noisy, and under-sampled measured images // IEEE Trans. Image Process., vol. 6, no. 12, 1997. - pp. 1646-1658.
13. *S. Borman, R.L. Stevenson* Super-resolution from image sequences - A review // Midwest Symp. Circuits and Systems, 1999. - pp.374 -378
14. *R. C. Hardie, K. J. Barnard, E. E. Armstrong* Joint map registration and high-resolution image estimation using a sequence of under-sampled images // IEEE Trans. Image Process., 1997. – pp. 1621–1633.

Поступила 28.03.2013р.

УДК 683.05

Б.В. Дурняк, О.В. Шевченко, УАД

ЗАБЕЗПЕЧЕННЯ ГАРАНТОВАНОЇ ДОСТАВКИ ДАНИХ

The article discusses the principles of network protocols UDP and TCP, as well as principles for guaranteed delivery of data.

З набуттям великої популярності у світі послуги Voice IP, потокового відео та Інтернет радіо зросли вимоги до доставки інформації через мережу Інтернет. Ці вимоги суттєво відрізняються від вимог звичайних послуг передачі інформації.

Передача мультимедійного трафіку у мережі Інтернет включає у себе роботу обладнання, у якому реалізується робота протоколів по контролю трафіка. Це стало можливим завдяки алгоритмам ущільнення та кодування інформації. Важливим чинником для більшості послуг мультимедіа є передача інформації через мережу з кінця в кінець. Але через виникнення завад і перешкод у каналах з`являється ймовірність втрати тої чи іншої інформації, що у свою чергу суттєво знижує якість надання послуг.

Впровадження нових протоколів та алгоритмів у мережі може забезпечити необхідний рівень якості послуг. Виходячи з критерію простоти, дешевизни, і швидкості реалізації цих протоколів можна здійснити на прикладному рівні.

Ієрархічна модель TCP/IP, на основі якої побудована робота мережі Інтернет, дає можливість задіяти нові можливості по ущільненню трафіка.

Реалізація більшості мультимедійних послуг «з кінця в кінець» здійснюється завдяки протоколам верхнього рівня:

RTP – протокол транспортного рівня

RTCP – протокол прикладного рівня

Допомагають контролювати мультимедійний трафік

UDP – протоколом, що не надає гарантій по якості обслуговування

TCP – протокол, що забезпечує гарантовану доставку даних.

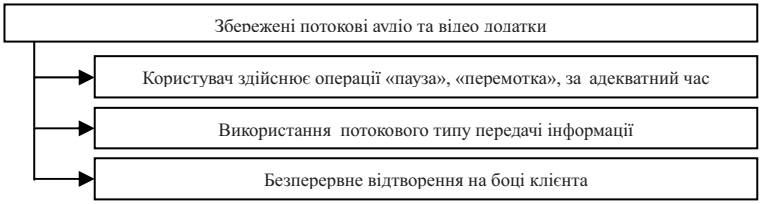

Рис.1. Блок-схема потокового відтворення інформації.

Для забезпечення реалізації послуги «з кінця в кінець» використовується протокол транспортного рівня UDP на прикладному рівні. Якщо відбувається випадкова втрата пакета, це не викликає завад у роботі протоколу.

Протокол передачі даних транспортного рівня UDP (user datagram protocol) запропонований організацією IETF. Він проектувався для створення у об`єднаній системі комп'ютерних мереж з комутацією пакетів режиму передачі датаграм клієнта. Робота протоколу UDP орієнтована на транзакції, отримання датаграм і захист від дублювання негарантовані.

TCP (transmission control protocol) – протокол гарантованої передачі даних через ненадійні канали і застосовується у тих випадках, коли необхідна гарантована доставка інформації від кореспондента до адресата. Для реалізації гарантованої доставки у протоколі TCP використовується алгоритм «ковзаючи вікно» (sliding window).

Протокол IP (Internet Protocol) передає всі частини отримувачу.

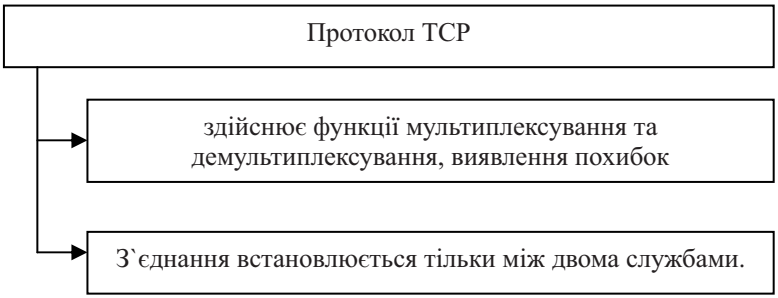

Рис.2. Блок-схема функціонування протоколу TCP

Мережний протокол – набір правил і дій (черговість дій), що дозволяє здійснити з`єднання і обмін даними між двома і більше включеними в мережу пристроями. Різні протоколи описують лише різні сторони одного типу зв`язку. Якщо їх об`єднати разом, вони утворять стек протоколів. Термін протокол і стек протоколів вказує на програмне забезпечення, за допомогою якого реалізується протокол.

За допомогою моделі OSI протоколи можна розділити на 7 рівнів від фізичного до прикладного.

Рис.3. Блок-схема 7-мирівневої моделі OSI

По своєму призначенню протоколи можна розділити на чотири групи:

Рис.4. Блок-схема розподілу протоколів.

Стек протоколів TCP/IP – протоколи нижчого рівня які є основою з`єднання в мережі Інтернет.

Данні транспортного рівня, що належать протоколу TCP називаються

сегментом TCP (segment TCP). Він складається з заголовку і поля даних. Інформація прикладного рівня є полем даних і визначаються наступними параметрами як на передаючій так і на приймаючій стороні:

- максимальний розмір сегмента передавача SMSS (sender maximum segment size);
- значення максимального розміру сегмента який може бути прийнятий приймачем RMSS;
- максимальний розмір сегмента який може бути відправлений MSS;
- повно розмірний сегмент даних максимально дозволеного розміру FSS (full sized segment).

Рис.5. Блок-схема сегменту TCP.

Склад поля сегменту TCP відображений у наступній блок схемі.

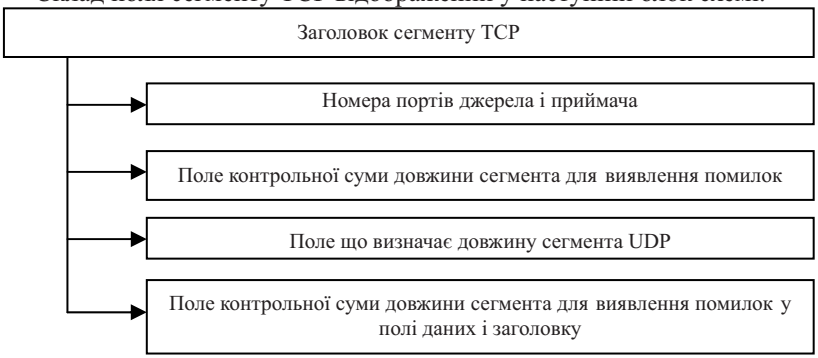

Рис. 6. Блок-схема складу поля сегменту TCP.

Управління з`єднанням у протоколі TCP здійснюється таким чином. Наприклад, додатку, що функціонує у межах якогось серверу гарантованої передачі даних необхідно встановити з`єднання TCP з іншим процесом, що функціонує у межах іншого сайту. Спочатку додаток отримувача повинен

надати інформацію протоколу транспортного рівня про те, що він бажає встановити з`єднання. Сервер клієнта розпочинає процедуру з`єднання з протоколом TCP іншого сайту у такій послідовності:

1. Клієнт надсилає до сервера дані прикладного рівня у вигляді службового сегмента TCP. Значення біту SYN у його заголовку дорівнює 1, і починаючи з цього сегменту клієнт повинен визначати нумерацію наступних сегментів.

2. Нехай SYN-сегмент успішно отриманий сервером TCP, і він готовий встановити і підтримувати з`єднання, але для цього йому необхідно встановити і визначити ресурси і параметри для з`єднання TCP і надіслати відповідний сегмент клієнту. Він, так само як і SYN, не вміщає прикладних даних, але має містити біт SYN якій відповідає значенню 1, номер наступного сегмента та інформація про наступний сегмент.

3. Після отримання клієнтським сервером TCP сегменту SYNACK (client_isn + 1), клієнту необхідно визначити параметри і ресурси для з`єднання. Таким чином клієнт підтверджує, що з`єднання відбулось.

Після проведення цих трьох дій клієнт отримує від сервера дані з гарантованою доставкою за допомогою протоколу TCP.

Для контролю і управлінням потоком сегментів протоколу TCP властиві функції контролю і управління потоком сегментів. Для доставки сегментів TCP використовується протокол мережного рівня IP, який не гарантує доставки пакетів, черговості їх доставки та цілісності даних в пакетах. Доставка пакетів у неправильному порядку, або їх втрата виникає внаслідок перевантаження маршрутизатора. Протоколу TCP властиві функції виділення нумерації сегментів даних, підтвердження прийнятих приймачем сегментів і управління таймером.

Нумерація і посилка сегментів даних здійснюється таким чином. У заголовку сегмента TCP є два поля, перше відображає нумерацію сегмента у послідовності, друге відображає номер наступного сегмента. Ці поля заповнюються по такому принципу. Програмні додатки транслюють потік байтів на транспортний рівень протоколу TCP, який у свою чергу розділяє отримані дані на сегменті з довжиною поля, яке дорівнює максимальному розміру сегментів, що може бути переданий приймачу. Для формування вихідного сегменту TCP кожний з них доповнюється заголовком, у якому призначається номер, що заноситься у поле номер сегмента у послідовності. Утворюється впорядкована послідовність сегментів, які обробляються протоколом TCP у відповідності з правилами.

Внаслідок того, що розмір поля має фіксований розмір, і вміщується у 32-х бітне поле, в протоколі TCP під час пересилання даних може виникнути ситуація під час якої номер сегмента може перевищити максимальне значення. У такому випадку передбачена циклічна нумерація сегментів, коли нумерація починається з початку.

Підтвердження про отримання відправлених сегментів здійснюється за допомогою сегментів ACK (asknowledgement), які мають свою нумерацію.

Наприклад сервер А відправив сегмент з даними серверу В поле даних містить 536 байтів, які пронумеровані від 0 до 536. Якщо сайт В прийняв сегмент коректно, у такому випадку він повинен підтвердити сайту А ,що він готовий прийняти наступний сегмент номер якого почнеться з байту 537. Після закінчення передачі сайт-джерело зупиняється і чекає підтвердження.

В іншому випадку, коли сервер-джерело А передав серверу-приймачу В сегменти нумерація яких відповідає значенням від 0 до 536 та 1073 до 1608, а сегмент з нумерацією байтів з 536 до 1072 був отриманий пізніше. Використовуючи номера сегментів, протокол TCP відновлює послідовність сегментів. Приймач надсилає до джерела сегмент АСК який містить номер байта, який він готовий прийняти. Це значно зменшує навантаження і збільшує пропускну здатність з`єднання. Цей тип передачі називається напівбезперевним і застосовується у протоколах GBN і Selective Repeat.

Протокол GBN (Go-back-N) протокол з поверненням на N пакетів назад дозволяє джерелу передавати декілька сегментів підряд, не очікуючи підтвердження АСК.

Протокол SR (Selective Repeat) – (вибіркове повторення) використовується при низькій пропускній здатності каналу. Коли у каналі знаходиться багато сегментів і втрата одного з них буде вимагати повторної передачі великої кількості сегментів. Функції цього протоколу допомагають запобігти повторної передачі тих сегментів, які були успішно прийняті, але не у тому порядку, у якому вони відправлялись. Повторно передаються тільки ті сегменти, які були передані з помилками.

Відмінність цих протоколів полягає у тому, що у вікні протоколу SR можуть знаходитись відправлені і підтверджені сегменти. Реалізований у цьому протоколі алгоритм «ковзаюче вікно» забезпечує прийняття сегмента підтвердження, який був прийнятий без помилок, незважаючи на його номер. Ресивер приймає позачергово сегменти, незважаючи на їх номер і поміщає у буфер до тих пір, поки не надійдуть сегменти з потрібним порядковим номером. Після цього сегменти передаються на прикладний рівень. Алгоритм роботи протоколу SR представлено на рис.7.

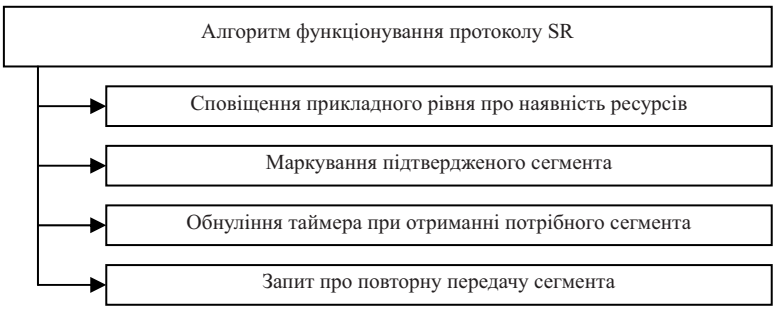

Рис.7. Блок-схема алгоритму функціонування протоколу SR.

Гарантована доставка даних у протоколі TCP забезпечуються реалізацією наступних функцій:

- нумерація сегментів;
- алгоритм «ковзаючи вікно»
- використання сегменту підтвердження АСК;
- таймер, вираховування контрольної суми сегментів.

Виходячи з того що з`єднання TCP є напівдуплексним і на боці приймача і передавача є буфери для обробки даних на транспортному рівні, але які ще не оброблені прикладним рівнем. Коли у ресивер приймача проступив сегмент, він проходить обробку на транспортному рівні, після чого переміщується у буфер, де проходить обробку на прикладному рівні. Момент переходу сегмента з транспортного рівня на прикладний може не співпасти, і тоді виникають перевантаження, що приводить до втрати даних.

Для такого випадку у протоколі TCP створена функція управління потоками, за допомогою якої вдається налаштувати швидкість поступлення даних у буфер у залежності від швидкості обробки сегментів прикладним рівнем.

Управління потоками здійснюється у протоколі TCP завдяки існуванню алгоритму «ковзаючи вікно» і накладає на нього такі обмеження:

- кількість даних, які приймач може прийняти підряд;
- кількість даних, які передавач може відправити приймачу без підтвердження доставки;
- розмір вікна після встановлення з`єднання;
- кількість відправлених і ще непідтверджених даних.

Зважаючи на те, що зв`язок, встановлений протоколом TCP є напівдуплексним, стає зрозумілим що на стороні як приймача так і передавача має бути встановлене вікно приймача. Якщо виникає необхідність передати великий файл, під час встановлення зв`язку між службами для нього встановлюють буфер фіксовано розміру, що дозволяє не переповнювати буфер приймача.

У протоколі TCP для управління потоками використовується параметр кількість відправлених, але ще непідтверджених приймачем сегментів. Якщо у приймачі кількість непідтверджених байтів буде зберігатись меншою, ніж розмір буфера, то у такому випадку переповнення буфера не відбудеться.

Для забезпечення гарантованої доставки даних через канал, у якому можуть виникнути завади, у протоколі TCP здійснюється управління перевантаженнями. Для управління перевантаженнями використовуються алгоритми:

- повільний старт;
- упередження перевантаження;
- швидка повторна передача;
- швидке відновлення.

Управління перевантаженням у протоколі TCP здійснюється адаптивно, в залежності від поточного навантаження у мережі. Таким чином, якщо джерелу інформації поступили данні про те, що на маршруті від передавача до приймача навантаження немає, то відповідно буде велика швидкість передачі сегментів. Але якщо виникне стан, близький до перевантаження, у такому випадку швидкість буде зменшена для уникнення перевантаження. У протоколі TCP визначено, що і приймач і передавач мають буфер, і завдяки зміні розміру вікна здійснюється управління кількістю даних, які передаються передавачем через мережу приймачеві, але непідтверджених. Кількість непідтверджених сегментів у буфері передавача не може перевищити максимальну кількість даних.

Коли максимальна кількість непідтверджених сегментів у передавачі дорівнює нулю і обмежена кількістю сегментів, які передавач може надіслати приймачеві без підтвердження про отримання. Це дає можливість обмежити кількість непідтверджених сегментів і обмежити швидкість передачі, при яких значення затримки і втрати сегмента незначні. У кожний момент часу розміри відправлених сегментів відповідає розміру вікна перевантаження. у кінці проміжку часу обороту сегменту TCP від відправки останнього сегменту даних до передавача надходить сегмент підтвердження.

Для проведення ефективної оцінки роботи з'єднання TCP використовується методика, при якій співвідношення часу обороту сегмента до кількості даних, які передавач може відправити приймачу без підтвердження доставки. Це дає можливість передавачу підібрати швидкість у відповідності з кількістю даних, які передавач може передати без підтвердження.

1. *Е.А. Кучерявый*. Управление трафиком и качество обслуживания в сети Интернет.— Спб.: Наука и техника, 2004. – 336 с.

Поступила 21.03.2013р.