

Передача данных в сетях ATM с использованием LANE

Документ описывает механизмы инкапсуляции кадров Ethernet в ячейки ATM при использовании сервиса LANE для передачи трафика локальных сетей через магистрали ATM.

1 Введение

Документ содержит некоторые теоретические основы передачи трафика ЛВС через сети ATM. Далее в качестве примера приведены результаты работы сетевого анализатора, установленного в сети с использованием LANE, и с соответствующие комментарии. Приведена схема сети, в которой производились измерения, а также вся необходимая адресная информация.

В качестве устройств для построения тестовой сети ATM использовались коммутаторы производства Nortel Networks моделей Centillion 100 и 5000BH.

2 Инкапсуляция данных в ATM/LANE

2.1 Общая схема

На рисунке 1 приведена общая схема сети ATM, использующей сервис LANE для передачи трафика ЛВС.

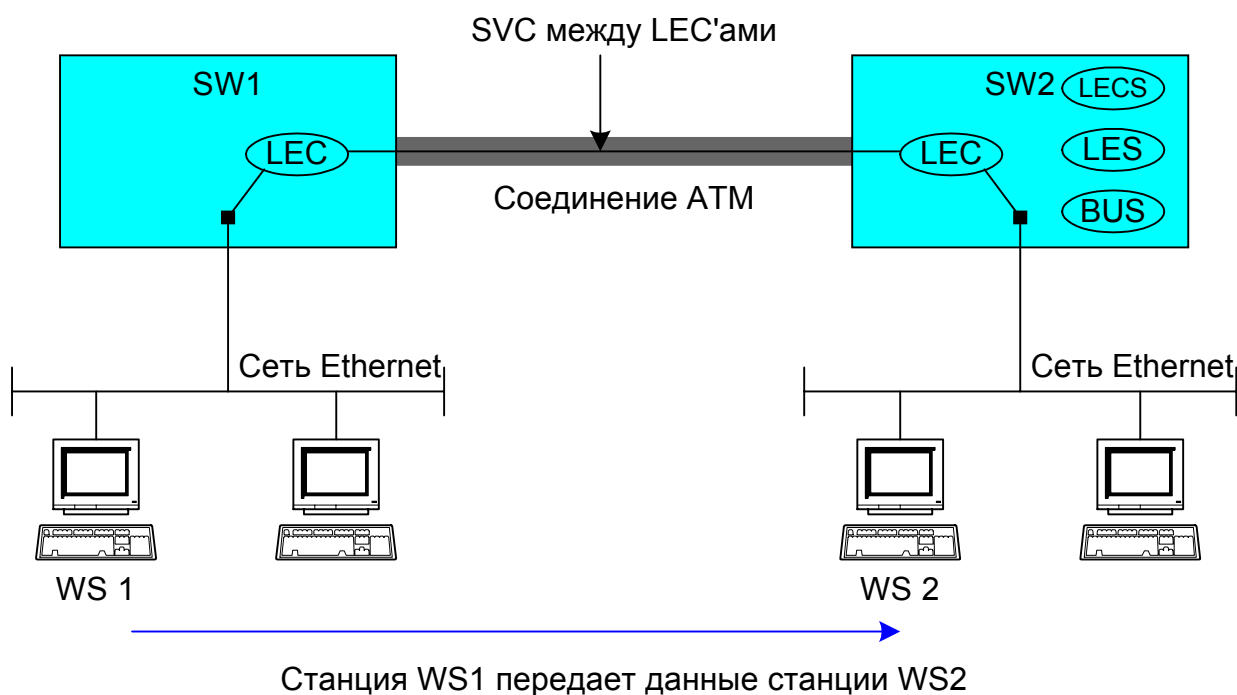


Рисунок 1 Сеть ATM с использованием сервиса LANE. Служебные каналы между LEC'ами и серверами LANE не показаны.

Рассмотрим типичный для такой сети случай. Станция WS1 должна передать данные станции WS2. Для этого клиент LEC (LAN Emulation Client), реализованный на пограничном коммутаторе SW1, должен установить соединение SVC с LEC'ом, реализованном на пограничном коммутаторе SW2. Пограничными коммутаторы в данном случае называются потому, что стоят «на границе» сети ATM и служат для передачи через нее трафика других сетей.

Используя соответствующие механизмы LANE и сигнализации ATM, LEC1 устанавливает соединение (SVC) с LEC2 и начинает передавать данные.

2.2 Использование AAL5

В LANE при передаче трафика традиционных ЛВС используется AAL5 – верхний уровень модели ATM, отвечающий за обработку данных и преобразование их в 48-байтовые ячейки. Уровень AAL (как пятый, так и все остальные) делится на два подуровня – Convergence Sublayer или CS (уровень преобразования) и Segmentation and Reassembly - SAR (уровень сегментации/сборки).

В AAL уровень CS отвечает за формирование пакета, пригодного для дальнейшей сегментации – CS-PDU (Protocol Data Unit – элемент данных протокола). Структура CS-PDU для AAL5 показана на рисунке 2.

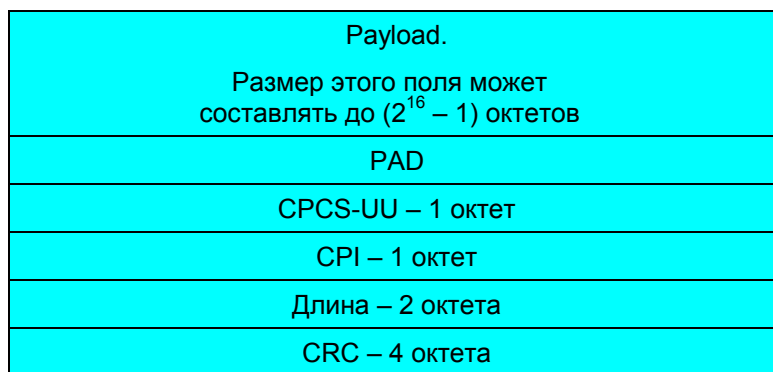


Рисунок 2 Формат AAL5 CS-PDU.

Payload

Поле Payload содержит информацию и имеет размер до $2^{16} - 1$ октетов.

PAD

Поле PAD (заполнение) используется для выравнивания размера CS-PDU по 48-октетной границе (т. е. при делении общей длины пакета на 48 не должно быть остатка). Пакет AAL5 CS-PDU будет затем передаваться уровню SAR, который поделит его на 48-байтовые ячейки.

CS-UU

Поле CS-UU (User-to-User indication – индикация пользователь-пользователь) может использоваться для прозрачной передачи пользовательской информации.

CPI

Поле CPI (Common Part Indicator – индикатор общей части) используется для выравнивания окончания CPCS-PDU по 64-битовой границе. В дальнейшем ITU-T может определить для этого поля дополнительные функции, если же поле используется только для выравнивания по 64-битовой границе, оно должно содержать значение 0x00.

Длина

Данное поле указывает длину Payload в октетах. Максимальное значение этого поля – 65535.

CRC

Поле контрольной суммы используется для контроля ошибок в пакете. Значение самого поля CRC при расчете контрольной суммы не учитывается.

Что же помещается в поле Payload? В поле Payload помещается пакет, сформированный LEC'ом из кадра Ethernet для передачи последнего через сеть ATM. Формат LEC-PDU показан на рисунке 3.

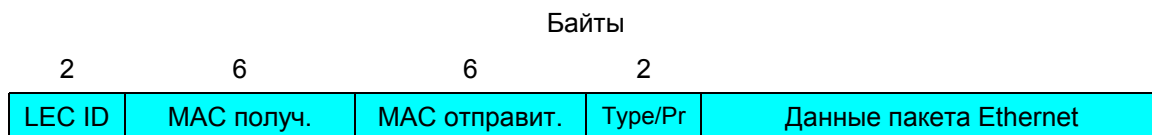


Рисунок 3 Структура LEC-PDU для IEEE 802.3.

LEC ID

Идентификатор клиента LEC.

MAC получ.

MAC-адрес станции – получателя пакета.

MAC отправит.

MAC-адрес станции – отправителя пакета.

Type/Pr

≥ 0x0600 Поле содержит тип протокола вышележащего (сетевого) уровня

46-1500 Длина пакета LLC, инкапсулированного в пакет Ethernet..

Данные пакета Ethernet

Информационное поле пакета Ethernet.

3 Пример инкапсуляции данных.

Рассмотрим на примере передачу данных через сеть ATM (рисунок 4). На станции WS1 запустим программу ping, в качестве параметра которой укажем адрес станции WS2. Механизм передачи данных будет следующим: на WS1 формируется ICMP-пакет -> инкапсуляция в IP -> инкапсуляция в Ethernet -> передача данных коммутатору SW1 -> обработка LEC (добавление заголовка, содержащего LEC ID, формирование CS-PDU, деление его на 48-байтовые ячейки). 48-байтовые ячейки передаются уровню ATM, который добавит к ним 5-байтовые заголовки. Принцип формирования заголовков мы здесь не рассматривать не будем.

На приведенных ниже рисунках показана схема инкапсуляции данных.

Сначала формируется пакет ICMP.

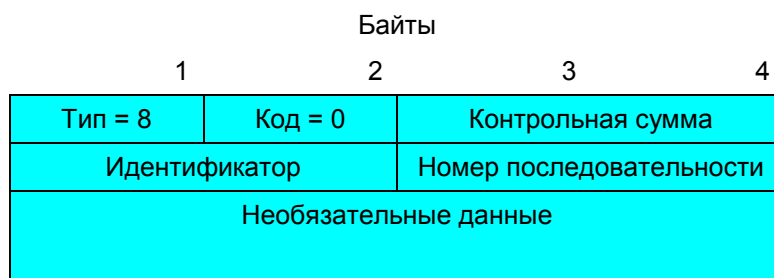


Рисунок 4 Пакет ICMP. Тип 8 – Echo Request.

Упакуем получившуюся структуру в пакет IP. В нашем примере IP-адрес отправителя (WS1) - 155.155.10.112, IP-адрес получателя (WS2) - 155.155.10.118.



Рисунок 5 Пакет IP. Поле протокола содержит значение 1 - идентификатор ICMP.

Пакет IP передается через сеть Ethernet в соответствующем кадре. На рисунке 6 приведены MAC-адреса рабочих станций WS1 и WS2, т. е. отправителя и получателя. Все значения полей пакета приведены в шестнадцатеричной системе счисления.



Рисунок 6 Пакет Ethernet.

После того, как пакет будет обработан LEC'ом коммутатора SW1, к нему будет добавлен соответствующий LEC ID, который в нашем случае равен 512 (десятичное значение).



Рисунок 7 LEC PDU.

Далее пакет передается уровню AAL. Первоначально формируется CP-PDU (рисунок 2), в который в качестве поля Payload упаковывается наш LEC PDU. CP-PDU передается уровню SAR, который делит его на 48-байтовые ячейки, передаваемые через сеть АТМ. Заглянем внутрь этих ячеек.

4 Декодирование ячеек ATM.

Ниже приведен пример декодирования двух ячеек ATM. В одной содержится запрос ICMP Echo Request, посланный станцией WS1 станции WS2, во второй – отклик ICMP Echo Reply от WS2 станции WS1. Комментарии выделены цветом.

Ячейка 1

ATM: VPI - 0	Заголовок ячейки ATM. 5 байт
ATM: VCI - 37	
ATM: AAL Type - 5	
ATM/SAR: Length - 76	
LE 802.3: LEC ID: 512	Начались данные. LEC ID
Ethernet: Destination Address 000092B60251	Заголовок Ethernet - 12 байт
Ethernet: Source Address 000092B607EF	
Ethernet: Ethernet V.2, Type DOD IP	
IP: Version = 4	IP-заголовок - 20 байт
IP: Total Length = 60	
IP: Identification = 56840	
IP: Flags & Fragment Offset: 0x0000	
IP: .0..... May Fragment	
IP: ..0..... Last Fragment	
IP: Fragment Offset = 0 [Bytes]	
IP: Time to Live = 32 [Seconds/Hops]	
IP: Protocol: 1 ICMP	
IP: Header Checksum = 0x709C	
IP: Source Address = 155.155.10.112	IP-адрес отправителя
IP: Destination Address = 155.155.10.118	IP-адрес получателя
ICMP: Type: 8 Echo request	Начался пакет ICMP
ICMP: Code: 0	
ICMP: CheckSum: 0x4F56	
ICMP: Identifier: 256	Идентификатор. Должен совпадать
ICMP: Sequence Number: 64773	с аналогичным полем в ответе
User Data	
OFFSET DATA	ASCII
002C: 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70	abcdefghijklmnop
003C: 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69	qrstuvwxyzabcdefghijklmnop
Frame Tail	
OFFSET DATA	ASCII
004C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4CL
005C: A5 3A 44 7F	.:D.

Ячейка 2

```

ATM: VPI - 0
ATM: VCI - 37
ATM: AAL Type - 5
ATM/SAR: Length - 76
LE 802.3: LEC ID: 512
Ethernet: Destination Address 000092B607EF
Ethernet: Source Address 000092B60251
Ethernet: Ethernet V.2, Type DOD IP
IP: Version = 4
IP: Total Length = 60
IP: Identification = 46924
IP: Flags & Fragment Offset: 0x0000
IP:  .0..... May Fragment
IP:  ..0..... Last Fragment
IP:  Fragment Offset = 0 [Bytes]
IP: Time to Live = 32 [Seconds/Hops]
IP: Protocol: 1 ICMP
IP: Header Checksum = 0x9758
IP: Source Address = 155.155.10.118
IP: Destination Address = 155.155.10.112
ICMP: Type: 0 Echo Reply
ICMP: Code: 0
ICMP: CheckSum: 0x5756
ICMP: Identifier: 256
ICMP: Sequence Number: 64773
User Data
OFFSET DATA
002C: 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70  abcdefghijklmnop
003C: 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69  qrstuvwabcdefghi
Frame Tail
OFFSET DATA
004C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4C  .....L
005C: E2 F1 12 94

```

Заголовок ячейки АТМ. 5 байт

Начались данные. LEC ID
Заголовок Ethernet - 12 байт

IP-заголовок - 20 байт

IP-адрес отправителя
IP-адрес получателя
Начался пакет ICMP

Идентификатор. Должен совпадать
с аналогичным полем в запросе

ASCII

ASCII