

температура эксплуатации лазера достигает 74 градусов. При таких температурах неизбежны насыщение ватт-амперных характеристик, падение мощности, а так же ускоренная деградация. Возможно, при таких амплитудах, частоту следования необходимо будет уменьшить до значения 10 Гц чтобы получить полное остывание лазерного диода в промежутках между импульсами.

ЗАКЛЮЧЕНИЕ

Высокие значения перегрева полученные для лазера с вытекающей модой объясняется высоким порогом генерации и низким значением КПД. И требует дальнейшего технологического совершенствования структуры.

Литература

1. *Тарасов И.С.* «Квантовая Электроника», 40, №8 (2010) С. 661–680.
2. *Слипченко С.О., Соколова З.Н., Пихтин Н.А., Борщев К.С., Винокуров Д.А., Тарасов И.С.*, ФТП, 40, №8, 2006, С. 1017–1023.
3. *Алешкин В.Я., Бабушкина Т.С.* Квантовая Электроника, 40 №10(2010) // С. 855–857.
4. *Некоркин С.М., Колесников М.Н., Дикарёва Н.В., Алёшкин В.Я., Дубинов А.А., Звонков Б.Н.* Вестник Нижегородского университета им. Лобачевского, 2012, №1, С. 30–32

РАЗРАБОТКА АЛГОРИТМА РАСПРЕДЕЛЕННОГО ХРАНЕНИЯ ДАННЫХ ДЛЯ БЕСПРОВОДНЫХ САМООРГАНИЗУЮЩИХСЯ СЕТЕЙ

Д. А. Наумович

ВВЕДЕНИЕ

Распределенное хранение данных (РХД) является единственным возможным способом дистанционного хранения информации в децентрализованных одноранговых, в том числе и беспроводных самоорганизующихся (ad hoc) сетях, однако на данный момент не существует широко распространенных алгоритмов РХД, которые бы обеспечивали гибкость при работе с данными, имели высокую производительность и устойчивость к динамике ad hoc сетей. Таким образом, важной и актуальной задачей является разработка такого алгоритма с целью его применения в беспроводных самоорганизующихся сетях.

ОСНОВЫ РХД И ДИНАМИЧЕСКОЙ МАРШРУТИЗАЦИИ

Правильно организованное и продуманное распределенное хранение данных имеет ряд преимуществ [2], [3]:

- повышает надежность и конфиденциальность данных;
- обеспечивает экономию ресурсов устройства;
- защищает передачу данных между узлами;
- делает возможным доступ к информации в любой точке сети;
- обеспечивает независимость данных от их физического расположения.

У такой технологии также есть и некоторые недостатки, в частности, можно выделить угрозу целостности данных, увеличенную загрузку отдельных узлов, возможность ведения вредоносной деятельности (с чем можно бороться), а также сложный процесс разработки.

Динамическая маршрутизация (ДМ) определяет, каким образом будут устанавливаться соединения между узлами, так как в ad hoc сети каждый пакет данных при отправке его с одного устройства на другое проходит некоторое количество промежуточных узлов, в данной ситуации выступающих в роли маршрутизаторов. Протокол ДМ говорит о том, как строится топология сети, какие узлы будут «соседями», как будут перемещаться пакеты данных [1].

РХД подразумевает, что каждый файл, который узел хочет хранить в сети, разбивается на определенное количество криптографически зашифрованных фрагментов, каждый из которых путешествует по сети, а затем «находит пристанище» на одном из узлов. В данном случае динамическая маршрутизация определяет пути к узлам, на которые будут пересылаться эти самые фрагменты данных. Таким образом, РХД невозможно без ДМ.

РАЗРАБОТКА АЛГОРИТМА РАСПРЕДЕЛЕННОГО ХРАНЕНИЯ ДАННЫХ

Разработка любого алгоритма на первых этапах обязательно включает в себя определение требований к нему. В данном случае, они непосредственно коррелируют с ограничениями, которые ad hoc сети накладывают на разрабатываемый алгоритм. Были определены следующие требования/ограничения:

- устойчивость к динамике беспроводных самоорганизующихся сетей;
- справедливость по отношению ко всем узлам;
- согласованность и независимость от других алгоритмов сети;

- наличие организованной системы оповещения об ошибках;
- алгоритм должен обеспечивать конфиденциальность, доступность и целостность данных;
- независимость данных от их территориального расположения;
- возможность корректного параллельного обращения к данным с помощью средств синхронизации.

Разработанный АРХД является абстрактной моделью, с использованием которой можно реализовать реальную децентрализованную сеть. Таким образом, алгоритм не затрагивает ни физический, ни канальный уровень модели OSI/ISO.

Опишем некоторые особенности разработанного алгоритма. Во-первых, количество фрагментов, на которые разбивается файл, не постоянно, зависит от его размера и определяется в соответствии с табл. 1.

Во-вторых, метрикой алгоритма мы выбрали XOR-функцию, которая применяется к ID узла и ID фрагмента для определения узла-назначения, куда отправится пакет с распределенными данными. Количество копий, создаваемых для каждого фрагмента, равно 4.

При отключении от сети некоторого узла, хранящего распределенные данные, производится их *резервное копирование* для сохранения целостности и доступности файла.

Таблица 1

Зависимость количества фрагментов разбиения от размера файла

Размер файла	До 5 кб	От 5 до 50 кб	От 50 до 500 кб	От 500 кб до 5 Мб	От 5 до 50 Мб	От 50 до 500 Мб	От 500 Мб
Количество фрагментов	8	16	32	64	128	256	512

МОДЕЛИРОВАНИЕ СЕТИ И ТЕСТИРОВАНИЕ АЛГОРИТМА

Для проверки работоспособности алгоритма, а также с целью оценки производительности, устойчивости и справедливости к узлам, работа АРХД в сети была смоделирована с помощью языка Java. На первом этапе был введен ряд упрощений, например, вместо динамической маршрутизации используется механизм, когда каждый узел с помощью алгоритма Дейкстры может рассчитать оптимальный маршрут до узла назначения [4]. Также, на первом этапе мы не шифровали фрагменты данных, распределяемые по сети, не разделяли устройства по их типу, а ID фрагментов и узлов генерировали произвольно. В модели узел имеет вид шара с радиусом R пикселей, а его соседями считаются узлы-шары, находящиеся от него на расстоянии меньшем или равным 10R. Все шары

двигаются прямолинейно и равномерно, отскакивая от стенок ограниченной области, где они находятся. То есть, в некоторый момент времени сеть имеет вид, представленный на рисунке 1.

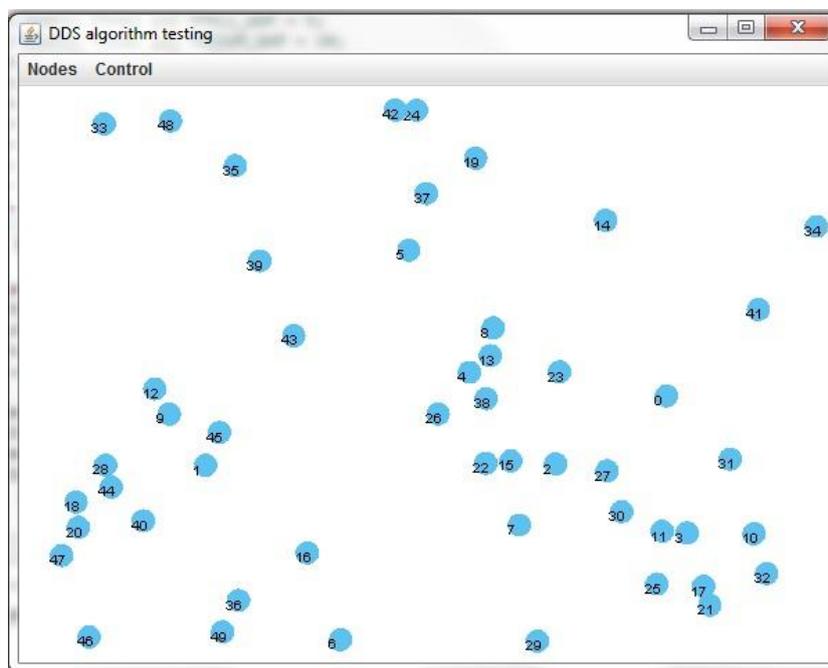


Рис. 1. Модель сети в произвольный момент времени

РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Во время моделирования и тестирования мы столкнулись со значимой проблемой: в виду того, что не использовалась динамическая маршрутизация и каждому узлу приходилось самостоятельно проверять состояние всех других узлов, у которых хранятся фрагменты файлов, принадлежащих ему, модель имеет проблемы, связанные со скоростью работы сети. Постоянные периодические проверки состояния узлов оказали негативное влияние на функционирование модели, загружая каждый узел.

ОЦЕНКА СПРАВЕДЛИВОСТИ АЛГОРИТМА

Для оценки произвольности и равномерности алгоритма, приведем некоторую статистику, касающуюся распределения фрагментов по сети и количества ретранслируемых пакетов с данными.

На диаграмме 1 приведена статистика по количеству хранимых фрагментов в сети, состоящей из 50 узлов, после того как пять произвольных узлов распределили по одному файлу. Видно, что, во-первых, каждый узел хранит хотя бы один фрагмент данных, а во-вторых, фрагменты

распределены практически равномерно, с некоторыми случайными отклонениями.



Диаграмма 1. Количество хранимых фрагментов на 50 узлах

На диаграмме 2 приведена та же статистика, но для сети, состоящей из 100 узлов, после того как десять произвольных узлов распределили по одному файлу. В этом случае каждый узел хранит уже как минимум четыре фрагмента данных, и распределение снова носит равномерно-случайный характер.



Диаграмма 2. Количество хранимых фрагментов на 100 узлах

На диаграмме 3 количество узлов уже равняется пятистам. Сто произвольных узлов распределили по одному файлу. Данная статистика подтверждает равномерность и произвольность «размазывания» фрагментов по сети.



Диаграмма 3. Количество хранимых фрагментов на 500 узлах

Приведенные диаграммы показывают: алгоритм распределяет фрагменты равномерно, с редкими отклонениями на большую величину. Эти отклонения носят случайный характер, то есть, не зависят от топологии, что подтверждает справедливость алгоритма по отношению к узлам в плоскости распределения файлов.

Ниже приведена диаграмма 4, которая отображает количество ретранслируемых узлами пакетов в сети, состоящей из 40 конечных устройств, после распределения пяти файлов и перехода восьми произвольных узлов в режим офлайн, то есть их отключения.



Диаграмма 4. Количество перенаправленных пакетов на 40 узлах

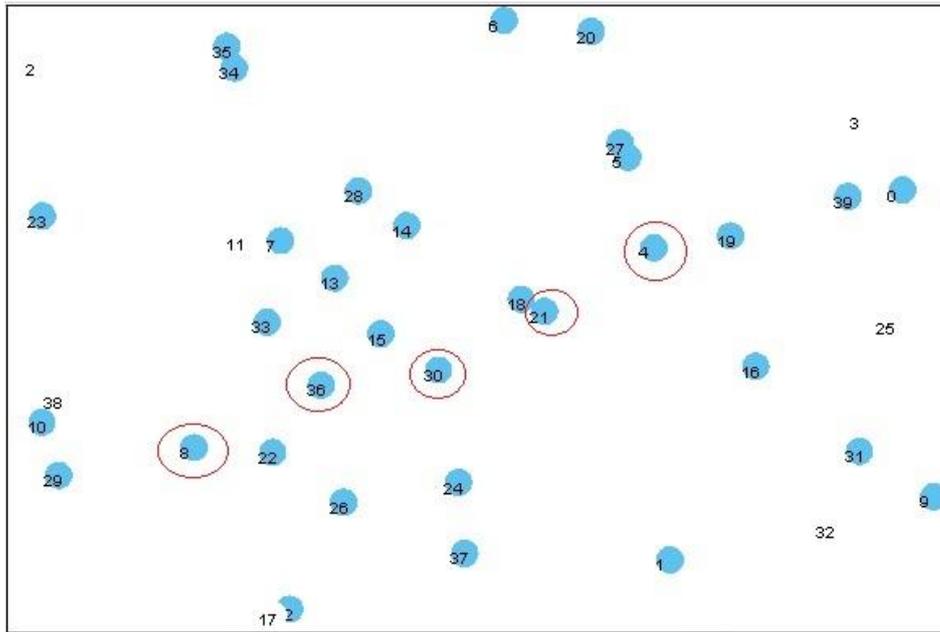


Рис.2. Сеть в некоторый произвольный момент времени

Из диаграммы видно, что некоторые узлы совершили за небольшой промежуток времени гораздо больше ретрансляций, чем иные, а некоторые узлы, наоборот, не совершили ни одной. С чем же это связано? Взгляните на рисунок 2, который показывает топологию сети в момент времени сразу же после сбора данных для диаграммы 4. Узлы, которые совершили наибольшее количество ретрансляций, обведены красным эллипсом. Видно, что узлы 21, 30, 36 находятся в центре сети, а узлы 8 и 4 соединяют виртуальные физические группы узлов по краям топологии с центральной ее частью. То есть, наиболее загружены проходящим мимо трафиком именно те устройства, которые располагаются на «перекрестках» видимых групп узлов. Следовательно, в этом плане алгоритм не справедлив, однако, все-таки построение маршрутов – сфера компетенции протокола динамической маршрутизации, а не АРХД. К тому же, необходимо помнить о том, что узлы постоянно перемещаются, что ведет к непостоянству топологий, а, значит, в разные моменты времени больше всего ретранслировать будут разные узлы, что, в конце концов, при рассмотрении большого промежутка времени, должно привести к равномерности и этого параметра.

Что касается устойчивости к динамике, то при большом количестве моделируемых нами сценариев алгоритм показал себя надежным, хотя в некоторых ситуациях все же были обнаружены сбои в работе, чаще всего связанные с проблемами поиска маршрутов.

ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ

Моделирование сети и тестирование алгоритма показало, что хоть алгоритм работает корректно в большинстве сетевых сценариев, связанных с отключением и перемещением файлов, все же модель оказалась медленно работающей. Это связано с большой загрузкой каждого узла, так как в виду упрощенной нами маршрутизации и отсутствия алгоритмов ДМ, каждому узлу приходится постоянно самостоятельно проверять состояние других устройств, которые хранят распределенные фрагменты. За 20 секунд в модели каждый узел проводит порядка 10^6 проверок состояний, что несомненно существенно загружает модель. Таким образом, самое сильное ограничение на модель накладывает именно построение и поиск маршрутов. Поэтому, первоочередной задачей при дальнейшем проектировании будет имплементация реактивного протокола динамической маршрутизации в модель.

Литература

1. *Адуцкевич И.* Анализ протоколов динамической маршрутизации в беспроводных ячеистых сетях / И. Адуцкевич, В. Садов // Инженер. вестн. 2006. № 1/2. С. 117–124.
2. *Hekmat R.* Ad-hoc Networks fundamental properties and network topologies. Delft, 2006.
3. *Subramanian R.* Peer-to-peer computing: The Evolution of a Disruptive Technology / R. Subramanian, B. Goodman. Hershey, PA, USA, 2005.
4. *Vogel L.* Dijkstra's shortest path algorithm in Java Tutorial [Электронный ресурс]. Режим доступа к источнику:
<http://www.vogella.com/tutorials/JavaAlgorithmsDijkstra/article.html>

МОДЕЛИРОВАНИЕ ЭЛЕКТРОФИЗИЧЕСКИХ СВОЙСТВ GAAS МЕТОДОМ МОНТЕ-КАРЛО

И. С. Новиков

ВВЕДЕНИЕ

Известно, что метод статистического моделирования или Монте-Карло в настоящее время является одним из наиболее перспективных численных методов моделирования современных полупроводниковых приборов и приборных структур микро- и наноэлектроники. Этот метод, применительно к полупроводниковым приборам впервые был предложен Куросавой в 1966г и являлся одночастичным методом для расчёта стационарных процессов в полупроводниках [1]. Он позволил отказаться от использования прямого решения кинетического уравнения Больцмана