Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра метрологии и стандартизации

А. Л. Гурский, Н. А. Певнева

ТЕЛЕКОММУНИКАЦИОННЫЕ И ИНФОРМАЦИОННЫЕ СИСТЕМЫ И СЕТИ

*Рекомендовано УМО по образованию
в области информатики и радиоэлектроники
для специальности 1-54 01 04 «Метрологическое обеспечение информационных систем и сетей» и направления специальности 1-54 01 02-01 «Инфокоммуникационные системы (стандартизация, сертификация и контроль параметров)» в качестве пособия*

Минск БГУИР 2013

УДК [621.391+004.7](076)
ББК 32.88я73+32.973.202я73
Г95

Р е ц е н з е н т ы:
кафедра телекоммуникационных систем учреждения образования
«Высший государственный колледж связи»
(протокол №8 от 19.04.2013);

доцент кафедры робототехнических систем
Белорусского национального технического университета,
кандидат технических наук, доцент Ю. Е. Лившиц

Гурский, А. Л.

Г95 Телекоммуникационные и информационные системы и сети : пособие / А. Л. Гурский, Н. А. Певнева. – Минск : БГУИР, 2013 – 62 с. : ил.
ISBN 978-985-488-981-8.

Пособие содержит учебно-методические материалы для проведения практических занятий по основам локальных телекоммуникационных сетей, использующих стек протоколов TCP/IP. Основное внимание уделено вопросам корректного построения сетей Ethernet, распределения адресного пространства IP-сетей, формату кадров сетей Ethernet и анализу протоколов локальных сетей.

УДК [621.391+004.7](076)
ББК 32.88я73+32.973.202я73

ISBN 978-985-488-981-8

Содержание

Практическое занятие №1

МЕТОДИКА РАСЧЕТА КОНФИГУРАЦИИ СЕТИ ETHERNET 10 МБИТ/С

Цель занятия: изучение методики анализа работоспособности сети Ethernet 10 Мбит/с с помощью проверочного расчета, обучение применению данной методики при расчете сетей Ethernet 10 Мбит/с.

1.1 Теоретические сведения

1.1.1 Понятие коллизии и домена коллизий

Все виды стандартов Ethernet (в том числе Fast Ethernet и Gigabit Ethernet) используют один и тот же метод разделения среды передачи данных – метод CSMA/CD. При описанном подходе возможна ситуация, когда две станции одновременно пытаются передать кадр данных по общей среде. Механизм прослушивания среды и пауза между кадрами не защищают от возникновения такой ситуации, когда две или более станции одновременно решают, что среда свободна, и начинают передавать свои кадры. Говорят, что при этом происходит **коллизия** (collision), т. к. содержимое обоих кадров сталкивается на общем кабеле и происходит искажение информации – методы кодирования, используемые в Ethernet, не позволяют выделять сигналы каждой станции из общего сигнала. Для возникновения коллизии необязательно, чтобы несколько станций начали передачу абсолютно одновременно, такая ситуация маловероятна. Гораздо вероятнее, что коллизия возникает из-за того, что один узел начинает передачу раньше другого, но до второго узла сигналы первого просто не успевают дойти к тому времени, когда второй узел решает начать передачу своего кадра. То есть коллизии – это следствие распределенного характера сети.

Домен коллизий – часть сети Ethernet, все станции (узлы) которой конкурируют за единую разделяемую среду передачи, и, следовательно, каждый узел которой может создать коллизию с любым другим узлом этой части сети. Если сеть построена на повторителях или концентраторах, она является доменом коллизий. Четкое распознавание коллизий всеми станциями сети является необходимым условием корректной работы сети Ethernet. Если какая-либо передающая станция не распознает коллизию и решит, что кадр данных ею передан верно, то этот кадр данных будет утерян. Из-за наложения сигналов при коллизии информация кадра исказится, и он будет отбракован принимающей станцией (возможно из-за несовпадения контрольной суммы).

Для надежного распознавания коллизий должно выполняться следующее соотношение:

$$T_{min} >= PDV,$$

где T_{min} – время передачи кадра минимальной длины;

PDV – время, за которое сигнал коллизии успевает распространиться до самого дальнего узла сети.

Так как в худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга станциями сети (в одну сторону проходит неискаженный сигнал, а на обратном пути распространяется уже искаженный коллизией сигнал), то это время называется временем двойного оборота (Path Delay Value, PDV). При выполнении этого условия передающая станция должна успевать обнаружить коллизию, которую вызвал переданный ею кадр, еще до того, как она закончит передачу этого кадра. Выполнение этого условия зависит, с одной стороны, от длины минимального кадра и пропускной способности сети, а с другой стороны, от длины кабельной системы сети и скорости распространения сигнала в кабеле (эта скорость различна для разных типов кабеля).

Все параметры протокола Ethernet подобраны таким образом, чтобы при нормальной работе узлов сети коллизии всегда четко распознавались. При выборе параметров учитывалось и приведенное выше соотношение, связывающее между собой минимальную длину кадра и максимальное расстояние между станциями в сегменте сети.

По стандарту Ethernet минимальная длина поля данных кадра составляет 46 байт (вместе со служебными полями это дает минимальную длину кадра 64 байт, а вместе с преамбулой – 72 байт или 576 бит). Отсюда может быть определено ограничение на расстояние между станциями.

В 10-мегабитном Ethernet время передачи кадра минимальной длины (57 бит) равно 575 битовых интервалов, следовательно, время двойного оборота должно быть меньше 57,5 мкс. Расстояние, которое сигнал может пройти за это время, зависит от типа кабеля и для толстого коаксиального кабеля равно примерно 13 280 м. Учитывая, что за это время сигнал должен пройти по линии связи дважды, расстояние между двумя узлами не должно быть больше 6635 м. В стандарте величина этого расстояния выбрана гораздо меньше с учетом других более строгих ограничений. Соблюдение всех ограничений, установленных для различных стандартов физического уровня сетей Ethernet, гарантирует корректную работу сети при ее исправном состоянии. Параметры сегментов сети для различных сред передачи приведены в таблице 1.1.

Таблица 1.1 – Параметры сетевых сегментов Ethernet 10 Мбит/с

Тип среды передачи	Скорость	Длина
10Base-5	10 Мбит/с, толстый коаксиал RG-8, RG-11	500 м
10Base-2	10 Мбит/с, тонкий коаксиал RG-58	185 м
10Base-T	10 Мбит/с, неэкранированная витая пара кат. 3,4,5	100 м
10Base-FB	10 Мбит/с, оптоволоконный кабель (многомодовый)	2 км
10Base-FL	10 Мбит/с, оптоволоконный кабель	2 км
FOIRL	10 Мбит/с, оптоволоконный кабель	1 км

* FOIRL (fiber optic inter-repeater link) применяется, как и 10Base-FB, только для соединения повторителей.

Наиболее часто приходится проверять ограничения, связанные с длиной отдельного сегмента кабеля, а также количеством повторителей и общей длиной сети. Эти ограничения формулируются обычно как «правило 5-4-3» и «правило 4-х хабов». «Правило 5-4-3» означает, что в сети допустимо не более пяти сегментов, разделенных четырьмя репитерами, при этом только 3 сегмента могут быть нагруженными (т. е. к ним могут быть подключены узлы сети). «Правило 4-х хабов» состоит в том, что между любыми двумя узлами сети должно быть не более 4-х хабов (повторителей). Например, если посчитать время двойного оборота в сети, состоящей из четырех повторителей l0 Base-5 и пяти сегментов максимальный длины 500 м, то окажется, что оно составляет 537 битовых интервалов (bt, англ. bit time). А так как время передачи кадра минимальной длины вместе с преамбулой 72 байт равно 575 битовым интервалам, то видно, что разработчики стандарта Ethernet оставили 38 битовых интервала в качестве запаса для надежности. Комитет IEEE 802.3 определил, что и 4 дополнительных битовых интервала создают достаточный запас надежности. Правила «5-4-3» для коаксиальных сетей и «4-х хабов» для сетей на основе витой пары и оптоволокна не только дают гарантии работоспособности сетей, но и оставляют большой «запас прочности» сети.

Комитет IEEE 802.3 установил исходные данные о задержках, вносимых повторителями и различными средами передачи данных для тех, кто хочет самостоятельно рассчитывать максимальное количество повторителей и максимальную общую длину сети, не довольствуясь теми значениями, которые приведены в правилах «5-4-3» и «4-х хабов». Такие расчеты особенно полезны для сетей, состоящих из смешанных кабельных систем, например, коаксиала и оптоволокна. Для таких сетей правила о количестве повторителей не определены. При этом максимальная длина каждого отдельного физического сегмента должна строго соответствовать стандарту, т. е. 500 м для «толстого» коаксиального кабеля, 100 м для витой пары и т. д.

Чтобы сеть Ethernet, состоящая из сегментов различной физической природы, работала корректно, необходимо выполнение четырех основных условий:

- количество станций в сети не более 1024;
- максимальная длина каждого физического сегмента не более величины, определенной в соответствующем стандарте физического уровня;
- время двойного оборота сигнала (Path Delay Value – PDV) между двумя самыми удаленными друг от друга станциями сети не более 575 битовых интервалов;
- сокращение межкадрового интервала IPG (InterPacket Gap) при прохождении последовательности кадров через все повторители должно быть не больше чем 49 битовых интервалов. Так как при отправке кадров конечные узлы обеспечивают начальное межкадровое расстояние в 96 битовых интервалов, то после прохождения повторителя оно должно быть не меньше чем 96 – 49 = 47 (т. е. 47 битовых интервалов).

Соблюдение этих требований обеспечивает корректность работы сети даже в случаях, когда нарушаются простые правила конфигурирования, опре-

деляющие максимальное количество повторителей и общую длину сети в 2500 м.

1.1.2 Расчет времени двойного оборота сигнала PDV

Для упрощения расчетов обычно используются справочные данные IEEE, содержащие значения задержек распространения сигналов в повторителях, приемопередатчиках и различных физических средах. В таблице 1.2 приведены данные для расчета значения PDV для физических стандартов сетей Ethernet.

Комитет IEEE 802.3 старался максимально упростить выполнение расчетов, поэтому данные, приведенные в таблице 1.2, включают сразу несколько этапов прохождения сигнала. Например, задержки, вносимые повторителем, состоят из задержки входного трансивера, задержки блока повторения и задержки выходного трансивера. В таблице 1.2 все эти задержки представлены одной величиной, названной базой сегмента. Чтобы не нужно было два раза учитывать задержки, вносимые кабелем, в таблице даны удвоенные величины задержек для каждого типа кабеля.

Таблица 1.2 – Данные для расчета значения PDV

Тип сегмента	База левого сегмента, bt	База промежуточного сегмента, bt	База правого сегмента, bt	Задержка среды на 1м, bt	Максимальная длина сегмента, м
10Base-5	11,8	46,5	169,5	0,0866	500
10Base-2	11,8	46,5	169,5	0,1026	185
10Base-T	15,3	42	165	0,113	100
10Base-FB		24		0,1	2000
10Base-FL	12,3	33,5	156,5	0,1	2000
FOIRL	7,8	29	152	0,1	1000
AUI* (>2 м)	0	0	0	0,1026	50

* Интерфейс стандарта 10Base-5, соединяющий узел с шиной на «толстом» коаксиальном кабеле.

В таблице 1.2 используются также такие понятия, как левый сегмент, правый сегмент и промежуточный сегмент. Поясним эти термины на примере сети, приведенной на рисунке 1.1. Левым сегментом называется сегмент, в котором начинается путь сигнала с выхода передатчика конечного узла. На примере это сегмент 1. Затем сигнал проходит через промежуточные сегменты и доходит до приемника наиболее удаленного узла наиболее удаленного сегмента 6, который называется правым. Именно здесь в худшем случае происходит столкновение кадров и возникает коллизия, что и подразумевается в таблице 1.2. С каждым сегментом связана постоянная задержка, названная базой, которая зависит только от типа сегмента и от положения сегмента на пути сигнала (левый, промежуточный или правый). База правого сегмента, в котором возникает коллизия, намного превышает базу левого и промежуточных сегментов. Кроме этого, с каждым сегментом связана задержка распространения сигнала вдоль кабеля сегмента, которая зависит от длины сегмента и вычисляется путем

умножения времени распространения сигнала по одному метру кабеля (в битовых интервалах) на длину кабеля в метрах.

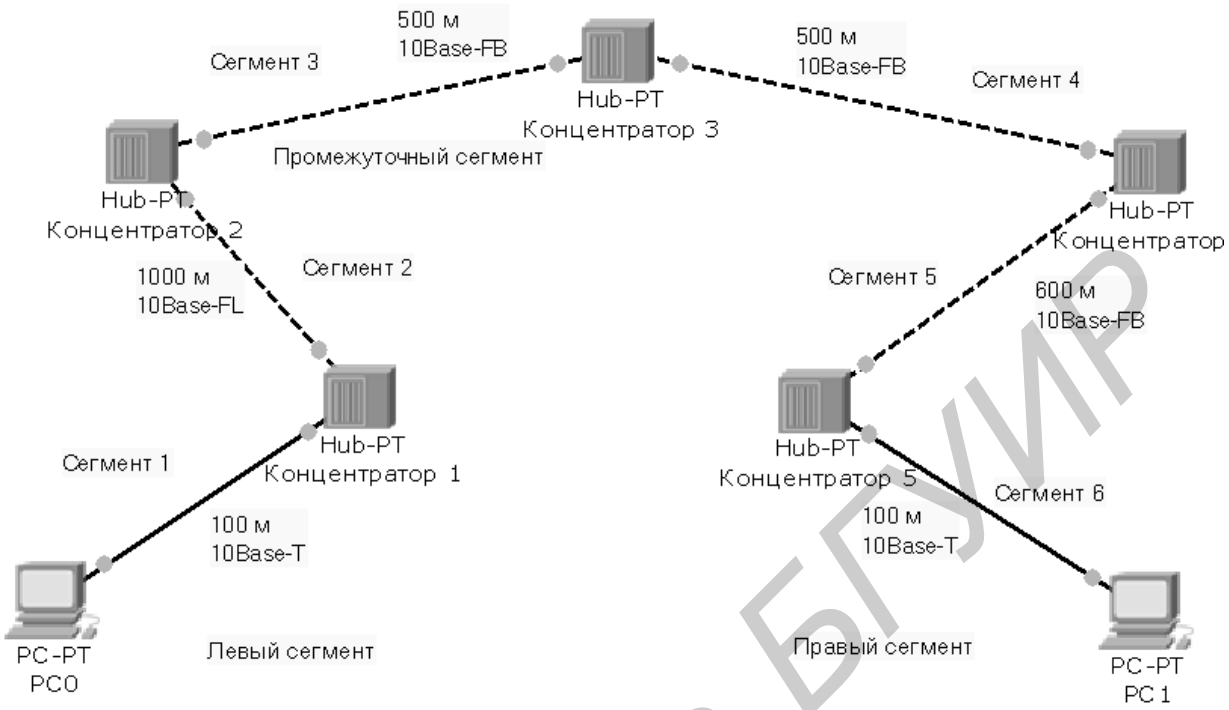

Рисунок 1.1 – Пример сети Ethernet, состоящей из сегментов различных физических стандартов

Расчет заключается в вычислении задержек, вносимых каждым отрезком кабеля (приведенная в таблице задержка сигнала на 1 м кабеля умножается на длину сегмента), а затем суммировании этих задержек с базами левого, промежуточных и правого сегментов. Общее значение PDV не должно превышать 575 bt.

Так как левый и правый сегменты имеют различные величины базовой задержки, то в случае различных типов сегментов на удаленных краях сети необходимо выполнить расчеты дважды: один раз принять в качестве левого сегмента сегмент одного типа, а затем – сегмент другого типа. Результатом следует считать максимальное значение PDV из двух полученных.

В рассматриваемом примере крайние сегменты сети принадлежат к одному типу – стандарту 10Base-T, поэтому двойной расчет не требуется, но если это сегменты разного типа, то в первом случае нужно принять в качестве левого сегмент между станцией и концентратором 1, а во втором считать левым сегмент между станцией и концентратором 5.

Приведенная на рисунке 1.1 сеть в соответствии с правилом 4-х хабов не является корректной, в сети между узлами сегментов 1 и 6 имеется 5 хабов, хотя не все сегменты являются сегментами 10Base-FB. Кроме того, общая длина сети равна 2800 м, что превышает максимальную длину 2500 м.

Пример – Рассчитаем значение PDV для сети, приведенной на рисунке 1.1.

Левый сегмент 1: 15,3 (база) + 100·0,113 = 26,6

Промежуточный сегмент 2: 33,5 + 1000·0,1 = 133,5
Промежуточный сегмент 3: 24 + 500·0,1 = 74,0
Промежуточный сегмент 4: 24 + 500·0,1 = 74,0
Промежуточный сегмент 5: 24 + 600·0,1 = 84,0
Правый сегмент 6: 165(база) + 100·0 113 = 176,3
Сумма всех составляющих дает значение PDV, равное 568,4.

Так как значение PDV меньше максимально допустимой величины 575, то эта сеть проходит по критерию времени двойного оборота сигнала, несмотря на то, что ее общая длина больше 2500 м, а количество повторителей – больше 4-х.

1.1.3 Расчет IPG

Чтобы признать конфигурацию сети корректной, нужно рассчитать также уменьшение межкадрового интервала повторителями, т. е. величину IPG (InterPacket Gap. Для расчета IPG можно воспользоваться значениями максимальных величин уменьшения межкадрового интервала при прохождении повторителей для различных физических сред, рекомендованными IEEE (таблица 1.3).

Таблица 1.3 – Сокращение межкадрового интервала повторителями

Тип сегмента	Передающий сегмент, bt	Промежуточный сегмент, bt
10Base-5	16	11
10Base-2	16	11
10Base-T	10,5	8
10Base-FL	10,5	8
10Base-FB	Не предусмотрен	2

В соответствии с этими данными рассчитаем значение IPG для рассматриваемого примера.
Левый сегмент 1 10Base-T: сокращение в 10,5 bt
Промежуточный сегмент 2 10Base-FL: 8 bt
Промежуточный сегмент 3 10Base-FB: 2 bt
Промежуточный сегмент 4 l0Base-FB: 2 bt
Промежуточный сегмент 5 10Base-FB: 2 bt
Сумма этих величин дает значение PDW, равное 24,5 bt, что меньше предельного значения в 49 битовых интервалов. В результате приведенная в примере сеть соответствует стандартам Ethernet по параметрам, связанным с длинами сегментов и с количеством повторителей.

1.2 Задание для выполнения

1 Выполните расчет параметров сети, заданной с помощью таблиц 1.3 и 1.4.
2 Оформите расчеты в виде отчета, в котором укажите параметры сети, приведите требуемые расчеты и их результаты, сформулируйте мотивированный вывод о работоспособности сети.

Таблица 1.4 – Состав сегментов рассчитываемой сети

Вари-ант	Сегмент 1 (левый)		Сегмент 2 (промеж.)		Сегмент 3 (промеж.)		Сегмент 4 (промеж.)		Сегмент 5 (промеж.)		Сегмент 6 (правый)	
	технология	длина	технология	длина	технология	длина	технология	длина	технология	длина	технология	длина
1	10Base-T	10	10Base-FL	1000	10Base-FB	690	10Base-5	470	10Base-T	10	10Base-2	100
2	10Base-2	185	10Base-T	20	10Base-FL	1600	10Base-FB	1380	10Base-5	480	10Base-T	20
3	10Base-T	30	10Base-5	490	10Base-T	30	10Base-FL	1700	10Base-FB	1200	10Base-2	130
4	10Base-2	140	10Base-FB	1100	10Base-5	500	10Base-T	40	10Base-FL	1800	10Base-T	40
5	10Base-T	50	10Base-FL	1900	10Base-FB	1650	10Base-T	50	10Base-5	450	10Base-2	150
6	10Base-2	160	10Base-T	60	10Base-FB	1250	10Base-FL	2000	10Base-5	368	10Base-T	60
7	10Base-T	70	10Base-5	350	10Base-FL	700	10Base-FB	1200	10Base-T	70	10Base-2	170
8	10Base-2	180	10Base-FB	540	10Base-T	80	10Base-FB	540	10Base-5	360	10Base-T	80
9	10Base-T	90	10Base-FL	890	10Base-5	370	10Base-T	90	10Base-FB	860	10Base-2	185
10	10Base-2	100	10Base-T	100	10Base-5	380	10Base-FB	1790	10Base-FL	750	10Base-T	100
11	10Base-T	95	10Base-T	95	10Base-FB	1905	10Base-5	390	10Base-FL	1600	10Base-2	110
12	10Base-2	100	10Base-5	400	10Base-FL	1850	10Base-FB	1600	10Base-T	100	10Base-T	100
13	10Base-T	90	10Base-FL	1200	10Base-2	170	10Base-5	455	10Base-FB	1700	10Base-2	170
14	10Base-2	180	10Base-FB	1800	10Base-5	470	10Base-T	80	10Base-FL	540	10Base-T	80
15	10Base-T	70	10Base-2	170	10Base-FL	860	10Base-5	380	10Base-FB	1900	10Base-2	170
16	10Base-2	155	10Base-T	55	10Base-5	350	10Base-T	55	10Base-FL	1790	10Base-T	55
17	10Base-T	60	10Base-5	250	10Base-FB	1200	10Base-2	60	10Base-FL	1805	10Base-2	60
18	10Base-2	65	10Base-FL	1500	10Base-5	365	10Base-FB	1100	10Base-2	65	10Base-T	65
19	10Base-T	75	10Base-FB	1650	10Base-2	175	10Base-5	430	10Base-FL	2000	10Base-2	175
20	10Base-2	65	10Base-FL	690	10Base-5	340	10Base-FB	2000	10Base-5	340	10Base-T	70
21	10Base-T	80	10Base-FB	540	10Base-T	80	10Base-5	240	10Base-FL	1380	10Base-2	180
22	10Base-2	185	10Base-T	80	10Base-FL	1200	10Base-5	400	10Base-FB	860	10Base-T	85
23	10Base-T	95	10Base-5	470	10Base-FB	1790	10Base-FL	1100	10Base-2	160	10Base-2	175
24	10Base-2	105	10Base-T	85	10Base-FL	1650	10Base-2	105	10Base-FB	1435	10Base-T	80
25	10Base-T	15	10Base-2	80	10Base-FB	350	10Base-FL	800	10Base-5	150	10Base-T	27
26	10Base-2	180	10Base-T	30	10Base-FL	1800	10Base-FB	400	10Base-5	20	10Base-2	175
27	10Base-T	25	10Base-FB	250	10Base-5	70	10Base-2	142	10Base-T	92	10Base-2	137
28	10Base-2	105	10Base-FL	80	10Base-FB	300	10Base-T	90	10Base-2	150	10Base-T	60
29	10Base-T	19	10Base-5	200	10Base-5	170	10Base-FB	1850	10Base-2	135	10Base-2	155
30	10Base-2	85	10Base-FL	1700	10Base-FB	950	10Base-FL	1250	10Base-T	70	10Base-T	88

Практическое занятие №2

МЕТОДИКА РАСЧЕТА КОНФИГУРАЦИИ СЕТИ FAST ETHERNET 100 МБИТ/С

Цель занятия: изучение методики расчета конфигурации сети Fast Ethernet 100Мбит/с, обучение применению данной методики для расчета сетей Fast Ethernet 100Мбит/с.

2.1 Теоретические сведения

2.1.1 Правила построения сегментов Fast Ethernet

Технология Fast Ethernet, как и все некоаксиальные варианты Ethernet, изначально была рассчитана на использование концентраторов-повторителей для образования связей в сети. Правила корректного построения сегментов сетей Fast Ethernet включают:

- ограничения на максимальные длины сегментов, соединяющих DTE с DTE;
- ограничения на максимальные длины сегментов, соединяющих DTE с портом повторителя;
- ограничения на максимальный диаметр сети;
- ограничения на максимальное число повторителей и максимальную длину сегмента, соединяющего повторители.

2.1.1.1 Ограничения длин сегментов DTE-DTE

В качестве DTE (Data Terminal Equipment) может выступать любой источник кадров данных для сети: сетевой адаптер компьютера, порт моста, порт маршрутизатора, модуль управления сетью и другие подобные устройства. DTE вырабатывает новый кадр для разделяемого сегмента (мост или коммутатор, хотя и передают через выходной порт кадр, который выработал в свое время сетевой адаптер, но для сегмента сети, к которому подключен выходной порт этих устройств, этот кадр является новым). Порт повторителя не является DTE, так как он побитно повторяет кадр, уже появившийся ранее в сегменте.

В типичной конфигурации сети Fast Ethernet несколько DTE подключается к портам повторителя, образуя сеть звездообразной топологии. Соединения DTE-DTE в разделяемых сегментах не встречаются (кроме редкого случая, когда сетевые адаптеры двух компьютеров соединены прямо друг с другом кабелем). Для мостов/коммутаторов и маршрутизаторов такие соединения являются нормой. Тогда сетевой адаптер прямо соединен с портом одного из этих устройств либо эти устройства соединяются друг с другом. Спецификация IEEE 802.3u определяет следующие максимальные длины сегментов DTE-DTE, приведенные в таблице 2.1.

Таблица 2.1 – Максимальные длины сегментов DTE-DTE

Тип кабеля	Длина сегмента
UTP кат. 5	100 м
Многомод. оптоволокно 62.5/125 мкм	412 м (полудуплекс), 2 км (дуплекс)
UTP кат. 3 (+4 и 5)	100 м

2.1.1.2 Ограничения сетей Fast Ethernet, построенных на повторителях

Повторители Fast Ethernet делятся на два класса. Повторители класса I поддерживают два типа логического кодирования данных: 4B/5B и 8B/6T. Повторители класса II поддерживают только какой-либо один тип кодирования – либо 4B/5B, либо 8B/6T. Повторители класса I позволяют выполнять трансляцию логических кодов с битовой скоростью 100 Мбит/с, а повторителям класса II эта операция недоступна. Поэтому повторители класса I могут иметь порты всех трех типов физического уровня: l00Base-TX, 100Base-FX и 100Base-T4. Повторители класса II имеют либо все порты 100Base-T4, либо порты 100Base-TX и 100Base-FX, т. к. последние используют один логический код 4B/5B.

В одном домене коллизий допускается наличие только одного повторителя класса I. Это связано с тем, что такой повторитель вносит большую задержку распространении сигналов из-за необходимости трансляции различных систем кодирования сигналов – 70 bt.

Повторители класса II вносят меньшую задержку при передаче сигналов: 46 bt для портов TX/FX и 33,5 bt для портов T4. Поэтому максимальное число повторителей класса II в домене коллизий – 2, причем они должны быть соединены между собой кабелем длиной до 5 метров.

Небольшое количество повторителей Fast Ethernet не является серьезным препятствием при построении больших сетей, т. к. применение коммутаторов и маршрутизаторов делит сеть на несколько доменов коллизий, каждый из которых будет строиться на одном или двух повторителях. Общая длина сети не будет иметь в этом случае ограничений.

В таблице 2.2 приведены правила построения сети на основе повторителей класса I.

Таблица 2.2 – Параметры сетей на основе повторителей класса I

Тип кабелей	Максимальный диаметр сети, м	Максимальная длина сегмента, м
Только UTP (TX)	200	100
Только опт. (FX)	272	136
Несколько TX + 1 FX	260	100 (TX), 160 (FX)
По нескольку TX и FX	272	100 (TX), 136 (FX)

Таким образом, правило 4-х хабов превратилось для технологии Fast Ethernet в правило одного или двух хабов, в зависимости от класса хаба.

При определении корректности конфигурации сети необязательно руководствоваться правилами одного или двух хабов, а рассчитывать время двойного оборота (PDV), как это делается для сети Ethernet 10 Мбит/с.

Как и для технологии Ethernet 10 Мбит/с, комитет 802.3 дает исходные данные для расчета времени двойного оборота сигнала. При этом сама форма представления этих данных и методика расчета несколько изменились. Комитет предоставляет данные об удвоенных задержках, вносимых каждым элементом сети, не разделяя сегменты сети на левый, правый и промежуточный. Кроме того, задержки, вносимые сетевыми адаптерами, учитывают преамбулы кадров, поэтому время двойного оборота нужно сравнивать с величиной 512 битовых интервала (bt), т. е. со временем передачи кадра минимальной длины без преамбулы.

Для повторителей класса I время двойного оборота рассчитывают следующим образом.

Задержки, вносимые прохождением сигналов по кабелю, рассчитываются на основании данных таблицы 2.3, в которой учитывается удвоенное прохождение сигнала по кабелю.

Таблица 2.3 – Задержки, вносимые кабелем

Удвоенная задержка в битовых интервалах	
На 1 м	На кабеле максимальной длины
1,14	114 (100 м)
1,14	114 (100 м)
1,112	111.2 (100 м)
1,112	111.2 (100 м)
1	412 (412 м)

Задержки, которые вносят два взаимодействующих через повторитель сетевых адаптера (или порта коммутатора), берутся из таблицы 2.4.

Учитывая, что удвоенная задержка, вносимая повторителем класса I, равна 140 bt, можно рассчитать время двойного оборота для произвольной конфигурации сети, учитывая максимально возможные длины непрерывных сегментов кабелей, приведенные в таблице 2.4.

Таблица 2.4 – Задержки, вносимые сетевыми адаптерами

Тип сетевых адаптеров	Максимальная задержка при полном обороте в битовых интервалах
Два адаптера TX/FX	100
Два адаптера T4	138
Один адаптер TX/FX и один T4	127

Если получившееся значение меньше 512, значит, по критерию распознавания коллизий сеть является корректной.

Комитет 802.3 рекомендует оставлять запас в 4 bt для устойчиво работающей сети, но разрешается выбирать эту величину из диапазона от 0 до 5 bt.

Пример – Рассчитаем корректность конфигурации сети, состоящей из одного повторителя и двух оптоволоконных сегментов FX длиной по 136 метров. Каждый сегмент вносит задержку по 136 bt, пара сетевых адаптеров FX дает задержку в 100 bt, а сам повторитель вносит задержку в 140 bt. Сумма задержек равна 512 bt, что говорит о том, что сеть корректна, но запас получился равным нулю.

2.2 Задание для выполнения

1 Выполните расчет параметров сети Fast Ethernet с двумя концентраторами, заданной с помощью таблицы 2.5.

2 Оформите отчет, включая схему сети, исходные данные, расчеты и мотивированный вывод о работоспособности сети.

Таблица 2.5 – Параметры сегментов рассчитываемой сети

Вариант	Сегмент 1	Конц. 1 (класс)	Сегмент 2	Конц. 2 (класс)	Сегмент 2
1	100Base-TX (90 м)	1	100Base-FX (9 м)	1	100Base-TX (60 м)
2	100Base-T4 (90 м)	1	Нет	Нет	100Base-FX (120 м)
3	100Base-TX (90 м)	Нет	Нет	1	100Base-FX (70 м)
4	100Base-T4 (60 м)	1	100Base-FX (5 м)	1	100Base-FX (90 м)
5	100Base-TX (50 м)	1	100Base-FX (10 м)	2	100Base-FX (100 м)
6	100Base-T4 (70 м)	1	100Base-FX (10 м)	2	100Base-FX (80 м)
7	100Base-TX (80 м)	Нет	Нет	2	100Base-TX (90 м)
8	100Base-T4 (60 м)	1	Нет	Нет	100Base-FX (80 м)
9	100Base-TX (50 м)	1	100Base-FX (10 м)	1	100Base-FX (80 м)
10	100Base-T4 (90 м)	1	нет	Нет	100Base-FX (90 м)
11	100Base-TX (60 м)	Нет	нет	1	100Base-FX (90 м)
12	100Base-T4 (90 м)	1	100Base-FX (5 м)	1	100Base-TX (80 м)
13	100Base-TX (80 м)	1	100Base-T4 (5 м)	2	100Base-FX (90 м)
14	100Base-T4 (90 м)	1	100Base-FX (5 м)	2	100Base-FX (70 м)
15	100Base-TX (105 м)	Нет	Нет	2	100Base-FX (70 м)
16	100Base-T4 (120 м)	1	Нет	Нет	100Base-FX (40 м)
17	100Base-TX (75 м)	1	100Base-FX (5 м)	1	100Base-FX (80 м)
18	100Base-T4 (95 м)	Нет	Нет	1	100Base-FX (80 м)
19	100Base-TX (96 м)	1	Нет	Нет	100Base-FX (87 м)
20	100Base-T4 (60 м)	1	100Base-FX (10 м)	2	100Base-FX (580 м)
21	100Base-TX (86 м)	2	100Base-TX (3 м)	1	100Base-FX (35 м)
22	100Base-T4 (90 м)	1	100Base-FX (90 м)	2	100Base-FX (90 м)
23	100Base-TX (57 м)	1	Нет	Нет	100Base-T4 (97 м)
24	100Base-T4 (90 м)	1	100Base-FX (10 м)	2	100Base-TX (120 м)
25	100Base-T4 (120 м)	Нет	Нет	2	100Base-FX (20 м)
26	100Base-T4 (150 м)	1	100Base-FX (10 м)	2	100Base-TX (50 м)
27	100Base-TX (120 м)	1	Нет	Нет	100Base-TX (185 м)
28	100Base-T4 (50 м)	1	100Base-TX (100 м)	1	100Base-T4 (70 м)
29	100Base-TX (10 м)	1	Нет	Нет	100Base-T4 (90 м)
30	100Base-T4 (30 м)	1	100Base-FX (50 м)	1	100Base-TX (50 м)

Практическое занятие №3

ПРИНЦИПЫ ПОСТРОЕНИЯ НЕБЛОКИРУЮЩИХ КОММУТИРУЕМЫХ СЕТЕЙ

Цель занятия: изучение приемов построения неблокирующих коммутируемых сетей, обучение проектированию неблокирующих коммутируемых сетей.

3.1 Теоретические сведения

Коммутатор является неблокирующим, если он может передавать через свои порты кадры с той же скоростью, с которой они поступают на эти порты.

Различают два вида неблокирующих режимов работы коммутатора – устойчивый и мгновенный. Устойчивый неблокирующий режим работы коммутатора означает, что коммутатор передает кадры со скоростью их поступления в течение любого произвольного промежутка времени. Для обеспечения такого режима нужно распределить потоки кадров по выходным портам таким образом, чтобы, во-первых, порты справлялись с нагрузкой, во-вторых, коммутатор мог всегда в среднем передать на выходы столько кадров, сколько их поступило на входы. Если же входной поток кадров (просуммированный по всем портам) в среднем будет превышать выходной поток кадров (также просуммированный по всем портам), то кадры будут накапливаться в буферной памяти коммутатора и при ее переполнении – отбрасываться. Для поддержания устойчивого неблокирующего режима работы коммутатора необходимо выполнение условия

$$C_k = (\Sigma C_{pi})/2, \qquad (3.1)$$

где C_k – производительность коммутатора,

C_{pi} – максимальная производительность протокола, поддерживаемого i-м портом коммутатора.

Суммарная производительность портов учитывает каждый проходящий кадр дважды: как входящий и как выходящий, а т. к. в устойчивом режиме входной трафик равен выходному, то достаточная производительность коммутатора для поддержки неблокирующего режима равна половине суммарной производительности портов. Если порт, например, Ethernet 10 Мбит/с, работает в полудуплексном режиме, то его производительность C_{pi} равна 10 Мбит/с, а если в дуплексном – 20 Мбит/с.

Мгновенный неблокирующий режим коммутатора означает, что коммутатор может принимать и обрабатывать кадры от всех своих портов на максимальной скорости протокола, независимо от того, обеспечиваются ли условия устойчивого равновесия между входным и выходным трафиком. Обработка некоторых кадров при этом может быть неполной, при занятости выходного порта кадр помещается в буфер коммутатора. Для поддержки мгновенного небло-

кирующего режима коммутатор должен обладать большей собственной производительностью, а именно она должна быть равна суммарной производительности его портов:

$$C_k = \Sigma C_{pi}. \quad (3.2)$$

Первый коммутатор для локальных сетей появился для технологии Ethernet. Широкому применению коммутаторов способствовало то, что их внедрение не требовало замены установленного в сетях оборудования – адаптеров, концентраторов, кабельной системы. Порты коммутаторов работали в обычном полудуплексном режиме, к ним прозрачно можно было подключить как конечный узел, так и концентратор, организующий целый логический сегмент. Так как коммутаторы и мосты прозрачны для протоколов сетевого уровня, то их появление в сети не оказывало никакого влияния на маршрутизаторы сети, если они имелись.

Коэффициент перегрузки сегмента К вычисляется как отношение суммы максимальных скоростей входящих потоков данных в некоторую часть сети или в отдельное сетевое устройство (предполагаемая максимальная нагрузка) S_{in} к имеющейся максимальной пропускной способности для этих потоков C_k внутри сети или устройства (реальная максимальная нагрузка):

$$K = \frac{\sum S_{in}}{\sum C_k}. \quad (3.3)$$

Например, если считать что у коммутатора с 24 портами Ethernet и одним портом Fast Ethernet внутренняя архитектура неблокирующая, а типичным распределением потоков будет работа всех низкоскоростных портов на один высокоскоростной порт, то коэффициент перегрузки составит $(24 \cdot 10) : 100 = 2{,}4$.

Когда величина К становится больше чем 1:1, то ситуация рассматривается как перегрузка.

Существуют различные рекомендации по максимальным значениям коэффициента перегрузки для различных частей сети. Например, компания Extreme Networks рекомендует следующие значения коэффициентов перегрузки для различных участков сетей:

- коммутация сети персональных компьютеров – 3:1;
- коммутация сегментов – 2:1;
- пул серверов – 1:1;
- магистраль – 1:1.

3.2 Задание для выполнения

Для сети, показанной на рисунке 3.1, рассчитайте пропускную способность линий связи магистрального сегмента, для того чтобы обеспечить неблокирующую коммутацию в сети. При этом учтите, что соотношение трафика в локальных сетях корпусов равно J_i, где i – номер корпуса.

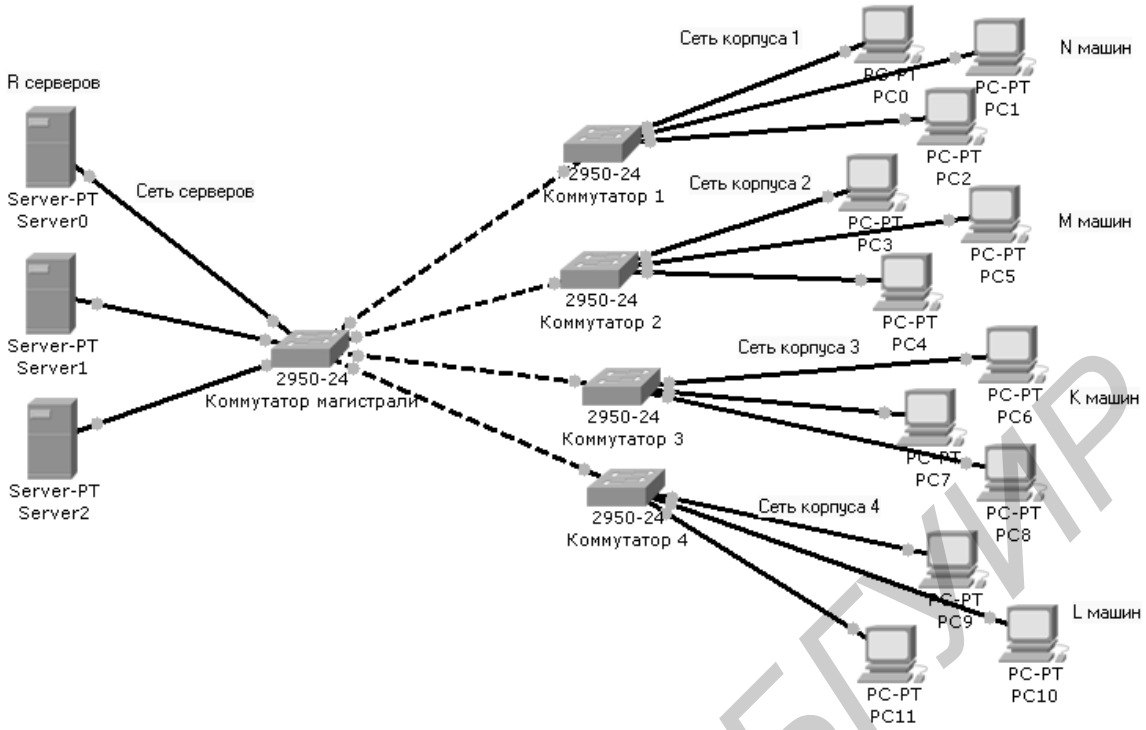

Рисунок 3.1 – Схема сети организации

Например, J_i = 20/80 означает, что 20 % трафика составляет внутренний обмен в пределах корпуса, а 80 % трафика связано с обращением пользователей к корпоративным серверам R_i.

3.3 Порядок выполнения задания

1 Занести данные варианта задания, указанного преподавателем, из таблицы 3.1 в таблицу 3.2.

Считать, что сегменты сетей корпусов используют технологию Ethernet 10 Мбит/с.

В магистральном сегменте могут использоваться технологии Ethernet 100 Мбит/с либо 1 Гбит/с (указывает преподаватель). Для обеспечения необходимой пропускной способности можно использовать технологию объединения линий в одну линию связи суммарной пропускной способности (агрегирование, или транкинг).

Варианты заданий приведены в таблице 3.1.

2 Представить в отчете:

- текст задания;
- рисунок сети организации;
- заполненную таблицу 3.2.

Таблица 3.1 – Варианты заданий

Вариант	Корпус 1		Корпус 2		Корпус 3		Корпус 4		Число серверов
	N	J_n	M	J_m	K	J_k	L	J_i	R
1	50	30/70	70	60/40	80	20/80	90	50/50	5
2	70	20/80	80	50/50	30	30/70	10	70/30	7
3	30	10/90	20	40/60	10	80/70	25	10/90	3
4	10	40/60	20	30/70	35	30/70	10	50/50	4
5	100	50/50	35	80/20	65	60/40	20	30/70	9
6	25	60/40	80	70/30	20	40/60	70	20/80	5
7	55	70/30	30	20/80	30	80/20	80	45/55	7
8	70	80/20	10	50/50	40	70/80	20	60/40	3
9	40	30/70	35	35/65	50	25/75	20	70/30	4
10	90	20/80	65	45/55	40	20/80	35	30/70	5
11	30	10/90	80	80/20	90	30/70	85	70/30	5
12	10	40/60	30	20/80	10	50/30	20	40/60	7
13	100	50/50	10	30/70	25	60/40	70	70/30	3
14	50	60/40	35	70/30	10	55/45	40	35/65	4
15	70	70/30	65	60/40	20	45/55	90	25/75	3
16	30	80/20	20	90/10	70	70/30	50	60/40	5
17	10	30/70	30	10/90	80	80/20	70	50/50	4
18	100	20/80	40	25/75	20	60/40	30	70/30	7
19	35	10/90	50	30/70	20	90/10	10	50/50	4
20	70	40/60	40	70/30	35	10/90	100	55/45	6
21	80	50/50	80	60/40	35	70/30	70	15/85	6
22	30	60/40	20	40/60	80	65/45	40	50/50	2
23	10	70/30	20	80/20	30	50/50	90	45/65	3
24	35	80/20	35	50/50	10	60/40	30	40/60	4
25	65	30/70	25	45/65	35	70/30	10	50/50	5
26	45	10/90	37	40/60	70	65/35	20	60/40	6
27	75	20/30	82	90/10	60	55/45	70	70/30	3
28	110	30/70	135	15/85	150	60/40	180	80/20	4
29	315	40/60	400	85/15	90	40/60	50	90/10	5
30	20	50/50	156	30/70	80	70/30	35	10/90	2

Примечания
1 N, M, K, L – число компьютеров.
2 R – число серверов.
3 J_i – соотношение трафика в %: внутренний/обращения к серверам.

3 Подготовить таблицу для выполнения расчетов по следующему шаблону (см. таблицу 3.2).

Таблица 3.2 – Результаты расчетов

№ сети	Количество машин	Трафик в сети	Допустимый коэффициент перегрузки	Трафик во внешней сети	Реальный трафик во внешней сети	Коэффициент загрузки линий связи	Необходимый транкинг
Корпус 1							
Корпус 2							
Корпус 3							
Корпус 4							
Магистральная сеть							
Сеть серверов							

Трафик в i-м сегменте сети T_i вычисляется как произведение $T_i = N_i \cdot P$ (P – пропускная способность сетевого подключения компьютеров сети (10, 100 Мбит/с и т. д., N_i – число машин в сети).

Допустимый коэффициент перегрузки задается типом сети. Коммутация сети ПК – 3:1. Коммутация сегментов – 2:1. Пул серверов –1:1. Магистраль – 1:1.

Трафик во внешней сети $T_{iвн}$ вычисляется как

$$T_{iвн} = T_i \cdot J_{iвн},$$

где $T_{iвн}$ – трафик из сегмента во внешнюю сеть;

T_i – трафик в i-м сегменте сети;

$J_{iвн}$ – доля внешнего трафика.

Реальный трафик во внешней сети $T_{iвн.р}$ определяется как

$$T_{iвн.р} = T_{iвн} \cdot K_п,$$

где $T_{iвн.р}$ – реальный трафик во внешнюю сеть;

$T_{iвн}$ – расчетный трафик;

$K_п$ – коэффициент перегрузки для данного типа сегмента.

Для магистральной сети поля заполняются путем суммирования соответствующих значений для отдельных корпусов.

Для сети серверов заполняется только поле «Реальный трафик во внешней сети».

Необходимый транкинг определяется исходя из величины перегрузки сети.

4 Сделать вывод и обосновать принятые решения.

Практическое занятие №4

УПРАВЛЕНИЕ АДРЕСНЫМ ПРОСТРАНСТВОМ IP-СЕТЕЙ

Цель занятия: изучение правил распределения адресного пространства IP-сетей.1

4.1 Теоретические сведения

4.1.1 Использование масок в IP-адресации

Традиционная схема деления IP-адреса на номер сети и номер узла основана на понятии класса, который определяется значениями нескольких старших (называемых начальными, начало при этом отсчитывается от старшего разряда) битов адреса в соответствии с рисунком 4.1. Эти биты называют также префиксом.

Количество начальных битов префикса	1	7	24
Класс A: значение префикса	0	Сетевые биты	Биты узла

Количество начальных битов префикса	2	14	16
Класс B: значение префикса	10	Сетевые биты	Биты узла

Количество начальных битов префикса	3	21	8
Класс C: значение префикса	110	Сетевые биты	Биты узла

Количество начальных битов префикса	4	28
Класс D: значение префикса	1110	Адрес

Количество начальных битов префикса	4	28
Класс E: значение префикса	1111	Адрес

Рисунок 4.1 – Начальные биты, образующие классы IP адресов

Обычно вместо двоичной записи адреса его записывают в виде набора из четырех десятичных чисел (соответствующим 8-разрядным частям 32-разрядного двоичного числа). Такой способ написания адреса называется **точечно-десятичным форматом**. В таком виде каждый IP-адрес состоит из четырех частей, разделенных точками. Каждая из частей называется **октетом**, поскольку состоит из восьми двоичных цифр. Октет по сути представляет собой один байт.

Количество и значения битов в префиксах каждого класса фиксированы и не могут изменяться. Поэтому число в первом (старшем) байте адреса не может принимать всех возможных значений от нуля до 255.

Именно потому, что первый байт адреса 185.23.44.206 попадает в диапазон 128–191, можно сказать, что этот адрес относится к классу B, значит, номером сети являются первые два байта, дополненные двумя нулевыми байтами – 185.23.0.0, а номером узла – 0.0.44.206.

Другим способом определения границы между номером сети и номером узла в IP-адресе является использование масок. Маска – это число, которое используется в паре с IP-адресом: двоичная запись маски содержит единицы в разрядах, которые должны интерпретироваться как номер сети, а нули – в разрядах, интерпретируемых как номер узла. Поскольку номер сети является цельной частью адреса, единицы в маске должны представлять непрерывную последовательность.

Для стандартных классов сетей маски имеют следующие значения:

- класс A – 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс B – 1111 111 1. 11111111. 00000000. 00000000 (255.255.0.0);
- класс C – 11111111.11111111.11111111.00000000 (255.255.255.0).

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации. Например, если рассмотренный выше адрес 185.23.44.206 ассоциировать с маской 255.255.255.0, то номером сети будет 185.23.44.0, а не 185.23.0.0, как это определено системой классов. В масках количество единиц в последовательности, определяющей границу номера сети, необязательно должно быть кратным 8, чтобы повторять деление адреса на байты. Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, т. е. в двоичном виде:

- IP-адрес 129.64.134.5 – 10000001.01000000.10000110. 00000101;
- маска 255.255.128.0 – 11111111.11111111.10000000. 00000000.

Если игнорировать маску, то в соответствии с системой классов адрес 129.64.134.5 относится к классу B, а значит, номером сети являются два первых байта – 129.64.0.0, а номером узла – 0.0.134.5.

Если же использовать для определения границы номера сети маску, то 11 последовательных единиц в маске, «наложенные» на IP-адрес, дадут в качестве номера сети число:

100000001. 01000000. 10000000. 00000000 или в десятичной форме записи – номер сети 129.64.128.0, а номер узла 0.0.6.5.

Механизм масок широко распространен в IP-маршрутизации. С их помощью администратор может структурировать свою сеть, не требуя от поставщика услуг дополнительных номеров сетей. На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения так называемых «префиксов» с целью уменьшения объема таблиц маршрутизации и повышения за счет этого производительности маршрутизаторов.

4.1.2 Порядок распределения IP-адресов

Номера сетей назначаются либо централизованно, если сеть является частью Интернет, либо произвольно, если сеть работает автономно. Номера узлов при этом администратор может назначать по своему усмотрению, не выходя из разрешенного для этого класса сети диапазона.

Координирующую роль в централизованном распределении IP-адресов до некоторого времени играла организация InterNIC, однако с ростом сети задача

распределения адресов стала слишком сложной, и InterNIC делегировала часть своих функций другим организациям и крупным поставщикам услуг Интернет (региональным и локальным регистраторам).

Уже сравнительно давно возник дефицит IP-адресов. В феврале 2011 года было официально объявлено об исчерпании лимита адресов IPv4. При этом надо отметить, что дефицит обусловлен не только ростом сетей, но и тем, что имеющееся множество IP-адресов используется нерационально. Очень часто владельцы сети класса С расходуют лишь небольшую часть из имеющихся у них 254 адресов. Рассмотрим пример, когда две сети надо соединить глобальной связью. В таких случаях для связи используют два маршрутизатора, соединенных по схеме «точка – точка» (рисунок 4.2).

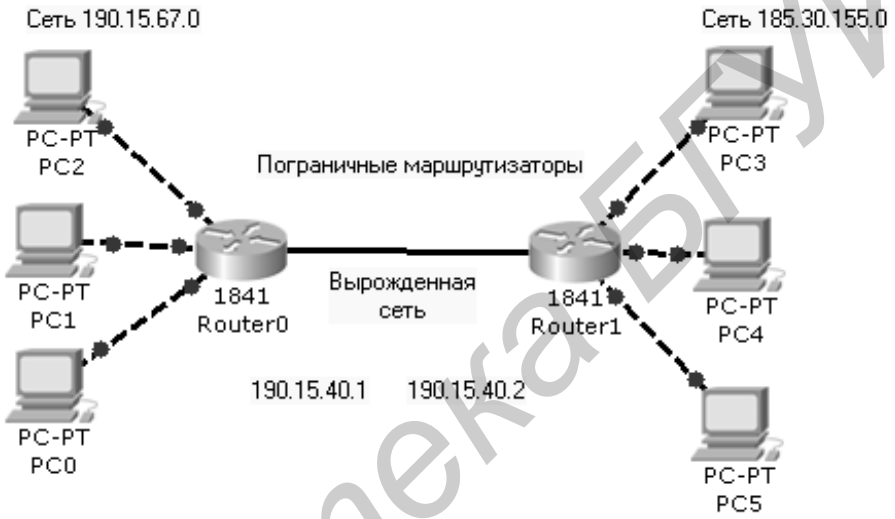

Рисунок 4.2 – Пример соединения двух сетей глобальной связью

Для вырожденной сети, образованной каналом, связывающим порты двух смежных маршрутизаторов, приходится выделять отдельный номер сети, хотя в ней имеются всего два узла.

Если же некоторая IP-сеть создана для работы в «автономном режиме» без связи с Интернет, тогда администратор этой сети может назначить ей произвольно выбранный номер. Чтобы избежать каких-либо коллизий, в стандартах Интернет определено несколько диапазонов адресов, рекомендуемых для локального использования (частные адреса). Эти адреса не обрабатываются маршрутизаторами Интернет ни при каких условиях. Адреса, зарезервированные для локальных целей, выбраны из разных классов (таблица 4.1).

Таблица 4.1 – Блоки IP-адресов для использования в частных сетях

Сеть	Маска	Блок
10.0.0.0	255.0.0.0	1 сеть класса А
172.16.0.0	255.255.0.0	16 сетей класса В
192.168.0.0	255.255.255.0	256 сетей класса С

4.1.3 Технология VLSM

Для смягчения проблемы дефицита адресов разработчики стека TCP/IP предлагают разные подходы. Принципиальным решением является переход на новую версию IPv6, в которой резко расширяется адресное пространство за счет использования 16-байтных адресов. Однако и текущая версия IPv4 поддерживает некоторые технологии, направленные на более экономное расходование IP-адресов. Одной из таких технологий является технология масок переменной длины (VLSM – Variable Length of Subnet Mask) и ее развитие – технология бесклассовой междоменной маршрутизации (Classless Inter-Domain Routing – CIDR). Технология CIDR отказывается от традиционной концепции разделения адресов протокола IP на классы, что позволяет получать в пользование столько адресов, сколько реально необходимо. Благодаря CIDR, поставщик услуг получает возможность «нарезать» блоки из выделенного ему адресного пространства в соответствии с требованиями каждого клиента, при этом у него остается пространство для маневра на случай будущего роста.

4.1.4 Технология NAT

Другая технология, которая может быть использована для смягчения дефицита адресов, – трансляция адресов (Network Address Translation – NAT). Узлам внутренней сети адреса назначаются произвольно (в соответствии с общими правилами, определенными в стандарте), как будто эта сеть работает автономно. Внутренняя сеть соединяется с Интернет через некоторое промежуточное устройство (маршрутизатор, межсетевой экран). Это промежуточное устройство получает в свое распоряжение некоторое количество внешних глобально-уникальных IP-адресов, согласованных с поставщиком услуг или другой организацией, распределяющей IP-адреса. Промежуточное устройство способно преобразовывать внутренние адреса во внешние, используя для этого таблицы соответствия. Для внешних пользователей все многочисленные узлы внутренней сети выступают под несколькими внешними IP-адресами. При получении внешнего запроса это устройство анализирует его содержимое и при необходимости пересылает его во внутреннюю сеть, заменяя IP-адрес на внутренний адрес этого узла. Процедура трансляции адресов определена в RFC 1631.

4.1.5 Бесклассовая междоменная маршрутизация (CIDR)

Бесклассовая междоменная маршрутизация (Classless Inter-Domain Routing – CIDR) – это промышленный стандарт назначения числа битов подсети, используемых в IP-адресе хоста или сети. Если, например, есть адрес 172.16.10.1 и маска 255.255.255.0, то вместо того чтобы записывать их порознь, можно их совместить. Например, запись 172.16.10.1/24 означает, что маска подсети содержит в себе 24 бита из 32-х, равных единице. Число 24 называют префиксом CIDR.

4.1.6 Маски подсети переменной длины (VLSM)

Основной задачей VLSM является сохранение IP-адресов. Например, если имеется адрес сети класса С и необходимо иметь 14 подсетей, 10 из которых являются глобальными каналами, использующими только два IP-адреса, то придется впустую потратить большой объем адресного пространства, если все интерфейсы на маршрутизаторе будут использовать одну и ту же маску.

Список всех возможных вариантов масок и префиксов CIDR

Маски

255.0.0.0=/8	255.252.0.0=/14	255.255.192.0=/18	255.255.255.192=/26
255.128.0.0=/9	255.254.0.0=/15	255.255.224.0=/19	255.255.255.224=/27
255.192.0.0=/10	255.255.252.0=/22	255.255.240.0=/20	255.255.255.240=/28
255.224.0.0=/11	255.255.254.0=/23	255.255.248.0=/21	255.255.255.248=/29
255.240.0.0=/12	255.255.0.0=/16	255.255.255.0=/24	255.255.255.252=/30
255.248.0.0=/13	255.255.128.0=/17	255.255.255.128=/25	

Список значений префикса CIDR начинается с минимального значения 8. Поскольку необходимо оставить минимум два бита для хостов (2 адреса для машин, адрес сети и широковещательный адрес сети), префикс не может превосходить 30.

4.1.7 Порядок создания VLSM-масок

Для того чтобы создавать VLSM-маски быстро и эффективно, требуется знать, каким образом можно совместно использовать для их создания размеры блоков и диаграммы. В таблице 4.2 показаны размеры блоков, используемые для создания VLSM-масок для сетей класса С.

Таблица 4.2 – Размеры блоков для создания VLSM-масок для сетей класса С

Префикс	Маска	Размеры блоков	Число хостов	Префикс	Маска	Размеры блоков	Число хостов
/30	255.255.255.252	4	2	/18	255.255.192.0	16384	16382
/29	255.255.255.248	8	6	/17	255.255.128.0	32768	32766
/28	255.255.255.240	16	14	/16	255.255.0.0	65536	65534
/27	255.255.255.224	32	30	/15	255.254.0.0	131072	131070
/26	255.255.255.192	64	62	/14	255.252.0.0	262144	262142
/25	255.255.255.128	128	126	/13	255.248.0.0	524288	524286
/24	255.255.255.0	256	254	/12	255.240.0.0	1048576	1048574
/23	255.255.254.0	512	510	/11	255.224.0.0	2097152	2097150
/22	255.255.252.0	1024	1022	/10	255.192.0.0	4194304	4194302
/21	255.255.248.0	2048	2046	/9	255.128.0.0	8388608	8388606
/20	255.255.240.0	4096	4094	/8	255.0.0.0	16777216	16777214
/19	255.255.224.0	8192	8190	–	–	–	–

Например, если требуется иметь 25 хостов, то необходим блок размером 32. При необходимости иметь 11 хостов размер используемого блока составит 16 и т. д. Далее необходимо создать VLSM-таблицу. На рисунке 4.3 по-

казаны три шага, которые требуется выполнить для создания VLSM-таблицы с помощью диаграмм.

Можно завершить создание диаграммы, выполнив шаги 4 и 5, которые разобьют таблицу на группы по 8 и по 4 адреса, что окажется полезным для соединений с глобальными сетями.

Далее возьмем размер нашего блока и VLSM-диаграмму и попробуем создать VLSM-таблицу, использующую адреса сетей класса C для сети, изображенной на рисунке 4.4.

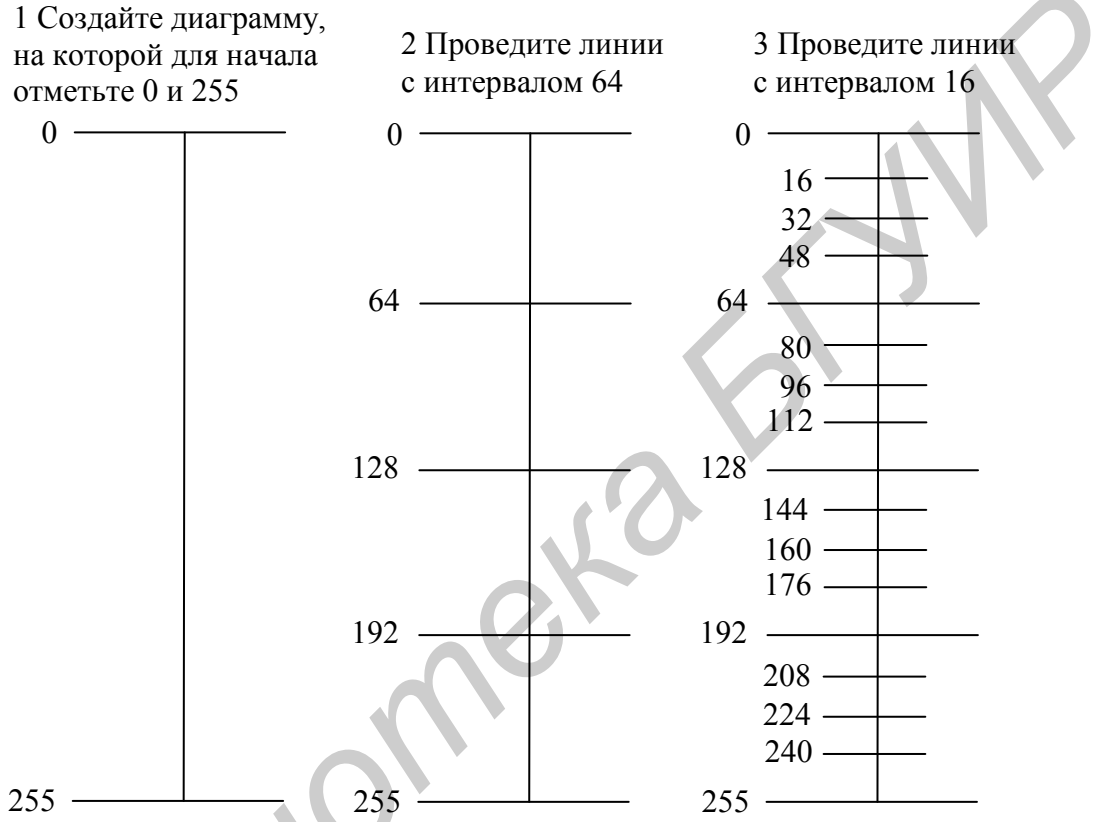

Рисунок 4.3 – Три шага построения диаграммы для создания VLSM-таблиц

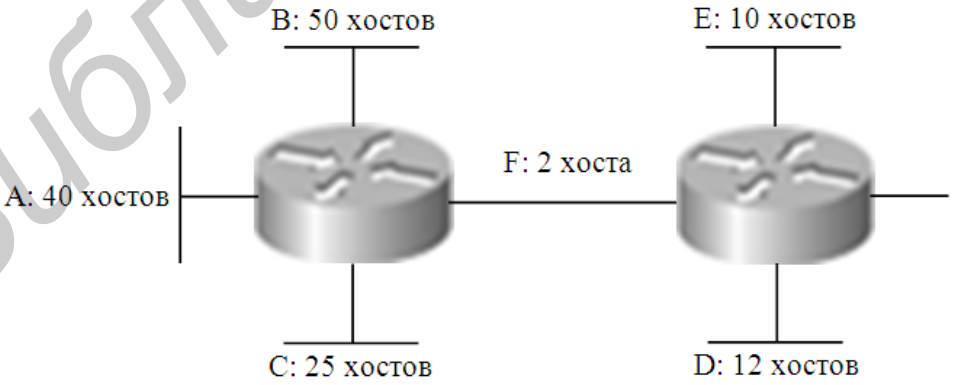

Рисунок 4.4 – Пример сети для выделения IP-адресов с использованием VLSM

Следующим шагом будет заполнение VLSM-таблицы, как показано на рисунке 4.5.

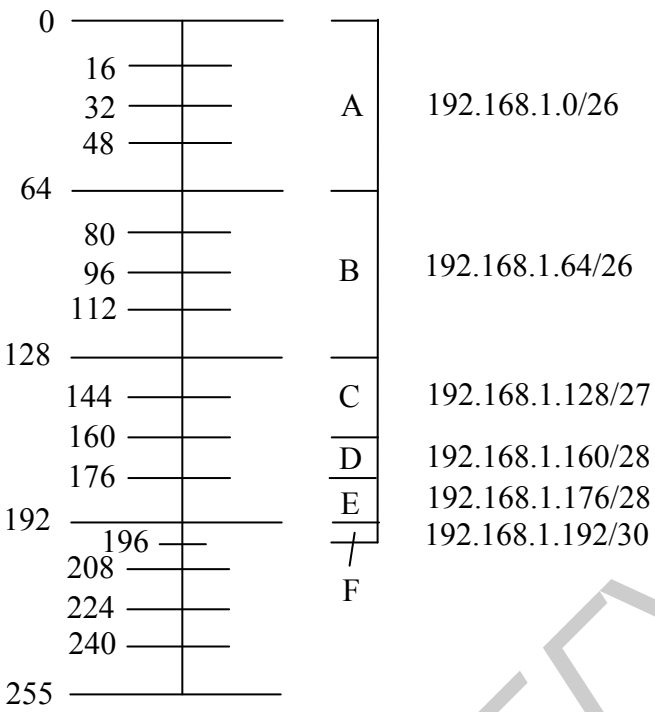

Рисунок 4.5 – Пример VLSM-таблицы

4.2 Задание для выполнения

1 Распределите адресное пространство для сети, топология которой приведена на рисунке 4.6. Параметры сети заданы в таблице 4.3.

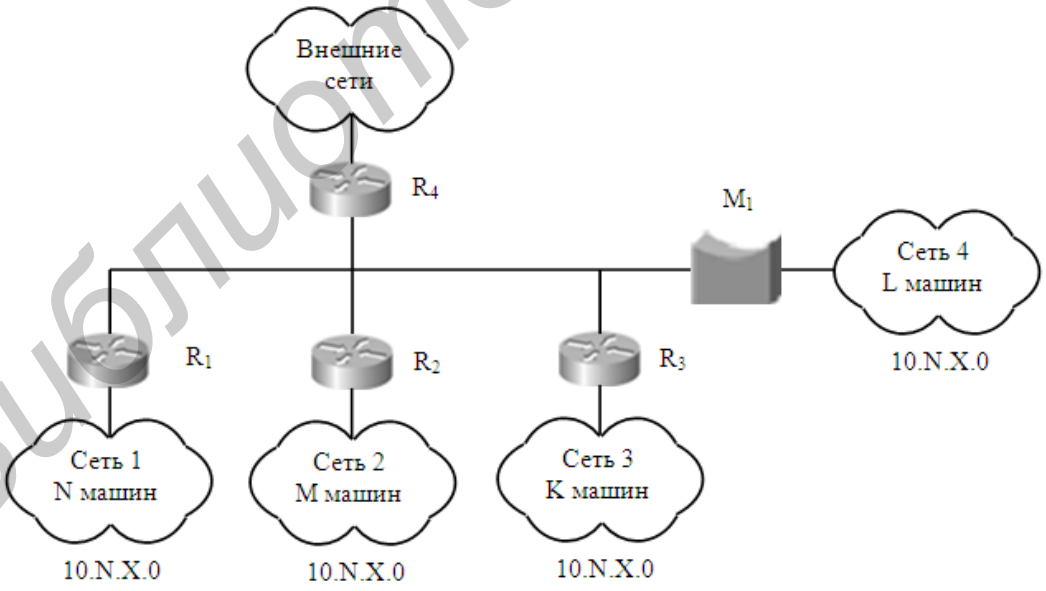

Рисунок 4.6 – Схема сети

Таблица 4.3 – Параметры сети

Вариант	N	M	K	L
1	50	70	300	900
2	1000	200	600	80
3	2000	3000	30	500
4	1500	700	150	2000
5	90	1500	2000	800
6	200	2000	1500	900
7	2000	200	900	1500
8	70	900	200	900
9	300	70	700	200
10	900	300	70	700
11	50	200	300	70
12	10	50	900	300
13	10000	10	50	900
14	750	10000	10	50
15	1500	750	1000	10
16	2000	900	750	10000
17	10	2000	200	750
18	150	370	2000	100
19	900	700	300	2000
20	35	550	780	1150
21	350	55	720	135
22	350	210	1300	600
23	510	350	780	450
24	800	710	350	60
25	750	15	255	350
26	800	375	90	1100
27	520	75	190	55
28	420	990	55	315
29	270	55	115	75
30	55	800	620	380

2 Результаты выполнения задания оформите в виде таблиц и краткого описания принятых проектных решений. Результаты выполнения пункта 1 запишите в отчет по практической работе.

Практическое занятие №5

ТЕХНОЛОГИЯ ETHERNET. ФОРМАТЫ КАДРОВ ETHERNET

Цель занятия: ознакомление с работой сетевого анализатора; обучение применению сетевого анализатора для исследования трафика локальных сетей, изучение форматов кадров Ethernet.

5.1 Теоретические сведения

5.1.1 Формат кадров Ethernet

Технология Ethernet является старейшей и наиболее популярной в мире технологией локальных сетей. В ходе эволюционного развития она постоянно совершенствовалась и развивалась, чем и определяется ее широкое использование. В ходе ее развития появилось несколько вариантов формата кадров Ethernet (рисунок 5.1).

Кадр 802.3/LLC

6	6	2	1	1	1 (2)	46-1497 (1496)	4
DA	SA	L	DSAP	SSAP	Control	Data	FCS
			Заголовок LLC				

Кадр Raw 802.3/Novell 802/3

6	6	2	46-1500	4
DA	SA	L	Data	FCS

Кадр Ethernet DIX (II)

6	6	2	46-1500	4
DA	SA	T	Data	FCS

Кадр Ethernet SNAP

6	6	2	1	1	1	3	2	46-1497 (1496)	4
DA	SA	L	DSAP	SSAP	Control	OUI	T	Data	FCS
			AA	AA	03	000000			
			Заголовок LLC			Заголовок SNAP			

Рисунок 5.1 – Форматы кадров Ethernet

Сокращения и обозначение полей кадров приведены в таблице 5.1.

Таблица 5.1 – Обозначение полей кадров Ethernet

Обозначение поля	Значение поля
DA	Destination Address – MAC адрес станции назначения
SA	Source Address – MAC адрес станции-источника
L	Length – длина поля данных кадра
T	Type – тип протокола сетевого уровня

Продолжение таблицы 5.1

Обозначение поля	Значение поля
DSAP	Destination Service Access Point – адрес точки входа службы назначения
SSAP	Source Service Access Point – адрес точки входа службы источника
Control	Управляющее поле. В зависимости от содержимого поля Control все кадры уровня LLC подразделяются на три типа: - информационные (1-кадры) - управляющие (S-кадры) - ненумерованные (U-кадры)
1 OUI	Поле OUI – уникальный идентификатор организации: 000000 – IEEE
1 Data	Поле данных
PCS	Frame Check Sequence – поле контрольной суммы

Для обеспечения совместимости сетевое оборудование должно уметь распознавать все четыре типа кадров. Алгоритм распознавания кадров приведен на рисунке 5.2.

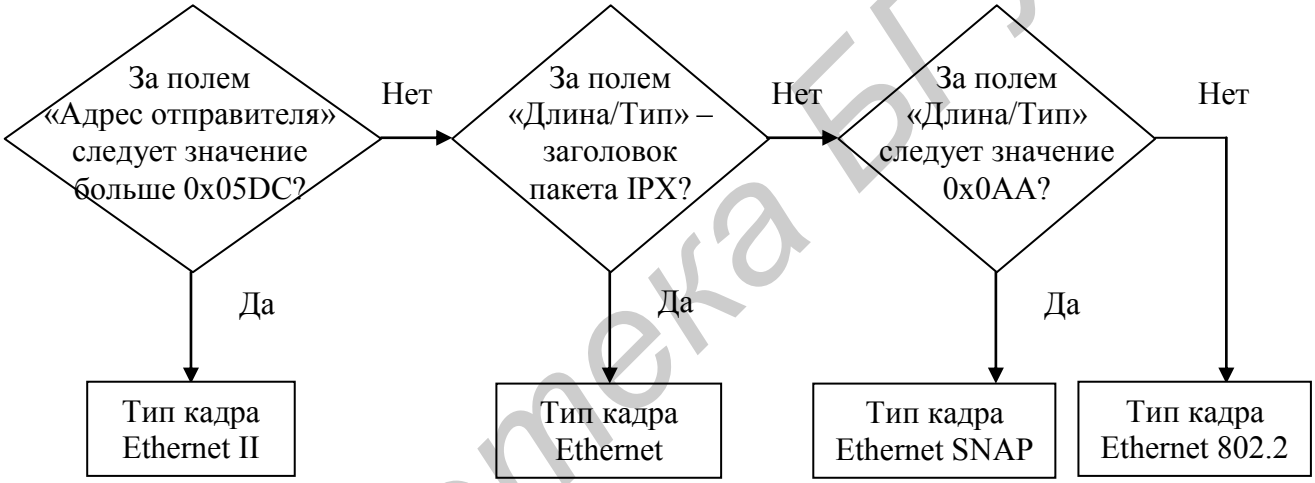

Рисунок 5.2 – Алгоритм определения формата кадров Ethernet

5.1.2 Назначение и принцип работы сетевого анализатора

Для анализа циркулирующих в сети кадров удобно использовать сетевой анализатор (network sniffer). Он прослушивает (sniff – англ. нюхать) пакеты, циркулирующие в сети. Для работы анализатора требуется, чтобы сетевой интерфейс работал в режиме захвата всех пакетов, поступающих на его порт. Этот режим получил название **promiscuous mode** – режим беспорядочного захвата. Он позволяет собирать трафик, циркулирующий в сети, проводить его анализ на соответствие некоторым шаблонам (фильтрам), выявлять подозрительную активность, изучать текущее состояние сети и выполнять ряд других работ, связанных с поддержанием сети в рабочем состоянии.

Если обратиться к эталонной модели ISO OSI, то анализаторы инспектируют два нижних уровня – физический и канальный. На канальном уровне происходит первоначальное кодирование данных для передачи через конкретную среду. Поэтому анализаторы являются сетезависимыми – они зависят от типа

сети, в которой они работают. Например, для анализа трафика в сети Ethernet необходимо иметь анализатор, способный обрабатывать кадры Ethernet.

К основным функциям сетевых анализаторов относятся:

- захват сетевых пакетов в реальном времени;
- анализ ошибок физического уровня (требует применения сертифицированных сетевых адаптеров);
- отображение статистической информации в реальном времени;
- визуализация трафика в виде диаграмм, графов, таблиц и т. д. (необязательная опция);
- декодирование протоколов, которое значительно облегчает анализ собранной информации;
- анализ полнодуплексных коммутируемых сегментов Fast Ethernet (требуется внешний адаптер Fast Ethernet Full Duplex).

Существуют методы декодирования протоколов, с помощью которых можно просматривать заголовки и содержимое сетевых пакетов в удобочитаемом виде. Это позволяет анализировать перехваченные кадры, обнаруживать неисправности в сравнительно небольших сетях, определять аномальное поведение сетевых станций.

5.1.3 Особенности применения сетевых анализаторов. Анализ сетевого трафика и информационная безопасность

Несанкционированный анализ сети является одним из видов пассивных сетевых атак. Так как большинство систем не шифрует свой трафик в локальной сети, то, перехватывая данные, передаваемые по сети, вполне возможно подсмотреть пароли для различных систем, содержимое почтовых сообщений и другие критичные данные. Поэтому, как правило, сетевые анализаторы запрещается устанавливать на машины рядовых пользователей сети, а обнаруженная активность такого рода пресекается с максимальной быстротой. ***Прежде чем приступить к выполнению задания, получите разрешение у преподавателя на использование сетевого анализатора.***

Перед использованием анализатора необходимо детально изучить физическую и логическую организацию сети. Проводя анализ в неправильном месте сети, можно либо получить ошибочные результаты, либо не увидеть то, что необходимо. Нужно удостовериться, что между анализирующей рабочей станцией и наблюдаемым участком сети нет маршрутизаторов. Аналогично в коммутируемой сети понадобится сконфигурировать порт, к которому подключается анализатор, как порт «монитора» или «зеркала». Производители используют различную терминологию, но, по сути, необходимо, чтобы данный порт действовал как концентратор – иначе будет «виден» не весь трафик.

Одним из широко распространенных анализаторов является Ethereal. Этот анализатор приобрел широкую популярность в среде UNIX-систем. Имеется его версия для Windows. К числу достоинств Ethereal относятся ясный формат вывода, поддержка большого числа форматов протоколов (более 300), большого

числа физических форматов сетей, возможность интерактивно просматривать и сортировать данные, наличие режима фильтрации вывода с широкими возможностями, включая выделение цветом некоторых пакетов. Имеется графический интерфейс создания фильтра, облегчающий данный процесс.

Ethereal позволяет следить за потоком TCP и просматривать его содержимое в текстовом виде. Данная возможность позволяет оперативно следить за общением между взаимодействующими узлами. Программа может работать с рядом внешних программ и библиотек перехвата.

Независимо от применяемой версии, Windows или Linux, почти все операции и интерфейсы схожи. После запуска Ethereal открывается экран с тремя зонами (рисунок 5.3). В этих окнах отображаются перехваченные данные и другая информация о трафике.

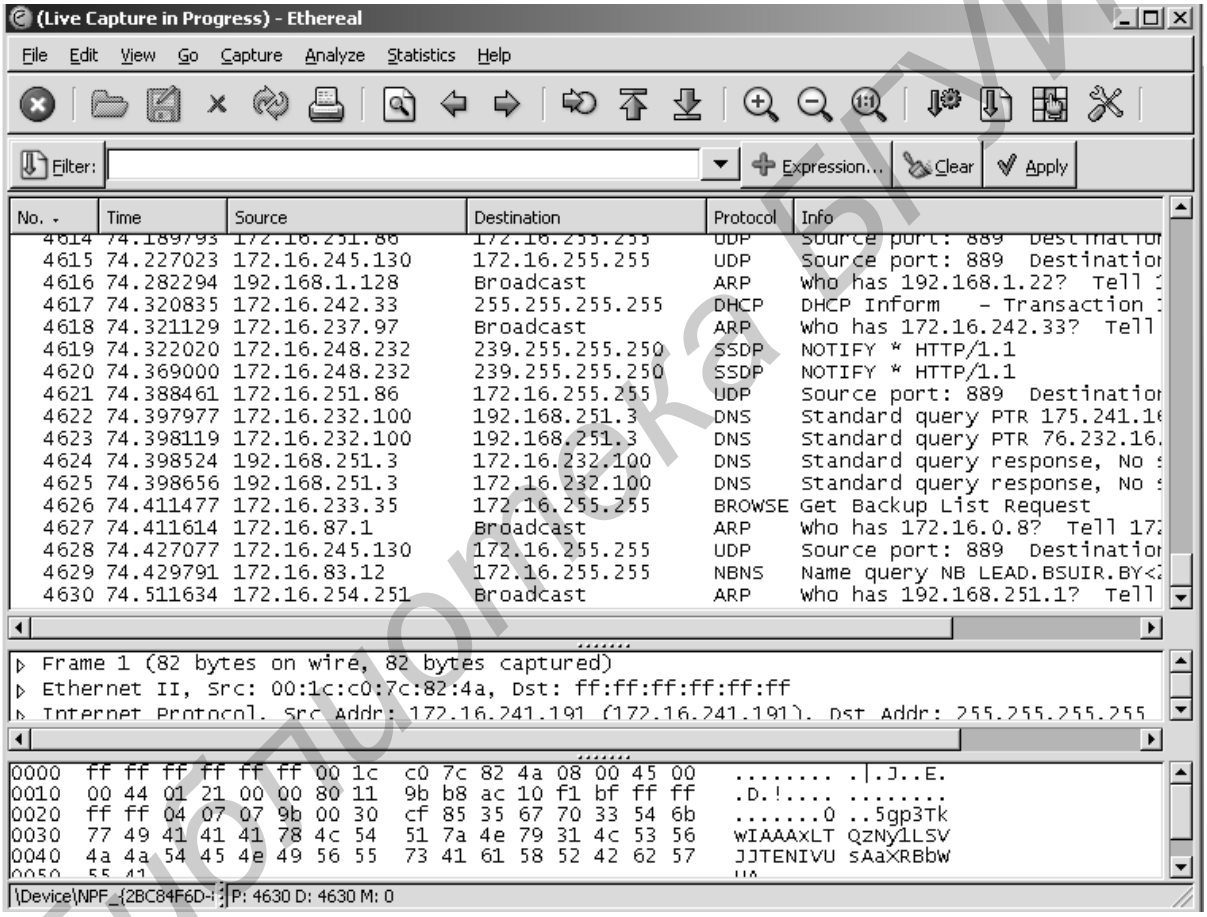

Рисунок 5.3 – Рабочее окно программы Ethereal

В верхней части экрана выводится поток пакетов в порядке получения, хотя результаты можно отсортировать почти любым образом, щелкая мышью на заголовках колонок. В таблице 5.2 перечислены выводимые данные для каждого пакета или кадра. В средней части экрана более детально отображается каждый выделенный пакет. Вывод организован таким образом, чтобы в целом соответствовать модели OSI, поэтому сначала идут детали канального уровня, затем сетевого и т. д. Небольшие символы «+» можно раскрыть, чтобы отобразить еще больше информации на каждом уровне.

Таблица 5.2 – Данные, выводимые Ethereal в окне собранных пакетов

Данные	Описание
Номер пакета	Присваивается Ethereal
Время (Time)	Время получения пакета. По умолчанию оно устанавливается как время, прошедшее с начала сеанса перехвата, но можно сконфигурировать вывод астрономического времени, даты и времени или даже интервалов между пакетами (это полезно для анализа функционирования сети)
Исходный адрес (Source)	Адрес, откуда пришел пакет. В IP-сетях это IP-адрес, в локальных сетях Ethernet – MAC-адрес
Целевой адрес (Destination)	Адрес, куда направляется пакет, также обычно IP-адрес
Протокол (Protocol)	Протокол четвертого уровня, используемый пакетом
Информация (Info)	Некоторая сводная информация о пакете

В нижнем разделе отображается реальное содержимое пакета в шестнадцатеричном и, где возможно, в текстовом виде. Двоичные файлы и зашифрованный трафик по-прежнему будут выглядеть как нечитаемый текст, но весь открытый текст будет виден.

5.1.4 Запуск открытого сеанса перехвата

Освоение программы можно начать с максимально открытого сеанса перехвата. Для этого нужно выбрать крайнюю левую кнопку в панели инструментов Ethereal. Появится окно опций (рисунок 5.4).

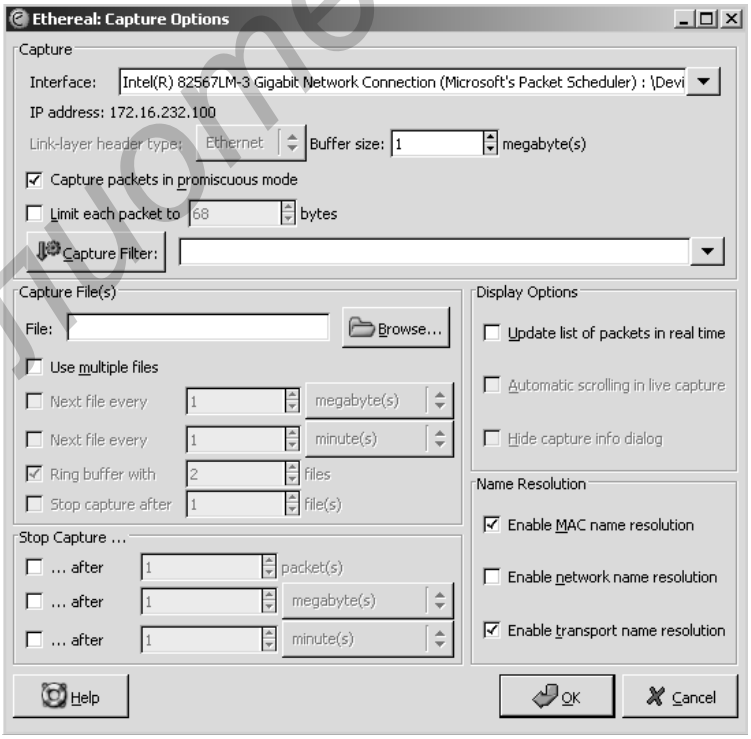

Рисунок 5.4 – Окно опций

5.1.5 Опции режима перехвата

Перечень всех доступных опций приведен в таблице 5.3.

Таблица 5.3 – Опции режима перехвата

Опция	Описание
Interface (Интерфейс)	Выбирает сетевой интерфейс для перехвата из выпадающего меню. Ethereal автоматически определяет все доступные интерфейсы и выдает их. Можно также задать одновременный перехват на всех интерфейсах, совсем как в Tcpdump
Limit each packet to x bytes (Ограничить каждый пакет x байтами)	Задает максимальный размер перехватываемых пакетов. Это полезно, если есть вероятность, что некоторые из пакетов могут быть очень большими
Capture packets in promiscuous mode (Перехват пакетов в режиме прослушивания)	Подразумеваемая опция. Выключите ее, если хотите перехватывать только потоки данных, направленные в вашу машину-анализатор
Filter (Фильтр)	Позволяет создать фильтр, используя выражения в стиле Tcpdump. Следует задать имя фильтра (которое можно использовать в будущих сеансах) и ввести выражение
Capture file(s) (Файлы перехвата)	Кнопка File используется, если необходимо читать данные из файла, а не перехватывать их «вживую»
Display options (Опции отображения)	По умолчанию отключены, но их можно включить, если нужно наблюдать движение пакетов в реальном масштабе времени. Отображение весьма полезно, если нужно понаблюдать за трафиком, чтобы получить общее представление о природе потоков данных в сети
Capture limits (Пределы перехвата)	Здесь представлено несколько дополнительных опций для задания условий завершения перехвата. Помимо остановки вручную, можно заставить Ethereal остановиться после перехвата некоторого числа x пакетов или килобайт данных или после того, как пройдет определенное число секунд
Name resolution (Разрешение имен)	Можно указать, должен или нет Ethereal разрешать имена на различных уровнях сетевой модели. Можно выборочно разрешать имена MAC-адресов, сетевые имена (SMB или имена хостов) и/или имена транспортного уровня. Включение этих опций, особенно DNS, может существенно замедлить перехват

После установки опций щелкните мышью на ОК, и сеанс начнется. Появится окно (см. рисунок 5.4), в котором в реальном масштабе времени отображается статистика сеанса. Если сеанс настроен для показа пакетов в реальном времени, вы будете наблюдать их в окне по мере того, как они проходят по среде передачи. Сеанс можно остановить в любое время, щелкнув мышью на кнопке Stop в окне статистики или выбрав Stop в меню Capture.

5.1.6 Просмотр и анализ результатов

Щелкая клавишей мыши на заголовках вверху окна, можно переупорядочить результаты по этому заголовку так, что можно сортировать вывод по ис-

ходным и целевым адресам, протоколу или информационному полю. Это помогает организовать данные, если требуется найти трафик определенного вида, например, все запросы DNS или весь трафик, связанный с почтой. В Ethereal имеется возможность создать фильтр для отображения трафика определенного вида. Для этого используется поле «Filter» в рабочем окне программы (см. рисунок 5.3).

5.1.7 Опции отображения результатов перехвата

В таблице 5.4 перечислены команды из меню Display, при помощи которых можно воздействовать на способ отображения пакетов на экране.

Таблица 5.4 – Опции отображения

Пункт меню	Описание
Подменю Options	Здесь можно установить глобальные параметры, такие как способ вычисления поля времени. Можно также включить автоматическую прокрутку трафика и разрешение имен, т. к. по умолчанию они отключены
Colorize display	Можно указать, чтобы пакеты определенных видов окрашивались определенным цветом. Это облегчает восприятие вывода и фокусирует внимание на нужной информации
Collapse/expand all	Показывать либо все детали каждого элемента, либо только верхний уровень 1

5.1.8 Инструменты (tools) Ethereal

Вместе с Ethereal поставляется несколько встроенных аналитических средств. Данная программа построена в архитектуре со встраиваемыми модулями (plug-in). Доступ к этим возможностям находится в меню Tools (таблица 5.5).

Таблица 5.5 – Инструментальные средства Ethereal

Пункт меню	Описание
Summary	Показывает список данных верхнего уровня сеанса перехвата, например, затраченное время, число пакетов, средний размер пакета, обшее количество перехваченных пакетов и среднюю плотность данных в среде передачи во время перехвата
Protocol hierarchy statistics	Выдает статистическое представление графика вашей сети. Показывает, какой процент сеанса перехвата составляет каждый тип пакетов. Можно свернуть или распахнуть представление, чтобы увидеть основные уровни или второстепенные протоколы определенного уровня
Statistics	Содержит ряд отчетов, специфичных для определенных типов протоколов. Дополнительную информацию по этому вопросу можно найти в документации Ethereal
Plugins	Показывает встраиваемые модули анализатора пакетов, которые мы загрузили. Это декодировшики для новых протоколов, которые можно добавлять к Ethereal, не изменяя основной версии программы. И поскольку это архитектура со встраиваемыми модулями, можно писать свои собственные модули

5.1.9 Сохранение вывода Ethereal

Выполнив перехват и анализ данных в Ethereal, можно сохранить их либо для анализа дополнительными средствами, либо для предоставления другим пользователям. С помощью опции Save As меню File можно выбрать подходящий формат, включая libpcap (по умолчанию), Sun Snoop, LANalyser, Sniffer, Microsoft Network Monitor и Visual Networks.

5.1.10 Применение Ethereal

Выполняя открытый перехват сети и используя затем статистические отчеты, можно понять, насколько загружена сеть и на какой вид пакетов приходится основная доля трафика. Проанализировав эти данные, можно определить действия, которые необходимо предпринять для улучшения работы сети. Если возникли проблемы, например, с почтовым сервером или службой DNS, можно, применяя Ethereal, подключиться к сети и понаблюдать за коммуникациями между серверами. Можно увидеть реальные сообщения серверов для таких протоколов, как SMTP или HTTP, и определить, где возникает проблема.

5.2 Задание для выполнения

1 Ознакомьтесь с графическим интерфейсом и элементами управления программы Ethereal. Выполните пробный запуск открытого перехвата. Остановите его. Ознакомьтесь с формой представления результатов. Включите в отчет описание всех действий по выполнению данного задания. Сформулируйте выводы и включите их в отчет.

2 Выполните перехват пакетов, относящихся к сессии подключения с вашего компьютера к WWW-серверу БГУИР: http://www.bsuir.by/. При назначении опций перехвата используйте документ «Опции и выражения команды tcpdump» для задания условий фильтрации в поле «Capture Filter». Сохраните результаты перехвата. Включите в отчет описание всех действий по выполнению задания данного пункта. Сформулируйте выводы и включите их в отчет.

3 Подробно проанализируйте все поля кадров Ethernet, относящихся к перехваченной сессии. В отчете опишите все поля кадров Ethernet. Сформулируйте выводы по заданию.

Практическое занятие №6

АУДИТ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ В СЕТЕВЫХ ОПЕРАЦИОННЫХ СИСТЕМАХ WINDOWS 2000/XP

Цель занятия: ознакомление с организацией аудита информационных процессов в сетевых операционных системах Windows 2000, XP.

6.1 Теоретические сведения

6.1.1 Журналы событий

Аудит – это процесс, позволяющий фиксировать события, происходящие в ОС. Сообщения о критических событиях, таких как переполнение жесткого диска или сбой в питании компьютера, выдаются на экран дисплея. Однако большинство событий записывается в три журнала событий (рисунок 6.1):

- системный журнал содержит информацию о событиях, относящихся к компонентам NT-XP, например, сообщения о сбое драйвера или службы при загрузке;
- журнал безопасности – события, связанные с безопасностью;
- журнал приложений – события, записываемые приложениями. Какие события будут зафиксированы в этом журнале, решают разработчики соответствующих приложений.

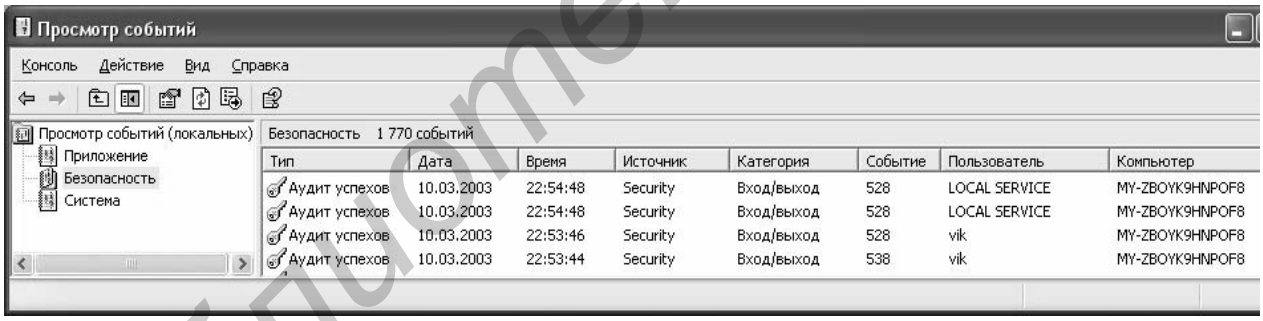

Рисунок 6.1 – Отображение журналов событий

По умолчанию системный журнал и журнал приложений могут просматривать все пользователи, журнал безопасности – только администраторы. Пользователи с привилегией управления аудитом и журналом безопасности могут читать и очищать журнал безопасности (по умолчанию это только администраторы) (таблица 6.1).

Журналы NT-XP находятся в папке *winnt_root**system32*\config в трех файлах (рисунок 6.2):

- AppEvent.evt (журнал приложений);
- SecEvent.evt (журнал безопасности);
- SysEvent.evt (системный журнал).

Таблица 6.1 – Права доступа пользователей к журналам

Журнал	Системный			Безопасности			Приложений		
Права доступа Локальная группа	Чтение (R)	Запись (W)	Очистка (C)	R	W	C	R	W	C
Система	+	+	+	+	+	+	+	+	+
Администраторы	+	+	+	+	–	+	+	+	+
Операторы сервера	+	–	+	–	–	–	+	+	+
Все	+	–	–	–	–	–	+	+	–

При запуске их открывает и блокирует ОС, и даже на дисках с файловой системой FAT получить доступ к файлам журналов, применяя стандартные средства, невозможно. Информация о событиях хранится на диске в двоичном виде. Журнал можно просмотреть в Windows 2000/XP с помощью программы просмотра событий из группы программ «Администрирование» (см. рисунок 6.2).

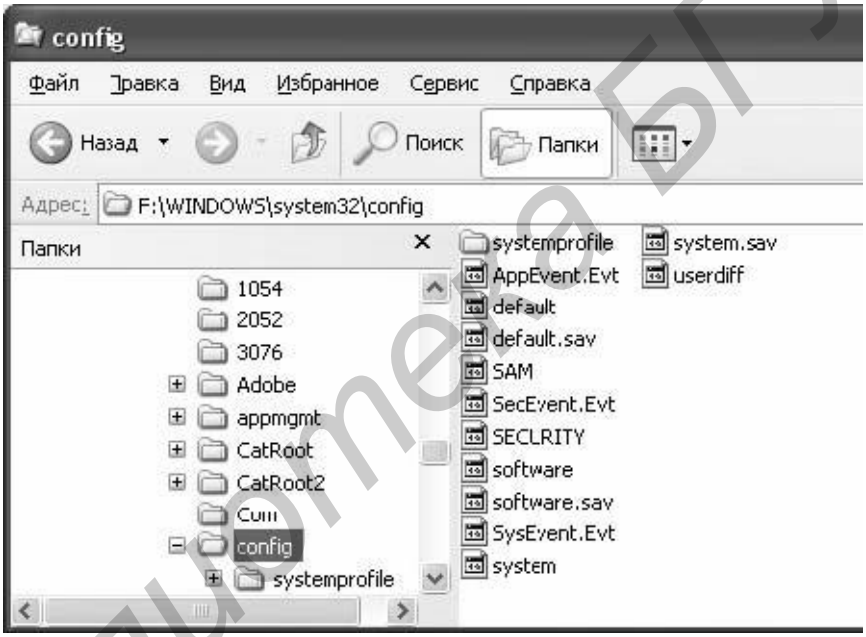

Рисунок 6.2 – Расположение файлов журналов событий

В журнал для событий записывается следующая информация:

- тип события;
- дата;
- время;
- источник, т. е. ПО, произведшее запись;
- категория;
- код события;
- имя пользователя, действия которого привели к возникновению события;
- имя компьютера, где произошло событие.

При нажатии левой клавиши мыши на определенном событии из журнала можно получить более детальную информацию о данном событии (рисунок 6.3).

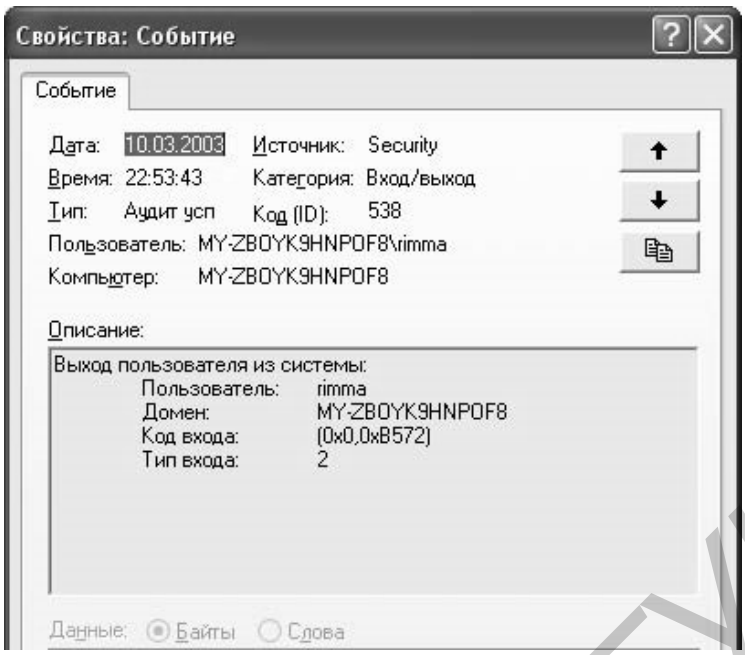

Рисунок 6.3 – Детализация информации о событии

Командой «Сохранить файл журнала как» можно сохранить данные в текстовом виде.

В свойствах журнала можно определить действия при заполнении файла данного журнала (рисунок 6.4):

- затирать старые события при необходимости;
- затирать события старше N дней, иначе новые события будут проигнорированы;
- не затирать события (очистка журнала вручную).

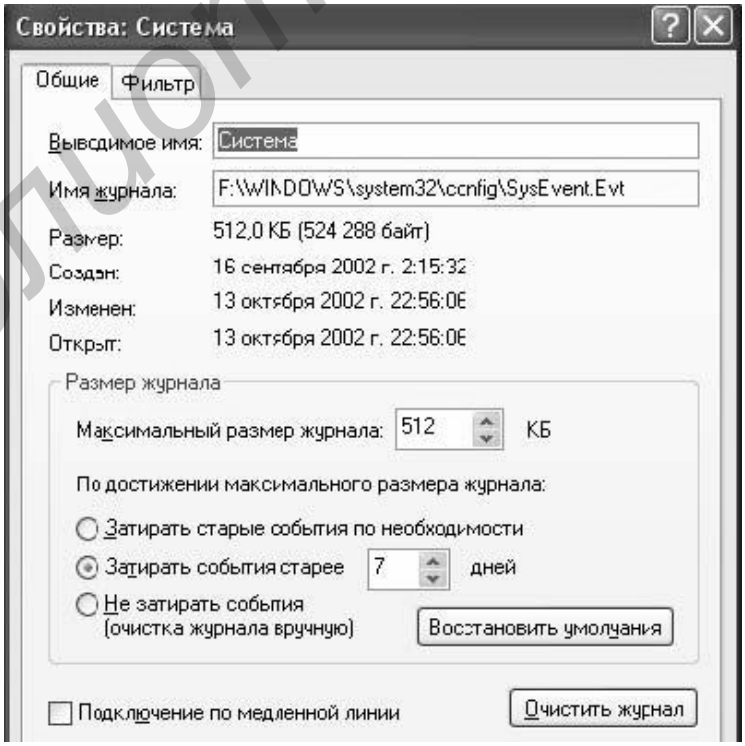

Рисунок 6.4 – Настройка параметров журнала

Для сортировки и фильтрации служит лист «Фильтр» свойств журнала (рисунок 6.5).

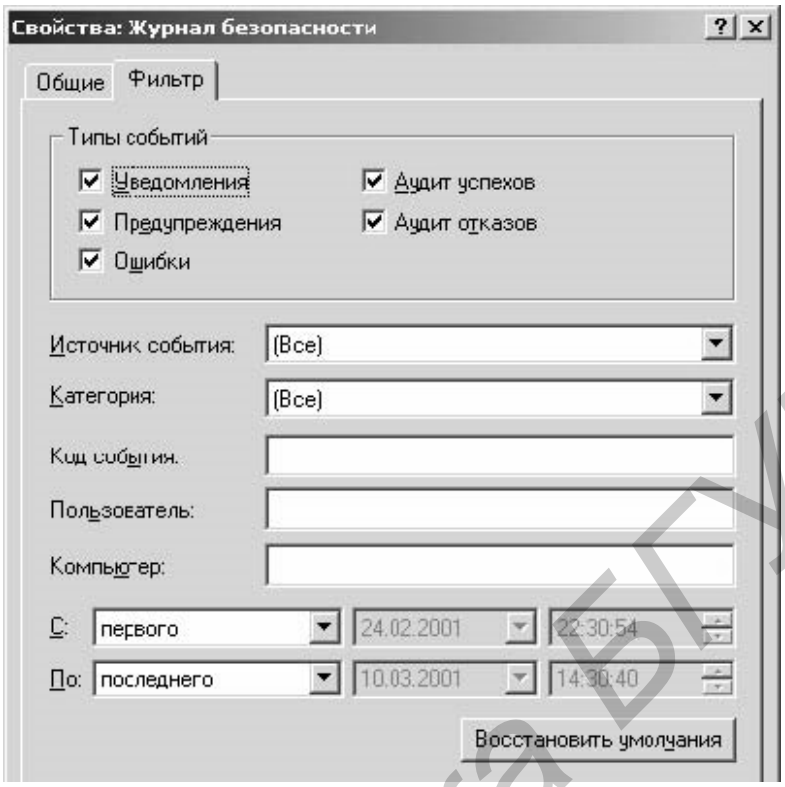

Рисунок 6.5 – Настройка свойств фильтров

6.1.2 Определение необходимости записи аудита

Каждая запись в SACL обрабатывается следующим образом:

- записи с типом не SystemAudit игнорируются;
- при отсутствии совпадений SID в записи с набором SID-субъекта запись пропускается;
- маска требуемого доступа сравнивается с маской доступа в ACE, если ни один тип доступа, указанный в маске ACE, не требовался пользователем, запись пропускается, в противном случае проверяются биты SUCCESSFUL_ACCESS_ACE_FLAG и FAILED_ACCESS_ACE_FLAG;
- если пользователь получил доступ, а бит SUCCESSFUL_ACCESS ACE_FLAG не установлен или пользователю доступ был запрещен, а бит FAILED_ACCESS_ACE_FLAG не установлен, запись пропускается;
- если запись прошла все проверки, монитор безопасности генерирует событие о доступе к объекту, которое должно быть помещено в журнал аудита.

По умолчанию аудит безопасности не ведется. В Windows 2000, XP аудит включается администратором в оснастке «Локальные параметры безопасности» (рисунок 6.6).

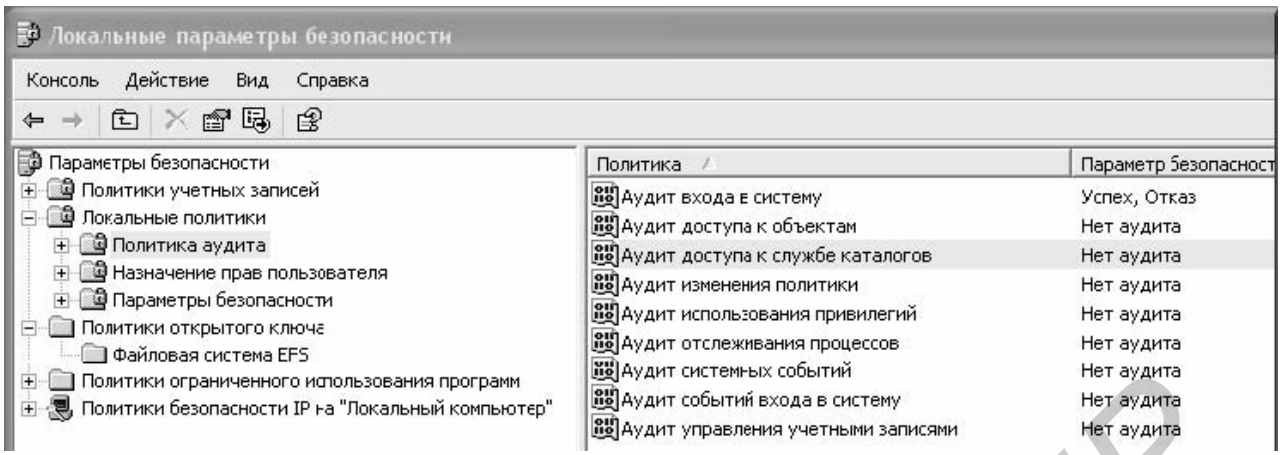

Рисунок 6.6 – Настройка аудита

На контроллере домена правила аудита определяются для всех контроллеров в этом домене. На рабочей станции и сервере, не являющемся контроллером домена, правила аудита задаются индивидуально для каждого компьютера.

6.1.3 Аудит входа в систему (Audit loon events)

Событие успешной регистрации пользователя в системе имеет номер (ID) 528. Событие с номером 538 означает завершение сеанса, начало которого зафиксировано событием 528 (рисунок 6.7).

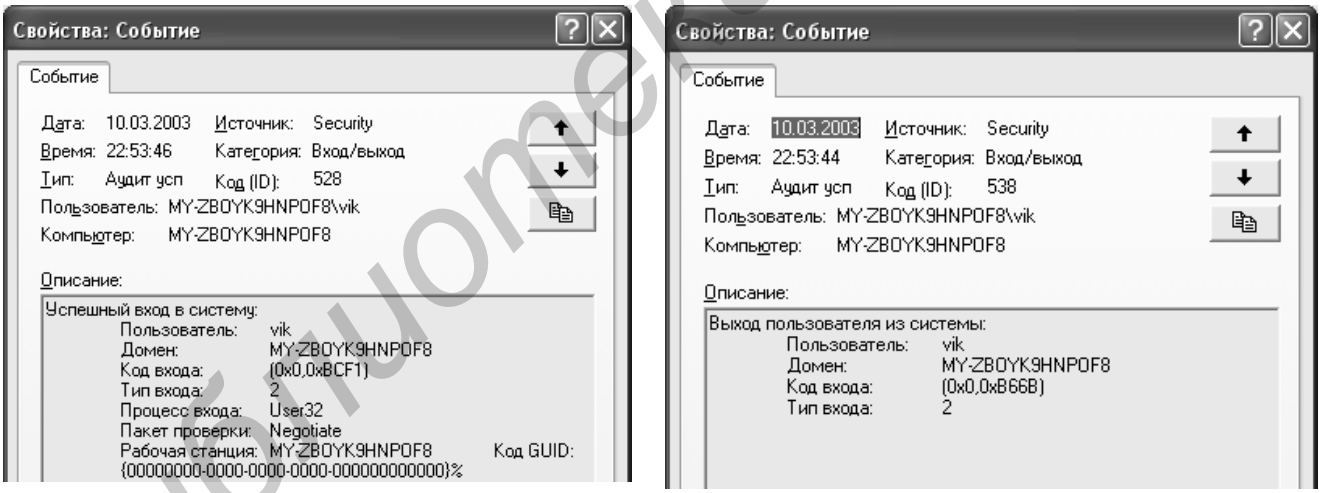

Рисунок 6.7 – Идентификация номера события

Событие 528 имеет несколько очень важных дополнительных параметров. Имя пользователя и домен определяют вошедшего в систему пользователя или то, чья учетная запись была при этом задействована. Любому активному сеансу работы пользователя с системой присваивается уникальный код входа. Именно он будет записан в событии завершения сеанса, что позволяет определить общее время работы пользователя при анализе событий 528 и 538 с одинаковым кодом входа. В случае нескольких одновременных сессий одного пользователя (например, когда вошедший интерактивно пользователь подключился к своему компьютеру еще и через сервер SMB) данный идентификатор позво-

ляет однозначно определить, в рамках какой сессии совершено то или иное событие.

Тип входа показывает, как пользователь вошел в систему:

2 – интерактивный вход с консоли, например, с помощью монитора и клавиатуры;

3 – сетевой вход, пользователь подключился по сети к диску этого ПК или использует сетевой ресурс по-другому;

4 – задание на выполнение командного файла при использовании планировщика задач независимых компаний;

5 – фиксируется при запуске службы с указанием конкретной учетной записи пользователя;

7 – разблокирование рабочей станции;

8 – сетевая регистрация незашифрованным паролем;

9 – ролевая (impersonated) регистрация.

В Windows NT событие 528 применялось для регистрации события любого типа. С Windows 2000 при подключении к диску на сервере, соединении с реестром сервера и выполнении других операций с использованием сетевой регистрации в журнал – новое событие ID 540. Это позволяет отделить сетевую регистрацию от других типов регистрации.

В событиях 540 следует обратить внимание на поле «Пользователь». Обычная пользовательская учетная запись свидетельствует о том, что пользователь зарегистрировался в системе через сеть; на эти события следует обратить внимание. Запись SYSTEM указывает на то, что одна системная служба устанавливает соединение с другой службой на той же машине. Имя компьютера с символом $ означает, что системная служба на удаленной машине устанавливает связь с системными службами на данном компьютере.

В поле «Домен» события 528, 540 указывается NetBios-имя домена, в котором расположена учетная запись пользователя. При регистрации с помощью локальной учетной записи в локальной базе SAM (диспетчер учетных записей), в имени «Домен» содержится имя NetBIOS компьютера. С помощью полей «Процесс входа» и «Пакет проверки» события ID 540 можно определить используемый протокол аутентификации Windows 2000 (NTLM или Kerberos).

При отметке Negotiate о регистрации NTLM в поле события «Рабочая станция» можно видеть имя NetBIOS клиентского компьютера. Если Windows 2000 использует механизм Kerberos, то это поле остается пустым.

Фиксируются также все неудачные попытки входа в систему. Событие 529 соответствует указанию неверного имени пользователя или пароля.

Если учетная запись пользователя недоступна или заблокирована, то записывается событие с номером соответственно 531 или 539.

Событие 530 указывает, что пользователь пытался войти в систему в недозволенное ему время или день недели.

Если учетная запись пользователя просрочена или устарел пароль, то фиксируется соответственно событие 532 или 535.

Если пользователь ограничен входом лишь на некоторые рабочие станции, а он пытается войти с другого компьютера, то запишется событие 533.

Можно ограничить права пользователя на выполнение определенных типов входа в различные системы. Если пользователь, которому запрещен доступ к какому-то компьютеру по сети, все же пытается обратиться к его ресурсу или реестру, то он получит отказ, а в журнал безопасности запишется событие с номером 534. Такое же событие будет зафиксировано при попытке пользователя войти в систему с консоли, если это ему запрещено. При попытке запустить службу с использованием учетной записи пользователя, не имеющей права на запуск служб (право «Вход в качестве службы»), также будет зафиксировано событие 534. Кроме того, событие 534 запишется и при попытке запуска задания с исполнением командного файла от имени учетной записи без права «Вход в качестве пакетного задания».

При всех других отказах в аутентификации фиксируется событие с номером 537 – отказ по неизвестной причине. Тип входа фиксируется при всех попытках входа в систему, независимо от их результата.

События категории «Аудит входа в систему» регистрируются в локальном журнале безопасности рабочей станции.

6.1.4 Аудит событий входа в систему (Audit account logon events)

Audit account logon events появился с Windows 2000. Данная категория событий используется для отслеживания аутентификации пользователей на контроллерах доменов.

При использовании протокола аутентификации Kerberos событие с ID 672 позволяет контролировать первичные подключения к домену при помощи процедуры получения билета TGT, а событие с ID 673 контролирует доступ к сетевым службам при помощи процедуры получения билета доступа к службе.

При вводе и аутентификации неправильного пароля контроллер домена записывает в журнал событие с ID 675 (ошибка предварительной аутентификации) с кодом ошибки 24. В сообщении указывается не только имя пользователя и имя домена, но и IP-адрес станции, с которой осуществлялась попытка несанкционированного доступа. Событие с ID 675 записывается еще и в том случае, если пользователь зарегистрировался на станции с одним именем, а затем пытался подключиться к серверу с другим.

При неправильном имени пользователя регистрируется событие с ID 676 с кодом ошибки 6. Код ошибки 12 указывает на попытку регистрации в неразрешенное время. Код ошибки 23 означает, что срок действия пароля пользователя истек. Код ошибки 18 указывает на завершение срока действия учетной записи или запрета администратора.

Событие 677 регистрируется, когда запрос на получение билета доступа к службе не удовлетворен.

При использовании протокола NTLM, когда контроллер домена успешно аутентифицировал пользователя, в журнал записывается событие с ID 680. В

событии с ID 680 указывается имя пользователя и имя станции, откуда поступил запрос на подключение. Если аутентификация NTLM по каким-то причинам не может быть выполнена, то контроллер записывает в журнал событие с ID 681.

6.1.5 Аудит доступа к объектам

На дисках, отформатированных с файловой системой NTFS, можно проводить аудит доступа к отдельным файлам и папкам. Это позволяет отследить выполняемые действия и идентифицировать пользователей, ответственных за эти действия.

Включение аудита доступа к объектам в политике аудита не приводит (в отличие от других категорий аудита) к автоматической регистрации событий, связанных с доступом к объектам. Администратору требуется сформировать SACL у объектов, к которым планируется осуществить аудит доступа. В свойствах файла или папки необходимо выбрать лист «Безопасность», нажать кнопку «Дополнительно» и выбрать лист «Аудит» (рисунок 6.8).

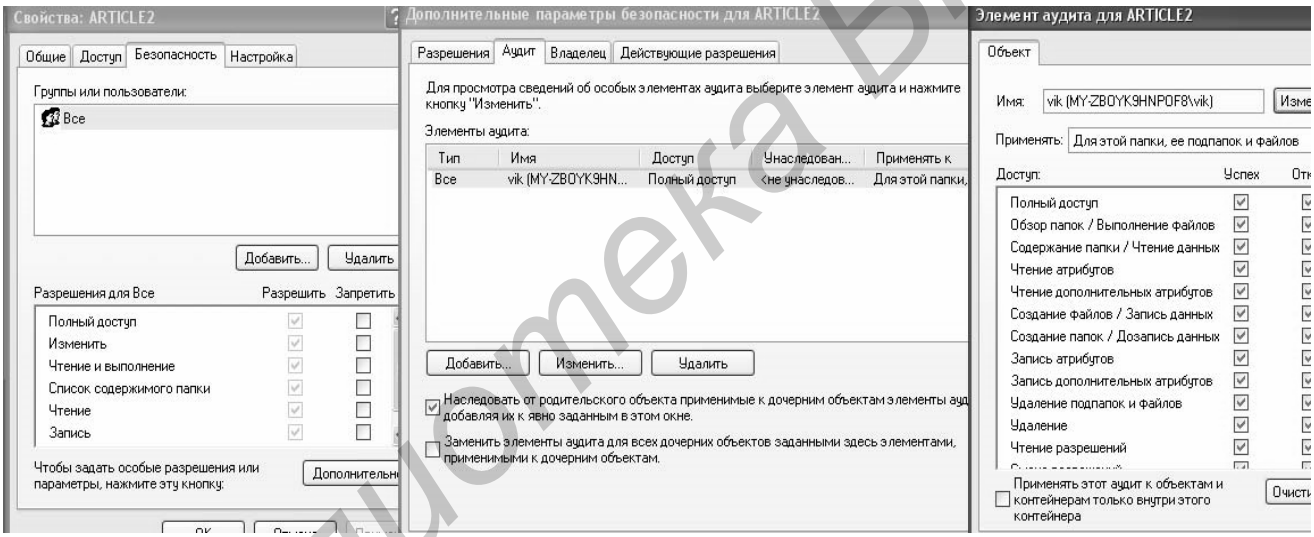

Рисунок 6.8 – Настройка аудита доступа к объектам

Для настройки аудита для нового пользователя или группы нажмите кнопку «Добавить». Выберите имя нужного пользователя или группы. В окне «Элемент аудита» можно указать необходимые параметры аудита. В поле «Применить» укажите, где следует выполнять аудит (это поле ввода доступно только для папок). В группе «Доступ» укажите, какие события следует отслеживать: окончившиеся успехом, отказом или оба типа событий. Флажок «Применить этот аудит к объектам и контейнерам только в пределах данного контейнера» определяет, распространяются введенные вами настройки аудита на файлы и папки вниз по дереву каталогов файловой системы или нет.

Для отключения аудита файла или папки в окне «Дополнительные параметры безопасности» выберите нужную запись и нажмите кнопку «Удалить». Если она недоступна, то настройки аудита наследуются от родительской папки.

Если вы хотите просмотреть или изменить настройки аудита для уже существующего пользователя или группы, нажмите кнопку «Изменить».

При наличии взаимодополняющих друг друга событий – событие «Открытие объекта (560)» фиксирует открытие объекта (рисунок 6.9), а событие «Закрытие дескриптора (562)» – его закрытие (рисунок 6.10).

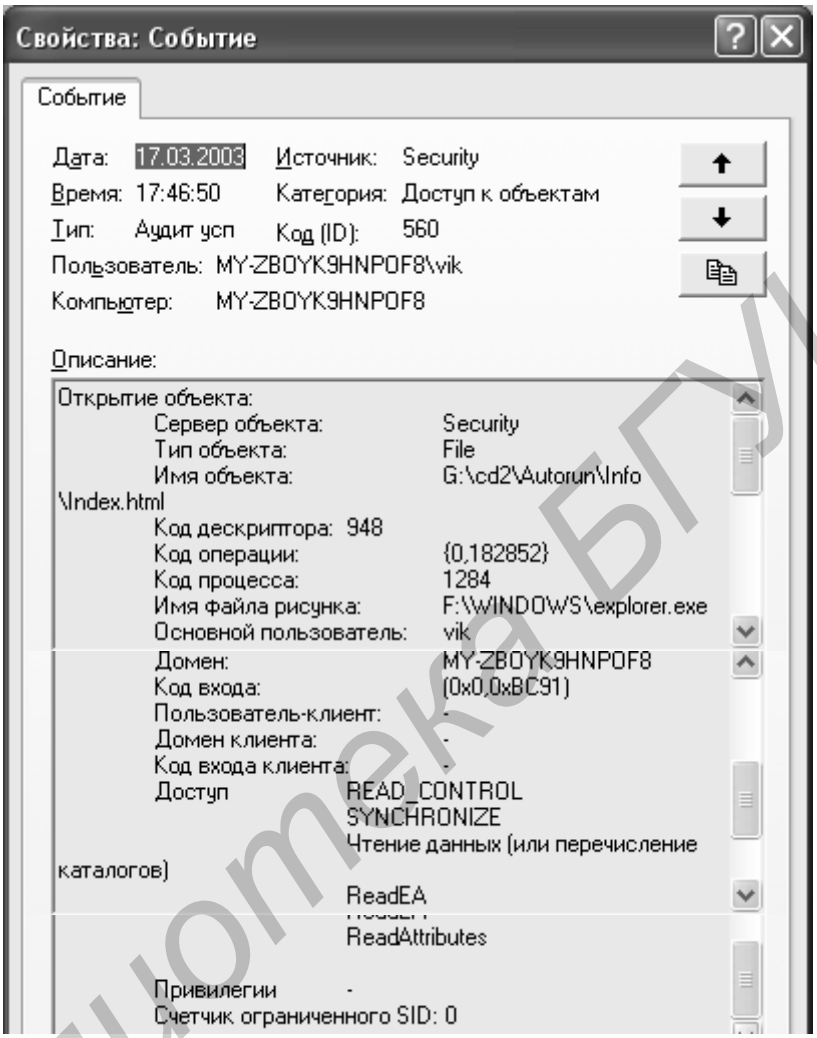

Рисунок 6.9 – Фиксация открытия объекта

Значение параметра «Сервер объекта» – всегда Security. Поле «Тип объекта» идентифицирует объект аудита – файл, папка, раздел реестра, принтер или служба. Успешное событие 560 записывает информацию об открытом объекте, а также имя пользователя и название приложения, которое воспользовалось объектом, тип доступа и код дескриптора.

Код дескриптора (handle Id) является уникальным кодом для контроля операционной системы за каждым объектом. Найдя пару событий открытия и закрытия (560 и 562) с одним кодом дескриптора, можно выяснить время работы пользователя с данным объектом. Код дескриптора указывается лишь при предоставлении пользователю доступа к объекту.

Код входа позволяет выяснить, в какой именно сессии пользователь обращался к объекту.

Код операции (Operation ID) – просто число, которое увеличивается на единицу для каждой операции, выполняемой в Active Directory (AD).

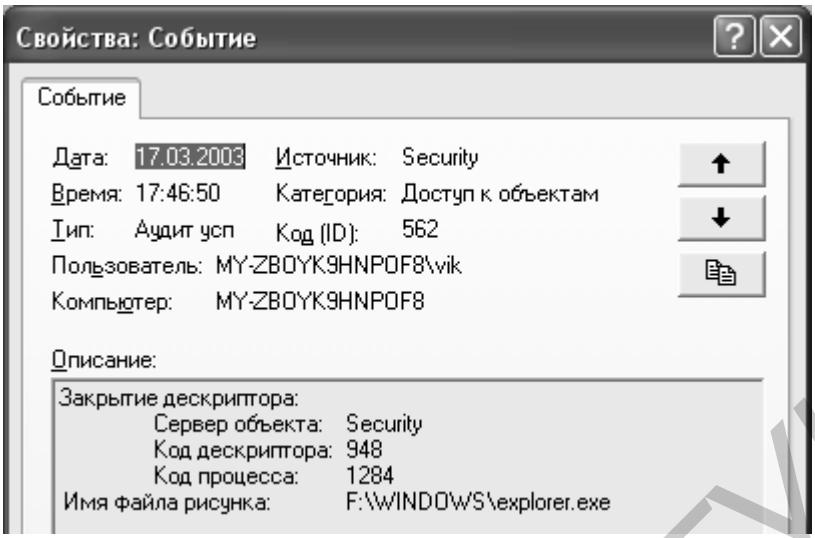

Рисунок 6.10 – Фиксация закрытия объекта

События хранят информацию о двух пользователях: об основном и о клиенте. При открытии файла на локальном компьютере с помощью обычного приложения, такого, как Microsoft Word, существенна только информация об основном пользователе. Однако при доступе к объекту из клиент-серверных приложений фиксируются оба типа пользователей: основной соответствует учетной записи, под которой запускается серверное приложение (например System), а клиентский – соответствует пользователю, от имени которого работает сервер.

Код процесса (Process ID, или PID) позволяет определить, какая именно программа обратилась к объекту.

Третьим важным событием в данной категории является событие 564 «Удаление объекта». Оно записывает только код дескриптора и код процесса. Чтобы разобраться, какой именно объект и кем был удален, надо найти по коду дескриптора соответствующее событие 560 открытия объекта. В событии 560 есть вся необходимая информация, и событие 564 удаления объекта следует связывать именно с ним.

В Windows XP дополнительно регистрируется событие 567 «Попытка доступа к объекту».

6.1.6 Аудит использования привилегий

Если сотрудник успешно воспользовался своей привилегией, то в журнал безопасности в зависимости от типа привилегии записывается событие 577 (вызов привилегированной службы) или 578 (операции с привилегированным объектом). Поле привилегии показывает условное обозначение использованной привилегии (рисунок 6.11).

Использование привилегий на регистрацию отражается категорией аудита входа в систему. Windows 2000/XP также не заносит в журнал информацию о

использовании привилегий архивирования и восстановления файлов и каталогов, а также восстановления файлов и каталогов, вызываемых так часто, что они быстро переполнили бы журнал безопасности. Чтобы система выполняла аудит использования этих полномочий, следует параметру реестра HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa: Set the Full-PrivilegeAuditing присвоить значение 1.

Рисунок 6.11 – Обозначение использованной привилегии

Windows 2000 никогда не регистрирует использование полномочий:

- создание маркерного объекта (SeCreateTokenPrivilege);
- отладка программ (SeDebugPrivilege);
- обход перекрестной проверки (SeChangeNotify-Privilege);
- замена маркера уровня процесса (SeAssignPrimaryTokenPrivilege);
- генерирование проверки безопасности (SeAudit-Privilege).

Но если регистрируется пользователь, обладающий одним или несколькими из этих полномочий, то Windows 2000 записывает в журнал событие с ID 576 с указанием кратких имен привилегий, которыми наделен данный пользователь (рисунок 6.12).

Рисунок 6.12 – Регистрация события

6.1.7 Аудит управления учетными записями

Позволяет определить, какая учетная запись была изменена, добавлена или удалена и кем именно (рисунок 6.13). Но какое именно свойство пользователя или группы было изменено – не сохраняется. События этой категории аудита записываются на той системе, где хранится учетная запись. Так, при создании новой локальной учетной записи соответствующее событие записывается в журнал безопасности данного компьютера, а при изменении учетной записи пользователя домена – в журнале на контроллере домена.

Рисунок 6.13 – Регистрация изменения учетной записи

В таблице 6.2 указаны идентификационные номера событий и их значения. В событии указывается учетная запись, с которой производилась операция, ее домен, код записи (Account ID), используемый для связи с кодом SID. По имени исполнителя можно определить пользователя, производившего действие с учетной записью. Его код входа позволяет встроить данное событие в цепочку других действий в данном сеансе.

Таблица 6.2 – Идентификационные номера и значения событий

Объект	Создание	Удаление	Изменение	Добавление члена	Удаление члена
Пользователь	624	630	642	–	–
Компьютер	645	646	647	–	–
Локальная группа	635	638	639	636	637
Глобальная группа	631	634	641	632	633
Универсальная группа	658	662	659	660	661

Если пользователь изменил свой пароль, записывается событие 627, при этом отмечается, успешно или нет завершена данная операция. Это зависит от права пользователя на смену пароля и от политики формирования паролей в домене. При смене пароля администратором записывается событие 628.

Рисунок 6.14 – Регистрация смены пароля

При нескольких безуспешных попытках войти в систему учетная запись домена блокируется и записывается событие 644.

У данной категории есть особенность: форма события аудита неудачи не предусмотрена, хотя можно поставить соответствующий флажок в диалоговом окне «Политика аудита». Для просмотра неудачных попыток редактирования базы учетных записей можно использовать следующее. Если установлены флажки для категории «Доступ к файлам и объектам», то в случае редактирования базы учетных записей в дополнение к событиям категории «Управление пользователями и группами» в журнале безопасности появятся события, связанные с доступом (или его запретом) к внутренним объектам диспетчера учетных записей.

6.1.8 Аудит доступа к службе каталогов

Данная категория аудита впервые появилась в Windows 2000 и применяется только на контроллере домена. Данная категория генерирует события с кодом 565. Она позволяет определить, какое свойство объекта AD изменено. В событии указывается тип, имя объекта. Чтобы определить, кем изменена запись, используются поля имя клиента, домен клиента, код входа клиента. В поле «Свойства» показывается какое свойство изменено.

6.1.9 Аудит изменений политики

К данной категории относят следующие события:

- добавление привилегии к учетной записи пользователя (608) (рисунок 6.15);
- удаление привилегии от учетной записи пользователя (609);
- установление новых доверительных отношений (610);
- удаление доверительных отношений (611);
- изменение информации о доверенном домене (620);
- изменение политики аудита (612);

- изменение политики Kerberos (617);

- изменение политики восстановления файлов, зашифрованных с помощью EFS (618);

- изменение политики безопасности IP (615, 616).

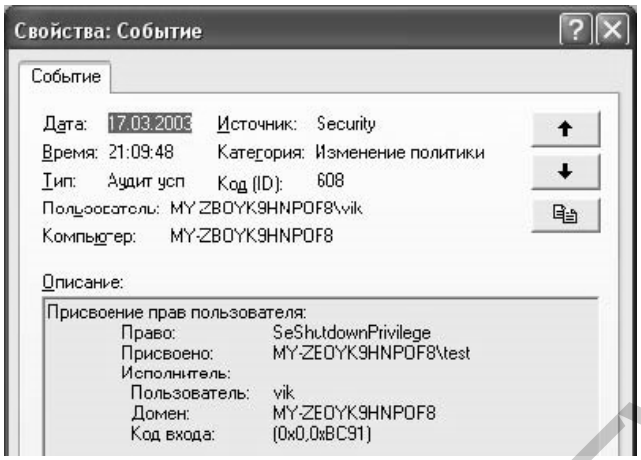

Рисунок 6.15 – Аудит изменений политики

6.1.10 Аудит системных событий

В данную категорию входят следующие события:

- перезапуск операционной системы (512);

- завершение работы операционной системы (513);

- загрузка пакета проверки подлинности (514);

- регистрация процесса проверки подлинности пользователя при входе в систему (515);

- очистка журнала безопасности (517);

- загрузка пакета уведомления обо всех изменениях в учетных записях пользователя (518).

6.1.11 Аудит отслеживания процессов

Данная категория позволяет проследить за тем, какие именно программы были запущены на рабочей станции и какие программы выполнялись на сервере.

В этой категории можно выделить следующие основные события:

- создание процесса – 592 (рисунок 6.16);

- завершение процесса – 593.

Найдя пару событий 592 и 593 с одинаковым кодом процесса, можно определить общее время работы того или иного приложения, которое указывается в поле имени файла образа. В поле имени пользователя хранится информация о том, кто запустил приложение. По полю кода входа можно отыскать соответствующее событие регистрации с кодом 528 и выяснить все подробности о сеансе, в котором запускалось приложение.

Для идентификации процесса, запустившего новый процесс, можно использовать поле кода создателя процесса. Достаточно найти предыдущее событие с кодом 592 с этим же кодом процесса.

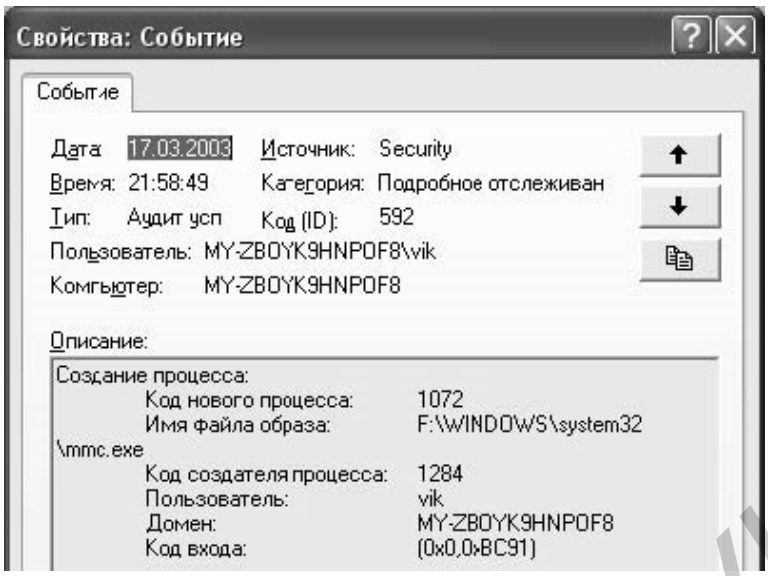

Рисунок 6.16 – Регистрация создания процесса

6.2 Задание для выполнения

1 Познакомьтесь с программой просмотра событий, с различными видами журналов, их структурой. Приведите отрывки журналов в отчете, дайте интерпретацию отдельных записей журналов.

2 Познакомьтесь с возможностями настройки журналов, параметрами фильтрации записей. Сохраните журнал в текстовом виде и экспортируйте его в Excel.

3 Ознакомьтесь и приведите в отчете настройки политики аудита.

4 Ознакомьтесь с аудитом доступа к объектам. Установите определенные права аудита на созданный вами каталог и вложенные в него файлы. Приведите их в вашем отчете. Произведите операции с этими файлами, приведите их в отчете и проанализируйте события, появляющиеся в журнале безопасности.

5 Включите аудит входов в систему и событий входа в систему, исследуйте события, отнесенные к данной категории аудита, дайте интерпретацию информации, выдаваемой для отдельных событий. Проведите анализ связи отдельных событий, сделайте выводы по результатам анализа.

6 Исследуйте аудит управления учетными записями.

7 Исследуйте аудит использования привилегий.

8 Исследуйте аудит изменения политик.

9 Познакомьтесь с аудитом системных событий

10 Исследуйте возможности аудита процессов.

Результаты исследования отдельных категорий аудита должны включать описание исследований, примеры различных событий данной категории, интерпретацию информации, выдаваемой для отдельных событий, анализ связи отдельных событий, полученные результаты и сделанные выводы.

Практическое занятие №7

ДИАГНОСТИКА И УПРАВЛЕНИЕ КОМПЬЮТЕРНЫМИ СЕТЯМИ С ПОМОЩЬЮ СЕТЕВЫХ СКАНЕРОВ

Цель занятия: обучение проведению исследования сети на наличие активных узлов, запущенных сервисов и определению типа операционной системы.

7.1 Теоретические сведения

7.1.1 Сканирование портов

Сканирование как метод вскрытия каналов передачи данных существует уже достаточно долгое время. Идея заключается в том, чтобы исследовать как можно больше каналов и отслеживать те из них, которые находятся в состоянии ожидания соединения и являются полезными для исследователя.

Сам по себе термин «сканирование» появился в процессе слияния компьютеров с телефонными системами. В результате была сформирована глобальная сеть телефонных коммуникаций, доступ в которую можно получить, всего лишь набрав номер на телефонном аппарате. С одного телефона доступны миллионы абонентов телефонной сети, однако полезными могут оказаться лишь сотни, а то и десятки абонентов, к телефонам которых подключен модем.

Огромное число компьютеров объединено в сеть с помощью специального оборудования (сетевых адаптеров, кабельных модемов), а также выделенных линий и не использует коммутируемые линии АТС. В этом случае определить нужно не *телефонный номер*, а номер *порта* сервера, ожидающего запрос на соединение.

Термин «порт» является абстрактным понятием, используемым для упрощенного описания механизма установления соединения между узлами, и представляет собой потенциальный канал передачи данных. Использование механизма портов существенно облегчает процесс установления соединения и обмена информацией между сервером и узлом. Кроме того, имеется возможность исследования сетевого окружения сервера методом опроса его портов (т. н. «сканирование» портов). На все возможные номера портов (1-65535) сервера посылается «лавина» пакетов, и поэтому от каких портов будут (или не будут) получены ответы, определяются открытые порты и службы, работающие на исследуемом сервере.

7.1.2 Техники сканирования портов

Было разработано огромное количество различных методов для поиска протоколов и портов, которые «прослушивает» удаленная машина. Все методы имеют определенные преимущества и недостатки. Ниже рассмотрены наиболее часто используемые из них.

7.1.2.1 Сканирование сервера с использованием ICMP-эха

Перед непосредственным сканированием портов удаленного узла необходимо выяснить, какие узлы в сети являются функционирующими, и определить их адреса. Особенно это важно при сканировании группы узлов либо при сканировании определенного сегмента сети.

Данный метод работает аналогично команде ping. В качестве запроса узел отправляет серверу ICMP-сообщение и ожидает получения ответа, также представляющего собой ICMP-сообщение (т. н. ICMP-эхо). Варьируя время ожидания ответа на ping-запрос, можно сканировать большие сети.

7.1.2.2 Сканирование TCP-портов функцией connect()

Данный метод является основным для сканирования портов по протоколу TCP. Функция connect() позволяет узлу соединиться с любым портом сервера. Если порт, указанный в качестве параметра функции, прослушивается сервером (т. е. порт открыт для соединения), то результатом выполнения функции будет установление соединения с сервером по указанному порту. В противном случае, если соединение не установлено, то порт с указанным номером является закрытым.

Этот метод обладает одним серьезным преимуществом: его может применить любой пользователь, не обладающий никакими привилегиями на узле. Другое преимущество – скорость исследования. Последовательный перебор портов путем вызова функции connect() для очередного номера порта, определение его состояния и закрытие соединения – достаточно долгий процесс. Однако его можно ускорить, применив метод «параллельного просмотра» с использованием неблокированного соединения (non-blocked socket). Такой метод позволяет определить состояние практически всех портов сервера одновременно.

Недостатком данного метода можно считать возможность обнаружения и фильтрации такого рода сканирования, причем сделать это достаточно легко. Log-файл сканируемого сервера укажет службам, отвечающим за внешние подключения, на наличие многочисленных попыток подключения с одного адреса и ошибок установления соединения (поскольку узел после установления соединения сразу обрывает его), а те в свою очередь немедленно заблокируют доступ к серверу для узла с данным адресом.

7.1.2.3 Сканирование TCP-портов флагом SYN

Данный метод известен под названием «сканирование с установлением наполовину открытого соединения» (half-open scanning), поскольку установление полного TCP-соединения не производится. Вместо этого узел отправляет на определенный порт сервера SYN-пакет, якобы намереваясь создать соединение, и ожидает ответ. Наличие в ответе флагов SYN/ACK означает, что порт открыт и прослушивается сервером. Получение в ответ TCP-пакета с флагом RST означает, что порт закрыт и не прослушивается.

В случае приема SYN/ACK-пакета узел немедленно отправляет RST-пакет для сброса устанавливаемого сервером соединения. Преимущество дан-

ного метода прежде всего заключается в том, что лишь немногие серверы способны зарегистрировать такого рода сканирование. Пользователь, однако, должен обладать статусом Root на узле, с которого производится сканирование. Если статус будет ниже Root, то пользователь попросту не сможет программно сформировать одиночный SYN-пакет.

7.1.2.4 Сканирование TCP-портов флагом FIN

Идея заключается в том, что согласно RFC 793 на FIN-пакет, прибывший на закрытый порт, сервер должен ответить RST-пакетом. FIN-пакеты на открытые порты игнорируются сервером. Однако не все ОС придерживаются этой рекомендации. Так ОС Windows 95/98/NT, по всей видимости, имеют иммунитет к такому сканированию, однако большинство ОС являются восприимчивыми. Таким образом, совместно используя SYN- и FIN-сканирование, можно с успехом обойти средства защиты сервера и просканировать его порты.

7.1.2.5 Сканирование TCP-портов флагами SYN/FIN с использованием IP-фрагментации

Данный метод представляет собой комбинацию SYN- и FIN-сканирования с небольшим усовершенствованием. TCP-пакет (SYN- или FIN-пакет, имеющий небольшой размер) разбивается на стороне узла на пару IP-фрагментов меньшего размера, и эта пара IP-фрагментов отправляется серверу. На стороне сервера IP-фрагменты «собираются» в один TCP-пакет и производится его обработка (те же действия, как и при SYN- или FIN-сканировании).

Фрагментация позволяет уменьшить вероятность обнаружения сканирования фильтрами пакетов и другим подобным оборудованием. Однако при этом следует быть очень осторожным, поскольку некоторые программы имеют обыкновение «зависать» при попытке обработки такого маленького IP-фрагмента.

7.1.2.6 Сканирование TCP-портов методом reverse-ident (обратной идентификации)

Протокол ident (RFC 1413) позволяет определить указанное при входе в систему имя (*username* или *login*) владельца любого запущенного на сервере процесса, связанного с ним, даже если сам этот процесс не инициализировал TCP-соединение. Так, например, имеется возможность подключиться к http-порту и затем использовать identd, чтобы определить, работает ли на сервере пользователь root. Это может быть сделано только при установлении «полного» TCP-соединения с портом исследуемого сервера.

7.1.2.7 Сканирование UDP-портов проверкой ICMP-сообщения «Порт недоступен»

Этот метод также предназначен для определения состояния портов сервера. Основным отличием является использование протокола UDP вместо протокола TCP. Несмотря на то, что организация протокола UDP проще, чем TCP, сканировать UDP-порты гораздо труднее. Это связано прежде всего с концеп-

цией протокола UDP как протокола с *негарантированной* доставкой данных. Поэтому UDP-порт не посылает подтверждение приема запроса на установление соединения, и нет никакой гарантии, что отправленные UDP-порту данные успешно дойдут до него.

Большинство серверов в ответ на пакет, прибывший на закрытый UDP-порт, отправляют ICMP-сообщение «Порт недоступен» (Port Unreachable – PU). Таким образом, если в ответ на UDP-пакет пришло ICMP-сообщение PU, то сканируемый порт является закрытым, в противном случае (при отсутствии PU) порт открыт. Поскольку нет гарантии, что запросы от узла дойдут до сервера, пользователь должен позаботиться о повторной передаче UDP-пакета, который по всей видимости оказался потерянным.

Этот метод работает очень медленно из-за использования на некоторых машинах так называемой «компенсации» (подпункт 4.3.2.8 стандарта RFC 1812), ограничивающей частоту генерирования ICMP-сообщений об ошибке. Например, ядро Linux ограничивает частоту генерирования ICMP-сообщения «адресат недостижим» (Destination Unreachable) до 80 сообщений за 4 с с простоем 0,25 с, если это ограничение было превышено. Кроме того, для использования данного метода (а именно – для обнаружения ICMP-сообщений об ошибке) пользователь должен обладать статусом Root на узле, с которого производится сканирование.

7.1.2.8 Сканирование UDP-портов с использованием функций recvfrom() и write()

Этот метод используется в случае, когда пользователь, проводящий сканирование, не обладает статусом Root на узле. Поскольку не root-пользователь не может «читать» ICMP-сообщение PU в ОС, поддерживающих механизм сокетов (например в Linux), имеется возможность получения информации о состоянии UDP-порта косвенным способом. Так, например, попытка вызова функции write() на закрытый порт обычно приводит к возникновению ошибки.

Функция recvfrom() в этом плане более информативна. Вызов ее на неблокированный UDP-сокет сервера обычно возвращает ошибку EAGAIN (Try Again – «попытайтесь еще раз», код 13) в случае, когда ICMP-сообщение не было принято и ECONREFUSED (Connection Refused – «соединение закрыто», код 111), если ICMP-сообщение было принято. Таким образом, по этим признакам также возможно определить состояние портов сканируемого сервера.

7.1.3 Сетевой сканер Nmap

До возникновения идеи о создании программы Nmap были исследованы возможности многих сканеров, таких, как strobe (автор – Julian Assange – WikiLeaks), netcat (Hobbit), stcp (Uriel Maimon), pscan (Pulvius), ident-scan (Dave Goldsmith) и Satan (Wietse Venema). Вначале были попытки доработать код одних сканеров для поддержки лучших возможностей других. Затем было решено написать абсолютно новый сканер, который использовал бы лучшие возможно-

сти его предшественников, и конечно, имел новые, такие, как «фрагментированное» сканирование и др. Так получился сетевой сканер Nmap – the Network Mapper. Ниже приведены наиболее отличительные его возможности.

Динамическое вычисление времени задержки. Некоторым сканерам для работы необходимо указать время задержки между передачей двух пакетов. Можно использовать данные ping-запроса, но это займет достаточно много времени, и кроме того, время задержки постоянно меняется и зависит от «загруженности» узла, состояния сети и т. д. Nmap самостоятельно определяет время задержки и подстраивает его в процессе сканирования. Для root-пользователей используется наилучший способ определения времени задержки – функция ping. Для остальных этот параметр определяется функцией connect () на закрытый порт. Кроме того, у пользователя имеется возможность самостоятельно задать время задержки, но обычно делать это нет необходимости.

Повторная передача пакетов. Некоторые сканеры сразу отправляют все запросы, а затем «собирают» ответы на них. Весьма некорректный подход, поскольку при этом не принимается во внимание тот факт, что иногда пакеты могут просто не дойти до адресата по самым различным причинам. В такой ситуации сканер примет решение об отсутствии ответа и ошибочно укажет, что сканируемый порт закрыт. Больше всего ошибок возникает при использовании «негативного» сканирования типа UDP или FIN, решение в которых принимается на основе отсутствия ответа. Nmap автоматически выбирает число повторных передач пакетов на порты, от которых не был получен ответ.

Параллельное сканирование портов. Некоторые сканеры последовательно сканируют все 65 535 портов за один раз. Этот метод нормально работает лишь при сканировании TCP-портов в высокоскоростных локальных сетях. Глобальные сети типа Internet высокой скоростью не отличаются. Nmap использует неблокированный ввод/вывод (non-blocked i/o) и параллельное сканирование во всех режимах TCP и UDP. Вы можете задать число параллельных процессов сканирования самостоятельно. На очень быстрых сетях эффективность сканирования уменьшается при указании этого значения больше 18. На медленных сетях – наоборот, чем больше значение, тем выше эффективность.

Гибкое указание сканируемых портов. Часто бывает необходимо отсканировать какие-либо конкретные порты, а не все 65 535 портов сразу. Большинство сканеров позволяет задавать диапазон портов типа 1-N, что также не всегда приемлемо. Nmap позволяет задать любое количество произвольных диапазонов и портов, например '21-25,80,113,6000-'. При использовании «быстрого» режима Nmap будет сканировать только порты, перечисленные в файле /etc/services.

Гибкое задание цели сканирования. Часто необходимо просканировать более чем один узел, однако большинство сканеров позволяют задать лишь один адрес. Все, что не является опцией или ее аргументом, Nmap воспринимает как адрес целевого узла. Таким образом, вы можете абсолютно произвольно указать адреса и диапазоны адресов, которые хотите просканировать. Кроме то-

го, вы можете использовать маску для сканирования групп адресов различных классов.

Определение неактивных узлов. Некоторые сканеры позволяют сканировать большие сети, однако они тратят очень много времени на сканирование 65 535 портов узла, который по каким-либо причинам не функционирует. По умолчанию перед сканированием Nmap опрашивает каждый узел и определяет его состояние, для того чтобы не тратить время на сканирование неактивных узлов.

Определение IP-адреса сканирующего узла. По некоторым причинам большинство сканеров требуют указать используемый сканирующим узлом IP-адрес в качестве одного из параметров. Nmap автоматически определяет IP-адрес машины, на которой он работает, на стадии ping-опроса и использует тот адрес, на который пришел ответ. Если он не смог это сделать (например, пользователь отключил ping-опрос), Nmap пробует определить первичный сетевой интерфейс и использует его IP-адрес. Наконец пользователь сам может указать IP-адрес его узла.

Рассмотрены лишь основные методы, применяемые для сканирования портов сервера, определения их состояния и работающих на сервере служб. Комплексное использование рассмотренных методов, примененное в программе Nmap, позволяет не только получить достоверную информацию об открытых портах, но и обойти возможные средства защиты исследуемого сегмента сети.

Системный администратор может воспользоваться Nmap для сканирования своего сегмента сети с целью выяснения уязвимостей и возможности их применения против него самого, после чего модернизировать средства защиты сети.

Интерфейсом программы Nmap служит режим командной строки.

При работе программы Nmap под операционной системой Windows необходима система прямого доступа к ядру операционной системы Windows, которая называется WinPcap. Рассмотрим основные механизмы, используемые в системе WinPcap.

7.1.3.1 WinPcap

WinPcap – бесплатная, общедоступная система для прямого доступа к сети под ОС Windows. WinPcap дает возможность приложениям Win32 (32-разрядные Windows-приложения) иметь доступ к низкоуровневому представлению сетевого трафика для его непосредственной обработки, используя сетевой адаптер компьютера.

WinPcap позволяет:

- захватывать пакеты низкого уровня, которые предназначены непосредственно как для узла, на котором работает WinPcap, так и для других узлов (при использовании технологии общая шина);

- производить фильтрацию пакетов, согласно установленным правилам, перед их отправкой приложению;

- передавать пакеты низкого уровня по сети;
- собирать статистические значения о сетевом трафике.

Этот набор функций предоставляется драйвером устройств, который устанавливается в сетевую часть ядра win32, и несколькими DLL (dynamic-link library – динамически подсоединяемая библиотека).

Все эти возможности предоставляются через мощный интерфейс программирования, который легко используется приложениями и может быть перенесен на различные операционные системы.

WinPcap может использоваться различными видами инструментальных средств для анализа сетевой активности, поиска неисправностей, безопасности и контроля. Классические инструментальные средства, которые используют WinPcap:

- сетевые анализаторы протоколов;
- сетевые мониторы;
- журналы регистрации сетевой активности;
- генераторы трафика;
- соединения пользовательского уровня и маршрутизаторы;
- сетевые системы обнаружения вторжения (NIDS);
- сетевые сканеры;
- инструментальные средства безопасности.

7.2 Задание для выполнения

1 Запустите режим командной строки.

Нажмите Пуск-Выполнить (рисунок 7.1), в появившемся окне введите команду cmd и нажмите кнопку «ОК». В результате появиться окно, показанное на рисунке 7.2.

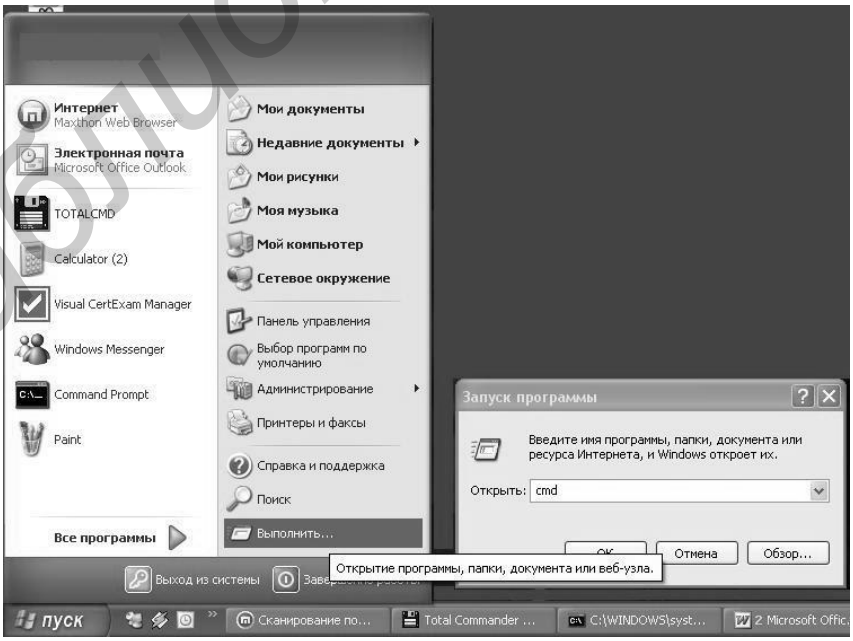

Рисунок 7.1 – Запуск режима командной строки

Рисунок 7.2 – Режим командной строки

2 Просмотрите основные команды и их описание. Введите команду Nmap для ознакомления с основными командами (см. рисунок 7.2).

3 Введите команду, которая производит поиск активных узлов для подсети вашего варианта (таблица 7.1) и сохраняет результат в текстовом файле.

Таблица 7.1 – Исходные данные

Вариант	Исследуемая подсеть	Техника сканирования
1	192.168.2.0 – 192.168.2.255	TCP SYN
2	192.168.2.0 – 192.168.2.128	Connect()
3	192.168.2.128 – 192.168.2.255	ACK

Введите команду Nmap -sP [адресное пространство, указанное для вашего варианта] -oN [путь к файлу, в котором сохраняется результат] (рисунок 7.3).

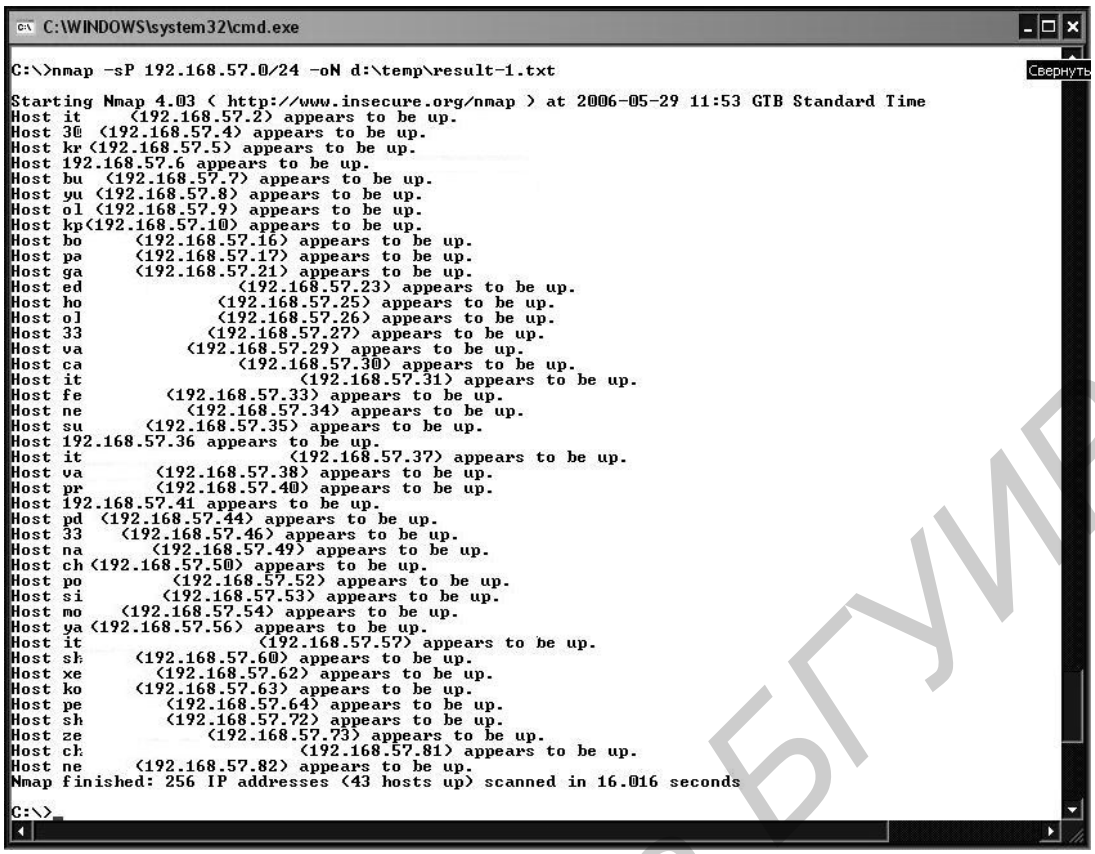

Рисунок 7.3 – Результат команды Nmap с ключами -sP и -oN

4 Введите команду, которая исследует открытые порты и тип запущенных сервисов для всех выявленных узлов, и сохраните результат в текстовый файл.

Введите команду Nmap [ключ техники сканирования, указанный в варианте] [адрес узла] -oN [путь к файлу, в котором сохраняется результат]. Пример на рисунке 7.4.

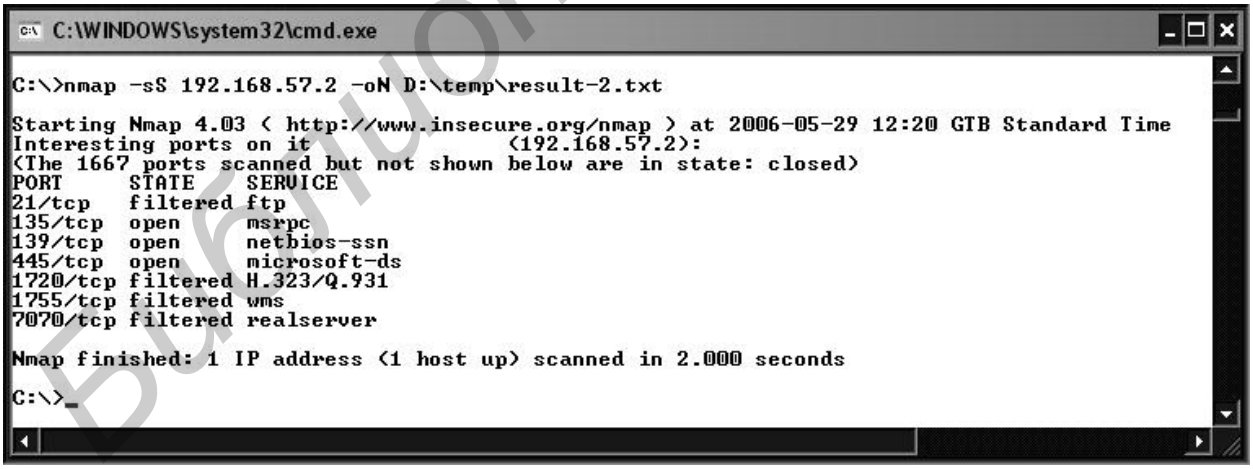

Рисунок 7.4 – Результат команды Nmap с ключами -sS и -oN

5 Введите команду, которая определяет тип операционной системы для всех выявленных узлов и сохраните результат в текстовый файл.

Введите команду Nmap -O [адрес узла] -oN [путь к файлу, в котором сохраняется результат]. Пример на рисунке 7.5.

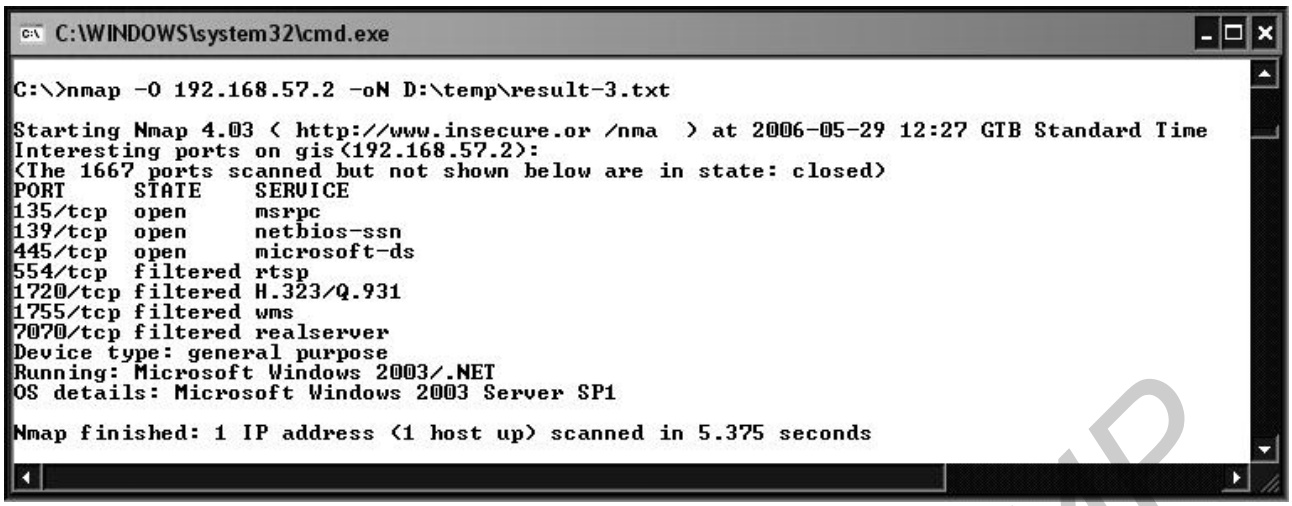

Рисунок 7.5 – Результат команды Nmap с ключами -O и –oN

6 Введите команду, которая определяет поддерживаемые IP-протоколы для всех выявленных узлов, и сохраните результат в текстовый файл.

Введите команду Nmap -sO [адрес узла] -oN [путь к файлу, в котором сохраняется результат]. Пример на рисунке 7.6.

Рисунок 7.6 – Результат команды Nmap с ключами -sO и -oN

7 Постройте таблицу 7.2 по результатам проведенного исследования.

Таблица 7.2 – Результаты исследования

Доменное имя узла	IP-адрес узла	Тип протокола /порт	Сервис	Операционная система	IP-протоколы

8 На основании полученных данных сделайте предположение о функциональных возможностях узлов.

Практическое занятие №8

ИТОГОВЫЙ КОМПЬЮТЕРНЫЙ ТЕСТ ПО КУРСУ

Цель занятия: контроль знаний по тематике проведенных практических занятий.

Тестирование выполняется на ПЭВМ в программной оболочке Hypertest 1.1

Материал, выносимый на тестовое занятие, включает ряд теоретических вопросов по темам занятий №1–7.

Студентам предлагается 15 вариантов тестов, каждый из которых включает 30 вопросов.

Тестирование проводится в течение 15 мин. Подгруппы для тестирования состоят из 6–8 человек.

Процедура тестирования заключается в выборе одного из нескольких альтернативных вариантов ответа на каждый вопрос и решении задач, аналогичных задачам, рассмотренным в практических занятиях №1–4.

Ответы заносятся в числовое поле.

Результаты тестирования предоставляются испытуемому в виде процентного содержания верных ответов непосредственно после окончания выделенного на это времени.

ЛИТЕРАТУРА

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

2 Танненбаум, Э. Компьютерные сети / Э. Танненбаум. – 4-е изд. – СПб. : Питер, 2007. – 993 с.

3 Тимонович, Г. Л. Управление коммуникациями в сети Интернет : практикум / Г. Л. Тимонович, А. С. Гринберг, Н. А. Свириденко. – Минск : Академия управления при Президенте Республики Беларусь, 2006. – 218 с.

4 Левин, М. Безопасность в сетях Internet и Intranet. Руководство пользователя / М. Левин. – М. : Познавательная книга Плюс, 2001. – 320 с.

Св. план 2013, поз. 52

Учебное издание

Гурский Александр Леонидович
Певнева Наталья Алексеевна

ТЕЛЕКОММУНИКАЦИОННЫЕ И ИНФОРМАЦИОННЫЕ СИСТЕМЫ И СЕТИ

ПОСОБИЕ

Редактор *Е. И. Герман*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *В. М. Задоля*

Подписано в печать 03.12.2013. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 3,84. Уч.-изд. л. 4,0. Тираж 100 экз. Заказ 140.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0494371 от 16.03.2009. ЛИ №02330/0494175 от 03.04.2009.
220013, Минск, П. Бровки, 6