

СЕВЕРО-КАВКАЗСКИЙ ФИЛИАЛ
ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«Московский технический университет связи и информатики»
Кафедра «Инфокоммуникационные технологии и системы связи»

НЕРСЕСЯНЦ А.А.

**Методические указания и задание
к практическим занятиям**

**Анализ распределённой системы
клиент-сервер**

**Дисциплина:
Теория телетрафика**

Направление подготовки 11.03.02
«Инфокоммуникационные технологии и системы связи»,
профиль «Сети связи и системы коммутации»

Ростов-на-Дону
2019

Методические указания к практическим занятиям по теме:

Анализ распределённой системы клиент-сервер

Дисциплина: Теория телетрафика

Направление подготовки 11.03.02

«Инфокоммуникационные технологии и системы связи»,
профиль «Сети связи и системы коммутации»

Практическое занятие предназначено для изучения принципов анализа телекоммуникационных систем с очередями на примере организации информационного обмена в локальной вычислительной сети между клиентами и серверами. Рассматриваются варианты с одним или группой серверов, а также с общей или индивидуальной очередями к серверам.

Данное пособие может быть использовано при разработке дипломных проектов по специальности «Сети связи и системы коммутации» при расчете времени реакции информационных систем в локальных, городских и глобальных сетях.

Автор:

профессор кафедры ИТСС, д.т.н., с.н.с. Нерсесянц А.А.,

Рецензент: доцент кафедры ИТСС - к.т.н. Борисов Б.П.

Рассмотрено и одобрено на заседании кафедры ИТСС

Протокол №11 от 26.08.2019

Анализ распределённой системы клиент-сервер

1. Цель практического занятия. Овладеть основами расчёта клиент-серверных систем, рассматриваемых как системы с очередями, с определением основных вероятностно-временных характеристик систем.

2. Постановка задачи в общем виде

В условиях стремительного роста интенсивности информационного обмена в современных сетях часто возникает необходимость в применении научно обоснованных методов предсказания последствий изменений в сети, смены топологии сети и т. д. Последствия могут оцениваться с точки зрения влияния на производительность, время ответа сети, доступность тех или иных сервисов и пр. Желательно также уметь проводить априорную оценку параметров сети до ее развертывания. Представим себе следующую ситуацию. В организации установлено определенное количество рабочих станций, подключенных к сети Ethernet 10 Мбит/с. Руководитель недавно сформированного отделения организации собирается подключить новые рабочие станции своих сотрудников к этой действующей сети. Перед всей организацией сразу встает вопрос — сможет ли существующая локальная сеть справиться с возросшей нагрузкой или для этого отделения необходимо будет создавать вторую локальную сеть и объединять обе сети мостом или, наконец, повысить скорость работы локальной сети до 100 Мбит/с, т.е. перейти к сети fastEthernet?

Существуют и другие случаи, в которых достаточно сложно быстро получить ответ на вопрос о том, насколько возрастет нагрузка на сеть при тех или иных изменениях, и справится ли с ней сеть. С точки зрения проектирования сети это означает, что не существует четкого однозначного метода, позволяющего на основе существующих требований к сети вычислить параметры и конфигурацию будущей системы. Рассмотрим другой пример. Одно из отделений организации намеревается оборудовать все свои рабочие места персональными компьютерами и настроить их для работы в локальной сети с сервером. Конечно, можно, основываясь на опыте какой-либо родственной организации, которая уже проделала подобную работу, оценить примерную загрузку, генерируемую каждым персональным компьютером, и на этом основании оценить требуемую производительность всей локальной сети и сервера в частности. Естественно, основным критерием при оценке совокупности параметров сети в данном случае является ее производительность в целом. При использовании интерактивных приложений реального времени в качестве основной оценочной характеристики обычно используется время ответа сети (иногда оно называется временем реакции сети). В других случаях ориентируются на пропускную способность сети.

Очевидно, что в таких случаях при проектировании сети необходимо иметь аналитические инструменты, позволяющие предсказывать производительность по

модели сети. Одним из таких инструментов, предназначенных для разработки сетевых и коммуникационных структур, может быть аналитическая модель, основанная на теории очередей. Оказывается, большое количество проблемных вопросов находят свое решение при использовании математического метода анализа очередей.

Хотя теория очередей математически достаточно сложна, применение этой теории для анализа производительности сети во многих случаях дает желаемые результаты. Знание основ статистики и понимание базовых принципов применения теории очередей — это все, что может потребоваться для получения оценки производительности сети с необходимой степенью точности. Аналитик может провести анализ очередей в заданной сетевой структуре, используя уже готовые таблицы очередей или простые компьютерные программы, которые занимают несколько строк кода.

Перед рассмотрением теории очередей, представляемой далее в виде, удобном для практического использования, можно привести пример конкретного использования этой теории. Рассмотрим Web-сервер, который тратит на обработку одного запроса какое-то заранее известное, фиксированное время — допустим одну миллисекунду (очевидно, что это будет также средним временем, затрачиваемым на обработку). Теперь, если средняя скорость поступления запросов равна одному запросу в одну миллисекунду (1000 запросов в секунду), то вполне можно считать, что сервер справится с этой нагрузкой. Действительно, это произойдет в том случае, когда запросы поступают с одинаковой скоростью (равной, очевидно, одному запросу в каждую миллисекунду). После поступления запроса сервер немедленно обрабатывает его. После того как сервер обработал текущий запрос, поступает новый запрос, сервер начинает его обработку и снова укладывается во время.

Рассмотрим более реальную ситуацию и предположим, что средняя скорость поступления запросов по-прежнему равна одному запросу в миллисекунду, но существует некоторая флуктуация поступления запросов. Тогда в течение любого миллисекундного периода времени может не приходиться запросов вообще, а может поступить сразу несколько запросов. Но средняя скорость поступления запросов все равно равна одному запросу в миллисекунду и является достаточно четким критерием того, насколько загружен сервер. С флуктуациями, нерегулярностями в поступлении запросов можно справиться, введя буферную память. Действительно, в течение времени занятости, когда необходимо обработать множество запросов, сервер может сохранять невыполненные запросы в своей буферной памяти. Или, иными словами, сервер помещает эти запросы в *очередь*. Когда сервер завершит обработку текущего запроса, он возьмет запрос из очереди и тем самым уменьшит ее. С этой точки зрения, основным вопросом при разработке сетевой структуры является вопрос о том, насколько большим должен быть буфер сервера.

Довольно часто возникает необходимость в проведении оценки производительности на основе имеющихся данных о загрузке действующей сети или по предполагаемой загрузке вновь проектируемой сети. Для проведения таких оценок существуют различные подходы:

- Проведение анализа производительности сети после ее внедрения, основываясь на значениях показателей, которые актуальны в данном конкретном случае;

- Выполнение простой оценки работоспособности будущей среды, основанной на существующем опыте разработки и построении подобных сетей;

- Разработка и применение аналитической модели, основанной на теории очередей;

- Разработка и применение простейшей программы, моделирующей поведение сети.

Первый вариант предполагает пассивную позицию разработчика сети. Разработчик просто ожидает результатов своей деятельности. Естественно, такой метод чреват непредсказуемыми последствиями. Полученным результатом, как правило, оказываются недовольны и пользователи, и руководители организации. Их можно понять — они понесли неоправданные затраты, но, в итоге, так и не получили сети с желаемыми параметрами.

Второй вариант может дать, как правило, лучшие результаты. При анализе будущей сети на основании имеющегося опыта можно увидеть, что при наличных возможностях (в том числе, финансовых) и ограничениях бессмысленно ожидать, что сеть будет удовлетворять тем или иным требованиям. То есть, этот метод позволяет достаточно уверенно предсказать, что *не сможет* делать проектируемая сеть. С точки зрения выполнения предъявляемых требований, метод, основывающийся на опыте, может дать только достаточно расплывчатые предложения, носящие качественный характер. Абсолютно бессмысленно пытаться выполнять на основе этого метода некую более или менее точную количественную оценку необходимых параметров. Другая проблема, связанная с этим подходом, заключается в том, что поведение большинства систем при изменении загрузки будет не таким, как интуитивно ожидалось. Если существует среда, в которой есть разделяемые каналы связи, то производительность такой системы, как правило, экспоненциально уменьшается при увеличении нагрузки. В результате наблюдается расхождение ожидаемых значений и наблюдаемых (рис. 1).

Загрузка сети в данном случае определяется долей задействованной пропускной способности. Следовательно, если рассматривать мост, который способен обрабатывать 1000 кадров в секунду, то загрузка 0.5 означает скорость передачи 500 кадров в секунду. Время ответа есть сумма средних времен, затрачиваемых на передачу входящих в сеть кадров.

На рис. 1 верхняя кривая показывает изменение реального времени ответа сети на разделяемых каналах связи при увеличении нагрузки. Нижняя кривая описывает ожидаемые разработчиком значения. Две кривые совпадают только в пределах той нагрузки, с которой реально имел дело наш гипотетический разработчик. Как видим, опыт является достаточно надежным проводником только при половинной загрузке сети. При дальнейшем росте нагрузки производительность сети будет резко снижаться.

Для проведения оценки поведения системы практически на всем диапазоне загрузки может быть использован аналитический метод. При его практическом

применении приходится решать набор уравнений, после чего удастся получить параметры, необходимые для оценки системы (время ответа, пропускную способность и т. д.). Использование теории очередей дает достаточно точную оценку, которая, в большинстве случаев, хорошо соответствует действительности. Недостатком теории очередей является то, что при выводе формул, на которых она основывается и которые используются для расчета интересующих нас параметров, необходимо принять определенные допущения. Тем не менее, оказывается, что эти допущения вполне оправданны, а получающиеся результаты близки к тем, которые получаются при программном моделировании сети с такими же параметрами. Преимуществом теории очередей по сравнению с моделированием является то, что анализ очередей может быть выполнен за сравнительно короткий срок (для большинства реальных ситуаций), в то время как моделирование может занять дни или даже недели — создать программную модель, описывающую требуемую ситуацию, достаточно непросто.

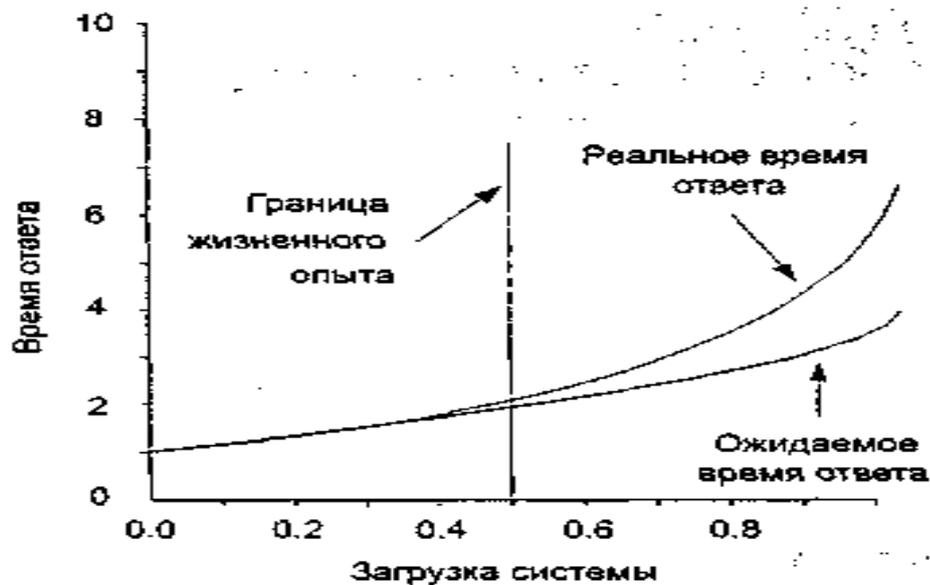


Рис. 1. Ожидаемое и реальное времена ответа системы

3. Определение параметров системы с очередями

Для проведения расчета параметров систем с очередями необходимо определить, что, собственно, входит в состав этой системы и то, какие параметры подлежат оценке. Простейшая система с организацией очереди к серверу показана на рис. 2.

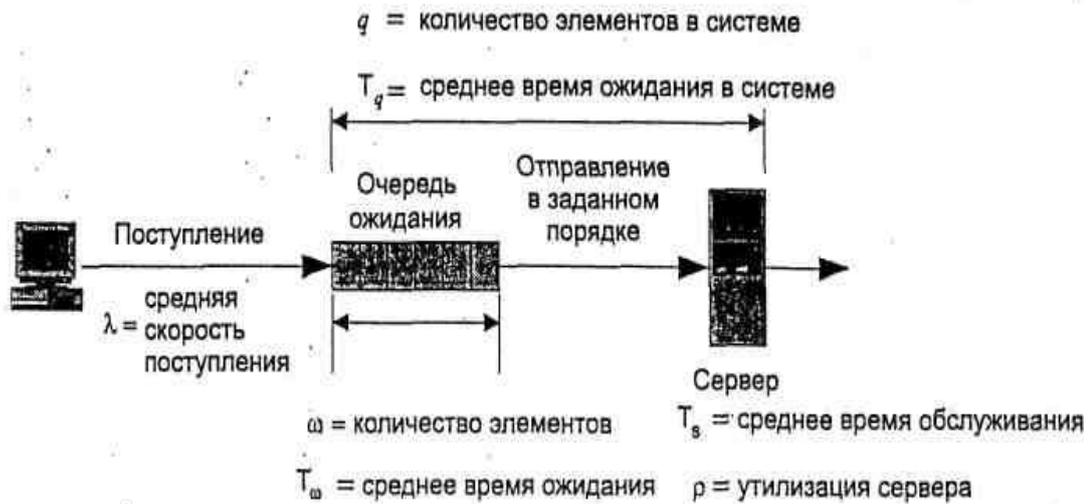


Рис. 2. Базовая схема системы с очередью к серверу

Центральным элементом этой системы является сервер, который производит определенные действия с элементами данных (пакетами, кадрами, дейтаграммами и т. п.). Для рассмотрения этой системы примем некоторые допущения. Все элементы данных, поступающие в систему, сохраняются. Если сервер в определенный момент времени простаивает (свободен), элемент данных обрабатывается немедленно. В противном случае поступающие элементы данных (заявки) сохраняются в очереди. После выполнения сервером обработки определенного элемента данных, он немедленно отправляется по назначению. Если в очереди находятся другие элементы данных, то один из них немедленно поступает на обработку в сервер.

На рис. 2 также показаны некоторые важные параметры, которые используются при расчетах. Элементы данных поступают в эту систему с некоторой средней скоростью поступления λ (она измеряется в числе заявок в секунду). На определенный момент времени некоторое количество элементов данных будет находиться в очереди. Давайте обозначим среднее число элементов данных, находящихся в очереди, буквой ω , а среднее время, которое элементы данных должны ожидать в очереди — символом T_ω . Этот параметр определен и имеет смысл для всех входящих элементов данных, включая и те, которые не ожидали вовсе. Сервер обрабатывает входящие элементы данных, затрачивая на это среднее время обработки T_s . Этот временной интервал отсчитывается от момента поступления элемента данных на обработку вплоть до окончания обработки его сервером (то есть отправки). Утилизация (степень загрузки) сервера ρ — это доля общего времени, в течение которой сервер был занят.

Кроме того, существуют еще два параметра, характеризующие систему в целом. К ним относятся: среднее число элементов данных, находящихся во всей системе, включая элементы данных, которые начали обрабатываться, и элементы, ожидающие обработки, — q и среднее время, которое эти элементы данных находятся в системе, ожидая своей очереди или уже находясь в обработке, — T_q .

Если предположить, что емкость очереди бесконечна, то в такой системе не будет потерянных элементов данных - элементы просто ожидают в очереди до тех пор, пока не будут обработаны. С учетом этого допущения, средняя скорость отправления элементов данных равна средней скорости поступления. Если скорость поступления элементов данных (которая определяется трафиком, входящим в систему) увеличивается, то, естественно, возрастает нагрузка на сервер, а значит, и утилизация сервера. Из таких же естественных соображений мы заключаем, что увеличение размеров очереди повышает (или, по крайней мере, может повысить, но никак не уменьшить) время ожидания элементов данных в ней. При $\rho = 1$ сервер загружается до предела, работая 100 % своего времени. Следовательно, теоретическая максимальная скорость поступления элементов данных, при которой они могут быть обработаны сервером, вычисляется по следующей формуле:

$$\lambda_{max} = \frac{1}{T_S}$$

Однако размер очереди резко возрастает при вхождении системы в режим насыщения, стремясь к бесконечности при $\rho = 1$. Поэтому на практике обычно ограничивают скорость поступления данных на сервер до 70-90 % от теоретического максимума.

Можно показать процессы, происходящие в очереди, в их динамике. На рис. 3 показан график, иллюстрирующий работу системы с очередью. По оси ординат откладывается общее число элементов данных в системе. Затененные области по оси абсцисс определяют периоды времени, в течение которых сервер занят. Вдоль этой же оси располагаются отрезки, означающие события двух типов: поступление элемента данных; (элемент поступает во время A_j) и завершение обработки этого же элемента j (обработка завершается в момент времени D_j), когда элемент покидает систему. Очевидно, время, которое элемент данных j находился в системе, определяется по следующей формуле: $T_j = D_j - A_j$. Минимальное время обслуживания для элемента j обозначается символом S_j .

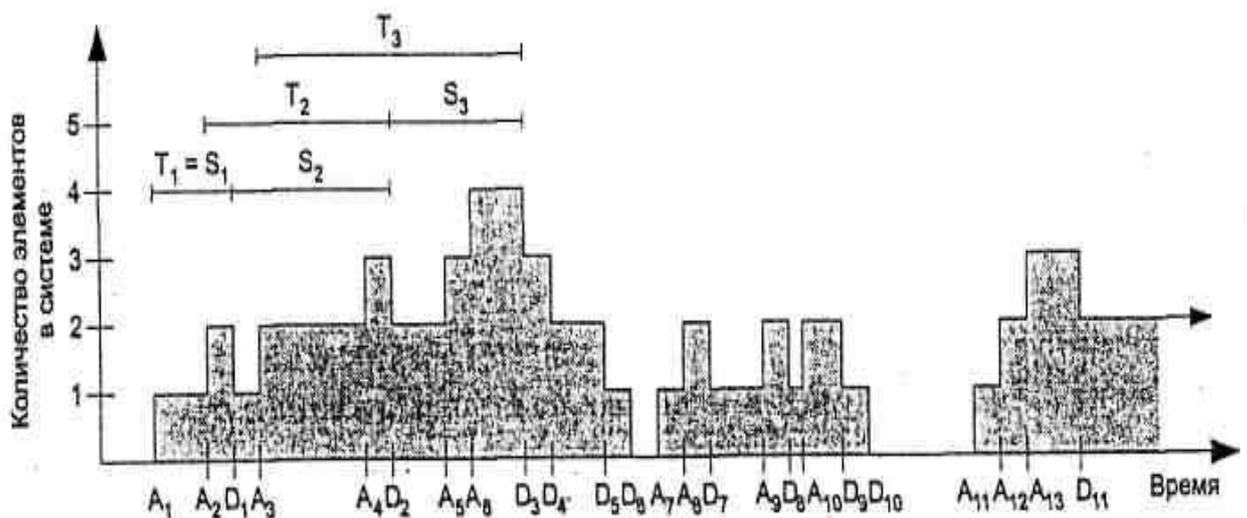


Рис. 3. Поступление элементов и их обработка в системе с очередью

В нашем примере время обработки первого элемента данных T_1 равно минимальному времени обслуживания для этого элемента S_1 , так как когда элемент 1 поступает в систему, она находится в режиме простоя, и элемент сразу же поступает на обслуживание. Время обслуживания второго элемента данных T_2 определяется суммированием двух времен: времени, которое элемент 2 ожидает обслуживания в очереди и которое определяется выражением $(D_1 - A_2)$, и времени, затраченного на его обслуживание S_2 . Аналогично, $T_3 = (D_3 - A_3) = (D_3 - A_3) + (D_3 - A_3) = S_3 + (D_3 - A_3)$.

Перед тем как производить любые вычисления параметров для системы с очередью, необходимо определить условия работы этой системы и выявить диапазон изменений параметров. Ниже приведены условия, при которых нами будет рассматриваться система с очередями:

□ *Определение количества элементов данных.* Предполагается, что в систему может поступать бесконечное количество элементов данных. Это можно интерпретировать как требование о том, что скорость поступления элементов данных в систему никак не зависит от числа элементов, находящихся в системе. Если бы количество элементов данных было ограничено, это означало бы, что скорость поступления элементов в систему снижалась бы при увеличении числа уже обрабатываемых элементов.

□ *Очередь может неограниченно расти.* При рассмотрении системы нами будет предполагаться бесконечный размер очереди. Следовательно, очередь, может расти безгранично. Если ввести ограничения для размера очереди, то элементы данных в системе стали бы отбрасываться при заполнении очереди. Если очередь заполнена, а дополнительные элементы данных продолжают поступать в систему, то сервер ничего не сможет сделать с ними, кроме того как отбрасывать. На практике любая очередь имеет конечный размер, но в большинстве случаев теоретическое допущение о ее безграничной вместительности не приводит к существенным ошибкам, так как реальные устройства используют различные механизмы предотвращения ситуаций, при которых будет производиться отбрасывание данных.

□ *Определенный порядок обслуживания элементов данных.* Когда сервер становится свободным, и в очереди находится несколько элементов данных, необходимо определить правила, в соответствии с которыми определяется элемент данных, выбираемый сервером для обработки. Простейшим правилом является обслуживание очереди по принципу FIFO (First In, First Out — первым пришел, первым и ушел). Другим возможным правилом обслуживания очереди может быть LIFO (Last In, First Out — первым пришел, последним ушел). Кроме того, на практике приходится иметь дело с порядком обслуживания, базирующимся на времени обслуживания. К сожалению, обслуживание очереди, основанное на временных параметрах, достаточно сложно для моделирования. Более общим случаем является обслуживание очереди на основе приоритетов. Однако, ниже рассматривается поступление пакетов с одинаковым приоритетом.

В табл. 1 перечислены все параметры, определенные на рис. 2, а также добавлены новые параметры, которые будут использованы далее при проведении расчетов, в том числе и для систем с множеством серверов.

Таблица 1. Используемые параметры

Символ	Описание
Λ	Средняя скорость поступления элементов данных в систему (число элементов в секунду)
T_S	Среднее время обслуживания поступивших элементов (в секундах)
σ_{T_S}	Стандартное отклонение во времени обслуживания элемента (в се-
P	Утилизация сервера при обслуживании (доля времени, когда сервер
U	Интенсивность трафика (Эрл)
Q	Общее количество элементов данных в системе
\bar{Q}	Среднее количество элементов данных в системе
T_σ	Время, которое элементы данных проводят в системе (в секундах)
T_q	Среднее время, которое элементы данных проводят в системе (в се-
σ_q	Стандартное отклонение q
σ_{T_S}	Стандартное отклонение T_q , (в секундах)
Ω	Среднее количество элементов данных, ожидающих обслуживания в очереди (размер очереди)
T_ω	Среднее время, которое элементы данных ожидают обслуживания (в секундах)
T_d	Среднее время ожидания обслуживания для элементов данных, находившихся в очереди (то есть, не включая элементы, для кото-рых время ожидания равно 0)
σ_ω	Стандартное отклонение ω
N	Число серверов
$m_x(r)$	x меньше или равно $m_x(r)$ в r процентах случаев (примеры см. ни-

На рис. 4,а показана простая модель, которая будет использована нами в случае наличия в системе множества серверов, разделяющих одну общую очередь. При поступлении элементов данных в такую систему, если на данный момент времени хотя бы один сервер свободен, элементы данных немедленно направляются на этот сервер. Предполагается, что в системе все сервера идентичны. Это значит, что если доступны несколько серверов, то не делается никаких различий между серверами для выбора того, который будет обрабатывать очередной элемент данных. Можно сказать, что вероятность поступления элементов данных для обслуживания на разные сервера одинакова. Если все сервера заняты, то начинает формироваться очередь. Очередь одна для всех серверов. При освобождении одного из серверов очередь покидает элемент данных, выбранный в соответствии с установленным порядком.

За исключением параметра утилизации серверов, все остальные параметры, приведенные в табл. 1, соответствуют ситуации, показанной на рис. 2. Если в системе N идентичных серверов, а ρ обозначает утилизацию каждого сервера, то можно рассматривать $N\rho$ как утилизацию всей системы. Этот показатель часто рассматривают как интенсивность трафика (в табл. 1 обозначена символом u)

или интенсивность работы системы. Следовательно, теоретическая максимальная утилизация такой системы (при $\rho = 1$) будет равна N . Максимальная скорость поступления элементов данных в такую систему будет определяться по формуле:

$$\lambda_{max} = \frac{N}{T_S}$$

В случае множества идентичных серверов выбор определенного сервера для обслуживания определенного элемента данных не влияет на время обслуживания. На рис.4,б показана структура с организацией нескольких очередей для множества серверов. Такое изменение структурной схемы в значительной степени влияет на производительность всей системы в целом.

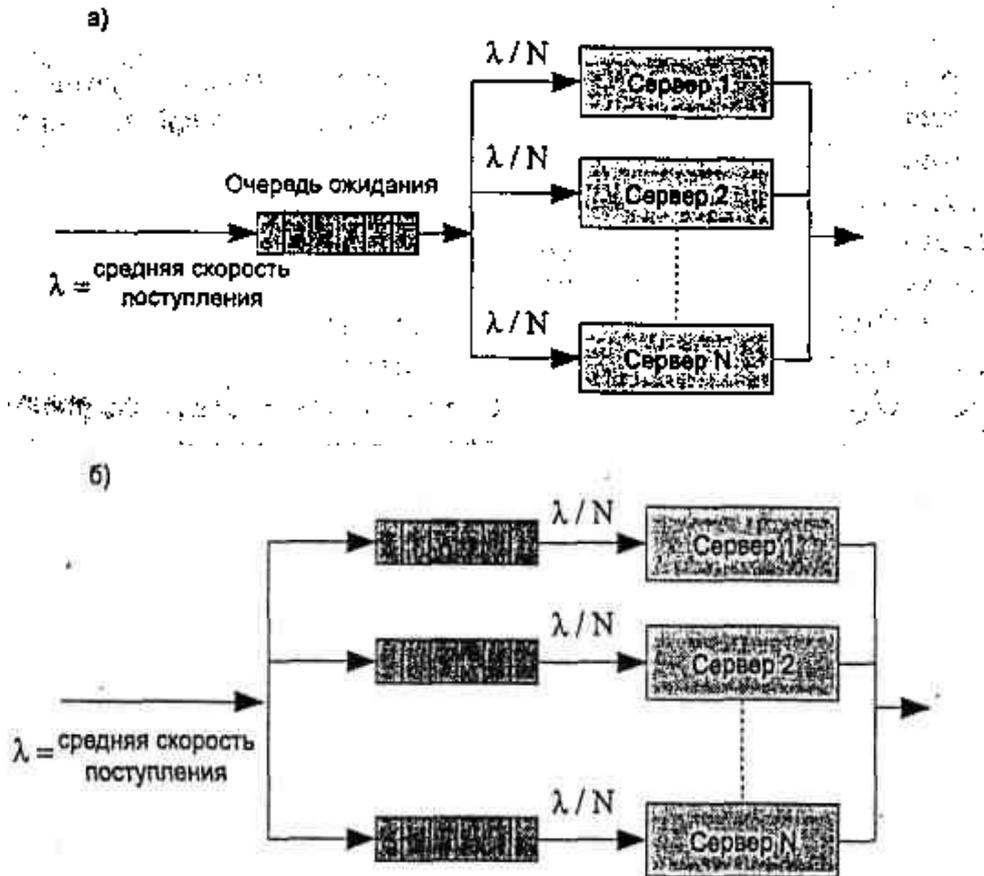


Рис. 4. Различные схемы организации очереди к N серверам

Искомые параметры можно вычислить с помощью нескольких несложных формул, перечисленных в табл. 2. Такие формулы могут быть использованы для вычисления качественных и количественных характеристик систем с очередями, представленных на рис. 2 и рис. 4. Следует подчеркнуть, что вычисления с применением этих формул носят приближенный характер.

Таблица 2. Формулы для расчета параметров систем с очередями

Основные	Один сервер	Множество серверов
$q = \lambda T_S$	$\rho = \lambda T_S$	$\rho = \frac{\lambda T_S}{N}$

$\omega = \lambda T_\omega$	$q = \omega + \rho$	$\omega = \omega + Nq$
$T_q = T_\omega + T_S$		$u = \lambda T_S = \rho N$

Эти формулы могут быть полезны для вычисления некоторых параметров при «интуитивном» выборе структуры системы. Для получения формулы $\rho = \lambda T_S$ достаточно заметить, что для скорости поступления элементов данных λ , среднее время между поступлениями элементов будет определяться выражением $T = 1/\lambda$. Если интервал времени T меньше интервала времени T_S , то можно будет записать $T_S/T = \lambda T_S$. Аналогичные рассуждения подходят и в случае со множеством серверов: $\rho = (\lambda T_S)/N$ (в данном случае ρ – это утилизация одного сервера).

Рассмотрим момент времени прихода очередного элемента данных. При поступлении этого элемента в систему в очереди находится в среднем ω элементов данных, которые ожидают обслуживания. В момент, когда этот элемент покидает очередь для обслуживания, он оставляет после себя такое же среднее количество элементов в очереди (на то оно и среднее количество, что не меняется). Среднее время, которое элемент ждет своей очереди до обслуживания, равно T_S . А так как элементы поступают в очередь со скоростью λ , то можно утверждать, что за промежуток времени T_ω должно поступить λT_ω элементов данных. Следовательно, $\omega = \lambda T_\omega$. Рассуждая аналогично, можно заключить, что $q = \lambda T_q$.

Из табл. 2 видно, что время, которое элемент данных находится в системе, равно сумме времени ожидания обслуживания и времени самого обслуживания. На любой момент времени количество элементов во всей системе равно сумме количества элементов, ожидающих обслуживания, и количества элементов, которые уже обслуживаются. Для одного сервера среднее число элементов, которые обслуживаются в данный момент времени, равно ρ . Следовательно, $q = \omega + \rho$ для одного сервера. Аналогично, $q = \omega + N\rho$, если рассматривать случай с N серверами.

Основная задача при проведении анализа очередей заключается в получении следующей информации:

- скорость поступления элементов данных в очередь;
- время обслуживания этих элементов на сервере на входе в систему;
- общее количество ожидающих элементов;
- время ожидания элементов в системе.

При этом важно знать средние значения этих параметров и диапазон их изменений. Следовательно, большое значение при анализе очередей имеет знание стандартных (среднеквадратичных) отклонений каждого из перечисленных параметров; эти отклонения обозначаются σ_q , σ_{T_S} , σ_ω и σ_{T_ω} .

Для анализа систем или отдельных модулей сетевых устройств могут быть полезны и другие показатели. Например, при вычислении размеров буферной памяти для моста или мультиплексора, предназначенного для того или иного сегмента рынка, его производителям могут потребоваться данные о размере буфера, при котором вероятность его переполнения будет меньше, допустим, 0.001. Для ответа на эти вопросы, в основном, необходимо знать закон изменения ско-

рости поступления элементов данных в систему и закон распределения времени обслуживания элементов данных сервером. Следует отметить, что, даже имея эти данные, очень непросто получить результат элементарными методами, так как исходные формулы для вычислений достаточно сложны. Для того чтобы упростить процесс вычислений, необходимо сделать некоторые естественные допущения. Наиболее важное из этих допущений заключается в том, что изменение скорости поступления элементов данных подчиняется закону Пуассона. Для применимости закона Пуассона необходимы три условия простейшего потока: стационарность, ординарность, отсутствие последействия.

При соблюдении этих условий вероятность поступления элемента данных подчиняется закону Пуассона, который описывается следующей формулой:

$$P_i(t) = \frac{e^{-\lambda t} (\lambda t)^i}{i!},$$

где $P_i(t)$ — вероятность поступления ровно i вызовов за время t ; λ — скорость поступления вызовов.

Закон Пуассона часто используется в различных приложениях теории вероятности и статистики. Его преимуществом является простота получаемых формул. Практически всегда, когда последовательность каких-либо событий разделена случайными интервалами времени и справедливы три перечисленные выше условия, то, с некоторым приближением, можно использовать пуассоновский закон.

Для продолжительности обслуживания элементов на сервере обычно используется закон интервалов или экспоненциальный закон. Для подтверждения этого закона используется большое количество продолжительностей времен обслуживания. Рассмотрим пример с использованием 1000 продолжительностей времен обслуживания (каждый промежуток из этой 1000 отличается по времени). Такое количество продолжительностей позволяет повысить точность вычислений. Для упрощения расчетов интервалы времени обслуживания группируются. Например: первая группа с временами обслуживания, лежащими в промежутке времени от 0 до 15 с, следующая группа с временами обслуживания, лежащими в промежутке времени от 15 до 30 с, следующая — от 30 до 45 с и т. д. Приведем таблицу, поясняющую смысл сказанного (табл. 3).

Для проведения оценки системы можно определить параметр, характеризующий интенсивность работы системы. Так как рассматриваются средние величины, важно, чтобы интенсивность поступления элементов данных не превосходила пропускной способности системы обслуживания, то есть $\lambda < \rho N$, или $\lambda/\rho N < 1$. Эта величина и определяет интенсивность работы системы.

Таблица 3. Пример распределения интервалов обслуживания для экспоненциального закона

Сгруппированные интервалы времени	
0 -15	202
15 – 30	167

30 – 45	127
45 – 60	102
60 – 75	82
75 – 90	65
90 – 105	52
105 – 120	42
120 – 135	34
135 – 150	26
150 – 165	21
165 – 180	17
180 – 195	14
195 – 210	11
210 – 225	8
225 – 240	7
240 – 255	6
255 – 270	4
270 – 285	3
285 – 300	3
>300	11

Существуют классические формулы, которые были выведены датским инженером Эрлангом при проведении им аналитического изучения очередей. Эрланг изучал очереди применительно к работе телефонной сети. Составлены специальные таблицы, с помощью которых можно определить ряд параметров для системы с очередями. Например, можно определить среднее время ожидания элементов в очереди как функцию интенсивности обслуживания, уровня утилизации и количества серверов.

Для обобщения всех возможных (или точнее, всех вероятных) случаев организации системы с очередями, к которым применимы рассмотренные выше допущения, был разработан удобный подход – нотация Кендалла. Все такие системы можно разделить, исходя из применяемых законов распределения времен обслуживания и поступления в систему. Система (в том аспекте, который мы рассматриваем) может быть определена тройкой $X/Y/N$, где X — это закон распределения времени поступления элементов данных в систему; Y — закон распределения времени обслуживания элементов данных сервером и N — число серверов. Для рассматриваемых здесь систем характерны следующие возможные законы распределения (ниже также указаны буквы, которыми обозначаются эти законы):

□ G — произвольное распределение времени поступления или времени обслуживания элементов данных;

□ M — пуассоновское распределение времени поступления; пуассоновское или экспоненциальное распределение времени обслуживания элементов данных;

□ D — детерминированное время поступления или время обслуживания элементов данных.

Следовательно, модель $M/M/1$ определяет систему с одним сервером, пуассоновским распределением времени поступления элементов данных в систему и

экспоненциальным временем обслуживания элементов на сервере.

4. Система с одним сервером

В первом столбце табл. 4 показаны формулы для определения некоторых параметров системы с одним сервером, которая подчиняется модели $M/G/1$. В соответствии с этой моделью скорость поступления элементов данных подчиняется пуассоновскому закону, а время обслуживания — произвольному распределению. Использование масштабирующего коэффициента A в значительной мере упрощает формулы для вычисления основных выходных параметров. Следует учесть, что коэффициент масштабирования зависит только от отношения стандартного (среднеквадратичного) отклонения времени обслуживания к среднему времени обслуживания (см. формулу). При этом не требуется никакой другой информации о времени обслуживания.

Другие два случая, разобранные в табл. 4, это — модель с распределением времени ожидания по пуассоновскому закону, а времени обслуживания по экспоненциальному закону ($M/M/1$, второй столбец) и модель, в которой время обслуживания всех элементов одинаково (а значит, отклонение времени обслуживания равно нулю), а время поступления элементов подчиняется пуассоновскому закону ($M/D/1$, третий столбец в табл. 4). Как мы уже отмечали, вычисления при помощи этих формул носят приближенный характер, но для практического применения их точности вполне достаточно.

Таблица 4. Формулы для определения параметров системы с одним сервером

Модель с произвольным распределением времени обслуживания ($M/G/1$)	Модель с экспоненциальным распределением времени обслуживания ($M/M/1$)	Модель с постоянным временем обслуживания ($M/D/1$)
$A = \frac{1}{2} \left[1 + \left(\frac{\sigma r_s}{T_s} \right)^2 \right]$	$q = \frac{\rho}{1-\rho}; \omega = \frac{\rho^2}{1-\rho}$	$q = \frac{\rho^2}{2(1-\rho)} + \rho$
$q = \sigma + \frac{\rho^2 A}{1-\rho}$	$T_q = \frac{T_s}{1-\rho}; T_\omega = \frac{\rho T_s}{1-\rho}$	$\omega = \frac{\rho^2}{2(1-\rho)}$
$\omega = \frac{\rho^2 A}{1-\rho}$	$\sigma_q = \frac{\sqrt{\rho}}{1-\rho}; \sigma_{T_q} = \frac{T_q}{1-\rho}$	$T_q = \frac{T_s(2-\rho)}{1(1-\rho)}$
$T_q = T_s + \frac{\rho T_s A}{1-\rho}$	$Pr[Q = N] = (1-\rho)\rho^N$	$T_\omega = \frac{\rho T_s}{1(1-\rho)}$
$T_\omega = \frac{\rho T_s A}{1-\rho}$	$Pr[Q \leq N] = \sum_{i=1}^N (1-\rho)\rho^i$	$\sigma_q = \frac{1}{1-\rho} \sqrt{\rho + \frac{3\rho^2}{2} + \frac{5\rho^2}{6}}$
	$Pr[Q \leq t] = 1 - e^{-(1-\rho)t/T}$	$\sigma_{T_q} = \frac{T_s}{1-\rho} \sqrt{\frac{\rho}{3} - \frac{\rho^2}{12}}$
	$m_{T_q}(r) = T_q \ln \frac{100}{100-r}$	
	$m_{T_\omega}(r) = T_\omega \ln \frac{100}{100-r}$	

Практика показывает, что наихудшую производительность демонстрирует система с экспоненциальным распределением времени обслуживания, а наилучшую производительность — система с постоянным временем обслуживания (что, впрочем, неудивительно). Поэтому обычно можно рассматривать систему с экспоненциальным распределением времени обслуживания, как систему с худшими параметрами.

5. Система с несколькими серверами

В табл. 5 приведены формулы для определения основных параметров в случае работы с системой со множеством серверов. Эти формулы применимы только для случая использования модели $M/M/N$. То есть предполагается пуассоновский характер распределения количества поступающих элементов данных и экспоненциальный характер времени обслуживания этих элементов. Во всех выражениях используется функция Эрланга - C , которая, в одних случаях, определяет вероятность того, что все сервера заняты в определенный момент времени, а в других случаях — вероятность того, что количество элементов данных, находящихся в данный момент времени в системе (ожидающих в очереди или обслуживаемых), будет больше или равно количеству серверов. Для вычисления функции C применима следующая формула:

$$C(N, u) = \frac{1 - K}{1 - \rho K},$$

где K — коэффициент пуассоновского распределения.

Значение этой функции зависит от количества серверов (N) и их утилизации (ρ). Функцию Эрланга приходится часто применять при расчете очередей, что значительно усложняет вычисления. Следует отметить, что для системы с одним сервером эта функция значительно упрощается. А именно: $C = (1, u) = \rho$. Такое упрощение как раз и позволяет для системы с одним сервером получить красивые законченные формулы (табл. 5).

Таблица 5. Формулы для определения параметров системы со множеством серверов

$K = \frac{\sum_{i=1}^{N-1} \frac{(N\rho)^i}{i!}}{\sum_{i=1}^N \frac{(N\rho)^i}{i!}}$	$\sigma_\omega = \frac{1}{(1-\rho)} \sqrt{C\rho(1+\rho-C\rho)}$
$q = C \frac{\rho}{1-\rho} + N\rho$	$Pr[T_\omega = t] = C e^{-N(1-\rho)t/T_q}$
$T_q = \frac{C}{N} \frac{T_s}{1-\rho} + T_s$	$T_d = \frac{T_s}{N(1-\rho)}$
$T_\omega = \frac{C}{N} \frac{T_s}{1-\rho}$	$\omega = C \frac{\rho}{1-\rho}$

$\sigma_{T_q} = \frac{T_s}{N(1-\rho)} \sqrt{C(2-C) + N^2(1-\rho)^2}$	$m_{T_\omega} = \frac{T_s}{N(1-\rho)} \ln \frac{100C}{100-r}$
--	---

Рассмотренная теория очередей достаточно эффективно может быть использована на практике в различных ситуациях.

6. Пример практического применения теории очередей.

Рассмотрим локальную сеть, имеющую в своем составе 100 рабочих станций и один сервер ($N = 1$), который обслуживает общую базу данных. Среднее время ответа сервера на запрос — 0.6 с. Стандартное отклонение этого времени также равно 0.6 с. В пиковые периоды работы локальной сети скорость поступления запросов к серверу достигает значения 20 запросов в минуту.

Ответим на следующие вопросы:

- Чему равно среднее время ответа сервера?
- Если время ответа, равное 1.5 с, рассматривается как максимально приемлемое, то насколько может вырасти процент загрузки до достижения насыщения сервера?
- Если ожидается, скажем, 20-процентное увеличение утилизации сервера, то насколько увеличится время ответа (на 20 %, больше чем 20 %, меньше чем 20 %)?

Предположим, что в рассматриваемой ситуации применима модель $M/M/1$. Будем игнорировать задержки, вносимые сетью, полагая, что задержки в ней можно не принимать в расчет.

Вычислим некоторые параметры сети. Сначала найдем интенсивность поступления вызовов λ :

$$\lambda = 20 \text{ поступлений в минуту} = 20/60 \text{ поступлений в секунду} = 1/3 \text{ поступлений в секунду.}$$

Утилизация сервера вычисляется по формуле:

$$\rho = \lambda T_s = (1/3 \text{ поступлений в секунду}) (0.6 \text{ секунд на обслуживание}) = 0.2.$$

Вычислим среднее время ответа (ожидание + обслуживание):

$$T_q = T_s / (1 - \rho) = 0.6 / (1 - 0.2) = 0.75 \text{ с.}$$

На второй вопрос однозначно ответить сложно, так как существует ненулевая вероятность того, что в некоторых случаях время ответа сервера будет превышать 1.5 с. Поэтому можно предположить, что в 90 % ответы сервера будут даны менее чем за 1.5 с. Если сделать такое допущение, то мы сможем воспользоваться формулой из второго столбца в табл. 4:

$$m_{T_q}(r) = T_q \ln(100 / (100 - r))$$

Получаем:

$$m_{T_q}(90) = T_q \ln(10) = 2.3 T_q = 2.3 T_s / (1 - \rho) = 1.5 \text{ с.}$$

Учитывая, что $T_s = 0.6 \text{ с}$, получаем для данного случая утилизацию сервера $\rho = 0.08$, то есть 8 %. Итак, можно сказать, что при изменении загруженности сервера в диапазоне от 8 % до 20 % (см. выше) время ответа сервера будет менее 1.5 с в 90 % случаев.

В заключение определим зависимость между возрастанием нагрузки и увеличением времени ответа. Время ответа будет увеличиваться несколько медленнее, чем утилизация. Действительно, в нашем случае, если утилизация сервера выросла с 20% до 40 %, то значение T_q изменится от 0.75 с до 1.0 с (как нетрудно подсчитать), что означает увеличение на 33.3 %. Однако, при приближении ρ к единице значение T_q начинает резко возрастать, устремляясь к бесконечности за счёт возрастания до бесконечности длины очереди.

Литература

1. Максимов В.А. Маршрутизация в IP-сетях. Питер. М. 2003.