

مقدمه

هدف از این تمرین آشنایی اولیه با ابزار شبیه‌سازی NS2 و تحلیل الگوریتم‌های کنترل کردن ازدحام در سه نوع مدل از TCP با نام‌های Cubic، Reno و yeah است.

ابزار شبیه‌سازی NS2

شبیه‌ساز ns2 یک ابزار شبیه‌سازی رخداد-گسسته¹ است که امکان ارزیابی پارامترهای مختلف شبکه‌های کامپیوتری را برای پروتکل‌های استاندارد یا دلخواه فراهم می‌کند. این نرم‌افزار امکان شبیه‌سازی لایه‌های TCP/IP را در بسترهای مختلف سیمی و بی‌سیم مانند LAN، WAN، Ad Hoc، Satellite ... فراهم می‌کند. همچنین انواع پروتکل‌ها در لایه‌های مختلف مانند TCP، UDP، FTP، AODV، DSR و ... در ns2 پیاده‌سازی شده است. در سال‌های اخیر به دلیل انعطاف و قدرت بالای ns2 در شبیه‌سازی شبکه‌های بی‌سیم توجه بسیاری به این شبیه‌ساز شده است. همچنین به دلیل استفاده‌ی اکثر مقالات پژوهشی خصوصاً در حوزه‌ی شبکه‌های بی‌سیم از این شبیه‌ساز، اعتبار این شبیه‌ساز در جامعه پژوهشی بسیار بالا می‌باشد. برای پیاده‌سازی سناریوهایی که می‌خواهیم به کمک این ابزار شبیه‌سازی کنیم از زبان tcl استفاده می‌شود.

برای آشنایی با NS2 و یادگیری کار با زبان tcl می‌توانید از منابعی که روی سایت قرار گرفته شده‌است استفاده کنید.

کنترل ازدحام در PCT²

کل ازدحام در شبکه شبکه‌های کامپیوتری به مفهوم جلوگیری از تجمع بیش از اندازه داده‌های انتقالی در صف انتظار و در نتیجه از بین رفتن بسته‌ها می‌باشد. ازدحام زمانی رخ می‌دهد که میزان داده ورودی به گذرگاه داده بیشتر از حد تحمل آن گذرگاه باشد. یکی

¹ Discrete event

² TCP congestion control

از مهم‌ترین ویژگی‌های TCP همانطور که در درس یاد گرفتید کنترل ازدحام است که آن را به کمک چند مرحله شروع آهسته³، پیشگیری از تصادم⁴، ارسال دوباره سریع⁵ و بهبود سریع⁶ پیاده‌سازی می‌کند. الگوریتم‌های متفاوتی برای پیاده‌سازی این مراحل و کنترل ازدحام به وجود آمده‌اند که از جمله آنان Reno، Cubic و yeah است.

شرح تمرین

برای انجام پروژه گام‌های زیر باید طی شوند:

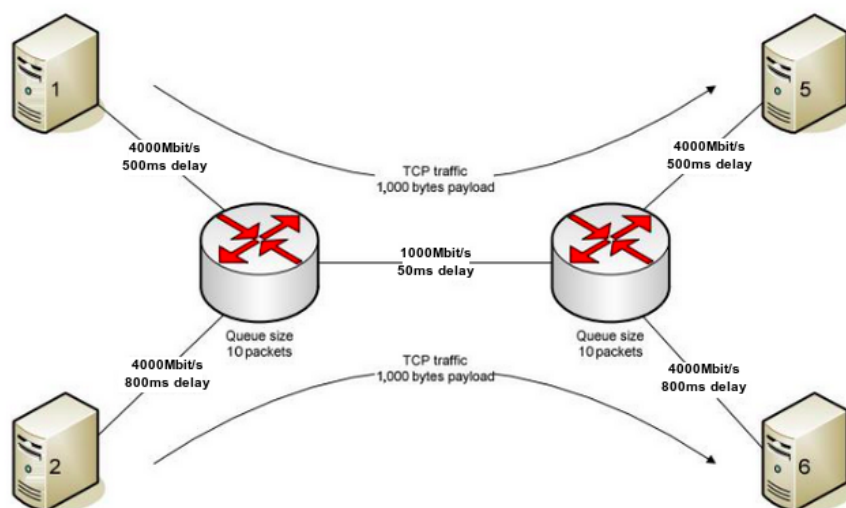
۱. تعریف توپولوژی که شبیه‌سازی روی آن انجام می‌شود در محیط NS2 به کمک زبان tcl

۲. شبیه‌سازی هر یک از روش‌های کنترل ازدحام Reno، Cubic و yeah در NS2

۳. تولید نمودارهای گفته‌شده برای کمک به تحلیل سه روش کنترل ازدحام

۴. تحلیل سه روش به کمک نمودارهای به دست‌آمده از مرحله قبل

توپولوژی شبکه



³ Slow start

⁴ Congestion avoidance

⁵ Fast retransmit

⁶ Fast recovery

در این توپولوژی پهنای باند هر لینک به همراه میزان تاخیر هر کدام از لینکها نمایش داده شده است. شرایط زیر برای شبکه بالا مدنظر می باشد:

- جریانی از گره ۱ به ۵ و همچنین جریانی از گره ۲ به ۶ وجود داشته و مدام داده در حال انتقال است.
- اندازه صف در روترها برابر با ۱۰ بسته است. (راهنمایی: ابتدای لینک بین دو روتر سمت چپ و راست صف را قرار دهید)
- مقدار TTL برابر با ۶۴ می باشد.
- اندازه پنجره اولیه را ۸۰۰۰ در نظر بگیرید.
- ظرفیت شبکه همواره ثابت در نظر گرفته شود. از اندازه بسته TCP پیشفرض NS2 یعنی ۱۰۰۰ بایت استفاده شود (مطابق تصویر).

شبیه سازی با توجه به شرایط زیر انجام شود:

- شبیه سازی برای ۱ مرتبه و برای ۱۰۰۰ ثانیه انجام شود.
- فرض شود که جریان مدام در حال ارسال داده است و در لینکهای داده ای Lost نمی شود (البته در روترها بعلت محدود بودن بافر، از دست رفتن بسته خواهیم داشت). فرض شود که پهنای باند هر لینک همواره ثابت است.

شبیه سازی روش های کنترل ازدحام در محیط NS2

برای شبیه سازی روش های کنترل ازدحام ذکر شده، اسکریپت هایی در اینترنت موجود هستند که می توانید از آنها استفاده کنید و تنها کافیست کد شبکه ای که شبیه سازی روی آن انجام می شود را تغییر دهید. برای دنبال کردن پارامترهای مورد نیاز برای تولید نمودار می توانید از کد نمونه ای که در سایت قرار گرفته است استفاده کنید.

نمودارهای خروجی

هدف نهایی از شبیه سازی بررسی و مقایسه معیارهای زیر در ۳ نوع پروتکل TCP مدنظر میباشد:

- تغییرات اندازه پنجره ازدحام CWND
- توضیح: اندازه پنجره ازدحام در هر ثانیه را رسم کنید. معیار را اندازه پنجره ازدحام در انتهای هر ثانیه بگیرید.
- نرخ GoodPut

توضیح: این نرخ تعداد بیت‌های اطلاعات مفید جابجا شده توسط شبکه به مقصد مشخص در واحد زمان را نشان می‌دهد. برای مثال اگر تا انتهای ثانیه ۵، ۴۰ بسته و تا انتهای ثانیه ۴، ۲۵ بسته توسط مقصد دریافت شده باشد نرخ goodput در ثانیه ۵ برابر ۱۵ خواهد بود.

راهنمایی: می‌توانید با کمک ackهایی که از طرف مقصد فرستاده می‌شوند این نرخ را محاسبه کنید.

- نرخ از دست رفتن بسته

توضیح: تعداد بسته‌های از دست‌رفته در واحد زمان (ثانیه) را نشان می‌دهد.

- نرخ RTT

توضیح: مقدار زمانی که از ارسال بسته تا دریافت acknowledge آن طول می‌کشد را RTT می‌گویند. نرخ RTT مشخص می‌کند در هر ثانیه آخرین مقدار RTT ثبت‌شده چه میزان بوده است (در حقیقت باید در اینجا میانگین RTTهای ثبت‌شده در ثانیه گذشته را در نظر گرفته شود اما برای سادگی آخرین RTT را تنها در نظر می‌گیریم).

به ازای هر کدام از ۲ جریان، برای تمامی معیارهای بالا یک نمودار در طول زمان برای ۳ نوع پروتکل TCP رسم نمایید. لذا خروجی حاوی ۴ نمودار خواهد بود که هر نمودار دارای ۶ خط نمودار⁷ به ازای ۲ جریان برای ۳ نوع پروتکل TCP شبکه می‌باشد. (نمودار اول حاوی خطوط معیار تغییر اندازه پنجره برای دو جریان به ازای هر ۳ نوع پروتکل TCP می‌باشد و الی آخر). فایل‌های حاوی داده‌های این نمودارها در کنار فایل‌های شبیه‌سازی باید در هنگام تحویل پروژه قرار داده شود.

گزارش کار

در گزارش کار انتظار می‌رود موارد زیر بررسی شده باشند:

- تعریف و توضیح نحوه کارکرد هر یک از روش‌های کنترل ازدحام Reno، Cubic و yeah
- خلاصه‌ای از روند شبیه‌سازی که طی کردید و نحوه اجرای کدها
- مقایسه سه الگوریتم به کمک نمودارهای خروجی
- نتیجه‌گیری کلی از مقایسه‌ها

⁷ Line chart

نحوه بارمبندی

10 نمره	درستی کدهای tcl	کد
10 نمره	درستی نمودارهای خروجی	
20 نمره	فهم درست قسمت‌های مختلف کد tcl	
20 نمره	تعریف و توضیح سه نوع TCP	گزارش
10 نمره	توضیح روند اجرا	
30 نمره	تحلیل و مقایسه درست نمودارها	

نکات پایانی

- در صورتی که سوال یا ابهامی برایتان به وجود آمد تا جای امکان آن را در فروم مطرح کنید تا دوستانتان هم بتوانند استفاده کنند.
- در صورت داشتن هرگونه پیش‌فرضی (که در صورت پروژه در نظر گرفته نشده است) در انجام شبیه‌سازی باید در مستندات به طور کامل توضیح داده شود.
- کدهای tcl، نمودارهای خروجی و گزارش کار خود را در قالب زیپی با نام A1-SID1-SID2.zip در سایت بارگذاری کنید که SID1 و SID2 شماره دانشجویی اعضای گروه است.
- برای تولید نمودارها می‌توانید از هر زبان برنامه‌نویسی و روشی استفاده کنید (برای این قسمت استفاده از زبان برنامه‌نویسی پایتون پیشنهاد می‌شود). اما روش مورد استفاده‌تان را در گزارش کار توضیح دهید.
- این لینک منبع خوبی برای تحقیقات در زمینه سه نوع TCP و پیاده‌سازی آنان در NS2 است.
- هدف از این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.