Лекція 6.

Тема. Мережеве забезпечення глобальних систем обробки та обміну інформацією.

Розділ. Структура протокола транспортного рівня на прикладі протокола TCP.

План лекції:
4.1. Структура протоколу TCP.
4.2. Додатки, що використовують протокол TCP.
4.3. Встановлення та розрив TCP з’єднання.
4.4. Протокол встановлення з'єднання.
4.5. Протокол розриву з'єднання.

4.1. Структура протоколу TCP.

Протокол TCP (англ. Transmission Control Protocol) належить до транспортного рівня моделі OSI надає додаткам заснований на з'єднанні надійний сервіс потоку байтів. Термін "заснований на з'єднанні" (connection-oriented) означає, що два додатки, що використовують TCP (як правило, це клієнт і сервер), повинні встановити TCP з'єднання один з одним, після чого у них з'являється можливість обмінюватися даними. Завжди існують дві кінцеві точки, які спілкуються один з одним за допомогою TCP з'єднання. Широкомовна розсилка та групова розсилка не мають відношення до TCP. Протокол TCP інкапсулюється в протокол IP, тобто сегменти TCP вкладаються в протокол IP в якості даних.

<----------IP датаграма---------->		
	<----------TCP сегмент---------->	
Заголовок IP	Заголовок TCP	Дані TCP
20 байт	20 байт	

Формат заголовка TCP

<table>
<tr><td colspan="5">Номер порту відправника, 16 біт</td><td colspan="4">Номер порту отримувача, 16 біт</td></tr>
<tr><td colspan="9">Номер сегменту, 32 біт</td></tr>
<tr><td colspan="9">Номер підтвердження (Номер ACK), 32 біт</td></tr>
<tr><td>Довжина заголовка, 4 біт</td><td>Зарезервовано, 6 біт</td><td>URG</td><td>ACK</td><td>PSH</td><td>PST</td><td>SYN</td><td>FIN</td><td>Розмір вікна, 16 біт</td></tr>
<tr><td colspan="5">Контрольна сума, 16 біт</td><td colspan="4">Покажчик терміновості, 16 біт</td></tr>
<tr><td colspan="9">Опції, якщо є……</td></tr>
<tr><td colspan="9">Дані, якщо є …..</td></tr>
</table>

Кожен TCP сегмент містить номер порту (англ. port number) відправника і отримувача, з допомогою яких ідентифікуються додатки, що відправляють та отримують дані. Ці два значення разом з IP адресою відправника і отримувача в IP заголовку унікально ідентифікують кожне з'єднання.

Комбінація IP адреси і номера порту називається сокетом (англ. socket). Пара сокетів (містить IP адресу клієнта, номер порту клієнта, IP адресу сервера і номер порту сервера) описує дві кінцеві точки, які унікально ідентифікують кожне TCP з'єднання.

Номер сегменту (англ. sequence number) ідентифікує байт в потоці даних від відправляючого TCP сокета до приймаючого. Якщо ми уявимо потік байтів, що тече в одному напрямку між двома додатками, TCP нумерує кожен байт номером послідовності і вказує в полі Номер сегменту номер першого байту фрагмента. Номер послідовності являє собою 32-бітне беззнакове число, яке переходить через 0, по досягненню значення $2^{32}-1$.

При встановленні нового з'єднання, зводиться прапорець SYN. Поле номера послідовності (sequence number field) містить вихідний номер послідовності (ISN - initial sequence number), який вибирається хостом для даного з'єднання. Номер послідовності першого байта даних, який надсилається цим хостом, буде дорівнює ISN плюс один, тому що прапор SYN займає собою номер послідовності.

Так як кожен байт, який бере участь в обміні, пронумерований, номер підтвердження (acknowledgment number) це наступний номер послідовності, який очікує отримати відправник підтвердження. Це номер послідовності плюс 1 останнього успішно прийнятого байта даних. Це поле приймається в розгляд, тільки якщо прапор ACK зведено.

Відправка ACK не варто нічого (це означає, що на підтвердження не витрачається сегмент), тому що 32-бітне поле номера підтвердження завжди є частиною заголовка, так само як і прапор ACK. Коли з'єднання встановлено прапор ACK завжди зведено.

TCP надає для прикладного рівня повнодуплексний сервіс. Це означає, що дані можуть передаватися в кожному напрямку незалежно від іншого напрямку. Однак, на кожному кінці з'єднання необхідно відстежувати номер послідовності даних, переданих в кожному напрямку.

TCP є протоколом із змінним вікном без селективних або негативних підтверджень. В TCP немає селективних підтверджень, тому що номер підтвердження в TCP заголовку означає, що відправник успішно прийняв, всі байти за винятком цього байта. Таким чином, в даний час не існує можливості підтвердити окремо обрану частину потоку даних. Наприклад, якщо байти 1-1024 прийняті нормально, а наступний сегмент містить байти з номерами 2049-3072 (тобто деякі байти відсутні), приймач не підтверджує цей новий сегмент, а це послає ACK з номером підтвердження 1025. Також немає ніякого сенсу посилати негативні підтвердження на сегмент. Наприклад, якщо сегмент з байтами 1025-2048 прибув, проте була визначена помилка в контрольній сумі, приймач посилає ACK з номером підтвердження рівним 1025.

Довжина заголовка (header length) містить довжину заголовка в 32-бітових словах. Це пояснюється тим, що довжина поля опцій змінна. З 4-бітовим полем у TCP є обмеження на довжину заголовка в 60 байт. Без опцій, проте, стандартний розмір становить 20 байт.

В TCP заголовку існують 6 прапорових бітів. Один або декілька з них можуть бути встановлені в одиницю одночасно.

URG - Покажчик терміновості (urgent pointer)

ACK - якщо встановлений, то Номер підтвердження необхідно прийняти в розгляд (Acknowledgment).

PSH - якщо встановлений, то Одержувач повинен передати ці дані додатку якомога швидше (PUSH).

RST - якщо встановлений, то необхідно скинути з'єднання (Reset).

SYN - Синхронізуючий номер послідовності для встановлення з'єднання.

FIN - якщо встановлений, то Відправник закінчує посилку даних.

Контроль потоку даних TCP здійснюється на обох кінцях з використанням розміру вікна (window size). Це кількість байт, що починається з вказаного в полі номера підтвердження, яке додаток збирається прийняти. Це 16-бітове поле обмежує розмір вікна в 65535 байт. Проте Опція масштабування вікна дозволяє змінювати це значення, при цьому можуть бути використані вікна більшого розміру.

Контрольна сума (checksum) охоплює собою весь TCP сегмент: TCP заголовок і TCP дані. Це обов'язкове поле, яке має бути розраховане і збережено відправником, а потім перевірено отримувачем. Контрольна сума TCP розраховується з використанням псевдозаголовка.

Покажчик терміновості (urgent pointer) враховується тільки в тому випадку, якщо встановлено прапор URG. Цей покажчик є позитивним зміщенням, яке повинно бути додано до поля номера послідовності сегмента, щоб отримати номер послідовності останнього байта термінових даних.

Режим терміновості TCP це спосіб, за допомогою якого відправник передає термінові дані на віддалений кінець.

Найбільш загальне поле опцій - це опція максимального розміру сегмента (MSS - maximum segment size). На кожному кінці з'єднання ця опція зазвичай вказується в першому сегменті, з якого починається обмін (сегмент з встановленим прапором SYN, який використовується для встановлення з'єднання). Вона вказує на максимальний розмір сегмента, який може бути прийнятий відправником.

Поле даних в TCP сегменті необов'язковий. Коли встановлюється з'єднання або коли з'єднання розривається, сегменти містять тільки TCP заголовки з можливими опціями. Заголовок без даних також використовується, щоб підтвердити прийняті дані, якщо в цьому напрямку не треба передавати дані. Також існує декілька випадків тайм-аутів, коли сегмент може бути відправлений без даних.

Таким чином TCP надає надійний, орієнтований на з'єднання, і орієнтований на потік байтів сервіс транспортного рівня.

TCP будує пакети з користувацьких даних, упаковуючи їх в сегменти, встановлює тайм-аути в той момент, коли він посилає дані, підтверджує прийняті дані з віддаленого кінця, сортує дані, які прийшли хаотично, відкидає продубльовані дані, здійснює контроль потоку даних, розраховує і перевіряє контрольну суму в кінцевих точках передачі.

4.2. Додатки, що використовують протокол TCP.

TCP використовується безліччю популярних додатків, таких Telnet, Rlogin, FTP, електронна пошта (SMTP) тощо.

4.3. Встановлення та розрив TCP з’єднання.

Нехай при встановленні і розриві TCP з'єднання на системі svr4 виконану наступну команду:

```
svr4% telnet bsdi discard
Trying 192.82.148.3 ...
Connected to bsdi.
Escape character is '^]'.
^] Вводимо Control + праву квадратну дужку,
telnet> quit щоб Telnet клієнт розірвав з'єднання
Connection closed.
```

Команда telnet встановлює TCP з'єднання з хостом bsdi на порт, відповідний discard сервісу.

Нижче наведені сегменти, згенеровані програмою tcpdump для команди telnet.

```
1 0.0 svr4.1037> bsdi.discard: S 1415531521:1415531521 (0)
                                win 4096 <mss 1024>
2 0.002402 (0.0024) bsdi.discard> svr4.1037: S 1823083521:1823083521 (0)
                                ack 1415531522 win 4096
                                <mss 1024>
```

3 0.007224 (0.0048) svr4.1037> bsdi.discard:. ack 1823083522 win 4096
4 4.155441 (4.1482) svr4.1037> bsdi.discard: F 1415531522:1415531522 (0)
ack 1823083522 win 4096
5 4.156747 (0.0013) bsdi.discard> svr4.1037:. ack 1415531523 win 4096
6 4.158144 (0.0014) bsdi.discard> svr4.1037: F 1823083522:1823083522 (0)
ack 1415531523 win 4096
7 4.180662 (0.0225) svr4.1037> bsdi.discard:. ack 1823083523 win 4096

Ці сім TCP сегментів містять тільки TCP заголовки. Обмін даними не здійснювався.

Для TCP сегментів кожна вихідна рядок починається з
source> destination: flags (джерело> призначення: прапори),

де прапори (flags) являють собою чотири з шести прапорових бітів TCP заголовка, а саме

прапор	3-символьне скорочення	Опис
S	SYN	синхронізуючий номер сегменту
F	FIN	Відправник закінчив передачу
R	RST	Скидання з’єднання
P	PST	Іідправляти дані приймаючому процесу настільки швидко, наскільки це можливо
.	Жоден не встановлений	Жоден з чотирьох прапорів не встановлений

У даному прикладі ми бачимо прапори S, F і точку. Два інших біта прапорів в TCP заголовку - ACK і URG - надруковані командою tcpdump.

В одному сегменті може бути присутнім більше ніж один з чотирьох прапоровим бітів, проте, зазвичай зведений буває тільки один прапор.

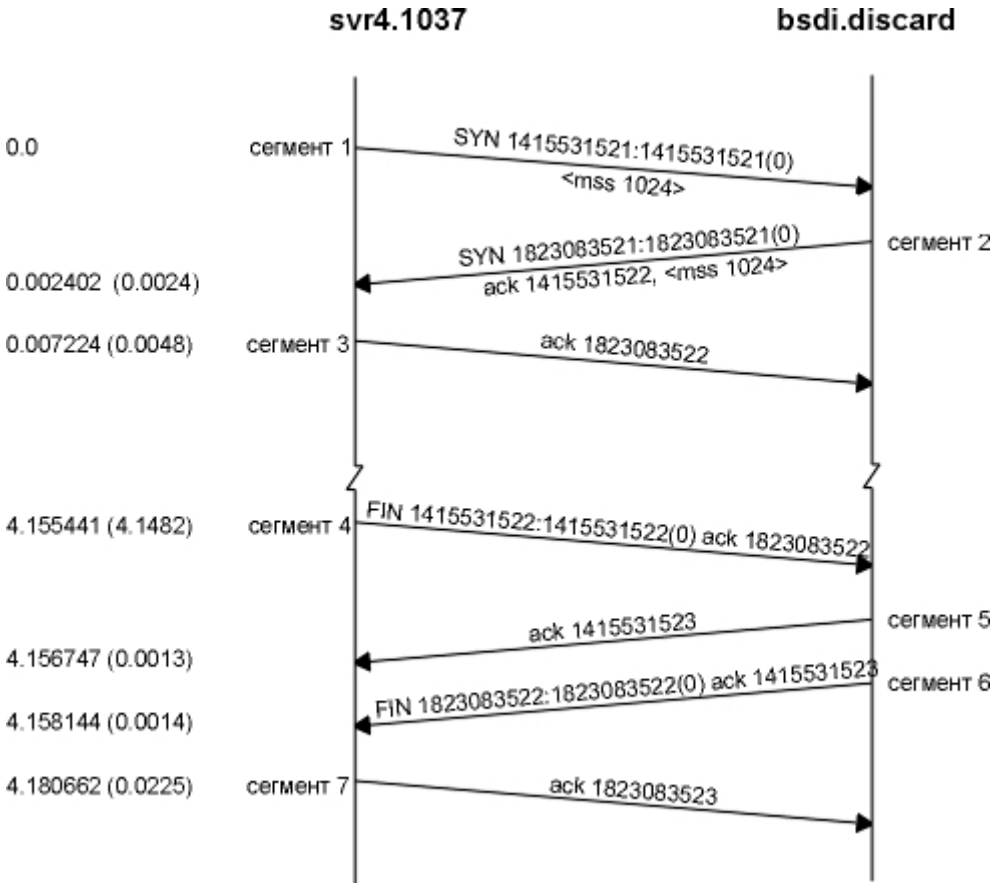
У рядку 1 поле 1415531521:1415531521 (0) означає, що номер послідовності пакета дорівнює 1415531521, а кількість байт даних в сегменті дорівнює 0. Команда tcpdump друкує початковий номер послідовності, двокрапка, передбачуваний заключний номер послідовності і потім в дужках кількість байт даних. При цьому існує можливість переглянути передбачуваний остаточний номер послідовності, коли кількість байтів більше ніж 0. У полі з'являється (1), якщо сегмент містить один або кілька байт даних користувача, або (2), якщо зведений прапор SYN, FIN або RST. У рядках 1, 2, 4 і 6 це поле з'являється, тому що зведений прапорцевий біт, а обмін даними не проводився.

У рядку 2 поле ack 1415531522 містить номер підтвердження. Воно друкується тільки в тому випадку, якщо прапор ACK зведено. Поле win 4096 в кожному рядку виводу показує розмір вікна, який був оголошений відправником. У цьому прикладі, де не здійснювався обмін даними, розмір вікна залишався незмінним і використовувалася величина за замовчуванням - 4096.

І останнє поле <mss 1024> показує максимальний розмір сегмента (MSS - maximum segment size), опція, яку встановлює відправник. Відправник не хоче отримувати TCP сегменти більше ніж це значення. Це робиться зазвичай для того, щоб уникнути фрагментації.

4.4. Протокол встановлення з'єднання..

На Мал. 6.1. показана часова діаграма, що відповідає цьому обміну пакетами. На цьому малюнку показано, яка сторона відправляє пакети. У цій тимчасовій діаграмі видалено значення розміру вікна, так як це не суттєво для нашого обговорення.



Мал. 6.1. Протокол встановлення з'єднання

Для встановлення з’єднання запитуюча сторона (яка, як правило, називається клієнт) відправляє SYN сегмент, вказуючи номер порту сервера, до якого клієнт хоче приєднатися, і вихідний номер послідовності клієнта (в даному прикладі ISN, 1415531521). Це сегмент номер 1.

Сервер відповідає своїм сегментом SYN, що містить вихідний номер послідовності сервера (сегмент 2). Сервер також підтверджує прихід SYN клієнта з використанням ACK (ISN клієнта плюс один). На SYN використовується один номер послідовності.

Клієнт повинен підтвердити прихід SYN від сервера з використанням ACK (ISN сервера плюс один, сегмент 3).

Цих трьох сегментів достатньо для встановлення з'єднання. Часто це називається триразовим рукостисканням (three-way handshake).

Вважається, що сторона, яка посилає перший SYN, активізує з'єднання (активне відкриття). Інша сторона, яка отримує перший SYN і відправляє наступний SYN, приймає пасивну участь у відкритті з'єднання (пасивне відкриття).

Коли кожна сторона відправила свій SYN, щоб встановити з'єднання, вона вибирає початковий номер послідовності (ISN) для цього з'єднання. ISN повинен мінятися кожен раз, тому кожне з'єднання має свій, відмінний від інших ISN. RFC 793 [Postel 1981c] вказує, що ISN є 32-бітовим лічильником, який збільшується на одиницю кожні 4 мікросекунди. Завдяки номерам послідовностей, пакети, що затрималися в мережі і доставлені пізніше, не сприймаються як частина існуючого з'єднання.

Як вибирається номер послідовності? У 4.4BSD (і в більшості Berkeley реалізацій) при ініціалізації системи початковий номер послідовності встановлюється в 1. Подібна практика засуджується вимогою до хостів Host Requirements RFC. Потім ця величина збільшується на 64000 кожні півсекунди і повертається в значення 0 через кожні 9,5 години. (Це відповідає лічильнику, який збільшується на одиницю кожні 8 мікросекунд, а не кожні 4 мікросекунди.) Крім того, кожного разу, коли встановлюється з'єднання, ця величина збільшується на 64000.

Проміжок в 4,1 секунди між сегментами 3 і 4 відповідає часу між встановленням з'єднання і введенням команди quit для telnet, щоб розірвати з'єднання.

4.5. Протокол розриву з'єднання.

Для того щоб встановити з'єднання, необхідно 3 сегмента, а для того щоб розірвати - 4. Це пояснюється тим, що TCP з'єднання може бути в наполовину закритому стані. Так як TCP з'єднання повнодуплексне (дані можуть пересуватися в кожному напрямку незалежно від іншого напрямку), кожен напрям має бути закритий незалежно від іншого. Правило полягає в тому, що кожна сторона повинна послати FIN, коли передача даних завершена. Коли TCP приймає FIN, він повинен повідомити додаток, що віддалена сторона розриває з'єднання і припиняє передачу даних у цьому напрямку. FIN зазвичай відправляється в результаті того, що додаток було закрито.

Отримання FIN означає тільки, що в цьому напрямку припиняється рух потоку даних. TCP, що отримав FIN, може все ще посилати дані. Незважаючи

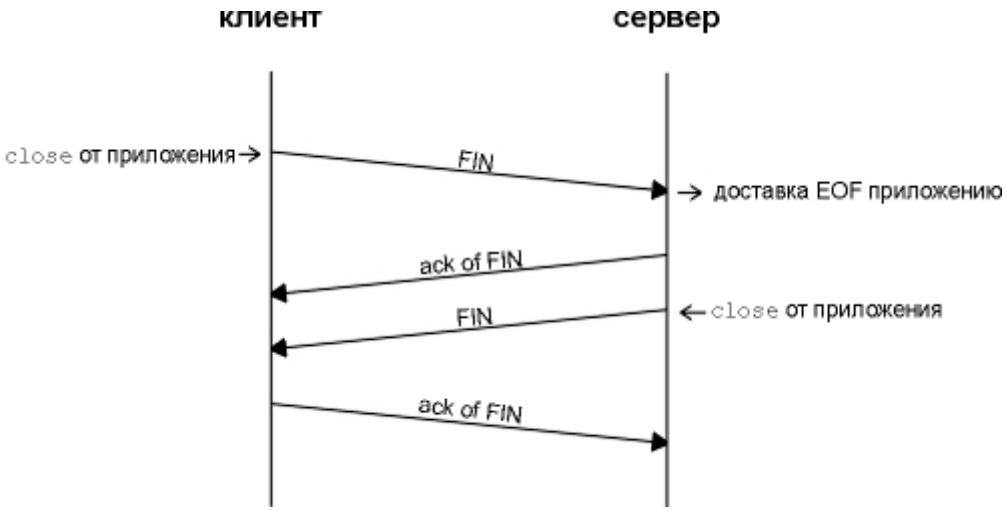
на те, що додаток все ще може посилати дані при наполовину закритому TCP з'єднанні, на практиці тільки деякі (зовсім небагато) TCP додатків використовують це. Звичайним є той сценарій, який показаний на Мал.6.1.

Можна сказати, що та сторона, яка першою закриває з'єднання (відправляємо перший FIN), здійснює активне закриття, а інша сторона (яка прийняла цей FIN) здійснює пасивне закриття. Зазвичай, одна сторона здійснює активну закриття, а інша пасивне, проте можлово, що обидві сторони можуть здійснити активне закриття.

Сегмент номер 4 на Мал.6.1. призводить до закриття з'єднання і надсилається, коли Telnet клієнт припиняє роботу. Це відбувається, коли ми вводимо quit. При цьому TCP клієнт змушений послати FIN, закриваючи потік даних від клієнта до сервера.

Коли сервер отримує FIN, він відправляє назад ACK з прийнятим номером послідовності плюс один (сегмент 5). На FIN витрачається один номер послідовності, так само як на SYN. У цей момент TCP сервер також доставляє додатку ознаку кінця файлу (end-of-file) (щоб вимкнути сервер). Потім сервер закриває своє з'єднання, що змушує його послати FIN (сегмент 6), на який клієнт повинен підтвердити (ACK), збільшивши на одиницю номер прийнятої послідовності (сегмент 7).

На Мал.6.2. показано типовий обмін сегментами при закритті з'єднання.



Мал. 6.2. Типовий обмін сегментами при закритті з'єднання.

Номери послідовності на малюнку не вказано. На цьому малюнку FIN надсилаються через те, що додатки закривають свої з'єднання, тоді як ACK для цих FIN генерується автоматично програмним забезпеченням TCP.

З'єднання зазвичай встановлюється клієнтом, тобто перший SYN рухається від клієнта до сервера. Однак будь-яка сторона може активно закрити з'єднання (відправивши перший FIN). Часто, однак, саме клієнт визначає, коли з'єднання повинно бути розірване, так як процес клієнта в основному

керується користувачем, який вводить щось подібне "quit", щоб закрити з'єднання. На Мал.6.2. можна поміняти місцями мітки, наведені вгорі малюнка, назвавши ліву сторону сервером, а праву сторону клієнтом. Однак навіть у цьому випадку все буде працювати саме так, як показано на малюнку.