به نام خداوند هستی بخش

شبکه‌های کامپیوتری – پاییز ۱۴۰۴

پروژه شماره ۱ – فاز دوم

مهلت تحویل: ۳۰ دی ماه

---

۱. **مسأله** — طراحی یک UDP-based application با کنترل نرخ «TCP-Friendliness» (پیاده‌سازی ساده‌TFRC-like) در ns-2

توضیحات: پیاده‌سازی یک UDP که نرخ ارسالش را بر اساس حضور ترافیک TCP تنظیم کند تا « fairness » نسبت به TCP رفتار کند. مقایسه ترافیک UDP با TCP در یک لینک مشترک و ارزیابی معیارهای گذردهی، fairness وpacket loss . یعنی میخواهیم ببینیم:

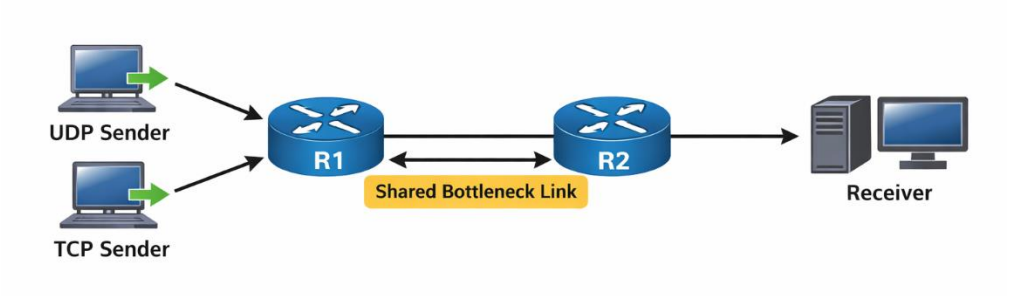
۱) UDP معمولی وقتی کنار TCP قرار می‌گیرد چه رفتاری دارد (معمولاًunfair)
۲) UDP با کنترل نرخ چطور می‌تواند شبیه TCP رفتار کند و منابع را عادلانه‌تر تقسیم کند

اهداف مدنظر برای این فاز شامل موارد زیر می باشد:

۱) پیاده‌سازی یک sender UDP که به صورت ساده نرخش را با توجه به packet loss و RTT دریافتی از receiver تنظیم می‌کند (الگوریتم ساده شبیه TFRC(TCP Friendly Rate Control): کاهش تدریجی نرخ به هنگام شناساییloss ، افزایش آهسته نرخ در absence of loss).

۲) مقایسه سناریوها: ۱) UDP (بدون کنترل)، ۲) UDP با کنترل جدید، در برابر یک یا چند جریان TCP در لینک مشترک.

۲. **توپولوژی شبکه (Shared Bottleneck)**



- ✓ لینک R1–R2 ⟵ گلوگاه (bottleneck)
- ✓ UDP و TCP روی یک لینک مشترک رقابت می‌کنند

۳. نحوه پیاده‌سازی کنترل نرخ:

- ساده‌ترین راه: در سطح application در TCL یک Application/Traffic/CBR را طوری تنظیم کنید که نرخ آن با یک تابع کنترل از سمت receiver تغییر کند؛ یا
- برای دقیق‌تر: یک Agent جدید در ns-2 (C++ + OTcl binding) — پیاده سازی کنید مثلاً Agent/UDP/TCPFriendly که متغیر send_rate دارد و هر T ثانیه بر اساس درصد پکت‌های ACKed یا مقدارloss ، نرخ را کاهش/افزایش می‌دهد.
- الگوریتم پیشنهادی کنترل نرخ (TFRC-like ساده):
  - در صورت تشخیص از دست رفتن هر بسته (Packet Loss):

نرخ ارسال به‌صورت ضربی کاهش می‌یابد:

$$rate = rate \times 0.85$$

  - در صورت سپری شدن یک RTT کامل بدون هیچ‌گونه Loss:

نرخ ارسال به‌صورت افزایشی و آهسته افزایش می‌یابد:

$$rate = rate + \alpha$$

که در آن:

$$\alpha = 0.01 \times rate$$

این مکانیزم باعث می‌شود:

- ✓ کاهش نرخ سریع و محافظه‌کارانه انجام شود (Multiplicative Decrease)
- ✓ افزایش نرخ تدریجی و سازگار با TCP باشد (Additive Increase)

۴. تحلیل و مقایسه:

- شبیه سازی را برای سه حالت باید اجرا نمایید: فقط TCP (baseline) ، TCP + UDPبدون کنترل، TCP + UDPبا کنترل. برای هر ۳ حالت معیارهای زیر استخراج و مقایسه شود:
  - Throughput هر جریان،
  - packet loss
  - fairness index (Jain's fairness)
  - و میانگین تاخیر delay
- انتظارات رفتار خروجی:
  - UDP بدون کنترل، TCPرا نابود می‌کند UDP قالبا سهم بیشتری از باند می‌گیرد، TCP starves)
  - UDP با کنترل، سهم بین UDP و TCP نزدیک‌تر به منصفانه می‌شود و Jain index بهتر می‌شود.

در نهایت به سوال زیر در بخش تحلیل نمودارهای پاسخ دهید:
"نشان دهید که الگوریتم ساده کاهش-افزایش چطور باعث می‌شود UDP از TCP-friendly بودن برخوردار شود".

نکته: در صورت داشتن هر گونه پیش فرضی(که در صورت پروژه در نظر گرفته نشده است) در انجام شبیه‌سازی باید در مستندات به طور کامل توضیح داده شود.

۱. مراجع و لینک‌ها:

https://en.wikipedia.org/wiki/TCP_congestion_control

https://en.wikipedia.org/wiki/TCP_Friendly_Rate_Control

https://www.educative.io/answers/what-is-tcp-fairness

https://en.wikipedia.org/wiki/Fairness_measure

https://www.researchgate.net/figure/Jains-Fairness-Index-vs-bottleneck-capacity-for-TCP-New-Reno-TCP-Vegas-TCP-Cubic_fig10_225675247

در صورت داشتن هرگونه سوال و ابهامی به me.esmaeili@ut.ac.ir ایمیل بزنید و یا از طریق نماینده کلاس پیگیری نمایید.

موفق باشید