

*Е.Б.Замятина, Г.А.Мальцев, С.Порязов, Э.Саранова*¹

Национальный исследовательский университет «Высшая школа экономики», ezamyatina@hse.ru,

Пермский государственный национальный исследовательский университет,

ОПЫТ МОДЕЛИРОВАНИЯ АЛГОРИТМОВ МАРШРУТИЗАЦИИ РАЗАБОТАННЫХ ДЛЯ SDN И SON СЕТЕЙ

Введение

В настоящее время можно заметить рост объемов трафика и изменение его структуры. Причиной тому явилось: а) появление огромного количества мобильных пользователей; б) развитие новых технологий (грид-технологии, облачные технологии, Интернет вещей). Следствием развития новых технологий явилась, по замечанию [1], необходимость в формировании высокопроизводительных кластеров для обработки Больших Данных, а также, хорошо масштабируемых виртуализированных сред для предоставления облачных сервисов. Получилось так, что сдерживающим фактором развития вычислительной инфраструктуры стали компьютерные сети, поддерживающие традиционные протоколы и использующие традиционные, и уже устаревшие, среды управления.

Так, например, отмечают препятствующее масштабированию ограничение на число логических групп [1] сетей. Известно, что стандартная технология VLAN обеспечивает поддержку 4096 виртуальных локальных сетей. Однако развертывание облачных сервисов IaaS для коммерческих ЦОД требует большего числа виртуальных сетей. (Таким образом, если услуги IaaS предоставить 100 пользователям, и у каждого есть 100 VLAN, то число логических сетей составляет уже 10 тыс.)

Таким образом, возникает необходимость в развитии технических средств поддержки передачи сетевых трафиков. На сегодняшний день сетевые устройства пополнились SDN и SON сетями. Более подробно

¹ Работа выполнена при финансовой поддержке РФФИ: грант № 16 - 47 -230336 и грант Администрации Краснодарского края
© Замятина Е.Б., Мальцев Г.А., Порязов С., Саранова Э., 2016

рассмотрим их и укажем на преимущества их применения. Однако сделаем это несколько позже.

Так или иначе, возникает необходимость в тщательном проектировании программно-конфигурируемых сетей. Одним из методов, используемых при проектировании, является метод имитационного моделирования.

Далее рассмотрим подробнее программно-конфигурируемые сети, алгоритм маршрутизации SBARC и представим имитационную модель этого алгоритма, описанную с помощью языка имитационного моделирования Triad. Таким образом, покажем возможности систем моделирования Triad для исследования программно-конфигурируемых сетей.

Программно-конфигурируемые сети

SDN-сети используют в сенсорных сетях, а сенсорные сети, в свою очередь, могут быть использованы во многих прикладных областях, таких как:

- системы обороны и обеспечение безопасности;
- контроль окружающей среды;
- мониторинг промышленного оборудования.

Основными преимуществами и особенностями таких сетей являются:

- большие масштабы с большой плотностью размещения узлов в пространстве;
- высокая отказоустойчивость и надежность системы (достигаются за счет большого количества различных маршрутов доставки данных);
- устойчивость к изменениям топологии сети;
- самоорганизация узлов.

Разберемся, что такое SDN-сеть. Итак, SDN (Software-defined Networking) сеть — это программно-конфигурируемая сеть, т.е. сеть передачи данных, в которой уровень управления сетью отделён от устройств передачи данных и реализуется программно. Это одна из форм виртуализации вычислительных ресурсов [2].

Суть концепции SDN заключается в разделении функций управления и пересылки данных, передаче функций маршрутизации контроллеру сети, и реализации на основе этого принципа легко масштабируемой, быстро и гибко настраиваемой виртуальной сети.

Разделение уровней управления (Control Plane) и коммутации (Forwarding Plane) избавляет коммутаторы и маршрутизаторы от

значительной доли вычислительной нагрузки. Все, что от них теперь требуется, это максимально быстро пересылать пакеты из одного порта в другой, согласно таблице маршрутизации, поступающей извне от контроллера сети. Контроллер же, вместо построения маршрута для каждого пакета, как это предусмотрено в традиционной схеме, принимает решение однократно, а потом пересылает все однотипные пакеты потоком по готовому маршруту, пока не изменится состояние сети или характер трафика[1].

Наряду с разделением процессов передачи и управления данными поддерживаются: а) централизация управления сетью при помощи унифицированных программных средств; б) виртуализация физических сетевых ресурсов. Протокол OpenFlow, реализующий независимый от производителя интерфейс между логическим контроллером сети и сетевым транспортом, является одной из реализаций концепции программно-конфигурируемой сети.

В архитектуре программно-конфигурируемой сети выделяется три уровня:

- инфраструктурный уровень, на котором функционируют сетевые коммутаторы и каналы передачи данных;
- уровень управления — набор программных средств, физически отделённых от инфраструктурного уровня, обеспечивающий реализацию механизмов управления устройствами инфраструктурного уровня;
- уровень сетевых приложений.

Ядром уровня управления программно-конфигурируемой сетью является сетевая операционная система — программное средство, обеспечивающее, с одной стороны, интерфейс со средствами инфраструктурного уровня (например, динамическое изменение таблиц маршрутизации), и, с другой стороны, прикладной программный интерфейс для уровня сетевых приложений. Этот интерфейс сформулирован в терминах более высокого уровня абстракции (например, «имя узла», «имя пользователя») в противовес тому, что используется в параметрах конфигурации сетевых устройств (IP-адрес, маска подсети, MAC-адрес).

Беспроводные сети связи разделяются на централизованные инфраструктуры и самоорганизующиеся. Отличительной особенностью самоорганизующихся сетей SON (self-organization) является ее способность в отсутствие централизованной инфраструктуры обмениваться данными любой паре находящихся в зоне радиопокрытия узлов сети. Узлы в SON могут быть одновременно конечными хостами и маршрутизаторами. Соединение организуется на

длинные расстояния с помощью специализированных протоколов маршрутизации в промежуточных узлах-маршрутизаторах. Такое соединение называют «многоэтапным или многошаговым» (multihop). Этапом является участие в этом соединении одного узла-маршрутизатора. Узлы этих сетей обладают способностью сами находить друг друга и формировать сеть, а в случае выхода из строя какого-либо узла могут устанавливать новые маршруты для передачи сообщений[3,4].

Алгоритм маршрутизации SBARC

Рассмотрим пример реализации функций маршрутизации в программно-конфигурируемых сетях. Рассмотрим более подробно алгоритм маршрутизации SBARC, описанный в [3] и успешно применяемый в таких сетях.

В отличие от других алгоритмов маршрутизации, SBARC делит вычислительные узлы (пиры) на суперузлы и простые узлы. Большая часть рабочей нагрузки ложится на суперузлы. Кроме того, в SBARC а) данные маршрутизации кэшируются для дальнейшего снижения стоимости маршрутизации; б) схема кэширования обеспечивает эффективное использование свободного дискового пространства.

Известно, что в традиционной C / S архитектуре функциональные возможности клиентов и серверов строго разделены, клиенты не могут нести нагрузку серверов даже в том случае, когда серверы перегружены. В P2P архитектуре напротив нагрузки системы равномерно распределены по всем вычислительным узлам без учета разнородных ресурсов вычислительных узлов. На самом же деле такие характеристики вычислительных узлов как пропускная способность сети, вычислительная мощность и память существенно различаются.

Алгоритм SBARC различает вычислительные узлы сети: вычислительная нагрузка на узлах не является ни равномерно распределенной на всех узлах, ни сосредоточенной на небольшом числе централизованных серверов. Суперузлы берут на себя большую часть нагрузки (в основном выполняя системную работу), простые действуют в основном как клиенты, но четкого распределения ролей нет (как в компьютерных системах с архитектурой клиент-сервер). Суперузлы должны соответствовать определенным критериям: у вычислительного узла должно быть широкополосное подключение, достаточно высокая вычислительная мощность, узел не должен очень часто присоединяться и отсоединяться, должен иметь достаточно большую память. Из простых топологически смежных узлов и ряда суперузлов (вторичных) организуются домены. У домена есть главный

суперузел, он хранит информацию об узлах домена и копии их таблиц маршрутизации, тем самым управление маршрутизацией переходит суперузлам. Более подробно об алгоритме SBARC можно прочитать в [3]. Теперь представим имитационную модель алгоритма SBARC на языке имитационного моделирования TriadNS. Кратко система имитационного моделирования и автоматизированного проектирования TriadNS описана ниже.

Система имитационного моделирования TriadNS

Система имитационного моделирования TradNS предназначена для проектирования и моделирования компьютерных сетей, а также их исследования[5]. СИМ TriadNS является развитием системы автоматизированного проектирования Triad[6,7], которая была разработана на кафедре математического обеспечения вычислительных систем Пермского государственного университета и предназначалась для проектирования встроенных вычислительных систем. В 2003 году появилась новая версия системы, написанная на C#(Triad.Net). В последующие год система TriadNS усовершенствовалась, о чем свидетельствуют публикации, например [8,9].

В настоящее время существует достаточное большое количество симуляторов компьютерных сетей, в том числе: COMNET[10], OPNET[11], OMNeT++[12]. Все они обладают привлекательными чертами, которые позволяют выполнять автоматизированное проектирование компьютерных сетей, получать временные оценки алгоритмов.

Так COMNET III – объектно-ориентированная система моделирования локальных и глобальных сетей. Позволяет моделировать уровни: приложений, транспортный, сетевой, каналный. Использует все известные на сегодня технологии и протоколы, а также системы клиент-сервер. Модель создается из объектов, “строительных блоков”, с COMNET поставляется большая библиотека объектов-моделей реального сетевого оборудования и методов доступа к среде.

OPNET – также представляет собой программное средство для проектирования и моделирования локальных и глобальных сетей, компьютерных систем, приложений и распределенных систем. Включает следующие программные продукты: Netbiz (проектирование и оптимизация вычислительной системы), Modeler (моделирование и анализ производительности сетей, компьютерных систем, приложений и распределенных систем), ITGuru (оценка производительности коммуникационных сетей и распределенных систем).

OMNeT++ представляет собой симулятор дискретных событий, которые происходят внутри простых модулей (simple modules). В системе OMNeT++ заложена детальная реализация протоколов, начиная с сетевого уровня, возможность написания и подключения собственных модулей, развитый графический интерфейс.

Можно сделать заключение, что описанные выше программные средства для проектирования, анализа и исследования компьютерных сетей являются в низкоуровневыми.

В TriadNS с помощью специально разработанного высокоуровневого языка моделирования Triad можно описать новое устройство, новый протокол или алгоритм (например, алгоритм маршрутизации). Именно попытка описать и исследовать алгоритм маршрутизации SBARC и является целью написания настоящей статьи.

Итак, СИМ TriadNS представляет собой совокупность лингвистических и программных средств имитационного моделирования и включает следующие компоненты: компилятор, ядро, графический редактор, подсистему отладки, подсистему валидации, подсистему синхронизации распределенных объектов модели, подсистему балансировки (распределенная версия), подсистему организации удаленного доступа, подсистему защиты от внешних и внутренних угроз, подсистему автоматического доопределения модели.

Назначение каждого из компонентов представлено ниже: TriadCompile (компилятор языка Triad, переводит описание имитационной модели с языка Triad во внутреннее представление); TriadDebugger (отладчик, использует механизм информационных процедур алгоритма исследования, локализует ошибки и вырабатывает рекомендации для их устранения на основании правил из базы данных, для каждого класса ошибок осуществляется поиск по онтологии соответствующего обработчика ошибок); TriadCore (ядро системы, включает библиотеки классов основных элементов модели), TriadEditor (редактор моделей, предназначен для работы с моделью как в удаленном, так и локальном режимах, локальный режим предполагает работу с системой в том случае, если нет удаленного доступа), TriadBalance (подсистема балансировки), TriadSecurity (подсистема безопасности, этот компонент используют при удаленном доступе к системе моделирования), TriadBuilder (подсистема автоматического доопределения частично описанной модели), база данных, где хранятся экземпляры элементов модели, TriadMining набор процедур для исследования результатов модели методами DataMining, TriadRule — алгоритм синхронизации объектов распределенной модели,

использующей для вычислительного эксперимента ресурсы нескольких вычислительных узлов.

Описание модели в системе TriadNS можно определить следующим образом: $M = (STR, ROUT, MES)$, где STR — слой структур, ROUT — слой рутин, MES — слой сообщений [6].

Слой структур представляет собой совокупность объектов, взаимодействующих друг с другом посредством посылки сообщений. Каждый объект имеет полюсы (входные и выходные), которые служат соответственно для приема и передачи сообщений. Основа представления слоя структур — графы. В качестве вершин графа следует рассматривать отдельные объекты. Дуги графа определяют связи между объектами.

Объекты действуют по определенному алгоритму, который описывают с помощью рутины. Рутинa представляет собой последовательность событий e_i , планирующих друг друга (E — множество событий; множество событий рутины является частично упорядоченным в модельном времени).

Выполнение события сопровождается изменением состояния объекта. Состояние объекта определяется значениями переменных рутины. Таким образом, система имитации является событийно-ориентированной.

Рутинa так же, как и объект, имеет входные и выходные полюса. Входные полюса служат соответственно для приема сообщений, выходные полюса — для их передачи. В множестве событий рутины выделено входное событие. Все входные полюса рутины обрабатываются входным событием. Обработка выходных полюсов осуществляется остальными событиями рутины.

Для передачи сообщения служит специальный оператор out (*out* <сообщение> *through* <имя полюса>). Совокупность рутин определяет слой рутин ROUT.

Слой сообщений (MES) предназначен для описания сообщений сложной структуры. Система Triad реализована таким образом, что пользователю необязательно описывать все слои. Так, если возникает необходимость в исследовании структурных особенностей модели, то можно описать только слой структур.

В слое структур определены стандартные процедуры, с помощью которых можно определить множество вершин графа, множество ребер, дуг и т.д., найти кратчайшее расстояние между двумя вершинами, компоненты связности $GetStronglyConnectedComponents(G)$, выделение слоя из структур модели $GetGraphWithoutRoutines(M)$.

В системе Triad.Net для анализа функционирования компьютерной сети можно использовать стандартные и пользовательские информационные процедуры. Пользовательские информационные процедуры описывают на языке Triad.

Для каждого элемента сети можно указать список необходимых информационных процедур, которые будут вести наблюдение во время моделирования за переменными, событиями и полюсами элемента. Система также предоставляет лингвистические средства для создания собственных условий моделирования, в которых можно описывать оригинальные алгоритмы сбора статистики и алгоритмы преобразования модели в динамике.

В системе TriadNS с помощью графического редактора пользователь описывает имитационную модель, данные о модели подаются на вход подсистемы построения модели. Используя информацию о структуре компьютерной сети, построенной с помощью графического редактора, а также, информацию о наложенных рутинах и подключенных информационных процедурах, подсистема построения модели получает внутреннее представление модели. При этом создаются все объекты, которые непосредственно участвуют в моделировании, динамически подключаются сборки с используемыми рутинными, информационными процедурами и условиями моделирования. Подсистема выполнения производит запуск модели на выполнение. Также в процессе моделирования эта подсистема отвечает за передачу сообщений между объектами системы.

Подсистема сбора статистики, взаимодействуя с подсистемой выполнения модели, получает уведомления обо всех изменениях состояний указанных объектов модели. После завершения моделирования все данные, полученные подсистемой сбора статистики, передаются в графический редактор и отображаются на форме для пользователя.

Реализация алгоритма SBARC в TriadNS

Итак, для реализации алгоритма маршрутизации SBARC была выбрана система моделирования TriadNS.

В симуляторе TriadNS можно использовать онтологии. Известно, что онтологии дают возможность разделения и повторного использования знаний.

Разработанная онтология предметной области импортирует онтологию понятий системы Triad.Net как базовую. Кроме того, онтология системы TriadNS дополняет базовую онтологию. Введены

специализированные для области компьютерных сетей подклассы основных классов базовой онтологии.

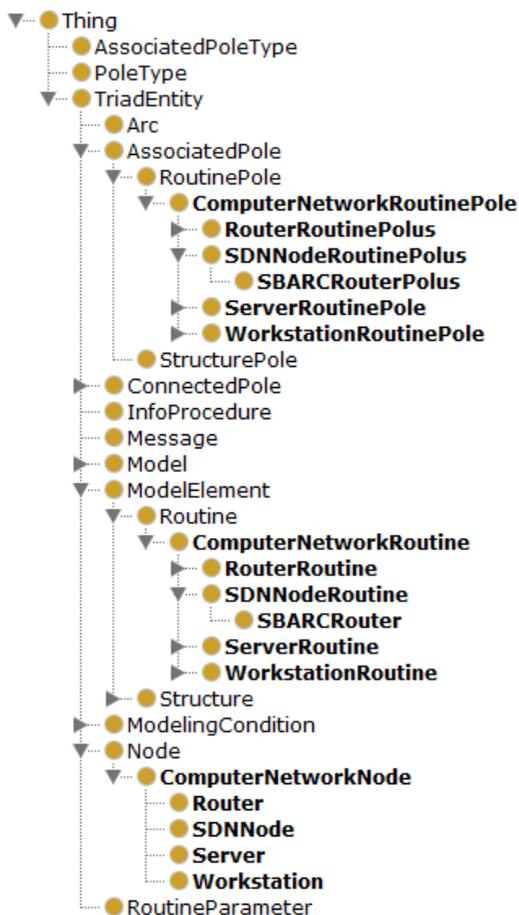


Рисунок 1. Онтология, дополненная классами для реализации алгоритма маршрутизации SBARC

Элементы сети загружаются из онтологии предметной области, это все подклассы класса ComputerNetworkNode. Следовательно, для реализации алгоритма SBARC необходимо добавить подкласс «Узел» (SDNNode), описывающий элемент сети, в таком виде, в котором он необходим для реализации алгоритма.

Использование онтологий для представления знаний предметной области также дает нам возможность легко ориентировать систему на другие области применения имитационного моделирования.

С помощью графического редактора построена структура модели компьютерной сети (см. рис.2).

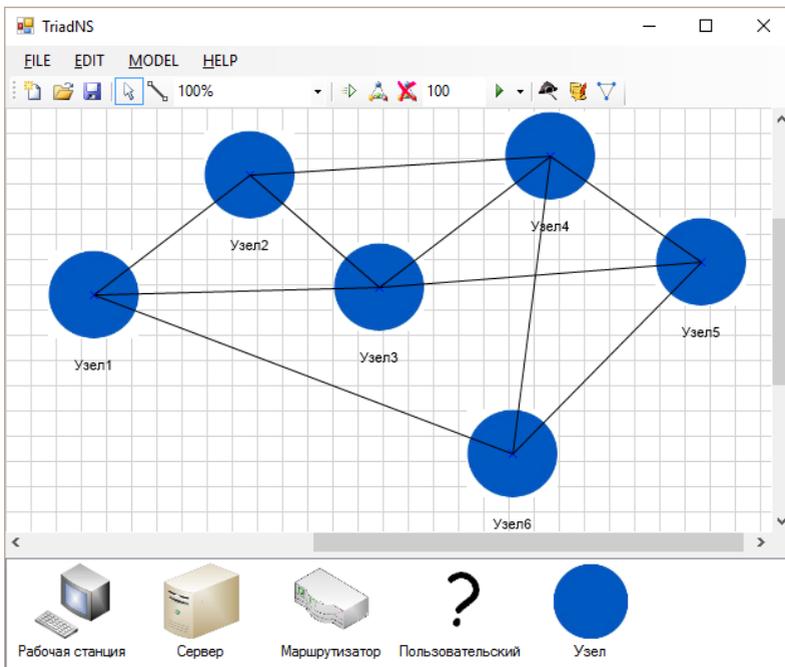


Рисунок 2. Пример сети составленной из узлов поддерживающих маршрутизацию по алгоритму SBARC

В СИМ TriadNS при помощи онтологий решается задача автоматического доопределения, которая состоит в том, чтобы на каждую вершину с неопределенным поведением наложить какую-либо подходящую рутину. Работа алгоритма доопределения начинается перед началом моделирования для того, чтобы автоматически доопределить все элементы с неопределенным поведением, либо после выбора пользователем конкретной рутины для элемента (для проверки возможности наложения рутины). Во втором случае пропускается этап поиска всех экземпляров рутин соответствующего семантического типа в онтологии. То есть, элементы сети «Узел» могут быть

соединены дугами только с элементами такого же типа. А в качестве рутин будет выбран «Алгоритм маршрутизации SBARCRouter».

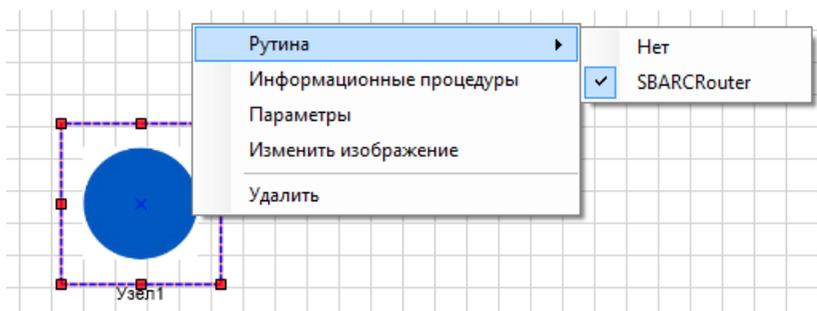


Рисунок 3. Элемент SDN-сети «Узел»

После перемещения элементов на рабочую область они имеют только определенный семантический тип («Узел») (см. рис.3), поведение у этого элемента не определено.

Поведение элементу можно задать через контекстное меню, все возможные поведения определяются из онтологии. Туда попадают все экземпляры рутин данного семантического типа. Каждая из рутин имеет некоторое количество параметров, которые пользователь может изменять после наложения рутины на элемент.

Далее необходимо реализовать механизм по выбору суперузлов, а так же разработать рутину на языке Triad, описывающую сам алгоритм маршрутизации в зависимости от типа узла (обычный узел или суперузел) (SBARCRouter). Работа алгоритма маршрутизации представлена в виде блок-схем на рисунке 4. Рутин SBARCRouter посылает сообщения случайно выбранному узлу в сети. Полюса рутины позволяют соединять эту рутину с другими узлами. Определены несколько событий, такие как: а) посылка сообщения случайно выбранному узлу; б) посылка ответного сообщения на запрос суперузла, выполняющего функции маршрутизатора. Входное сообщение в зависимости от текущего режима работы и от вида узла может рассматриваться как параметры соседнего узла (идентификатор, вид узла и производительность (если сообщение от суперузла)), сообщения от других узлов.

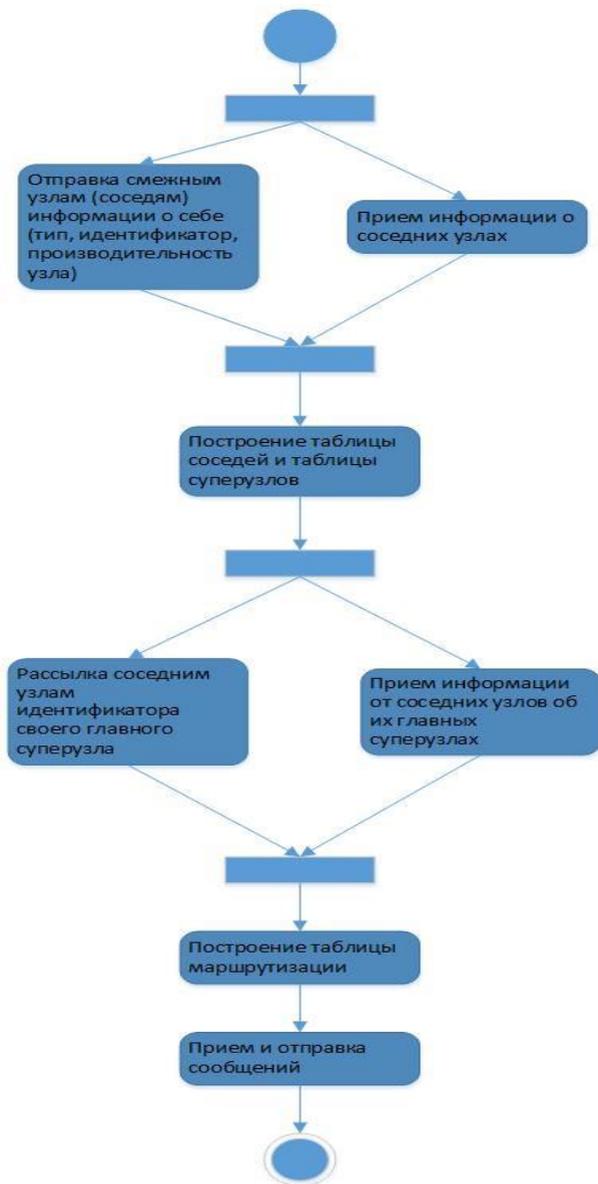


Рисунок 4. Работа узла по рутине SBARCRouter (алгоритм маршрутизации SBARC)

Имеется также несколько других параметров:

- два параметра (ST11 и ST12), влияют на время обработки сообщения ($T1 + \text{Random}(T2)$);
- два параметра (ST21 и ST22), влияющие на частоту посылки сообщений узлам;
- SQueueLen — параметр, который определяет размер буфера;
- SBBCon — параметр, определяющий имеется ли широкополосное подключение;
- STFlops — параметр, который указывает значение производительности или вычислительной мощности узла и измеряется во флопсах (количество операций с плавающей запятой в секунду);
- SNodeID — идентификатор узла;
- IDMessage — ключ (идентификатор узла), по которому будут отсылаться сообщения.

Узлы в сети логически сгруппированы на основе их идентификаторов. Принадлежность той или иной группе определяется идентификатором SNodeID. Узлы расположенные ближе друг к другу имеют идентификаторы с более близкими значениями. Каждый узел знает адрес хотя бы одного узла в других группах и все адреса внутренней группы (рис. 5).

После того как определено поведение всех элементов можно запустить процесс моделирования. Используя информацию о структуре компьютерной сети, построенной с помощью графического редактора, о наложенных рутинах и подключенных информационных процедурах строится внутреннее представление модели. При этом динамически подключаются сборки с используемыми рутинами, информационными процедурами и условиями моделирования, создаются соответствующие объекты.

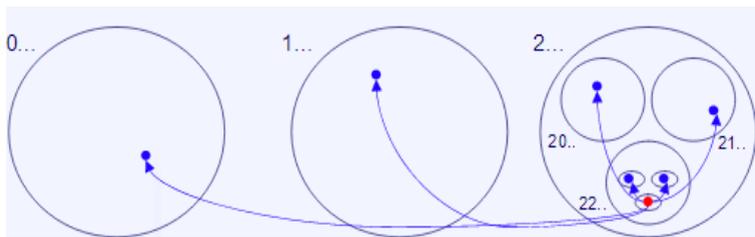


Рисунок 5. Группы узлов в сети

После того как построено внутренне представление модели, запускается имитационный процесс. По окончании процесса моделирования появится форма с результатами моделирования, в данном случае форма представляет собой трассировку сообщений при выполнении алгоритма SBARC (рис. 6).

Time	Element	Message
16,5654	Узел2	0102 шлёт узлу 0103 сообщение: '27' для узла 0103
16,5654	Узел3	0103 получил сообщение '0102 0103 27 13'
16,5654	Узел3	Сообщение: '27' от узла(0102) доставлено по назначению!
18,059	Узел3	0103 шлёт узлу 0101 сообщение: '983' для узла 0101
18,059	Узел1	0101 получил сообщение '0103 0101 983 13'
18,059	Узел1	Сообщение: '983' от узла(0103) доставлено по назначению!
18,0886	Узел1	0101 шлёт узлу 0102 сообщение: '769' для узла 0102
18,0886	Узел2	0102 получил сообщение '0101 0102 769 13'
18,0886	Узел2	Сообщение: '769' от узла(0101) доставлено по назначению!
26,5654	Узел2	0102 хочет отправить узлу(0103) сообщение: '462'
28,059	Узел3	0103 хочет отправить узлу(0101) сообщение: '395'
28,0886	Узел1	0101 хочет отправить узлу(0102) сообщение: '305'
30,0017	Узел2	0102 шлёт узлу 0103 сообщение: '462' для узла 0103
30,0017	Узел3	0103 получил сообщение '0102 0103 462 26,5654'
30,0017	Узел3	Сообщение: '462' от узла(0102) доставлено по назначению!
34,5583	Узел1	0101 шлёт узлу 0102 сообщение: '305' для узла 0102

Рисунок 6. Трассировка сообщений при выполнении алгоритма SBARC.

Заключение

Итак, в статье кратко представлены компьютерные сети нового поколения: программно-конфигурируемые сети (SDN и SON-сети).

Эти сети позволяют преодолеть технические сложности, которые в настоящее время возникают при передаче существенно возросшего трафика в связи с развитием новых информационных технологий (мобильные пользователи, облачные вычисления).

Авторы используют высокоуровневый язык Triad для описания алгоритма маршрутизации SBARC, а затем выполняют анализ этого алгоритма с использованием подсистемы сбора и обработка статистики, которая включает специальные языковые и программные средства, а именно, информационные процедуры и условия моделирования. При описании алгоритма маршрутизации были

использованы онтологии, позволяющие гибко настроиться на конкретную предметную область.

Проведенные исследования показали жизнеспособность системы моделирования TriadNS.

Библиографический список

1. Барсков А. SDN: кому и зачем это надо? [Электронный ре-сурс] // Открытые системы: Журнал сетевых решений/LAN. URL: <http://www.osp.ru/lan/2012/12/13033012/> (дата обращения: 10.10.16)
2. Смелянский Р.И. Программно-конфигурируемые сети. [Электронный ресурс] // Открытые системы: Журнал сетевых решений/LAN. URL: <http://www.osp.ru/os/2012/09/13032491/> (дата обращения: 10.10.16)
3. Zhiyong X. and Yiming H. SBARC: A Supernode Based Peer-to-Peer File Sharing System. // Department of Electrical & Computer Engineering and Computer Science University of Cincinnati, Cincinnati, OH 45221-0030.
4. Druschel P. and Rowstron A. *Past: A large-scale, persistent peer-to-peer storage utility*. May 2001.
5. Mikov A., Zamiatina E. Program Tools and Language for Networks Simulation and Analysis. /Mikov A./Proceedings SDN & NFV – The Next Generation of Computational Infrastructure: 2014 International Science and Technology Conference «Modern Networking Technologies (MoNeTec)», October 27-29, 2014 Lomonosov Moscow State University pp. 94-102
6. Mikov A.I. Simulation and Design of Hardware and Software with Triad // Proc.2nd Intl. Conf. on Electronic Hardware Description Languages. Las Vegas, USA. 1995. P. 15-20.
7. Миков А.И. Автоматизация синтеза микропроцессорных управляющих систем — Иркутск: издательство Иркут. ун-та, 1987.
8. Zamyatina E., Mikov A, Mikheev R. Linguistic and Program Tools For Debugging and Testing Of Simulation Models Of Computer Networks // International Journal "Information Models and Analyses". 2013. Vol. 2. No. 1. P. 70-80.
9. Миков А.И., Замятина Е.Б. Особенности моделирования распределенных информационных систем. // Научный журнал. Вестник Пермского университета 2013. Математика. Механика. Информатика. Выпуск 4(23).

10. *COMNET III*. Planning for Network Managers. Release 1.3. [Электронный ресурс] [Режим доступа: <http://eent3.lsbu.ac.uk/staff/baoyb/acs/Comnet/comnet%20III.pdf>] [Проверено: 30.10.2016].
11. *OMNeT++ Community Site*. [Электронный ресурс] [Режим доступа: <http://www.omnetpp.org>] [Проверено: 30.10.2016].
12. *OPNET*. [Электронный ресурс] [Режим доступа: <http://www.riverbed.com/products/performance-management-control/opnet.html?redirect=opnet>] [Проверено: 30.10.2016].

Сведения об авторах:

Фамилия, имя и отчество автора (русский/английский)	Замятина Елена Борисовна
	Zamyatina Elena Borisovna
Учёная степень (и специальность ВНК), звание (русский/английский)	к.ф.-м.н.
Место работы (вуз, кафедра и т.п.) (русский/английский)	Кафедра информационных технологий
Должность (русский/английский)	доцент
Почтовый адрес (для отправки сборника по почте)	Пермь, 17, ул.Тургенева, д.33,40
Адрес электронной почты	e_zamyatina@mail.ru
Телефон/факс	+79194476944

Фамилия, имя и отчество автора (русский/английский)	Мальцев Георгий Валерьевич
	Maltcev Georgii Valerevich
Учёная степень (и специальность ВНК), звание (русский/английский)	

Место работы (вуз, кафедра и т.п.) (русский/английский)	ПГНИУ, кафедра математического обеспечения вычислительных систем
Должность (русский/английский)	студент магистратуры
Почтовый адрес (для отправки сборника по почте)	
Адрес электронной почты	georgemalcev@rambler.ru
Телефон/факс	+79194819168

ОПЫТ МОДЕЛИРОВАНИЯ АЛГОРИТМОВ МАРШРУТИЗАЦИИ РАЗРАБОТАННЫХ ДЛЯ SDN И SON СЕТЕЙ

Опыт моделирования алгоритмов маршрутизации разработанных для SDN и SON сетей: Пермь 2016. Авторы: Е.Б.Замятина, Г.А.Мальцев, С.Порязов, Э.Саранова.

В данной работе кратко рассмотрены SDN-сети и SON-сети, а также средство имитационного моделирования TriadNS, в котором разрабатывается имитационная модель SDN-сети. Рассмотрен алгоритм маршрутизации SBARC, который подходит для SDN-сетей, и его реализация в СИМ TriadNS. Работа будет полезна при построение имитационных моделей компьютерных сетей в СИМ TriadNS.

EXPERIENCE IN SIMULATION OF ROUTING ALGORITHMS DESIGNED TO SDN AND SON NETWORKS

Experience in simulation of routing algorithms designed to SDN and SON networks: Perm 2016. Authors: E.B.Zamyatina, G.A.Maltcev, S.Poryazov, E.Saranova.

In this research, we briefly discuss the SDN-networks and SON-networks and a simulation tool TriadNS, in which the simulation model of SDN-network is developed. Considered SBARC routing algorithm that is suitable for SDN-networks and its implementation in TriadNS. The research will be useful for the construction of simulation models of computer networks in TriadNS.