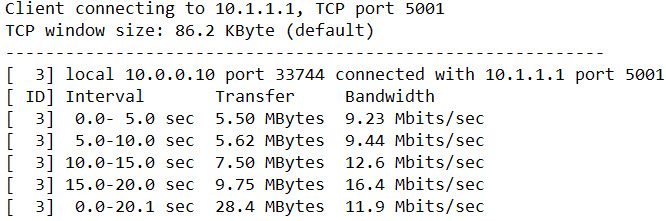
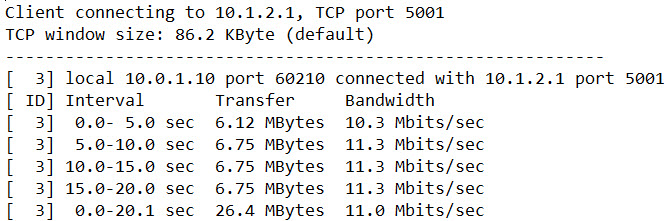
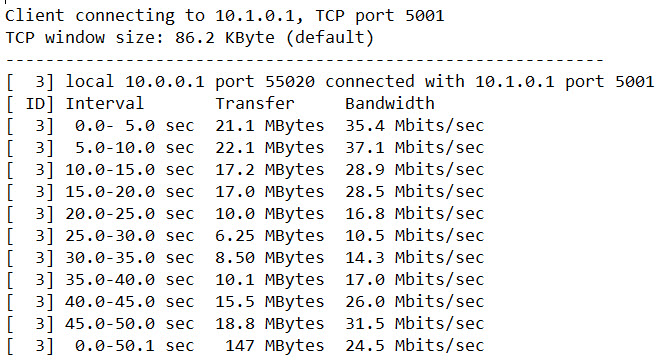
در این الگوریتم به ازای یک زوج آدرس IP کلاینت و سرور بیش از یک زیر جریان ترافیکی (به صورت پیشفرض دو) ایجاد می شود. روش این کار این است که شماره پورت source را متفاوت قرار می دهد.

**آزمایش**

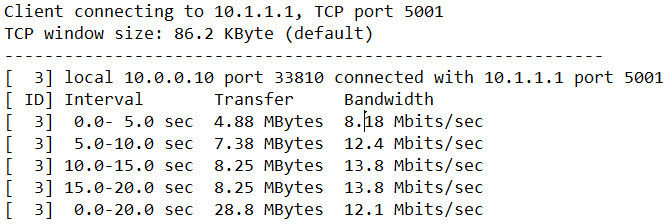
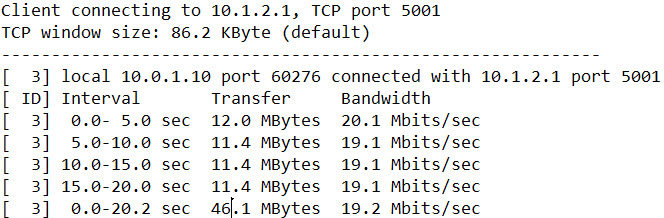
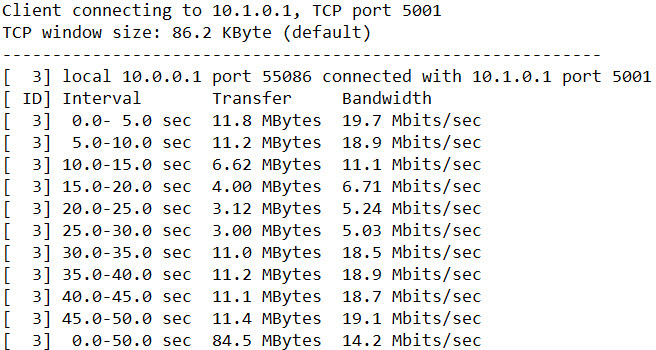
حالت 1 : کنترل ازدحام : LIA، مدیریت مسیر : Fullmesh، زمانبند : LR



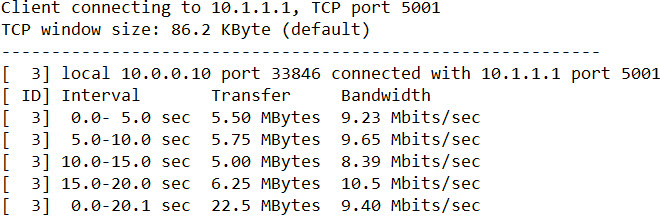
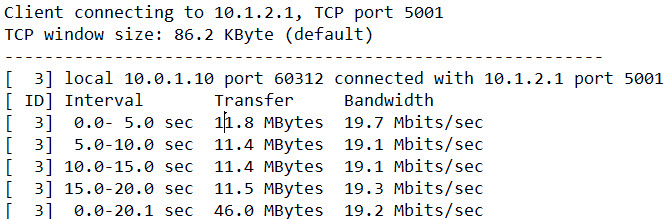
حالت 2 : کنترل ازدحام : LIA، مدیریت مسیر : Fullmesh، زمانبند : RR



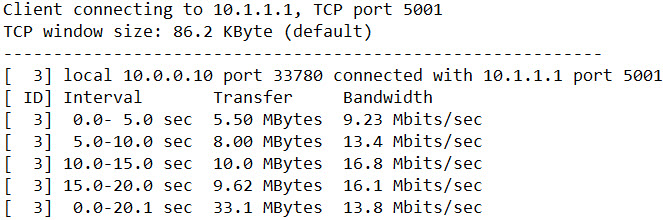
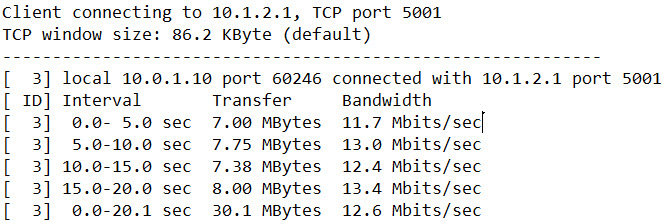
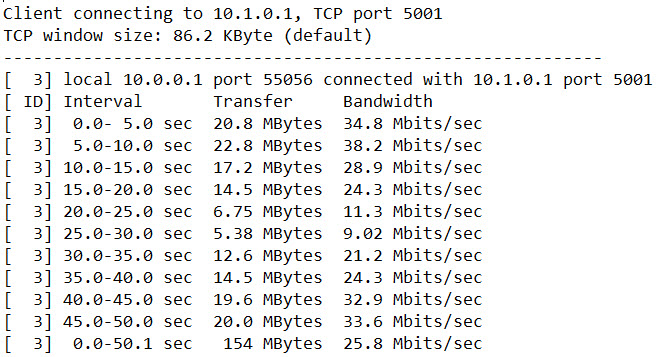
حالت 3 : کنترل ازدحام : LIA، مدیریت مسیر : ndiffports، زمانبند : LR



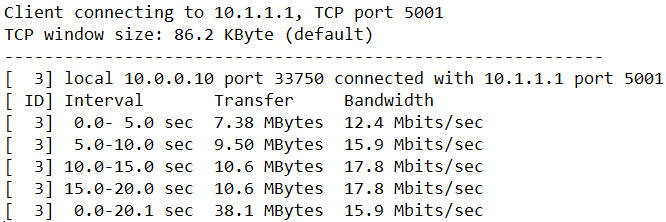
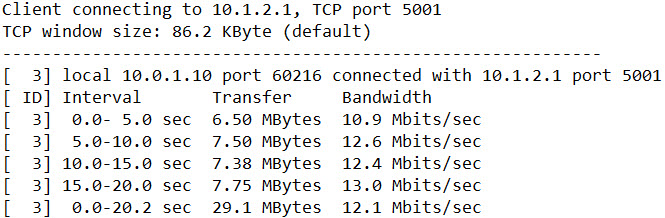
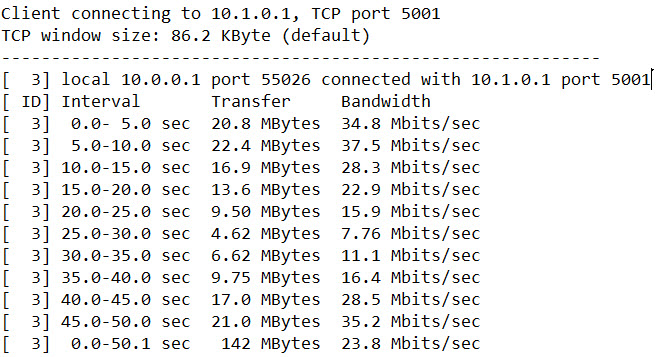
حالت 4 : کنترل ازدحام : LIA، مدیریت مسیر : ndiffports، زمانبند : RR



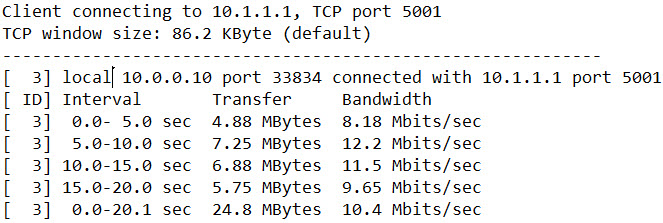
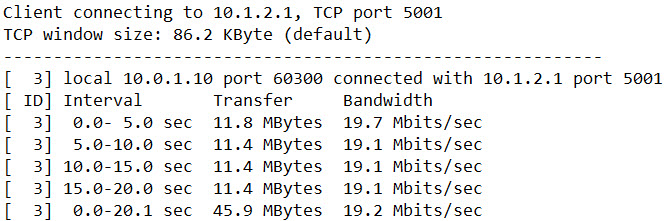
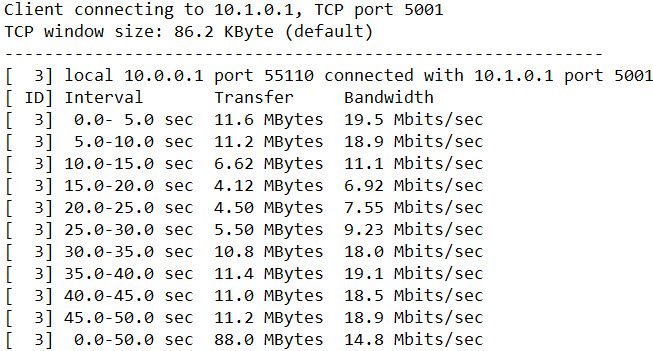
حالت 5 : کنترل ازدحام : OLIA، مدیریت مسیر : Fullmesh، زمانبند : LR



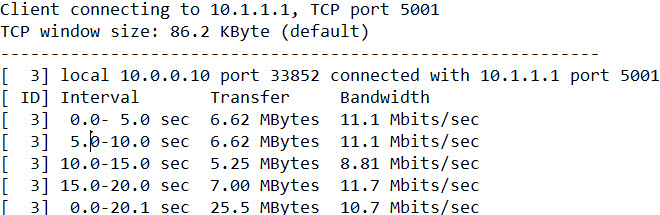
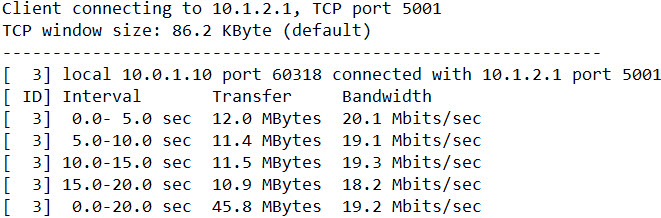
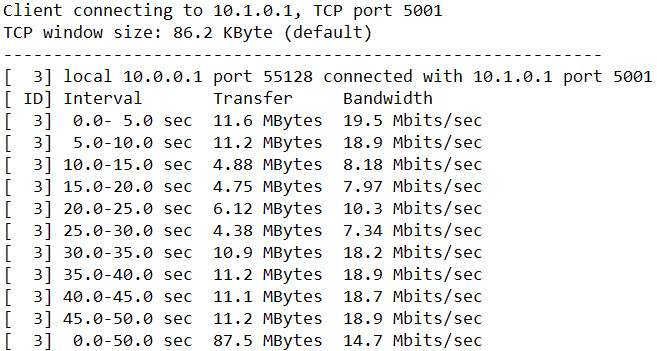
حالت 6 : کنترل ازدحام : OLIA، مدیریت مسیر : Fullmesh، زمانبند : RR



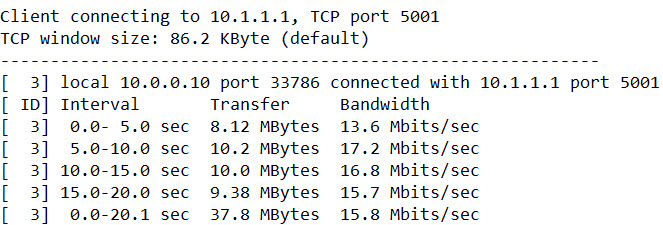
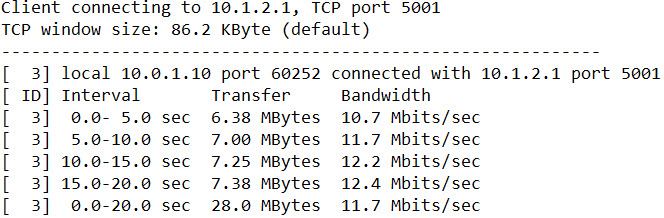
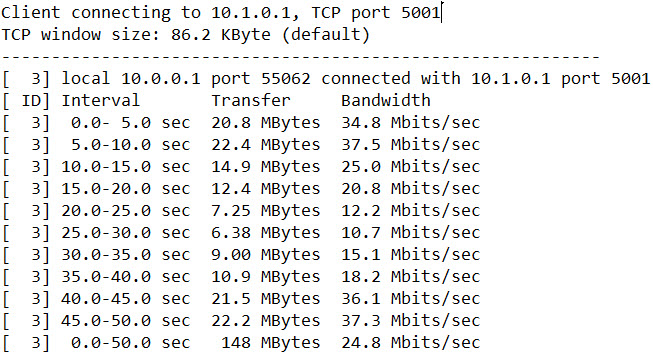
حالت 7 : کنترل ازدحام : OLIA، مدیریت مسیر : ndiffports، زمانبند : LR



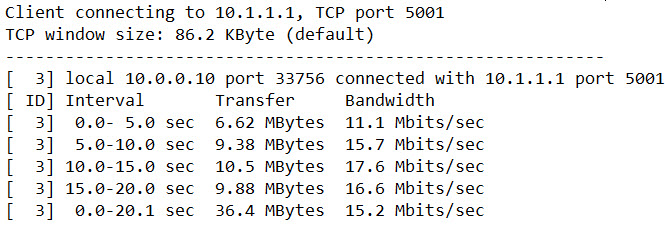
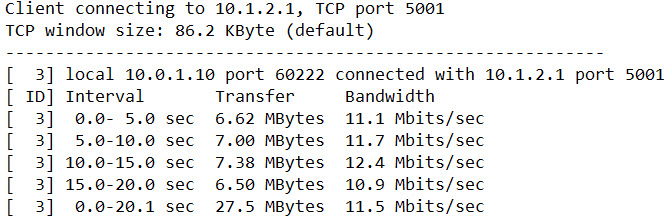
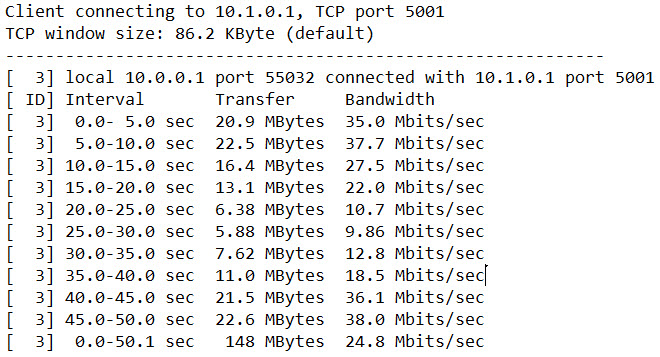
حالت 8 : کنترل ازدحام : OLIA، مدیریت مسیر : ndiffports، زمانبند : RR



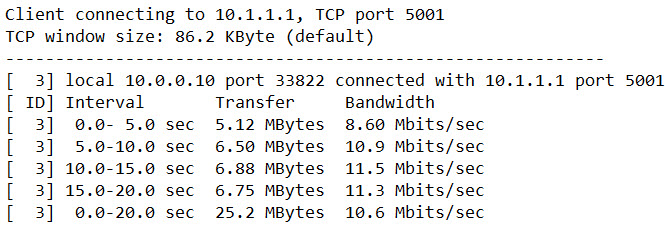
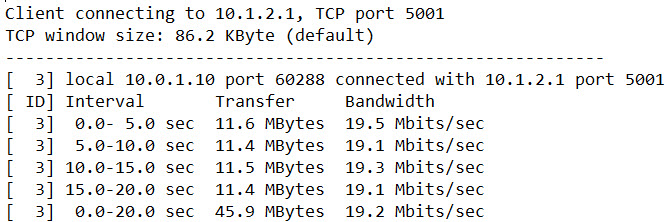
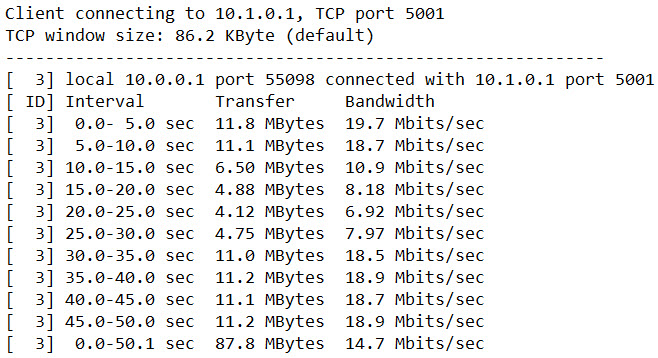
حالت 9 : کنترل ازدحام : BALIA، مدیریت مسیر : Fullmesh، زمانبند : LR



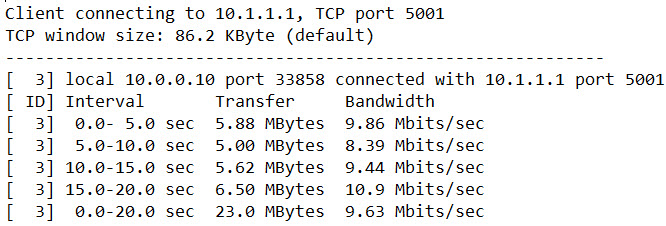
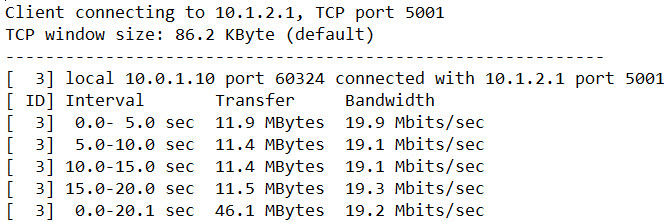
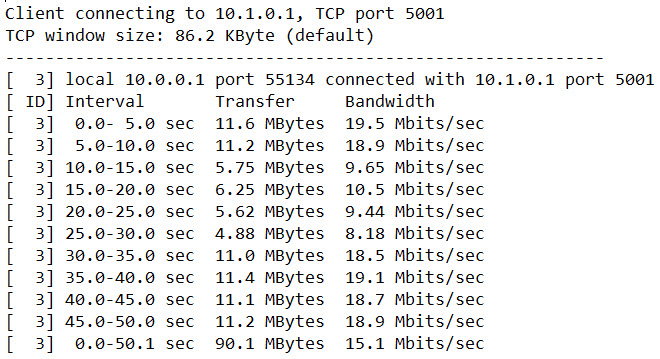
حالت 10 : کنترل ازدحام : BALIA، مدیریت مسیر : Fullmesh، زمانبند : RR



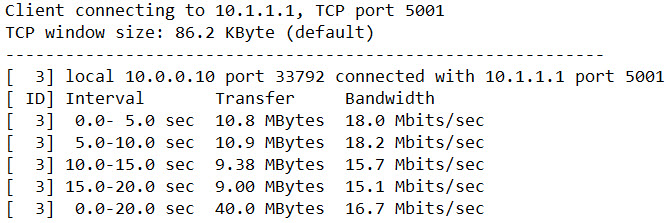
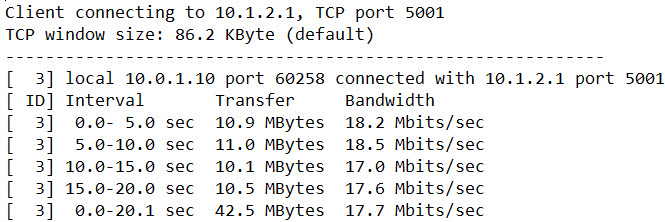
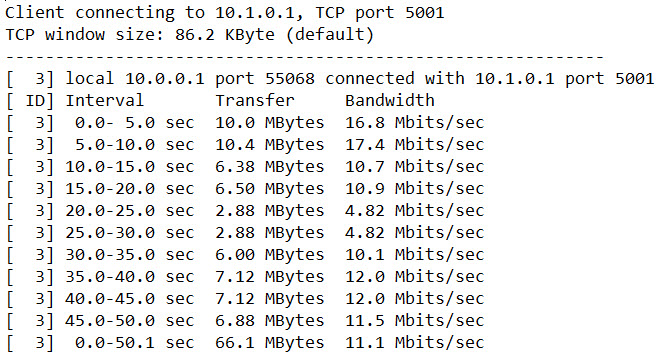
حالت 11 : کنترل ازدحام : BALIA، مدیریت مسیر : ndiffports، زمانبند : LR



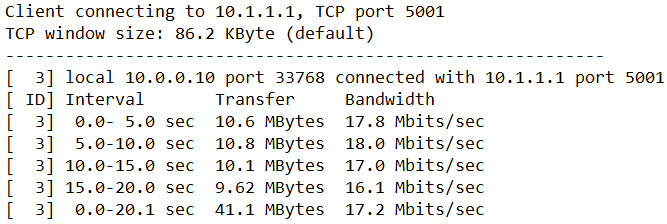
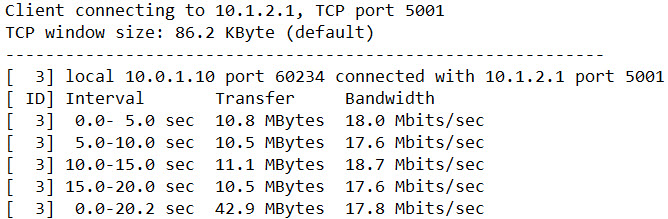
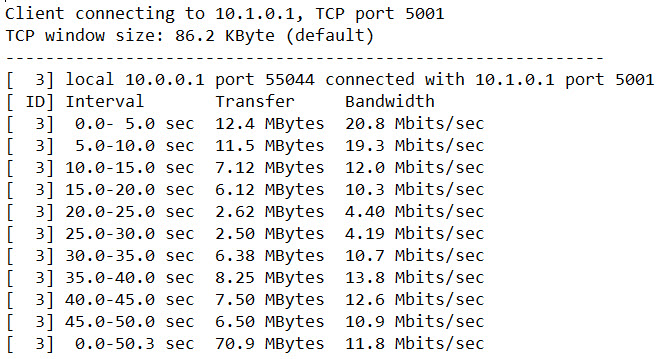
حالت 12 : کنترل ازدحام : BALIA، مدیریت مسیر : ndiffports، زمانبند : RR



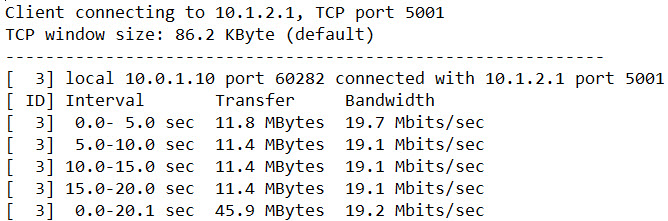
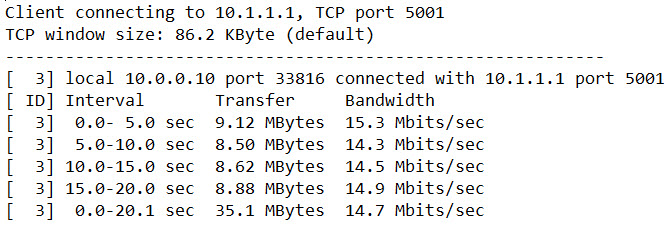
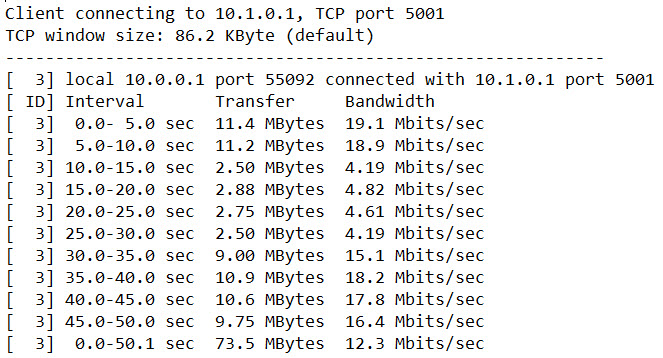
حالت 13 : کنترل ازدحام : WVEGAS، مدیریت مسیر : Fullmesh، زمانبند : LR



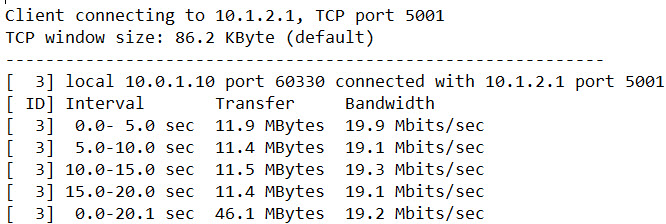
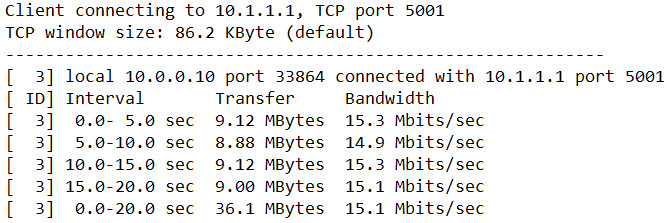
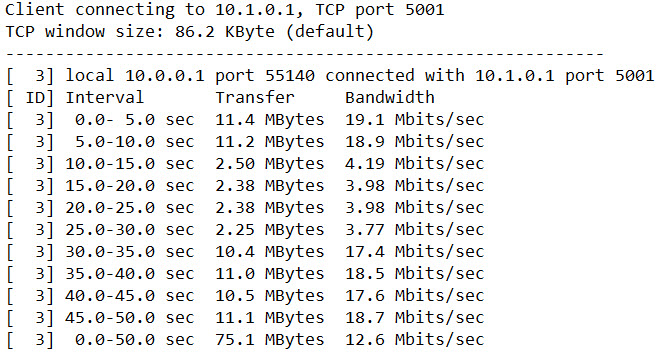
حالت 14 : کنترل ازدحام : WVEGAS، مدیریت مسیر : Fullmesh، زمانبند : RR



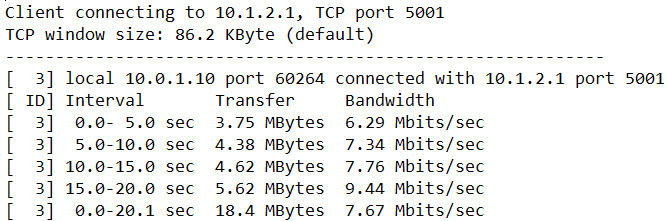
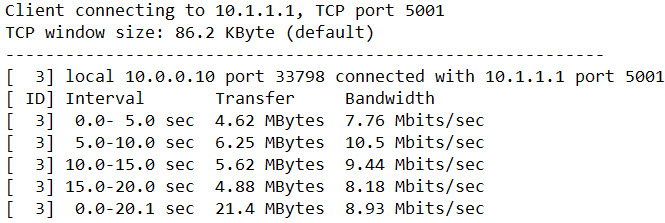
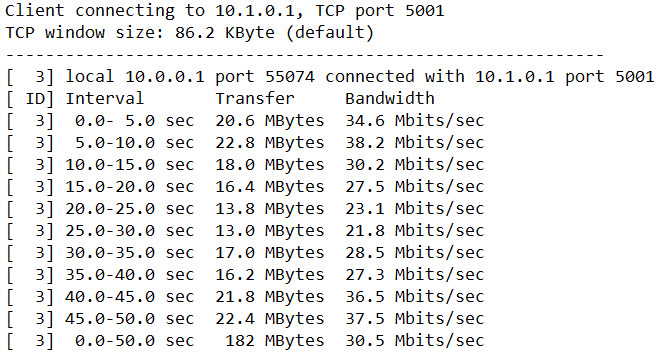
حالت 15 : کنترل ازدحام : WVEGAS، مدیریت مسیر : ndiffports، زمانبند : LR



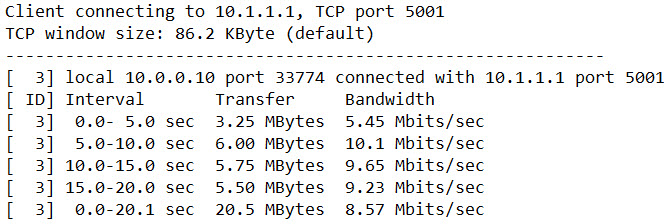
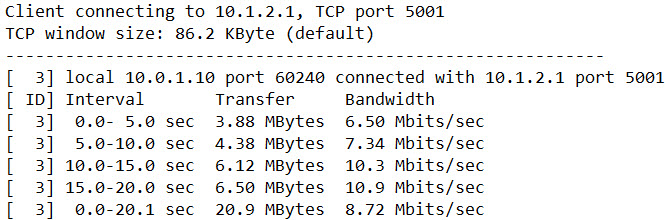
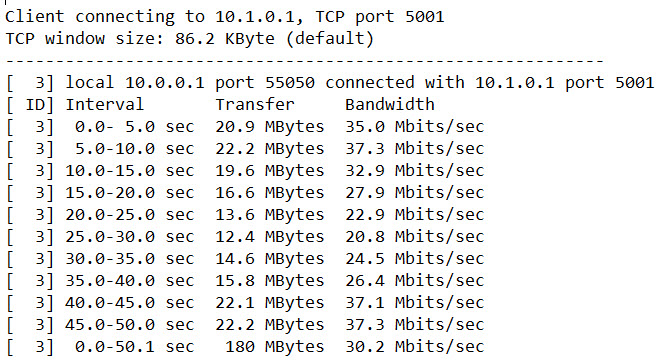
حالت 16 : کنترل ازدحام : WVEGAS، مدیریت مسیر : ndiffports، زمانبند : RR



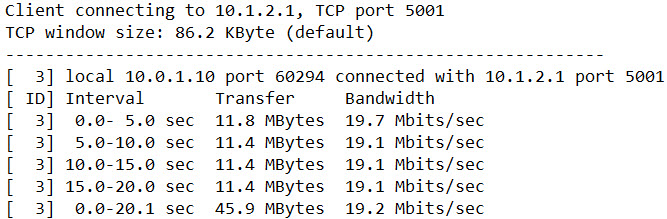
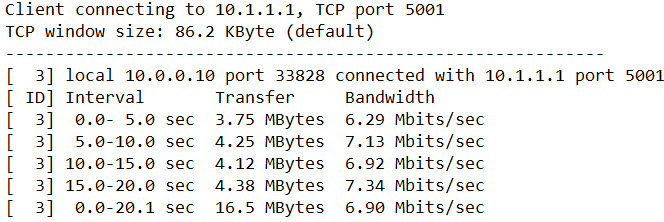
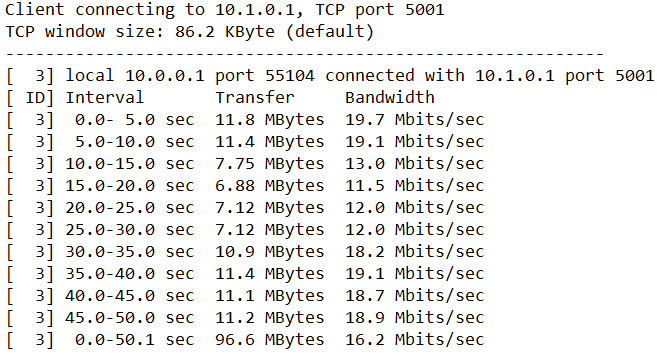
حالت 17 : کنترل ازدحام : RENO، مدیریت مسیر : Fullmesh، زمانبند : LR



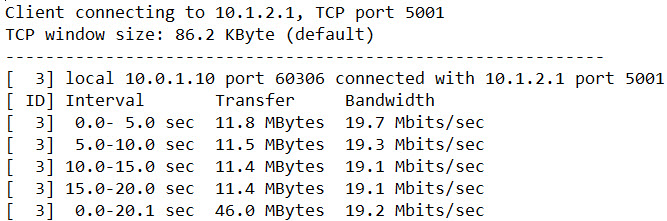
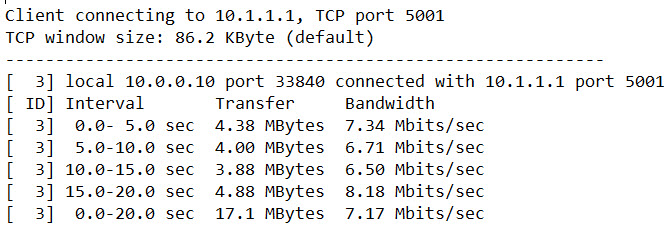
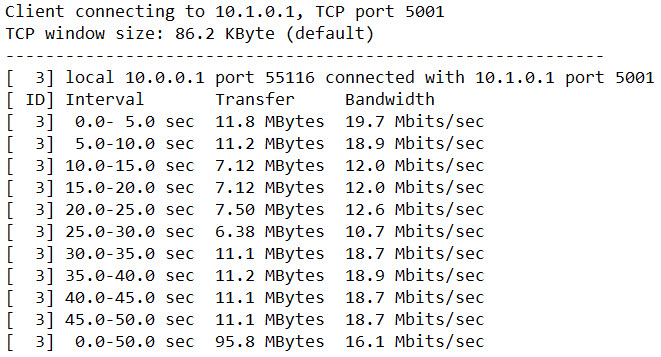
حالت 18 : کنترل ازدحام : RENO، مدیریت مسیر : Fullmesh، زمانبند : RR



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



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



**نتیجه گیری**

1. از مقایسه مدل های LIA,ndiffports,RR با LIA,ndiffports,LR به این نتیجه می رسیم که

* به دلیل استفاده از ndiffports، mptcp فقط از لینک بالا استفاده می کند در نتیجه از ثانیه 30 تا 40 که client 2 داده هایی را از لینک پایین ارسال می کند تاثیری بر پهنای باند و میزان داده انتقال یافته در mptcp ندارد.
* تفاوت پهنای باند این دو مدل در ثانیه های 10 تا 30 مشهود است دلیل آن هم این است که از LIA به عنوان الگوریتم کنترل ازدحام استفاده کرده که یک الگوریتم coupled است و عملاً پهنای باند لینک بالایی را نصف کرده و mptcp نیز دو زیرجریان در لینک بالایی دارد پس هر کدام از آن ها در بیشترین حالت پهنای باند 5mbps را دارند. در زمانبند RR از هر دوی این زیرجریان ها برای ارسال داده استفاده می شود اما در LR تنها از یکی از آنها استفاده می شود به همین دلیل پهنای باند نصف RR را داریم.

1. از مقایسه مدل های LIA,fullmesh,RR با LIA,ndiffports,RR به این نتیجه می رسیم که

* 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 client 1 شروع به ارسال داده می کند حداکثر پهنای باند به 10mbps می رسد.

1. از مقایسه مدل های LIA,ndiffports,RR با OLIA,ndiffports,LR به این نتیجه می رسیم که

* در زمان های 40 تا 50 ثانیه الگوریتم OLIA سریعتر به پهنای باند ماکسیمم نزدیک شده است که این تفاوت به دلیل فرمول افزایش پنجره ازدحام پس از رخ دادن ازدحام است.

1. از مقایسه مدل های LIA,fullmesh,RR با WVEGAS, fullmesh,LR به این نتیجه می رسیم که

* در حالت WVEGAS به صورت محتاطانه ای پنجره ازدحام افزایش می یابد در نتیجه نتایج کلی آن در مقایسه با دیگر الگوریتم ها بسیار پایین تر است.

1. از مقایسه مدل های WVEGAS,fullmesh,RR با WVEGAS,ndiffports,RR به این نتیجه می رسیم که

* افزایش سرعتی که توسط fullmesh به خاطر استفاده از دو مسیر ایجاد می شود در الگوریتم WVEGAS کاملا از بین می رود.

1. از مقایسه مدل های LIA,fullmesh,RR با RENO,ndiffports,LR به این نتیجه می رسیم که

* در زمان های 10 تا 40 که client 1 و client 2 داده ای را ارسال می کنند الگوریتم های coupled مانند LIA باعث می شوند که ترافیک لینک بین ارتباط TCP معمولی با MPTCP تقسیم شود لذا پهنای باند کمتری در مقایسه با الگوریتم RENO دریافت می کنند.