

УДК 004.716

ПОСТРОЕНИЕ ИМИТАЦИОННОЙ МОДЕЛИ СЕТЕВОГО КОНТРОЛЛЕРА КАНАЛЬНОГО УРОВНЯ

© 2017 Е.Н. Яблоков

Санкт-Петербургский государственный университет аэрокосмического приборостроения

Статья поступила в редакцию 21.03.2017

Статья посвящена первичной оценке характеристик сетевых контроллеров на канальном уровне. Рассматривается методика построения моделей сетевого контроллера с помощью имитационного моделирования на основе систем массового обслуживания.

Ключевые слова: *система массового обслуживания, канальный уровень, сетевой контроллер, точка-точка*

Канальный уровень является вторым уровнем из семи модели OSI для построения вычислительных сетей. Этот уровень представляет собой механизм, который позволяет передавать данные между соседними узлами глобальной сети или между узлами в локальном сегменте той же сети. Канальный уровень описывает различные функции и методы для передачи данных между объектами в сети и так же предоставляет средства для обнаружения и исправления ошибок (если такая задача ставится), которые могут возникнуть на физическом уровне при передаче данных.

Попробуем рассмотреть систему канального уровня как систему массового обслуживания (СМО). В любую систему канального уровня поступает полезная информация, которая передается через ее структуру. В качестве полезной информации системы канального уровня рассматривается информация, которая поступает из более высоких уровней и передается посредством канального уровня и выдается на более высокий уровень, или информация, которая генерируется на канальном уровне и так же передается на более высокий уровень. Данную информацию можно рассматривать как поток заявок, которые генерируются по определенным законам и являются входным потоком для СМО канального уровня. Необходимо учитывать, что данные потоки могут быть как с одним приоритетом, так и с разными приоритетами обслуживания.

В зависимости от типа устройства канального уровня разрабатываемая система канального уровня может являться как СМО с ожиданием – при наличии некоторого физического буфера на входе системы, так и системой с отказами (при отсутствии такого буфера или в зависимости от конфигурации создаваемого оборудования). На практике моделирование устройств канального уровня осуществляется редко. Практически всегда модель сетевого контроллера встраиваются в модели более высокого уровня не на основании конкретных математических расчетов, а на основании предположений о

пропускной способности более высоких уровней, совершенно игнорируя данные, которые ходят только на канальном уровне и которые не надо передавать выше. Широко распространена идея, что издержки канального уровня, такие как размеры памяти, которая требуется для хранения данных, пропускная способность внутренних каналов или блоков (например, подсчет контрольной суммы), не влияют на пропускную способность и аппаратные издержки. Важная функция управления потоков совершенно не учитывается на подобных моделях. Поэтому построение моделей канального уровня необходимо.

При разработке СМО сетевого контроллера канального уровня можно поставить задачу расчета следующих характеристик:

- среднее время прохождения данных через контроллер (важно для данных, которые критичны ко времени)
- средняя пропускная способность контроллера, т.е. сколько данных возможно передать через контроллер
- средняя длина очереди – для расчета глубины буферов FIFO, используемых в сетевом контроллере

Данные характеристики являются основными и должны быть рассчитаны до начала создания устройства в железе, так как они смогут дать общее представление о характеристиках, которыми будет обладать конечное аппаратное устройство.

Модель будет строиться из расчета характеристик сетевого контроллера для канала точка-точка. Все данные, которые поступили в сетевой контроллер, должны быть переданы, потеря данных или отказ от обработки внутри контроллера не допускается. Сетевой контроллер канального уровня состоит из 2 основных частей: приемная часть, передающая часть и коммуникационная среда, через которую осуществляется передача данных. Два сетевых контроллера осуществляют передачу с использованием коммуникационной среды. В подавляющем числе случаев используется дуплексный режим передачи, когда передача данных идет в обе стороны (рис. 1).

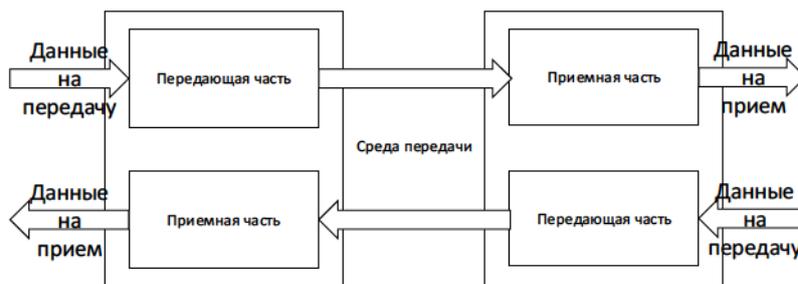


Рис. 1. Простая схема сетевого контроллера канального уровня канала точка-точка

Как видно из рис. 1, осуществляется параллельная передача в оба направления. Данные могут передаваться с одним или с различными приоритетами. Сетевой контроллер может или передавать данные о готовности (контроллер с управлением потоком), или быть готовым принять любое количество данных (контроллер без управления потоком). Контроллеры без управления встречаются сильно реже и используются при низкой интенсивности данных. Представим обмен канального уровня в качестве модели на основе СМО. В общем случае модель контроллера канального уровня представляет

собой (рис. 2) устройство для передачи заявок различных типов. Данная модель будет отправлять заявки через всю описанную систему. В такой модели часть заявок могут быть более приоритетны, чем другие. Данная СМО описывает канал соединения точка-точка между передатчиком и приемником через коммуникационную среду. Возможность реализации коммуникаций на других структурах не рассматривается, так как современные высокоскоростные протоколы используют исключительно каналы точка-точка. Реализации сетевых контроллеров на других структурах не рассматриваются.

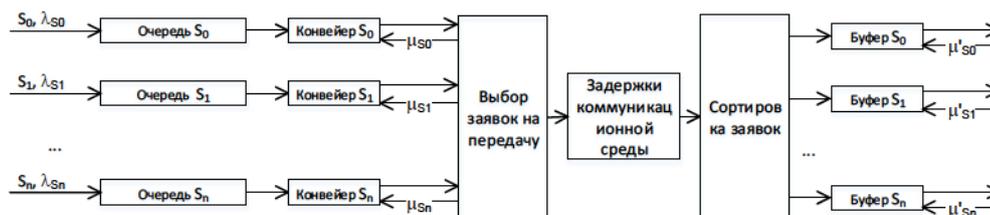


Рис. 2. Модель сетевого контроллера канального уровня

В этой статье рассмотрены коммуникационные системы на базе непосредственных связей между узлами, определен набор характеристик, существенных для разработки сетевого контроллера, разработана методика оценки данных с точки зрения их применимости для сетевого контроллера. Так как данные не могут не быть обработаны в сетевом контроллере – весь сетевой контроллер будем рассматривать как СМО с ожиданием, то есть без отказов. Поток заявок, поступающих в данную СМО, будем считать пуассоновским с учетом аппроксимации Клейнрока [1] и теоремы Джексона [2], которые допускают независимость потока в каждом узле, независимости времени поступления заявки и ее длины от предыдущих заявок, при этом считается, что поток заявок стационарен и время поступления и обслуживания заявок распределены по экспоненциальному закону.

Практически любая система канального уровня представляет собой тот или иной вариант конвейера. Конвейерная обработка может находиться как в приемной, так и в передающей части блока. Если рассматривать конвейер с точки зрения СМО, то конвейер, который состоит из n ступеней, можно рассматривать как многофазную СМО с бесконечными очередями между фазами (рис. 3). Время обработки одной заявки одной фазой конвейера будет зависеть от количества времени, которое будет потрачено на этом этапе конвейера. В этом случае время прохождения заявки через конвейер будет равно сумме затраченного времени на каждом шаге конвейера, по которой формируется интенсивность обработки заявки. Все СМО являются одноканальными с бесконечной очередью. Для этого класса СМО выходной поток является пуассоновским.

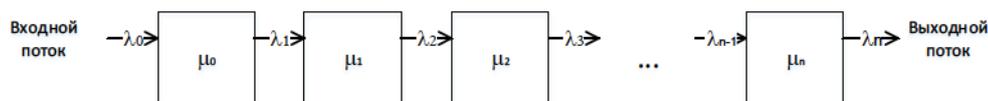


Рис. 3. Аппаратный конвейер с точки зрения СМО

В данной общей схеме могут передаваться заявки только одного типа (если аппаратный конвейер передает только данные одного типа), и заявки

нескольких типов (если идет передача всех типов данных и разрешений по одному каналу). В общем случае конвейер передает заявки разных типов,

которые могут находиться на разных фазах конвейера. Для систем M/M/1 теорема Берга [3] позволяет считать выходной поток так же пуассоновским.

Общий вид конвейера редко используется в аппаратуре. Аппаратные конвейеры чаще всего можно разбить на несколько подтипов, конвейеры с обработкой одного типа данных (или одной заявки), данный тип обработки встречается очень часто, конвейер с одинаковым временем обработки на каждой фазе с возможностью блокирования на последнем этапе, такой тип конвейера используется при подготовке определенного типа данных к передаче. Как пример можно рассмотреть двухфазный конвейер, на первой фазе которого происходит добавление заголовка и подсчет CRC на втором этапе.

Введем следующие обозначения: $T_{\text{конв}}$ – время прохождения заявки через все стадии конвейера, $T_{\text{конв}}^i$ – для i -й заявки; $\mu_{\text{конв}}$ – интенсивность обработки конвейера, $\mu_{\text{конв}}^i$ – для i -й заявки; T_s – время прохождения заявки через стадию конвейера s , T_s^i – для i -й заявки; μ_s – интенсивность обработки заявки стадией конвейера s , μ_s^i – для i -й заявки; $L_{\text{оч}}$ – длина очереди заявок; $L_{\text{сист}}$ – общее количество заявок в системе. Для расчета характеристик данной СМО можно разделить конвейер на составные части вида A1|B1|1, A2|B2|1 ... AN|BN|1, где каждая стадия конвейера будет представлена отдельной СМО. В этом случае общее время задержки заявки в системе будет равно сумме задержек каждого этапа конвейера, а именно:

$$T_{\text{конв}} = T_1 + T_2 + \dots + T_N$$

Интенсивность же обработки заявки конвейером будет определяться временем обработки самой медленной заявки

$$\mu_{\text{конв}} = \min(\mu_1, \mu_2, \dots, \mu_N)$$

Количество заявок, которое находится непосредственно в конвейере, равно количеству ступеней конвейера, количество заявок в очередях не ограничено. В нашем случае мы будем рассматривать эти СМО как M/M/1 [4], так как на практике каждый этап конвейера в сетевых контроллерах определяется физической структурой и все задержки, как и время обработки, можно предположить предварительно. Среднюю задержку заявки на каждом этапе можно рассчитать по формуле:

$$T_{\text{сист}} = \frac{p}{\lambda(1-p)}$$

Таким образом, задержки для одной конкретной СМО можно рассчитать по этой формуле, как и количество заявок, которые находятся в данной конкретной СМО. Рассчитав задержку каждой СМО можно выяснить общую задержку каждой конкретной СМО всего конвейера для заявки i :

$$T_{\text{конв}}^i = \sum_{k=1}^M T_k$$

где M – количество стадий конвейера.

Исходя из вышеизложенного, интенсивность потока входных заявок для конвейерных СМО для

каждой заявки не должна быть больше интенсивности обслуживания заявок каждой ступенью конвейера, т.е.

$$\begin{cases} \lambda_{\text{конв}}^i < \mu_{\text{конв}}^i \\ \mu_{\text{конв}}^i = \min(\mu_1^i, \mu_2^i, \dots, \mu_M^i) \end{cases}$$

Среднее количество заявок в системе, как и среднее количество заявок в очередях считается как сумма всех заявок во всех стадиях конвейера.

Второй неотъемлемой частью любой системы канального уровня является блок выбора передаваемых данных (или заявок). В зависимости от ситуации – выбор может осуществляться или в абсолютно-приоритетном доступе, или относительно приоритетном доступе, или бесприоритетном доступе. При абсолютно приоритетном доступе все поступающие заявки имеют свой приоритет. Чем выше приоритет канала, тем больше вероятность обработки заявки данного канала. При относительном приоритете некоторая группа, или группы заявок имеют приоритет над другой группой, или группами заявок, внутри группы приоритетов нет. При бесприоритетном доступе все заявки имеют одинаковый приоритет и отправляются последовательно из каждого канала (рис. 4).



Рис. 4. Модель СМО с одним каналом и несколькими заявками

Блок выбора канала изначально будем рассматривать как общий случай блока с приоритетом обслуживания заявок M/G/1, причем с тем свойством, что поступающие заявки делятся на несколько классов различного приоритета, причем 1-й класс имеет наивысший приоритет, 2-й класс – второй по величине приоритет и т. д. Данное доказательство для приоритетного доступа было взято из доказательств для доступа с приоритетом для различных сетевых устройств [5].

СМО с одной очередью, в которой заявки поступают в соответствии с пуассоновским процессом с интенсивностью λ , длительности обслуживания заявок имеют экспоненциальное распределение. Предположим, что заявки обслуживаются в порядке поступления, X_i – длительность обслуживания i -й заявки, случайные величины (X_1, X_2, \dots) одинаково распределены, взаимно независимы и не зависят от интервалов между моментами поступления. Рассматривая эту систему необходимо учитывать, что $\bar{X}_1 = E\{X\} = \frac{\lambda}{\mu}$ – средняя длительность обслуживания;

$\bar{X}_2 = E\{X^2\}$ – второй момент длительности обслуживания.

Для начала рассмотрим систему, где часть заявок находится в очереди и на вход поступает i -я заявка, в обслуживании находится заявка j , а в очереди некоторое количество заявок. Система обладает следующими характеристиками:

W_i – время ожидания в очереди для i -й заявки;

R_i – остаточное время обслуживания в момент поступления i -й заявки. Это означает, что если заявка находится на обслуживании в момент появления заявки i , то R_i равно промежутку времени от этого момента до завершения обслуживания заявки. Если нет заявки, которая обслуживается (т. е. система пуста в момент поступления заявки i), R_i равно нулю;

X_i – длительность обслуживания i -й заявки

N_i – число требований, которые в очереди при поступлении i -й заявки

Для формирования задержек в очередях для каждой заявки воспользуемся следующей формулой

$$W_i = R_i + \sum_{j=i-N_j}^{i-1} X_j$$

Проводя соответствующие преобразования для системы M/G/1, - получаем следующий результат для задержек в системе и очередях. Задержки для заявок k -го приоритета рассчитываются по формуле

$$W_k = \frac{R}{(1-p_1-p_2-\dots-p_{k-1})(1-p_1-p_2-\dots-p_k)}$$

Из данной формулы единственным неизвестным параметром представляется среднее остаточное время дообслуживания. Его можно вычислить графическим способом, где получается

$$R = \frac{1}{2} \sum_{i=1}^n \lambda_i \bar{X}_i^2$$

Как итог – среднее время ожидания в очереди будет составлять

$$W_k = \frac{\frac{1}{2} \sum_{i=1}^n \lambda_i \bar{X}_i^2}{(1-p_1-p_2-\dots-p_{k-1})(1-p_1-p_2-\dots-p_k)}$$

Как говорилось выше, второй момент длительности обслуживания i -й заявки представляется как $\bar{X}_i^2 = E\{X^2\}$. Для системы M/G/1 расчет может вызвать затруднения, однако, в нашем случае наша система представляет собой СМО M/M/1, и в этом случае \bar{X}_i^2 соответственно равен $\bar{X}_i^2 = \frac{2}{\mu^2}$ для системы M/M/1. Средняя задержка в данном блоке заявки k -го приоритета T_k будет составлять

$$T_{\text{выбр}}^k = \frac{1}{\mu_k} + W_k$$

Теперь рассмотрим вариант, когда вместо относительных приоритетов данный блок использует абсолютный приоритет, то есть заявки с более высоким приоритетом поступают в обслуживание сразу, а заявки с более низким – ждут и дообслуживаются после. При вычислении среднего времени

ожидания в очереди заявки k -го порядка необходимо учитывать приоритеты с номером меньше k , в то время как все заявки с приоритетом больше k игнорируются. В этом случае задержку для данного блока вычисляем по формуле абсолютного приоритета

$$T_{\text{выбр}}^k = \frac{\left(\frac{1}{\mu_k}\right) (1-p_1-p_2-\dots-p_k) + \frac{1}{2} \sum_{i=1}^k \lambda_i X_i^2}{(1-p_1-p_2-\dots-p_k)(1-p_1-p_2-\dots-p_{k-1})}$$

Как писалось выше, - второй момент длительности обслуживания равен $\bar{X}_i^2 = \frac{2}{\mu^2}$ для системы M/M/1.

Задержка коммуникационной среды вычисляются исходя из законов физики. Среду передачи можно разделить на 2 типа кабелей – медный и оптический. Задержка, с которой информация передается в проводах ограничена соответственно скоростью света (в 300 тыс. км/с), что и является предельной скоростью передачи информации. Рассчитаем минимальную задержку в кабеле из расчета задержки на 1 метр:

$$t = \frac{1}{c} = \frac{1}{300000000} \approx 3,33 \cdot 10^{-9} = 3,33 \text{ нс.}$$

Соответственно – минимальная задержка каждого метра кабеля дает задержку в 3.33 наносекунды. Дополнительно в задержках среды можно учитывать ошибки, которые могут произойти. Пусть $P_{\text{ош}}$ – вероятность ошибки, $T_{\text{обр}}$ – время обработки ошибки, которое будет затрачено на обработку ошибочной ситуации. В этом случае время, которое будет потеряно на ошибке передачи заявки $T_{\text{ош}}$ будет

$$T_{\text{ош}} = P_{\text{ош}} T_{\text{обр}}$$

Вероятность возникновения ошибки зависит от физических характеристик коммуникационной среды и в данной работе не рассматривается. Итоговая формула для задержки среды для кабеля, длиной L , рассчитывается по формуле:

$$T_{\text{сред}} = Lt + T_{\text{ош}}$$

Что касается блока сортировки заявок, – общая задержка i -й заявки $T_{\text{сорт}}^i$ считается довольно просто, учитывая, что приемная часть не может задерживать приходящую информацию, ввиду отсутствия очереди, поэтому – в базовом случае – расчет задержки представляется простым значением в зависимости от количества тактов, потраченных на обработку очередной заявки. Задержка в выходном буфере $T_{\text{вых}}$ рассчитывается как СМО M/M/1, которая выводится из описанных выше формул для конвейерной системы. В этом случае интенсивность поступления заявок сохраняется равной изначальной λ , а μ интенсивности обработки задается извне системой, которая забирает данные из очереди. Подводя итог, можно сказать, что общая задержка $T_{\text{общ}}^i$ любой заявки i в системе будет составляться как сумма полученных задержек при прохождении через систему, а именно как

$$T_{\text{общ}}^i = T_{\text{конв}}^i + T_{\text{сред}} + T_{\text{выбр}}^i + T_{\text{сорт}}^i + T_{\text{вых}}^i$$

Максимальная интенсивность λ_i потока заявок типа i должна быть меньше интенсивности обслуживания каждого блока

$$\lambda_i < \min(\mu_{\text{конв}}^i, \mu_{\text{выбр}}^i, \mu_{\text{вых}}^i),$$

где $\mu_{\text{конв}}$ – интенсивность обработки заявки блока конвейера, $\mu_{\text{выбр}}^i$ – интенсивность обработки заявки блока выбора заявки i , $\mu_{\text{вых}}$ – интенсивность обработки выходной очереди.

Сумма коэффициентов использования дает нам общий коэффициент использования системы, причем

$$p_1 + p_2 + p_3 + \dots + p_n < 1,$$

где $p_1, p_2, p_3 \dots p_n$ – коэффициенты использования системы.

Данная формула позволяет высчитать полную среднюю задержку любой заявки, которая рассчитывается для сетевого контроллера и является временем прохождения заявки через сетевой контроллер, включая очереди.

Размеры буферов (аппаратные затраты на очередь) для заявок i определяется с помощью формулы Литтла

$$N_i = \lambda_i T_i.$$

Получив среднее число заявок в очередях (или в системе) становится просто рассчитать аппаратные затраты для одного или нескольких FIFO. Средняя пропускная способность зависит от типа самой заявки и количества полезной информации, которая она передает через систему. Если предположить, что количество полезной информации, которая передает заявка i , составляет некоторое число бит и равно b_i , то средняя пропускная способность Q_i , характеризующая полезную информацию заявки i , которая передается через нашу модель, вычисляется по формуле

$$Q_i = \frac{\lambda_i b_i}{t_i}$$

где t_i – период времени, за который идет расчет пропускной способности

Одной из важнейших функций является функция управления потоком. Сетевой контроллер представленный на начальном рис. 2 подходит к сетевым контроллерам без ярко выраженного управления потоком на канальном уровне, – например, для сетевых контроллеров по стандарту Ethernet или AFDX. В этих контроллерах изначально интенсивность потока определяется более высоким уровнем и зависит только от интенсивности

обработки каждого блока модели. Однако большинство сетевых контроллеров используют другие формы управления потоком. Самой сложной системой управления потоком (но одной из самых эффективных) является система так называемого кредитования, где, при обмене данными между сторонами, вторая сторона сообщает первой о том, сколько заявок она может принять. Такой способ управлением поток называется кредитованием. Система кредитования разделяется на 2 типа – абсолютную и относительную. При относительном кредитовании каждый сетевой контроллер хранит у себя количество данных, которое он может отправить. Специальный пакет или флит, который приходит от второй стороны увеличивает это число на определенное значение.

В системе с абсолютным кредитованием сообщение о величине свободных мест в приемном буфере сообщается в абсолютных цифрах. Такие сообщения должны регулярно отсылаться либо в составе других заявок, либо самостоятельной заявкой. Приоритет данных заявок может быть любой. В этом случае используется базовая модель для сетевого контроллера на рисунке 2. Дополнительное ограничение, которое накладывается на заявки такого типа – это количество заявок в системе. Количество заявок, для которого используется абсолютное кредитование, не должно быть больше определенного количества, которое было откредитовано. Соответственно для каждой заявки i , для которой используется абсолютное кредитование, необходимо учитывать формулу Литтла, где N_i – количество заявок типа i в СМО, которое может находиться в данный конкретный момент времени в системе. На заявку i накладывается следующее ограничение

$$\begin{cases} \lambda_i = \frac{T_i}{N_i} \\ T_{\text{общ}}^i = T_{\text{конв}}^i + T_{\text{сред}} + T_{\text{выбр}}^i + T_{\text{сорт}}^i + T_{\text{вых}}^i \\ \lambda_i < \min(\mu_{\text{конв}}^i, \mu_{\text{выбр}}^i, \mu_{\text{вых}}^i) \end{cases}$$

Последовательно вычисляются интенсивность λ_i и задержка $T_{\text{общ}}^i$, на основании этого рассчитываются пропускная способность Q_i и общая длина очереди L_i . Система с относительным кредитованием является еще более сложным случаем для работы сетевого контроллера канального уровня. Модифицируем нашу модель под эту систему кредитования (рис. 5).

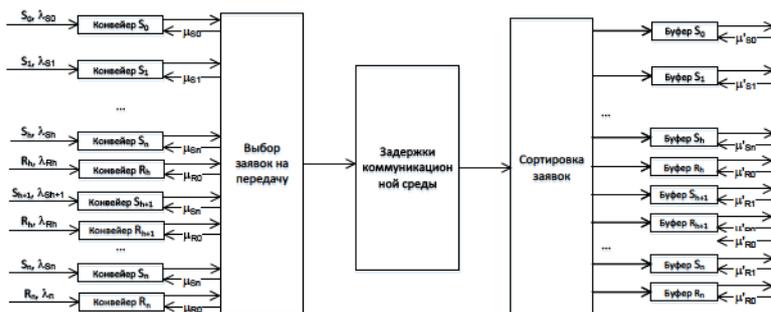


Рис. 5. Модель сетевого контроллера с кредитованием

Часть заявок (на рис. 5 это заявки с $h+1$ до n) используют систему относительного кредитования, где произвольная S_i является самой заявкой, а R_i – разрешающей заявкой на отправку. Построение модели следует следующим дополнительным принципам:

- произвольное разрешение R_i на отправку заявки S_i обладает большим приоритетом, чем сама заявка (однако – может обладать меньшим приоритетом, чем разрешения и заявки другого типа);

- заявка R_i может разрешать отправку сразу n заявок S_i , где n больше 0;

- если заявка S_i требует наличия разрешения R_i , то такая заявка формируется только при наличии разрешения R_i в соответствующем буфере, и поток λ_{Si} формируется в зависимости от наличия разрешений в соответствующем буфере;

- количество разрешений, которые выдает заявка R_i , которое может находиться в системе, ограничено (чаще всего - приемным буфером заявок S_i);

- количество заявок S_i ограничено количеством выданных разрешений заявками R_i ;

- заявка разрешения R_i , считается обработанной после того, как заявка S_i поступила в канал.

Данный метод предполагает обмен информацией между двумя сторонами. Приведем ряд дополнительных ограничений, которые появятся в данной модели. Так как количество заявок ограничено – то для произвольной R_i поток λ_{Ri} будет рассчитываться по формуле Литта как

$$\lambda_{Ri} = \frac{N_{Ri}}{T_{Si}},$$

где N_{Ri} – количество заявок R_i во всей СМО, T_{Si} – задержка передачи заявки S_i по системе. В стационарном режиме в лучшем случае поток λ_{Si} должен быть равен λ_{Ri} . Если же в результате расчетов задержек на основе приведенных формул получится, что $\lambda_{Si} > \lambda_{Ri}$, это означает, что или система кредитования неэффективна и N_{Ri} необходимо увеличить, что предполагает увеличение физического буфера, так как N_{Ri} зависит от этого физического параметра. Так же возможно увеличить время на

обработку заявки S_i , что улучшит аппаратные характеристики устройства например путем снижения внутренней частоты. Если же получится, что $\lambda_{Si} < \lambda_{Ri}$, это означает, что либо задержки обработки заявки S_i слишком велики или же (если уменьшение этих заявок невозможно или не требуется), – возможно сокращение количества заявок R_i , а значит – уменьшение размера буфера N_{Ri} .

Расчет задержек для всех заявок осуществляется по выведенным формулам для $T_{общ}^i$. Размер буфера для заявок без кредитования делает по формуле, выведенными для очередей. Пропускная способность заявок с кредитованием будет определяться из системы для каждого λ_{Si} , причем

$$\left\{ \begin{array}{l} \lambda_{Si} < \min(\mu_{конв}^{Si}, \mu_{выбр}^{Si}, \mu_{вых}^{Si}) \\ \lambda_{Ri} \geq \frac{N_{Ri}}{T_{Si}} = \frac{N_{Si}}{nT_{Si}} \end{array} \right.,$$

что в итоге дает

$$\left\{ \begin{array}{l} \lambda_{Si} < \min(\mu_{конв}^{Si}, \mu_{выбр}^{Si}, \mu_{вых}^{Si}) \\ \frac{\lambda_{Si}}{\lambda_{Ri}} \leq n \end{array} \right.,$$

где n – количество заявок S_i , которое разрешает R_i .

Выводы: построенная имитационная модель позволяет сделать расчет вышеозначенных характеристик, таких как средняя задержка передачи, количество буферных элементов и среднюю пропускную способность устройств канального уровня.

СПИСОК ЛИТЕРАТУРЫ:

1. Клейнрок, Л. Теория массового обслуживания, пер. с англ. – М.: Машиностроение, 1979. 432 с.
2. Jackson, R. Queueing systems with phase type service // Operational Research Quarterly. 1954. Vol. 5. P. 109-120.
3. Burke, P. The output of a queuing system // Operations Research. 1956. Vol. 4. P. 699-704.
4. Вентцель, Е.С. Исследование операций: задачи, принципы, методология. – М.: Наука. Гл. ред. физ.-мат. лит., 1988. 208 с.
5. Бертсекас, Д. Сети передачи данных, пер. с англ. – М.: Мир, 1989. 544 с.

BUILDING A SIMULATION-BASED DATA LINK LAYER CONTROLLER MODEL

© 2017 E.N. Yablokov

St-Petersburg State University of Aerospace Instrumentation

The article is devoted to a preliminary assessment of the characteristics of the data link controllers. The described method allows to make the data link controller model using simulation-based queuing systems.

Key words: *queuing systems, link layer, data controller, point-point*