Формування черг трафіка..

Аналіз черг трафіка.

Резервування ресурсів. Забезпечення заданого рівня затримок.

**Алгоритми управління чергами**

Список ключових слів: алгоритм FIFO, алгоритм пріоритетного обслуговування, пріоритет, класифікація трафіку, точка класифікації трафіку, розмір буфера, клас трафіку, гранулярность, агрегат трафіку, алгоритм зважених черг, зважене справедливе обслуговування, комбіновані алгоритми обслуговування черг, засоби профілювання трафіку.

Алгоритми управління чергами потрібні для роботи в періоди тимчасових перевантажень, коли мережевий пристрій не справляється з передачею пакетів на вихідний інтерфейс в тому темпі, в якому вони надходять. Якщо причиною перевантаження є недостатня продуктивність процесорного блоку мережевого пристрою, то необроблені пакети тимчасово накопичуються у вхідній черзі відповідного вхідного інтерфейсу. Черг до вхідного інтерфейсу може бути кілька, якщо ми диференціюючи запити на обслуговування по декількох класах. У тому ж випадку, коли причина перевантаження полягає в обмежену пропускну здатність вихідного інтерфейсу, пакети тимчасово зберігаються в вихідний черги (або чергах) цього інтерфейсу.

**алгоритм FIFO**

У традиційному алгоритмі FIFO в разі перевантаження все пакети поміщаються в одну загальну чергу і вибираються з неї в тому порядку, в якому надійшли. У всіх пристроях з комутацією пакетів алгоритм FIFO використовується за умовчанням. Перевагами його є простота реалізації і відсутність потреби в конфігурації. Однак йому притаманний і корінний недолік - неможливість диференційованої обробки пакетів різних потоків. Всі пакети коштують в загальній черзі на рівних підставах. Разом виявляються і пакети чутливого до затримок голосового трафіку, і пакети нечутливого до затримок, але дуже інтенсивного трафіку резервного копіювання, тривалі пульсації якого можуть надовго затримати голосовий пакет.

**пріоритетне обслуговування**

Алгоритми пріоритетного обслуговування дуже популярні в багатьох областях обчислювальної техніки, зокрема в операційних системах, коли одним з додатками потрібно віддати перевагу перед іншими при обробці їх в мультипрограммной суміші. Застосовуються ці алгоритми і для переважної в порівнянні з іншими обробки одного класу трафіку.

Механізм пріоритетного обслуговування заснований на поділі всього мережевого трафіку на невелику кількість класів і подальшого призначення кожного класу деякого числового ознаки - пріоритету.

Класифікація трафіку є окреме завдання. Пакети можуть розбиватися на пріоритетні класи на підставі різних ознак: адреси призначення, адреса джерела, ідентифікатора додатка, що генерує цей трафік, будь-яких інших комбінацій ознак, які містяться в заголовках пакетів. Правила класифікації пакетів є частиною політики адміністрування мережі.

Точка класифікації трафіку може розміщуватися в кожному комунікаційному пристрої. Більш масштабується, - розміщення функцій класифікації трафіку в одному або декількох пристроях, розташованих на кордоні мережі (наприклад, в комутаторах корпоративної мережі, до яких підключаються комп'ютери користувачів, або у вхідних маршрутизаторах мережі постачальника послуг). В цьому випадку необхідно спеціальне поле в пакеті, в якому можна запам'ятати призначене значення пріоритету, щоб ним могли скористатися інші мережеві пристрої, що обробляють трафік після классифицирующего пристрою. Таке поле є в заголовку багатьох протоколів. У тих же випадках, коли спеціального поля пріоритету в заголовку немає, розробляється
 додатковий протокол, який вводить новий заголовок з таким полем (так сталося, наприклад, з протоколом Ethernet).

Пріоритети можуть призначатися не тільки комутатором або маршрутизатором, а й додатком на вузлі-відправнику. Необхідно також враховувати, що якщо в мережі відсутня централізована політика призначення пріоритетів, кожне мережеве пристрій може не погодитися з пріоритетом, призначеним даному пакету в іншій точці мережі. У цьому випадку воно перепише значення пріоритету відповідно до локальної політикою, прийнятої безпосередньо на цьому пристрої.

В мережевому пристрої, що підтримує пріоритетне обслуговування, є кілька черг (буферів), по одній для кожного пріоритетного класу. Пакет, що надійшов в період перевантажень, поміщається в чергу, відповідну його пріоритетному класу[18]. На рис. 7.9 наведено приклад використання чотирьох пріоритетних черг з високим, середнім, нормальним і низьким пріоритетом. До тих пір поки з більш пріоритетною черги не будуть обрані всі наявні в ній пакети, пристрій не переходить до обробки наступної, менш пріоритетною черзі. Тому пакети з низьким пріоритетом обробляються тільки тоді, коли порожні всі вищі черзі: з високим, середнім і нормальним пріоритетами.

|  |
| --- |
|  Черги різних пріоритетів |
|  високий |

|  |
| --- |
|  Вибір з черги, якщо все більш пріоритетні черги порожні |
|  Планувальник (абсолютні пріоритети) |
|  вихідний трафік |
|  Вихідна чергу |
|  Буфери різної довжини Рис. 7.9. Пріоритетні черги |

Розмір буфера мережевого пристрою визначає максимальну довжину черги чекають обслуговування пакетів, якщо пакет надходить при заповненому буфері, то він просто відкидається. Зазвичай за умовчанням всіх пріоритетних черг відводяться однакові буфери, але багато пристроїв дозволяють адміністратору призначати кожної черги буфер індивідуального розміру. Розмір буфера визначається в ідеальному випадку таким чином, щоб їх було вдосталь з деяким запасом для зберігання черги середньостатистичної довжини. Однак встановити це значення досить складно, так як воно змінюється в залежності
 від навантаження мережі, тому потрібне постійне і тривале спостереження за роботою мережі. У загальному випадку, чим вище значимість трафіку для підприємства, чим більше його інтенсивність і пульсації, тим більший розмір буфера потрібно цього трафіку. У прикладі, наведеному на рис. 7.9, з точки зору трафіку вищого і нормального пріоритету обрані великі розміри буферів, а для інших двох класів - менші. Мотиви прийнятого рішення для вищого пріоритету очевидні, а трафік нормального пріоритету має, очевидно, високу інтенсивність і значний коефіцієнт пульсацій.

Пріоритетне обслуговування черг забезпечує високу якість обслуговування для пакетів з найпріоритетнішу черзі. Якщо середня інтенсивність їх надходження в пристрій не перевищує пропускну здатність вихідного інтерфейсу (і продуктивності внутрішніх просувають блоків самого пристрою), то пакети вищого пріоритету завжди отримують ту пропускну здатність, яка їм потрібна. Рівень затримок високопріоритетних пакетів також мінімальний. Однак він не нульовий і залежить в основному від характеристик потоку цих пакетів - чим вище пульсації потоку і його інтенсивність, тим імовірніше виникнення черги, утвореної пакетами даного високопріоритетного потоку. Трафік всіх інших пріоритетних класів майже прозорий для пакетів вищого пріоритету. Слово «майже» відноситься до ситуації, коли високопріоритетний пакет змушений чекати завершення обслуговування низькопріоритетного пакета, якщо його прихід збігається за часом з початком просування низькопріоритетного пакета на вихідний інтерфейс.

Що ж стосується інших пріоритетних класів, то якість їх обслуговування буде нижче, ніж у пакетів найвищого пріоритету, причому рівень зниження може бути дуже суттєвим. Якщо коефіцієнт навантаження вихідного інтерфейсу, який визначається тільки трафіком вищого пріоритетного класу, наближається в якийсь період часу до одиниці, то трафік інших класів просто на цей час заморожується. Тому пріоритетне обслуговування зазвичай застосовується для класу трафіку, чутливого до затримок, що має невелику інтенсивність. При таких умовах обслуговування цього класу не занадто обмежує обслуговування решти трафіку. Наприклад, голосовий трафік чутливий до затримок, але його інтенсивність зазвичай не перевищує 8-16 Кбіт / с, так що при призначенні йому вищого пріоритету шкоду іншим класам трафіку буде не дуже значним.

Однак в мережі можуть спостерігатися і інші ситуації. Наприклад, відеотрафік також вимагає пріоритетного обслуговування, але має набагато більш високу інтенсивність. Для таких випадків розроблені алгоритми обслуговування черг, що дають фонову трафіку деякі гарантії навіть в періоди підвищення інтенсивності високопріоритетного трафіку.

Уважний читач, очевидно, вже звернув увагу на те, що при описі роботи пріоритетних черг фігурували не окремі потоки, а класи трафіку. Це важлива особливість, яка відноситься не тільки до пріоритетних алгоритмам, а й до багатьох інших механізмів підтримки якості обслуговування.

Мережа може обслуговувати трафік з різним ступенем гранулярності. Окремий потік являє собою мінімальну одиницю обслуговування, яку беруть до уваги механізми забезпечення заданих параметрів QoS. Якщо ми забезпечуємо кожному потоку власні параметри QoS, то це підтримка якості обслуговування на рівні потоків. Якщо ми об'єднуємо декілька потоків в єдиний потік і перестаємо розрізняти окремі потоки при забезпеченні параметрів QoS, то це підтримка якості обслуговування на рівні класів трафіку. Такі класи також називають агрегатами трафіку.

**УВАГА --------------------------------------------------------------------------------------------------------**

Для того щоб потоки можна було об'єднати в агрегат, потрібно, щоб вони пред'являли однакові вимоги до якості обслуговування і мали спільні точки входу в мережу і виходу з мережі.

**зважені черги**

Алгоритм зважених черг розроблений для того, щоб можна було надати всім класам трафіку певний мінімум пропускної здатності або гарантувати деякі вимоги до затримок. Під вагою даного класу розуміється відсоток надається класу трафіку пропускної здатності від повної пропускної здатності вихідного інтерфейсу.

При зваженому обслуговуванні так само, як при пріоритетному, трафік ділиться на кілька класів, і для кожного класу ведеться окрема черга пакетів. Але з кожною чергою зв'язується не пріоритет, а відсоток пропускної здатності ресурсу, який гарантується даному класу трафіку при перевантаженнях цього ресурсу. Для вхідного потоку таким ресурсом є процесор, а для вихідного потоку (після виконання комутації) - вихідний інтерфейс.

**приклад**

Показане на рис. 7.10 пристрій для 5 класів трафіку підтримує 5 черг до вихідного інтерфейсу комутатора. Цим чергам при перевантаженнях виділяється відповідно 10%, 10%, 30%, 20% і 30% пропускної здатності вихідного інтерфейсу.

Досягається поставлена мета тим, що черги обслуговуються послідовно і циклічно, і в кожному циклі обслуговування з кожної черги вибирається таке число байтів, яке відповідає вазі даної черги. Наприклад, якщо цикл перегляду черг в розглянутому прикладі дорівнює одній секунді, а швидкість вихідного інтерфейсу становить 100 Мбіт / с, то при перевантаженнях в кожному циклі першої черги приділяється 10% часу, тобто 100 мс і вибирається 10 Мбіт даних, з другої - теж 10 Мбіт, з третьої - 30 Мбіт, з четвертої - 20 Мбіт, з п'ятої - 30 Мбіт.

В результаті кожного класу трафіку дістається гарантований мінімум пропускної здатності, що в багатьох випадках є більш бажаним результатом, ніж придушення фонових класів високопріоритетним.

|  |
| --- |
|  |

|  |
| --- |
|  вхідний трафік |
|  Класифікатор (протокол, адреси) |

|  |
| --- |
|  вихідний трафік |
|  Вихідна чергу |
|  Планувальник про (зважене циклічне просування) |
|  Мал. 7.10. зважені черги |

Так як дані вибираються з черги пакетами, а не бітами, то реальний розподіл пропускної спроможності між класами трафіку завжди трохи відрізняється від планованого. Наприклад, замість 10% перший клас трафіку може отримати при перевантаженнях 9 або 12%. Чим більше час циклу, тим точніше дотримуються необхідні пропорції між класами трафіку, так як з кожної черги вибирається велике число пакетів і вплив розміру кожного пакета усредняется.

З іншого боку, тривалий цикл призводить до більших затримок передачі пакетів. Наприклад, при обраному нами для прикладу циклі в одну секунду затримка може скласти одну секунду і більше - адже арбітр повертається до кожної черги не частіше, ніж раз в секунду, крім того, в черзі може перебувати більше одного пакета. Тому при виборі часу циклу потрібно забезпечити баланс між точністю дотримання пропорцій пропускної здатності і прагненням до зменшення затримки.

Для нашого прикладу час циклу в 1000 мкс є прикладом такого балансу. З одного боку, воно забезпечує обслуговування черги кожного класу кожні 1000 мкс, а значить - більш низький рівень затримок. З іншого боку, цього часу достатньо, щоб вибрати з кожної черги в середньому по кілька пакетів (першої черги в нашому прикладі буде відводитися 100 мкс, що досить для передачі в вихідний канал одного пакета Fast Ethernet або десяти пакетів Gigabit Ethernet).

На рівень затримок і варіації затримок пакетів для деякого класу трафіку при зваженому обслуговуванні в значній мірі впливає коефіцієнт використання. У цьому випадку коефіцієнт підраховується як відношення інтенсивності вхідного трафіку класу до пропускної спроможності, виділеної цього класу відповідно до його вагою. Наприклад, якщо ми виділили першої черги 10% від загальної пропускної здатності вихідного інтерфейсу, тобто 10 Мбіт / с, а дредняя інтенсивність потоку, який потрапляє в цю чергу, дорівнює 3 Мбіт / с, то коефіцієнт використання для цього потоку складе 3/10 = 0,3. Залежність на рис. 7.5 показує, що затримки в такому значенні коефіцієнта використання будуть незначними. Якби інтенсивність вхідного потоку цієї черги була 9 Мбіт / с, то черги були б значними, а при перевищенні меж 10 Мбіт / с частина пакетів потоку постійно б відкидалася через переповнення черги.

Якісне поведінку черзі і, відповідно, затримок тут виглядає приблизно так само, як і в разі черги FIFO - чим менше коефіцієнт завантаження, тим менше середня довжина черги і тим менше затримки.

Як і для пріоритетного обслуговування, при зваженому обслуговуванні адміністратор може призначати різних класів черг буфери різних розмірів. Зменшення розмірів буферів для черг веде до зростання числа втрат пакетів при перевантаженнях, але зате знижує час очікування для тих пакетів, що не були відкинуті і потрапили в чергу.

Існує також такий вид зваженого обслуговування, як зважене справедливе обслуговування (Weighted Fair Queuing, WFQ). У разі подібного обслуговування пропускна здатність ресурсу ділиться між всіма потоками порівну, тобто «справедливо».

**УВАГА --------------------------------------------------------------------------------------------------------**

Виважена обслуговування забезпечує необхідні співвідношення між інтенсівностя- ми трафіку різних черг тільки в періоди перевантажень, коли кожна чергу постійно заповнена. Якщо ж якась із черг порожня (тобто з точки зору трафіку даного класу поточний період не є періодом перевантаження), то при перегляді черг вона пропускається, і її час обслуговування розподіляється між іншими чергами відповідно до їх вагою. Тому в окремі періоди трафік певного класу може мати більшу інтенсивністю, ніж відповідний відсоток від пропускної здатності вихідного інтерфейсу.

**Комбіновані алгоритми обслуговування черг**

Кожен з описаних підходів має свої достоїнства і недоліки. Пріоритетне обслуговування, забезпечуючи мінімальний рівень затримок для черги найвищого пріоритету, не дає ніяких гарантій щодо середньої пропускної здатності для трафіку черг нижчих пріоритетів. Виважена обслуговування забезпечує заданий розподіл середньої пропускної здатності, але не враховує вимог до затримок.

Існують комбіновані алгоритми обслуговування черг. У найбільш популярному алгоритмі подібного роду підтримується одна пріоритетна чергу при обслуговуванні інших черг відповідно до зваженим алгоритмом. Зазвичай пріоритетна чергу використовується для чутливого до затримок трафіку, а решта - для еластичного трафіку декількох класів. Кожен клас еластичного трафіку отримує певний мінімум пропускної здатності при перевантаженнях. Цей мінімум обчислюється як відсоток від пропускної здатності, що залишилася від пріоритетного трафіку. Очевидно, що потрібно якось обмежити пріоритетний трафік, щоб він не поглинав всю пропускну здатність ресурсу. Зазвичай це робиться засобами профілювання трафіку, які розглядаються далі.

**Зворотній зв'язок**

Алгоритми управління чергами не запобігають перевантажень, а лише деяким «справедливим» чином в умовах дефіциту перерозподіляють ресурси між різними потоками або класами трафіку. Алгоритми управління чергами відносяться до механізмів контролю перевантажень, які починають працювати, коли мережа вже перевантажена.

Існує інший клас засобів, які носять назву механізмів запобігання перевантажень. Очевидно, що запобігти перевантаженню мережі можна в тому випадку, коли сумарна інтенсивність всіх потоків, що передаються кожним інтерфейсом кожного комутатора мережі, менша за пропускну спроможність цього інтерфейсу. Домогтися цього можна двома способами - збільшуючи пропускну здатність інтерфейсу або зменшуючи інтенсивності потоків. Перший варіант відноситься до засобів проектування і планування мережі і тому тут не розглядається.

Другий варіант - зменшення інтенсивності потоків - можна реалізувати також двома принципово різними способами. Перший спосіб заснований на використанні механізму зворотного зв'язку, за допомогою якого перевантажений вузол мережі, реагуючи на перевантаження, просить попередні вузли, розташовані уздовж маршруту проходження потоку (або потоків, що належать до одного класу), тимчасово знизити швидкість трафіку. Після того як перевантаження в даному вузлі зникне, він посилає ще одне повідомлення, яке дозволяє підвищити швидкість передачі даних. Інший спосіб заснований на попередньому резервуванні пропускної здатності для потоків, що протікають через мережу. Для цього йому необхідна попередня інформація про интенсивностях потоків. Принципи резервування ресурсів ми розглянемо пізніше, а зараз зупинимося на механізмах зворотного зв'язку.

# Інжиніринг трафіку

При розгляді системи забезпечення якості обслуговування, заснованої на резервування, ми не стали порушувати питання маршрутів слідування потоків через мережу. Точніше, ми вважали, що вони якимось чином обрані, причому цей вибір робиться без урахування вимог QoS. І в умовах заданий ™ маршрутів ми намагалися забезпечити проходження по цих маршрутах такого набору потоків, для якого можна гарантувати дотримання вимог QoS.

Очевидно, що завдання підтримки вимог QoS можна вирішити більш ефективно, якщо вважати, що маршрути проходження трафіку не фіксовані, а також підлягають вибору. Це дозволило б мережі обслуговувати більше потоків з гарантіями QoS при тих же характеристиках самої мережі, тобто пропускної здатності каналів і продуктивності комутаторів і маршрутизаторів.

Завдання вибору маршрутів для потоків (або класів) трафіку з урахуванням дотримання вимог QoS вирішують методи інжинірингу трафіку (Traffic Engineering, ТІ). За допомогою цих методів прагнуть домогтися ще однієї мети - по можливості максимально і збалансовано завантажити всі ресурси мережі, щоб мережа при заданому рівні якості обслуговування володіла якомога більш високою сумарною продуктивністю.

Методи ТЕ, як і інші розглянуті раніше методи, засновані на резервуванні ресурсів. Тобто вони не тільки дозволяють знайти раціональний маршрут для потоку, але і резервують для нього пропускну здатність ресурсів мережі, що знаходяться вздовж цього маршруту.

Методи інжинірингу трафіку є порівняно новими для мереж з комутацією пакетів. Це пояснюється багато в чому тим, що передача еластичного трафіку не висувала строгих вимог до параметрів QoS. Крім того, Інтернет довгий час не з'являвся комерційною мережею, тому максимальне використання ресурсів не вважалося першочерговим завданням для IP-технологій, що лежать в основі Інтернету.

Сьогодні ситуація змінилася. Мережі з комутацією пакетів повинні передавати різні види трафіку із заданою якістю обслуговування, максимально використовуючи можливості своїх ресурсів. Однак для цього їм потрібно змінити деякі, що стали вже традиційними, підходи до вибору маршрутів.