Лобжанидзе Лили

Повышение производительности компьютерной сети путем выбора оптимальной маршрутизации и решения задачи управления потоками

представлена на соискание академической степени
доктора

Грузинский Технический Университет
Тбилиси, 0175, Грузия
2008

საქართველოს ტექნიკური უნივერსიტეტი

ინფორმატიკისა და მართვის სისტემების ფაკულტეტი

ჩვენ, ქვემორე ხელისმომწერნი ვადასტურებთ, რომ გავეცანით ლილი ლობჟანიძეს მიერ შესრულებულ სადისერტაციო ნაშრომს დასახელებით: “კომპიუტერული ქსელის წარმადობის ამაღლება ოპტიმალური მარშრუტიზაციის შერჩევით და ნაკადების მართვის ამოცანის გადაწყვეთით” და ვაძლევთ რეკომენდაციას საქარველოს ტექნიკური უნივერსიტეტის ინფორმატიკისა და მართვის სისტემების ფაკულტეტის სადისერტაციო საბჭოში მის განხილვას დოქტორის აკადემიური ხარისხის მოსაპოვებლად.

თარიღი

ხელმძღვანელი: _____

ხელმძღვანელი: _____

რეცენზენტი: _____

რეცენზენტი: _____

რეცენზენტი: _____

საქართველოს ტექნიკური უნივერსიტეტი

2008

ავტორი:	ლობჟანიძე ლილი
დასახელება:	“კომპიუტერული ქსელის წარმადობის ამაღლება ოპტიმალური მარშრუტიზაციის შერჩევით და ნაკადების მართვის ამოცანის გადაწყვეთით”
ფაკულტეტი :	ინფორმატიკისა და მართვის სისტემების
აკადემიური ხარისხი:	ასპირანტი
სხდომა ჩატარდა:	

ავტორის ხელმოწერა

რეზუმე

თანამედროვე საზოგადოებაში ფრიად გავრცელებულია კომპიუტერული ქსელების გამოყენება. ძალიან ფართო გახდა შექმნილი და მომუშავე ტერიტორულად განაწილებული და ლოკალური ქსელების სპექტრი. კომპიუტერული ქსელების შემუშავებაში ჩამოყალიბდა ახალი ტენდენცია, რომელიც გამოიხატება დიდი რაოდენობის განსხვავებული, ტერიტორიულად განაწილებული და ლოკალური ქსელების გაერთიანებაში. მოცემული საკითხი მოითხოვს ახალ მიდგომას კომპიუტერული ქსელების არქიტექტურისადმი და მათი ეფექტური ურთიერთქმედების მოდელებისადმი.

კომპიუიტერული ქსელების გამოყენების თანამედროვე ამოცანები აყენებს ძალიან მაღალ მოთხოვნებს ქსელების მომსახურეობის ხარისხისადმი QoS (Quality of Service). მომსახურეობის ხარისხის ამაღლების მიზნით შეიძლება იყოს ჩატარებული მრავალი ღონისძიება. ერთერთი მათგანი არის რესურსების მართვა ქსელში. კომპიუტერულ ქსელებში რესურსებად შეიძლება იყოს განხილული გარკვეული აპარატურული და პროგრამული კომპლექსი. რა თქმა უნდა, რესურსებად შეიძლება განხიული იყოს ქსელური აპარატურა და პროგრამული უზრუნველყოფა, როგორიც არის ოპეარციული სისტემები და მონაცემთა ბაზების მართვის სისტემები. მაგრამ მომსახურეობის ხარისხის QoS -ის თვალსაზრისით რესურსებად განიხილავენ დაყოვნებას და გამტარუნარიანობას.

დაყოვნებას უწოდებენ იმ დროით ინტერვალს, რომელიც საჭიროა პაკეტის გადასაადგილებლად წყაროდან მიმღებამდე დამაკავშირებელი ქსელის საშუალებით.

გადაცემის დაყოვნება განისაზღვრება როგორც დაყოვნება ქსელის რომელიმე მოწყობილობის შემოსასვლელზე პაკეტის შემოსვლის მომენტსა და ამ მოწყობილობის გამოსასვლელზე გამოჩენის მომენტს შორის. წარმადობის ეს პარამეტრი თავისი შინაარსით ახლოს არის ქსელის რეაქციის დროსთან, მაგრამ განსხვავდება მისგან იმით, რომ ყოველთვის ახასიათებს მონაცემთა დამუშავების მხოლოდ ქსელურ ეტაპებს, კომპიუტერების დამუშავების დაყოვნების გარეშე. ჩვეულებრივად, ქსელის ხარისხი ხასიათება გადაცემის მაქსიმალური დაყოვნებითა და დაყოვნების ვარიაციით.

ქსელის გამტარუნარიანობა არის მომსახურეობათა რაოდენობა, რომელიც შეუძლია უზრუნველყოს ქსელს დროის ერთეულში. გამტარუნარიანობა ასახავს მონაცემთა მოცულობას, გადაცემულს ქსელით ან მისი ნაწილით დროის ერთეულში. გამტარუნარიანობა არ არის სამომხმარებლო მახასიათებელი, ვინაიდან ის მეტყველებს ქსელის შიდა ოპერაციების შესრულების სიჩქარეზე - ანუ მონაცემთა პაკეტების გადაცემაზე ქსელის კვანძებს შორის კომუნიმაციური მოწყობილობების საშუალებით. სამაგიეროდ ის უშუალოდ ახასიათებს ქსელის ძირითადი ფუნქციის შესრულებას - შეტყობინებების ტრანსპორტირებას, და ამიტომაც,

ის უფრო ხშირად გამოიყენება ქსელის წარმადობის ანალიზის დროს, ვიდრე რეაქციის დრო.

ქსელის გამტარუნარიონაბა იზომება ან ბიტებით წამში ან პაკეტებით წამში. გამტარუნარიანობა შეიძლება იყოს წამიერი, მაქსიმალური და საშუალო. საშუალო გამტარუნარიანობა იზომება გადაცემული მონაცემების მთელი მოცულობის გაყოფით მათი გადაცემის დროზე, ამისათვის ირჩევა საკმაოდ დიდი დროის მონაკვეთი - საათი, დღე ან კვირა. წამიერი გამტარუნარიანობა განსხვავდება საშუალოსგან იმით, რომ გასაშუალებისათვის შეირჩევა ძალიან პატარა დროის ინტერვალი. მაქსიმალური გამტარუნარიანობა - ეს არის ყველაზე დიდი წამიერი გამტარუნარიანობა, რომელიც ფიქსირდება დაკვირვების პერიოდის განმავლობაში.

აღნიშნულ პარამეტრებზე (დაყოვნება და ქსელის გამტარუნარიანობა) გავლენას ახდენს მრავალი ფაქტორი. მაგრამ ყველაზე მეტად - მარშრუტიზაცია და ნაკადების მართვა. ამასთან ერთად აღნიშნულ პარამეტრებზე გავლენას ახდენს ქსელის საიმედობა და გადაცემის შეცდომების გასწორება.

ქსელის საიმედობის მიმართ შეიძლება ითქვას შემდეგი. აქ განიხილავენ საიმედობის რამდენიმე ასპექტს. ტექნიკური მოწყობილობებისათვის გამოიყენება საიმედობის ისეთი მაჩვენებლები, როგორიც არის მტყუნებათა შორისი ნამუშევრის საშუალო დრო, მტყუნების ალბათობა, მტყუნებების ინტენსიურობა. მაგრამ ეს მაჩვენებლები სასარგებლოა უბრალო ელემენტების საიმედობის შესაფასებლად, რომლებიც შეიძლება იყოს მხოლოდ ორ მდგომარეობაში - მუშა და არამუშა. რთულ სისტემებს , რომელიც შედგება მრავალი ელემენტისაგან , მუშა და არამუშა მდგომარეობის გარდა შეიძლება გააჩნდეს სხვა შუალედური მდგომარეობა, რომელსაც აღნიშნული მახასიათებლები არ ითვლისწინებენ. ამასთან დაკავშირებით, საიმედობის შეფასების მიზნით გამოიყენება მახასიათებლების სხვა ნაკრები. ეს არის მზადყოფნა ან მზადყოფნის კოეფიციენტი, კვანძთან დამახინჯების გარეშე პაკეტის მიწოდების ალბათობა, უსაფრთხოება და მტყუნებათა მიმართ მდგრადობა.

იმისათვის, რომ სისტემა ჩაითვალოს მაღალსაიმედოდ, მას როგორც მინიმუმ უნდა გააჩნდეს მაღალი მზადყოფნა, მაგრამ ეს არ არის საკმარისი. საჭიროა უზრუნველყოფილი იყოს მონაცემთა შენახულობა და მათი დაცვა დამახინჯებისაგან. გარდა ამისა უნდა იყოს მხარდაჭერილი მონაცემთა შეთანხმებულობა. მაგალითად, თუ საიმედობის ამაღლების მიზნით რამდენიმე ფაილურ სერვერზე ინახება რამდენიმე ასლი, მაშინ მუდმივად უნდა იყოს უზურნველყოფილი მათი იდენტურობა.

შეცდომების აღმოფხვრა და გასწორება მოიცავს ქსელის მუშაობაში შეფერხებებისა და მტყუნებების შედეგების გამოვლენას, განსაზღვრასა და აღმოფხვრას, ამ დროს სრულდება არა მარტო შეცდომების შესახებ შეტყობინებების რეგსიტრაცია, არამედ მათი ფილტრაცია, მარშრუტიზაცია და ანალიზი.

ქსელის საიმედობის ამაღლების, შეცდომების გასწორებისა და აღმოფხვრის შედეგად მცირდება განმეორებით გადაცემული პაკეტების რაოდენობა, რაც დადებითად მოქმედებს ქსელის დაყოვნებაზე და გამტარუნარიანობაზე.

მარშრუტიზაციაში გულისხმობენ უწყვეტი გზის შერჩევას (რომელიც მოიცავს რამდენიმე შუალედურ ქსელურ ხაზს) ნებისმიერ ორ კვანძს შორის (წყაროსა და მიმღებს შორის). მარშრუტიზაციის ალგორითმს უნდა გააჩნდეს გარკვეული თვისებები: სისწორე, სიმარტივე, საიმედობა, მდგრადობა, სამართლიანობა და ოპტიმალურობა. მარშრუტიზაცია ურთიერთქმედებს ნაკადების მართვასთან უკუკავშირის მექანიზმის საშუალებით მახასიათებლების განსაზღვრისას. არსებობს მარშრუტიზაციის ალგორითმების დიდი რაოდენობა, რომლებშიც მიიღწევა სხვადასხვა მიზნები: გზის, დროის შემცირება. შესაბამისად, წარმოიშვება მარშრუტიზაციის პრობლემა, და ეს იძლევა დაყოვნების გარკვეულ დონეს.

ნაკადების მართვა - ეს არის იმ ღონისძიებების უძრუნველყოფა, რომელიც რეგულირებას უკეთებს ქსელის ტრაფიკს. ნაკადების მართვის ამოცანა წარმოიშვება იმ შემთხვევებში, როდესაც ქსელს ესაჭიროება დაყოვნების შენარჩუნება გარკვეულ დაბალ დონეზე ხოლო ქსელის დატვირთვა კი მატულობს, რაც შესაძლებელი ხდება მხოლოდ ზოგიერთი მოთხოვნის (მომხმარებლის) უარყოფის გზით.

სადისერტაციო ნაშრომის მიზანი იყო კომპიუტერული ქსელის რესურსების მართვის ამოცანების გადაჭრა ქსელის წარმადობის ამაღლების მიზნით.

სადირტაციო ნაშრომი შედგება შესავლისა და ხუთი თავისაგან.

შესავალში მოცემულია კომპიუტერული ქსელების განვითარების მოკლე მიმოხილვა. აგრეთვე განხილულია OSI მოდელის დონეები.

პირველ თავში განხილულია კომპიუტერული ქსელების მართვის ამოცანების ფუნქციონალური ჯგუფები, რომელიც მოიცავს ხუთ ჯგუფს. ესენია: ქსელის კონფიგურაციისა და დასახელებების მართვა, შეცდომების დამუშავება, წარმადობისა და საიმედობის ანალიზი, უსაფრთხოების მართვა და ქსელის მუშაობის ანალიზი.

მეორე თავში მოცემულია მარშრუტიზაციის ცნება. აღწერილია მარშრუტიზაციის ძირითადი ალგორითმები, მოცემულია ისეთი პარამეტრები, როგორიც არის მარშრუტის სიგრძე, საიმედობა, დაყოვნება, გამტარუნარიანობა, დატვირთვა და ქსელის ღურებულება. გარდა ამისა განხილულია ნაკადების მართვის საკითხები. ნაჩვენებია კავშირი მარშრუტიზაციასა და ნაკადების მართვას შორის.

მესამე თავში მოყვანილია მომსახურეობის ხარისხის ცნება, და აგრეთვე მისი ფუნქციები. განხილულია მარშრუტიზაცია ხარისხის მომსახურეობის თვალსაზრისით და მისი ძირითადი მაჩვენებლები კომპიუტერულ ქსელებში, რომელიც დაკავშირებულია მარშრუტიზაციასთან.

მეოთხე თავში გადაწყვეტილია რესურსების მართვის ძირითადი ამოცანები. დამუშავებულია მასობრივი მომსახურების მოდელი M/M/m m-

მომსახურე ხელსაწყოთი და მოდელი M/M/m/m m- დანაკარგებითა და m - მომსახურე ხელსაწყოთი. მოცემული მოდელები საშუალებას იძლევა ყველაზე ზუსტად შეფასდეს ქსელში დაყოვნების მნიშვნელობა. მოყვანილია მარშრუტიზაციის ოპტიმალურობის შეფასების მეთოდიკა და დამუშავებულია მოდელები და სქემები ნაკადების მართვისათვის ოპტიმალურ მარშრუტიზაციასთან შეხამებაში.

РЕЗЮМЕ

Применение компьютерных сетей в современном обществе широко распространено. Стал необычайно широким спектр создаваемых и работающих компьютерных сетей как территориально распределенных так и локальных. В разработке компьютерных сетей появилась тенденция объединения большого числа различных территориальных и локальных сетей. Эта вопрос требует нового подхода к архитектуре сетей, моделям их эффективного взаимодействия.

Современные задачи использования компьютерных сетей ставят высокие требования к качеству обслуживания сетей - QoS (Quality of Service). С целью повышения качества обслуживания может быть проведен ряд мероприятий. Одним из них является управления ресурсами. Ресурсами можно расссматривать определнный аппаратный и программный комплекс компьютерных сетей. Безусловно, ресурсами сети можно считать сетевую апппаратуру и программное обеспечение (ОС, СУБД). Но с точки зрения QoS ресурсами считаются задержка и пропускная способность.

Задержкой называют отрезок времени, необходимый для передвижения пакета от источника до пункта назначения через проводящую сеть. Задержка передачи определяется как задержка между моментом поступления пакета на вход какого-либо сетевого устройства или части сети и моментом появления его на выходе этого устройства. Этот параметр производительности по смыслу близок ко времени реакции сети, но отличается тем, что всегда характеризует только сетевые этапы обработки данных без задержек обработки компьютерами сети. Обычно, качество сети характеризуют величинами максимальной задержки передачи и вариацией задержки.

Пропускной способностью является количество обслуживаний, которое может обеспечить сеть в единицу времени. Пропускная способность отражает объем данных, переданных сетью или ее частью в единицу времени Пропускная способность не является пользовательской характеристикой, так как она говорит о скорости выполнения внутренних операций сети - передачи пакетов данных между узлами сети через различные коммуникационные устройства. Зато она непосредственно характеризует качество выполнения основной функции сети - транспортировки сообщений - и поэтому чаще используется при анализе производительности сети, чем время реакции.

Пропускная способность измеряется либо в битах в секунду, либо в пакетах в секунду. Пропускная способность может быть мгновенной, максимальной и средней.

Средняя пропускная способность вычисляется путем деления общего объема переданных данных на время их передачи, причем выбирается достаточно длительный промежуток времени - час, день или неделя.

Мгновенная пропускная способность отличается от средней тем, что для усреднения выбирается очень маленький промежуток времени. Максимальная пропускная способность - это наибольшая мгновенная пропускная способность, зафиксированная в течение периода наблюдения.

На указанные параметры (задержка и пропускная способность) влияют множество факторов, но более всего - маршрутизация и управление потоками, надежность и и устранение ошибок передачи.

Относительно надежности компьтерной сети можно отметить следующее. Здесь различают несколько аспектов надежности. Для технических устройств используются такие показатели надежности, как среднее время наработки на отказ, вероятность отказа, интенсивность отказов. Однако эти показатели пригодны для оценки надежности простых элементов и устройств, которые могут находиться только в двух состояниях - работоспособном или неработоспособном. Сложные системы, состоящие из многих элементов, кроме состояний работоспособности и неработоспособности, могут иметь и другие промежуточные состояния, которые эти характеристики не учитывают. В связи с этим для оценки надежности сложных систем применяется другой набор характеристик. Это- готовность или коэффициент готовности, вероятность доставки пакета узлу назначения без искажений, безопасность, отказоустойчивость.

Чтобы систему можно было отнести к высоконадежным, она должна как минимум обладать высокой готовностью, но этого недостаточно. Необходимо обеспечить сохранность данных и защиту их от искажений. Кроме этого, должна поддерживаться согласованность (непротиворечивость) данных, например, если для повышения надежности на нескольких файловых серверах хранится несколько копий данных, то нужно постоянно обеспечивать их идентичность.

Устранение и исправление ошибок включает в себя выявление, определение и устранение последствий сбоев и отказов в работе сети, выполняется не только регистрация сообщений об ошибках, но и их фильтрация, маршрутизация и анализ.

В результате повышения надежности сети, исправления и устранения ошибок сокращается количество повторно передаваемых пакетов, следовательно, загруженность сети, что положительно влияет на задержку и пропускную способность сети.

Под маршрутизацией понимают выбор непрерывного пути (включающего несколько промежуточных линий сети) между любыми двумя узлами сети (источником и приемником). Алгоритм маршрутизации должен обладать определенными свойствами: правильностью, простотой, надежностью, устойчивостью, справедливостью и оптимальностью. Маршрутизация взаимодействует с управлением потоками в определении характеристик посредством механизма обратной связи. Существует большое количество алгоритмов маршрутизации, в которых достигаются различные цели: сокращение пути, времени.

Управление потоками - это обеспечение мер, регулирующих объем трафика в сети. Задача управления потоками возникает в тех случаях, когда в сети требуется сохранить задержку на определенном низком уровне при повышении нагрузки в сети. что становится возможным только путем отказа в обслуживании некторым пользователям.

Целью диссертационной работы было решение задач управления ресурсами компьютерной сети с целью повышения производительности сети.

Диссертационная работа состоит из введения и пяти глав.

Во введении дается краткий обзор развития компьютерных сетей, а также рассмотрены уровни модели OSI.

В первой главе рассмотрены функциональные группы задач управления компьютерными сетями, которые включают в себя пять групп. Это - управление конфигурацией сети и именованием, обработка ошибок, анализ производительности и надежности, управление безопасностью и учет работы сети.

Во второй главе даны основные понятия маршрутизации. Описаны основные типы алгоритмов маршрутизаций, даны такие параметры, какими являются длина маршрута, надежность, задержка, пропускная способность, нагрузка и стоимость сети. Кроме того, рассмотрен вопрос управления потоками. Также показана взаимосвязь между управлением потоками и маршрутизацией.

В третьей главе приведены общие понятия качества обслуживания, а также ее функции. Рассмотрена маршрутизация с точки зрения качества обслуживания и основные показатели в компьютерных сетях, связанных с маршрутизацией.

В четвертой главе решены основные задачи управления ресурсов. Разработана система массового обслуживания M/M/m с m обслуживающими приборами и M/M/m/m m-потерями и m-обслуживающими приборами. Данные модели дают возможность наиболее точно оценить величину задержки в сети. Приведена методика определения оптимальности маршрутизации и разработаны модели и схемы управления потоками в сочитании с оптимальной маршрутизацией.

ABSTRACT

It is wide-spread to use computer networks in modern society. The spectrum of created and worked computer networks both territorial and local is unusual wide. In the elaboration of computer networks it is appeared the tendency of unification of different territorial and local area networks large quantity. This problem requires a new approach to the networks architecture, models of their effective interaction.

The modern tasks of using computer networks put high requirements to the quality of service (QoS). Number actions can be done in order to raise a quality of service. One of them is resources control. Resources may be considered as hardware and software complex of computer networks. Absolutely, network resources can be considered as network hardware and software (Operating System, Database System Control). But at a point of QoS view delay and capacity are resources.

Delay is called the space of time, which is necessary for the packet movement from the source to the destination point through the network. Delay transference is defined as delay between the time of the packet entering on any input network device or the part of the network and the time of its appearance on the output device. By sense this productivity parameter is nearer to the time of the network reaction, but it differs that it s always characterized only network stages of data processing without delay of computer processing in the network. The quality of network is usually characterized the value of maximum transference delay and delay variation.

Capacity is equal either bit per second or packet per second. Capacity can be momentary, maximum or average.

Average capacity is calculated by division of the whole size of transmitted data on the time of their transmission, it must take note of choosing enough long time interval – an hour, a day or a week. As regards the difference of momentary capacity and average capacity, there is a little time interval is chosen for average. Maximum capacity is the biggest momentary capacity which is fixed during observation period.

A great number of factors have influence on the shown parameters as delay and capacity, but most at all – routing and flow control, safety and removal transfer errors.

As concerns computer network safety it must note several safety aspects. For technical devices there are used such safety parameters as the average working time on rejection, rejection probability, rejection intensity. However these parameters are useful for safety appraisement of simple elements and devices which can only be in two conditions – efficient and disabled. Complex systems which consist of many elements besides efficient and disabled conditions can have other intermediate conditions which not to take into consideration these parameters. That's why the other set of parameters is used for complex systems

safety appraisement. They are readiness or the readiness coefficient, packet delivery probability to the destination node without distortion, security.

To say that system has high safety it must as minimum possess high readiness but it is not enough. It is necessary to provide data safety and their protection from distortions. Besides it must be supported data coordination, for example, if for safety increasing several data copies are kept on some servers, it needs to guarantee their identity constantly.

Correction and elimination of errors includes ascertainment, definition and elimination of refusal consequence in the network. It is not only executed the registration of messages about errors but their filtration, routing and analysis.

As a result of increase of reliability of a network, correction and elimination of errors the quantity of repeatedly transmitted packages is reduced, consequently, network congestion a positively influences a delay and flow control of a network.

As routing it is understood a choice of a continuous way (including some intermediate lines of a network) between any two nodes of a network (the source and the receiver). The algorithm of routing should possess certain properties: correctness, simplicity, reliability, stability, justice and optimality. Routing co-operates with flow control in definition of characteristics by means of a feedback mechanism. There is a considerable quantity of algorithms of routing in which the various purposes are reached: reduction of a way, time.

Flow control is maintenance of the measures regulating volume of the traffic in a network. The problem of flow control arises when in a network it is required to keep a delay at certain low level at loading increase in a network that becomes possible only by refusal in service to some users.

The decision of problems of resource management of a computer network for the purpose of increase of productivity of a network was the purpose of dissertation.

Dissertation consists of introduction and five chapters.

In the introduction it is the short review of development of computer networks is given, and also levels of model OSI are considered.

In chapter one the functional groups of problems of management are considered by computer networks which include five groups. They are configuration management, fault management, performance management, security management and accounting management.

In the second chapter the basic concepts of routing are given. The basic types of algorithms of routings are described; such parameters as the route length, reliability, a delay, capacity, loading and network cost are given. Besides, the question of flow control is considered. Also the interrelation between flow control and routing is shown.

In the third chapter the general concepts of quality of service, and also its function are resulted. Routing from the point of view of quality of service and the basic indicators in the computer networks connected with routing is considered.

In the fourth chapter the primary goals of management of resources are solved. The system of mass service M/M/m with m serving devices and M/M/m/m m-losses and m-serving devices is developed. The given models give the chance to estimate the most precisely delay size in a network. The technique of definition of an optimality of routing is resulted and models and schemes of flow control in combination with optimum routing are developed.

Содержание

Список рисунков

Список абривиатур используемых в диссертации

ISO/OSI	–	Open Systems Interconnection/International Organization for Standartization,
LLC	–	Logical Link Control – управление логическим каналом,
MAC	–	Media Access Control – управление доступом к среде,
QoS	–	Quality of Service – качество обслуживания,
ОС	–	операционная система,
СУБД	–	система управления базой данных,
CRC	–	Cyclical Redundancy Code – циклический избыточный код,
УЛПД	–	управление линией передачи данных,
ЦИП	–	циклические избыточные проверки,
TPDU	–	Transport Protocol Data Unit – модуль данных транспортного уровня ,
SLA	–	Service Level Agreement – согллашение об уровне обслуживания,
FTP	–	File Transfer Protocol – протокол передачи данных,
IP	–	Internet Protocol – протокол интернет,
DSCP	–	Differentiated Service Code Point – поле кода дифференцированной услуги,
URL	–	Universal Resource Locator – универсальный указатель информационного ресурса,
NBAR	–	Network Based Application Recognition – приложения на основе сетевых параметров,
CAR	–	Committed Access Rate – механизм согласования скорости доступа,
QPPB	–	QoS Policy Propagation using Border Gateway Protocol –механизм распространения политик QoS с помощью протокола пограничного поля,
TS	–	Traffic Shaping – функция выравнивания трафика,
TI	–	Time Interval – интервал времени,
FIFO	–	First-in,First-out – первым пришел, первым обслужен,
WFQ	–	Weighted Fair Queuing – взвешенный механизм равномерного обслуживания очередей,
CIR	–	Committed Information Rate – согласованная скорость передачи информации,
TCP	–	Transmission Control Protocol – протокол управления передачей,
GPS	–	Generalized Processor Sharing – разделение процессорного времени,
ARP	–	Address Resolution Protocol – протокол преобразования адресов,
CEF	–	Cisco Express Forwarding – метод коммутации пакетов,
OSPF	–	Open Shortest Path First – протокол выбора кратчайшего пути,
IS-IS	–	Intermediate System-to-Intermadiate System – протокол обмена информацией о маршрутах между промежуточными системами.

Благодарности.

В благодарность моим лекторам Ассист.-Проф. М.Т.Тевдорадзе и

Проф. Н. Н. Ломинадзе

Введение

Концепция вычислительных сетей является логическим результатом эволюции компьютерной технологии. Первые компьютеры 50-х годов - большие, громоздкие и дорогие - предназначались для очень небольшого числа избранных пользователей. Такие компьютеры не были предназначены для интерактивной работы пользователя, а использовались в режиме пакетной обработки.

Системы пакетной обработки, как правило, строились на базе мэйнфрейма - мощного и надежного компьютера универсального назначения. Пользователи подготавливали перфокарты, содержащие данные и команды программ, и передавали их в вычислительный центр. Операторы вводили эти карты в компьютер, а распечатанные результаты пользователи получали обычно только на следующий день.

Конечно, для пользователей интерактивный режим работы, при котором можно с терминала оперативно руководить процессом обработки своих данных, был бы гораздо удобней. Но интересами пользователей на первых этапах развития вычислительных систем в значительной степени пренебрегали, поскольку пакетный режим - это самый эффективный режим использования вычислительной мощности, так как он позволяет выполнить в единицу времени больше пользовательских задач, чем любые другие режимы. Во главу угла ставилась эффективность работы самого дорогого устройства вычислительной машины - процессора, в ущерб эффективности работы использующих его специалистов.

По мере удешевления процессоров в начале 60-х годов появились новые способы организации вычислительного процесса, которые позволили учесть интересы пользователей. Начали развиваться интерактивные многотерминальные системы разделения времени. В таких системах компьютер отдавался в распоряжение сразу нескольким пользователям.

Каждый пользователь получал в свое распоряжение терминал, с помощью которого он мог вести диалог с компьютером. Причем время реакции вычислительной системы было достаточно мало для того, чтобы пользователю была не слишком заметна параллельная работа с компьютером и других пользователей. Разделяя таким образом компьютер, пользователи получили возможность за сравнительно небольшую плату пользоваться преимуществами компьютеризации.

Терминалы, выйдя за пределы вычислительного центра, рассредоточились по всему предприятию. И хотя вычислительная мощность оставалась полностью централизованной, некоторые функции - такие как ввод и вывод данных - стали распределенными. Такие многотерминальные централизованные системы внешне уже были очень похожи на локальные вычислительные сети. Действительно, рядовой пользователь работу за терминалом мэйнфрейма воспринимал примерно так же, как сейчас он воспринимает работу за подключенным к сети персональным компьютером. Пользователь мог получить доступ к общим файлам и периферийным устройствам, при этом у него поддерживалась полная иллюзия единоличного владения компьютером, так как он мог запустить нужную ему программу в любой момент и почти сразу же получить результат. Таким образом, многотерминальные системы, работающие в режиме разделения времени, стали первым шагом на пути создания локальных вычислительных сетей.

Тем не менее потребность в соединении компьютеров, находящихся на большом расстоянии друг от друга, к этому времени вполне назрела. Началось все с решения более простой задачи - доступа к компьютеру с терминалов, удаленных от него на многие сотни, а то и тысячи километров. Терминалы соединялись с компьютерами через телефонные сети с помощью модемов. Такие сети позволяли многочисленным пользователям получать удаленный доступ к разделяемым ресурсам нескольких мощных компьютеров класса суперЭВМ. Затем появились системы, в которых наряду с удаленными

соединениями типа терминал-компьютер были реализованы и удаленные связи типа компьютер-компьютер. Компьютеры получили возможность обмениваться данными в автоматическом режиме, что, собственно, и является базовым механизмом любой вычислительной сети. Используя этот механизм, в первых сетях были реализованы службы обмена файлами, синхронизации баз данных, электронной почты и другие, ставшие теперь традиционными сетевые службы.

Таким образом, хронологически первыми появились глобальные вычислительные сети. Именно при построении глобальных сетей были впервые предложены и отработаны многие основные идеи и концепции современных вычислительных сетей. Такие, например, как многоуровневое построение коммуникационных протоколов, технология коммутации пакетов, маршрутизация пакетов в составных сетях.

В начале 70-х годов произошел технологический прорыв в области производства компьютерных компонентов - появились большие интегральные схемы. Их сравнительно невысокая стоимость и высокие функциональные возможности привели к созданию мини-компьютеров, которые стали реальными конкурентами мэйнфреймов.

Даже небольшие подразделения предприятий получили возможность покупать для себя компьютеры. Мини-компьютеры выполняли задачи управления технологическим оборудованием, складом и другие задачи уровня подразделения предприятия. Таким образом, появилась концепция распределения компьютерных ресурсов по всему предприятию. Однако при этом все компьютеры одной организации по-прежнему продолжали работать автономно.

Но шло время, потребности пользователей вычислительной техники росли, им стало недостаточно собственных компьютеров, им уже хотелось получить возможность обмена данными с другими близко расположенными компьютерами. В ответ на эту потребность предприятия и организации стали

соединять свои мини-компьютеры вместе и разрабатывать программное обеспечение, необходимое для их взаимодействия. В результате появились первые локальные вычислительные сети. Они еще во многом отличались от современных локальных сетей, в первую очередь - своими устройствами сопряжения. На первых порах для соединения компьютеров друг с другом использовались самые разнообразные нестандартные устройства со своим способом представления данных на линиях связи, своими типами кабелей и т.п.

В середине 80-х годов положение дел в локальных сетях стало кардинально меняться. Утвердились стандартные технологии объединения компьютеров в сеть - Ethernet, Arcnet, Token Ring. Мощным стимулом для их развития послужили персональные компьютеры. Эти массовые продукты явились идеальными элементами для построения сетей - с одной стороны, они были достаточно мощными для работы сетевого программного обеспечения, а с другой - явно нуждались в объединении своей вычислительной мощности для решения сложных задач, а также разделения дорогих периферийных устройств и дисковых массивов. Поэтому персональные компьютеры стали преобладать в локальных сетях, причем не только в качестве клиентских компьютеров, но и в качестве центров хранения и обработки данных, то есть сетевых серверов, потеснив с этих привычных ролей мини-компьютеры и мэйнфреймы.

Локальные сети в сравнении с глобальными сетями внесли много нового в способы организации работы пользователей [45]. Доступ к разделяемым ресурсам стал гораздо удобнее - пользователь мог просто просматривать списки имеющихся ресурсов, а не запоминать их идентификаторы или имена. После соединения с удаленным ресурсом можно было работать с ним с помощью уже знакомых пользователю по работе с локальными ресурсами команд. Последствием и одновременно движущей силой такого прогресса стало появление огромного числа

непрофессиональных пользователей, которым совершенно не нужно было изучать специальные (и достаточно сложные) команды для сетевой работы. А возможность реализовать все эти удобства разработчики локальных сетей получили в результате появления качественных кабельных линий связи, на которых даже сетевые адаптеры первого поколения обеспечивали скорость передачи данных до 10 Мбит/с.

Сегодня вычислительные сети продолжают развиваться, причем достаточно быстро. Разрыв между локальными и глобальными сетями постоянно сокращается во многом из-за появления высокоскоростных территориальных каналов связи, не уступающих по качеству кабельным системам локальных сетей. В глобальных сетях появляются службы доступа к ресурсам, такие же удобные и прозрачные, как и службы локальных сетей. Подобные примеры в большом количестве демонстрирует самая популярная глобальная сеть - Internet.

Изменяются и локальные сети. Вместо соединяющего компьютеры пассивного кабеля в них в большом количестве появилось разнообразное коммуникационное оборудование - коммутаторы, маршрутизаторы, шлюзы. Благодаря такому оборудованию появилась возможность построения больших корпоративных сетей, насчитывающих тысячи компьютеров и имеющих сложную структуру [43].

Но дело в том, что для взаимосвязи компьютеров, которые изготовленны различными корпорациями, объединениями, фирмами, и имеющими разные структуры и использующими всевозможные операционные системы требуют нового подхода к архитектуре сетей, моделям их эффективного взаимодействия. Для этого была создана модель ISO/OSI (Open Systems Interconnection/ International Organization for Standartization), которая помогла взаимодействию различных компьютерных сетей и обеспечила совместимость между продуктами разных производителей. Модель ISO/OSI предполагает, что все сетевые приложения

можно подразделить на семь уровней: прикладной, представительный, сеансовый, транспортный, сетевой, канальный и физический [37].

Теоретически, каждый уровень должен взаимодействовать с аналогичным уровнем удаленного компьютера. На практике каждый из них, за исключением физического, взаимодействует с выше или нижележащими уровнями.

Физический уровень имеет дело с передачей битов по физическим каналам связи, таким, например, как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. К этому уровню имеют отношение характеристики физических сред передачи данных, такие как полоса пропускания, помехозащищенность, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, передающих дискретную информацию, например, крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизуются типы разъемов и назначение каждого контакта.

Функции физического уровня реализуются во всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Канальный уровень обеспечивает перенос данных по физической среде. Он поделен на два подуровня: управления логическим каналом (Logical Link Control, LLC) и управления доступом к среде (Media Access Control, MAC) [22,24]. Такое деление позволяет одному уровню LLC использовать различные реализации уровня MAC.

В отличие от канального уровня, имеющего дело с физическими адресами, сетевой уровень работает с логическими адресами. Он обеспечивает подключение и маршрутизацию между двумя узлами. Сетевой уровень предоставляет транспортному уровню услуги с установлением или без установления логического соединения.

На сетевом уровне сам термин “сеть” наделяют специфическим значением. В данном случае под сетью понимается совокупность компьютеров, соединенных между собой в соответствии с одной из стандартных типовых топологий и использующих для передачи данных один из протоколов канального уровня, определенный для этой топологии.

Внутри сети доставка данных обеспечивается соответствующим канальным уровнем, а вот доставкой данных между сетями занимается сетевой уровень, который и поддерживает возможность правильного выбора маршрута передачи сообщения даже в том случае, когда структура связей между составляющими сетями имеет характер, отличный от принятого в протоколах канального уровня.

Протоколом сетевого уровня выполняются следующие функции [3,4,6]:

- формирование блока данных протокола, выполняется в соответствии с существующими правилами кодирования, декомпозиции (эта функция обратная формированию блока данных);
- анализ формата заголовка (проверяется подмножество протокола: бывают два подмножества - холостой протокол и подмножество без сегментации. Холостой протокол используется в том случае, если не нужна ни одна функция сетевого уровня - в простых сетях, а подмножество без сегментации, там где размер блока позволяет не производить сегментацию);
- контроль времени существования блока данных (устанавливается отправителем блока данных);
- сегментация (формирование двух или более блоков данных из полученного);
- сборка (восстановление блока данных из порожденных функцией сегментации);

- сброс блока данных (применяется при нарушении контрольной суммы блока данных; при занятости принимающего компьютера; если анализ заголовка блока данных невозможен; если адрес получателя в блоке данных недоступен; если размер блока данных никоим образом не может быть доведен до требований сети; если время жизни блока данных истекло);
- оповещание об ошибках (передается некоторый блок данных, может передаваться также полный источниковый маршрут, т.е. путь, пройденный от источника);
- обнаружение ошибок в заголовке блока данных (реализуется путем контрольного суммирования всего заголовка блока данных);
- услуги по защите (сюда входит безопасность и защита от несанкционированного доступа- аутентификации источника данных, гарантия конфенденциальности данных и их целостности);
- источниковая маршрутизация (позволяет отправителю задавать маршрут, по которому должен передаваться блок данных);
- запись маршрута (позволяет записать маршрут блока данных по мере его прохождения через промежуточные системы);
- функция сохранения качества обслуживания (обеспечивает сетевые объекты информацией, которая может использоваться для принятия маршрутных решений, если они влияют на общее качество обслуживания пользователей);
- приоритетная обработка (позволяет обрабатывать блоки данных с большим приоритетом по отношению к блокам данных с наименьшим приоритетом);

- оповещание о перегрузке (позволяет пользователю сетевой службы предпринять соответствующие действия при возникновении перегрузки внутри поставщика сетевых услуг).

Транспортный уровень предоставляет услуги аналогичные услугам сетевого уровня. На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Транспортный уровень обеспечивает приложениям или верхним уровням стека - прикладному и сеансовому - передачу данных с той степенью надежности, которая им требуется. Надежность гарантируют лишь некоторые реализации сетевых уровней, поэтому ее относят к числу функций, выполняемых транспортным уровнем. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное - способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

Протоколы нижних четырех уровней обобщенно называют сетевым транспортом или транспортной подсистемой, так как они полностью решают задачу транспортировки сообщений с заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями. Остальные три верхних уровня решают задачи предоставления прикладных сервисов на основании имеющейся транспортной подсистемы.

Сеансовый уровень обеспечивает управление диалогом: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки

в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов, хотя функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Представительный уровень имеет дело с формой представления передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов. На этом уровне может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных служб.

Прикладной уровень - высший в модели ISO/OSI. Прикладной уровень- это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например, с помощью протокола электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется сообщением.

Как уже было сказано выше, вплоть до 80-х годов компьютерные сети пребывали в «зародышевом» состоянии, что характеризоавлось низким объемом трафика и малым числом используемых сетевых приложений. Сети растут, и помимо передачи данных появилась необходимость передавать нетрадиционный тип информации, - например, передачу голоса или передачу трафика видеоконференций. Негарантированная доставка данных препятствует передаче трафика, требующего выделения заданного минимума

сетевых ресурсов и гарантии предоставления определенных услуг. Для разрешения этой проблемы было введено такое понятие, как качество обслуживания (Quality of Service, QoS). Технологии качества обслуживания позволяют максимально оптимизировать производительность сетей и обеспечить стабильное функционирование нового поколения мультимедийных и голосовых приложений.

Функции качества обслуживания в сетях заключается в обеспечении гарантированного и дифференцированного обслуживания сетевого трафика путем передачи контроля за ипользованием ресурсов и загруженностью сети ее оператору. QoS представляет собой набор требований, предъявляемых к ресурсам сети при транспортировке потока данных и обеспечивает сквозную гарантию передачи данных и основанный на системе правил контроль за средствами повышения производительности сети, такими, как механизм распределения ресурсов, коммутация, маршрутизация, механизмы обслуживания очередей и механизмы отбрасывания пакетов.

Существуют два подхода к обеспечению качества обслуживания сети [29,38,42]. Первый подход, очевидно, покажется наиболее естественным с точки зрения пользователя сети. Он состоит в том, что сеть (точнее, обслуживающий ее персонал) гарантирует пользователю соблюдение некоторой числовой величины показателя качества обслуживания. Второй подход состоит в том, что сеть обслуживает пользователей в соответствии с их приоритетами. То есть качество обслуживания зависит от степени привилегированности пользователя или группы пользователей, к которой он принадлежит. Качество обслуживания в этом случае не гарантируется, а гарантируется только уровень привилегий пользователя. Сеть старается по возможности более качественно обслужить пользователя, но ничего при этом не гарантирует.

Качество обслуживания в компьютерных сетях непосредственно связано с управлением ресурсами. Под ресурсами понимают полосу пропускания, задержку и уровень потери пактов.

Полоса пропускания относится к имеющейся мощности трафика какого-либо канала. Термин полоса пропускания используется для описания номинальной пропускной способности среды передачи информации, протокола или соединения.

Под задержкой обычно понимают отрезок времени, необходимый для передвижения пакета от источника до пункта назначения через объединенную сеть. Задержка зависит от многих факторов, включая полосу пропускания промежуточных каналов сети, очереди в порт каждого маршрутизатора на пути передвижения пакета, перегруженность сети и физическое расстояние, на каторое необходимо переместить пакет.

Уровень потери пакетов определяет количество пакетов, отбрасываемых сетью во время передачи. Основными причинами потери пакетов являются перегрузка сети и повреждение пакетов во время передачи по линии связи. Чаще всего отбрасывание пакетов происходит в местах перегрузки, где число поступающих пакетов намного превышает верхнюю границу размера выходной очереди. Кроме того, отбрасывание пакетов может быть вызвано недостаточным размером входного буфера. Как правило, уроовень потери пакетов выражается как доля отброшенных пакетов за определенный интервал времени.

Управление ресурсами связано с управлением потоками и маршрутизацией [10,11].

Несмотря на то, что реализация механизмов QoS в конечных системах является необходимым условием, она не позволяет говорить о сквозном качестве обслуживания до тех пор, пока соответствующие механизмы не будут реализованы в маршрутизаторах - устройствах ответственных за передачу трафика между конечными системами. Следовательно, с 1990-х годов акцент в разработке механизмов QoS вполне логично переместилсся на исследование возможности реализации функций качества обслуживания в маршрутизаторах.

Глава 1. Задачи управления сетями

1.1 Функциональные группы задач управления компьютерными сетями

Любая сложная вычислительная сеть требует дополнительных специальных средств управления помимо тех, которые имеются в стандартных сетевых операционных системах. Это связано с большим количеством разнообразного коммуникационного оборудования, работа которого критична для выполнения сетью своих основных функций. Распределенный характер крупной корпоративной сети делает невозможным поддержание ее работы без централизованной системы управления, которая в автоматическом режиме собирает информацию о состоянии каждого концентратора, коммутатора, мультиплексора и маршрутизатора и предоставляет эту информацию оператору сети. Обычно система управления работает в автоматизированном режиме, выполняя наиболее простые действия по управлению сетью автоматически, а сложные решения предоставляя принимать человеку на основе подготовленной системой информации. Система управления должна быть интегрированной. Это означает, что функции управления разнородными устройствами должны служить общей цели обслуживания конечных пользователей сети с заданным качеством.

Сами системы управления представляют собой сложные программно-аппаратные комплексы, поэтому существует граница целесообразности применения системы управления - она зависит от сложности сети, разнообразия применяемого коммуникационного оборудования и степени его распределенности по территории. В небольшой сети можно применять отдельные программы управления наиболее сложными устройствами, например коммутатором, поддерживающим технику VLAN. Обычно каждое

устройство, которое требует достаточно сложного конфигурирования, производитель сопровождает автономной программой конфигурирования и управления. Однако при росте сети может возникнуть проблема объединения разрозненных программ управления устройствами в единую систему управления, и для решения этой проблемы придется, возможно, отказаться от этих программ и заменить их интегрированной системой управления.

Системы управления корпоративными сетями существуют не очень давно. Одной из первых систем такого назначения, получившей широкое распространение, был программный продукт SunNet Manager, выпущенный в 1989 году компанией SunSoft. SunNet Manager был ориентирован на управление коммуникационным оборудованием и контроль трафика сети. Именно эти функции имеют чаще всего в виду, когда говорят о системе управления сетью. Кроме систем управления сетями существуют и системы управления другими элементами корпоративной сети: системы управления ОС, СУБД, корпоративными приложениями.

Независимо от объекта управления, желательно, чтобы система управления выполняла ряд функций, которые были бы определены международными стандартами, обобщающими опыт применения систем управления в различных областях. Системы управления делятся на пять функциональных групп задач [2]:

- управление конфигурацией сети и именованием;
- обработка ошибок;
- анализ производительности и надежности;
- управление безопасностью;
- учет работы сети.

Рассмотрим задачи этих функциональных областей управления применительно к системам управления сетями.

1.2. Управление конфигурацией сети и именованием.

Эти задачи заключаются в конфигурировании параметров как элементов сети, так и сети в целом. Для элементов сети, таких как маршрутизаторы, мультиплексоры и т. п., с помощью этой группы задач определяются сетевые адреса, идентификаторы (имена), географическое положение и пр.

Для сети в целом управление конфигурацией обычно начинается с построения карты сети, то есть отображении реальных связей между элементами сети и изменении связей между элементами сети - образование новых физических или логических каналов, изменение таблиц коммутации и маршрутизации.

Управление конфигурацией (как и другие задачи системы управления) могут выполняться в автоматическом, ручном или полуавтоматическом режимах. Например, карта сети может составляться автоматически, на основании зондирования реальной сети пакетами-исследователями, а может быть введена оператором системы управления вручную. Чаще всего применяются полуавтоматические методы, когда автоматически полученную карту оператор подправляет вручную. Методы автоматического построения топологической карты, как правило, являются фирменными разработками.

Для структуризации и конфигурации сети используются такие коммуникационные устройства, как мосты, коммутаторы, маршрутизаторы и шлюзы [17,18].

Мост делит разделяемую среду передачи сети на части (часто называемые логическими сегментами), передавая информацию из одного сегмента в другой только в том случае, если такая передача действительно необходима, то есть если адрес компьютера назначения принадлежит другой подсети. Тем самым мост изолирует трафик одной подсети от трафика другой, повышая общую производительность передачи данных в сети. Локализация трафика не только экономит пропускную способность, но и

уменьшает возможность несанкционированного доступа к данным, так как кадры не выходят за пределы своего сегмента и их сложнее перехватить злоумышленнику.

Мосты используют для локализации трафика аппаратные адреса компьютеров. Это затрудняет распознавание принадлежности того или иного компьютера к определенному логическому сегменту - сам адрес не содержит никакой информации по этому поводу. Поэтому мост достаточно упрощенно представляет деление сети на сегменты - он запоминает, через какой порт на него поступил кадр данных от каждого компьютера сети, и в дальнейшем передает кадры, предназначенные для этого компьютера, на этот порт. Точной топологии связей между логическими сегментами мост не знает. Из-за этого применение мостов приводит к значительным ограничениям на конфигурацию связей сети - сегменты должны быть соединены таким образом, чтобы в сети не образовывались замкнутые контуры.

Коммутатор по принципу обработки кадров ничем не отличается от моста. Основное его отличие от моста состоит в том, что он является своего рода коммуникационным мультипроцессором, так как каждый его порт оснащен специализированным процессором, который обрабатывает кадры по алгоритму моста независимо от процессоров других портов. За счет этого общая производительность коммутатора обычно намного выше производительности традиционного моста, имеющего один процессорный блок. Можно сказать, что коммутаторы - это мосты нового поколения, которые обрабатывают кадры в параллельном режиме.

Ограничения, связанные с применением мостов и коммутаторов - по топологии связей, а также ряд других, - привели к тому, что в ряду коммуникационных устройств появился еще один тип оборудования - маршрутизатор. Маршрутизаторы более надежно и более эффективно, чем мосты, изолируют трафик отдельных частей сети друг от друга. Маршрутизаторы образуют логические сегменты посредством явной

адресации, поскольку используют не плоские аппаратные, а составные числовые адреса. В этих адресах имеется поле номера сети, так что все компьютеры, у которых значение этого поля одинаково, принадлежат к одному сегменту.

Кроме локализации трафика маршрутизаторы выполняют еще много других полезных функций. Например, очень важной функцией маршрутизаторов является их способность связывать в единую сеть подсети, построенные с использованием разных сетевых технологий.

Кроме перечисленных устройств отдельные части сети может соединять шлюз. Обычно основной причиной, по которой в сети используют шлюз, является необходимость объединить сети с разными типами системного и прикладного программного обеспечения, а не желание локализовать трафик. Тем не менее шлюз обеспечивает и локализацию трафика в качестве некоторого побочного эффекта.

Крупные сети практически никогда не строятся без логической структуризации. Для отдельных сегментов и подсетей характерны типовые однородные топологии базовых технологий, и для их объединения всегда используется оборудование, обеспечивающее локализацию трафика, - мосты, коммутаторы, маршрутизаторы и шлюзы [21].

Ограничения, связанные с применением мостов и коммутаторов - по топологии связей, а также ряд других, - привели к тому, что в ряду коммуникационных устройств появился еще один тип оборудования - маршрутизатор. Маршрутизатор - это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. Чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач между сетями, каждый раз выбирая подходящий

маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет.

Маршрутизаторы более надежно и более эффективно, чем мосты, изолируют трафик отдельных частей сети друг от друга. Маршрутизаторы образуют логические сегменты посредством явной адресации, поскольку используют не плоские аппаратные, а составные числовые адреса. В этих адресах имеется поле номера сети, так что все компьютеры, у которых значение этого поля одинаково, принадлежат к одному сегменту, называемому в данном случае подсетью.

Кроме локализации трафика маршрутизаторы выполняют еще много других полезных функций. Так, маршрутизаторы могут работать в сети с замкнутыми контурами, при этом они осуществляют выбор наиболее рационального маршрута из нескольких возможных.

Другой очень важной функцией маршрутизаторов является их способность связывать в единую сеть подсети, построенные с использованием разных сетевых технологий.

Проблема выбора наилучшего пути называется маршрутизацией, и ее решение является одной из главных задач сетевого уровня [20]. Эта проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени. Некоторые алгоритмы маршрутизации пытаются приспособиться к изменению нагрузки, в то время как другие принимают решения на основе средних показателей за длительное время. Выбор маршрута может осуществляться и по другим критериям, например, надежности передачи.

Более сложной задачей является настройка коммутаторов и маршрутизаторов на поддержку маршрутов и виртуальных путей между пользователями сети.

Сущетвуют две принципиально различные философии организации компьютерной сети, одна с использованием соединений, а другая - без соединений. В контексте внутреннего устройства сети соединение обычно называют виртуальным каналом, по аналогии с физическими каналами, устанавливаемым телефонной станцией. Независимые пакеты в системе без установления соединений называются дейтаграммами, по аналогии с телеграммами.

Виртуальные каналы обычно используются в сетях, чья основная служба является ориентированной на соединение. Идея, стоящая за виртуальными каналами, состоит в том, что для каждого посылаемого пакета не нужно выбирать маршрут заново [27]. Вместо этого маршрут от отправителя к получателю выбирается и запоминается при установке соединения. Этот маршрут используется для всех данных, посылаемых и принимаемых за время соединения. При разрыве соединения виртуальный канал также перестает существовать.

В дейтаграммной подсети, напротив, маршруты не выбираются заранее, даже если служба является ориентированной на соединение. Маршрут каждого пакета выбирается независимо от его предшественников. Следующие один за другим пакеты могут перемещаться по различным маршрутам. Дейтаграммным сетям приходится выполнять больше работы, но зато они отличаются большей устойчивостью и легче приспосабливаются к неисправностям и заторам, чем сети виртуальных каналов [26].

Если пакеты, передаваемые по виртуальному каналу, всегда перемещаются по одному и тому же маршруту, каждый маршрутизатор должен помнить, куда направлять пакеты для каждого из открытых в данный момент виртуальных каналов. Каждый маршрутизатор должен хранить таблицу, записи которой соответствуют виртуальным каналам, проходящим через этот маршрутизатор. Каждый пакет, проходящий по сети, должен, помимо порядковых номеров, контрольной суммы и т.п., содержать в своем

заголовке поле номера виртуального канала. Когда пакет прибывает на маршрутизатор, последений знает номер виртуального канала пакета и по какой линии он прибыл. Этой информациии должно быть достаточно для отправки пакета в правильном направлении.

При установке сетевого соединения выбирается неиспользуемый номер виртуального канала. Поскольку номера виртуальных каналов выбираются независимо, эти номера имеют лишь местное значение. Если бы эти номера были глобальными, прохождение двух виртуальных каналов с одинаковыми номерами через один и тот же маршрутизатор могло бы привести к неопределенности.

Нужно обратить внимание на то, что от каждого процесса требуется, чтобы он сообщал, когда он закончил пользоваться виртуальным каналом. Это необходимо для того, чтобы маршрутизаторы могли удалить неиспользуемый виртуальный канал из своих таблиц.

В случае дейтаграммной службы маршрутизаторы пользуются таблицами, содержащими не виртуальные каналы, а номера выходных линий для пунктов назначения пакетов. Эти таблицы применяются также и при использовании виртуальных каналов на стадии определения маршрута во время установки соединения.

Каждая дейтаграмма должна содержать полный адрес получателя. Когда поступает пакет, маршрутизатор выбирает линию, по которой его послать дальше.

Согласованная ручная настройка таблиц маршрутизации при полном или частичном отказе от использования протокола маршрутизации (а в некоторых глобальных сетях, например X.25 [41], такого протокола просто не существует) представляет собой сложную задачу. Многие системы управления сетью общего назначения ее не выполняют, но существуют специализированные системы конкретных производителей, например,

система NetSys компании Cisco Systems, которая решает ее для маршрутизаторов этой же компании.

1.3 Обработка ошибок

Эта группа задач включает выявление, определение и устранение последствий сбоев и отказов в работе сети. На этом уровне выполняется не только регистрация сообщений об ошибках, но и их фильтрация, маршрутизация и анализ на основе некоторой корреляционной модели. Фильтрация позволяет выделить из весьма интенсивного потока сообщений об ошибках, который обычно наблюдается в большой сети, только важные сообщения, маршрутизация обеспечивает их доставку нужному элементу системы управления, а корреляционный анализ позволяет найти причину, породившую поток взаимосвязанных сообщений (например, обрыв кабеля может быть причиной большого количества сообщений о недоступности сетей и серверов).

Устранение ошибок может быть как автоматическим, так и полуавтоматическим. В первом случае система непосредственно управляет оборудованием или программными комплексами и обходит отказавший элемент за счет резервных каналов и т. п. В полуавтоматическом режиме основные решения и действия по устранению неисправности выполняют люди, а система управления только помогает в организации этого процесса - оформляет квитанции на выполнение работ и отслеживает их поэтапное выполнение (подобно системам групповой работы).

В этой группе задач иногда выделяют подгруппу задач управления проблемами, подразумевая под проблемой сложную ситуацию, требующую для разрешения обязательного привлечения специалистов по обслуживанию сети.

Вследствии особенностей физических процессов, порождающих их, ошибки в некоторых типах носителей чаще бывают не единичными, а групповыми. В этом есть как положительные, так и отрицательные стороны. Положительной чертой является то, что ошибки затрагивают лишь немногие блоки данных [46]. Например, блок размером в 1000 бит при вероятности ошибки 0,001 на бит. Если бы ошибки были независимыми, то очень большой процент блоков содержал бы ошибки. Однако если ошибки приходят пакетами, искажая по 100 бит подряд, то из 100 блоков будут испорчены в среднем только один или два блока. Неудобством группирования ошибок является то, что их значительно труднее обнаружить и исправить, чем изолированные ошибки.

Для борьбы с ошибками было создано две основные стратегии [55]. Каждый метод основывается на добавлении к передаваемым данным некоторой избыточной информации. В одном случае этой информации должно быть достаточно, чтобы с большой вероятностью обнаружить наличие искаженных битов. В другом случае избыточной информации должно быть достаточно даже для того,. чтобы востановить поврежденный блок данных. Коды, применяемые в обоих случаях, называются помехоустойчивыми. Коды, позволяющие исправлять ошибки, называются корректирующими или кодами с исправлением ошибок.

Чтобы понять, как могут обнаруживаться и исправляться ошибки, необходимо рассмотреть подробнее, что представляет собой ошибка. Обычно кадр состоит из m бит данных (то есть информационных) и r избыточных или контрольных битов. Пусть подная длина кадра равна n (то есть $n=m+r$). Набор из n бит, содержащий информационные или контрольные биты, часто называют n-битовым кодовым словом или кодовой комбинацией.

Если рассмотреть два кодовых слова, например 10001001 и 10110001, можно определить число отличающихся в них соответствующих разрядов. В данном примере отличаются три бита. Для нахождения этого числа нужно

сложить два кодовых слова по модулю 2 (операция «исключающее или») и сосчитать количество единиц в результате. Количество бит, которыми отличаются два кодовых слова, называется кодовым расстоянием (или расстоянием между кодовыми комбинациями в смысле Хэмминга). Смысл этого числа в том, что если два кодовых слова находятся на кодовом расстоянии d, то для преобразования одного кодового слова в другое понадобится d ошибок в одиночных битах.

В большинстве приложений передачи данных все 2^m возможных сообщений являются допустимыми, однако благодаря использованию контрольных битов не все 2^n возможных кодовых слов используются. Построив полный список всех допустимых кодовых слов, можно найти такую пару кодовых слов, кодовое расстояние между которыми будет минимальным. Это расстояние называется минимальным кодовым расстоянием (расстоянием всего кода в смысле Хэмминга) [23].

Способности кода к обнаружению и исправлению ошибок зависят от его минимального кодового расстояния. Для обнаружения d ошибок в одном кодовом слове со стопроцентной гарантией необходим код с минимальным кодовым расстоянием, равным d+1,. поскольку d однобитовых ошибок не смогут изменить одну допустимую комбинацию так, чтобы получилась другая допустимая комбинация. Когда премник встречает запрещенную кодовую комбинацию, он понимает, что при передаче произошла ошибка. Аналогично, для возможности исправления d ошибок в одном кодовом слове требуется код с минимальным кодовым расстоянием, равным $2d+1$, так как в данном случае даже при d однобитовых ошибках результат окажется ближе к исходному кодовому слову, чем к любому другому, и, следовательно, его можно будет однозначно восстановить.

В качестве простейшего примера кода с обнаружением ошибок рассмотрим код, в котором к данным добавляется один бит четности. Бит четности выбирается таким образом, чтобы количество единиц во всем

кодовом слове было четным (или нечетным). Например, при посылке числа 10110101 с добавлением бита четности в конце оно становится равным 101101011, тогда как 10110001 преобразуется в 101100010. Код с единственным битом четности имеет кодовое расстояние, равное 2, так как любая однократная ошибка в любом разряде образует кодовое слово с неверной четностью. Такой код может использоваться для обнаружения однократных ошибок.

В качестве простейшего примера корректирующего кода рассмотрим код, у которого есть всего четыре допустимые кодовые комбинации:

0000000000, 0000011111, 1111100000 и 1111111111.

Этот код имеет расстояние, равное 5, что означает, что он может исправлять двойные ошибки. Если приемник получит кодовое слово 0000000111, он поймет, что оригинал должен быть равен 0000011111. Однако если тройная ошибка изменит 0000000000 в 0000000111, ошибка будет исправлена неверно.

Попробуем создать код, состоящий из m информационных и r контрольных битов, способный исправлять одиночные ошибки. Каждому из 2^m допутимых сообщений будет соответствовать n недопустимых кодовых слов, отстоящих от сообщения на расстояние 1. Их можно получить инвертированием каждого из n битов n -битового кодового слова. Таким образом, каждому из 2^m допустимых сообщений должны соответствовать $n+1$ кодовых комбинаций. Поскольку общее количество возможных кодовых комбинаций равно 2^n, получается, что $(n+1)2^m \le 2^m$. Так как $n=m+r$, это требование может быть преобразовано к виду: $(m+r+1) \le 2^r$. При заданном m данная формула описывает нижний предел для требуемого количества контрольных битов для возможности исправления одиночных ошибок.

Коды Хэмминга могут исправлять только одиночные ошибки. Однако не очень хитрый трюк позволяет исправлять при помощи этого кода и пакеты ошибок. Для этого последовательность k кодовых слов организуется в виде матрицы, по одному кодовому слову в ряду. Обычно данные передаются по кодовым словам, слева направо. Чтобы иметь возможность исправления пакетов ошибок, данные из этой таблицы следует передавать по столбцам. В этом случае, если на блок данных наложится пакет ошибок, инвертирующий k соседних битов, она затронет не более одного бита в каждом кодовом слове. А поскольку код Хэмминга может исправлять одиночные ошибки, то можно будет восстановить весь блок. Данный метод использует kr проверочных битов, благодаря которым блок из km бит данных может выдержать один пакет ошибок длиной не более k бит.

Коды, исправляющие ошибки, иногда используются при передаче данных, например, где нельзя запросить повторную передачу. Однако чаще более эффективным оказывается обнаружение ошибки с последующей повторной передачей. Например, рассмотрим канал с изолированными ошибками, возникающими с вероятностью 10^{-6} на бит. Пусть блок данных соостоит из 1000 бит. Для создания кода, корректирующего однократные ошибки, потребуется 10 дополнительных битов на блок. Для мегабита данных это составит 10000 проверочных битов. Чтобы просто обнаруживать одиночную 1-битовую ошибку, достаточно одного бита четности на блок. На каждые 1000 блоков придется переслать повторно в среднем еще один блок. Таким образом, суммарные накладные расходы такого метода составят всего 2001 бит на мегабит данных против 10000 битов, необходимых для минимального кодового расстояния.

Если к блоку добавлять всего один бит четности, то в случае пакета ошибок вероятность обнаружения ошибки будет всего лишь 0,5, что абсолютно неприемлемо. Этот недостаток может бять исправлен, если рассматривать каждый посылаемый блок как прямоугольную матрицу n бит

шириной и k бит высотой. Бит четности должен вычисляться отдельно для каждого столбца и добавляться к матрице в виде последнего ряда. Затем матрица передается по рядам. Приемник проверяет все биты четности. Если хотя бы один из них неверен, он запрашивает повторную отсылку всего блока.

Такой метод позволяет обнаружить одиночный пакет ошибок длиной не более n, так как в этом случае будет изменено не более одного бита в каждом столбце. Однако пакет ошибок длиной n+1 не будет обнаружен, если будут инвентированы первый и последний биты, а все остальные биты останутся неизмененными. Если блок подвергнется при передаче длинному пакету ошибок или нескольким одиночным ошибкам, вероятность того, что четность любого из n столбцов будет верной (или неверной), равна 0,5, поэтому вероятность необнаружения ошибки будет равна 2^{-n}.

Хотя приведенная выше схема может в некоторых случаях быть приемлемой, тем не менее на практике широко используется другой метод: полиномиальный код, также известный как CRC (Cyclical Redundancy Code-циклический избыточный код). В основе полиномиальных кодов лежит представление битовых строк в виде многочленов с коэффициентами, равными 0 или 1. Кадр из k бит рассматривается как список коэффициентов многочлена степени k-1, состоящего из k членов от x^{k-1} до x^0. Старший (самый левый) бит кадра соответствует коэффициенту при x^{k-1}, следующий бит- коэффициенту при x^{k-2} и т.д. Например, число 110001 состоит из 6 бит и, следовательно, представляется в виде многочлена пятой степени с коэффициентами 1, 1, 0, 0, 0 и 1: x^5+x^4+1.

1001 1011	0011 0011	1111 0000	0101 0101
+1100 1010	+1100 1101	-1010 0110	-1010 1111
0101 0001	1111 1110	0101 0110	1111 1010

Рис. 1.1 Осуществление арифметических действий по модулю 2.

С данными многочленами осуществляются арифметические действия по модулю 2 в соответствии с алгебраической теорией поля. При этом вычитание не отличается от сложения, перенос в следующий или предыдущий разряд не производится (рис. 1.1).

Деление чисел осуществляется так же, как и деление обычных двоичных чисел, с той разницей, что вычитание производится по модулю 2.

При использовании циклического кода отправитель и получатель должны сначала договориться насчет образующего многочлена, *G(x).* Старший и младший биты образующего многочлена должны быть равны 1. При этом идея добавления контрольной суммы к некоему кадру представляется в виде добавления к этому кадру в конец такой последовательности бит, чтобы получившийся многочлен делился на образующийся многочлен *G(x)* без остатка. Получатель, приняв кадр, содержащий контрольную сумму, пытается разделить его на *G(x)*. Ненулевой остаток от деления означает ошибку.

В течение десятилетий предполагалось, что кадры, контрольные суммы которых вычисляются, содержат случайные биты. Все исследования алгоритмов подсчета контрольных сумм основывались на этом предположении. Недавние исследования реальных данных показали, что эти предположения были совершенно неверными. Как следствие, при некоторых обстоятельствах вероятность прохождения необнаруженных ошибок оказалось значительно выше, чем считалось раннее.

1.3.1. Однобитовые проверки на четность

Простейший способ обнаружения ошибки- это добавление одного бита, который называется битом проверки на четность, к последовательности битов данных. Этот бит проверки на четность имеет значение 1, если число единиц в последовательности битов нечетное, и значение 0- в противном

случае (рис. 1.3.1.). Другими словами, бит проверки на четность равен сумме по модулю 2 значений битов в исходной последовательности битов (k по модулю j, где k - целое и j - положительное целое, равное целому числу m, $0 \le m < j$, такому, что $k - m$ делится на j).

s_1	s_2	s_3	s_4	s_5	s_6	s_7	c
1	0	1	1	0	0	0	1

Рис.1.3.1. Однобитовая проверка на четность.

Нужно отметить, что общее число единиц в закодированной последовательности (т.е. в исходной последовательности плюс добавочный бит проверки на четность) всегда четное. Если закодированная последовательность передается и при передаче возникает единственная ошибка из-за перехода 0 в 1 или 1 в 0, то число единиц в последовательности становится нечетным и ошибка может быть обнаружена приемником. Также нужно отметить, что приемнику неизвестно, какой бит ошибочный и сколько ошибок возникло; имеется только информация, что ошибки возникли, так как число единиц нечетно.

При любой длине последовательности битов однобитовая проверка на четность позволяет обнаружить любую единичную ошибку в закодированной последовательности. К сожалению, две ошибки в закодированной последовательности всегда восстанавливают четность числа единиц, так что две ошибки обнаружить невозможно. В общем случае нечетное число ошибок обнаруживается, а любое четное число не обнаруживается.

Несмотря на привлекательную простоту однобитовой проверки на четность, она не пригодна для надежного обнаружения ошибок; во многих ситуациях она обнаруживает ошибки только примерно у каждой второй закодированной последовательности, содержащей ошибки. Существуют две причины такого плохого показателя работы. Первая состоит в том, что многие

модемы преобразуют несколько битов в один отсчет входного сигнала, и ошибка при приеме такого отсчета обычно порождает ошибки в нескольких битах. Второй причиной является то, что многие виды шума, порождают длинные пакеты ошибок. По двум этим причинам при возникновении одной или более ошибок в закодированной последовательности четное число ошибок почти также вероятно, как и нечетное число, и однобитовая проверка на четность является неэффективной.

1.3.2. Проверки на четность по вертикали и по горизонтали

Другой простой интуитивный подход к обнаружению ошибок состоит в том, что последовательность битов данных перестраивают в двумерный массив (рис.1.3.2.) и вычисляют биты проверки на четность для каждой строки и каждого столбца. Бит проверки на четность в правом нижнем углу может относиться к строке или столбцу битов проверки на четность или же к массиву данных. Если четное число ошибок находится в пределах одной строки, то каждую из них можно обнаружить с помощью соответствующего бита проверки на четность столбцу; аналогично ошибки только в одном столбце можно обнаружить с помощью битов проверки на четность по строкам. К сожалению, любой набор из четырех ошибок, принадлежащих двум строкам и двум столбцам (т.е. образующих прямоугольник, как показано на рис.4.2., б), обнаружить невозможно.

Эта схема чаще всего используется в случае, когда данные состоят из последовательности символов. Каждый закодированный символ можно считать строкой массива на рис.1.3.2; бит проверки на четность по строке в этом случае является просто последним битом закодированного символа. Биты проверки по столбцам тривиально вычисляются программным и аппаратным способом. Основной недостаток этой схемы заключается в том, что она допускает сбои при обнаружении довольно коротких пакетов

ошшибок (например, ошибок в двух соседних битах, в каждой из двух соседних строк). Поскольку идущие друг за другом ошибки довольно часто встречаются на практике, вероятность подобных сбоев недопустимо высока.

1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	0
0	0	1	1	0	0	1	1
1	0	1	1	1	1	1	0

Проверки по горизонталям

1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	**1**	0	0	**0**	1	0
1	0	0	0	1	1	1	0
0	0	**1**	1	0	**0**	1	1
1	0	1	1	1	1	1	0

Проверки по горизонталям

Проверки по вертикалям

(б)

Рис.1.3.2. Проверки на четность по горизонтали и вертикали. Каждый бит проверки на четность по горизонтали проверяет свою строку, а каждый бит проверки на четность по вертикали проверяет свой столбец. Заметим, что в случае (б), если каждый из отмеченных битов изменяет значение, то четность в каждом столбце и каждой строке тем не менее сохраняется.

Привлекательной особенностью проверок на четность по горизонтали и вертикали является еще и то, что лежащая в их основе идея легко распространяется на случай произвольных кодов с проверкой на четность. Эта идея сводится к тому, что в исходной последовательности битов (массив битов данных в случае, показанных на рис.1.3.2.) производятся проверки на четность различных подмножеств битов (строк и столбцов в случае на рис. 1.3.2б). Преобразование последовательности битов данных в последовательность битов данных и битов проверки на четность называется кодом с проверкой на четность или линейным кодом. Пример кода с проверкой на четность (отличного от проверок по горизонтали и вертикали) показан на рис. 1.3.3. Код с проверкой на четность определяется конкретным набором подмножеств, используемых для выполнения проверок на четность. Нужно отметить, что слово код относится к самому преобразованию;

закодированную последовательность битов (данные плюс проверки на четность)называют кодовыи словом.

Пусть K обозначает длину последовательности данных заданного кода с проверкой на четность и пусть L - число битов проверки на четность. Итак, $K+L$ есть длина кадра, которая считается постоянной. При заданном коде каждая из 2^K возможных последовательностей данных длины K отображается на кадр (т.е. кодовое слово) длины $K+L$. В системе с обнаружением ошибок при передаче кадра принимающий модуль УЛПД определяет, каждый ли бит проверки на четность равен по-прежнему сумме по модулю 2 соответсвующего подмножества битов данных. Если это так, то кадр считается свободным от ошибок, в противном случае обнаруживается наличие ошибок. Если ошибки при передаче по сети превратили одно кодовое слово в другое, то кадр считается свободным от ошибок, в этом случае говорят, что кадр слдержит необнаруживаемые ошибки.

Ниже приведены трудности по нахождению вероятности необнаруживаемых ошибок в кадре при передаче. Во-первых, ошибки в сетях появляются не независимо и в виде пакетов; не существует хороших моделей для длины и интенсивности этих пакетов, которые меняются в широком диапазоне для сетей одного и того же типа. Во-вторых, при любом разумном коде частота необнаруживаемых ошибок очень мала и, слеовательно, ее трудно измерить экспериментально и к тому же она зависит от редких, трудномоделируемых событий.

В следствии этих трудностей эффективность кода по обнаружению ошибок обычно измеряется тремя параметрами:

1 минимальное расстояние кода,

2 способность обнаружения пакетов ошибок,

3 вероятность того, что совершенно случайная последовательность будет воспринята как свободная от ошибок.

Минимальное расстояние кода определяется как наименьшее число ошибок, которое может превратить одно кодовое слово в другое. Как уже было установлено, минимальное расстояние кода с однобитовой проверкой на четность равно 2, а минимальное расстояние кода с проверками на четность по горизонтали и вертикали равно 4.

Длина пакета ошибок называется число ошибочных бит, считая от первой ошибки до посследней включительно. Способность кода обнаруживать пакеты ошибок определяется как наибольшее целое B, такое, что код может обнаруживать все пакеты длины B или меньше. У кода с однобитовой проверкой на четность способность обнаруживать пакеты равна 1, тогда как у кода с проверками на четность по горизонтали и вертикали способность обнаруживать пакеты равна 1 плюс длина строки (предполагается, что передача данных производится по строкам).

Под совершенно случайной последовательностью длины $K+L$ подразумевается такая последовательность длины $K+L$, которая принимается с вероятностью 2^{-K-L}. Так как число кодовых слов равно 2^K, то вероятность необнаруженной ошибки представляет собой вероятность того, что случайная последовательность совпадает с одим из кодовых слов; это имеет место с вероятностью 2^{-L} (вероятность того, что принятая случайная последовательность окажется такой же, как и переданный кадр, не учитывается). Обычно это дает чорошую оценку вероятности необнаруживаемых ошибок для кадра, в котором ошибки начительно превышают как минимальное расстояние, так и способность кода обнаруживать пакеты ошибок.

Коды с проверкой на четность могут использоваться не только для обнаружения ошибок, но и для исправления ошибок. Например, в случае проверок на четность по горизонтали и вертикали любая одиночная ошибка может быть исправлена посредством простого поиска той строки и того столбца, у которых проверка дает нечетный результат.

s_1	s_2	s_3	c_1	c_2	c_3	c_4	
1	0	0	1	1	1	0	
0	1	0	0	1	1	1	$c_1 = s_1 + s_3$
0	0	1	1	1	0	1	$c_2 = s_1 + s_2 + s_3$
1	1	0	1	0	0	1	$c_3 = s_1 + s_2$
1	0	1	0	0	1	1	$c_4 = s_2 + s_3$
1	1	1	0	1	0	0	
0	0	0	0	0	0	0	

Рис.1.3.3. Пример кода с проверкой на четность. Кодовые слова приведены слева, а правила вычислений битов проверки на четность - справа.

Циклические избыточные проверки

В настоящее время кодами с проверкой на четность, используемыми в компьютерных сетях, являютя коды с циклическими избыточными проверками (ЦИП). Биты проверки на четность называются ЦИП. Пусть L длина ЦИП (т.е. число проверочных битов), а K - длина последовательности битов данных (т.е. заголовка и пакета в кадре). Удобно обозначать биты данных через s_{K-1}, s_{K-2}, ..., s_1, s_0 и представлять последовательность в виде многочлена $s(D)$ с коэффициентами s_{K-1}, ..., s_0,

$$s(D) = s_{K-1}D^{K-1} + s_{K-2}D^{K-2} + \ldots + s_0. \qquad (1.3.1)$$

Можно считать, что степени переменной D сохраняют порядок битов; предполагается, что передача начинается с членов старшего порядка. ЦИП представляется другим многочленом

$$c(D) = c_{L-1}D^{L-1} + \ldots + c_1 D + c_0. \qquad (1.3.2)$$

Весь кадр из передаваемой информации и ЦИПа можно в этом случае представить как $x(D) = s(D)D^L + c(D)$, т.е. как

$$x(D) = s_{K-1}D^{L+K-1} + \ldots + s_0 D^L + c_{L-1}D^{L-1} + \ldots + c_0 \qquad (1.3.3)$$

Многочлен $c(D)$, представляющий ЦИП, является функцией информационного многочлена $s(D)$; эта функциональная связь определяется

порождающим многочленом $g(D)$; это многочлен степени L с двоичными коэффициентами, который задает конкретный код с ЦИП,

$$g(D) = D^L + g_{L-1}D^{L-1} + \ldots + g_1 D + 1. \qquad (1.3.4)$$

При заданном $g(D)$ отображение информационного многочлена в многочлен $c(D)$, представляющий ЦИП, задается равенством

$$c(D) = \text{остаток}\left[s(D)D^L / g(D)\right]. \qquad (1.3.5)$$

Операция деления многочленов, использованная выше, является обычным делением одного многочлена на другой за исключением того, что коэффициенты принимаются только двоичные значения,а арифметические операции над коэффициентами выполняются по модулю 2. Таким образом, например, (1+1)mod2=0, а (0-1)mod2=1. Нужно отметить, что вычитание в арифметике по модулю 2 эквивалентно сложению. Приведем пример выполнения операций в (1.3.5)

$$\begin{array}{rl} & D^2 + D \\ D^3 + D^2 + 1 \quad & D^5 + \qquad D^3 \\ & D^5 + D^4 + \qquad D^2 \\ & \\ & D^4 + D^3 + D^2 \\ & D^4 + D^3 + \qquad D \\ & \\ & D^2 + D = \text{остаток.} \end{array}$$

Поскольку степень многочлена $g(D)$ не превышает L, то степень остатка не превышает $L-1$. Если степень $c(D)$ меньше, чем $L-1$, то соответствующие старшие коэффициенты $c_{L-1,\ldots}$ в (1.3.5) полагаются равными нулю.

Это деление легко реализуется аппаратно при помощи цепи с регистром сдвига и обратными связями [34]. Сравнивая цепь с приведенным выше примером деления, можно убедиться, что последовательные значения разрядов регистра сдвига совпадают с коэффициентами промежуточных

остатков, появляющихся в процессе деления. На практике ЦИП обычно вычисляется в СБИС, которые часто выполняют также другие функции УЛПД.

Пусть $z(D)$ обозначает частное от деления $s(D)D^L$ на $g(D)$. Тогда $c(D)$ можно представить следующим образом:

$$s(D)D^L = g(D)z(D) + c(D). \qquad (1.3.6)$$

Вычитая $c(D)$ (по модулю 2) из обеих частей этого равенства и вспоминая, что вычитание по модулю 2 эквивалентно сложению, получаем

$$x(D) = s(D)D^L + c(D) = g(D)z(D). \qquad (1.3.7)$$

Таким образом, все кодовые слова делятся на $g(D)$ и все многочлены, делящиеся на $g(D)$ являются кодовыми словами. Пока еще не показано, что отображение $s(D)$ и $x(D)$ соответствует некоторому коду с проверкой на четность.

Теперь предположим, что $x(D)$ передается, а принятая последовательность представляет собой многочлен $y(D)$, где $y(D)$ отличается от $x(D)$ из-за ошибок в сети. Если последовательность ошибок представляется многочленом $e(D)$, тогда $y(D) = x(D) + e(D)$, где знак + здесь обозначает сложение по модулю 2; каждой ошибке в кадре соответствует ненулевой коэффициент $e(D)$, т.е. несовпадение коэффициентов $y(D)$ и $x(D)$. Можно вычислить REM $[y(D)/g(D)]$ (т.е. остаток от деления) при помощи цепи, оторая, по существу, аналогична приведенной выше. Так как было показано, что $x(D)$ делится на $g(D)$, то

$$\text{REM}\,[y(D)/g(D)] = \text{REM}\,[e(D)/g(D)]. \qquad (1.3.8)$$

Если ошибки не возникли, то $e(D)$=0 и остаток будет равен нулю. Правило, которое используется в приемнике, состоит в том, что кадр не содержит ошибок, если этот остаток равен нулю, и содержит ошибки в противном случае. Может случиться, что ошибки действительно возникают (т.е. $e(D) \neq 0$), а в приемнике не удается их обнаружить из-за того, что этот

остаток равен 0; это имеет место только тогда, когда $e(D)$ является некоторым кодовым словом. Другими словами, $e(D) \neq 0$ не обнаруживается только и только тогда,когда

$$e(D) = g(D)z(D) \qquad (1.3.9)$$

При некотором ненулевом многочлене $z(D)$. Теперь исследуем условия, при которых могут возникать необнаруживаемые ошибки.

Сначала предположим, что возникла одиночная ошибка, например, $e_i = 1$, так что $e(D) = D^i$. Так как $g(D)$ имеет по крайней мере ва ненулевых члена (т.е. D^L и 1), то произведение $g(D)z(D)$ должно также иметь по крайней мере два ненулевых члена при любом ненулевом $z(D)$. Итак, $g(D)z(D)$ не может быть равно D^i; поскольку это справедливо пр любом i, то все одиночные ошибки обнаруживаются. Поскольку разница между степенями самого старшего и самого младшего членов в $g(D)$ (т.е. D^L и 1 соответственно) равна L, то степени самого старшего и самого младшего членов $g(D)z(D)$ отличаются по крайней мере на L при любом ненулевом $z(D)$. Следовательно, если $e(D)$ является кодовым словом, то длина пакета ошибок не меньше $L+1$ (+1 возникает из-за определения длины пакета ошибок, как числа позиций начиная с первой ошибки и кончая последней ошибкой включительно).

Далее предположим, что возникла двойная ошибка, например, в позициях i и j, так что

$$e(D) = D^i + D^j = D^j\left(D^{i-1} + 1\right),\ i > j\,. \qquad (1.3.10)$$

Выше было установлено, что D^j не делится на $g(D)$ или любой множитель $g(D)$; поэтому $e(D)$ не удается обнаружить, если только $D^{i-j}+1$ делится на $g(D)$. Для произвольного двоичного многоччлена $g(D)$ степени L существтвует наименьшее n, для которого D^n+1 делится на $g(D)$. Из теории конечных полей известно, что это наименьшее n может быть не больше чем $2^L - 1$; более того, для любого $L > 0$ существуют специальные

многочлены степени L, называемые примитивными многочленами, для которых это наименьшее n равно $2^L - 1$. Итак, если в качестве $g(D)$ использовать такой примитивный многочлен степени L и если длина кадра ограничена сверху величиной $2^L - 1$, то $D^{i-j} + 1$ не может делится на $g(D)$; следовательно, все двойные ошибки обнаруживаются.

Фактически на практике для вычисления ЦИП обычно используется порождающий полином $g(D)$, который является произведением примитивного многочлена $D+1$. Следовательно, у любого кода такого типа минимальное расстояние не меньше 4, способность обнаруживать пакеты ошибок не меньше L, а вероятность необнаруужения ошибок в совершенно случайной последовательности равна 2^{-L}.

1.4. Анализ производительности и надежности

Вопросы производительности являются очень важными в компьютерных сетях. Когда сотни или тысячи компьютеров соединены вместе, их взаимодействие становится очень сложным и может привести к непредсказуемым последствиям. Часто эта сложность приводит к низкой производительности, причины которой довольно трудно определить.

К сожалению, понимание производительностей сетей представляет собой скорее искусство, нежели науку. Теоретическая база, допускающая хоть какое- либо практическое применение, очень мала. Лучшее, что можно сделать- это предоставит несколько приблизительных методов, полученных в результате напряженных экспериментов.

Следует отметить пять аспектов производительности сети [47].

1. Причины снижения производительности.
2. Измерение производительности сети.
3. Проектирование производительных систем.
4. Быстрая обработка TPDU- модулей.

5. Протоколы для будущих высокопроизводительных сетей.

Одной из причин снижения производительности являются заторы, вызываемые временной перегрузкой ресурсов. Если на маршрутизатор вдруг прибывает больше трафика, чем он способен обработать, создается затор и производительность резко падает.

Производительность также снижается, если возникает структурный дисбаланс. Например, если гигабитная линия связи присоединена к компьютеру с низкой производительностью, не способному достаточно быстро обрабатывать приходящие пакеты, так что некоторые пакеты будут терятся. Эти пакеты, в конце концов, будут передаваться повторно, что приведет к увеличению задержки, непроизводительному использованию пропускной способности и снижению общей производительности.

При анализе производительности сетей полезно обращать внимание на произведение пропускной способности и времени задержки. Пропускная способность канала (в битах в секунду) умножается на время прохождения сигнала в оба конца (в секундах). В результате получается емкость канала в битах. Отсюда следует, что для эффективного использования канала размер окна получателя должен быть, по меньшей мере, равен произведению пропускной способности и времени задержки, а лучше превосходить его, так как получатель может сразу и не ответить.

Когда качество работы сети неважное, ее пользователи часто жалуются сетевым операторам, требуя усовершенствований. Чтобы улучшить производительность сети, операторы должны сначала точно определить, в чем суть проблемы. Чтобы выяснить текущее состояние сети, операторы должны произвести измерения.

Основной цикл работ по совершенствованию производительности сети включает следующие этапы [31,32,35,36].

1. Измерить важные сетевые параметры и производительность сети.

2. Попытаться понять, что происходит.
3. Измерить один параметр.

Эти шаги повторяются до тех пор, пока производительность не увеличится достаточно или пока не станет ясно, что этими методами производительность уже более не увеличить.

Измерения могут быть произведены многими способами и во многих местах (как физичесски, так и в стеке протоколов). Наиболее обычный тип измерений представляет собой включение таймера при начале какой- либо активности и измерение, таким образом, продолжительности этой активности. Например, одним из ключевых измерений является измерение времени, необходимого для получения отправителем подтверждения в ответ на отправку TPDU- модуля (Transport Protocol Data Unit- модуль данных транспортного протокола). Другие измерения призводятся при помощи счетчиков, в которых учитывается частота некоторых событий (например, количество потерянных TPDU- модулей). Наконец, часто измеряются такие количественные показатели, как число байтов, обработанных за определенный временной интервал [21].

Измерения и настройки часто позволяют значительно улучшить производительность сети, однако они никогда не заменят хорошо разработанного проекта. Плохо спроектированная сеть может быть усовершенствованна только до определенного уровня. Для дальнейшего увеличения производительности ее потребуется переделать с нуля.

Каждая служба характеризуется качеством обслуживания. Некоторые службы являются надежными, в том смысле, что никогда не теряют данные. Обычно надежная служба реализуется при помощи подтверждений, посылаемых получателем в ответ на каждое принятое сообщение, так что отправитель знает, дошло ли очередное сообщение или нет. Процесс пересылки подтверждений требует некоторых накладных расходов и снижает пропускную способность канала. Обычно подобные затраты не очень велики

и окупаются, хотя иногда могут быть нежелательными. В некоторых приложениях требуется, чтобы все доставленные сообщения не содержали ошибок. Например, в банковских приложениях, при передаче программ для электронных вычислительных машин или файлов ошибка в единственном бите сообщения может иметь серьезные последствия. В других приложениях, таких как электронная почта, все сообщения должны быть доставлены, но случайный ошибочный бит в сообщении обычно может быть исправлен читателем визуально. Наконец, имеются приложения, в которых как случайная ошибка в одном бите, так и случайная потеря целых пакетов или сообщений являются допустимыми. Например, в распределенных измерительных системах передаваемые сообщения уже содержат шум, а случайная потеря сообщений вскоре восполняется более новыми сообщениями. При передаче пакетированной речи случайная потеря (или поздняя доставка) пакета или случайное искажение бита просто увеличивает зашумленность принятого речевого сигнала. Следует отметить, однако, что применение методов сжатия данных при передаче пакетированной речи, а также в других приложениях сильно повышает необходимость безошибочной передачи.

Задачи этой группы связаны с оценкой на основе накопленной статистической информации таких параметров, как время реакции системы, пропускная способность реального или виртуального канала связи между двумя конечными абонентами сети, интенсивность трафика в отдельных сегментах и каналах сети, вероятность искажения данных при их передаче через сеть, а также коэффициент готовности сети или ее определенной транспортной службы. Функции анализа производительности и надежности сети нужны как для оперативного управления сетью, так и для планирования развития сети.

Результаты анализа производительности и надежности позволяют контролировать соглашение об уровне обслуживания (Service Level

Agreement, SLA), заключаемое между пользователем сети и ее администраторами (или компанией, продающей услуги). Обычно в SLA оговариваются такие параметры надежности, как коэффициент готовности службы в течение года и месяца, максимальное время устранения отказа, а также параметры производительности, например, средняя и максимальная пропускная способности при соединении двух точек подключения пользовательского оборудования, время реакции сети (если информационная служба, для которой определяется время реакции, поддерживается внутри сети), максимальная задержка пакетов при передаче через сеть (если сеть используется только как транзитный транспорт). Без средств анализа производительности и надежности поставщик услуг публичной сети или отдел информационных технологий предприятия не сможет ни проконтролировать, ни тем более обеспечить нужный уровень обслуживания для конечных пользователей сети.

1.5. Управление безопасностью

В первые десятилетия своего существования компьютерные сети использовались, в первую очередь, университетскими исследователями для обмена электронной почтой и сотрудниками корпораций для совместного использования принтеров. В таких условиях вопросы безопасности не привлекали большого внимания. Однако теперь, когда миллионы обычных граждан пользуются сетями для управления своими банковскими счетами, заполнения налоговых деклараций, приобретают товары в Интернет-магазинах, проблема сетевой безопасности становится все более актуальной.

Безопасность включает в себя большое разнообразие вопросов, связанных с человеческими грехами. В простейшем виде службы безопасности гарантируют, что любопытные личности не смогут читать или, что еще хуже, изменять сообщения, предназначающиеся другим получателям. Службы

безопасности пересекают попытки получения доступа к удаленным службам неавторизованными пользователями. Кроме того, система безопасности предоставляет способ определить, послано ли сообщение действительно тем отправителем, чьим именем оно подписано, или же это фальсификация. Системы безопасности также решают вопросы, связанные с перехватом и повторным воспроизведением сообщений, и с людьми, отрицающими, что они посылали данные сообщения [33].

Большинство проблем безопасности вызываются злоумышленниками, пытающимися извлечь какую- либо пользу для себя или причинить вред другим.

Проблемы безопасности сетей могут быть грубо разделены на четыре пересекающиеся области: секретность, аутентификация, обеспечение строгого выполнения обязательств и обеспечение целостности. Секретность означает предотвращение попадания информации в руки неавторизованных пользователей [40]. Именно это обычно многим приходит в голову при упоминании безопасности сетей. Аутентификация позволяет определить, с кем вы разговариваете, прежде чем предоставить собеседнику доступ к секретной информации или вступить с ним в деловые отношения. Проблема обеспечения строгого выполнения обязательств имеет дело с подписями. Как доказать, что ваш клиент действительно прислал электронной почтой заказ по договорочной цене, если впоследствии он утверждает, что цена была совсем другой? Наконец, как можно быть уверенным, что принятое вами сообщение не подделано и не модифицированно по пути злоумышленником?

Задачи этой группы включают в себя, как было уже сказано выше, контроль доступа к ресурсам сети (данным и оборудованию) и сохранение целостности данных при их хранении и передаче через сеть. Базовыми элементами управления безопасностью являются процедуры аутентификации пользователей, назначение и проверка прав доступа к ресурсам сети, распределение и поддержка ключей шифрования, управления полномочиями

и т. п. Часто функции этой группы не включаются в системы управления сетями, а реализуются либо в виде специальных продуктов (например, системы аутентификации и авторизации Kerberos, различных защитных экранов, систем шифрования данных), либо входят в состав операционных систем и системных приложений.

1.6. Учет работы сети

Задачи этой группы занимаются регистрацией времени использования различных ресурсов сети - устройств, каналов и транспортных служб. Эти задачи имеют дело с такими понятиями, как время использования службы и плата за ресурсы – billing [16]. Ввиду специфического характера оплаты услуг у различных поставщиков и различными формами соглашения об уровне услуг, эта группа функций обычно не включается в коммерческие системы и платформы управления, а реализуется в заказных системах, разрабатываемых для конкретного заказчика.

Модель управления OSI не делает различий между управляемыми объектами- каналами, сегментами локальных сетей, мостами, коммутаторами и маршрутизаторами, модемами и мультиплексорами, аппаратным и программным обеспечением компьютеров, СУБД. Все эти объекты управления входят в общее понятие «система», и управляемая система взаимодействует с управляющей системой по открытым протоколам OSI.

Ниже приведена таблица функциональных групп систем управления сетями.

Задачи управления	Что выполняют	За что отвечают
Управление конфигурацией сети и именованием	Определяются сетевые адреса, имена, географичекое положение	Отвечают за поддержку маршрутов и виртуальных путей
Обработка ошибок	Обнаруживают, регистрируют и фильтруют ошибки	Выявляют, определяют и устранняют последствия сбоев и отказов в сети
Анализ производительности и надежности	Анализируют накопленную статистичес–кую информацию о состоянии сети	Обеспечивают производительность и надежность сети
Управление безопасностью	Контролируют доступ к ресурсам сети	Сохранение целостности данных при их хранении и передаче через сеть.
Учет работы сети	Регистрируют время использования различных ресурсов сети	Плата за ресурсы

Таблица 1. Функциональные группы систем управления сетями

1.7 Постановка задачи

Исходя из всего вышесказанного становится ясным, что для повышения производительности компьютерной сети и обеспечения качества

обслуживания на определенном уровне необходимо осуществление задач управления сети.

На сегодняшний день предъявляются высокие требования к качеству обслуживания. Функции качества обслуживания в сетях заключается в обеспечении гарантированного и дифференцированного обслуживания сетевого трафика. QoS представляет собой набор требований, предъявляемых к ресурсам сети при транспортировке потока данных, и обеспечивает сквозную гарантию передачи данных и основанный на системе правил контроль за средствами повышения производительности сети, такими, как механизм распределения ресурсов, коммутация, маршрутизация, механизмы обслуживания очередей и механизмы отбрасывания пакетов.

Высокая производительность - это одно из основных свойств распределенных систем, к которым относятся компьютерные сети. Это свойство обеспечивается возможностью распараллеливания работ между несколькими компьютерами сети. К сожалению, эту возможность не всегда удается реализовать. Существует несколько основных характеристик производительности сети: время реакции, пропускная способность и задержка передачи.

Пропускная способность отражает объем данных, переданных сетью или ее частью в единицу времени. Пропускная способность уже не является пользовательской характеристикой, так как она говорит о скорости выполнения внутренних операций сети - передачи пакетов данных между узлами сети через различные коммуникационные устройства. Зато она непосредственно характеризует качество выполнения основной функции сети - транспортировки сообщений - и поэтому чаще используется при анализе производительности сети, чем время реакции.

Задержка передачи определяется как задержка между моментом поступления пакета на вход какого-либо сетевого устройства или части сети и моментом появления его на выходе этого устройства. Этот параметр

производительности по смыслу близок ко времени реакции сети, но отличается тем, что всегда характеризует только сетевые этапы обработки данных, без задержек обработки компьютерами сети.

Можно отметить, что для повышения производительности компьютерной сети одним из значительных аспектов управления является управление конфигурированием и поименованием сети, а именно здесь можно выделить вопросы маршрутизации и управления потоками. То же самое можно сказать с точки зрения качества обслуживания. Маршрутизация на основе QoS - это механизм маршрутизации, в соответствии с которым пути для потоков трафика определяются на основе некоторых сведений о наличии ресурсов сети и на основе требований потоков к качеству обслуживания, а это в свою очередь дает возможность добиться наилучших показателей маршрутизации и, соответственно, управления потоками.

Как известно, существует множество различных маршрутизаций, которые преследуют различные цели, например, достижение минимального пути, наименьшей задержки, обеспечение наилучшей пропускной способности и т.д. Но в конктретном случае очень трудно бывает определить, какая маршрутизация даст наилучшие результаты, поэтому возникает задача определения оптимальной маршрутизации, а затем управления потоками в сочетании этой оптимальной маршрутизацией.

Поэтому в данной диссертационой работе задача повышения производительности сети поставлена следующим образом:

1. определить оптимальную маршрутизацию с учетом потоков сети, пропускной способности и задержки,
2. решить задачу управления потоками в сочетании с оптимальной маршрутизацией.

Глава 2. Маршрутизация

2.1 Основные понятия

Под маршрутизацией понимают выбор непрерывного пути (включающего несколько промежуточных линий сети) между любыми двумя узлами сети (источником и приемником). Обычно, для выбора маршрута используется довольно сложный набор алгоритмов, которые работают более или менее независимо, хотя и обмениваются информацией. Его сложность обусловлена рядом причин: маршрутизация требует координации работы всех узлов сети; система маршрутизации должна справиться с выходом из строя линий или узлов путем перенаправления трафика и обновления баз данных, используемых системой; для достижения наилучших характеристик алгоритм маршрутизации может изменить маршруты, когда некоторые области сети становятся перегруженными.

Задача маршрутизации имеет два аспекта: первый касается выбора маршрутов для достижения наилучших характеристик, а второй состоит в распространении между всеми узлами сети информации, необходимой для выбора маршрута.

Алгоритм маршрутизации выполняет две главные функции: выбор маршрутов для различных пар отправитель - адресат и обеспечение правильной доставки сообщений их адресату после того, как выбраны маршруты. Вторая функция называется коммутацией и обеспечивается путем использования различных протоколов и структур данных (называемых маршрутными таблицами).

Вообще можно сказать, что маршрут, прокладываемый в сети, реализуется в виде путевой процедуры на коммутационной среде. Системы коммутации сетей подразделяются на два класса: система коммутации каналов и система коммутации сообщений или пакетов. Основное отличие

этих классов состоит в следующем: при коммутации каналов между источником информации и адресатом с помощью коммутаторов устанавливается непосредственная линия связи, по которой и ведется передача. При коммутации сообщений источник и адресат не соединяются прямой линией связи, и передача ведется через ряд промежуточных пунктов. Для передачи информации в магистральной сети в настоящее время повсеместно используют метод коммутации пакетов, в соответствии с которым передаваемые сообщения (массивы данных, управляющие сигналы и др.) разбиваются на отдельные коммутируемые в промежуточных узлах блоки, называемые пакетами. Существуют две основные концепции построения процедур передачи пакетов в сети.

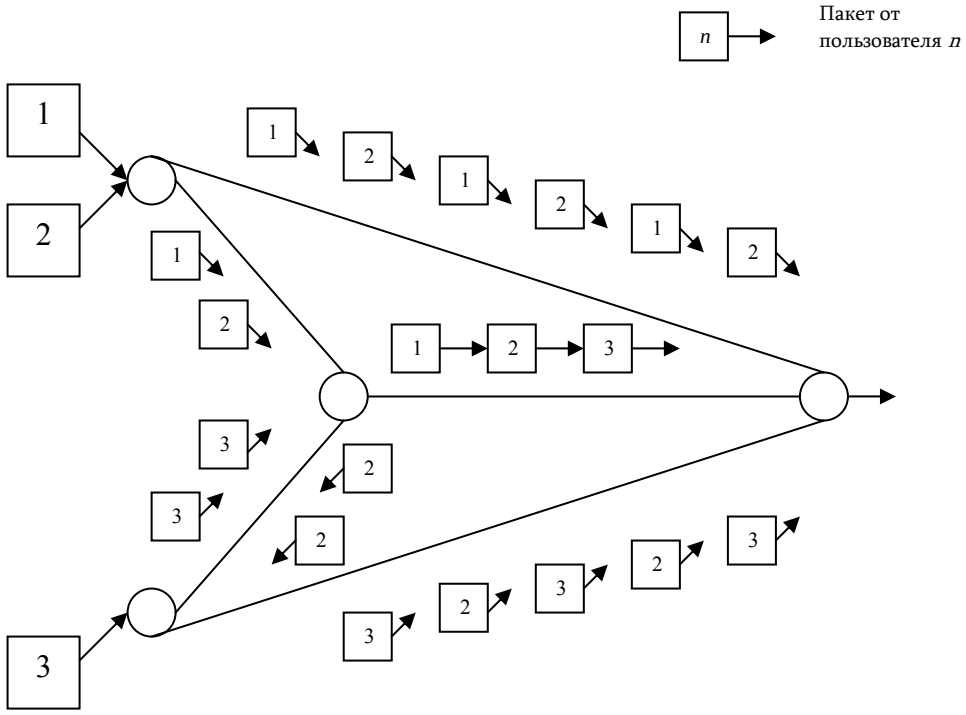


Рис. 2.1.1. Маршрутизация в дейтаграммной сети.

Первая концепция основана на полной независимости каждого из пакетов (даже если они части одного сообщения) и доставке их к получателю, в общем случае, различными маршрутами, определяемыми сложившейся

динамической ситуацией в сети. Для обеспечения независимой передачи по сети в каждый пакет включается полный заголовок с адресом получателя и необходимыми служебными маршрутами (сетевыми признаками). Такие пакеты называются дейтаграммами, а совокупность процедур управления их передачей по сети - дейтаграммной службой (рис. 2.1.1).

Вторая концепция предполагает предварительное установление маршрута передачи всего сообщения от отправителя до получателя с помощью специального служебного пакета - запроса на соединение. Маршрут в случае согласия получателя закрепляется для всего последующего трафика и получает номер соответствующего виртуального канала (соединения) для использования его другими пакетами того же сообщения, передаваемыми по тому же адресу. Совокупность процедур, обслуживающих виртуальные соединения, образует службу виртуальных соединений (рис. 2.1.2.).

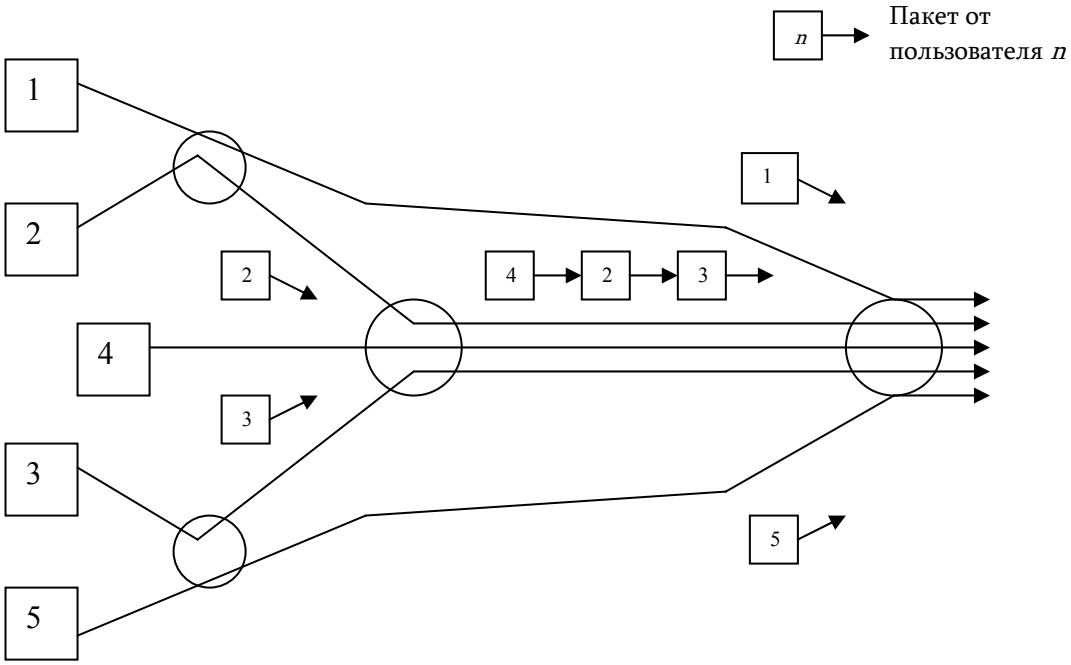


Рис.2.1.2. Маршрутизация в сети с виртуальными цепями.

Пакеты, передаваемые по одному виртуальному соединению, не являются независимыми и, поэтому, включают сокращенный заголовок, содержащий порядковый номер пакета в составе одного сообщения (диалога).

Преимущество режима виртуальных соединений перед дейтаграммной службой заключается в обеспечении упорядоченности пакетов, поступающих по адресу получателя, и сравнительной простоте управления потоком данных вдоль маршрута в целях ограничения нагрузки в сети и возможности предварительного резервирования ресурсов памяти на узлах коммутации, входящих в маршрут. К недостаткам режима виртуальных соединений следует отнести существенно большую сложность реализации, значительные накладные расходы для установления и разъединения соединений и отсутствие воздействия изменившейся ситуации в сети на маршрут виртуальных соединений. В настоящее время нет устоявшегося единого мнения относительно достоинств и недостатков режимов дейтаграммной службы и виртуальных соединений, и поэтому существующие рекомендации, как правило, ориентированны на применение обоих режимов, предоставляя пользователю право выбора одного из них или использования обоих сразу. В случаях, когда сеть ориентированна на передачу преимущественно больших массивов данных, предпочтительнее использовать режим виртуальных соединений, поскольку он обеспечивает упорядоченность поступления пакетов адресату. При обмене преимущественно короткими сообщениями выгоднее использовать дейтаграммную службу. Это позволяет более эффективно использовать пропускную способность сети.

В случае дейтаграммной передачи маршрутизация осуществляется на каждом шаге передачи, а в случае сети с виртуальным соединением алгоритм маршрутизации используется для выбора пути по сети для данного виртуального соединения.

2.2 Классификация маршрутизаций

Разработано большое число алгоритмов маршрутизации. Все эти алгоритмы могут быть классифицированны по типам. В качестве основы для классификации выберем следующие характеристики: метод управления потоком передаваемой информации, способ принятия решения при выборе маршрутов, способ обновления маршрутной информации и способ передачи пакетов.

С учетом вышеперечисленных характеристик можно привести следующие основные типы маршрутизаций (рис.2.2.1):

Рис. 2.2.1. Классификация маршрутизаций

- статические и динамические;
- централизованные, децентрализованные и смешанные;
- стохастические и детерминированные;
- одноуровневые и иерархические;
- направленные и ненаправленные;
- однопутевые и многопутевые;

- внутри и междоменные;
- маршрутизации хостом и маршрутизатором;
- канальные, векторные и принудительные;
- статические и динамические.

При статической маршрутизации таблица маршрутизации создается при настройке маршрутизатора и не изменяется динамически. Время от времени администратор сети может корректировать ее вручную. Статическая маршрутизация не в состоянии оперативно следовать всем изменениям сети, поэтому ее редко используют. Простейшим примером такого подхода является алгоритм выбора кратчайшего пути.

Алгоритмы динамической маршрутизации способны адаптироваться к изменениям конфигурации сети, поэтому они очень широко применяются. Алгоритмы динамической маршрутизации основываются на изменениях времен задержек длин очереди, интенсивности потоков с учетом информации не о всей сети, а только о близлежащих узлах.

При централизованной маршрутизации все таблицы маршрутизации составляются в одном месте, обычно именуемом центром управления маршрутизацией, и распределяются среди маршрутизаторов. Эта схема выгодна тем, что все маршрутизаторы имеют общее представление о сети, поскольку их таблицы маршрутизации одинаковы, кроме того вычислительные ресурсы сосредоточены в одном месте. Слабое место этой схемы - центр управления маршрутизацией, отказ которого может остановить всю сеть. Кроме того, звездообразная топология (с центром управления маршрутизацией в середине) требует высокоскоростных линий. Выгода от синхронизации таблиц всех маршрутизаторов зачастую несколько уменьшается от того, что в больших сетях маршрутизаторы, расположенные ближе к центру управления, получают новые таблицы гораздо раньше, чем сильно удаленные представления о сети.

В алгоритмах децентрализованной маршрутизации вычисления прводятся на нескольких узлах. Таблицы маршрутизации обновляются из нескольких мест и определяются непосредственно в процессе выполнения путевой процедуры. Примером децентрализованной маршрутизации являются алгоритмы лавинной маршрутизации, фиксированной маршрутизации, разветвления трафика (бифуркации) и маршрутизации «идеального наблюдателя».

При стохастической маршрутизации для принятия решения о маршруте используют оценки времени, необходимого для достижения узлов назначения, полученные из наблюдений за прхождением сообщений через узлы. Простейшим примером стохастической маршрутизации является выбор фиксированных правил передачи сообщений соседним узлам.

В алгоритмах детерминированной маршрутизации расчитываются времена, необходимые для достижения узлов назначения в идеальных, т.е. неменяющихся условиях по структуре сети. Примером детерминированного алгоритма является алгоритм выбора маршрута с наименьшей задержкой при передаче сообщения.

При направленной маршрутизации алгоритм рассматривает направление передачи.

При ненаправленной маршрутизации алгоритм рассматривает ненаправленную передачу данных. Примером ненаправленной маршрутизации является алгоритм по предыдущему опыту и метод скорейшей передачи.

При одноуровневой маршрутизации все маршрутизаторы равноправны, а при иерархической маршрутизации часть маршрутизаторов функционируют на равных, но доступ к другим узлам осуществляется только через один маршрутизатор более высокого уровня. Фактически, маршрутизаторы верхних уровней иерархии формируют межсетевые магистрали. Таким образом, внутридоменная маршрутизация является одноуровневой, междоменная -

иерархической, поскольку пакеты, направленные в другие домены, проходят через один магистральный маршрутизатор. Пакет, отправленный из поддомена М1 или М2, и покидающий домен через М3, маршрутизируется иерархически.

При однопутевой маршрутизации каждому адресату соответствует только один путь. За счет этого сокращается размер таблицы маршрутизации.

В алгоритмах многопутевой маршрутизации для каждого адресата вычисляется несколько путей. Это позволяет оптимально использовать емкость канала связи и повысить общую пропускную способность. Дополнительно обеспечивается некоторая отказоустойчивость сети. Недостаток многопутевой схемы в том,что таблицы маршрутизации занимают большой объем, а сами алгоритмы становятся сложнее.

При маршрутизации хостом в каждом пакете кроме адреса получателя указан полный список узлов, составляющих маршрут пакета. Они перечислены в порядке прохождения пакета. При этом маршрутизатор служит просто приемно-передающим устройством. Эта схема также известна как маршрутизация источника. Сначала хост генерирует специальный пакет и отправляет его по всем известным ему маршрутам. Каждый маршрутизатор, приняв такой пакет, добавляет в него запись о себе и рассылает по всем маршрутам, о которых знает сам. В такой схеме адресат иногда получает несколько пакетов, при этом все они отсылаются обратно отправителю, который изучает пути их следования и выбирает оптимальный. Это - значимое преимущество, а недостаток в том, что механизм определения путей наводняет сеть пакетами данных. Такая маршрутизация используется нечасто.

При маршрутизации маршрутизатором хост просто отправляет пакет данных на адрес получателя. Маршрутизаторы сами проводят вычисления и определяют дальнейший путь пакета. В этой схеме все решения принимают

маршрутизаторы. Эти маршрутизации могут быть канальная, векторная и принудительная.

При канальной маршрутизации маршрутизатор широковещательно рассылает список узлов, отражающий состояние канала, к которым подключен. При такой схеме маршрутизаторы отправляют маленькие объемы информации множеству узлов об изменении состояния канала. Маршрутизатор, использующий векторный алгоритм, рассылает полную таблицу маршрутизации, но только своим соседям. Суть векторной маршрутизации в минимизации числа узлов, проходимых пакетом. Протоколы принудительной маршрутизации не что иное, как модификация векторных. Эти алгоритмы при выборе маршрута помимо физических факторов (число переходов и стоимость) учитывают и политические (деловые соглашения с другими компаниями).

Как видно из всего вышесказанного, все охарактеризованные типы маршрутизации ориентированы на адресата и используют те или иные механизмы и параметры для достижения наилучшего показателя параметра при доставке информации адресату.

2.3. Параметры маршрутизации

В алгоритмах маршрутизации при формировании маршрута можно использовать различные показатели: длина маршрута, надежность, задержка, пропускная способность, нагрузка, стоимость связи. Их же можно охарактеризовать как основные параметры, с помощью которых можно охарактеризовать алгоритм любой маршрутизации [38].

Длина маршрута является наиболее общим показателем маршрутизации. Некоторые протоколы маршрутизации позволяют администраторам сети назначать произвольные цены на каждый канал сети. В этом случае длиной тракта является сумма расходов, связанных с каждым

каналом, который учавствует в передаче. Другие протоколы маршрутизации определяют «количество пересылок», т.е. показатель, характеризующий число проходов, которые пакет должен совершить на пути от источника до пункта назначения через устройства объединения сетей.

Под надежностью понимают надежности каждого канала сети. Некоторые каналы сети могут отказывать чаще, чем другие. Отказы одних каналов сети могут быть устранены легче или быстрее, чем отказы других каналов. При назначении оценок надежности могут быть приняты в расчет любые факторы надежности. Обычно, оценки надежности назначаются каналам сети администраторами сети. Надежность канала оценивается величиной - *бит/ошибка.*

Под задержкой маршрутизации понимают отрезок времени, необходимый для передвижения пакета от источника до пункта назначения через объединенную сеть. Задержка зависит от многих факторов, включая полосу пропускания промежуточных каналов сети, очереди в порте каждого маршрутизатора на пути передвижения пакета, перегруженность сети на всех промежуточных каналах сети и физическое расстояние, на которое необходимо переместить пакет, способ предотвращения или разделение конфликтов при обмене сообщений. Так как здесь имеет место конгломерация нескольких важных переменных, задержка является наиболее общим и полезным показателем.

Под пропускной способностью можно понимать количество обслуживаний. Но для ее оценки можно также рассмотреть полосу пропускания. Полоса пропускания относится к имеющейся мощности трафика какого-либо канала. Хотя полоса пропускания является оценкой максимально достижимой пропускной способности канала, маршруты, проходящие через каналы с большей полосой пропускания, не обязательно будут лучше маршрутов, проходящих через менее быстродействующие каналы.

Нагрузка представляет собой потоки информации между парами объектов сети. Под потоком понимают сообщения, запросы, сигналы, передаваемые в сети. Если анализируется пропускная способность сети, то рассчитываются максимальные потоки, которые могут возникнуть между каждой парой объектов.

Стоимость связи (передачи) является величиной, зависящей от пропускной способности и потока.

В качестве критерия выбора маршрутов можно использовать такие показатели, как длина маршрута, отсутсвие тупиков и стоимость.

Можно выделить два основных показателя, на которые существенное влияние оказывает алгоритм маршрутизации - пропускная способность (количество обслуживаний) и средняя задержка пакета (качество обслуживания).

2.4. Управление потоками

Надо отметить. что маршрутизация взаимодействует с управлением потоками в компьютерной сети.

В любой системе передачи информации основной задачей является обеспечение свободного движения информационных потоков. На первый взгляд может показаться, что для этого вполне достаточно, чтобы пропускные способности узлов и каналов всегда превосходили предлагаемую нагрузку, однако это не так. В большинстве сетей возникают ситуации, когда поступающая нагрузка больше той, которая может быть обслужена даже при оптимальной маршрутизации. Тогда, если не предпринять никаких мер по ограничению поступающего трафика, размеры очередей на наиболее нагруженных линиях будут неограниченно расти и в конце концов превысят размеры буферов в соответствующих узлах. Когда это происходит, пакеты поступающие в узлы, для которых нет свободного места в буфере, будут

сброшены и позднее переданы повторно, что приведет к пустой трате ресурса связи. В результате возникает эффект, аналогичный транспортной пробке на автостраде, когда при увеличении поступающей нагрузки действительная пропускная способность сети уменьшается, а задержка пакета становится чрезвычайно большой. Таким образом, иногда необходимо не допускать в сеть какую-то часть поступающего трафика для того, чтобы избежать подобных перегрузок. Обеспечение мер, регулирующих объем трафика в сети, и составляет задачу управления потоком.

В основном вопрос управления потоком возникает, когда ограничение на скорость передачи между двумя точками вследствии ограниченной пропускной способности линии передачи или узла обработки [56,49].

Нужно отметить и выделить несколько целей управления потоком. Во-первых, это - установление подходящего компромисса между притеснением пользователей и удержанием средней задержки сообщения на разумном уровне. Во-вторых, это - соблюдение справедливости по отношению ко всем пользователям, когда часть поступающего трафика не допускается в сеть. В-третьих, это - не допустить уменьшения пропускной способности и тупиковой ситуации из-за переполнения буферов.

Рассмотрим выше приведенные цели управления потоком [48,50].

Малая средняя задержка пакета, конечно, желательна с точки зрения пользователя. Однако важно понять, что управление потоком на сетевом уровне не обязательно уменьшает задержку для пользователей сети; оно просто перебрасывает задержку с сетевого уровня на более высокие уровни. Это означает, что путем ограничения входа в подсеть управление потоком заставляет пакеты ждать вне подсети, а не в очередях внутри ее. Таким образом, управление потоком, возможно, не окажет большую помощь пользователю, чьи пакеты вынуждены проходить очень медленно и, как оказывается в действительности, оно иногда увеличивает задержку. Польза от управления потоком в основном состоит в том, что оно может предотвратить

появление катастрофических трафиковых перегрузок в подсети, способных ухудшить положение многих пользователей. Единственный способ уменьшить неудовлетворительность пользователя по поводу больших задержек состоит в гарантировании того, чтобы потребность в управлении потоком не возникала слишком часто; этого можно достичь либо путем увеличения сетевых ресурсов связи (пропускных способностей линий и т.д.), либо путем улучшения алгоритма маршрутизации. Другой возможностью, конечно, является запрещение доступа новых пользователей к сети, если она находится в почти перегруженном состоянии, но это порой бывает трудно реализовать и вызывает неудовлетворенность другого характера.

Основной причиной, почему важно сохранять задержку малой внутри подсети, а не вне ее, является то, что при этом не тратятся ресурсы подсети на повторную передачу пакетов. Последнее происходит в двух случаях: во-первых, когда размеры очередей становятся такими, что происходит переполнение буферов в узлах и приходящие пакеты сбрасываются; во-вторых, когда подтверждения возвращаются настолько поздно, что узел-источник ошибочно думает, что некоторые пакеты потеряны, и начинает повторно передавать их. При повторных предачах тратятся впустую ресурсы сети; эти передачи существенно уменьшают пропускную способность сети и приводят к распространению перегрузок.

Заметим, что сохранить сетевую задержку малой, когда предлагаемая нагрузка велика, можно только путем уменьшения числа сообщений, передаваемых по сети. Поэтому должен существовать естественный компромисс между разрешением свободного доступа пользователя к сети и сохранением задержки на достаточно низком уровне для того, чтобы повторные передачи или другие нежелательные явления не ухудшали сетевые характеристики. Несколько упрощенная рекомендация состоит в том, чтобы управление потоком вообще не использовалось, когда сетевая задержка находится ниже некоторого критического уровня, а в условиях большой

нагрузки оно должно отвергать столько предлагаемой нагрузки, сколько необходимо для того, чтобы сохранить задержку на критическом уровне.

Когда часть поступающего трафика должна быть отвергнута, важно сделать это справедливо. Это не тривиальная задача, так как максимизация суммарной пропускной способности сети часто оказывается не совместимой со справедливостью. Как правило, необходим компромисс между справедливостью и притеснением тех пользователей, которые больше других ответственны за перегрузки. Нужно заметить также, что пользователи могут принадлежать разным приоритетным классам и в этом случае справедливость подразумевается только внутри приоритетного класса.

Термин «перегрузка» часто используется для обозначения явления, в котором увеличение поступающей нагрузки приводит к уменьшению пропускной способности и увеличению задержки. Ранее уже было упомянуто, что перегрузки могут возникать в результате переполнения буфера. Проблема, связанная с переполнением буфера, состоит в том, что могут возникнуть тупиковые ситуации, когда два или более узла не смогут дальше продвигать пакеты из-за того, что у всех потенциальных приемников нет свободного места в буфере. Существуют простые способы устранения подобных тупиковых ситуаций путем распределения пакетов по приоритетным уровням и выделения дополнительного места в буфере для пакетов с более высоким приоритетом [15].

Перегрузки и переполнения буферов не являются столь же важными вопросами в управлении потоком, как задержка. При существующей технике стоимость буферов невысока и сети следует проектировать таким образом, чтобы переполнения буферов возникали редко. Другими словами, буферы должны быть настолько большими, чтобы до того, как они начнут переполняться, задержка становилась весьма существенной.

Рассмотрим взаимосвязь между управлением потока и маршрутизацией.

Когда поступающая нагрузка - трафик, поступающий в сеть от внешних источников относительно мал, он полностью будет принят сетью и тогда

Пропускная способность= Поступающая нагрузка.

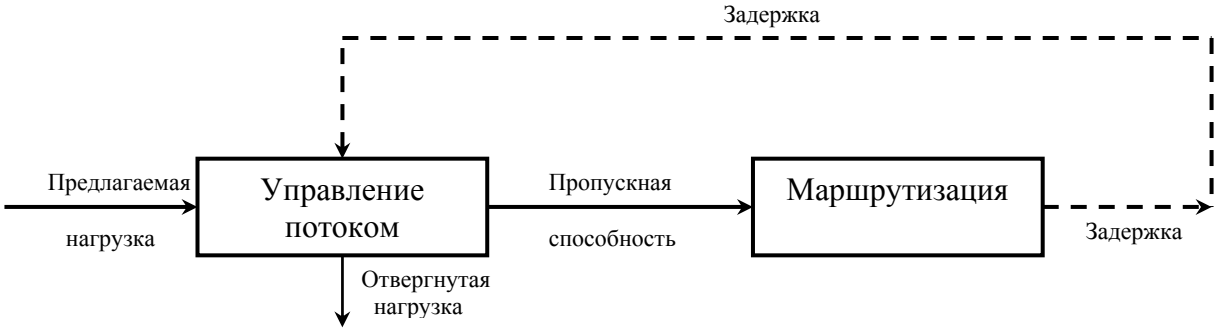


Рис. 2.3.1. Взаимодействие между управлением потоков и маршрутизацией

Когда поступающая нагрузка чрезмерно большая, часть этой нагрузки будет отвергаться алгоритмом управления потоками и тогда

Пропускная способность= Поступающая нагрузка- Отвергнутая нагрузка.

Трафик, принятый в сеть, будет иметь среднюю задержку пакетов, зависящую от того, какие маршруты были выбраны алгоритмом маршрутизации. Однако на пропускную способность существенное влияние оказывает также алгоритм маршрутизации, т.к. алгоритм управления потоками обычно действует на основе поддержания баланса между пропускной способностью и средней задержкой, например: поступающая нагрузка не принимается, как только задержка становится слишком большой. Поэтому, если алгоритму маршрутизации удается более успешно

поддерживать малую задержку, то алгоритм управления потоками разрешает принимать в сеть больше трафика. Хотя точный баланс между задержкой и пропускной способностью устанавливается алгоритмом управления потоками, хорошая маршрутизация в условиях большого предлагаемого трафика дает более предпочтительную зависимость задержка - пропускная способность, по которой действует алгоритм управления потоками.

Для того, чтобы добиться наилучших показателей маршрутизации и, соответственно, управления потоками, желательно использовать алгоритм маршрутизации, ориентированный не на адресат, а на сведения о наличии ресурсов сети и на основе требований потоков к качеству обслуживания. Маршрутизацию такого типа называют маршрутизацией на основе QoS (QoS-based routing). При маршрутизации на основе QoS -политик: протокол маршрутизации передает метрики с динамической информацией о наличии ресурсов (QoS); протокол маршрутизации должен вычислять не только оптимальный путь, но и другие возможные пути на основе их QoS-доступности; в каждом потоке передаются сведения о необходиомо качестве услуг (QoS).

Маршрутизация на основе QoS влечет существенные сложности, поскольку метрики QoS-доступности очень динамичны, и обновления маршрутов происходят гораздо чаще. Но с другой стороны, здесь возможно устанавливать значение приоритета, маршрутизация моржет быть основана на длине пакета, адресе источника, потоке, битами ToS (тип сервиса), битами приоритетов и т.д. В больших динамически маршрутизируемых окружениях все больше растет необходимость в применении функции качества обслуживания и управления трафиком, поскольку подобный подход является более гибким и значительно улучшает показатели маршрутизации и управления потоками.

Глава 3. Качество обслуживания (QoS)

3.1. Общие понятия

Важность внедрения механизма QoS (Quality of Service- качество обслуживания) в компьютерных сетях возросла благодаря увелечению популярности сетей и преобретению ими коммерческих черт. Технологии качества обслуживания позволяют максимально оптимизировать производительность сетей и обеспечить стабильное функционирование нового поколения мультимедийных и голосовых приложений.

Функции качества обслуживания в сетях заключается в обеспечении гарантированного и дифференцированного обслуживания сетевого трафика путем передачи контроля за ипользованием ресурсов и загруженностью сети ее операттору. QoS представляет собой набор требований, предъявляемых к ресурсам сети при транспортировке потока данных и обеспечивает сквозную гарантию передачи данных и основанный на системе правил контроль за средствами повышения производиткльности сети, такими, как механизм распределения ресурсов, коммутация, маршрутизация, механизмы обслуживания очередей и механизмы отбрасывания пакетов.

Способность сети обеспечивать различные уровни обслуживания, запрашиваемые теми или иными сетевыми приложениями, наряду с проведением контроля за характеристиками производительности - полосой пропускания, задержкой, дрожанием и потерей пакетов - может быть классифицированна по трем перечисленным ниже категориям [1].

- Негарантированная доставка данных. Обеспечение связности узлов сети без гарантии времени и самого факта доставки пакета в точку назначения. Следует отметить, что отбрасывание пакета может произойти только в случае переполнения буфера входной или выходной очереди маршрутизатора.

На самом деле негарантированная доставка пакетов не является частью QoS вследствии отсутствия гарантии качества обслуживания и гарантии обеспечения доставки пакетов. Следует отметить, что негарантированная доставка пакетов является на сегодняшний день единственной услугой, поддерживаемой в Internet.

Несмотря на некоторое снижение производительности, для большинства приложений, ориентированных на передачу информации (например, приложений, обеспечивающих взаимодействие по протоколу передачи файлов (File Transfer Protocol- FTP)), эта услуга является вполне достаточной. В целом же оптимальные условия функционирования всех приложений включают в себя требования к выделению определенных сетевых ресурсов в терминах полосы пропускания, задержки и уровня потери пакетов.

- Дифференцированное обслуживание. Дифференцированное обслуживание предполагает разделение трафика на классы на основе требований к качеству обслуживания. Каждый класс трафика дифференцируется и обрабатывается сетью в соответствии с заданными для этого класса механизмами QoS.

  Следует отметить, что дифференцированное обслуживание само по себе не предполагает обеспечения гарантий предоставляемых услуг. В соответствии с данной схемой трафик распределяется по классам, каждый из которых имеет свой собственный приоритет. По этой причине дифференцированное обслуживание довольно часто называют мягким QoS.

  Дифференцированное обслуживание удобно применять в сетях с интенсивным трафиком приложений. В этом случае важно обеспечить отделение административного трафика сети от всего остального трафика и назначить ему приоритет, позволяющий в любой момент времени быть уверенным в связности узлов сети.

- Гарантированное обслуживание. Гарантированное обслуживание предполагает резервирование сетевых ресурсов с целью удовлетворения специфических требований к обслуживанию со стороны потоков трафика.

  В соответствии с гарантированым обслуживанием выполняется предварительное резервирование сетевых ресурсов по всей траектории движения трафика. Гарантированное обслуживание довольно часто называют еще жестким QoS в связи с предъявлением строгих требований к ресурсам сети.

  К сожалению, резервирование ресурсов на всем пути следования отдельных потоков трафика невозможно реализовать в масштабах магистрали Internet, обслуживающей в отдельный момент времени тысячи потоков данных. Исправить положение призвано агрегированное резервирование ресурсов, требующее хранения в базовых маршрутизаторах Internet всего лишь небольшого количества информации.

  Приложения, требующие гарантированного обслуживания, включают в себя мультимедийные приложения, приводящие передачу голосовой информации и видеоизображений. Интерактивные приложения, ориентированные на передачу речи по Internet, могут функционировать нормально (т.е. не вызывая неудобства у пользователей) лишь в том случае, если значение латентности равно или меньше 100 мс. Следует отметить, что аналогичный уровень латентности является приемлемым для большинства мультимедийных приложений. А вот приложения Internet- телефонии уже понадобится канал передачи информации с пропускной способностью как минимум 8 Кбит/с и со значением задержки подтверждения приема, равном 100 мс. Для того чтобы удовлетворить подобные требования к

гарантированному обслуживанию, сеть должна обладать определенным запасом ресурсов.

3.2 Функции качества обслуживания

Функции качества обслуживания являются неотъемлемой частью современных легкомасштабируемых сетей. Ниже рассмотрены различные функции качества обслуживания и связанные с ними возможности и преимущества [1,9].

- Классификация и маркировка пакетов;
- Управление интенсивностью трафика;
- Распределение ресурсов;
- Предотвращение перегрузки и политика отбрасывания пакетов;
- Сигнальный протокол QoS;
- Коммутация;
- Маршрутизация.

Ниже рассмотрены подробнее функции качества обслуживания.

3.2.1 Классификация и маркировка пакетов

Функции формирования трафика, реализованные на границе сети, являются жизненно важными для обеспечения дифференцированного обслуживания. К этим функциям относятся функции классификации трафика, маркировки пакетов и управления интенсивностью трафика.

Классификация пакетов является необходимым условием для идентификации различных классов трафика в зависимости от требуемого уровня обслуживания. IP-пакет может быть классифицирован на основании одного или нескольких полей заголовка. После отнесения пакета к определенному классу он маркируется посредством установки

соответсвующего значения поля IP-приоритета, то есть функция маркировки пакетов используется для разметки классифицированного трафика путем установки значения поля IP- приоритета или поля дифференцированного обслуживания.

Управление интенсивностью трафика – это необходимое условие существования достаточного количества ресурсов и обеспечения функций качества обслуживания внутри базовой сети. Управление интенсивностью трафика предполагает соизмерение параметров поступающего в сеть трафика клиента с его профилем с помощью ограничивающей функции. И наоборот, компания, подключенная к сети поставщика услуг, может проверять параметры своего исходящего трафика с целью поддержания его интенсивности на уровне, удовлетворяющим всем ограничивающим функциям поставщика услуг.

Наличие формирователей трафика, расположенных на границе сети, является необходимым условием для обеспечения сетью функций дифференцированного обслуживания.

3.2.2. Классификация пакетов

Классификация пакетов представляет собой средство, позволяющее отнести пакет к тому или иному классу трафика в зависимости от значения одного или нескольких полей пакета. Распознающая функция может быть как очень простой, так и весьма сложной. Ниже перечислено несколько различных способов классификации пакетов [39].

- Распознающая функция IP- потока зависит от пяти параметров: адреса источника IP- пакета, адреса назначения IP- пакета, поля протокола IP, порта источника и порта назначения.
- Распознающая функция зависит от значения поля IP-приоритета или поля кода дифференцированной услуги (DSCP).

- Распознающая функция зависит от других параметров заголовка TCP/IP- пакета, таких, как длина пакета.
- Распознающая функция зависит от MAC- адреса источника и MAC- адреса назначения пакета.
- Распознающая функция зависит от используемых приложением номеров портов, адресов URL (Universal Resource Locator- универсальный указатель информационного ресурса) и т.д.

Чтобы задать критерий совпадения пакетов на основании различных параметров потока, можно использовать список доступа. Кроме того, списки доступа могут быть использованы и для идентификации пакетов на основе значения поля IP- приоритета или поля DSCP. Распознование приложений на основе сетевых параметров (NBAR) позволяет маршрутизаторам идентифицировать трафик отдельных приложений, позволяя таким образом проводить классификацию пакетов на основе сгенерировавших их программных средств.

Классификация пакетов может быть основана также и на внутренних параметрах маршрутизатора. Примером подобной классификации является идентификация пакетов на основании входного интерфейса маршрутизатора или идентификация пакетов на основании значения поля QoS - группы, относящегося к внутренней по отношению к маршрутизатору структуре данных пакета.

3.2.3. Маркировка пакетов

Маркировка пакетов используется для идентификации соответсвующего им класса трафика. Пакеты могут быть маркированы путем установки значения поля IP- приоритета или поля кода дифференцированной услуги (DSCP), расположенных в заголовке IP- пакета, а также путем

установки значения поля QoS- группы, относящегося к внутренней по отношению к маршрутизатору структуре данных пакета.

Классификацию пакетов часто называют также маркировкой или раскраской пакетов. Все пакеты, принадлежащие определенному классу трафика, «окрашиваются» в соответствующий цвет.

3.2.4. IP- приоритет

Поле IP- приоритета, расположенное в заголовке IP- пакета, указывает на относительный приоритет при обработке соответсвующего пакета данных [13]. Каждый кадр данных или маркер имеет приоритет, устанавливаемый битами приоритета (значение от 0 до 7, причем 7 - наивысший приоритет). Станция может воспользоваться маркером, если только у нее есть кадры для передачи с приоритетом равным или большим, чем приоритет маркера. Сетевой адаптер станции с кадрами, у которых приоритет ниже, чем приоритет маркера, не может захватить маркер, но может поместить наибольший приоритет своих ожидающих передачи кадров в резервные биты маркера, но только в том случае, если записанный в резервных битах приоритет ниже его собственного. В результате в резервных битах приоритета устанавливается наивысший приоритет станции, которая пытается получить доступ к кольцу, но не может этого сделать из-за высокого приоритета маркера.

Станция, сумевшая захватить маркер, передает свои кадры с приоритетом маркера, а затем передает маркер следующему соседу. При этом она переписывает значение резервного приоритета в поле приоритета маркера, а резервный приоритет обнуляется. Поэтому при следующем проходе маркера по кольцу его захватит станция, имеющая наивысший приоритет. При инициализации кольца основной и резервный приоритет маркера устанавливаются в 0.

3.2.5. DSCP

Поле кода дифференцированной услуги (DSCP) используется для идентификации PHB- политики обработки пакетов. Аналогично IP-приоритету, поле DSCP является частью заголовка IP- пакета. На самом деле поле DSCP

Представляет собой расширение поля IP- приоритета. Следовательно, способы использования и установки значения поля DSCP во многом напоминают рассмотренные нами ранее способы использования и установки значения поля IP- приоритета. Следует отметить, что поле кода дифференцироваггой услуги (DSCP) обладает обратной совместимостью с полем IP– приоритета.

3.2.6. QoS- группа.

QoS- группа представляет собой поле внутренней по отношению к маршрутизатору структуры данных пакета. QoS- группы применяется для маркировки пакетов на основании определенных пользователем критериев классификации. Следует отметить, что поле QoS- группы является внутренним по отношению к маршрутизатору и не входит в заголовок IP-пакета.

С помощью поля QoS- группы производится внутренняя по отношению к маршрутизатору «раскраска» пакета. Средства качества обслуживания, поддерживающие данную функцию маркировки пакетов, включают в себя механизм согласования скорости доступа (CAR) и механизм распространения политик QoS с помощью протокола пограничного поля (QPPB). Модульный интерфейс командной строки QoS позволяет проводить маркировку пакетов с использованием любого из трех рассмотренных выше механизмов.

Довольно часто пакеты поступают в пограничное устройство сети с уже установленным полем IP- приоритета или полем DSCP. Несмотря на то что поступивший пакет уже имеет «раскраску», сетевой оператор должен назначить новую маркировку пакета в зависимости от его класса и соответствующего этому классу качества услуг, предлагаемых сетью.

3.3. Управление интенсивностью трафика

С целью обеспечения функций качества обслуживания весь входящий в сеть поставщика услуг трафик должен проходить строгий контроль на границе сети на предмет соответствия его интенсивности поддерживаемым сетью параметрам [8]. Даже если небольшое число пограничных маршрутизаторов начнут передавать трафик с интенсивностью, превышающей максимально допустимое значение, увеличившаяся нагрузка потока трафика может привести к перегрузке сети. Следует отметить, что снижение производительности сети неизбежно приведет к невозможности обеспечения функций качества обслуживания для всего сетевого трафика.

Управление интенсивностью трафика может быть достигнуто за счет применения двух функций: функции ограничения трафика, предусмотренной механизмом (CAR), и функции выравнивания трафика (traffic shaping- TS), предусмотренной одноименным механизмом. Несмотря на одинаковое предназначение, названные выше функции отличаются способом обработки трафика в момент исчерпания маркеров. Понятие маркера исходит из схемы «корзины маркеров» - известного алгоритма дозированной передачи трафика, который будет рассмотрен в следующих разделах.

3.3.1. Корзина маркеров

В процессе работы механизм управления интенсивностью трафика полагается на функцию дозирования трафика. Одной из наиболее распространенных схем дозирования трафика является так называемая схема «корзина маркеров». Схема «корзина маркеров» используется как алгоритмом ограничения, так и алгоритмом выравнивания трафика и представляет собой средство сообщения о результатах сопоставления параметров пакета с заданными ограничениями интенсивности. В зависимости от результатов дозирования принимается соответствующее решение - передать пакет, отбросить и т.п.

Схема «корзина маркеров» предполагает наличие трех ключевых параметров [16,2].

- Средняя интенсивность или согласованная скорость передачи информации, бит/с. Как правило, интенсивность трафика не превышает согласованную скорость передачи информации.
- Согласованный размер всплеска (B_C), байт. Объем трафика, на который может быть превышен размер корзины маркеров в отдельно взятый момент времени. Иногда этот параметр называют также стандартным размером всплеска.
- Расширенный размер всплеска (B_E), байт. «Резервный фонд». Объем трафика, на который может быть превышен размер корзины маркера в экстренном случае. Всплеск, размер которого находится между согласованным и расширенным размером, разрешается, как правило, только для очень небольшой части трафика.

Четвертый ключевой параметр - интервал времени (time interval- TI)- зависит от средней интенсивности трафика и согласованного размера всплеска и вычисляется по формуле

$$TI = B_c + CIR. \qquad (3.3.1)$$

3.3.2. Выравнивание трафика

Выравнивание трафика (traffic shaping- TS) представляет собой механизм сглаживания поступающего на интерфейс потока трафика с целью недопущения перегрузки канала и удовлетворения требований поставщика услуг. В соответсвии с механизмом TS интенсивность пульсирующего трафика выравнивается до согласованной скорости передачи информации (CIR) путем постановки в очередь (буферизации) пакетов, интенсивность передачи которых превысила среднее значение. Буферизованные пакеты передаются по мере накопления достаточного числа маркеров. Передача поставленных в очередь пакетов планируется механизмом обслуживания очередей «первым пришел, первым обслужен» (first-in, first-out - FIFO) или взвешенным механизмом равномерного обслуживания очередей (Weighted Fair Queuing- WFQ) [7].

3.3.3. Дозирование трафика

В качестве инструмента дозирования трафика механизм TS использует уже выше упомянутую корзину маркеров, которая применяется для проверки поступающих пакетов на соответсвие заданному профилю.

Максимальный размер корзины равен сумме размеров согласованного (B_C) и расширенного (B_E) всплесков. Корзина пополняется маркерами, число которых равняется размеру согласованного всплеска (B_C), через каждый интервал времени $T= B_C/CIR$, где CIR представляет собой согласованную среднюю интенсивность потока трафика. Когда корзина становится полной, вновь прибывающие избыточные маркеры отбрасываются. Для каждого пакета из корзины вынимается число маркеров, которое равняется размеру пакета в байтах. Если для передачи пакета в корзине нашлось достаточное количество маркеров, пакет передается, а размер корзины уменьшается на

равное размеру переданного пакета количество маркеров. В противном случае пакет маркируется как не удовлетворяющий заданному профилю и ставится в очередь для последующей передачи. На рис.3.3.1. изображена реализация алгоритма «корзина маркеров» для механизма выравнивания трафика.

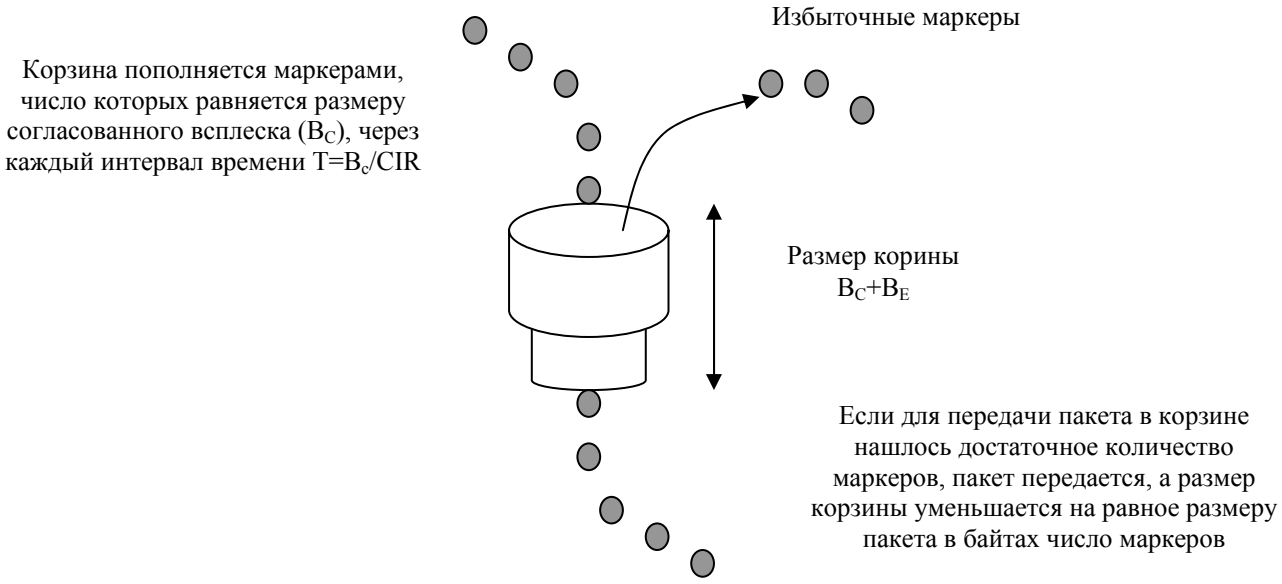


Рис. 3.3.1. Реализация алгоритма «корзина маркеров» для механизма выравнивания трафика.

3.4. Распределение ресурсов

В моменты перегрузки сети распределение ресурсов для отдельного потока трафика обуславливается порядком обслуживания поставленных в очередь пакетов. Порядок обслуживания поставленных в очередь пакетов определяет следующий пакет, который будет извлечен из очереди. Частота обслуживания пакетов, принадлежащих к одному и тому же потоку трафика, обуславливает его полосу пропускания или распределение ресурсов для данного потока трафика.

Традиционным для компьютерных сетей механизмом обслуживания очередей является механизм «первым пришел, первым обслужен» (first-in, first-out - FIFO), в соответствии с которым пакеты передаются в том порядке, в котором они были поставлены в выходную очередь. Несмотря на то что механизм FIFO досаточно прост и его легко реализовать, он не способен проводить различие между несколькими потоками трафика. Следовательно, механизм FIFO не может выделить требуемый объем ресурсов для потока или обеспечить его приоритетную по отношению к другим потокам обработку.

Взвешенный механизм равномерного обслуживания очередей (Weighted Fair Queuing- WFQ) представляет собой механизм обслуживания очередей с учетом принадлежности пакетов к тому или иному классу трафика. В соответствии с механизмом WFQ каждому потоку трафика назначается определенный вес, обуславливающий частоту обслуживания пакетов данного потока. Механизм WFQ поддерживает приоритетную обработку потоков с большим весом, а также защиту и равномерное обслуживание потоков с одинаковым весом путем применения максиминной схемы равномерного распределения ресурсов [5].

3.4.1. Поддержка функций QoS со стороны механизмов обслуживания очередей

Динамика передачи пакетов в сети делает ее уязвимой для случайных или постоянных перегрузок, наиболее часто возникающих в местах расположения маршрутизаторов, объединяющих сети с существенно различающимися полосами пропускания. В моменты нормального функционирования сети любая схема обслуживания очередей кажется идеальной, поскольку очередей как таковых в маршрутизаторах попросту нет. Однако при перегрузке сети маршрутизаторы начинают проводить

буферизацию пакетов и использовать механизмы обслуживания очередей с целью выбора пакета, который должен быть обработан на следующем шаге.

Алгоритм обслуживания очередей с поддержкой QoS должен как минимум обладать средством дифференцирования пакетов и средством определения уровня обслуживания каждого пакета. Кроме того, подобный алгоритм должен гарантировать качество обслуживания пакетов путем распределения ресурсов для каждого отдельного потока трафика или с помощью определения относительного приоритета потоков. Слеует отметить, что дифференцирование пакетов может осуществляться как на основании принадлежности пакета к потоку трафика, так и на основании принадлежности пакета к классу трафика, включающему в себя, ка правило, пакеты из различных потоков.

Кроме того, алгоритм обслуживания очередей с поддержкой QoS должен обеспечивать защиту и равномерную обработку всех потоков трафика с одинаковым приоритетом (например, трафика, доставляемого без гарантий).

Дополнительными требованиями к подобному алгоритму обслуживания очередей являются легкость реализации и поддержка управления доступом для потоков, нуждающихся в гарантированом предоставлении ресурсов.

Несмотря на то что алгоритм WFQ более сложен в реализации, нежели алгоритм FIFO, он удовлетворяет всем требованиям алгоритма обслуживания очередей с поддержкой QoS (чем, к сожалению, не может похвастаться алгоритм FIFO).

3.4.2. Алгоритм обслуживания очередей FIFO

FIFO представляет собой механизм ослуживания очередей, в соответсвии с которым порядок постановки пакетов в очередь совпалает с

порядком их извлечения из очереди для обработки (передачи) [51]. Очередь FIFO схематически преставлена на рис. 3.4.1.

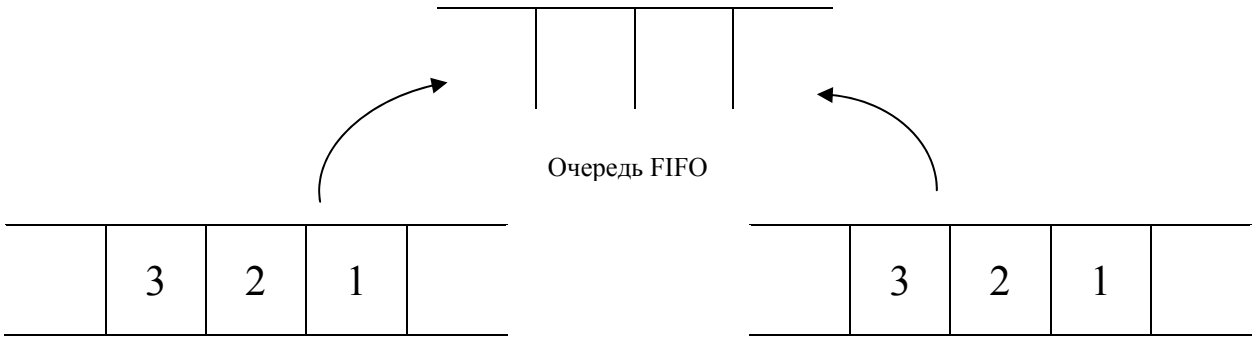


Рис. 3.4.1. Очередь FIFO

Как уже говорилось ранее, в соответствии с механизмом обслуживания FIFO порядок постановки пакетов в очередь совпадает с порядком их извлечения из очереди. На сегодняшний день механизм FIFO является наиболее распространенным механизмом обслуживания очередей, применяющимися в маршрутизаторах. Его реализация отличается своей простотой. К сожалению, механизм FIFO не способен обеспечить равномерное обслуживание потоков трафика с одинаковым приоритетом и защитить их от подавления потоками с неравномерной интенсивностью. Последние могут отобрать часть ресурсов у потоков, имеющих постоянную интенсивность и обслуживающихся сквозными адаптивными схемами управления потоком, такими, как схема управления динамическим окном протоколом TCP (Transmission Control Protocol- протокол управления передачей). Выясняется, что в соответствии с механизмом FIFO потоки трафика обслуживаются почти пропорционально их интенсивности. Подобная схема обслуживания очередей не является равномерной, поскольку она допускает преобладание потоков высокой интенсивности над всеми

остальными потоками трафика. Любой же равномерный алгоритм обслуживания очередей обеспечивает защиту от потоков с высокой интенсивностью по своей природе.

3.4.3. Максиминная схема равномерного распределения ресурсов

Если механизм FIFO не обеспечивает равномерного распределения ресурсов между потоками, то такой схемой равномерного распределения ресурсов, получившей широкое распространение, является максиминная схема равномерного распределения ресурсов.

Как правило, разные пользователи предъявляют различные требования к ресурсам. Следовательно, существует возможность классификации пользователей в порядке возрастания их требований к ресурсам. Ниже дано определение максиминной схемы равномерного распределения ресурсов.

- Ресурсы распределяются в порядке возрастания требований.
- Пользователь не может получить превышающий его потребности объем ресурсов.
- Ресурсы распределяются равномерно между пользователями с неудовлетворенными требованиями.

Рассмотрим пример. Предположим, что общий объем доступного ресурса равен 14 единицам. Требования к ресурсу пользователей A, B, C, D и E составляют 2, 2, 3, 5 и 6 единиц, соответсвенно. Распределение ресурса начинается с источника с наименьшими требованиями, который получает объем ресурса, равный отношению всего запаса ресурса к общему числу пользователей. Таким образом, в рассматриваемом нами случае пользователям A и B будет предоставлено $14 \div 5 = 2.8$ единиц ресурса. Однако требования пользователей A и B составляют всего лишь 2 единицы ресурса. В этом случае избыточный ресурс объемом 1.6 единиц (по 0.8 единиц с каждого

пользователя) равномерно распределяется между оставшимися тремя пользователями. Таким образом, пользователи C, D и E получают по $2.8 + (1.6 \div 3) = 3.33$ единицы ресурса. Следующим по объему предъявляемых требований к ресурсам является пользователь C. Требования пользователя C на 0.33 единицы «скромнее» предлагаемого ему объема ресурсов. Избыточный ресурс объемом 0.33 единицы равномерно распределяются между пользователями C и D, каждый из которых получает по $3.33 + (0.33 \div 2) = 3.5$ единицы ресурса.

Объем ресурсов, предоставляемый пользователю, рассчитывается по следующей формуле:

Объем ресурсов, предоставляемый пользователю=(весь запас ресурсов- объем уже распределенных ресурсов)÷ число пользователей, которым все еще требуются ресурсы

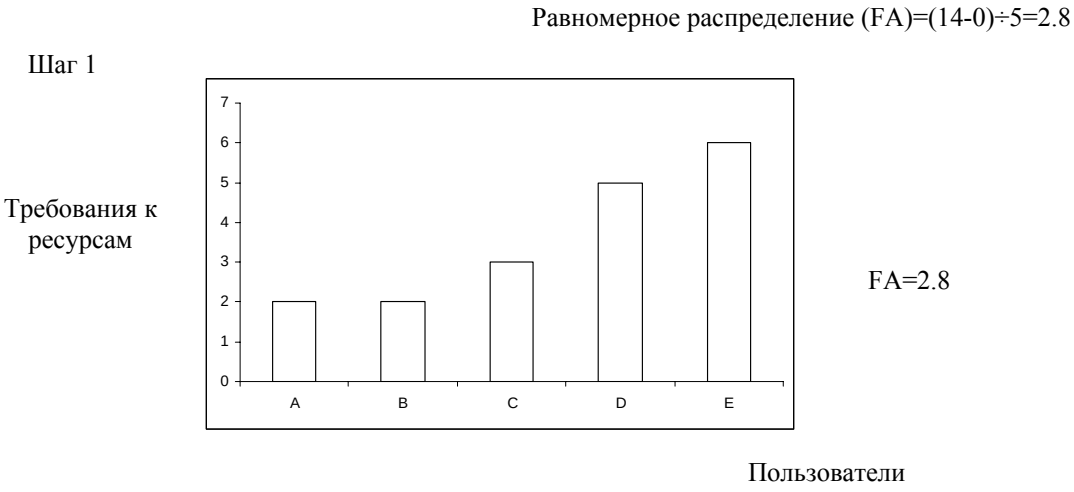


Рис. 3.4.2. Распределение ресурсов для пользователей A и B

Рассмотренный нами пример наглядно проиллюстрирован на рис 3.4.2. На шаге 1 (рис. 3.4.2) требования пользователей A и B удовлетворяются в полном объеме, так как они не превышают значения, полученного в результате равномерного распределения ресурсов между всеми пользователями.

Поскольку требования к ресурсам пользователей C, D и E превышают 2.8 единиц, они не могут быть удовлетворены на этом шаге. На следующем шаге объем невостребованных пользователями A и B ресурсов равномерно распределяются между оставшимися тремя пользователями - C, D и E.

На шаге 2 (рис. 3.4.3) в полном объеме удовлетворяется лишь требование пользователя C как не превышающее значения, полученного в результате равномерного распределения оставшихся ресурсов между пользователями C, D и E.

Рис. 3.4.3. Распределение ресурсов для пользователя C

Поскольку требования к ресурсам пользователей D и E превышают 3.33 единицы, они не могут быть удовлетворены на этом шаге. На следующем шаге объем невостребованных пользователем C ресурсов равномерно распределяется между оставшимися двумя пользователями - D и E.

На шаге 3 (рис. 3.4.4) значения, полученного в результате равномерного распределения ресурсов (3.5 единиц), недостаточно для покрытия запросов как пользователя D, так и пользователя E, объем

неудовлетворенных требований которых составляет 1.5 и 2.5 единиц, соответсвенно.

Рис. 3.4.4. Распределение ресурсов для пользователей D и E

Представленный выше способ распределения ресурсов получил название максиминной ссхемы равномерного распределения ресурсов. Нужно обратить внимание на то, что все пользователи с неудовлетворенными требованиями (т.е. с требованиями, объем которых превышает их максиминную равномерную долю) получают равные объемы ресурсов. Максиминная схема равномерного распределения ресурсов получила свое название в связи с тем, что пользователь с неудовлетворенными требованиями получает максимум из возможных минимальных равномеррных долей.

Максиминная схема равномерного распределения ресурсов, в которой каждому пользователю назначается определенный вес, получила название взвешенной максиминной схемы равномерного распределения ресурсов. В соответствии со взвешенной максиминной схемой равномерного распределения ресурсов каждому пользователю выделяется равномерная доля ресурсов, пропорциональная его весу.

3.4.4. Обобщенная схема разделения процессорного времени

При обработке потоков трафика, передаваемого по методу негарантированной доставки (а также всех других равновесных классов трафика), должна применяться схема, обеспечивающая справедливое обслуживание по типу максиминной схемы равномерного распределения ресурсов. Именно такой схемой и является обобщенная схема разделения процессорного времени (Generalized Processor Sharing - GPS).

В соотвветствии со схемой GPS каждый поток трафика помещается в собственную логическую очередь, после чего бесконечно малый объем данных из каждой непустой очереди обслуживается по круговому принципу. Необходимость обработки бесконечно малого объема данных на каждом круге обусловлена требованием обслуживания всех непустых очередей на любом конечном временном интервале. Таким образом, схема GPS фактически является справедливой в любой момент времени.

Если же всем потокам трафика назначить вес, то объем данных потока, обрабатываемый на каждом круге, будет пропорционален его весу. Подобное расширение схемы GPS фактически представляет собой взвешенную максиминную схему равномерного обслуживания.

3.5. Предотвращение перегрузки и политика отбрасывания пакетов

Политика отбрасывания пакетов представляет собой алгоритм управления очередью, применяющийся для регулирования ее длины. Традиционный алгоритм обслуживания очередей «первым пришел, первым обслужен» использует достаточно простую политику «отбрасывания хвоста», в соответсвии с которой любая попытка постановки пакета в полную очередь неминуемо завершится его отбрасыванием. Подобная «дискриминация»

пакетов продолжается до тех пор, пока длина очереди не уменьшится за счет передачи уже находящихся в ней пакетов. Алгоритм управления очередью, в соответствии с которым любая попытка постановки пакета в полную очередь неминуемо завершится его отбрасыванием, получил название алгоритма «отбрасывания хвоста». Поскольку отбрасывание пакета является сигналом о перегрузке сети, механизм «отбрасывания хвоста» сообщает о перегрузке сети лишь в момент фактического переполнения очереди. С целью поддержки механизма предотвращения заторов в сети используется так называемое окно перегрузки [19,25]. В результате отбрасывания пакета источник TCP-соединения уменьшает размер окна и перезапускает алгоритм, что приводит к резкому уменьшению трафика. С поведением TCP-источников в моменты работы алгоритма «отбрасывания хвоста» связана необходимость проведения управления очередью с целью сигнализации о перегрузке сети и контроля за размером очереди для снижения задержки обработки пакетов. Алгоритм произвольного раннего обнаружения представляет собой алгоритм предотвращения перегрузки, который вместо ожидания фактического переполнения очереди отбрасывает пакеты с ненулевой вероятностью, когда средний размер очереди превысит определенное минимальное пороговое значение.

3.5.1. Сигнальный протокол QoS

Для информирования сети о нуждах различных потоков трафика используется сигнальный протокол QoS- протокол резервирования ресурсов, который позволяет конечным приложениям, требующим определенные гарантированные услуги, проводить сквозную сигнализацию своих QoS-требований.

3.6. Коммутация

Главной заадачей маршрутизатора является эффективная и корректная коммутация пакетов. Широко используются и поддерживаются несколько методов коммутации - коммутация процессов, продвижение пакетов с помощью кэша маршрутов и скоростная коммутация пакетов.

В современных сетях, обеспечивающих передачу тысяч кратковременных потоков данных, рекомендуется использовать метод коммутации пакетов CEF. Этот метод играет важную роль и контексте реализации функций маршрутизации на основе политик, которые требуют просмотра таблицы маршрутизации.

3.6.1. Коммутация процессов

При коммутации процессов пакет поступает на входной интерфейс и ставится во входную очередь процесса, который коммутирует пакет. Когда планировщик запускает процесс, последний просматривает таблицу маршрутизации в поисках маршрута до точки назначения. Если маршрут найден, то из таблицы маршрутизации извлекается адрес точки следующей передачи пакета. Механизм управления доступом к среде передачи (Media Access Control- MAC) перезаписывает адрес точки слледующей передачи с помощью информации, взятой из таблицы протокола преобразования адресов (Address Resolution Protocol - ARP), после чего пакет устанавливается в выходную очередь интерфейса для отправки. Коммутация пакетов - это медленная, неэффективная и требующая интенсивного использования ресурсов процессора, поскольку каждое принятие решения о коммутации пакета требует просмотра таблицы маршрутизации и ARP- таблицы. При существовании нескольких путей к пункту назначения с одинаковой ценой

коммутация процессов предполагает распределение нагрузки для каждого пакета.

3.6.2. Продвижение пакетов с помощью кэша маршрутов

Продвижение пакетов с помощью кэша маршрутов решает некоторые проблемы, присущие коммутации процессов. Так, в соответствии с этим методом после того, как первый пакет был передан с помощью коммутации процессов, адрес пункта назначения, интерфейс точки следующей передачи и инкапсуляции MAC- адрес точки следующей передачи сохраняются в таблице под названием кэш маршрута. Таким образом, можно осуществить быструю коммутацию последующих пакетов, предназначенных для этого же пункта назначения, всего лишь найдя адрес пункта назначения в кэше маршрутов. При использовании этого метода принятие решения о коммутации производится в рамках того же прерывания, которое было вызвано поступлением пакета.

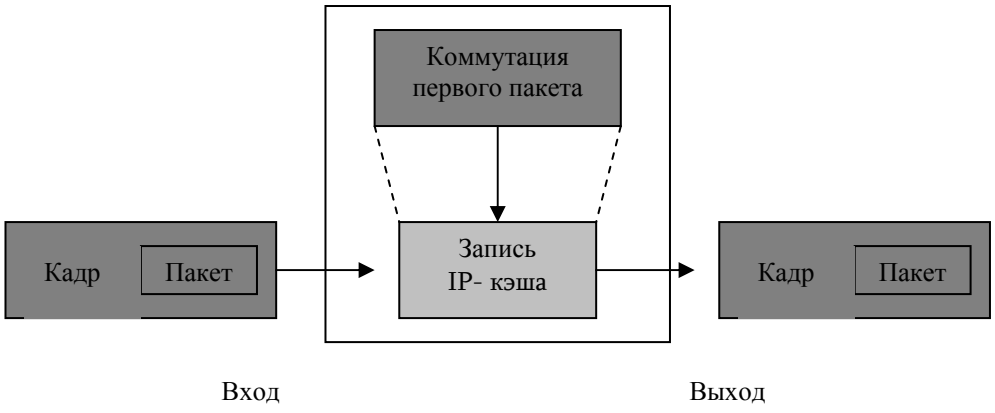


Рис. 3.6.1. Передача пакетов с помощью кэша маршрутов

Продвижение пакетов с помощью кэша маршрутов называют также быстрой коммутацией. Механизм продвижения пакетов с помощью кэша маршрутов схематически изображен на рис 3.6.1.

Механизм продвижения пакетов с помощью кэша маршрутов производит быстрый поиск префикса пункта назначения по требованию. Запись для пункта назначения в кэше маршрутов создается только после того, как маршрутизатор получит первый предназначенный к отправке в этот пункт пакет. Данный пакет передается с помощью коммутации процессов, однако все остальные пакеты, предназначенные для передачи в этот же пункт назначения, коммутируются путем их поиска в более быстром и эффективном кэше маршрутов. Записи в кэше маршрутов периодически устаревают. Более того, изменение в топологии сети может мгновенно сделать эти записи недействитнльными.

Подобная схема кэширования маршрутов по требованию эффективна только в тех случаях, когда большинство потоков трафика предназначаются для передачи в некоторое подмножество пунктов назначения в таблице маршрутизации.

3.6.3. CEF- коммутация

CEF является масштабируемым механизмом коммутации сетевого уровня. В механизме CEF представлено большое число улучшений по сравнению с традиционным методом коммутации с помощью кэша маршрутов. Механизм CEF избегает потенциального перенасыщения постоянными обращениями к кэшу с помощью топологической таблицы CEF, использующейся во время принятия решения о коммутации. В CEF- таблице зеркально отображается вся информация о маршрутах. Существует прямое соответствие между записями в CEF- таблице и префиксами таблицы маршрутизации. При создании записи CEF- таблицы автоматически разрешается проблема рекурсивного маршрута, вследствие чего механизм CEF позволяет достичь таких преимуществ, как производительность, масштабируемость, устойчивость сети и функциональность. Преимущества

механизма CEF особенно видны в больших, сложных сетях с динамической моделью трафика.

Наряду с CEF- таблицей поддерживается таблица смежностей. В таблице смежностей хранится информация заголовка канального уровня, которая заполняется любыми протоколами - ARP, открытым протоколом предпочтительного выбора кратчайшего пути и др., - которым необходимо установить отношение смежности. Каждый заголовок канального уровня смежного узла заранее вычисляется и сохраняется вместе с информацией об установке смежности.

Таблица CEF заполняется путем обратных вызовов из таблицы маршрутизации. После того как маршрут разрешен, соответсвующая ему запись в CEF- таблице указывает на точку следующей передачи пакета, которая должна представлять собой смежное устройство. Если данное отношение смежности найдено в таблице смежности, то указатель на него кэшируется в CEF- записи.

Кроме обычных физических смежностей, необходимо еще обработать особые типы смежности. Записи CEF- таблицы с префиксами, которые необходимо обработать специальным образом, заносятся в кэш с указанием на специальный тип смежности.

В отличие от модели кэша маршрутов, механизм CEF проводит поиск эффективной кэш- функции с целью выравнивания нагрузки для каждой пары «источник- пункт назначения». Более того, механизм CEF может производить выравнивание нагрузки для каждого пакета с помощью особого указателя.

3.7. Маршрутизация с точки зрения QoS

Главная функция маршрутизатора заключается в быстрой и эффективной коммутации входящего трафика на соответсвующие выходные

интерфейсы согласно информации, хранящейся в таблице продвижения пакетов. Метод продвижения пакетов, учитывающий топологию сети, называющийся методом скоростной коммутации пакетов, обладает преимуществами перед методом, базирующимся на кэшировании пакетов, что обусловлено совпадением таблицы продвижения пакетов с таблицей маршрутизации.

Маршрутизация на основе QoS - это механизм маршрутизации, в соответствии с которым пути для потоков трафика определяются на основе некоторых сведений о наличии ресурсов сети и на основе требований потоков к качеству обслуживания.

Механизм маршрутизации на основе QoS- политик использует следующие важные расширения.

- Протокол маршрутизации передает метрики с динамической информацией о наличии ресурсов (QoS) (например, информация о доступной полосе пропускания, уровне потери пакетов или задержке).
- Протокол маршрутизации должен вычислять не только оптимальный путь, но и другие возможные пути на основе их QoS- доступности.
- В каждом потоке передаются сведения о необходимости качестве услуг (QoS). QoS- информация может передаваться в байте типа обслуживания заголовка IP- протокола. Маршрутный путь для потока выбирается в соответствии с его QoS- требованиями.

Следует отметить, что маршрутизация на основе QoS влечет за собой существенные сложности. По своей природе метрики QoS- доступности очень динамичны. Таким образом, обновления маршрутов происходят более часто, поглащая ценные сетевые ресурсы и занимая циклы работы процессора маршрутизатора. Поток может часто «колебаться» между альтернативными

QoS- маршрутами из-за изменений QoS- метрики нестабильного пути. Более того, частая смена маршрутов может привести к увеличению дрожания, т.е. колебания задержек, которые испытывает конечный пользователь. За исключением упомянутых проблем, маршрутизация на основе QoS является весьма ценным атрибутом QoS- сети.

Открытый протокол предпочтительного выбора кратчайшего пути (Open Shortest Path First - OSPF) и протокол обмена информацией о маршрутах между промежуточными системами (Intermediate System-to-Intermediate System - IS-IS), являющиеся очень распространенными протоколами внутреннего шлюза сетей поставщиков услуг, вместе с объявлением о состоянии канала могут передавать и байт типа обслуживания. Тем не менее на сегодняшний день байт типа обслуживания все еще устанавливается равным нулю и, таким образом, не используется. Тема QoS-маршрутизации пока что находится в стадии обсуждения организациями по стандартизации.

Между тем протоколы OSPF и IS-IS были расширены с целью поддержки механизма управления трафиком на основе технологии многопротокольной коммутации меток путем включения информации о ресурсах канала в объявление маршрута. Несмотря на то что эти протоколы все еще основываются на адресе пункта назначения, в каждом маршруте теперь находится дополнительная информация, которую такие протоколы, как многопротокольный коммуттации меток, могут использовать для обеспечения механизма управления трафиком. Протоколы маршрутизации OSPF и IS-IS с расширениями управления трафиком представляют собой практический компромисс между современными протоколами на основе пункта назначения и QoS- маршрутизацией.

3.7.1. Маршрутизация на основе политики

Сегодня маршрутизация в компьютерных сетях основывается исключительно на IP- адресе пункта назначения пакета. Маршрутизация на основе другой информации, передаваемой в заголовке или теле IP- пакета невозможна при использовании современных динамических протоколов маршрутизации. Для решения этой проблемы и предназначена маршрутизация на основе политик.

Архитектура дифференцированного обслуживания была разработана с целью обеспечения поддержки легкомасштабируемых дифференцированных услуг в компьютерных сетях. Для маркировки IP- пакета в соответствии с требуемым уровнем QoS ипользуется недавно стандартизированное поле кода дифференцированной услуги, которое определяет PHB- политику, применяемую при транспортировке данного пакета в пределах домена дифференцированной услуги. Формирование и маркировка трафика осуществляется на границе сети дифференцированной услуги. PHB- политика- это наблюдаемая извне политика поведения сетевого узла в отношении пакетов с определенным значением поля кода дифференцированной услуги. PHB- политика может быть определена в терминах приоритета в предоставлении ресурсов по отношению к другим PHB- политикам или же с помощью таких измеряемых ресурсов сети, как задержка пакетов, уровень потери пакетов или дрожание трафика.

Предположим, что поставщику услуг компьютерной сети может понадобиться разделить трафик, направляющийся к определенному серверу, на два класса: с приоритетом 3 (передается по выделенной быстрой линии связи) и с приоритетом 0. Несмотря на то что пункт назначения один и тот же, трафик направляется по различным выделенным каналам, соответствующим каждому значению IP- приоритета. Аналогичным образом, маршрутизация может быть основана на длине пакета, адресе источника,

потоке, определенной парой «адрес источника - адрес пункта назначения» и портами протокола управления передачей и протокола передачи дейтаграмм пользователя, битами типа обслуживания, битами поля IP- приоритета и т.д. Подобная гибкая схема маршрутизации обычно и называется маршрутизацией на основе политики.

Маршрутизация на основе политики не опирается на какой-нибудь динамический протокол маршрутизации, а использует локальную статическую конфигурацию маршрутизатора. Она позволяет маршрутизировать трафик на основе определенной политики, даже если информация о маршруте к пункту назначения потока недоступна, или вообще игнорируя динамическую информацию о маршруте. Более того, можно указать маршрутизатору на необходимость установки значения поля IP–приоритета пакета для трафика, маршрутизируемого на основе политики.

Поскольку конфигурация маршрутизации на основе политики является статической, это может привести к отбрасыванию трафика в случае, если сконфигурированная точка следующей передачи пакета становится недоступной. Для проверки доступности точки следующей передачи механизм маршрутизации на основе политики может использовать протокол обнаружения. Когда механизм маршрутизации на основе политики не может обнаружить точку следующей передачи в таблице протокола обнаружения, он прекращает передавать соответствующие пакеты в эту точку и перенаправляет их, используя таблицу маршрутизации. Маршрутизатор возвращается к осуществлению маршрутизации на основе политики, когда точка следующей передачи опять становится доступной (это определяется с помощью протокола обнаружения). Эту функциональность можно применять только в том случае, если на интерфейсе активизирован протокол обнаружения.

3.8. Основные показатели в компьютерных сетях

Теперь необходимо отметить важность управления ресурсами сети с точки зрения качества обслуживания и получения наилучших характеристик сети, в частности производительности.

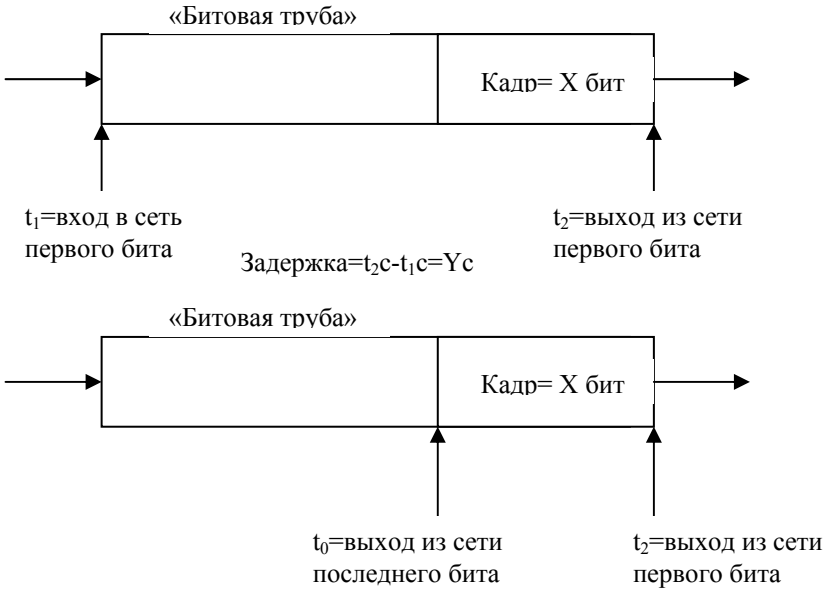
Внедрение механизмов QoS предполагает обеспечение со стороны сети соединения с определенными ограничениями по производительности. Основными характеристиками производительности сетевого соединения являются полоса пропускания, задержка, дрожание и уровень потери пакетов.

Полоса пропускания относится к имеющейся мощности трафика какого-либо канала. Термин полоса пропускания используется для описания номинальной пропускной способности среды передачи информации, протокола или соединения. Этот термин достаточно эффективно определяет «ширину канала», требующуюся приложению для взаимодействия по сети. Как правило, каждое соединение, нуждающееся в гарантированном качестве обслуживания, требует от сети резервирования минимальной полосы пропускания.

Под задержкой обычно понимают отрезок времени, необходимый для передвижения пакета от источника до пункта назначения через объединенную сеть. Задержка зависит от многих факторов, включая полосу пропускания промежуточных каналов сети, очереди в порт каждого маршрутизатора на пути передвижения пакета, перегруженность сети и физическое расстояние, на каторое необходимо переместить пакет.

На рис. 3.8.1 представлен способ вычисления задержки сети и полосы пропускания для работы с данными [53]. Передача информации происходит посредством пересылки последовательности кадров, которые содержат пакеты и имеют определенную длину в битах (X бит на рисунке). Путь передачи данных показан, как пассивная "битовая труба" с постоянными характеристиками и отсутствием хранения и преобразования битов. Время от

момента входа в сеть первого бита данного кадра и до момента выхода этого бита из сети определяет задержку. Отметим, что задержка может измениться во время пересылки кадра по сети, а также иметь максимум, минимум, среднюю величину, стандартное отклонение и так далее. Задержка должна измеряться от одного конца до другого, от отправителя до получателя, между двумя точками сети или в нескольких различных точках по пути следования.



Полоса пропускания=X бит/ (t_0c- t_2c)=Z бит/с

Рис.3.8.1. Взаимосвязь между полосой пропускания и задержкой

Полоса пропускания определяется через биты. Отметим, что в контексте приложений для работы с данными в определении используется цифровая версия понятия полосы пропускания, а не аналоговая версия диапазона частот. Полоса пропускания измеряется в битах в секунду. Поэтому цифровая полоса пропускания - это число битов кадра, деленное на время от момента выхода из сети первого бита кадра до момента выхода из сети последнего бита кадра. Это лишь один из возможных методов измерения. Отметим, что полоса пропускания для данного кадра может значительно

изменяться при перемещении кадров от линии доступа к другим частям сети и далее.

В рассмотренном примере (рис. 3.8.1) не учтена пересылка кадра ошибок передачи и избыточных битов. Если кадр и содержащийся в нем пакет будут приняты с ошибкой, то должна быть выполнена повторная пересылка, что увеличит общее запаздывание при передаче файла. Кроме того, некоторые биты кадра или пакета не несут информационного содержания, а используются внутри самой сети. Такие биты называются избыточными. Учитывая влияние ошибок и избыточных битов, можно вычислить общую пропускную способность сети. Чем меньше ошибок и избыточных битов, тем ближе пропускная способность к времени пересылки.

Задержка при передаче пакетов, или латентность, на каждом переходе состоит из задержки сериализации, задержки распространения и задержки коммутации. Ниже приведены определения каждого из названных выше типов задержки.

- **Задержка сериализации.** Время, которое требуется устройству на передачу пакета при заданной ширине полосы пропускания. Задержка сериализации зависит как от ширины полосы пропускания канала передачи информации, так и от размера передаваемого пакета. Довольно часто задержку сериализации называют задержкой передачи.
- **Задержка распространения.** Время, которое требуется переданному биту информации для достижения принимающего устройства на другом конце канала. Эта величина довольно существенна, поскольку в наилучшем случае скорость передачи информации соизмерима со скоростью света. Нужно обратить внимание на то, что задержка распространения зависит от расстояния и используемой среды передачи информации, а не от полосы пропускания. Для линий связи глобальных сетей задержка распространения измеряется в миллисекундах.

- **Задержка коммутации.** Время, которое требуется устройству, получившему пакет, для начала его передачи следующему устройству.

Обычно каждый из пакетов, принадлежащий одному и тому же потоку трафика, передается с различным значением задержки. Задержка при передаче пакетов меняется в зависимости от состояния промежуточных сетей.

В том случае, если сеть не испытывает перегрузки, пакеты не ставятся в очередь в маршрутизаторах, а общее время задержки при передаче пакета состоит из суммы задержки сериализации и задержки распространения на каждом промежуточном проходе. В этом случае можно говорить о минимальной возможной задержке при передаче пакетов через заданную сеть. Следует заметить, что задержка сериализации становится незначительной по сравнению с задержкой распространения при передаче пакета по каналу с большей пропускной способностью.

Если же сеть перегружена, задержки при организации очередей в маршрутизаторах начинают влиять на общую задержку при передаче пакетов и приводят к возникновению разницы в задержке при передаче различных пакетов одного и того же потока. Колебание задержки при передаче пакетов получило название дрожания при передаче пакетов.

Дрожание пакетов имеет большую важность, поскольку именно оно определяет максимальную задержку при приеме пакетов в конечном пункте назначения. Принимающая сторона, в зависимости от типа используемого приложения, может попытаться компенсировать дрожание пакетов за счет организации приемного буфера для хранения принятых пакетов на время, меньшее или равное верхней границе дрожания. К этой категории относятся приложения, ориентированные на передачу/прием непрерывных потоков данных.

Нужно обратить внимание, что задержка сериализации становится незначительной по сравнению с задержкой распространения по мере увеличения полосы пропускания канала. Задержка коммутации

пренебрежимо мала в случае отсутствия пакетов в очередях маршрутизаторов, однако склонна к существенному увеличению при росте размеров очередей.

Уровень потери пакетов определяет количество пакетов, отбрасываемых сетью во время передачи. Основными причинами потери пакетов являются перегрузка сети и повреждение пакетов во время передачи по линии связи. Чаще всего отбрасывание пакетов происходит в местах перегрузки, где число поступающих пакетов намного превышает верхнюю границу размера выходной очереди. Кроме того, отбрасывание пакетов может быть вызвано недостаточным размером входного буфера. Как правило, уроовень потери пакетов выражается как доля отброшенных пакетов за определенный интервал времени.

Некоторые приложения не способны нормально функционировать или же функционируют крайне неэффективно в случае потери пакетов. Подобные приложения требуют от сети гарантии надежной доставки всех пакетов.

Как правило, хорошо спроектированные сети характеризуются очень низким значением потери пакетов. Потеря пакетов также несвойственна приложениям, для которых были заранее зарезервированы требуемые этими приложениями ресурсы. Отбрасывание пакетов является неизбежным явлением при негарантированной доставке трафика, хотя и в этом случае оно обуславливается крайней необходимостью. Следует отметить, что отброшенные пакеты указывают на неэффективное ипользование ресурсов сети, часть которых была потрачена на доставку пакетов в точку, где они были потеряны.

Реализация в компьютерных сетях функций качества обслуживания позволяет обеспечить надежную доставку данных приложений посредством контроля за доступом к сети, задержкой, уровнем потери, качеством передаваемых пакетов и полосой пропускания каналов передачи информации.

Глава 4. Основные задачи управления ресурсов

4.1. Разработка модели M/M/m для расчета задержки в сети

В этой главе рассмотрена модель $M/M/m$ с m обслуживающих приборов (или каналов в линии передачи данных). Требования, стоящие в начале очереди, направляются к любому имеющемуся в наличии обслуживающему прибору.

Записывая уравнения равновесия для стационарных вероятностей p_n и устремляя $\delta \to 0$, получаем

$$\lambda p_{n-1} = n\mu p_n,\ n \le m$$
$$\lambda p_{n-1} = n\mu p_n,\ n > m.$$

Из этих уровнений следует

$$p_n = \begin{cases} p_0 \dfrac{(m\rho)^n}{n!},\ n \le m, \\ p_0 \dfrac{m^m \rho^n}{m!},\ n > m, \end{cases} \qquad (4.1.1)$$

где

$$\rho = \frac{\lambda}{m\mu} < 1 \qquad (4.1.2)$$

Можно вычислить p_0, используя формулу (4.1.1) и условие $\sum_{n=0}^{\infty} p_n = 1$.

Получаем

$$p_0 = \left[1 + \sum_{n=1}^{m-1} \frac{(m\rho)^n}{n!} + \sum_{n=m}^{\infty} \frac{(m\rho)^n}{m!} \frac{1}{m^{n-m}} \right]^{-1}$$

и окончательно

$$p_0 = \left[\sum_{n=0}^{m-1} \frac{(m\rho)^n}{n!} + \frac{(m\rho)^m}{m!(1-\rho)} \right]^{-1} \qquad (4.1.3)$$

Вероятность того, что поступившее требование обнаружит, что в системе все обслуживающие приборы заняты, и будет поставлено в очередь для ожидания, равна

$$P\{\text{встать в очередь}\}=\sum_{n=m}^{\infty} p_n = \sum_{n=m}^{\infty} \frac{p_0 m^m \rho^n}{m!} = \frac{p_0 (m\rho)^m}{m!} \sum_{n=m}^{\infty} \rho^{n-m}$$

и окончательно

$$P_Q \triangleq P\{\text{встать в очередь}\}=\frac{p_0 (m\rho)^m}{m!(1-\rho)}, \qquad (4.1.4)$$

где p_0 можно найти из (4.1.3).

Математическое ожидание числа требований, ожидающих в очереди, равно

$$N_Q = \sum_{n=0}^{\infty} n p_{m+n}$$

Используя (4.1.1), получаем

$$N_Q = \sum_{n=0}^{\infty} n p_0 \frac{m^m \rho^{m+n}}{m!} = \frac{p_0 (m\rho)^m}{m!} \sum_{n=0}^{\infty} n\rho^n .$$

Используя (4.1.4) и равенство $(1-\rho)\sum_{n=0}^{\infty} n\rho^n = \rho/(1-\rho)$ окончательно будем иметь

$$N_Q = P_Q \frac{\rho}{1-\rho} \qquad (4.1.5)$$

Заметим, что

$$\frac{N_Q}{P_Q} = \frac{\rho}{1-\rho}$$

дает условное математическое ожидание числа требований, ожидающих в очереди при поступлении требования, при условии, что это требование направляется в очередь для ожидания.

Используя теорему Литтла $N = \lambda T$ (где N - число требований в системе N, T - средняя задержка, λ - средняя скорость поступления требований) и равенство (4.1.5), получаем среднее время W, которое требование ожидает в очереди,

$$W = \frac{N_Q}{\lambda} = \frac{\rho P_Q}{\lambda(1-\rho)}. \qquad (4.1.6)$$

Следовательно, средняя задержка требования равна

$$T = \frac{1}{\mu} + W = \frac{1}{\mu} + \frac{\rho P_Q}{\lambda(1-\rho)}.$$

и учитывая, что $\rho = \lambda / m\mu$, окончательно получаем

$$T = \frac{1}{\mu} + W = \frac{1}{\mu} + \frac{P_Q}{m\mu - \lambda} \qquad (4.1.7)$$

Снова, используя теорему Литтла, находим, что среднее число требований в системе равно

$$N = \lambda T = \frac{\lambda}{\mu} + \frac{\lambda P_Q}{m\mu - \lambda}.$$

и с учетом $\rho = \lambda / m\mu$ получаем [52]

$$N = m\rho + \frac{\rho P_Q}{1-\rho}.$$

4.2. Система M/M/ ∞ - система с бесконечным числом обслуживающих приборов

В предельном случае, когда в системе $M/M/m$ имеем $m = \infty$, получаем

$$\lambda p_{n-1} = n\mu p_n,\ n = 1,2,...$$

и, таким образом,

$$p_n = p_0 \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!},\ n = 1,2,...$$

Из условия $\sum_{n=0}^{\infty} p_n = 1$ находим

$$p_0 = \left[1 + \sum_{n=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!}\right]^{-1} = e^{-\lambda/\mu},$$

поэтому окончательно

$$p_n = \left(\frac{\lambda}{\mu}\right)^n \frac{e^{-\lambda/\mu}}{n!},\ n = 0,1,...$$

Следовательно, в стационарном состоянии распределение вероятностей числа требований в системе – пуассоновское распределение с параметром λ/μ. Среднее число требований в системе равно

$$N = \lambda/\mu.$$

Согласно теореме Литтла, средняя задержка равна N/λ и

$$T = 1/\mu.$$

Последнее равенство можно получить иначе, используя тот факт, что в системе M/M/∞ нет ожидания в очереди. Можно показать, что распределение вероятностей числа требований в системе пуассоновское, даже если распределение времени обслуживания не является экспоненциальным.

4.3. Система M/M/m/m с потерями и с m обслуживающими приборами

Эта система подобна системе M/M/m за исключением того, что, если требование при поступлении в систему обнаружит, что все m обслуживающих приборов заняты, оно не поступит в систему. В сетях передачи данных такая модель может использоваться для исследования системы, в которой моменты поступления соответствуют запросам на установление виртуальных цепей между двумя узлами, а максимально возможное число виртуальных цепей равно m. Средняя длительность обслуживания $1/\mu$ в этом случае равна среднему времени использования виртуальной цепи.

Имеем

$$\lambda p_{n-1} = n\mu p_n,\ n = 1,2,...,m,$$

так что

$$p_n = p_0\left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!},\ n = 1,2,...,m.$$

С учетом равенства $\sum_{n=0}^{m} p_n = 1$ получаем

$$p_0 = \left[\sum_{n=0}^{m} \left(\frac{\lambda}{\mu} \right)^n \frac{1}{n!} \right]^{-1}$$

Вероятность того, что поступившее требование обнаружит, что все m обслуживающих приборов заняты, и, следовательно, будет потеряно, равна

$$p_m = \frac{(\lambda / \mu)^m / m!}{\sum_{n=0}^{m} (\lambda / \mu)^n / n!}.$$

4.4. Оптимальная маршрутизация

Для того, чтобы оценить характеристики алгоритма маршрутизации, необходимо количественно определить понятие перегрузки в сети. В этом разделе формулируются некоторые модели, называемыми потоковыми моделями, основанные на интенсивностях трафика, поступающего в линии сети. Эти модели будут ипользоваться при постановке задачи оптимальной маршрутизации.

Перегрузки в сетях передачи данных количественно можно определить с использованием статистик входных процессов, описывающих очереди в сети. Эти статистики определяют распределение длины очереди и время ожидания пакета в каждой линии [44]. Очевидно, что при хорошей маршрутизации должны быть малы средние значения и дисперсии задержки пакета в каждой очереди. К сожалению, обычно трудно бывает представить количественный параметр в виде единственного функционала, подходящего для оптимизации. Главной причиной этого является то, что, как правило, нет явного аналитического выражения для средних значений или дисперсий длин очередей в сетях передачи данных.

Удобной альтернативой является измерение нагрузок в линиях с использованием среднего трафика, проходящего по линии. Точнее говоря, мы

предполагаем, что статистика потока, поступающего в любую линию (i, j) меняется только из-за обновлений маршрутов, и под нагрузкой в линии (i, j) мы будем понимать интенсивность поступающего трафика F_{ij}. Будем называть F_{ij} потоком, проходящим по линии (i, j), и измерять его в единицах данных в секунду, где единицами данных могут быть биты, пакеты, сообщения и т.д.

В потоковых моделях делается неявное предположение, что статистика трафика, поступающего в сеть, не меняется во времени. Такое допущение является разумным, когда эта статистика меняется очень медленно по сравнению со средним временем, наобходимым для уменьшения очередей в сети, и когда потоки в линиях измеряются путем временного усреднения.

Выражение вида

$$\sum_{(i,j)} D_{ij}\left(F_{ij}\right), \tag{4.4.1}$$

где каждая функция D_{ij} является монотонно возрастающей, часто выбирают в качестве стоимостной функции при оптимизации. Также используется другая формула:

$$D_{ij}\left(F_{ij}\right) = \frac{F_{ij}}{C_{ij} - F_{ij}} + d_{ij} F_{ij} \tag{4.4.1.а}$$

где C_{ij} - пропускная способность линии (i, j), измеряемая в тех же единицах, что и F_{ij}, а d_{ij} - задержка из-за обработки и распространения. Другой стоимостной функцией с аналогичными качественными свойствами является

$$\max_{(i,j)} \left\{ \frac{F_{ij}}{C_{ij}} \right\} \tag{4.4.1.б}$$

т.е. максимум коэффициента использования линии.

Теперь сформулируем задачу оптимальной маршрутизации. Для каждой пары $\omega = (i, j)$ различных узлов i и j (также называемой парой

отправитель-адресат или ОА-пара) входной процесс поступающих пакетов предполагается стационарным и имеет интенсивность r_ω. Таким образом, r_ω - интенсивность входного трафика, поступающего в сеть в узле i и адресованного узлу j. Цель маршрутизации состоит в том, чтобы трафик интенсивности r_ω разделить между несколькими путями от отправителя к адресату так, чтобы общий получающийся в результате поток по линиям в сети минимизировал стоимостную функцию (4.4.1). Чтобы дать более точные формулировки задачи, введем обозначения

W - множество всех ОА-пар.

P_ω - множество всех ориентированных путей, соединяющих узлы отправителя и адресата ОА-пары ω.

x_p - поток (единицы данных в секунду) по пути p. Тогда набор всех путевых потоков $\{x_p \mid \omega \in W, p \in P_\omega\}$ должен удовлетворять ограничениям

$$\sum_{p \in P} x_p = r_\omega \quad \text{для всех } \omega \in W,$$

$$x_p \geq 0 \qquad \text{для всех } p \in P_\omega,\ \omega \in W.$$

Суммарный поток F_{ij} по линии (i, j) равен сумме путевых потоков проходящих по этой линии,

$$F_{ij} = \sum_{\substack{\text{По всем путям } p, \\ \text{содержащим } (i, j)}} x_p$$

Рассмотрим стоимостную функцию вида

$$\sum_{(i,j)} D_{ij}(F_{ij})$$

и задачу нахождения путевых потоков $\{x_p\}$, которые минимизируют эту стоимостную функцию при вышеуказанных ограничениях.

Если в стоимостной функции (4.4.1) заменить суммарные потоки F_{ij} на путевые, то задачу можно сформулировать в следующем виде

минимизировать $\sum_{(i,j)} D_{ij} \left[\sum_{\substack{\text{По всем путям } p, \\ \text{содержащим } (i, j)}} x_p \right],$

при ограничениях $\sum_{p \in P_\omega} x_p = r_\omega$ для всех $\omega \in W$,

$x_p \geq 0$ для всех $p \in P_\omega$, $\omega \in W$.

Таким образом, задача сформулирована в терминах неизвестных путевых потоков $\{x_p \mid p \in P_\omega, \omega \in W\}$. Это и является основной задачей оптимальной маршрутизации, которая будет рассмотрена.

Только что сформулированная задача оптимальной маршрутизации поддается аналитическому исследованию и распределенному численному решению. Однако она имеет некоторые ограничения, на которые стоит обратить внимание. Основное ограничение касается выбора стоимостной функции (4.4.1) в качестве меры. Этот выбор основан на гипотезе, что достаточно хорошую маршрутизацию можно сделать оптимизируя средние уровни проходящего по линиям трафика и не обращая внимания на другие вероятностные характеристики трафика. Таким образом. стоимостная функция (4.4.1) не чувствительна к нежелательным явлениям, связанным с большой дисперсией и корреляциями интервалов между моментами поступления пакетов и моментами передачи.

Также будет показано, что при оптимальной маршрутизации трафик направляется по тем путям, которые являются кратчайшими по отношению к некоторым «длинам» линий, зависящим от потоков, проходящих по этим линиям.

Вспомним вид стоимостной функции $\sum_{(i,j)} D_{ij}\left(F_{ij}\right)$

Здесь F_{ij}-суммарный поток (в единицах данных в секунду), проходящий по линии (i, j):

$$F_{ij} = \sum_{\substack{\text{По всем путям } p, \\ \text{содержащим } (i,j)}} x_p \qquad (4.4.2)$$

где x_p- поток, проходящий по пути p. Для каждой ОА-пары ω существуют ограничения

$$\sum_{p \in P_\omega} x_p = r_\omega ,$$

$$x_p \geq 0 \text{ для всех } p \in P_\omega ,$$

где r_ω - известная интенсивность входного потока трафика ОА-пары ω, а P_ω - множество ориентированных путей для ω. Задача, которая формулируется с использованием неизвестного вектора путевых потоков $x = \{x_p \mid p \in P_\omega, \omega \in W\}$, записывается в виде

минимизировать $\sum_{(i,j)} D_{ij} \left[\sum_{\substack{\text{По всем путям } p, \\ \text{содержащим } (i,j)}} x_p \right]$,

при ограничениях $\sum_{p \in P_\omega} x_p = r_\omega$ для всех $\omega \in W$, (4.4.3)

$$x_p \geq 0 \text{ для всех } p \in P_\omega ,\ \omega \in W .$$

Далее будем описывать оптимальную маршрутизацию с ипользованием производных функций D_{ij}. Предполагается, что каждая D_{ij} является дважды дифференцируемой функцией от F_{ij} и определена в полуинтервале $[0, C_{ij})$, где значение C_{ij} равно либо некоторой положительной константе, либо бесконечности. Предполагается, что первая и вторая производные D_{ij}, обозначаемые соответственно через D'_{ij} и D''_{ij}, строго положительны для всех F_{ij} из $[0, C_{ij})$. Отсюда, в частности, следует, что D_{ij} является выпуклой, монотонно возрастающей функцией от F_{ij} на всем полуинтервале $[0, C_{ij})$. Кроме того предполагается, что $D_{ij}(F_{ij} \to \infty)$ при $F_{ij} \to C_{ij}$.

Описание решений задачи оптимальной маршрутизации будет получено из следующего необходимого и достаточного условия оптимальности.

Лемма. Пусть f - некоторая дифференцируемая выпуклая функция n-мерного вектора $x=(x_1,...,x_n)$ и X - некоторое выпуклое множество векторов. Тогда $x^* \in X$ будет оптимальным решением задачи

$$\text{минимизировать } f(x)$$

$$\text{при ограничениях } x \in X \tag{4.4.4}$$

тогда и только тогда, когда

$$\sum_{i=1}^{n} \frac{\partial f(x^*)}{\partial x_i}(x_i - x_i^*) \geq 0 \text{ для всех } x \in X, \tag{4.4.5}$$

где $\partial f(x^*/\partial x_i)$- значение первой производной f по i-й координате x_i в точке x^*.

Предположим, что x^*- оптимальное решение и для каждого $x \in X$ рассмотрим функцию $g(\alpha)=f[x^*+\alpha(x-x^*)]$ скалярной переменной α. Тогда $g(\alpha)$ достигает минимума на отрезке $[0,1]$ в точке $\alpha=0$ и поэтому $dg(0)/d\alpha \geq 0$. Используя правило дифференцирования сложной функции, имеем

$$\frac{dg(0)}{d\alpha} = \sum_{i=1}^{n} \frac{\partial f(x^*)}{\partial x_i}(x_i - x_i^*),$$

и следовательно, неравенство (4.4.5) доказано.

Для доказательства достаточности предположим, что неравенство (4.4.5) справедливо, но x^* не является оптимальным решением. Тогда мы должны прийти к противоречию. Действительно, пусть $\tilde{x} \in X$. Такое, что $f(\tilde{x}) < f(x^*)$, и рассмотрим функцию $g(\alpha)=f[x^*+\alpha(\tilde{x}-x^*)]$. Тогда $dg(0)/d\alpha \geq 0$ (на основании (5.54)), $f(x^*)=g(0)>g(1)=f(\tilde{x})$. Эти условия противоречат выпуклости $g(\alpha)$ и, следовательно, выпуклости f. На этом доказательство заканчивается.

Так как и стоимостная функция и все ограничительные множества задачи минимизации (4.4.3) являются выпуклыми, то можно применить лемму. Пусть x будет вектором путевых потоков x_p. Для конкретности

предположим, что пути p последовательно пронумерованы. Тогда если $W = \{\omega_1, \omega_2, ..., \omega_m\}$ - множество ОА-пар, то соответствующими множествами путей являются

$$P_{\omega_1} = \{1,2,...,n_1\},$$

$$P_{\omega_2} = \{(n_1+1),...,n_2\},$$

...

$$P_{\omega_m} = \{(n_{m-1}+1),...,n_m\},$$

где $n_1, n_2, ..., n_m$ - некоторые целые числа, такие, что $n_1 < n_2 < ... < n_m$. Координатами вектора путевых потоков x будут путевые потоки $x_1, x_2, ..., x_{n_m}$, т.е.

$$x = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n_m} \end{Bmatrix}.$$

Обозначим через $D(x)$ стоимостную функцию задачи (4.4.3)

$$D(x) = \sum_{(i,j)} D_{ij} \left[\sum_{\substack{\text{По всем путям } p, \\ \text{содержащим } (i,j)}} x_p \right]$$

а через $\partial D(x)/\partial x_p$ - частную производную D по x_p. Тогда

$$\frac{\partial D(x)}{\partial x_p} = \sum_{\substack{\text{По всем линиям } (i,j) \\ \text{на пути } p}} D'_{ij} \qquad (4.4.6)$$

где первые производные D'_{ij} берутся при суммарных потоках, соответствующих x. Из равенства (4.4.6) видно, что $\partial D/\partial x_p$ является длиной пути p, если длину каждой линии (i,j) положить равной первой производной D'_{ij} взятой в x. Поэтому далее $\partial D/\partial x_p$ называется первопроизводной длиной пути p.

В соответсвтсии с леммой $x^* = \{x_p^*\}$ является вектором оптимальных путевых потоков, если он удовлетворяет ограничениям задачи маршрутизации и условию (4.4.5). Это условие можно переписать в виде

$$\sum_{\omega \in W} \sum_{p \in P_\omega} \frac{\partial D(x^*)}{\partial x_p} (x_p - x_p^*) \geq 0 \qquad (4.4.7)$$

для всех x_p, удовлетворяющих ограничениям

$$\sum_{p \in P_\omega} x_p = r_\omega \,,\ x_p \geq 0 \ \text{для всех} \ p \in P_\omega \,,\ \omega \in W \,. \qquad (4.4.8)$$

Условия (4.4.7) и (4.4.8) можно объединить по отношению к ОА-паре и для всех $\omega \in W$ записать в виде

$$\sum_{p \in P_\omega} \frac{\partial D(x^*)}{\partial x_p} (x_p - x_p^*) \geq 0 \ \text{для всех} \ x_p \geq 0 \,, \qquad (4.4.9)$$

и $p \in P_\omega$, таких, что $\sum_{p \in P_\omega} x_p = r_\omega$.

Условие (4.4.9) эквивалентно требованию, чтобы

для всех $\omega \in W$ $x_p^* > 0$ только тогда, когда

$$\frac{\partial D(x^*)}{\partial x_{p'}} \geq \frac{\partial D(x^*)}{\partial x_p} \ \text{для всех} \ p' \in P_\omega \,. \qquad (4.4.10)$$

Смысл условия (4.4.10) состоит в том, что набор путевых потоков оптимален тогда и только тогда, когда путевой поток положителен только для тех путей, которые имеют минимальную первопроизводную длину. Из условия (4.4.10) также следует, что точке оптимума пути, по которым проходит ненулевая часть входного потока r_ω ОА-пары ω, должны иметь одинаковую длину (меньшую или равную длину, чем все другие пути для ω).

4.5. Методы допустимого направления для оптимальной маршрутизации

В предыдущем разделе было рассмотрено, что оптимальная маршрутизация достигается только тогда, когда поток каждой ОА-пары

направляется по путям, имеющим минимальную первопроизводную длину. Это равносильно тому, что набор путевых потоков является строго подоптимальным, если только какая-либо положительная часть потока направлена по пути, не являющемуся минимальным первопроизводным путем. При этом предполагается, что подоптимальная маршрутизация может быть улучшена переброской на минимальный первопроизводный путь потока с других путей для каждой ОА-пары. Целесообразнее перебрасывать на кратчайший путь только часть потока с других путей. В этом разделе рассматриваются методы, основанные на этой идее. В основном эти методы численно решают задачу оптимальной маршрутизации уменьшением стоимостной функции путем малых изменений в путевых потоках.

Для заданного вектора допустимых путевых потоков $x = \{x_p\}$ рассмотрим изменение x по направлению $\Delta x = \{\Delta x_p\}$. К направлению Δx предъявляются два требования.

1. Первое требование состоит в том, чтобы Δx было допустимым направлением в том смысле, что x при малом изменении вдоль Δx оставался вектором допустимых путевых потоков.

Математически это требование означает, что для некоторого $\bar{\alpha} > 0$ и всех $\alpha \in [0, \bar{\alpha}]$ вектор $x + \alpha \Delta x$ должен быть допустимым, или, что одно и тоже,

$$\sum_{p \cup P_\omega} \Delta x_p = 0 \text{ для всех } \omega \in W \tag{4.5.1}$$

$$x_p + \alpha \Delta x_p \geq 0 \text{ для всех } \alpha \in [0, \bar{\alpha}],\ p \in P_\omega,\ \omega \in W \tag{4.5.2}$$

Равенство (4.5.1) вытекает из требования допустимости

$$\sum_{p \in P_\omega} \left(x_p + \alpha \Delta x_p \right) = r_\omega$$

и того факта, что вектор x сам является допустимым; последнее означает

$$\sum_{p \in P_\omega} x_p = r_\omega \ .$$

Оно просто выражает тот факт, что сохранение допустимости необходимо, чтобы все увеличения потоков по каким-либо одним путям

компенсировались уменьшением потоков по другим путям той же ОА-пары. Можно получить допустимое направление, например взяв любой другой допустимый вектор $\bar{x}$ и положив

$$\Delta x = \bar{x} - x .$$

В действительности легко заметить, что с точностью до скалярного множителя всевозможные допустимые направления могут быть получены этим способом.

2. Второе требование состоит в том, чтобы Δx было направллением спуска, т.е. чтобы стоимостная функция убывала в направлении Δx от x.

Так как градиент $\nabla D(x)$ является нормалью к поверхностям равной стоимости стоимостной функции D, то требование о спуске переходит в условие, что скалярное произведение $\nabla D(x)$ и Δx отрицательно, т.е.

$$\sum_{\omega \in W} \sum_{p \in P_\omega} \frac{\partial D(x)}{\partial x_p} \Delta x_p < 0 . \qquad (4.5.3)$$

Это условие можно проверить математически, заметив, что первая производная функции $G(\alpha) = D(x + \alpha \Delta x)$ при $\alpha = 0$ равна скалярному произведению в (4.5.3). Заметим, что частную производную $\partial D(x)/\partial x_p$ можно выразить в виде

$$\frac{\partial D(x)}{\partial x_p} = \sum_{i,j} D'_{ij}(F_{ij}),$$

и рассматривать как первопроизводную длину пути p. Для того чтобы удовлетворить условию (4.5.3), можно потребовать, чтобы Δx удовлетворяло условию сохранения потока (4.5.1) и чтобы

$\Delta x_p \le 0$ для всех некратчайших путей p, т.е. таких путей, для которых $\partial D(x)/\partial x_p > \partial D(x)/\partial x_{\bar{p}}$ для некоторого пути $\bar{p}$, соответствующего той же ОА-паре; (4.5.4)

$\Delta x_p < 0$ для по крайней мере одного некратчайшего пути p.

Условия (4.5.1) и (4.5.4) означают, что какая-то часть потока перебрасывается с некратчайших путей на кратчайшие. Так как $\partial D(x)/\partial x_p$ принимает наименьшее значения для тех кратчайших путей $\overline{p}$, для которых $\Delta x_{\overline{p}} > 0$ и соответсвенно (4.5.1) сумма Δx_p по $p \in P_\omega$ равна нулю, то видно, что из (4.5.1) и (4.5.4) выткает условие спуска (4.5.3).

Это приводит к широкому классу итеративных алгоритмов для решения задачи оптимальной маршрутизации. Основной итерацией этих алгоритмов является

$$x = x + \alpha\Delta x,$$

где Δx - допустимое направление спуска, а величина шага α выбирается таким образом, чтобы стоимостная функция убывала, т.е.

$$D(x + \alpha\Delta x) < D(x)$$

и вектор $x + \alpha\Delta x$ быыл допустимым. Размер шага α может выбираться для каждой итерации отдельно.

Из условий оптимальности видно, что допустимые направления спуска из точки x можно найти тогда и только тогда, когда x не является оптимальным решением.

4.6. Управление потоками

4.6.1.Оконное управление потоками

В процессе передачи между передатчиком A и приемником B используется оконное управление потоком, если установлена верхняя граница на число единиц данных, которые уже были переданы передатчиком A, но о которых передатчику A неизвестно, что они попали к B (рис. 4.6.1). Верхняя граница (целое положительное число) называется размером окна или просто окном. Приемник B уведомляет передатчик A о том, что к нему попала единица данных путем отправления специального сообщения к A,

называемого разрешением (другими названиями, используемые в литературе, являются подтверждение, резервирующее сообщение, квитанция и т.д.). После получения разрешения передатчик A может отослать еще одну единицу данных к B. Таким образом, разрешение можно рассматривать как пропуск, который обязана получить единица данных, прежде чем войти в логический канал связи между A и B. Число разрешений, находящихся в использовании, не должно превышать размер окна.

Разрешения либо содержатся в специальных управляющих пакетах, либо прицепляются к обычным информационным пакетам. Их можно реализовать различными способами. Управление потоком используется при передаче по одной виртуальной цепи, группе виртуальных цепей (например, по всем виртуальным цепям, использующим один и тот же путь) или управлению подвергается весь поток пакетов, возникающих в одном узле и адресованных другому узлу.

Основная идея оконной стратегии состоит в том, чтобы интенсивность входного трафика у передатчика уменьшалась при замедлении возвращения разрешений. Следовательно, если на коммуникационном пути процесса возникают перегрузки, то сопутствующее увеличение задержки возвращения разрешений приводит к естественному замедлению интенсивности передачи данных передатчиком. Однако оконная стратегия допускает еще одну возможность, а именно приемник может умышленно задержать отправление разрешения для того, чтобы ограничить интенсивность передачи в данном процессе. Например, приемник может сделать это для того, чтобы избежать переполнения буфера.

В последующем обсуждении рассматриваются две стратегии: оконное управление от конца до конца и поузловое оконное управление. Первая стратегия относится к управлению потоком между входным и выходными узлами подсети для некоторого процесса передачи, а вторая- к управлению

потоком между каждой парой последовательных узлов вдоль пути виртуальной цепи.

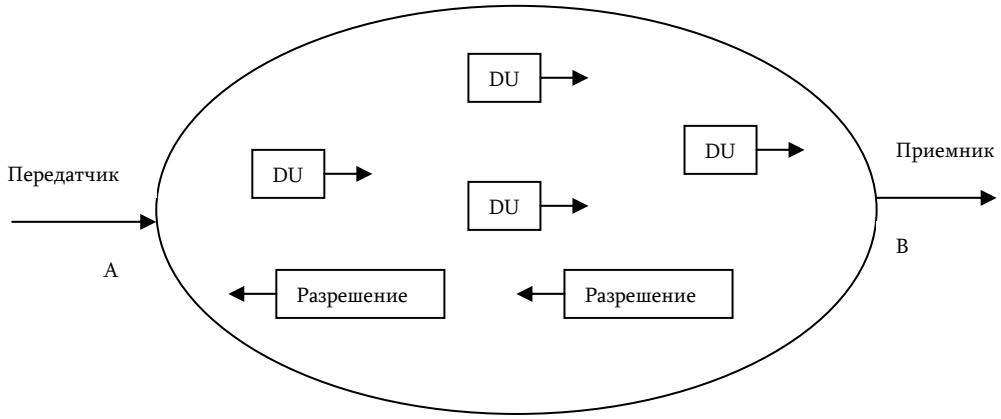


Рис. 4.6.1 Оконное управление потоком между передатчиком и приемником

4.6.2. Оконное управление от конца до конца

В самой распространенной версии управления потоком от конца до конца размер окна равен $W \cdot A$, где W и A - некоторые положительные числа. Каждый раз, когда новая партия из A единиц данных доходит до узла-адресата, назад к источнику отсылается разрешение для новых A единиц данных после получения только первой единицы из партии, состоящей из A единиц данных.

Будем считать, что $A = 1$. Обычно используется какя-либо схема нумерации пакетов и разрешений для того, чтобы можно было установить соответствие между разрешениями и ранее переданными пакетами. Одной из возможностей является использование скользящего оконного протокола, когда пакет содержит порядковый номер и следующий ожидаемый номер. Последнее число может служить в качестве одного или нескольких

разрешений для целей управления потоком. Например, предположим, что узел A получил от узла B пакет со следующим ожидаемым номером k. Тогда A знает, что до B дошли все пакеты, отосланные A с номерами, меньшими k, и поэтому узлу A разрешается отослать те пакеты, номера которых не превышают $k+W-1$ и которые он еще не отослал, где W - размер окна. В такой схеме оба числа- порядковый номер и следующий ожидаемый номер- представлены по модулю m, где $m \geq W+1$.

Для упрощения последующего изложения, предположим, что узел-источник просто считает число x пакетов, которые он передал, но для которых еще не получил назад разрешение, и передает новые пакеты только тогда, когда $x < W$.

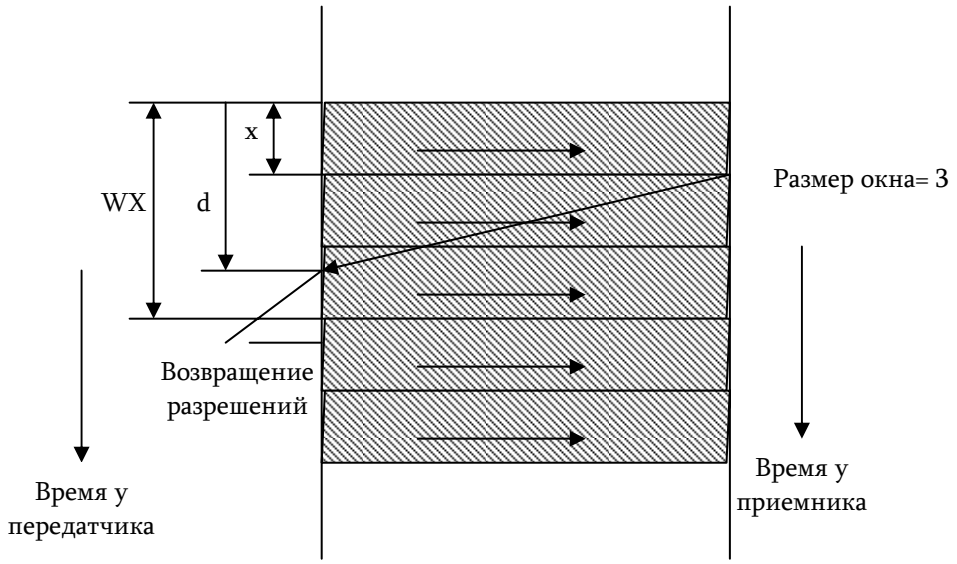


Рис 4.6.2 Пример полноскоростной передачи с размером окна $W = 3$

На рис 4.6.2 показан поток пакетов для случая, когда задержка между отправлением пакета и возвращением разрешения меньше, чем время, необходимое для передачи всего окна из W пакетов, т.е.

$$d \leq WX \text{ ,}$$

где X - время передачи одного пакета. В этом случае источник имеет возможность передавать пакеты с полной скоростью, равной $1/X$ пакетов/с, и управление потоком в данном случае оказывается неактивным.

Случай, в котором управление потоком оказывается активным, показан на рис 4.6.3 Здесь

$$d > WX$$

и задержка возвращения разрешения d оказывается настолько большой, что все W пакетов будут переданы до момента возвращения первого разрешения.

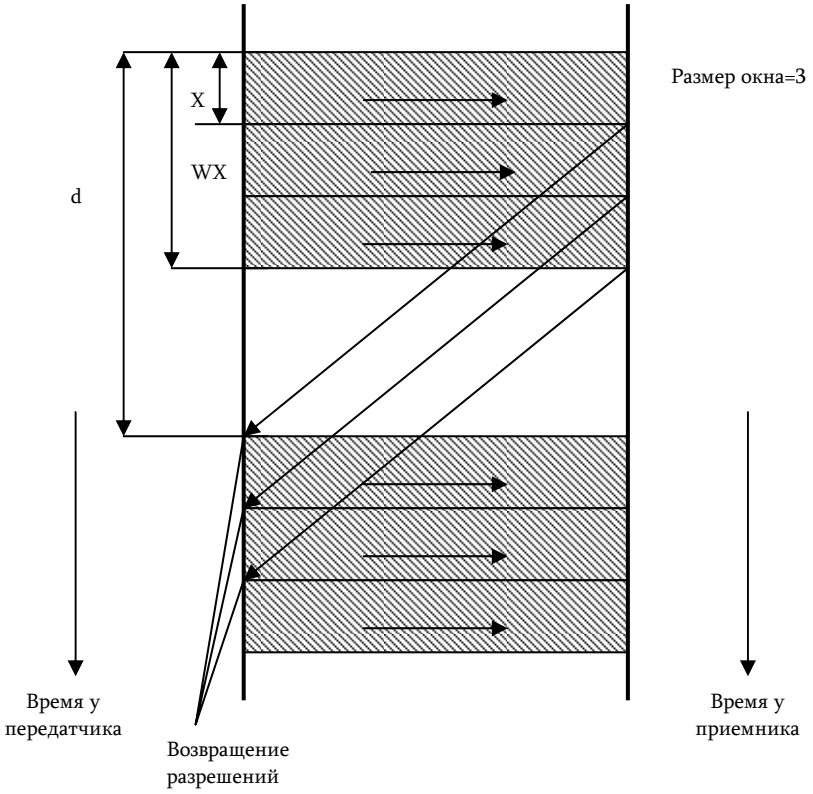


Рис 4.6.3. Пример полноскоростной передачи с размером окна $W = 3$

Предполагая, что источник всегда имеет ожидающий в очереди пакет, скорость передачи можно считать равной W/d пакетов/с. Если подытожить результаты, приведенные на рис 4.6.2 и 4.6.3, то видно, что максимальная скорость передачи, соответствующая задержке в оба конца d, равна

$$r = \min\left\{\frac{1}{X}, \frac{W}{d}\right\} \qquad (4.6.1)$$

На рис 4.6.4 иллюстрируется механизм управления потоком; скорость передачи источника уменьшается в ответ на перегрузки и присущие им большие задержки. Кроме того, оконные схемы реагируют очень быстро на перегрузки- не более чем за время передачи W пакетов. Такая быстрая реакция в сочетании с малыми дополнительными нагрузками является главным преимуществом оконных стратегий по сравнению с другими (не оконными схемами).

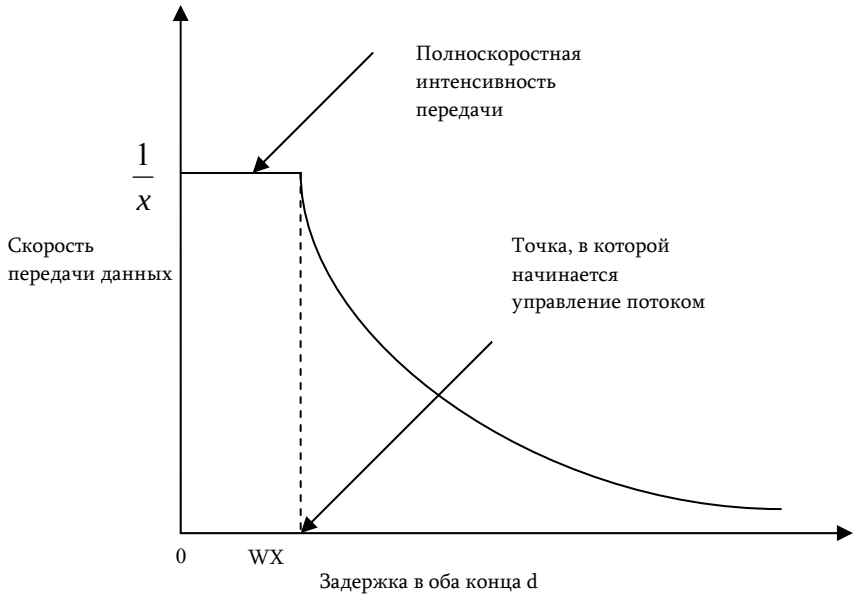


Рис. 4.6.4. Зависимость скорости передачи от величины задержки в оба конца в системе с оконным управлением потоком.

4.6.3. Недостатки оконного управления от конца до конца

Оконное управление от конца до конца имеет некоторые недостатки. Нужно отметить, что двумя главными целями управления потком являются установление разумного компромисса между малой задержкой и большой пропускной способностью и соблюдение справедливости по отношению ко

всем пользователям. Оказывается, что оконная стратегия от конца до конца (с фиксированным размером окна) не является полностью удовлетворительной ни в том ни в другом отношении.

Рассмотрим сначала задержку. Предположим, что имеется n процессов в сети, потоки которых активно управляются с размером окон $W_1,...,W_n$. Тогда суммарное число пакетов в сети равно $\sum_i^n =_1 \beta_i W_i$, где множитель β_i принимает значения между 0 и 1 в зависимости от относительной величины времени возвращения разрешения. В сответствии с теоремой Литтла средняя задержка пакета равна

$$T = \frac{\sum_{i=1}^{n} \beta_i W_i}{\lambda},$$

где λ - пропускная способность (суммарный принятый входной трафик процессов).

Рис 4.6.5. Средняя задержка пакета и пропускная способность как функции числа процессов в сети, для которых оконное управление потоком являетсяя активным.

По мере возрастания числа процессов с активно управляемым потоком пропускная способность λ начинает подходить к ограничению, которое задается пропускной способностью линий и поэтому будет приближаться к константе. Поэтому задержка T будет возрастать примерно пропорционально числу процессов с активно управляемым потоком (точнее, сумме их размеров окон), как показано на рис 4.6.5. Таким образом, если максимальное число процессов очень велико, то оконная схема от конца до конца может не удержать задержку на разумном уровне и предотвратить перегрузки. Эти трудности объясняются тем, что оконная схема ущемляет пользователей, когда задержки становятся большими, но не настолько, как это необходимо.

Можно рассмотреть использование малых размеров окон как средство борьбы с большими задержками в условиях большой нагрузки. К сожалению, существует предел, ниже которого нельзя уменьшить размеры окон без того, чтобы не ущемить пользователей в условиях малой нагрузки, в чем нет никакой необходимости. Действительно, если процесс использует путь из n линий с временем передачи пакета X на каждой линии, то время между моментом отправления пакета и возвращением разрешения будет не менее nX и оно будет значительно больше, если разрешениям не будет дан высший приоритет при их передаче по обратному каналу. Например, если разрешения на обратном пути прицепляются к пакетам, проходящим по тому же пути в обратном направлении, то время возвращения будет также менее nX . Таким образом, из рис 4.6.5 видно, что передача с полной скоростью будет невозможна для этих процессов даже в условиях малой нагрузки, если размер окна не превышает число линий n на пути. Поэтому размеры окон обычно рекомендуется выбирать между n и $3n$. Такая рекомендация предполагает, что время передачи на каждой линии намного больше времени обработки и распространения.

Что действительно необходимо в оконном управлении потоком от конца до конца для достижения хорошего соотношения между задержкой и

пропускной способностью, так это динамический выбор размеров окон. В условиях малой нагрузки окна должны быть большими и позволять вести беспрепятственную передачу, а в условиях большой нагрузки окна должны быть несколько «прикрыты» для того, чтобы задержка не становилась очень большой.

Оконное управление от конца до конца может быть также плохим по отношению справедливости. Ранее было показано, что подходящий размер окна для процесса должен быть пропорционален числу линий в его пути. Отсюда следует, что процессы с длинными путями могут иметь намного больше пакетов, ожидающих передачи по сильно перегруженной линии, чем процессы с короткими путями, в результате чего интенсивность проходящего трафика у этих процессов будет пропорционально больше.

Аспект справедливости оконного управления от конца до конца можно улучшить, если процессы, подчиненные управлению потоком и принадлежащие одному и тому же классу приоритетности, обслуживать методом кругового опроса в каждой очереди на передачу. Это, как правило, просто сделать, если процесс занимает одну виртуальную цепь. В дейтаграммной сети эффективность кругового опроса зависит от того, достаточно ли идентифицирующей информации несет каждый пакет для того, чтобы можно было в каждом узле поставить его соответствие с конкретным процессом, подчиненным управлению потоком.

4.6.4. Поузловое оконное управление для виртуальных цепей

В этой стратегии имеется отдельное окно для каждой виртуальной цепи и пары смежных узлов на пути виртуальной цепи. Многое из того, что относится к оконному управлению от конца до конца, применимо также и к этой стратегии. Так как здесь путь, вдоль которого осуществляется управление потоком, эквивалентен, по существу, одной линии, то размер

окна, измеряемый в пакетах, обычно равен двум или трем для наземных линий.

Нужно сосредоточиь внимание на паре последовательных узлов на пути виртуальной цепи, которые будем называть передатчиком и приемником. Основная идея поузловой схемы состоит в том, что приемник может избежать накопления большого числа пакетов в своей памяти путем уменьшения скорости, с которой он возвращает разрешения передатчику. В самой распространенной стратегии у приемника имеется буфер, в который можно записать W пакетов для каждой виртуальной цепи, и приемник возвращает разрешение передатчику только тогда, когда в его W-пакетном буфере имеется свободное место для записи еще одного пакета. Как только пакет покинет W-пакетный буфер, он либо будет отдан пользователю вне подсети, либо войдет в модуль управления линией передачи данных (УЛПД), ведущей к последующему узлу на пути виртуальной цепи.

Раасмотрим теперь взаимодействие окон вдоль трех последовательных узлов ($i-1$, i и $i+1$) на пути виртуальной цепи. Предположим, что W-пакетный буфер узла i полон. Тогда узел i отошлет разрешение узлу $i-1$, как только он вручит еще один пакет по модулю УЛПД на линии (i,$i+1$), а это в свою очередь произойдет, как только узел i получит разрешение, высланное узлом $i+1$. Таким образом, происходит сцепление последовательных окон вдоль пути виртуальной цепи. В частности, предположим, что на какой-то линии образовалась перегрузка. Тогда W-пакетное окно в узле, от которого идет поток по этой загруженной линии, заполнится для каждой виртуальной цепи, проходящей по этой линии.

В результате W-пакетные окна узлов, лежащих выше (по течению потока) от перегруженной линии, будут постепенно заполняться, включая окна узлов-отправителей виртуальных цепей, проходящих по перегруженной линии.

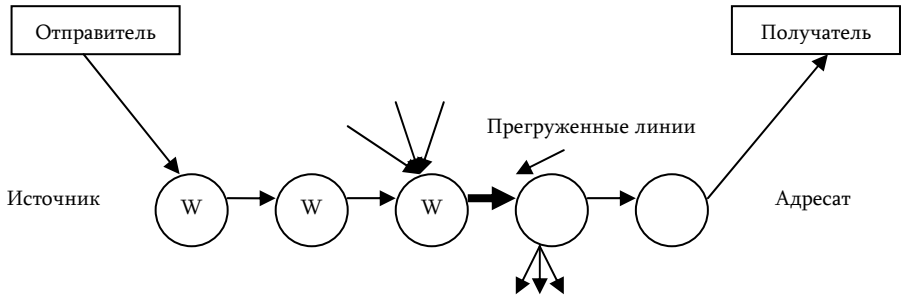


Рис. 4.6.6. Эффект обратного давления при поузловом управлении потоком.

В это время потоки этих виртуальных цепей будут активно управляться. Явление, когда окна постепенно заполняются в направлении от точки перегрузки до узлов-отправителей виртуальных цепей называется обратным давлением (рис. 4.6.6).

Одну привлекательную особенность поузлового оконного управления можно увидеть на рис 4.6.6. В худшем случае, когда перегрузка образовалась на последней линии (скажем, на n-ой) пути виртуальной цепи, суммарное число пакетов внутри сети для этой виртуальной цепи будет приближенно равно nW. Если бы оконное управление потоком этой виртуальной цепи было от конца до конца, то суммарное число пакетов внутри сети было бы примерно таким же. Важное отличие, однако, состоит в том, что в случае поузлового управления эти пакеты будут равномерно распределены вдоль пути виртуальнойй цепи, а в случае управления от конца до конца они будут сосредоточены у перегруженной цепи. Вследствие этого объем памяти, который необходим в каждом узле для избежания переполнения буфера, при поузловом оконном управлении может быть намного меньше, чем в случае оконного управления от коца до конца.

Равномерное распределение пакетов по виртуальной уепи вдоль ее пути ослабляет проблему нарушения справедливости, которое проявляется тогда, когда процессы с большими окнами монополизируют перегруженную линию за счет процессов с малыми окнами.

4.6.5. Изаритмический метод

Изаритмический метод можно рассматривать как разновидность оконного управления потоком, в котором имеется только одно всеобщее окно для всей сети. Идея здесь состоит в том, что для ограничения суммарного числа пакетов в сети нужно иметь фиксированное число разрешений, циркулирующих по сети. Пакет входит в сеть после того, как он захватит одно из этих разрешений. Попав в свой узел-адресат, он затем отпускает разрешение. Таким образом, суммарное число пакетов в сети ограничено числом разрешений. Можно установить верхнюю границу средней задержки пакета, которая не будет зависеть от числа сеансов в сети. К сожалению, вопросы справедливости и перегрузки внутри сети зависят от того, как распределены по сети разрешения, которые при изаритмическом подходе не имеют адресов. Существует также и другая трудность, связанная с тем, что разрешения могут исчезать вследствие различных аппаратурныхх неисправностей.

4.6.6. Оконное управление потоком на уровне пользователя

Многое из того, что было сказано до сих пор об оконных стратегиях, применимо к управлению потоком в сеансе между двумя пользователями либо на сетевом уровне, либо на транспортном уровне.

На рис 4.6.7 показана типичная ситуация. Данные из машины A отсылаются пользователем входному узлу NA подсети, а затем они направляются выходному узлу NB и машине B. Имеется управление потоком (сетевого уровня) между входным и выходным узлами NA и NB. Имеется также оконное управление потоком (сетевого уровня) между машиной A и входным узлом NA, которое удерживает машину A от

передачи узлу NA большего количества данных, чем то, с которым он может справиться. Точно также имеется оконное управление потоком между выход-

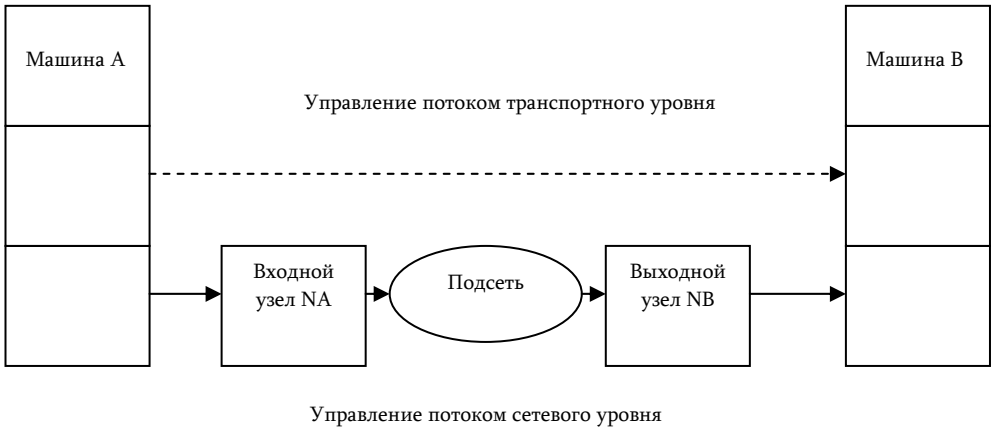


Рис.4.6.7. Управление потоком от пользователя до пользователя.

ным узлом NB и машиной B, которое удерживает NB от передачи машине B слишком большого количества данных. Таким образом, видно, что имеется система управления потоком сетевого уровня, простирающаяся от машины A до машины B, которая работает во многих отношениях так же, как система поузлового оконного управления потоком.

4.6.7. Схемы управления потоком, основанные на регулировании интенсивности входного трафика

Было показано, что одна из основных трудностей, с которыми приходится сталкиваться в процессе оконного управления потоком с фиксированными размерами окон, состоит в том, что средняя задержка пакета возрастает пропорционально числу процессов, активно управляемых по потоку. Средством борьбы с этим явлением может быть уменьшение размеров окон по мере увеличения числа таких потоков. К сожалению, не очень просто

найти хорошие пути осуществления этого на практике. Поэтому стоит сосредоточить внимание на более четких формулировках задачи управления потоком, в которых интенсивность входного трафика у процессов, потоки которых управляются, регулируются непосредственно в ответ на состояние трафика внутри сети.

4.6.8. Сочетание оптимальной маршрутизации и управления потоком

Рассмотрим возможность сочетания маршрутизации и управления потоком от конца до конца внутри подсети путем оптимального регулирования как маршрутных переменных, так и входных интенсивностей для пар отправитель-адресат (ОА-пар). Частным случаем является задача чистого управления потоком, в которой маршрутизация фиксируется, и единственными переменными, которые должны регулироваться, остаются входные интенсивности.

Обозначим через r_ω входную интенсивность ОА-пары ω. В некоторых случаях r_ω может измерятся в бит/с, и тогда ее оптимальное значение можно интерпритировать как требуеммое значение интенсивности, усредненное по достаточно длинному интервалу времени (например, передать не более $r_\omega T$ бит в течение временного интервала длины T). Если r_ω измеряется числом виртуальных цепей, то ее оптимальное значение можно интерпритировать как требуемое значение, которое узел-отправитель старается достигнуть путем блокировки или разрешения новых запросов на соединение, возникших во внешних пунктах.

Далее мы сначала сосредоточим внимание на формулировке задачи минимизации некоторой разумной стоимостной функции путем подгонки маршрутных переменных и входных интенсивностей r_ω. Затем покажем, что

эта задача математически эквивалентна задаче оптимальной маршрутизации (r_w фиксированы).

Если минимизировать стоимостную функцию $\sum_{(i,j)} D_{ij}(F_{ij})$ задачи маршрутизации по путевым потокам $\{x_p\}$ и входным интенсивностям $\{r_\omega\}$, то можно обнаружить, что оптимальное решение будет разочаровывающим: $x_p = 0$ и $r_\omega = 0$ для всех p и w. Это указывает на то, что стоимостная функция должна включать штраф за то, что входные интенсивности становятся слишком малыми; это приводит к задаче

$$\text{Минимизировать } \sum_{(i,j)} D_{ij}(F_{ij}) + \sum_{\omega \in W} e_\omega(r_\omega)$$

$$\text{при ограничениях } \sum_{p \in P_\omega} x_p = r_\omega \text{ для всех } \omega \in W, \qquad (4.6.2)$$

$$x_p \geq 0 \quad p \in P_\omega, \text{ для всех } \omega \in W.$$

Здесь минимизация должна проводиться по $\{x_p\}$ и $\{r_\omega\}$. Данные величины $\bar{r}_\omega$ означают желаемые входные интенсивности ОА-пары ω, т.е предлагаемую нагрузку для ω, определяемую как входные интенсивности для ω, которые будут при отсутствии управления потоками. F_{ij} означает суммарный поток по линии (i, j), т.е. сумму всех путевых потоков, проходящих по этой линии. Функции e_ω имеют такой вид, как показано на рис 4.6.8, и означают штраф за ущемление входной интенсивности r_ω. Они являются выпуклыми, монотонно убывающими функциями на множестве положительных чисел $(0, \infty)$ и стремятся к ∞, когда r_ω стремится к нулю. Мы предполагаем, что их первые и вторые производные e'_ω и e''_ω существуют на $(0, \infty)$ и строго отрицательны и положительны соответственно. Интересный класс функций e_ω задается следующей формулой для их первой производной:

$e'_\omega(r_\omega) = -\left(\frac{a_\omega}{r_\omega}\right)^{b_\omega}$, где a_ω и b_ω являются заданными положительными константами (4.6.3)

Параметры a_ω и b_ω влияют на оптимальные значения входных интенсивностей r_ω и на приоритет ОА-пары ω соответственно.

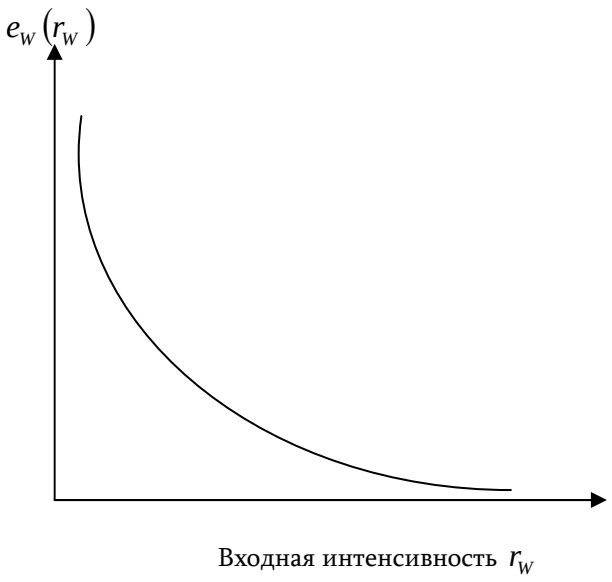


Рис. 4.6.8. Типичный вид функции штрафа за ущемление входной интенсивности r_W

Ценность предыдущей формулировки повысится, если мы расширим наше представление о ω и будем рассматривать ее как класс пользователей, использующих один и тот же набор путей P_ω. Это позволяет придавать разные приоритеты (т.е. разные функции e_ω) разным классам пользователей, даже если они вместе используют одни и те же пути.

Теперь покажем, что объединенная задача маршрутизации и управления потоком (4.6.2) математически эквивалентна задаче маршрутизации, которая рассматривалась в предыдущем параграфе. Введем новую переменную y_ω для каждого $\omega \in W$ посредством равенства

$$y_\omega = \bar{r}_\omega - r_\omega \qquad (4.6.4)$$

Можно рассматривать y_ω как избыточный поток (часть $\bar{r}_\omega$, которая блокируется сетью) и считать, что этот поток проходит по избыточной линии,

непосредственно соединяющей узел-отправитель с узлом-адресатом пары ω, как показано на рис 4.6.9. Если определить новую функцию F_ω равенством

$$E_\omega(y_\omega) = e_\omega(\bar{r}_\omega - y_\omega), \tag{4.6.5}$$

то задачу (4.6.2) с учетом (4.6.4) можно переписать в виде

Минимизировать $\sum_{(i,j)} D_{ij}(F_{ij}) + \sum_{\omega \in W} E_\omega(y_\omega)$ (4.6.6)

при ограничениях $\sum_{p \in P_\omega} x_p + y_\omega = \bar{r}_\omega$ для всех $\omega \in W$,

$x_p \geq 0$ для всех $p \in P_\omega$, $\omega \in W$,

$y_\omega \geq 0$ для всех $\omega \in W$.

Вид функции E_ω из (4.6.5) показан на рис 4.6.9. Так как $e_\omega(r_\omega) \to \infty$ при $r_\omega \to 0$ (т.е. назначается бесконечный штраф за полное запирание класса пользователей ω), то $E_\omega(y_\omega) \to \infty$, когда избыточный поток y_ω приближается к своему максимальному значению- максимальной входной интенсивности $\bar{r}_\omega$. E_ω можно рассматривать как задержку для избыточной линии, а $\bar{r}_\omega$ можно рассматривать как пропускную способность этой линии.

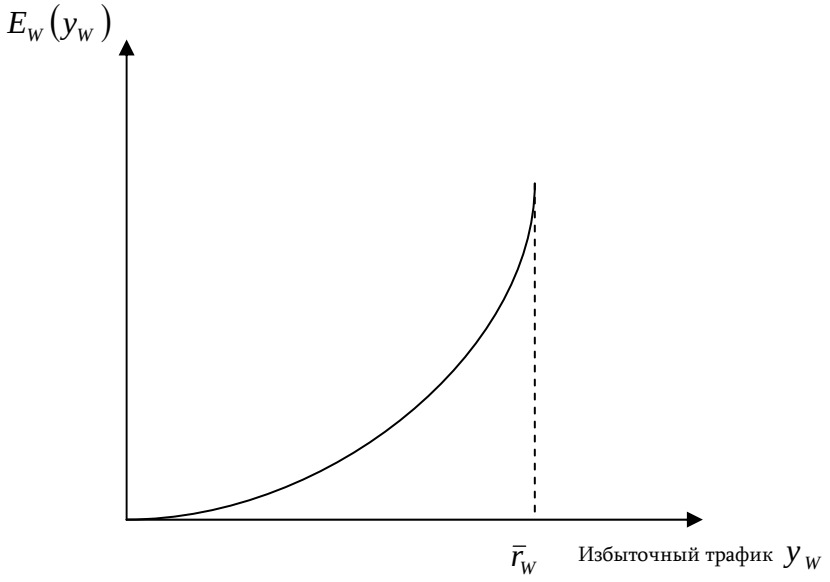


Рис. 4.6.9. Стоимостная функция для избыточной линии.

Теперь стало ясно, что задача (4.6.6) является задачей того же типа, что рассматривалась в предыдущем параграфе. В частности, применение условия оптимальности, основанного на понятии кратчайшего пути, приводит к следующему результату:

множество допустимых значений путевых потоков $\{x_p^*\}$
и входных интенсивностей $\{r_\omega^*\}$ является оптимальным
для задачи (6.2) тогда и только тогда, когда выполняются
следующие условия для всех $p \in P_\omega$ и $\omega \in W$:

$x_p^* > 0$ только тогда, когда $d_p^* \le d_{p'}^*$ для всех $p' \in P_\omega$,

$$d_p^* \le -e'_\omega(r_\omega^*), \tag{4.6.7а}$$

$r_\omega^* < \bar{r}_\omega$ только тогда, когда $-e'_\omega(r_\omega^*) \le d_p^*$ для всех $p \in P_\omega$, (4.6.7б)

где d_p^* - первопроизводная длина пути $p\left[d_p^* = \sum_{(i,j)} D'_{ij}(F_{ij}^*)\right]$
и F_{ij}^* - суммарный поток по линии (i,j), соответствующий $\{x_p\}$.

Заметим, что условие оптимальности (4.6.7) зависит только от производных функций D_{ij} и e_ω. Это означает, что добавление произвольных констант к D_{ij} или e_ω не влияет на оптимальное решение. Заметим также, что из (6.7б) следует, что оптимальная точка r_ω^* на зависит от $\bar{r}_\omega$, пока $\bar{r}_\omega > r_\omega^*$. Это является хорошим свойством стратегии управления потоком, мешающим пользователям (чей поток управляется активно) увеличить их долю ресурса путем повышения требований.

Смысл параметров a_ω и b_ω в стоимостной функции, определяемых формулой

$$e'_\omega(r_\omega) = -\left(\frac{a_\omega}{r_\omega}\right)^{b_\omega},$$

теперь может быть понят в свете условий оптимальности (4.6.7б). Рассмотрим два различных класса пользователей ω_1 и ω_2, использующих одни и те же пути $\left(P_{\omega_1} = P_{\omega_2}\right)$. Тогда из условий (4.6.7б) следует, что в точке оптимального

решения, в которой оба класса пользователей ущемлены $\left(r^*_{\omega_1} < \bar{r}_{\omega_1}, r^*_{\omega_2} < \bar{r}_{\omega_2}\right)$, должны выполняться равенства

$$-e'_{\omega_1}\left(r^*_{\omega_1}\right) = -e'_{\omega_2}\left(r^*_{\omega_2}\right) = \min_{p\in P_{\omega_1}}\left\{d^*_p\right\} = \min_{p\in P_{\omega_2}}\left\{d^*_p\right\} \qquad (4.6.8)$$

Если e'_{ω_1} и e'_{ω_2} определяются параметрами a_{ω_1}, b_{ω_1} и a_{ω_2}, b_{ω_2} согласно (4.6.3), то можно заметить следующее:

(а) Если $b_{\omega_1} = b_{\omega_2}$, то

$$\frac{r^*_{\omega_1}}{r^*_{\omega_2}} = \frac{a_{\omega_1}}{a_{\omega_2}},$$

и отсюда следует, что параметр a_ω влияет на оптимальную относительную входную интенсивность класса пользователей ω.

(б) Если $a_{\omega_1} = a_{\omega_2} = a$ и $b_{\omega_1} > b_{\omega_2}$, то условие (4.6.8) показывает, что, когда входные потоки вынуждены быть малыми $\left(r^*_{\omega_1}, r^*_{\omega_2} < a\right)$, классу пользователей ω_2 (тому, у которого параметр b_ω больше) позволено допустить в сеть больше трафика. Отсюда следует, что параметр b_ω влияет на относительный приоритет класса пользователей ω в условиях большой нагрузки.[54]

Заключение

Современные задачи использования компьютерных сетей ставят высокие требования к качеству обслуживания сетей - QoS (Quality of Service). С целью повышения качества обслуживания может быть проведен ряд мероприятий. Одним из важнейших является управление ресурсами. Ресурсами можно расссматривать определенный аппаратный и программный комплекс компьютерных сетей. Но с точки зрения QoS ресурсами считаются задержка и пропускная способность.

Задачей диссертационной работы являлось построение моделей управления, которые дали бы возможность оценить задержку – один из параметров сети и в тоже время один из ресурсов; определить оптимальность маршрутизации и ее связь с управлением потоками, решить задачу управления потоками в сочетании с оптимальной маршрутизацией.

С указанной целью были разработаны три модели массового обслуживания. Была разработана система массового обслуживания M/M/m с m обслуживающими приборами, M/M/∞ с бесконечным числом обслуживающих приборов и M/M/m/m с m-потерями и m-обслуживающими приборами. Предложен подход к определению оптимальности маршрутизации и разработан метод управления потоками.

По диссертационной работе можно сделать следующие выводы:

1. Предложенные модели дают возможность наиболее точно оценить величину задержки в сети и величину отказа работы в сети.

2. Первая модель дает возможность оценить значение задержки и определить среднее число требований в системе для сети, в которой m обслуживающих приборов с пуассоновским входным потоком, длительности обслуживания независимы от интервалов между моментами поступления запросов на обслуживание и экпоненциально распределены.

3. Предложена также модель для определения задержки и среднего числа требований для системы с беконечным числом обслуживающих приборов.

4. Разработана модель с m отказами, поскольку сети при управлении потоками часто приходится отказывать определенным запросам на обслуживание. В результате получено выражение для определения вероятности потери запроса на обслуживание, если все m обслуживающих приборов при его поступлении будут заняты.

Для получения наилучших значений параметров задержки и пропускной способности используют маршрутизацию и управление потоками. Маршрутизация является одной из сложнейших функций сети передачи данных, требующей согласованной работы узлов сети. Она влияет на среднюю задержку пакета и пропускную способность сети. В зависимости от способа реализации маршрутизация может привести к низкой пропускной способности, чувствительности к перекрузкам и к колебательному режиму. Эти недостатки более заметны в дейтаграммных сетях и менее заметны в сетях с виртуальными цепями.

Более сложным вопросом является определение оптимальной маршрутизации.

1. Приведенная в диссертационной работе методика определения оптимальности маршрутизации основана на потоковых моделях.

2. Для определения оптимальности вводится стоимостная функция, которая, которая зависит от пропускной способности, задержки и потока.

3. Решение задачи является в минимизации стоимостной функции при ограничениях на интенсивность трафика. Такая задача становится необходимой при увеличении интенсивности входного трафика.

В диссертациооной работе были определены главные цели управления потоками, такие как поддержание внутри сети относительно малой средней задержки и справедливости по отношению ко всем пользователям. Был

сделан обзор основных методов управления потоком. Можно отметить, что на практике преобладают стратегии, основанные на окнах. Для этого имеются основательные причины, так как оконные стратегии сочетают низкую избыточность в нагрузке и быструю реакцию при перегрузке. Однако оконные стратегии имеют также серьезные недостатки, которые состоят в том, что средняя задержка пакета возрастает пропорционально числу процессов, активно управляемых по потоку.

1. В данной работе была предложена схема регулирования входной интенсивности, которая пытается избавиться от недостатков характерных для управлления потоками методом окон.

2. Разработана схема, которая обобщает методы оптимальной маршрутизации и управления потоками.

Список используемой литературы

1. Шринивас Вегешна. Качество обслуживания в сетях .-Москва, 2003 г., стр 604

2. Олифер В.Г., Олифер Н.А. «Компьютерные сети». Санкт-Питербург: Питер, 2002 г., стр 668

3. «Модели информационных сетей и коммутационных систем», Издательство «Наука», Москва, 1982 г., стр. 375

4. Г.Янбых, Б. Столяров, «Оптимизация информационно-вычислительных сетей», Москва «Радио и связь», 1987 г., стр. 103

5. Д.Девис, Д.Барбер, «Вычислительные сети и сетевые протоколы», Москва «Мир», 1982 г., стр. 678

6. В.Морозов, А.Долганов, «Основы теории информационных сетей», Москва «Высшая школа», 1987 г., стр. 98

7. Р.Бесслер, А.Дойч, «Проектирование сетей связи», Москва «Радио и связь», 1988 г., стр. 218

8. Ephremides A., «The Routing Problem in Computer Networks», New-York, 1996, pp 299-324

9. Hayes J.F., «Modeling and Analysis of Computer Communications Networks», New-York, Plenum, 1984, p. 397

10. Abeysundara B.W., Kamal A.E., «High-Speed Local Area Networks and Their Perfomance», Computing Surveys, vol 23, pp 221-264, June 1991.

11. Adam J.A., «Privacy and Computers», IEEE Spectrum, vol 32, pp 46-52, Dec. 1995.

12. Baransel C., Dobosiewicz W., Gblrzynski P., «Routing in Multihop Packet Switching Networks», IEEE Network Magazine, vol 9, pp 38-61, May/June, 1995

13. Блэк Ю.Д., « TCP/IP и родственные протоколы», Нью-Йорк, «MacGraw-Hill», 1995 г., стр. 207

14. Зангвилл У.И., «Нелинейное програмирование. Единый подход», Москва, Сов. Радио, 1973 г., стр. 111

15. Lam S.S., Reiser M., «Congestion control of store-and-forward networks by unit buffer limits», IBM Research Report, 1987, p 235

16. Э. Таненбаум, «Компьютерные сети», Питер, 2002 г., стр. 993

17. Э. Таненбаум, «Распределенные операционные системы», Питер, 2002 г., стр. 895

18. Э. Таненбаум, «Современные операционные системы», Питер, 2002 г., стр. 956

19. Huitema C., «The New Internet Protocol»Prentice Hall, 1996, p 280

20. Huitema C., «Routing In the Internet»Prentice Hall, 1995, p 481

21. Комер Д., «Принципы функционирования Интернета», Питер, 2001 гю, стр. 169

22. Day J.D., Zimmermann H., «The OSI Reference Model», Computer Commun. Rev., vol. 71, Dec. 1983

23. Питерсон В.В., Браун Д.Т. «Циклические коды для предотвращения ошибок», Proc. IRE, 1991 г., стр. 178

24. Day J.D., «The (Un)Revised OSI Reference Model», Computer Commun. Rev., vol. 25, Oct. 1995

25. Deering S.E., «Simple Internet Protocol», IEEE Network Magazine, vol. 7, May/June 1993

26. Deering S.E., Cheriton D.R., «Multicast Routing in Datagram Internetworks and Extended LANs», ACM Trans. On Computer Systems, vol. 8, May 1990

27. Dorfman R., «Detection of Defective Members of a Large Population», Annals Math. Statistics, 1943, p 201

28. Ford P.S., Rekhter Y., Braun H., «Improving the Routing and Addressing of IP», IEEE Network Magazine, vol. 7, May/June 1993

29. IEEE: Communication magazine, vol. 33, Jan. 1995

30. IEEE: 802.3: Carrier Sense Multiple Access with Collision Detection, New York: IEEE, 1985a

31. IEEE: 802.4: Token-Passing Bus Access Method, New York: IEEE, 1985b

32. IEEE: 502.5: Token Ring Access Method, New York: IEEE, 1985c

33. Kahn D., «The Codebreakers», New York: Macmillan, 1996, p 305

34. Shacham N., Mckenney P., «Packet Recovery in High-Speed Networks Using Coding and Buffer Management», Proc/ INFOCOM'90, IEEE, 1990

35. Kavak N., «Data Communication in ATM Networks», IEEE Network Magazine, vol. 9, May/June 1995

36. Mcdysan D.E., Spochn D.L., «ATM- Theory and Application», NY, McGrawHill, 1995, p 238

37. Piscitello D.M., Chapin A.L., «Open Systems Networking: TCP/IP and OSI», Addison-Wesley, 1993, p 385

38. First Int'l. Worldwide Web Conference, 1994

39. Santifaller M., «TCP/IP and ONC/NFS», Addison-Wesley, 1994, p 239

40. Schneier B., «E-Mail Security», New York, John Wiley, 1995, p 93

41. Smith P., «Frame Relay and X.25», Addison-Wessley, 1993, p 190

42. Stevens W.R., «TCP/IP Illustrated», Addison-Wesley, 1994, p 205

43. Teraoka F., Yokte Y., Tokoro M., «Host Migration Transparency in IP Networks», Computer Commun. Rev., vol. 23, Jan. 1993

44. Gallager R.G., «A Minimum Delay Routing Algorithm Using Distributed Computation», IEEE Trans Commun., 1977

45. Кларк Д.Д., Погрен К.Т., Рид Д.П., «Локальные сети», ТИИЭР, 1987 гю, стр. 372

46. Кларк Д.Д., Кейн Д., «Кодирование с исправлением ошибок в системах цифровой связи», Москва, Радио и связь, 1987 г., стр. 407

47. Галлагер Р.Д., «Теория информации и надежная связь», Москва, Радио и связь, 1974 г., стр. 544

48. Форд Л.Р.. Фалкерсон Д.Р., «Потоки в сетях», Москва, Мир, 1986 г., стр. 498

49. Грей Д.П., «Линейное управление», ТИИЭР, т. 60, № 11, 1987

50. Клейнрок Л., «Коммуникационные сети. Стохастические потоки и задержки сообщений», Москва, Наука, 1980 г., стр. 627

51. Клейнрок Л., «Вычислительные системы с очередями», Москва, Мир, 1986 г., стр. 372

52. Клемени Д.Д., Снелл Д., Кнеп А., «Счетные цепи Маркова», Москва, Мир, 1987 г., стр. 749

53. Михайлов В.А., Цыбаков В.С., «Верхняя граница для пропускной способности системы случайного множественного доступа», Пробл. Передачи информ., 1981, т. 17, № 1

54. Полак Э., «Численные методы оптимизации», Москва, Мир, 1987 г., стр. 896

55. Куреши Ш., «Адаптивная коррекция», ТИИЭР, т. 7, № 9, 1985

56. Walrand J., Interconnections of Markov Chains and Quasi-reversible ueueing Networks, Stoc Proc Appl, 10, pp 209-219, 1980.