دانشگاه صنعتی شاهرود

شبکه‌های کامپیوتری

تمرین سوم

دکتر محسن رضـــوانی

مصطفی فضلی - ۹۸۲۲۸۰۳

۱۳ خرداد ماه ۱۴۰۱

در مدیریت کانال نقطه به نقطه براي انتقال اطلاعات از سه روش زیر استفاده می‌شود:

- Selective Repeat
- GBN
- Stop & Wait

هر یک از موارد زیر را براي سه کانال فوق با هم مقایسه کنید:

A. مقدار بافر مورد نیاز در سمت گیرنده
B. بازدهی کانال
C. استفاده از تایمر

سوال 1

A. مقدار بافر مورد نیاز در سمت گیرنده

- در روش Selective Repeat گیرنده بسته‌هایی که کامل و سالم دریافت می‌شوند را به صورت منفرد تصدیق می‌کند. بسته‌ها بافر شده تا سپس به ترتیب بیان شده تحویل لایه کاربرد داده شوند.
- در روش GBN گیرنده به اندازه پکت های ارسالی فرستنده می تواند بافری برای آن در نظر بگیرید لذا فرستنده تعداد پکت ها را ارسال و تعیین می کند و در سمت گیرنده درصورت دریافت یا عدم دریافت همان مقدار معلوم شده را بافر می کند.
- در روش stop & wait مقدار بافر سمت گیرنده به اندازه یک پکت است: فرستنده یک پکت می فرستد، دو حالت پیش می آید:

1- پکت نمی رسد که دوباره باید بفرستد یا

2- گیرنده پاسخ مناسب را ack می فرستد.

Buffer Size: Stop & wait < GBN ≈ Selective Repeat

B. بازدهی کانال

- در GBN با توجه به اینکه پکت های ارسال شده توسط فرستنده به اندازه پنجره توضیح داده شده می‌باشد، بدین معنا که ارسال همزمانی چندین پکت را داریم و بازدهی کانال نسبت به روش stop & wait بسیار بیشتر خواهد بود.
- در روش Selective Repeat با توجه به ارسال شدن همزمان پکت ها توسط فرستنده، همانند روش GBN عمل کرده با این تفاوت که ارسال پکت ها می تواند خارج از نوبت صورت گیرد لذا بازدهی این روش از روش GBN نیز بیشتر است.
- در stop & wait از آنجاییکه بعد از ارسال پکت توسط فرستنده باید برای پاسخ ack توسط گیرنده منتظر بمانیم لذا بازدهی کانال بسیار پایین می باشد.

Efficiency of Channel: stop & wait < GBN < Selective Repeat

C. استفاده از تایمر

- در روش stop & wait تایمر برای پکت ارسالی اگر در یک مدت مشخصی پاسخ ack دریافت نشود به timeout می‌خوریم و تایمر فعال می شود تا ack برسد یا فرستنده دوباره پکت را ارسال کند.
- در GBN اگر بین بسته های دریافت شده فاصله وجود داشته باشد گیرنده بسته را ack نمی کند و فرستنده برای بسته های ack نشده قدیمی از تایمر استفاده می کند و وقتی تایمر منقضی شود فرستنده تمامی بسته های ack نشده را دوباره می فرستد.
- در روش Selective Repeat با توجه به اینکه گیرنده برای هر بسته ack جدا می فرستد لذا فرستنده برای هر بسته ackنشده یک تایمر نگه می دارد و وقتی تایمر منقضی شود فرستنده فقط بسته ack نشده را دوباره می فرستد.

Use Timer: stop & wait < GBN < Selective Repeat

نقش دو فیلد # seq و # ack در سرایند پروتکل TCP را با یک مثال شرح دهید. آیا این فیلدها در پروتکل UDP هم استفاده می‌شوند؟ چرا؟

سوال 2

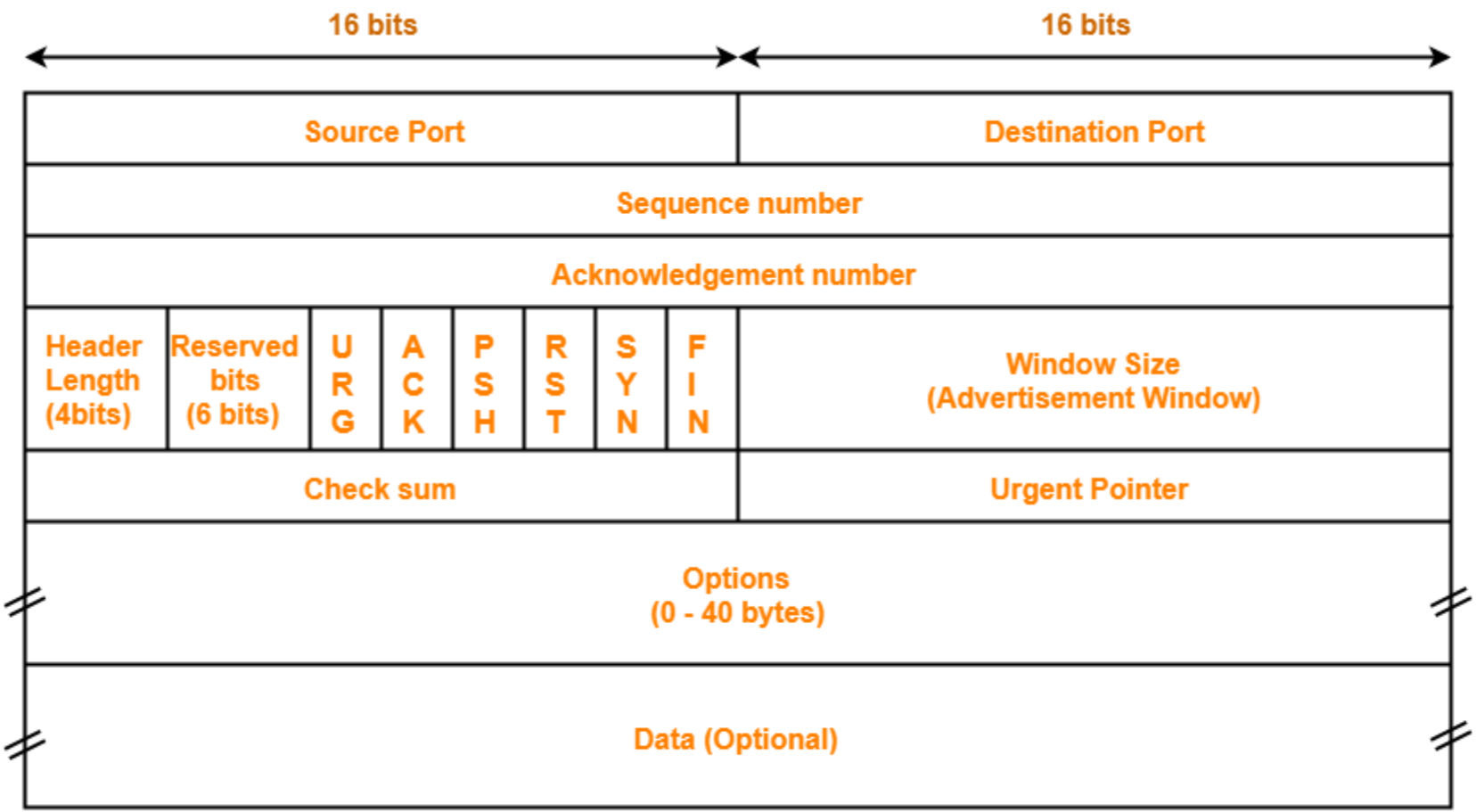
فیلد #seq مخفف شده sequence است که همانطور از نامش پیداست، ترتیب(پکت های ارسالی) را نشان می‌دهد.
برای مثال از طریق پروتکل TCP میان میزبان A و B ارتباطی برقرار می‌شود که در آن میزبان A داده ای را برای میزبان B ارسال می‌کند. این پکت های ارسالی در این جریان ارتباطی برای دریافت و بازسازی پکت ها در سمت مقصد شماره گزاری می‌شوند، بدین صورت که بسته به اندازه MSS که فرض کنیم اندازه آن 1000 بیت در نظر گرفته شود و شماره نخستین بایت از جریان داده را هم صفر بگیریم، اولین پکت با عدد 0 و بعدی 1000 و بعدی 2000 و الی آخر شماره‌گذاری خواهند شد. این فیلد یک عدد 32 بیتی می باشد.
فیلد #ack که مخفف Acknowledgment که همانطور از معنای آن پیداست، تصدیق کننده است، یک عدد 32 بیتی است که از سمت گیرنده برای فرستنده ارسال می شود و نشان دهنده دریافت شدن آن توسط گیزنده برای فرستنده است.

برای توضیح با مثال، بدین صورت که باز هم میزبان A داده ای را برای میزبان B فرستاده است و میزبان B پس از دریافت کامل و سالم داده ارسال شده، یک پیام که نشان دهنده دریافت آن است برای میزبان A میفرستند. این پیام تایید به همراه #seq برای فرستنده ارسال می شود.

این فیلد ها در پروتکل UDP وجود ندارند زیرا که در این پروتکل unreliable است، بدین معنا که منتظر ماندن برای پکت های در حال ارسال و همچنین گم شدن یا از بین رفتن پکت در طول مسیر اهمیتی ندارد.
به طور کلی segment شدن پیام برای بررسی کردن و همچنین شماره گزاری seq برای هر پیام در این پروتکل وجود ندارد و به همین دلیل است که سرعت بالا و قابلیت اطمینان پایین دارد.

سوال 3

ساختار سگمنت TCP را با شکل نشان داده و هر بخش را توضیح دهید؟


16 bits
16 bits
Source Port
Destination Port
Sequence number
Acknowledgement number
Header Length (4bits)
Reserved bits (6 bits)
URG
ACK
PSH
RST
SYN
FIN
Window Size (Advertisement Window)
Check sum
Urgent Pointer
Options (0 - 40 bytes)
Data (Optional)

TCP Header

بخش **Source port** دامنه آدرس پورت های مبدا را نمایش می دهد که مقداری 16 بیتی است.
بخش **Destination port** دامنه آدرس پورت های مقصد را نمایش می دهد که مقداری 16 بیتی است.
این دو بخش ذکر شده عملیات تسهیم داده که دو بخش Multiplexing و Demultiplexing را از سوی لایه های بالاتر و همچنین به آن ها مدیریت می کنند.

Sequence Number مقداری 32 بیتی است که ترتیب پکت ها(در قسمت قبلی توضیح داده شد) را نمایش می دهد.
Acknowledgment Number مقداری 32 بیتی است که تصدیق کننده پیام ها(در قسمت قبلی توضیح داده شد) را نمایش می دهد.
این دو قسمت با همدیگر برای انتقال مطمئن و ترتیب انتقال پیام های استفاده می شوند.

receive window: دامنه پنجره دریافت، طول آن 16 بیت است و برای کنترل جریان استفاده می شود. این بخش تعداد بایت های قابل قبول جهت دریافت را از سمت گیرنده مشخص می کند.

flag field: دارای 6 بیت است و از بیت ACK جهت اعتبارسنجی ارزش حمل شده در محدوده تایید استفاده می شود. بیت های FIN, SYN, RSTجهت برقراری و نیز قطع ارتباط مورد استفاده قرار می گیرد.

مشخص کردن وجود اطلاعات U: urgent

بررسی وجود A:ack

مشخص کردن ارسال اطلاعات P:

برقراری و قطع ارتباط بین دو host RSF:

هنگامی که بیت PSHبرابر یک می شود گیرنده را مجبور به تحویل بی درنگ داده (اولیت بالاتر) به لایه بالاتر می کند. بیت URGکه از آن برای مشخص کردن داده ای استفاده می شود که موجودیتی در لایه بالاتر در سمت ارسال به آن برچسب فوری اختصاص داده است.

header length: سایز هدر کخ 4 بیت است. هدر TCP می تواند طول متغیری را به دلیل وجود محدوده قابل گزینش در TCPاختیار نماید.

options: با طولی متغیر و اختیاری که فرستنده و گیرنده به تعیین حداکثر اندازه قطعه MSSو یا مقیاس پنجره برای کاربرد شبکه های پرسرعت می پردازند.

Internet checksum, Urgent data pointer: آدرس 16 بیتی که آخرین بایت این داده اضطراری urgent data pointer را مشخص می کند به هنگام وجود داده اضطراری لایه بالاتر را آگاه نموده و در سپس checksum آن جهت آشکار سازی بیت های خطا در یک بسته ارسال شده کاربرد دارد.

فرض کنید دو میزبان A و B در حال برقراري ارتباط و دریافت اطلاعات از سرور C هستند. تفاوت TCP و UDP در demultiplexing و multiplexing چگونه است؟

سوال 4

گرفتن داده ها از سمت فرستنده، قرداردن هدر و ارسال آن‌ها به عنوان یک پیام به گیرنده مورد نظر، مالتی پلکس نامیده می‌شود.
همچنین برای تحویل بخش های دریافتی در سمت گیرنده فرآیندهای لایه برنامه demultiplexing را برای گرفتن پیام انجام می‌دهند.
در پروتکل TCP، 4 بخش IP مبدا و مقصد، پورت مبدا و مقصد تعیین و شناسایی می شوند ولی در پروتکل UDP تنها دو بخش اطلاعات IP و پورت مقصد تعیین و شناسایی می شوند.
این تفاوت میان این دو پروتکل در واقع مفهوم شئی گرایی برای پروتکل TCP و ConnectionLess بودن برای UDP را نشان می‌دهند.

یک پروتکل قابل اطمینان را در نظر بگیرید که فقط از Negative Acknowledgement استفاده می‌کند. فرض کنید که فرستنده با فواصل زمانی طولامی داده‌هایی با مقدار کم را ارسال می‌کند.

آیا استفاده از فقط NAK بهتر است یا فقط ACK؟ شرح دهید.

حال فرض کنید فرستنده داده‌هاي زيادي براي ارسال دارد و احتمال از دست دادن بسته‌ها وجود دارد. در این صورت استفاده از فقط NAK بهتر است یا فقط ACK؟ شرح دهید.

سوال 5

الف) استفاده از ack بهتر است زیرا که دریافت ack که تصدیق رسیدن پیام است، هنگامی که پکت های کمی داشته باشیم از اینکه پیام نرسیدن آن ها را یکی یکی ارسال کنیم و منتظر آن بمانیم، بهینه تر عمل می کند و زمان کمتری صرف کرده و بازدهی را بالاتر می برد.

در این پروتکل، تنها زمانی متوجه از بین رفتن بسته در ارتباط خواهیم شد که بسته بعدی دریافت شود. بدین معنا که تا بسته n+1 دریافت نشود نخواهیم فهمید که بسته n دریافت شده است یا خیر.

همچنین اگر تاخیر زیادی مابین ارسال بسته n و n+1 داشته باشیم، زمان زیادی طول می کشد که n توسط پروتکل فقط NACK بازیابی شود.

ب) چون اغلب داده ها زیاد است و احتمال اینکه بسته ها خراب شده یا ارسال نشوند زیاد است اگر از Ackاستفاده کنیم بهتر است زیرا از درستی بسته های ارسال شده مطمئن می شویم و بسته هایی که خراب شده اند یا نرسیده اند را دوباره میفرستیم.

همچنین ممکن است به خطاهای نادر بخوریم که می توانیم از NACK استفاده کنیم ولی پروتکل تنها NACK این امکان را به ما نمی دهد که ACK را بفرستیم و صحت ارسال را در جا متوجه شویم.

فرض کنید 5 تا SampleRTT اندازه گرفته شده برابر 106،120،140،90،115 میلی ثانیه باشند. EstimatedRTT را بعد از هر sampleمحاسبه کنید. a=0.125و اولین تخمین را هم 100 در نظر بگیرید. همچنین DevRTT را بعد از هر نمونه محاسبه کنید با B=0.25 و مقدار اولیه 5ms و نهایتا TimeoutIntervalرا براي هر کدام محاسبه کنید.

سوال 6

فرمول محاسبه:

EstimatedRTT = (x)SampleRTT +(1−x)EstimatedRTT

DevRTT = y |SampleRTT −EstimatedRTT| +(1−y)DevRTT

TimeoutInterval = EstimatedRTT +4*DevRTT

90ms :sampleRTT

EstimatedRTT = 0.125*90 +0.875*107.76 = 105.54ms

DevRTT = 0.25∗|90 −105.54|+0.75*14.06 = 14.42ms

TimeoutInterval = 105.54 +4*14.42 = 163.22ms

106ms :sampleRTT

EstimatedRTT = (0.125*106) + (0.875*100) = 100.75ms

DevRTT = (0.25∗(|106 −100.75|)) + (0.75*5) = 5.06ms

TimeoutInterval = 100.75 + (4 *5.06) = 120.99ms

115ms :sampleRTT

EstimatedRTT = 0.125*115 +0.875*105.54 = 106.71ms

DevRTT = 0.25∗|115−106.71| +0.75*14.42 = 12.88ms

TimeoutInterval = 106.71+4*12.88 = 158.23ms

120ms :sampleRTT

EstimatedRTT = (0.125*120) + (0.875*100.75) = 103.15ms

DevRTT = (0.25∗(|120−103.15))| + (0.75*5.06) = 8ms

TimeoutInterval = 103.15 + (4*8) = 135.15ms

140ms :sampleRTT

EstimatedRTT = 0.125*140 +0.875*103.15 = 107.76ms

DevRTT = 0.25∗|140−107.76 |+0.75*8 = 14.06ms

TimeoutInterval = 107.76 +4*14.06 = 164ms

فرض کنید یک فایل بزرگ با حجم L بایت از میزبان Aبه Bارسال می‌شود.
فرض کنید مقدار MSSبرابر 536 بایت باشد.
بزرگترین مقدار Lچقدر باشد تا sequence numberتمام نشوند؟ با فرض اینکه در TCPفیلد sequence number 32بیتی می باشد.
با استفاده از حجمی که در قسمت قبل به دست آوردید فرض کنید درکل 66 بایت هدر به هر سگمنت اضافه می شود وبر لینکی با نرخ 200 Mbps فرستاده می شود. با صرف نظر از کنترل جریان و تراکم چقدر طول می کشد تا این فایل انتقال یابد؟

سوال 7

Sequence number با توجه به هر تکه(segment) افزایش نمی‌یابد و اندازه MSS در اینجا کاربردی ندارد.، حداکثر اندازه فایل به سادگی تعداد بایت های قابل نمایش با 32 بیت یعنی 2^32 بیت یا 4.294 گیگابیت است.

MSS = 536 byte

R = 200 Mbps

Header Size = 66 byte

segment = $\left\lceil \frac{2^{32}}{536} \right\rceil = 8{,}012.999$ bytes

+ 66 bytes of header

2^32 + 528,857,934 = 4.824 $*$ 109 $bytes$

Time to send = $\frac{42824 * 10^9}{200 * 10^6} = \mathbf{24.12}\ sec$

24.12ثانیه طول می کشد تا ارسال شود.

فرض کنید 10 بسته به سمت گیرنده ارسال می‌شود. اگر بسته سوم و هفتم در اولین دفعه ارسالشان گم شوند و به مقصد نرسند، با فرض اندازه پنجره ارسالی 5، براي سه حالت SR، GBNو TCP، تعداد کل دفعات ارسال بسته و نیز تعداد کل ACKها را محاسبه نمایید. فرض کنید مدت زمان Time outبسیار بیشتر از زمان ارسال 10 بسته باشد. با این توضیح، به نظر شما در کدام حالت سریعتر هر 10 بسته به مقصد می‌رسند. چرا؟

سوال 8

GBN

دفعات ارسال	15
دفعات ACK	13

Selective Repeat

دفعات ارسال	12
دفعات ACK	10

TCP

دفعات ارسال	12
دفعات ACK	10

سریع ترین حالت روش TCP می باشد زیرا در صورت وجود تصدیق تکراری (Ack duplicate) بلافاصله پکت را دوباره ارسال میکند.

Finish