

## СЕКЦИЯ 3. СИСТЕМЫ И СРЕДСТВА ТЕЛЕКОММУНИКАЦИЙ

### ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ ФУНКЦИОНИРОВАНИЯ ИНТЕГРИРОВАННЫХ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ

А.А. Громыко, В.В. Пытляк  
Беларусь, г. Минск

Интегрированные телекоммуникационные системы важны для разработки как провайдерских, так и корпоративных сетей. Процесс интеграции эволюционен, и первым его этапом становится интеграция сервиса передачи данных с телефонным сервисом. Исторически сложились два подхода к интеграции. Первый – это передача данных по телефонным сетям (технология ISDN) и второй – передача голоса по компьютерным сетям (IP-телефония). ISDN прижился слабо и занимает всего 5% в среде телекоммуникаций. Поэтому предложенное в статье формализованное описание применимо для описания интегрированных систем, построенных с применением второго подхода (VoIP). Уже сейчас технологии передачи речевых пакетов на 50 - 80% дешевле, а в будущем они будут обладать большими возможностями, чем сегодняшние речевые каналы [1].

Технология VoIP характеризуется в первую очередь тем, что в ней был впервые применён новый для речи метод – метод коммутации пакетов, а не каналов, как традиционной телефонии. Поэтому, учитывая коренным образом изменённый подход к передаче речи, важной является задача разработки формализованного описания функционирования такой интегрированной телекоммуникационной системы.

Можно говорить о следующих видах ресурсов в интегрированной телекоммуникационной системе: ресурс производительности сетевых устройств, ресурс буферов памяти, ресурс пропускной способности линии связи. Буфера памяти представляют собой переменную память, объём которой зависит от нагрузки на устройства телекоммуникационной системы. Ресурс же пропускной способности можно нормировать дискретными каналами определённой ширины пропускания, например В-каналами по 64 кбит/с. Ресурс производительности сетевых устройств определяет реактивность системы, которая выражается во времени отклика. Особенность данного ресурса в том, что он непрерывный, и при бесконечном ресурсе памяти буферов может считаться неисчерпаемым. Все эти три ресурса влияют на показатели качества телекоммуникационной системы, которыми являются вероятности явных и условных потерь пакетов. [3]

Специфика показателей качества состоит в том, что они задаются с помощью вероятности превышения определенных значений отказов в обслуживании со стороны сети из-за недостаточности ресурсов памяти и пропускной способности (явные потери) и превышения времени задержки на определенных стадиях обслуживания вызова (условные потери). Для получения названных вероятностей необходимо знать соответствующие функции распределения. На уровне функций распределения аналитического аппарата недостаточно, поэтому необходимо использовать имитационное моделирование.

Для этого необходимо построить формализованное описание телекоммуникационной системы, включающей сетевые устройства с реализованными в них протоколами. С точки зрения функционирования – это формализованное описание сетевого протокола, обеспечивающего сервис, а с точки зрения обслуживания – это формализованное описание сервиса, то есть его номенклатура и услуги. В информационно-вычислительных сетях существует стандарт на описание сервиса – рекомендация ISO 8509 [2], в традиционной телефонии подобных стандартов нет. Зато там есть стандартный язык формализованного описания протоколов SDL, подобного которому в сетях данных нет.

Для протоколов сетей данных рекомендован автоматный способ описания. Вообще существуют два класса спецификации протоколов и сервисов: автоматные модели и модели последовательностей. Для описания протоколов более пригодны первые, для сервисов – вторые. В целом же для описания сложных систем есть общие модели динамики, из которых частично применимы для описания интегрированных телекоммуникационных систем агрегатная модель, сети Петри и СМО. Но они сложны для применения и часто недостаточны.

Таким образом, встаёт вопрос о выборе некоторой альтернативной модели формализованного описания функционирования системы. В качестве такой модели была выбрана транзактная модель. Механизм применения транзактной модели достаточно хорошо известен в системах управления базами данных. Однако у него есть и ещё одна важная задача – управление параллелизмом и распределённой обработкой информации.

Транзакции обладают такими свойствами, как атомарность и вложенность, что очень удобно в нашем случае. Свойство атомарности говорит о том, что транзакция либо выполняется целиком, либо, если не может быть выполнена, то внесённые ей в систему изменения отменяются, будто транзакция не выполнялась. Свойство вложенности позволяет транзакции на данном уровне являться частью другой транзакции более

высокого уровня. Это свойство транзакций удобно для описания сервиса на семи уровнях эталонной модели OSI/ISO.

В нашем случае модель, построенная на использовании транзакций, очень удобна, поскольку она не рассматривает чётко оформленные элементы в программном или аппаратном исполнении, то есть позволяет проводить произвольное разбиение системы по структурным единицам. Таким образом, с помощью данного подхода мы можем обобщённо описывать сервис, и предоставляющие его протоколы. Работа телекоммуникационной системы, по сути своей, заключается в обработке запросов, которые, имеют естественную транзактную природу.

Остальные модели непригодны для описания сервиса протокольных уровней, потому что не в полной мере затрагивают качество функционирования, не говоря уже о качестве обслуживания. Ни одна из них не даёт чёткого описания картины сервиса, потому как для этого описания данных моделей попросту недостаточно. В свою очередь одно только описание сервиса не приспособлено для анализа функционирования интегрированной телекоммуникационной системы. В модели сервисов нет описания ресурсов. И поскольку формализованное описание сервиса говорит только об услугах, то чтобы иметь возможность количественного описания, необходимо говорить о формализованном описании функционирования. А задать его это можно только с помощью параметризуемых элементов.

Таким образом, целесообразно совместить богатство описания сервиса (с его примитивами, определёнными стандартом [2] и позволяющими извлекать из них необходимые параметры элементов описания функционирования) и собственно само описание функционирования в одно общее формализованное описание интегрированной телекоммуникационной системы.

Оно может быть многоуровневым, поскольку наш моделируемый «чёрный ящик» представляет собой всю сеть, а не отдельные сетевые устройства. Первый уровень спецификации – стандарт ISO на описание сервиса. Второй уровень спецификации – переход к алгоритму описания функционирования. Как видно из изложенного выше, для этой цели наиболее подходит транзактный метод формализованного описания.

Путём уточнения деталей, а точнее благодаря описанию сервиса на разных уровнях, мы сужаем «чёрный ящик» до тех пор, пока он сам, наконец, сможет быть описан с помощью параметризуемых элементов, а из характеристик самой системы удастся получить реальные параметры этих элементов формализованного описания. Таким образом, мы заменяем элементы описания сервиса этими параметризуемыми элементами,

после чего появляется возможность моделировать систему с применением данных элементов.

К характеристикам телекоммуникационной системы в данном случае относятся её ресурсы, которые упоминались ранее (производительность, буфера памяти, пропускная способность), а также данные, полученные с использованием средств мониторинга, и технические данные и показатели, характеризующие сетевые устройства, входящие в систему. Многие из этих характеристик сетевых устройств можно найти в их технических паспортах.

Описание процесса функционирования модели должно включать описание прохождения отдельной заявки в программе по сетевому устройству в соответствии с последовательной цепочкой обслуживания (транзакцией) заявок данного типа, и совместное обслуживание заявок.

В соответствии с принятым подходом обслуживание заявки состоит в выполнении ряда последовательных процессов, задаваемых цепочкой обслуживания. Транзакция состоит из элементов нескольких типов, различные варианты перестановок из которых позволяют задавать схемы обслуживания для любой из транзакций. Количество элементов в ней может быть произвольным, количество типов элементов - ограничено. Охарактеризуем типы элементов транзакции и параметры процессов, задаваемых ими [3, 4].

**"ФУНКЦИОНАЛЬНАЯ ПРОГРАММА"** (ФП). Использование данного элемента наиболее предпочтительно в таком варианте, когда ему соответствует реальная программа по сетевому устройству, подвергающаяся диспетчеризации на одном из процессоров системы. До перехода на выполнение этого элемента цепочки обслуживания каким-то образом должен быть задан номер процессора, на котором программа должна выполняться. Выполнение ФП заключается в занятии процессора на время, являющееся характеристикой программы и процессора. При приоритетной дисциплине диспетчирования ФП имеет ещё один параметр - приоритет, каждое назначение ФП на процессор можно считать актом диспетчирования, поэтому в модели в этом месте производится учёт затрат на диспетчирование.

**"ОБМЕН"** (0). Данный элемент цепочки означает, что необходимо произвести групповой обмен между оперативным запоминающим устройством процессора, к которому в данный момент приписана заявка, и устройством, являющимся внешним по отношению к этому процессору. Такой обмен непосредственно не затрагивает ресурса процессора. Но опосредованное влияние все же есть - через ресурсы памяти и связи, совместно используемые с процессором. Степень этого влияния зависит от

времени обмена, характера обмена, количества одновременно проводимых обменов. В статике (в исходных данных) необходимо задать для элементов типа 0 такие параметры, как время обмена и характер обмена.

Характер обмена задается как коэффициент замедления  $K_i^{\xi}$ , где  $i$  - тип обмена. Выполнение элемента 0 для одиночной заявки означает задержку её выполнения на какую-то величину в течение всего времени обмена, время обмена  $T_i^{обм} = t_i * l$ , где  $l$  - количество передаваемых единиц информации,  $t_i$  - скорость передачи, определяемая типом внешнего устройства.  $t_i$  является известной характеристикой внешнего устройства, и может задаваться функцией распределения. Таким образом,  $T_i^{обм}$  может задаваться функцией, имеющей такое же распределение, что и  $l$ .

"ЦИКЛ" (Ц). Данный элемент сам по себе ресурс процессора не использует. Его назначением является повторение уже пройденного участка цепочки обслуживания заданное число раз. Параметры циклов могут быть заданы априори и определению в ходе вычислительного процессора не подлежат. Кроме числа повторений параметром элемента Ц является номер элемента, с которого он берет начало.

"РАЗМНОЖЕНИЕ" (РАЗМ.) Элемент означает, что определенный участок последовательной транзакции превращается в некоторое множество параллельных идентичных процессов. Параметр элемента: количество размноженных процессов.

"СЛИЯНИЕ" (СЛ) Состоит в том, что несколько последовательно поступающих сообщений объединяются в одно. Параметр элемента - количество объединяемых сообщений.

"ЗАДЕРЖКА" (ЗА,ЗП). Ресурс процессора этим элементом также не потребляется. Элемент позволяет учитывать задержки, как прогнозируемые алгоритмом обслуживания, так и задержки, связанные с принятой в сетевых устройствах организацией вычислительного процесса. Что означают последние? Сетевые устройства по преобладающему в них способу организации вычислительных процессов можно отнести к одному из двух классов - циклическим либо асинхронным. Задание схемы обслуживания только с помощью перечисленных выше элементов позволяет описывать только асинхронные сетевые устройства: в цепочках обслуживания нет элемента, позволяющего искусственно упорядочивать заявки во времени. Таким элементом может быть "ЗАДЕРЖКА". При этом возможны два варианта задержки: задержка активная (ЗА) и задержка пассивная (ЗП). Для элемента ЗА определено время, через кото-

рое заявка выйдет из состояния задержки по отношению к моменту входа в неё. Для ЗП момент выхода из состояния задержки определяется посторонним источником, например, первым сигналом таймера. Итак, для ЗА задается интервал времени, для ЗП - имя (номер) источника, первый же сигнал из которого заканчивает процесс задержки.

"ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР" (ЦПР). Элемент не требует ресурса процессора. Он означает, что следующий за ним участок цепочки будет относиться к одиночному по фазе процессору.

"ПЕРИФЕРИЙНЫЙ ПРОЦЕССОР" (ППРj). Элемент означает, что следующий за ним участок цепочки относится к одному из процессоров j - ой группы.

Заявки, требующие ресурсов производительности различных процессоров, обслуживаются параллельно. Возможна одновременная реализация процессов, относящихся к одному процессору и нескольким заявкам. На процессоре может выполняться функциональная программа, относящаяся к одной заявке, и в этом же процессоре параллельно с ней могут выполняться процессы типа "ОБМЕН" и "ЗАДЕРЖКА" для других заявок. "ЗАДЕРЖКА" не влияет на ход выполняемой на процессоре программы. "ОБМЕН" оказывает замедляющее влияние. Итак, функционирование сетевого устройства на данном уровне рассматривается как параллельное исполнение цепочек обслуживания фаз с учетом ограничений, накладываемых на параллельно исполняемые процессы, и их взаимного влияния.

Предложенное формализованное описание интегрированной телекоммуникационной системы может использоваться как на этапе индивидуального проектирования (для провайдеров сетей), так и для построения имитационных моделей для поиска узких мест в реальных системах или для изучения их поведения при изменении структуры системы.

## Литература

1. Cisco Systems и др. Руководство по технологиям объединённых сетей, 3-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 1040 с.: ил. – парал. тит. англ.
2. ISO/IS 8509. Information Processing Systems. Open System Interconnection. Basic reference model.
3. Громыко А.А., Пытляк В.В. Разработка показателей оценки и формализованного описания функционирования мультисервисной телекоммуникационной системы с интеграцией речи и данных, Труды международной конференции "IST'2004". – Минск, 2004.
4. Пытляк В.В. Метод и средства оценки характеристик многопроцессорных управляющих комплексов коммутационных узлов. – Минск.: БГУ, 1990. – 20 с.