

УДК 004.725.5(045)

Яковин К.Н.,  
Чуприн В.М.,  
Ткалич О.П.

## РЕКОМЕНДАЦИИ ПО ПРОЕКТИРОВАНИЮ И РАСЧЕТУ МУЛЬТИСЕРВИСНЫХ ЛОКАЛЬНЫХ СЕТЕЙ

Национальный авиационный университет

*Изложены рекомендации по проектированию мультисервисных локальных сетей. Приведен конкретный пример расчета. Детально рассмотрены построение информационно-геометрической модели сети, построение модели идеального коммутатора, реализация идеального коммутатора в заданном классе промышленных коммутаторов, настройка параметров промышленного коммутатора*

### Основные этапы проектирования и расчета LAN

Можно порекомендовать несколько хороших источников [1–3] для ознакомления с отдельными элементами технологии проектирования локальных вычислительных сетей (LAN). Однако найти алгоритм проектирования и тем более пример расчета этих сетей трудно.

Анализ технологий проектирования некоторых фирм и доступные публикации показывают, что алгоритм проектирования и расчета LAN можно разбить на три последовательные части – это:

– проектирование и расчет активного оборудования, то есть аппаратной конфигурации транспортных и конечных узлов сети;

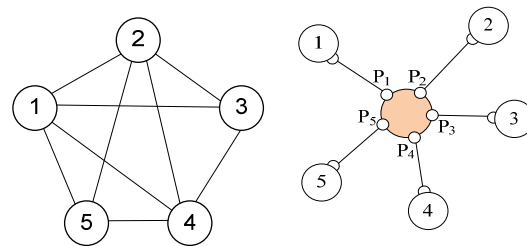
– проектирование и расчет параметров настройки узлов сети (коммутаторов, компьютеров конечных узлов, их операционных систем и приложений);

– проектирование и расчет пассивного оборудования, включая расчет топологии и характеристик каналов связи.

### Формальное описание информационных и компьютерных сетей в виде графов и матриц. Топология сетей

В общем случае проектируемую сеть можно представить набором графов и матриц. Вершинам графов будут

соответствовать узлы сети (источники/потребители информации), а ребрам – информационные или физические связи между вершинами. Конкретный граф может иметь ту или иную конфигурацию, так называемую топологию. В случае, когда каждая вершина графа непосредственно связана со всеми остальными вершинами, называется полностью связанным графом представленным на рис.1.



Вершины графа:  $n=1,2,\dots,N$

Ребра графа:  $A=||a_{i,j}||$ , где  $i,j=1,2,\dots,N$  (1)

Рис.1. Представление сети графом

### Начальные данные для проектирования сети. Приложения

В первом приближении моделью сети можно считать следующий набор графов и матриц:

$$A^0, A^1, \dots, A^m, \dots, A^M, m=0,1,2,\dots,M, \quad (2)$$

где  $A^0$  – геометрическая конфигурация сети (матрица расстояний между узлами сети),  $A^1, \dots, A^M$  – конфигурация приложений.

Конфигурация приложений представляет собой функционально-информационную модель сети. Эта модель отражает распределение приложений по уз-

лам сети, а также интенсивность индивидуальных потоков между серверами и клиентами приложений сети. Примерами приложений могут служить: *WWW*, *FTP*, *VoIP*, *NC* (сервер управления сетью) и др.

Интенсивности приложений – функции времени. Однако в большинстве случаев для расчета сети можно ограничиться средними значениями интенсивностей трафиков за определенные промежутки времени.

Пусть для определенности проектируется сеть, состоящая из  $N=34$  узлов, среди которых есть серверы: *VoIP* (узел  $n=1$ ), *Video* ( $n=2$ ), *WWW* ( $n=3$ ), *FTP* ( $n=4$ ), а также рабочие станции с клиентскими частями приложений ( $n=5, \dots, 34$ ).

В результате модель сети будет представлена матрицей расстояний между узлами сети  $A^0$  и матрицами интенсивностей трафиков: *VoIP* ( $A^1$ ), *Video* ( $A^2$ ), *WWW* ( $A^3$ ), *FTP* ( $A^4$ ).

### **Трансформация модели сети с полносвязной топологией в модель сети со звездообразной топологией с помощью идеального коммутатора**

В сети с полносвязной топологией для связи  $N$  узлов требуется  $N(N-1)/2$  физических дуплексных связей, то есть имеет место квадратичная зависимость от числа узлов. Кроме того, каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров в сети. Оба этих фактора сводят “на нет” главное преимущество полносвязных сетей – их логическую простоту.

На практике стремятся минимизировать число физических связей между узлами сети. Такая цель достижима, если узлы сети связать между собой через коммутатор. В результате количество связей между узлами сети становится минимальным, а топология сети из полносвязной трансформируется в звездообразную (рис. 1).

Легко видеть, что количество ( $N$ ) портов ( $p_i=1, 2, \dots, N$ ) в коммутаторе должно совпадать с числом узлов сети ( $N$ ), а информационные потоки через каждый ( $i$ -й) порт коммутатора ( $p_i$ ) – те же самые, что и через связанный с ним порт ( $i$ -го) узла сети. Другими словами, интенсивности трафиков через порты коммутатора также описываются набором матриц:  $A^1, \dots, A^M$ .

Важно и нужно отметить, что идеальный коммутатор должен быть для сети прозрачным, то есть никоим образом не влиять на проходящие через коммутатор информационные битовые потоки.

### **Расчет распределения трафиков приложений по портам идеального коммутатора**

Если просуммировать интенсивности трафиков ( $a^m_{i,j}$ ) какого-либо конкретного приложения  $m$  ( $m=1, 2, \dots, M$ ), протекающие через каждый ( $i$ -й) порт коммутатора, то получим суммарную интенсивность трафиков этого конкретного приложения для этого ( $i$ -го) порта коммутатора, т.е.:

$$b_i^m = \sum_{j=1}^{i=n} a^m_{i,j}. \quad (3)$$

Так для проектируемой нами сети на основании значений матриц  $A^1, A^2, A^3, A^4$  определяются суммарные интенсивности трафиков приложений в портах коммутатора – это:

$B^1$  (трафик *VoIP*),  $B^2$  (трафик *Video*),  $B^3$  (*www*-трафик),  $B^4$  (*ftp*-трафик).

### **Упрощенный расчет распределения трафиков приложений по портам идеального коммутатора**

Рассчитать распределения трафиков приложений ( $B^m$ ) по портам идеального коммутатора можно и более простым способом. Рассмотрим это на нашем примере.

На первом шаге построим таблицу  $\|a^1_{ij}\|$  размерностью  $34 \times 34$ , в которой каждый элемент матрицы будет равен 0 или 1:  $\|a^1_{ij}\|=1$ , если между  $i$ -м и  $j$ -м узлами коммутатора возможна связь по *IP*-телефону,  $\|a^1_{ij}\|=0$ , в противном случае.

На втором шаге рассчитаем количество возможных сеансов переговоров по *IP*-

телефону для всех портов коммутатора – матрицу  $B_0^1$  (табл. 1).

На третьем шаге определим количество одновременных переговоров по IP-телефону (одновременных сеансов приложения VoIP) через каждый порт коммутатора. Очевидно, что в портах коммутатора, связанных с рабочими станциями, число одновременных сеансов будет равно 1, а через порт-1 может быть установлено 30 сеансов одновременно. В результате получим еще одну строку-матрицу  $B_1^1$ .

На четвертом шаге укажем для каждого сеанса количество дуплексных связей и битовую скорость передачи голоса (с учетом выбранного стандарта кодирования). В нашем случае для одного сеанса потребуется две дуплексные линии связи с битовой скоростью 64 Кбит/с (при избрании стандарта ко-

дирования ИКМ). В результате получим очередную матрицу-строку  $B^1$ .

Аналогично для видео-приложений вычислим матрицу-строку  $B^2$ . Здесь единственное отличие будет в скорости видеотрафика. Для определенности пусть это будет компрессированный видео-сигнал с битовой скоростью 1,024 Мбит/с.

Для определенности также зададим интенсивности WWW-трафиков ( $B^3$ ) и интенсивности FTP-трафиков ( $B^4$ ).

**Расчет распределения классов трафиков по портам идеального коммутатора**

Одна из главных целей проектирования современной локальной сети – это обеспечение нормальной работы приложений с мультимедийным трафиком.

Таблица 1. Пример расчета производительности портов коммутатора

	1	2	3	4	5	6	7-33	34	Всего
	VoIP - шлюз	Video - шлюз	www - сервер	ftp - сервер	PC1	PC2	PC-(3-29)	PC - 30	
$B_0^1$	30				30	30	30	30	
$B_1^1$	30				1	1	1	1	
$B^1$	3,840				0,128	0,128	3,46	0,128	8
$B^2$		61,440			2,048	2,048	55,296	2,048	123
$B^3$			570,00		19,00	19,00	513,00	19,00	1140
$B^4$				750,00	25,00	25,00	675,00	25,00	1500
$T^0$			570,00	750,00	44,00	44,00	1188,00	44,00	2640
$H^{Vo}$	3,84	61,44	0,00	0,00	2,18	2,18	58,75	2,18	131
$H^{Be}$	0,00	0,00	570,00	750,00	44,00	44,00	1188,00	44,00	2640
$V$	19	307	633	833	51	51	1385	51	3332

Для решения этой задачи приложения делят на классы. В локальных сетях согласно рекомендациям стандарта 802.1D-1998 весь трафик локальных сетей делится на восемь классов (табл. 2).

С учетом этой классификации матрицы распределения интенсивностей

трафиков приложений по портам коммутатора  $B^m$  могут быть легко пересчитаны соответственно в восемь матриц интенсивностей классов трафиков:

$$\{B^1, \dots, B^m, \dots, B^M\} \Rightarrow \{T^1, \dots, T^0\}. \quad (4)$$

Таблица 2. Типы трафика в локальной сети

Приоритет	Обозначение	Тип трафика
1	ВК	Фоновый трафик
2	–	Экономный трафик
0 (вариант по умолчанию)	BE	Трафик, передаваемый с максимальными усилиями («по возможности»)

3	EE	Приоритетный трафик
4	CL	Контролируемый трафик
5	VI	Видео (задержка и джиттер менее 100 мс)
6	VO	Голос (задержка и джиттер менее 10 мс)
7	NC	Управление сетью

В нашем примере:

$T^6=B^1$  (трафик *VoIP* отнесен к классу *VO*),  
 $T^5=B^2$  (трафик *Video*-приложений отнесен к классу *VI*),

$$T^0=B^3+B^4$$

Как видно, *WWW*- и *FTP*-трафики просуммированы и отнесены к одному классу (матрица-строка  $T^0$  в табл. 1).

### Упрощение модели идеального коммутатора путем слияния классов трафика

Коммутатор, поддерживающий параметры *QoS*, позволяет использовать несколько очередей для дифференцированной обработки классов трафика. Очереди могут обслуживаться в соответствии с алгоритмом приоритетной обработки, взвешенного обслуживания или на основе комбинации этих алгоритмов.

Коммутатор обычно поддерживает некоторое максимальное количество очередей, которое может оказаться меньше, чем требуемое число классов трафика. В этой ситуации несколько классов будут обслуживаться одной очередью, то есть фактически сольются в один класс. Стандарт 802.1D-1998 дает рекомендации в отношении того, какие классы трафика нужно реализовывать в сети в условиях ограниченного количества очередей в коммутаторах.

При существовании только одной очереди в сети существует только один класс трафика – *BE* (*Best Effort*).

Две очереди дают возможность разделить трафик на два класса – *BE* и *VO* (*Voice*). При таком делении к классу *VO* следует относить любой чувствительный к задержкам трафик, то есть не только голос, но и видео, а также трафик управления сетью.

Дальнейшее увеличение количества очередей позволяет более дифференцированно обслуживать трафик, вплоть до рекомендуемых восьми классов.

С учетом сказанного не трудно вычислить два основных класса потоков в идеальном коммутаторе (*VO* и *BE*) по следующей формуле:

$$\begin{aligned} H^{VO} &= T^7 + T^6 + T^5, \\ H^{BE} &= T^4 + T^3 + T^2 + T^1 + T^0. \end{aligned} \quad (5)$$

В нашем примере:  $T^7=T^4=T^3=T^2=T^1=0$ ,  $H^{VO}=T^6+T^5=B^1+B^2$  (трафики *VoIP* и *Video* слиты в один класс)  $H^{BE}=T^4+T^3+T^2+T^1+T^0=B^3+B^4$  (в одном классе трафики *WWW*, *FTP*).

### Резервирование пропускной способности портов идеального коммутатора

Известно, что основная идея, лежащая в основе всех методов поддержания характеристик *QoS*, заключается в следующем: общая производительность каждого ресурса должна быть разделена между разными классами трафика неравномерно. В частном случае производительность портов коммутатора также должна быть разделена между разными классами трафика неравномерно.

Известно, что главным фактором, влияющим на величину задержек пакетов в коммутаторе, а значит, и на качество обслуживания, является коэффициент использования ресурса. Поэтому для обеспечения определенного качества обслуживания важно, чтобы коэффициент использования порта коммутатора не превышал определенной величины.

Итак, для упрощения задачи все потоки разделены на два класса – чувствительный к задержкам (трафик реального времени, например голосовой) и эластич-

ный, допускающий большие задержки, но чувствительный к потерям данных.

Согласно [2] зависимость задержек пакетов от коэффициента использования ресурса иллюстрирует функция  $W=W(P)$  на рис. 2. Здесь  $W_s$  – задержки чувствительного к задержкам трафика, а  $W_e$  – задержки эластичного трафика.

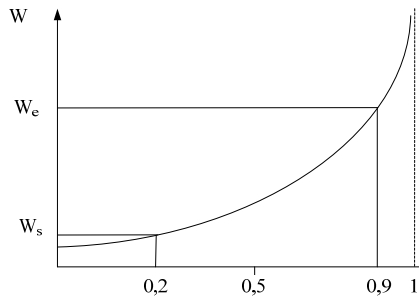


Рис. 2. Обслуживание эластичного и чувствительного к задержкам трафика

Как видно, предельная загрузка порта коммутатора чувствительным трафиком не должна превышать 0.2, а предельная общая загрузка порта чувствительным и эластичным трафиком не должна превышать 0.9.

С учетом принятых обозначений и названных ограничений пропускную способность портов коммутатора ( $v_i=1, \dots, N$ ) для двух классов трафика  $H^{VO}$  и  $H^{BE}$  можно определить следующей формулой:

$$v_i = \max((h^{VO}_i + h^{BE}_i)/0.9, h^{VO}_i/0.2). \quad (5)$$

Результат расчета пропускной способности всех портов коммутатора (матрица-строка  $V$ ) приведен в табл. 1.

Резервирование пропускной способности портов коммутатора по формуле (5) основано на двух классах обслуживания. Но эту формулу не трудно модифицировать для более чем двух классов обслуживания, причем так, чтобы каждый класс работал на своей части кривой задержек.

### **Вычисление параметров конфигурации идеального коммутатора**

На основании табл. 1 можно определить параметры конфигурации идеального коммутатора (табл. 3). Здесь общая производительность коммутатора равна удвоенной сумме производительностей всех портов коммутатора.

Таблица 3. Конфигурация идеального коммутатора

№	Название параметра	Значение
1	Общая производительность коммутатора	6,7 Гбит/с
2	Количество портов	34
3	Производительность портов:	
	производительность порта-1	19 Мбит/с
	производительность порта-2	307 Мбит/с
	...	
	производительность порта-34	51 Мбит/с
4	Количество приоритетных очередей в портах	2

### **Алгоритм реализации идеального коммутатора. Определенные критерии выбора**

Простейший алгоритм реализации идеального коммутатора может быть сведен к поиску такой промышленной модели коммутатора, конфигурация которой охватывает и наиболее близка к конфигурации идеального коммутатора – это, во-первых. А, во-вторых, необходимо, чтобы параметры промышленной модели могли программно изменяться администратором сети так, чтобы после соответствующей

настройки промышленный коммутатор был идентичен идеальному коммутатору.

Определим последовательность параметров промышленной модели, которые следует учитывать. Конечно, в первую очередь, промышленная модель должна иметь параметры не хуже тех, какие отображены в таблице 3. Во вторую очередь следует учесть в приоритетном порядке следующие параметры: число уровней управления, возможность агрегирования портов, возможность профилирования портов, возможность поддержа-

ния параметров качества обслуживания мультимедийного трафика (*QoS*, *CoS*), желательность наличия *Web*-интерфейса для настройки, фирма-производитель, цена.

Выбрав параметр “L3” (число уровней управления), мы тем самым сужаем поиск искомой промышленной модели в классе так называемых маршрутизаторов локальных сетей, которые предназначены

для разделения крупных локальных сетей на подсети.

Пошаговое сравнение параметров конфигурации идеального коммутатора (табл. 3) с параметрами конфигурации фирменных коммутаторов класса “L3” фирмы “D-link” показывает, что параметры модели *DES-3852* (табл. 4) совпадают или не хуже параметров идеального коммутатора.

Таблица 4. Структура параметров модели коммутатора *DES-3852*

№	Название параметра	Значение
1	Производительность	15,7 Гбит/с
2	Количество портов 10/100/1000/ <i>STP</i>	2
3	Количество портов 10/100/1000	2
4	Количество портов 10/100	48
5	Количество приоритетных очередей	8
6	Число уровней управления	L3
7	802.3ad Link Aggregation: 32 группы, 8 портов на группу	+
8	Управление полосой пропускания: шаг для каждого порта 64 Кб/с	+
9	<i>QoS/CoS</i>	+
10	Цена (\$)	1030

### **Конфигурирование промышленного коммутатора и узлов сети**

Модель коммутатора *DES-3852* обладает мощным набором команд и функций по структуризации сети – это:

- агрегирование портов и каналов связи;
- поддержка приоритетов очередей (для обеспечения *QoS* по протоколу 802.1p);
- создание виртуальных сетей.

### **Агрегирование портов и каналов связи**

Транки, то есть объединения портов в группы, создаются, чтобы объединить несколько портов вместе для образования одного единого широкополосного канала связи.

При этом коммутатор работает со всеми портами в группе, как с одним отдельным портом. Данные, переданные в определенный узел сети (по определенному адресу назначения), будут всегда передаваться через тот же самый порт в группе портов. Это позволит пакетам в

потоке данных прибывать в узел назначения в том же самом порядке, в котором они были посланы. Таким образом, агрегирование каналов связи позволяет группировать несколько портов вместе и получать единый канал связи. В результате полоса частот этого единого канала связи будет представлять собой сумму полос частот отдельных портов коммутатора, сгруппированных в группу.

Агрегирование портов наиболее часто используется, чтобы соединить высокоскоростные устройства, такие, например, как серверы, с магистралями сети.

Коммутатор позволяет создавать до 32 групп (объединений портов в транки), причем каждая группа может включать от 2 до 8 портов. Все порты в группе должны принадлежать одной и той же *VLAN*, а их *STP* статус, управление трафиком, сегментация трафика и конфигурация приоритетов очередей (согласно протоколу 802.1p) должны быть идентичны. Захват порта, порт *mirroring* и 802.X нельзя использовать в транках. Более того, объединенные порты должны иметь одну и ту

же самую скорость и должны быть сконфигурированы как полно-дуплексные.

*Spanning Tree Protocol (STP)* будет

обращаться с группой связей портов, как с одним отдельным каналом связи. *STP* будет автоматически распределять нагрузку между всеми портами в транке и в случае отказа какого-либо одного канала связи между портами заставит трафик сети перераспределиться между оставшимися действовать портами группы.

### **Поддержка приоритетов очередей для обеспечения QoS по протоколу 802.1p**

*IEEE 802.1p* – это стандарт поддержания качества сервиса (*QoS*), который позволяет администраторам сети резервировать полосу частот для важных функций с высоким приоритетом, типа *VoIP* (передача голоса по *IP*-протоколу), *web browsing* приложения, *file server* приложения или видео-конференции.

Возможно не только создание широкополосных каналов, но и ограничение полосы частот других менее критичных трафиков сети. Коммутатор имеет отдельные очереди в каждом физическом порту, к которым пакеты данных от различных приложений могут быть привязаны. Каждой очереди порта может быть назначен свой приоритет.

Коммутатор имеет восемь приоритетов для назначения очередям портов и может ограничивать потолок скорости передачи данных через любой свой порт.

В режиме строго приоритетного управления трафиком любые пакеты, живущие в очередях с более высоких приоритетом обслуживания, должны быть переданы первыми. Только, когда очереди с высоким приоритетом пусты, передаются пакеты из очередей более низкого приоритета.

В режиме взвешенного управления трафиком число пакетов, посланных от каждой очереди, зависит от назначенного ей веса. В этом режиме очереди, имеющие один и тот же вес, имеет равную воз-

можность послать свои пакеты внутри каждого цикла обслуживания очередей.

В режиме взвешенного управления трафиком в случае, если вес для какой-либо очереди установлен в 0, то передача пакетов из этой очереди продолжается до тех пор, пока очередь не опустеет. Другие очереди, имеющие ненулевой вес, будут передавать свои пакеты с учетом веса.

Параметры настройки *QoS* в коммутаторе могут быть изменены администратором сети. При этом требуется осторожность, чтобы трафик с низким приоритетом не повредить. Изменения в настройках могут привести к неприемлемому числу потерь пакетов или значительным задержкам пакетов. При изменениях настроек очень важно выполнять мониторинг работы сети, особенно во время пиковых нагрузок.

### **Выводы**

Изложены рекомендации по проектированию мультисервисных локальных сетей. Приведен конкретный пример расчета.

Детально рассмотрены построение информационно-геометрической модели сети, построение модели идеального коммутатора, реализация идеального коммутатора в заданном классе промышленных коммутаторов, настройка параметров промышленного коммутатора.

Авторы надеются, что изложенная в работе концепция проектирования мультисервисных локальных сетей будет полезна для студентов, аспирантов и технических специалистов в области телекоммуникаций.

В основу рассмотренных выше рекомендаций положены концепция и идеи, опубликованные в трудах конференций [4-5] в 2008 году.

### **Список литературы**

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е изд. – СПб.: Питер, 2007. – 958 с.