

ОТЧЁТ ПО ЗАДАНИЮ №4

Построение модели сети на основе среды моделирования NS3.

Вариант №2

Никифоров Никита Игоревич, 321 группа

Оглавление

Постановка задачи	2
Описание модели	2
Методы измерения	2
Техническая реализация	3
Измерения	3
Выводы	3

Постановка задачи

Необходимо построить модель сети, канала Fast Ethernet на основе среды моделирования NS3. Модель должна состоять из канала FastEthernet и некоторого количества хостов, подключённых 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 необходимо было рассчитать количество хостов, чтобы среднее число попыток переотправки пакета не превышало 2.0

Описание модели

Использовалась среда моделирования 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 распространение информации, так как непосредственно 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 интерпретировалась как миллисекунды. Приложение при отправке пакета кладёт его в очередь, как только пакет отправляется, устройство достаёт следующий пакет из очереди, если он есть, и пытается отправить его. Устройство в случае занятости канала пытается переотправить пакет через случайное время в соответствии с моделью 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 использовались встроенные в NS3 методы `TraceConnect`, которые позволяют на определённое событие запустить определённую функцию. Причём для каждого хоста или канала отдельно назначается список триггеров. Для подсчёта сброшенных пакетов на очередь на каждом хосте назначался триггер `Drop`, в функции просто увеличивался счётчик сброшенных пакетов. Для подсчёта переотправок пакетов на каждый хост ставился триггер `MacTxBackoff`, в функции вызываемой этим триггером 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 каждой 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 PrintLog(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 эксперимента используются лог файлы и рсар файлы.

Измерения

БЛА БЛА БЛА переделать нормально табличку тут
описание время моделирования количество хостов средняя длина очереди.

Выводы

Таким образом канал длиной 30 метров может поддерживать общение только блабла хостов для.