به نام خدا

درس: شبکه مخابرات داده‌ها

استاد: دکتر محمدرضا پاکروان

گزارش پروژه شماره ۵

سیّدمحمّدامین منصوری طهرانی

۹۴۱۰۵۱۷۴

۱. در ابتدا به صورت خلاصه TCP Tahoe را شرح می‌دهیم. در این روش طول congestion window به نحوی هوشمندانه متناسب با ازدحام تنظیم می‌شود. به این صورت که در ابتدا توسط فرآیندی که slow start نام دارد(که آرام هم نیست و به صورت نمایی زیاد می‌شود!) طول این پنجره با گرفتن ack برای هر segment از داده دو برابر می‌شود. سپس به حد این مرحله که برسد(slow start threshold) به صورت خطی زیاد می‌شود(که به آن congestion avoidance نیز می‌گویند.). در ادامه این فرآیند اگر packet loss/time-out اتفاق بیفتد پروتکل TCP فرض می‌کند که این گم شدن بسته به دلیل ازدحام بوده است و پکت گم شده را بازارسال می‌کند(fast retransmit) طول congestion window را برابر یک می‌گذارد و پروسه slow start را از اول انجام می‌دهد. هم‌چنین حدی که بعد از آن congestion avoidance یا افزایش طول پنجره به صورت خطی می‌باشد را نیز به نصف طول پنجره‌ای که در آن packet loss/time-out رخ داده ست می‌کند. مشکل این روش این است که pipe برای مدتی خالی می‌ماند و مدت زمان رسیدن به بهره‌وری بهینه از شبکه طی پروسه slow start نسبتاً طولانی است. در TCP Reno این وضعیت بهبود بخشیده شده است.

ایده این است که اگر یک پکت گم شده پس این بسته از pipe خارج شده و لزومی ندارد ما با شدت نرخ ارسال(congestion window) را تا ۱ کم کنیم و می‌توانیم هنوز بسته‌هایی را وارد شبکه کنیم پس اگر طول congestion window را نصف هم بکنیم مشکلی پیش نمی‌آید و از منابع به طرز بهینه‌تری استفاده می‌شود. در این روش در واقع slow start حذف می‌شود و تنها در شروع ارتباط و یا زمانی که time-out رخ می‌دهد(بسته‌ها یا دچار time-out می‌شوند یا duplicate ack) شاهد slow start خواهیم بود. در زیر به توضیح بیشتر این پروتکل که از روش fast recovery استفاده می‌کند می‌پردازیم.

Fast Recovery: پس از دریافت سه duplicate ack پشت سر هم عملیات زیر انجام می‌شود:

۱. ssthresh به نصف طول congestion window فعلی کاهش می‌یابد.

۲. segment گم شده retransmit می‌شود.

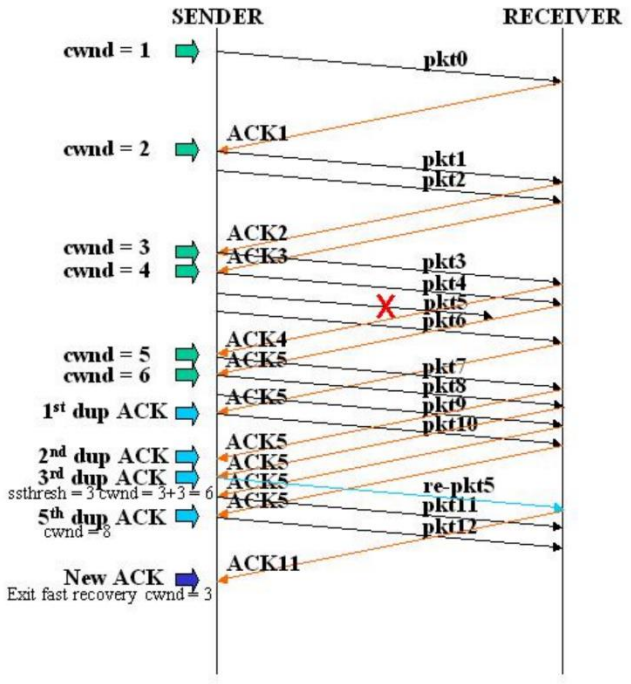
۳. طول پنجره congestion برابر ssthresh+3 تنظیم شود(در نظر گرفتن ۳ بسته‌ای که از شبکه خارج شده‌اند و ack بسته نیامده را فرستاده‌اند.)

۴. با اضافه شدن duplicate ack های بیشتر طول پنجره congestion یکی اضافه شود.(بسته‌ها از شبکه خارج می‌شوند پس جا برای بسته‌های جدید هست.)

۵. اگر ack جدید دریافت شد(که segment هایی که در حد فاصل ارسال شدن بسته گم شده و رسیدن اولین duplicate ack را acknowledge کند) از فاز fast recovery خارج می‌شویم و طول congestion window را برابر ssthresh گذاشته و بر اساس congestion avoidance به صورت خطی طول این پنجره را زیاد می‌کنیم.

* نکته مهم: این الگوریتم برای وقتی مناسب است که فقط یک بسته در هر پنجره گم شود و در غیر این‌صورت مزیت چندانی نسبت به Tahoe ندارد.

در ادامه یک مثال برای توضیح بیشتر آورده شده است.



پس از گرفتن سومین dup ack بسته ۵، $ssthresh = 6/2 = 3$ تنظیم می‌شود. cwnd نیز ۶ گذاشته می‌شود (طبق الگوریتم). سپس بسته ۵ بلافاصله بازارسال می‌شود و به خاطر الگوریتم fast recovery بعد از گرفتن dup ack های بعدی به دلیل خالی شدن شبکه از آن بسته‌ها طول پنجره congestion را زیاد می‌کند و به فرستنده اجازه می‌دهد بسته‌های ۱۱ و ۱۲ را نیز بفرستد. سپس با فرستاده شدن ack جدید بسته‌های قبلی به صورت piggybacked شده از فاز fast recovery خارج می‌شویم و cwnd را برابر ssthresh قرار داده و ادامه مسیر با congestion avoidance پیگیری می‌شود.

در شکل زیر مقایسه عملکرد دو پروتکل قابل مشاهده است که به وضوح در پروتکل Reno استفاده بهتری از منابع می‌شود.(مرحله slow start حذف شده‌است.)