Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибирнетики
Кафедра автоматизации систем вычислительных комплексов

Отчёт по заданию №4

Построение модели сети на основе среды моделирования NS3.

Вариат №2

Никифоров Никита Игоревич, 321 группа

Москва
2019

Оглавление

Постановка задачи

Необходимо построить модель сети, канала Fast Ethernet на основе среды моделирования NS3. Модель должна состоять из канала FastEthernet и некоторого количества хостов, подлючённых к нему. Требуется написать приложение для этих хостов, отсылающее в сеть пакеты с задержкой имеющей экспоненциальное распределение с заданным математическим ожиданием.
На основе построенной модели необходимо:

1. Рассчитать среднее число попыток повторной передачи
2. среднее и максимальное количество пакетов в буфере устройства

Также необходмо было рассчитать количество хостов, чтобы среднее число попыток переотправки пакета не превышало 2.0

Описание модели

Использовалась среда модделирования NS3, за основу была взята модель CSMA канала и устройств. Было написано приложение, которое с использованием протокола UDP отправляет пакеты в канал, не дожидаясь отправки предыдущего пакета. Задержка между отправкой пакетов задаётся случайной величиной с экспоненциальным распределением $\lambda * e^{-\lambda}$, задающимся параметром λ. Данная случайная величина была реализована с помощью стандартной библиотеки C++
`std::exponential_distribution`. Длина канала была представлена в модели как задержка на распространие информации, так как нанапрямую длину канала в условиях NS3 задать нельзя. Задержка вычислялась по формуле:

```
T = L * 0.01
```

Где T - искомая задержка, L - длина канала, 0.01 – задержка в микросекундах на метр.

По данной формуле получаем, что задержка канала - 300нс. В связи с этим случайная величина задержки между пакетами итерпритировалась как милисекунды. Приложение при отправке пакета кладёт его в очередь, как только пакет отправляется, устройство достаёт следующий пакет из очереди, если они есть, и пытается отправить его. Устройство в случае занятости канала пытается переотправить пакет через случайное время в соответствии с моделей Backoff и стандартом `IEEE 802.3`. Также в модели присутствует дополнительный хост, который имеет функцию сбора пакетов с хостов, и указывается в сокетах как пункт назначения. Пакет сбрасывался, когда при попытке добавить его в очередь, очередь была полностью занята.

Методы измерения

Для измерения искомых характеристик испольвались встроенные в NS3 методы `TraceConnect`, которые позволяют на определённое событие запустить определённую функцию. Причём для каждого хоста или канала отдельно назначается список тригеров. Для подсчёта сброшенных пакетов на очередь на каждом хосте назначался тригер `Drop`, в функции просто увелисивался счётчик сброшенных пакетов. Для подсчёта переотправок пакетов на каждый хост ставился тригер `MacTxBackoff`, в функции вызываемой этим тригером просто увеличивался счётчик попыток переотправки. Для подсчёта количества пакетов, которые должны были отправить все хосты использовался счётчик, который увеличивался в функции отправки в приложении. Размер очереди вычислялся как суммарное количество значений очереди на кажой генерации пакета (перед помещением его в очередь), делённый на количество сгенерированных пакетов.

Техническая реализация

Для реализации приложения был написан класс `MyApp`, который имеет следующие методы для решения задач поставленных в этой работе:

1. `void Setup(...)` – метод для задания уникальных параметров этому приложению
2. `void StartApplication(void)` – метод для запуска приложения
3. `void StopApplication(void)` – метод для остановки приложения
4. `void ScheduleTx(void)` – метод для генерации времени отправки нового пакета и указания этого времени симулятору
5. `void SendPacket(void)` – метод для генерации и отправки пакета через UDP сокет
6. `void PritLog(void)` – метод, работающий только для 0-го хоста, для отображения текущего времени симуляции в лог

Для реализации отправки пакетов использовались UDP сокеты, так как UDP протокол позволяет не ждать отправки предыдущего пакета, для генерации нового. Для измерения характеристик и логирования были реализованы следующие функции:

1. `void RxBegin(...)` – метод для отслеживания доставленных пакетов на сервер
2. `void QueDrop(...)` – метод для отслеживания сброса пакетов с очереди
3. `void MacTxBackoff(...)` – метод для отслеживания переотправки пакетов

В главной функции был использован механизм NS3 для передачи параметров программе.

1. `hosts` – количество хостов в симуляции, по умолчанию 10
2. `distr` – параметр распределения случайной величины времени задержки пакетов
3. `delay` – задержка канала связи

Данные параметры были использованы для быстрого тестирования программы. В главной функции сначала создаётся CSMA канал, затем создаются Nodes, которые мы присоединяем к каналу и получаем хосты. Затем для каждого хоста создаётся очередь и передаётся хосту, затем создаётся пул IPv4 адресов `10.1.0.0 - 10.1.255.255`, которые потом раздаются хостам. Далее для каждого хоста создаётся объект класса приложения, и устанавливается на хост. Устанавливаются временные рамки моделирования, и модель запускается. После окончания моделирования, выводятся значения всех счётчиков и программка завершается. Для получения дополнительной информации о ходе экспериментка используются лог файлы и pcap файлы.

Измерения

номер теста	параметр распределения	количество хостов	средняя длина очереди	максимальная длина очереди	Средние число переотправок пакета	количество отправленных пакетов	количество сброшенных пакетов
1	10	40	1.005	3	1.97	4057	0
2	10	41	1.006	3	2.06	4168	0
3	50	1	1.09	5	1.04	545	0
4	50	2	1.1	5	3.34	1090	0
5	100	1	1.19	5	1.09	1212	0
6	100	2	1.2	5	3.32	2224	0
7	100	80	45.1	100	22.35	84881	10639

Интерпретация измерений

В тесте 1, мы видим, что максимальное число хостов, для которых параметр средней переотправки не привосходит 2, равно 40. Из теста 2, понятно, что уже для 41 хоста этот параметр превышает 2. Для параметра распределения - 50, среднее число переотправок получилось меньше 1 только для одного хоста, что видно из тестов 3 и 4. Для параметра распределения - 100, среднее число переотправок также получилось меньше 1 только для одного хоста, что видно из тестов 5 и 6. Последний тест нужен, чтобы показать работу механизма сбрасывания пакетов, в нём очередь ограничена 100 пакетами, как видно была попытка отправить почти 85 тысяч пакетов, что невозможно, так как канал за 10 секунд может пропустить только 83333 пакета. Соответственно канал в данном случае был использован на 90%.

Выводы

Таким образом, для установленного параметра среднего числа попыток переотправки пакета равного 2, очереди на устройствах не будут бесконечно увеличиваться.