Краткое пособие по сквозной диагностике Информационной Системы

1 ВВЕДЕНИЕ 4

2 ЭЛЕМЕНТЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ 5

3 УРОВЕНЬ ПАССИВНОГО ОБОРУДОВАНИЯ 6

3.1 Классификация пассивного оборудования 6

3.2 Средства диагностики 6

3.3 Диагностика пассивного оборудования 6

3.4 Примеры 9

4 УРОВЕНЬ АКТИВНОГО ОБОРУДОВАНИЯ 11

4.1 Типы активного оборудования 11

4.2 Средства диагностики 11

4.3 Диагностика в коммутируемой сети 13

4.4 Классификация сбоев работы сетей Ethernet 15

4.5 Различия ошибок повреждения пакетов 16

4.6 Способы обнаружения коллизий и ошибок 17

4.7 Описание эксперимента по исследованию влияния коллизий и ошибок на скорость работы приложения 18

4.8 Примеры 22

5 УРОВЕНЬ ДРАЙВЕРОВ И СЕРВИСОВ 25

5.1 Интеллектуальное ядро и возможные сбои его работы 25

5.2 Предварительная диагностика 25

5.3 Диагностика потери данных 28

5.4 Определение скорости передачи данных 29

5.5 Обработка и передача информации сетевыми устройствами 30

5.6 Примеры 31

6 ДИАГНОСТИКА ПО 32

6.1 Вопросы диагностики ПО 32

6.2 Средства и методика определения зависимости работы приложения от других элементов ИС 32

6.3 Функции анализа программы Trend Analyst..... 33

6.4 Примеры 36

1 Введение

Развитие информационных технологий привело к полному объединению ранее разрозненных элементов систем получения, обработки и передачи информации. Сейчас мы можем в полной мере говорить о существовании информационных систем (ИС), которые включают в себя программное обеспечение (ПО), сервера, персональные компьютеры, сети, телефоны, системы видеоконференций и безопасности и т.д. Таким образом, ИС – это совокупность всех элементов, предназначенных для обеспечения работы приложения.

Бурное развитие и усложнение информационных технологий привело к тому, что обслуживающий персонал все больше разделяется по специализациям (программисты, специалисты по сетям и электронной технике и т.д.); усложняется контроль и диагностика элементов ИС. В то же время сквозная диагностика ИС требует комплексного подхода и понимания работы всех элементов ИС.

К сожалению, данным вопросом практически никто не занимается, имеющиеся книги говорят или в общем или о проблемах и методах решения для конкретного оборудования. На базе собственного опыта и наработках компании Пролан (www.prolan.ru) рассмотрены средства и методики проведения сквозной диагностики, т.е. полной диагностики всех элементов и уровней ИС.

Данное пособие затрагивает только вопросы диагностики. В случае необходимости ознакомления с теоретическими основами работы компьютерных сетей рекомендуем изучить специальную литературу (см. по окончании главы).

Диагностика ИС рассматривается в основном на базе Ethernet сетей и Intel-совместимых компьютеров.

Примеры приводятся в виде задач, т.е. вначале описываются симптомы проблемы, проведенные действия, полученные результаты и только в конце выводы.

Литература:

1. Компьютерные сети. Принципы, технологии, протоколы. В.Г. Олифер, Н.А. Олифер – “Питер”, 1999;
2. Fast Ethernet. Лаем Куин, Ричард Рассел – К.: Издательская группа BHV, 1998.
3. Структурированные кабельные системы. Семенов А.Б., Стрижаков С.К., Сунчелей И.Р. – 3-е изд., перераб. и доп. – М.:ЛАЙТ Лтд., 2001.

2 Элементы Информационной Системы

ИС состоит из взаимосвязанных элементов, которые можно с точки зрения диагностики представить в виде четырех уровней.

Элементы (уровни) ИС начиная с нижнего:

1. пассивное оборудование;
2. активное оборудование;
3. драйверы и сервисы ОС;
4. приложение (ПО).

Очень важно понимать, что все уровни взаимосвязаны и появление проблем в каком-либо из элементов всегда проявляется на более высоких уровнях. Однако, обнаружив проблемы на уровне приложения (например, замедление работы ПО), невозможно понять, на каком уровне они возникли. Из этого следует основное правило – диагностику необходимо проводить пошагово, на всех уровнях, начиная с нижнего.

На каждом уровне применяются свои средства диагностики. Заранее стоит предостеречь – ни одно из средств не делает выводов и не говорит о причинах проблемы. Это всегда приходится делать эксперту.

3 Уровень пассивного оборудования

3.1 Классификация пассивного оборудования

Пассивным оборудованием являются:

1. магистральные кабели;
2. кабели горизонтальной подсистемы;
3. информационные розетки;
4. рабочие соединительные кабели;
5. соединительные панели;
6. коммутационные шнуры.

3.2 Средства диагностики

Диагностика пассивного оборудования проводится кабельными тестерами. Они подразделяются на 3 группы:

1. простейший тестер;
2. кабельный тестер;
3. совмещенный кабельный тестер и сетевой анализатор.

Первые определяют правильность разводки витой пары, обрывы, короткое замыкание, а также показывают расстояние до обрыва или короткого замыкания. Стоят они в десятки раз меньше, чем кабельные тестеры.

Кабельные тестеры проводят полную диагностику состояния кабельной системы и сертифицируют ее на соответствие определенной категории. Также они могут иметь минимальные функции по мониторингу сети (загрузка сегмента, ошибки и т.д.) и возможность фиксирования внешних импульсных помех.

Приборы третьей группы не могут соперничать ни с тестерами, ни с сетевыми анализаторами. В первую очередь они предназначены администраторам сетей для выявления несложных и часто встречающихся проблем.

Более подробную информацию о тестерах и ссылки на статьи, посвященные диагностике пассивного оборудования, можно получить на сервере http://www.fluke.ru.

3.3 Диагностика пассивного оборудования

Методика диагностики довольно простая – достаточно нажать только одну кнопку на кабельном тестере, чтобы получить полный отчет о состоянии пассивного оборудования и соответствии его определенной категории.

На рис. 3.1 представлен пример отчета OMNIScanner™2.

OMNI Scanner

PASS

OMNIScanner2 Certification Report

Circuit ID:	test5e
Project:	TIA Project
Owner:	OmniScanner
Autotest:	*Cat 5E Chan*
Cable:	Cat 5E UTP
NVP:	72
Site:	----
Building:	----
Floor:	----
Closet:	----

OMNIScanner 50D01K00127 **SW:** V05.01
Adapter MT CAT6 SSTP

OMNIRemote 50E01J00054
Adapter CHAN 5/5E/6

	Limit	12	36	45	78
Length m	(100,0)	2,0	2,0	2,0	2,0
Delay (ns):	(555)	9	9	9	9
Resistance (Ohms):	(----)	----	----	----	----

Wiremap	Expected	Actual			
OMNI:	12345678	12345678S	Skew (ns):	(50)	0
Remote:	12345678	12345678S	Bandwidth (MHz):		----

Attenuation

Overall Margin (dB)[1] 22,6

Pairs	dB	Margin	MHz
12	0,7	23,3	99,4
36	0,7	23,3	99,4
45	0,6	22,9	96,7
78	0,6	22,6	94,3

Return Loss

Overall Margin (dB)[1] ----

	OMNIScanner			OMNIRemote		
Pairs	dB	Margin	MHz	dB	Margin	MHz
12	23,7	11,9	66,0	19,8	9,8	99,9
36	22,3	12,3	99,9	19,8	9,5	94,5
45	20,9	9,4	71,4	21,1	11,1	99,4
78	21,6	11,5	99,4	20,3	10,3	99,9

NEXT

Overall Margin (dB)[1] 23,6

	OMNIScanner			OMNIRemote		
Pairs	dB	Margin	MHz	dB	Margin	MHz
12/36	77,6	32,8	13,6	60,1	30,0	99,7
12/45	61,5	30,1	84,4	60,0	29,3	92,7
12/78	78,1	31,3	10,4	80,2	34,2	11,5
36/45	92,3	33,4	1,9	82,2	35,7	10,9
36/78	77,2	30,6	10,6	78,8	35,6	16,9
45/78	53,9	23,6	97,9	58,0	27,8	99,0

ACR

Overall Margin (dB)[1] ----

	OMNIScanner			OMNIRemote		
Pairs	dB	Margin	MHz	dB	Margin	MHz
12/36	----	----	----	----	----	----
12/45	----	----	----	----	----	----
12/78	----	----	----	----	----	----
36/45	----	----	----	----	----	----
36/78	----	----	----	----	----	----
45/78	----	----	----	----	----	----

ELFEXT

Overall Margin (dB)[1] 24,0

	OMNIScanner			OMNIRemote		
Pairs	dB	Margin	MHz	dB	Margin	MHz
12/36	64,1	28,3	12,0	63,7	28,8	13,3
12/45	60,6	26,4	14,5	61,7	26,6	13,1
12/78	75,6	40,0	12,4	77,9	41,7	11,5
36/12	63,7	28,8	13,3	64,2	28,3	12,0
36/45	90,3	34,3	1,2	92,2	34,4	1,0
36/78	71,9	36,7	12,9	72,3	36,6	12,2
45/12	61,6	26,6	13,1	60,6	26,4	14,5
45/36	92,2	34,4	1,0	90,2	34,3	1,2
45/78	59,9	24,1	12,0	78,4	24,0	1,4
78/12	77,9	41,7	11,5	75,5	40,0	12,4
78/36	72,3	36,6	12,2	71,9	36,7	12,9
78/45	78,5	24,0	1,4	59,9	24,1	12,0

PSNEXT

Overall Margin (dB)[1] 26,0

	OMNIScanner			OMNIRemote		
Pairs	dB	Margin	MHz	dB	Margin	MHz
12	60,2	31,8	84,4	57,1	29,9	98,8
36	76,3	32,6	10,6	59,5	32,4	99,7
45	53,4	26,0	96,5	55,8	28,6	99,0
78	53,5	26,2	97,9	57,8	30,6	99,0

PSELFEXT

Overall Margin (dB)[1] 25,1

	OMNIScanner			OMNIRemote		
Pairs	dB	Margin	MHz	dB	Margin	MHz
12	59,8	27,7	13,1	60,6	27,7	12,0
36	85,3	30,5	1,0	63,3	31,4	13,3
45	75,3	25,1	1,6	58,9	25,4	11,1
78	59,7	26,8	12,0	78,2	26,7	1,4

PSACR

Overall Margin (dB)[1] ----

	OMNIScanner			OMNIRemote		
Pairs	dB	Margin	MHz	dB	Margin	MHz
12	----	----	----	----	----	----
36	----	----	----	----	----	----
45	----	----	----	----	----	----
78	----	----	----	----	----	----

[1] Overall margin value is the worst margin for OMNI and Remote.

MICROTEST®

Date: 23.05.2002 16:43:00 Page 1

рис. 3.1 Образец отчета кабельного тестера OMNIScanner™2

В недавно построенных сетях проблемы с пассивным оборудованием практически не возникают, т.к. системные интеграторы стараются придерживаться стандартов и тестируют кабельную систему при сдаче. В старых сетях наиболее часто проблемы возникают при подключении компьютеров UTP кабелем горизонтальной проводки к активному оборудованию и неправильной прокладкой и подключением коаксиальных кабелей.

Проблемы, возникающие на уровне пассивного оборудования, можно разделить на две основные группы:

1. нарушение стандартов при проектировании и установке пассивного оборудования;
2. внешнее воздействие на пассивное оборудование.

Диагностирование проблем первой группы представляет собой наиболее простую задачу, т.к. стандартизованы не только пассивное оборудование, но и правила его установки.

Проблемы второй группы диагностировать очень тяжело. Если в сети возникают ошибки, то с помощью доступных средств диагностики практически невозможно понять, что их вызывает. Если помеха от электроприбора накладывается на передаваемый пакет – мы просто увидим пакет с неверной суммой CRC. То же самое мы увидим и во время коллизии. Конечно, если мы увидим некий “мусор” без положенной для пакета преамбулы, можно предполагать, что это внешняя помеха. Однако существует вероятность того, что наложение пакетов произошло во время передачи преамбулы. С достаточной долей уверенности можно говорить о причинах ошибок после нагрузочного тестирования. Ошибки уровня пассивного оборудования слабо влияют на скорость работы компьютеров, и их количество не увеличивается с повышением нагрузки тестирования. Если ошибки вызваны более высокими уровнями – скорость работы компьютеров в сети падает катастрофически.

Импульсные помехи на кабеле без подключенного активного оборудования может обнаруживать OMNIScanner™2. Он регистрирует импульсы, превышающие 30 mv. Согласно тестам, проведенным компанией ITT NS&S, наводки от 38 до 123 mv в зависимости от пары начинают появляться при внешнем шуме 3 v/m (оригинал статьи с описанием теста – http://www.ittnss.com/files/whitepapers/emctx100.pdf). Также специалисты ITT NS&S утверждают, что при этих помехах концентратор показывал загрузку в сети более 80% (при отсутствии передачи данных). Однако по следующим причинам к данному тесту стоит относиться весьма скептически:

- компания откровенно продвигает свои экранированные кабельные системы;
- непонятно, что за кабель неэкранированной витой пары они использовали;
- концентратор не может служить измерительным устройством;
- не проверялось воздействие на передаваемые данные.

Да, в кабеле появились наводки, но остался невыясненным вопрос – как же все-таки они повлияют на реально передаваемые данные, тем более на современном оборудовании (тест проводился в 1995 году). К сожалению, никакой другой информации по внешним помехам найти не удалось, что свидетельствует о сложности проблемы. Практически невозможно ответить на вопрос: как внешние шумы повлияют на передаваемые данные. Даже если и будет зафиксировано изменение передаваемого сигнала при определенном уровне помех (для чего нужен записывающий осциллограф) – нельзя сказать, что точно такие же помехи вызовут сбой на другой кабельной системе с другим активным оборудованием. Наши эксперименты с OMNIScanner™2 показали, что в то время, когда

OMNIScanner™2 регистрировал импульсные шумы, аппаратный анализатор WinPharaoh не зафиксировал никаких ошибок при передачи данных.

Несмотря на то, что диагностирование внешних помех является достаточно сложной задачей, нахождение проблемы достаточно простое. Обычно сами администраторы предупреждают о холодильниках и электрощитах. О них следует вспомнить, когда вы провели полную диагностику и не нашли причин ошибок в сети. Недопущение и устранение подобных проблем еще более простое – следуйте стандартам на установку пассивного оборудования.

3.4 Примеры

В качестве примеров - реальные случаи проблем с пассивным оборудованием.

Пример 1. *Диагностика сети на 100 пользователей, ориентированной на бухгалтерские задачи.*

Проблемы в сети:

Периодическое пропадание связи с сервером Oracle.

Этап 1

Цель:

Предварительное обследование.

Структура ЛВС:

В результате предварительного обследования было выяснено, что в сети находится несколько серверов под управлением OS Novell, а в качестве системы управления базами данных *используется Oracle. Компьютерная сеть построена с использованием только концентраторов. Существует центральный коммутационный шкаф, в котором находится концентратор 3Com SuperStack II Dual Speed Hub 500 и подключенные к нему концентраторы SynOptics. Удаленные комнаты имеют свои концентраторы, подключенные к 3Com. Таким образом, сеть существует в виде одного сегмента. К нему подключено около 100 пользователей. Кабель горизонтальной проводки обжат RJ-45 и напрямую подключен к активному оборудованию.*

Методика и средства:

Диагностика сети проводилось во время обычной работы пользователей в сети с помощью аппаратно-программного анализатора WinPharaoh компании GNNettest. Анализатор подключался к каждому концентратору в центральном коммутационном шкафу.

Результаты:

Статистическая система анализатора зафиксировала достаточно большое количество ошибок типа Bad CRC и Runt.

Этап 2

Методика тестирования и средства:

Тестирование сети проводилось на следующий день во время обычной работы пользователей в сети с помощью аппаратно-программного анализатора WinPharaoh компании GNNettest. Никаких изменений после первого тестирования замечено не было. Выборочно была проверена кабельная проводка с помощью PentaScanner. Исследовались линии подключения серверов, отдельных сегментов и пользователей к центральному коммутационному шкафу.

Результаты:

На линии 700 был обнаружен разрыв 6-го провода витой пары, на линии 10 – 6 и 4-го.

Вывод:

С большой долей вероятности можно предположить, что плохие контакты в витой паре и являются источником снижения помехоустойчивости витой пары, а соответственно и обнаруженных в ЛВС ошибок. Повреждения кабельной системы обусловлены тем, что кабельная проводка ЛВС выполнена с грубыми нарушениями стандартов.

Рекомендации:

До следующих этапов диагностики необходимо полностью заменить кабельную проводку.

На самом деле довольно сложно понять, что именно вызывает ошибки в сети, однако в данном случае во время дополнительных работ по диагностике было выявлено следующее: работа компьютеров в сети не замедляется и не влияет на количество ошибок, что свидетельствует о возникновении ошибок на уровне пассивного оборудования. В любом случае, в кабельной системе были найдены проблемы, и их необходимо было устранять.

Пример 2. *Диагностика городской оптической сети, объединяющей несколько крупных локальных сетей.*

Цель:

Выявление причин крайне медленной работы компьютеров в сети.

Структура ЛВС:

В результате предварительного обследования было выяснено, что сеть объединяет несколько зданий и построена с помощью повторителей *расстояние между максимально удаленными точками – 5 км.*

Методика и средства:

Диагностика сети проводилось во время обычной работы пользователей в сети с помощью аппаратно-программного анализатора WinPharaoh компании GNNettest. Анализатор подключался к сегменту сети.

Результаты:

Статистическая система анализатора зафиксировала большое количество ошибок типа Bad CRC, ALIGNMENT и JABBER.

Вывод:

Превышен диаметр коллизионного домена, что вызывает повреждение и ошибки в сети.

Рекомендации:

Сегментировать сеть.

Проводить диагностику данной сети особого смысла не было, т.к. после первых же вопросов о топологии выяснилось, что превышен диаметр коллизионного домена, поэтому и ожидать чего-то иного от этой сети не приходилось.

4 Уровень активного оборудования

4.1 Типы активного оборудования

Под активным оборудованием условимся понимать устройства, занимающиеся передачей и приемом информации в сети. Это коммутаторы, концентраторы, маршрутизаторы, сетевые адаптеры сетевые сервисные устройства и т.д. Нас интересует только их работа на физическом и канальном уровне модели OSI, т.е. работа их сетевых интерфейсов.

Проблемы активного оборудования можно разделить на две основные группы:

1. поломка оборудования;
2. неверная настройка.

Методика диагностики не зависит от того, к какой группе принадлежит проблема, но понимание деления необходимо для правильного решения проблемы.

Проблемы на уровне оборудования проявляются только в виде ошибочных сетевых пакетов. Для их выявления необходим сетевой анализатор.

4.2 Средства диагностики

Сетевые анализаторы подразделяются на программные и программно-аппаратные. Первые представляют собой программы, запускаемые на обычных компьютерах (ноутбуки). Программно-аппаратные состоят из специализированных аппаратных решений, предназначенных для диагностики сетей с различными интерфейсами, и программных средств, исполняемых на отдельном или интегрированном компьютере. И то и другое решение имеет свои плюсы и минусы.

Программные анализаторы:

- Дешевле, чем аппаратные;
- имеют больший набор функций;
- проще обновляются.

Аппаратные анализаторы:

- могут иметь одновременно различные интерфейсы;
- имеют интерфейсы, с которыми не могут работать компьютеры;
- не зависят от производительности компьютера (подключаемый компьютер предназначен для отображения результатов диагностики, выполненного аппаратным интерфейсом);
- доподлинно отображают все происходящее на канальном уровне.

Методику и примеры диагностики канального уровня сетей Ethernet будем рассматривать на базе аппаратно-программного анализатора WinPharaoh компании GNNettest (http://www.gnnettest.com/pages/wp.htm). Его изображение приведено на рис. 4.1**.**

рис. 4.1 Аппаратно-программный анализатор WinPharaoh.

Диагностику транспортного и более высоких уровней – на базе анализатора Observer компании Network Instruments

(http://www.netinst.com/html/observer_suite.html). Изображение Observer приведено на рис. 4.2.

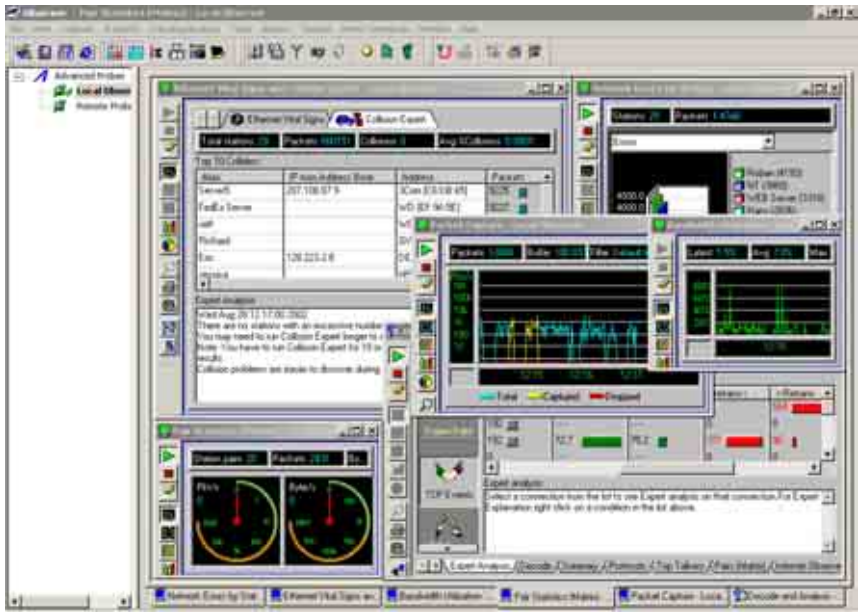

рис. 4.2 Observer

Не будем подробно останавливаться на функциях и возможностях сетевых анализаторов, определимся только с наиболее необходимыми из них для полноценной диагностики. Анализатор должен уметь:

- захватывать трафик;
- отображать трафик во время захвата;
- декодировать пакеты;
- отображать сетевые ошибки;
- проводить долговременный сбор статистики с последующим сравнением по выбранным диапазонам;
- анализировать потоки как минимум на 2 и 3 уровне модели OSI;
- фиксировать повтор пакетов;
- проводить стрессовое тестирование;
- создавать отчеты;
- сигнализировать о проблемах в сети;
- снимать и протоколировать статистику с сетевых устройств по протоколам SNMP и RMON;
- показывать скорость потока и задержки от каждой из сторон потока в реальном режиме времени;
- работать с удаленными агентами.

Если с аппаратными анализаторами все ясно, – они предназначены для того, чтобы диагностировать сети – то на некоторых вопросах в использовании программных анализаторов следует остановиться подробнее.

Для того, чтобы диагностировать сеть, ее необходимо прослушивать, т.е. сетевая карта компьютера, на котором установлен анализатор, должна включаться в режим promiscuous mode (режим передачи уровню драйвера всех захваченных пакетов, а не только отправленных на данный сетевой адаптер). Но если с этим проблем практически не возникает, и карты захватывают из сети все пакеты, то с поврежденными пакетами все гораздо хуже. Сетевые карты их игнорируют, что, в принципе, абсолютно правильно. В качестве решения производители программных анализаторов предлагают вместе со своими продуктами специализированные

карты. В компании Network Instruments пошли другим путем, они написали свои драйверы для распространенных сетевых карт.

Существует еще одна проблема. Дело в том, что программный анализатор обрабатывает поступающие из сети данные программно, т.е. производительность анализатора зависит от мощности компьютера. Кстати, во время тестирования программных анализаторов с помощью аппаратного выяснилось, что их производительность очень сильно зависит от них самих. Например, Network Monitor показывал загрузку 10Мбит/с сегмента - 60% при 100% нагрузке от аппаратного анализатора, в то время как на этом же компьютере (Windows NT, PII-266, ОЗУ 64М) Observer показал загрузку 100Мбит/с сегмента - 80% при 100% нагрузке. Мощности современных компьютеров достаточно для полной дешифровки и экспертного анализа сетевого трафика канала 100Мбит/с и даже 1000Мбит/с. Системные требования для программного анализатора Observer приведены в табл. 4.1.

табл. 4.1

Системные требования для Observer

	Минимум		**Рекомендуется**	
	Windows 98/ME	**NT/2000/XP**	**Windows 98/ME**	**NT/2000/XP**
802.11 Wireless	-	Pentium 266 128MB RAM	-	Pentium 400 128MB RAM
10MB Ethernet	Pentium 266 64MB RAM	Pentium 266 128MB RAM	Pentium 400 128MB RAM	Pentium 400 128MB RAM
100MB Ethernet	Pentium 400 128MB RAM	Pentium 400 128MB RAM	Pentium III 900 128MB RAM	Pentium III 900 256MB RAM
4MB Token Ring	Pentium 266 64MB RAM	Pentium 266 128MB RAM	Pentium 400 128MB RAM	Pentium 400 128MB RAM
16MB Token Ring	Pentium 266 128MB RAM	Pentium 266 128MB RAM	Pentium 400 128MB RAM	Pentium 400 128MB RAM
FDDI	Pentium 400 128MB RAM	Pentium 400 128MB RAM	Pentium II 600 256MB RAM	Pentium III 600 256MB RAM
Gigabit (1000Mbit)	Pentium III 600 128MB RAM	Pentium III 800 128MB RAM	Pentium 1.4Ghz 256MB RAM	Pentium III 1.4Ghz 512MB RAM
Gigabit (full-duplex, wire-speed) Требуется Network Instruments' Dual-Receive Gigabit NIC	-	-	-	Dual Processor Pentium III 1.4Ghz 512MB RAM

4.3 Диагностика в коммутируемой сети

Диагностика в коммутируемой сети имеет свои особенности, т.к. подключившись к порту сетевого оборудования, мы можем прослушивать данные, передаваемые только с этого или на этот порт. Для прослушивания данных, передаваемых через другие порты на сетевом оборудовании необходимо включать режим зеркального отображения если, конечно, такая функция в оборудовании есть.

Возможности режима зеркалирования могут отличаться для разного оборудования. Порт, включенный на прослушивание данных с другого порта, может перейти в блокированный режим, и выполнять только функции прослушивания. Некоторое оборудование может позволять перенаправлять данные сразу из

нескольких прослушиваемых портов, что, правда, не совсем корректно, и может вызвать потерю части данных. Ошибки, зафиксированные прослушиваемым портом, могут перенаправляться или уничтожаться.

Однако обычно достаточно просмотреть статистику работы портов. Практически любое управляемое активное сетевое оборудование позволяет это сделать при помощи протоколов SNMP (Simple Network Management Protocol - простой протокол сетевого управления) и RMON (Remote MONitoring – удаленный мониторинг). Рассмотрим их возможности, применяемые в диагностике.

По протоколу SNMP можно получить сведения о настройках оборудования, информацию о работе портов. А также можно узнать общее количество широковещательных пакетов, байтов, ошибок, пакетов и отброшенных не ошибочных пакетов за время работы агента SNMP.

SNMP не показывает загрузку канала в процентном отношении. Некоторые средства мониторинга самостоятельно ее рассчитывают на основе данных о принятых и переданных байтах.

В отличие от протокола SNMP, предназначенного в основном для управления, его расширение – протокол RMON – является сетевым анализатором и позволяет собирать большое количество информации о передаваемых данных и работоспособности портов. RMON состоит из 19 групп. Наиболее необходимая при диагностике сети – первая группа (статистика). Из группы статистики мы можем получить следующую информацию о работе порта (общее количество за время работы агента RMON):

- проигнорированных событий – etherStatsDropEvents;
- принятых из сети байт – etherStatsOctets;
- полученных пакетов – etherStatsPkts;
- широковещательных пакетов – etherStatsBroadcastPkts;
- многоадресных пакетов – etherStatsMulticastPkts;
- ошибок CRC – etherStatsCRCAlignErrors;
- пакетов нестандартной длины – etherStatsUndersizePkts и etherStatsOversizePkts;
- пакетов менее 64 байт и ошибкой CRC – etherStatsFragments;
- пакетов более 1518 байт и ошибкой CRC – etherStatsJabbers;
- коллизий – etherStatsCollisions;
- пакетов размером 64 байта – etherStatsPkts64Octets;
- пакетов размером от 65 до 127 байт – etherStatsPkts65to127Octets;
- пакетов размером от 128 до 255 байт – etherStatsPkts128to255Octets;
- пакетов размером от 256 до 511 байт – etherStatsPkts256to511Octets;
- пакетов размером от 512 до 1023 байт – etherStatsPkts512to1023Octets;
- пакетов размером от 1024 до 1518 байт – etherStatsPkts1024to1518Octets.

К сожалению, из-за ограниченных функциональных возможностей сетевое оборудование обычно поддерживает не более четырех групп. А группы после десятой относятся к информации транспортного уровня и оборудованием, работающим только на канальном уровне, поддерживаться не могут.

Пример информации RMON группы статистики коммутатора 3Com SuperStack II 3300 приведен на рис. 3.1.

Также начинает внедряться и новый протокол – SMON (Switch Monitoring), собирающий информацию о работоспособности и передачи данных на уровне интеллектуального ядра устройства.

4.4 Классификация сбоев работы сетей Ethernet

Четкое деление и классификация сбоев передачи в сетях Ethernet имеет очень важное значение для проведения диагностики, что будет показано на опытах и примерах ниже.

Сбои в сетях Ethernet можно разделить на две группы. Это сбой, в результате которого произошло либо повреждение пакета, либо возникновение нестандартного пакета.

Дальнейшую детализацию лучше всего отобразить в табличном виде (табл. 4.2).

табл. 4.2

Классификация сбоев передачи в сетях Ethernet

<table>
<tr><th>Результат</th><th>Тип сбоя</th><th>Причина сбоя</th></tr>
<tr><td rowspan="3">повреждение пакета</td><td rowspan="2">Fragment
ошибка контрольной суммы,
пакет менее 64 Байт</td><td>коллизия</td></tr>
<tr><td rowspan="3">сбойная работа сети</td></tr>
<tr><td>CRC Error
ошибка контрольной суммы,
пакет больше 64 Байт</td></tr>
<tr><td>Jabber
ошибка контрольной суммы,
пакет больше 1518 Байт</td></tr>
<tr><td rowspan="2">нестандартный пакет</td><td>Undersize
верная контрольная сумма,
пакет менее 64 Байт</td><td rowspan="2">сбойная работа оборудования</td></tr>
<tr><td>Oversize
верная контрольная сумма,
пакет больше 1518 Байт</td></tr>
</table>

Примечание: в случае, если размер пакета не кратен 8, фиксируется ошибка выравнивания – Alignment Error. Такая ошибка может встречаться при всех типах повреждения пакетов.

Классификация приведена согласно стандарту RFC 1757 “Remote Network Monitoring Management Information Base”. В качестве примера на рис. 4.3 приведен фрагмент таблицы etherStatsTable статистики RMON коммутатора 3Com SuperStack II 3300.

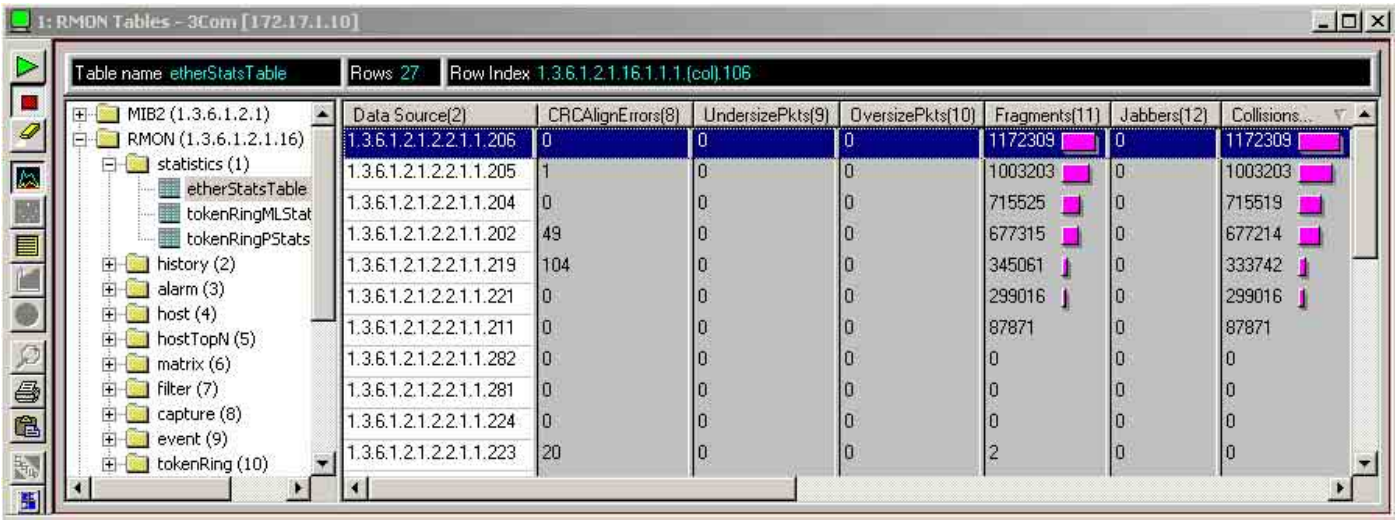

рис. 4.3 Фрагмент таблицы etherStatsTable статистики RMON коммутатора 3Com SuperStack II 3300

Повреждение пакета всегда проявляется в виде ошибки контрольной суммы, а вот понять, в результате чего пакет был поврежден – задача сложная и неординарная.

Повреждение пакетов может произойти в следующих случаях:

1. в результате наложения пакетов во время нормальной работы сети;
2. если одна из станций перестала прослушивать сеть перед передачей;
3. если работающие друг с другом сетевые интерфейсы установлены в разные режимы (FullDuplex на HalfDuplex);
4. если превышен диаметр коллизионного сегмента;
5. при повреждении пассивного оборудования;
6. от внешних помех.

Во всех этих случаях сетевой анализатор зафиксирует в сети пакет с неверной контрольной суммой.

4.5 Различия ошибок повреждения пакетов

Коллизия – столкновение (искажение содержимого) при одновременной передаче пакетов в сеть двумя или более станциями в одном сегменте, произошедшее во время нормальной работы сети.

Коллизия не является результатом сбоя работы сети. Возникновение ситуации наложения пакетов в сети подразумевается протоколом Carrier Sense, Multiple Access with Collision Detection (CSMA/CD). Самое главное отличие коллизии от ошибки в том, что коллизия обрабатывается и исправляется средствами сетевого интерфейса. В то время как при любом повреждении пакета в результате сбоя работы сети (сбои на физическом уровне; передача данных не соответствует протоколу CSMA/CD или стандартам физического уровня), можно говорить, что в сети произошла ошибка, которая с большой долей вероятности будет обрабатываться транспортным уровнем (по классификации модели OSI).

Сложность установления факта, что произошла именно коллизия, связана с функциональными особенностями работы Ethernet сетей.

Анализатор, подключенный к порту 10BASE-5/10BASE-2 в режиме прослушивания, фиксирует коллизию только в случае, если три или более станций начали одновременную передачу; в режиме передачи – сравнивает передаваемые данные с принимаемыми, и при их различии фиксирует коллизию. Порт повторителя 10BASE-5/10BASE-2 должен зафиксировать коллизию если две или больше станций начали одновременную передачу.

Порт 10BASE-T/100BASE-T фиксирует коллизию только в том случае, если во время передачи у него на приемном порту появился сигнал. Соответственно анализатор сети 10BASE-T/100BASE-T, работающий в режиме прослушивания, даже теоретически не может зафиксировать коллизию на своем или другом порту.

Однако даже если мы увидели, что действительно произошло наложение (искажение содержимого), а не просто получили поврежденный пакет, все равно однозначно не известно, что именно вызвало наложение, и произошло ли оно во время нормальной работы.

4.6 Способы обнаружения коллизий и ошибок

В результате диагностики мы всегда видим только результат коллизии или ошибки – пакет с неверной контрольной суммой или вообще “мусор”. К тому же, после наложения пакетов можно увидеть MAC адрес только первого пакета, и то только в том случае если наложение произошло после его заголовка. Понять, что было причиной сбоя – ошибка или коллизия, и какое именно сетевое оборудование вызвало сбой, – можно только косвенными способами.

На первый вопрос можно ответить достаточно просто. Если в сегменте нет поврежденных пакетов, превышающих 64 байта (притом, что пакеты более 64 байт используются), можно практически с полной уверенностью говорить о том, что это только коллизии. В случае, если зафиксированы даже единичные поврежденные пакеты более 64 байт, пакеты менее 64 байт тоже могут быть вызваны сбоем работы сети.

Для определения оборудования, вызывающего коллизии, GNNettest рекомендует поочередно отключать оборудование и контролировать уровень коллизий. Естественно, данный способ применим только в том случае, если других вариантов нет.

Observer для выявления коллизий и подозрительных станций создает небольшую нагрузку сети и контролирует станции, начавшие передачу непосредственно до или после коллизии. Observer Collision Expert приведен на рис. 4.4.

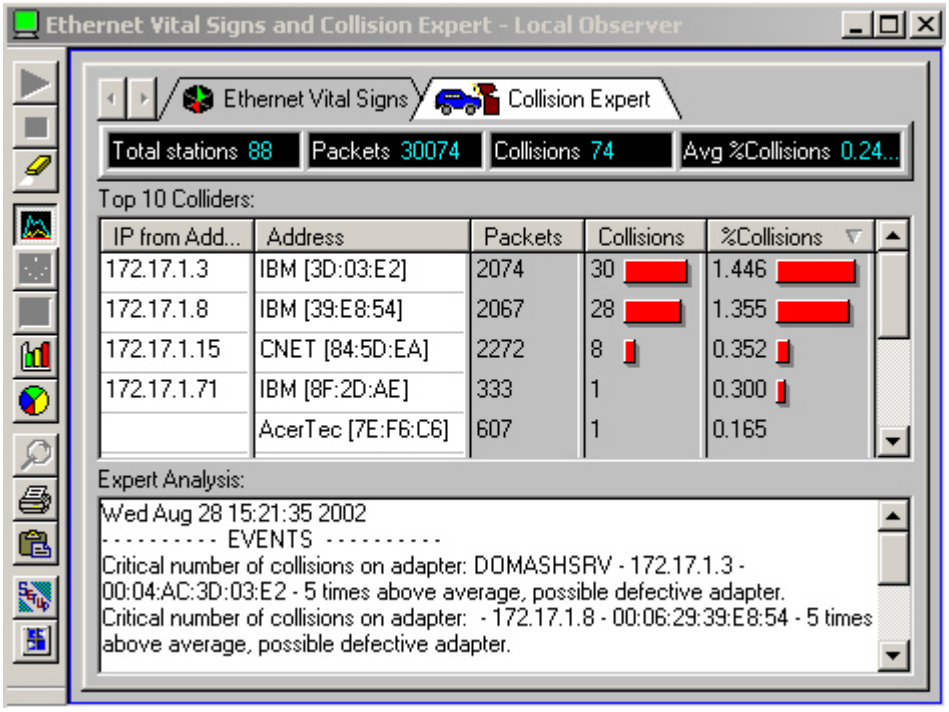

рис. 4.4 Observer Collision Expert

Искать оборудование, которое вызывает коллизии, особого смысла не имеет, т.к. крайне маловероятно, что коллизии будут создавать проблемы в работе сети. Как будет видно ниже из эксперимента, при уровне коллизий около 5% работоспособность сети остается на приемлемом уровне. Во время диагностики одной из сетей мы с помощью 8 станций, работающих с бухгалтерским сервером, довели нагрузку 10Мбит/с сегмента до 80% без существенного падения скорости работы приложений.

В качестве диагностического устройства бессмысленно использовать встроенный в концентратор индикатор уровня коллизий. Даже если индикатор загорелся и не гаснет во время работы сети – это свидетельствует только о том, что за секунду в сети происходит не менее 25 коллизий. Для сегмента 10Мбит/с при нагрузке 80% это составляет примерно четверть процента от переданных пакетов.

Если все же в сети есть проблемы с коллизиями, то, скорее всего, они возникают от слишком большого количества компьютеров, работающих в одном сегменте. Рекомендация может быть только одна – рассегментировать сеть.

Обнаружить компьютер, создающий в сети ошибки, более просто. Но для этого нужно точно знать, что в сегменте есть ошибки, и они не связаны с проблемами уровня пассивного оборудования. Для подобного тестирования можно применять пакет стрессового тестирования сети FTest, компании Пролан (www.prolan.ru). Более подробно о FTest будет рассказано в разделе диагностики уровня драйверов и сервисов.

4.7 Описание эксперимента по исследованию влияния коллизий и ошибок на скорость работы приложения

Для наглядного представления отличий между коллизиями и ошибками были проведены два эксперимента. В первом исследовалась зависимость скорости работы приложения от коллизий, во втором – зависимость скорости работы приложения от ошибок.

Схема стенда приведена на рис. 4.5.

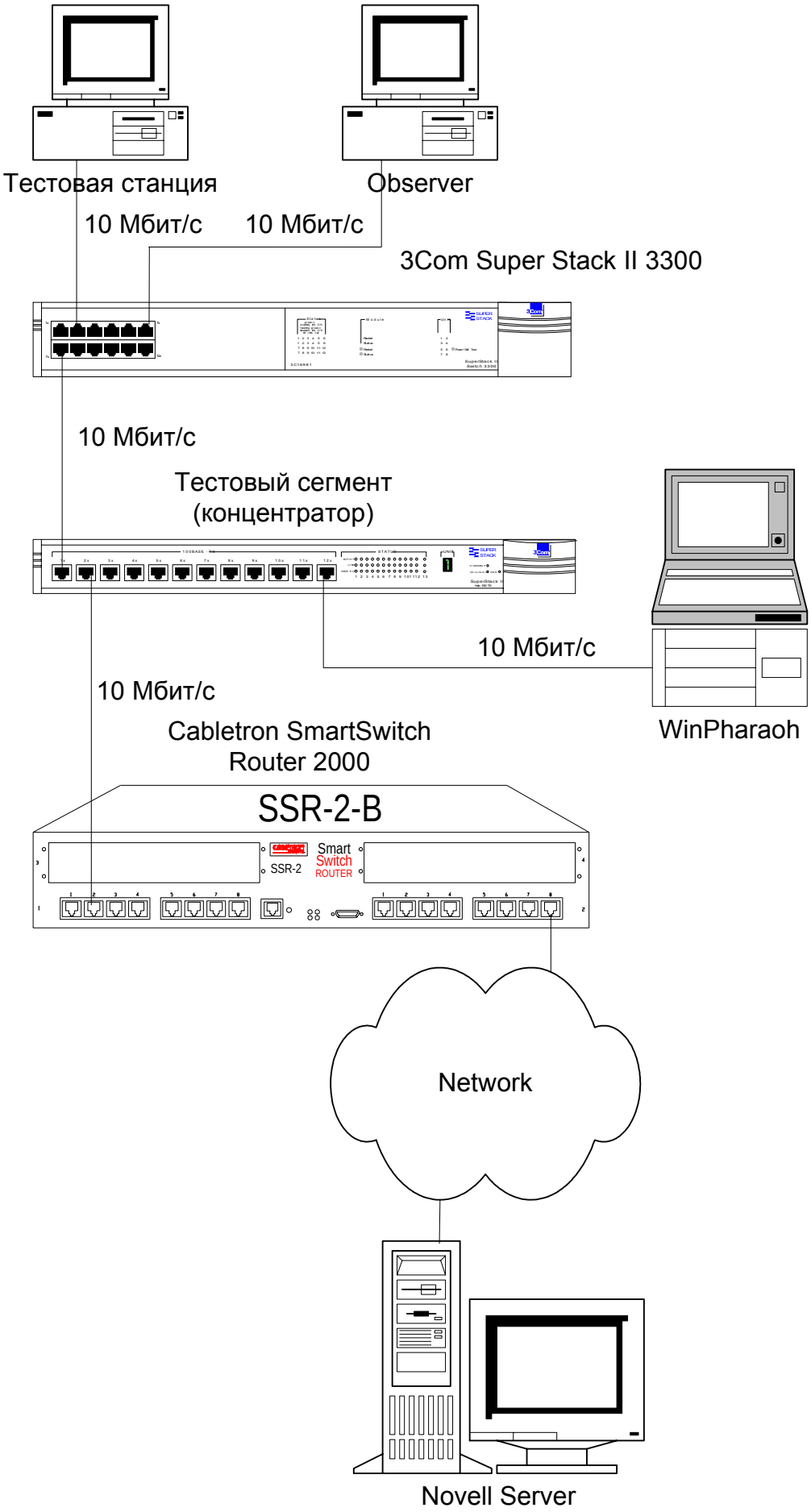

рис. 4.5 Схема стенда.

В качестве тестового приложения на рабочей станции запускалась задача обращения к файловой базе на сервере Novell, транспортный протокол – SPX. Используемая задача при работе в сети только сервера и клиента загружает канал 10 Мбит/с примерно на 3,6% и выполняется в течение 1,03 мин.

Для исследования коллизий между коммутаторами был создан сегмент сети Ethernet.. Коммутаторы Cabletron SmartSwitch Router 2000 (http://www.enterasys.com/products/items/SSR-2-B128) и 3Com Super Stack II 3300 (http://www.3com.com/products/switches/sw1100_3300_family.html) были подключены через концентратор 10 Мбит/с. Подобный канал можно было бы создать и путем конфигурации портов коммутаторов в режим 10 Мбит/с HalfDuplex, но концентратор также предназначался для подключения аппаратно-программного анализатора WinPharaoh, с помощью которого создавалась изменяющаяся фоновая нагрузка данного сегмента сети. Для создания в сети ошибок CRC (потерь пакетов с их повтором), порт коммутатора 3Com Super Stack II 3300, подключенный к концентратору, принудительно переводился в режим 10 Мбит/с FullDuplex. Кстати, это весьма распространенная ошибка конфигурирования активного сетевого оборудования. В результате Super Stack II 3300 не “слушал” сеть, и пакеты уничтожались при столкновении с пакетами от WinPharaoh. Статистика количества коллизий и ошибок снималась с порта Super Stack II 3300, подключенного к концентратору, с помощью RMON модуля Observer.

Нагрузка в сети создавалась искусственным образом и никак не зависела от сетевых ошибок. Если бы нагрузка создавалась реальными приложениями, она бы снижалась при появлении ошибок и коллизий. Сервер был подключен через слабо загруженную рабочую сеть, нагрузка сервера другими пользователями не контролировалась. Однако эти условия крайне мало влияли на результат тестирования, в чем можно будет убедиться далее.

Тестирование проводилось поэтапно. На каждом этапе нагрузка тестового сегмента сети с помощью WinPharaoh увеличивалась на 10%, одновременно измерялась общая нагрузка тестового сегмента, которая состояла из нагрузки от WinPharaoh и нагрузки от приложения. Среднее количество коллизий и ошибок в секунду в тестовом сегменте, т.е. их относительное число, измерялось с помощью опции анализатора Observer - Ethernet Vital Signs. Т.к. их количество во время выполнения тестовой задачи колебалось значительно, приведены минимальные и максимальные значения.

Перейдем к результатам тестирования. В первом тесте (табл. 4.3) видно, что при нагрузках сети до 60% время выполнения тестовой задачи росло незначительно. После 70% время выполнения тестовой задачи начинает резко расти, уменьшается нагрузка сети от тестовой задачи и начинает уменьшаться количество коллизий. Во-первых, столь большое количество коллизий, достигающее 180 в секунду при нагрузке сети 63,9%, слабо влияет на скорость работы тестового приложения. Во-вторых, относительное количество коллизий начинает снижаться после достижения пика производительности сети. Практически такой же эффект наблюдается при анализе ошибок и говорит только об одном – в результате слишком большого количества коллизий и ошибок начинает падать скорость работы приложения в сети, а соответственно, и относительное количество коллизий и ошибок.

табл. 4.3

Результаты первого теста

№	Время выполнения тестовой задачи, мин.	Нагрузка тестового сегмента сети, создаваемая анализатором, %	Нагрузка тестового сегмента сети (анализатор + приложение), %	Среднее количество коллизий в секунду
1	1,03	10	13,8	29
2	1,03	20	23,8	60-64,8
3	1,03	30	33,9	77-96
4	1,03	40	44	135-148
5	1,03	50	54	150-170
6	1,10	60	63,9	160-181
7	1,38	70	73,4	138-201
8	2,55	80	81,2	120-150
9	9	90	90,7	68-95
10	17,4	99	97,8	40-60

Во втором тесте (табл. 4.4) при 10% нагрузке сети время выполнения тестового задания начинает катастрофически расти, и уже при 50% нагрузке стало бессмысленным продолжать тестирование. Однако относительное количество ошибок в десятки раз меньше и на первый взгляд практически не может влиять на работу сетевых приложений.

Для понимания результатов экспериментов разберемся, как в сети отрабатываются коллизии и ошибки. В случае возникновения коллизии сетевой интерфейс сам повторяет передачу пакета, и происходит это примерно за 0,5 мсек. Повтором уничтоженных пакетов занимается по тайм-ауту транспортный уровень протокола передачи и, естественно, он не может быть столь малым, как в случае коллизии. Начальное значение тайм-аута для данного тестирования составляет примерно 0,2 сек. Данные были получены с помощью WinPharaoh при анализе захваченных пакетов. Учитывая время передачи уничтоженного пакета и обработки ошибки, суммарное время повтора составляет примерно 0,5 сек, что в тысячу раз дольше, чем для обработки коллизии. Начальные и последующие значения тайм-аута повтора на транспортном уровне зависят от протокола и могут вычисляться на основе средней скорости передачи.

табл. 4.4

Результаты второго теста

№	Время выполнения тестовой задачи, мин.	Нагрузка тестового сегмента сети, создаваемая анализатором, %	Нагрузка тестового сегмента сети (анализатор + приложение), %	Среднее количество ошибок CRC в секунду
1	2	10	12	4,6-7,2
2	3,08	20	21,3	4,5-7,2
3	4,26	30	31	4,5-7,7
4	10,42	40	40,5	0,7-5,7
5	25,40	50	50,4	0,5-4,3
6	-	-	-	-
7	-	-	-	-
8	-	-	-	-
9	-	-	-	-
10	-	-	-	-

Из проведенных экспериментов видно, что коллизии мало влияют на производительность сети. Даже при уровне до 5% от всех переданных пакетов скорость работы приложения падала не значительно. В то же время при единичных случаях ошибок CRC скорость упала катастрофически, и сеть стала практически неработоспособной.

4.8 Примеры

Пример 1. *Поиск причин неработоспособности некоторых приложений, после переподключения компьютеров с концентратора на коммутатор.*

Проблемы в сети:

После переподключения отдела программистов с концентратора на коммутатор некоторые приложения на нескольких компьютерах перестали работать в сети. Проблемы были только на компьютерах с картами на одинаковых чипах.

Методика и средства:

Диагностика проводилась с помощью RMON модуля анализатора Observer.

Результаты:

Было зафиксировано большое количество ошибок Alignment Error на портах коммутатора. В результате более подробного разбирательства с сетевыми картами было установлено, что вероятность сбоя напрямую зависит от размера передаваемого пакета. И если пакеты размером 64 байта передавались практически без проблем, то пакеты максимальной длины практически всегда вызывали сбой на порту коммутатора. Аналогичные результаты были получены и при работе этих сетевых карт с коммутаторами разных производителей. Никаких ошибок при работе с концентраторами зафиксировано не было.

Тестирование работы приложений, используемых в этом отделе, показало, что одни приложения использует NetBIOS с размерами пакетов до 512 байт, а другие – TCP/IP с размерами пакетов до 1518 байт.

Вывод:

Единственным выводом в данной ситуации может быть только то, что сетевые карты неисправны и во время передачи начинают терять синхронизацию.

***Примечание**. Можно лишь предполагать, почему сетевые карты работали с концентраторами. Возможно, концентраторы более либерально относятся к форме и синхронизации импульсов, к тому же они выступают в роли повторителей, улучшая физические характеристики сигнала. Для того, чтобы с этим досконально разобраться, необходим записывающий осциллограф, работающий на частоте 10 МГц. Но это уже неоправданная роскошь, сетевую карту стоимостью $10 никто ремонтировать не будет, ее проще заменить.*

Рекомендации:

Заменить сетевые карты. В случае невозможности замены подключить через концентраторы.

***Пример 2.** Диагностика бухгалтерского сегмента заводской сети.*

Цель:

Выявление причин замедления работы бухгалтерского приложения с сервером на первом компьютере после включения второго компьютера, используемого периодически.

Структура ЛВС:

Оба удаленных бухгалтерских компьютера подключены к концентратору (также к концентратору подключены 3 компьютера, используемых редко). Концентратор и сервер подключены к центральному коммутатору.

Методика и средства:

Диагностика проводилась с помощью аппаратно-программного анализатора WinPharaoh. Анализатор подключался к концентратору.

Результаты:

При работе только первого бухгалтерского компьютера были зафиксированы коллизии и единичные ошибки типа Bad CRC. Скорость работы приложения с сервером была нормальной. После включения второго скорость работы обоих катастрофически падала, количество ошибок Bad CRC резко возрастало. При работе только второго компьютера ошибки не зафиксированы.

Вывод:

Сетевая карта первого компьютера не прослушивает сеть перед началом передачи. Т.к. при работе только с сервером передача идет по принципу запрос-ответ, ошибок практически не возникало. При включении второго компьютера, первый начинал уничтожать и свои и чужие пакеты.

Рекомендации:

Заменить сетевую карту, или настроить ее, если она при работе с концентратором была включена в режим FullDuplex.

***Пример 3.** Диагностика работы бухгалтерской ИС.*

Цель:

Определение причин периодических сбоев работы сервера 1С.

Структура ЛВС:

Семь бухгалтерских компьютеров подключены к концентратору (к остальным портам концентратора подключено подразделение менеджеров). Концентратор и сервер 1С подключены к центральному коммутатору.

Все проведенные действия и результаты были аналогичны второму примеру. Данный пример приведен только для демонстрации того, что одинаковые сбои проявляются в виде различных симптомов. Единственное отличие в том, что в третьем примере интенсивность работы бухгалтерских приложений была низкой, и снижение скорости в результате сбоев не было заметно. Но потери пакетов для сервера 1С при общении со своими клиентами оказались критическими, и вызывали сбои в работе сервера.

5 Уровень драйверов и сервисов

5.1 Интеллектуальное ядро и возможные сбои его работы

Под драйверами и сервисами будем понимать интеллектуальное ядро любого сетевого оборудования, обрабатывающее или передающее данные между приложением и уровнем активного оборудования.

Как-то классифицировать этот уровень сложно. Можно только грубо разделить активное сетевое оборудование на универсальные компьютеры и специализированные сетевые устройства. Для первых более характерна передача данных между уровнями, для вторых – обработка перед дальнейшей пересылкой.

Результаты сбоя работы интеллектуального ядра проявляются в виде:

1. потери/искажении данных;
2. замедлении передачи;
3. неверной обработке/передаче данных.

5.2 Предварительная диагностика

5.2.1 FTest

Для распознавания проблем с потерей, замедлением передачи или неверной обработкой данных необходим сетевой анализатор с функциями декодирования пакетов и экспертной системой. Однако, при большом количестве компьютеров, и особенно в коммутируемых сетях, диагностирование каждого сетевого устройства может занять очень много времени. И практически невозможно будет провести таким методом диагностику, если проблемы возникают только при совместной работе многих устройств, например, перегрузка маршрутизатора. Также бывает необходимо предварительно сравнить работоспособность одинаковых устройств.

Для предварительного тестирования существует достаточно простое, и в то же время очень эффективное средство – FTest, компании Пролан (http://www.prolan.ru/solutions/testing/ftest/index.html).

Программа FTest предназначена для диагностики локальных сетей методом нагрузочного (стрессового) тестирования.

В чем суть работы Ftest? На компьютерах устанавливаются агенты, которые, используя стандартные вызовы операционной системы, читают или записывают данные в тестовых файлах на любом сервере, к которому компьютеры имеют полный доступ. Управление агентами осуществляется с любой рабочей станции, на которой устанавливается базовая программа FTest. Есть три режима работы агентов:

1. все агенты работают вместе, постепенно наращивая нагрузку;
2. агенты добавляются по очереди; нагрузка, создаваемая агентами одинакова и не изменяется;
3. агенты запускаются и работают в одиночном режиме (калибровочный режим).

На стадии поиска установленных агентов (рис. 5.1) отображается информация об основных параметрах компьютеров (тип процессора, объем ОЗУ, тип сетевой карты), а также рассчитанный на базе этих и других параметров индекс производительности. В дальнейшем эти данные помогают более адекватно оценивать полученные результаты тестирования.

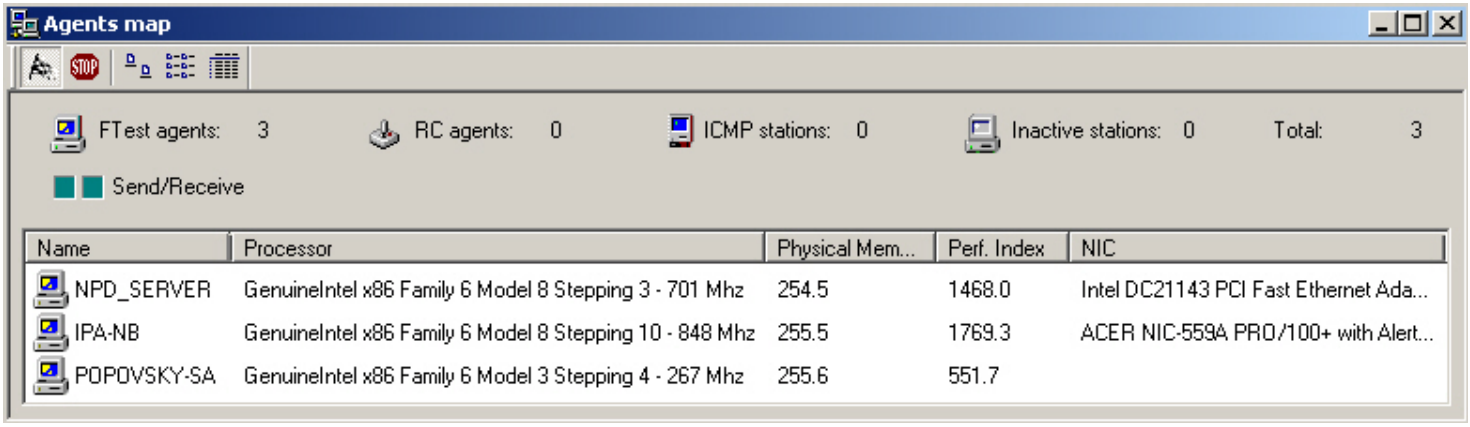

рис. 5.1 Окно отображения обнаруженных агентов

Выбор типа теста и параметров зависит от того, что мы хотим получить.

5.2.2 Совместная работа агентов

Тест при совместной работе всех агентов предназначен, в первую очередь, для определения максимальной нагрузочной способности ИС. По результатам теста видно, как и какой максимальной нагрузочной способности достигают тестируемые станции в данной сети с определенным сервером. Тест может применяться как во время сдачи-приемки ИС, так и для периодического контроля ее работоспособности. В качестве параметров задаются:

- Minimum Offered Load (Минимальная предлагаемая нагрузка) – значение нагрузки, которую предлагается создать всем станциям-агентам на первом шаге теста “FTest all stations”;
- Maximum Offered Load (Максимальная предлагаемая нагрузка) – значение нагрузки, которую предлагается создать всем станциям-агентам на последнем шаге теста “FTest all stations”;
- Number of steps (Число шагов) – число шагов, выполняемых всеми станциями-агентами в ходе теста;
- Duration of one step (Длительность шага) – продолжительность каждого шага теста в секундах;
- File Transaction Mode (Режим транзакций) – режим работы теста, при котором все файловые операции выполняются транзакциями;
- Share of read operations (%) (Доля операций чтения (%)) – процентная доля операций чтения в общем числе файловых операций, которые выполняет каждая станция-агент;
- Size of file (Размер файла) – размер тестового файла на тестовом сервере;
- Size of record (Размер записи) – размер записи в тестовом файле.

Если необходимо увидеть только максимальную нагрузочную способность данной ИС, можно задать минимальную и максимальные нагрузочные способности заведомо недостижимыми (например, 10 000 Кбайт/с) и один этап теста.

Результаты теста отображаются в виде графика или отчета.

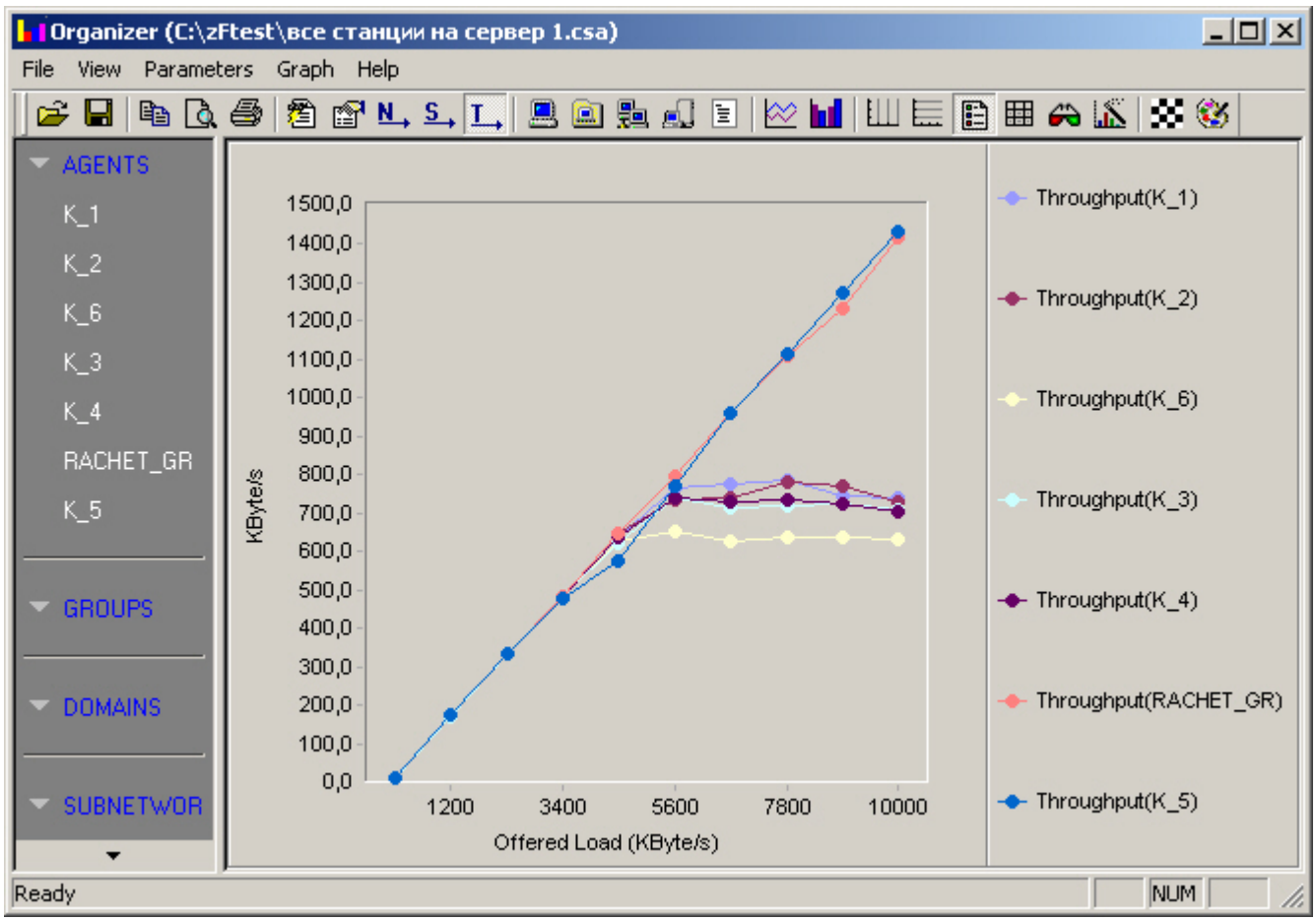

рис. 5.2 Результаты теста работы семи станций с сервером.

В процессе тестов всех типов измеряются следующие параметры:

- Read Rate (Скорость чтения) – характеризует среднюю скорость Агента при выполнении операций чтения на каждом этапе теста. Вычисляется как размер записи (параметр “Size of record”/“Размер записи”), деленный на время выполнения операции чтения. Полученные значения усредняются за время шага (параметр “Duration of one step”/“Длительность шага”). Измеряется в Кбайт/сек.
- Write Rate (Скорость записи) – характеризует среднюю скорость Агента при выполнении операций записи на каждом шаге теста. Вычисляется как размер записи (параметр “Size of record”/“Размер записи”), деленный на время выполнения операции записи. Полученные значения усредняются за время шага. Измеряется в Кбайт/сек.
- Read Throughput (Пропускная способность при выполнении операций чтения) – характеризует пропускную способность, с которой каждый Агент реально выполнял операции чтения на каждом шаге теста. Вычисляется как общий объем прочитанных с сервера данных, деленный на время шага (параметр “Duration of one step” / “Длительность шага”). Измеряется в Кбайт/сек.
- Write Throughput (Пропускная способность при выполнении операций записи) – характеризует пропускную способность, с которой каждый Агент выполнял операции записи на каждом шаге теста. Вычисляется как общий объем записанных на сервер данных, деленный на время шага (параметр “Duration of one step” / “Длительность шага”). Измеряется в Кбайт/сек.

– Throughput (Пропускная способность) - Throughput = Read Throughput + Write Throughput

Для большей информативности на рис. 5.2 отображена только пропускная способность.

5.2.3 Тест с пошаговым добавлением агентов

В отличие от совместной работы всех агентов, тест с пошаговым добавлением больше предназначен для диагностики, т.к. в результате теста мы не только видим суммарную нагрузочную способность ИС, но и влияние от каждой рабочей станции на ИС. Как было ранее показано, очень часто одна сбойная станция приводит к сбою работы всей ИС. Данный тест наилучшим образом подходит для выявления станций, создающих проблемы в сети.

При проведении полной диагностики ИС необходимо проводить диагностику всех уровней, начиная с нижнего, однако бывает целесообразно вначале проверить наличие ошибок в сегменте или протестировать сразу все станции и сервера с помощью FTest. К тому же при стрессовом тестировании мы сразу увидим проблемные станции, на которые нужно обратить внимание.

Калибровочный тест является дополнением теста с добавлением агентов. Проводится как для предварительного выявления возможностей станций, так и как самостоятельный тест для выявления проблемных станций.

В качестве параметров задаются время работы агента и нагрузка, создаваемая им.

На рис. 5.3 показан пример калибровочного теста.

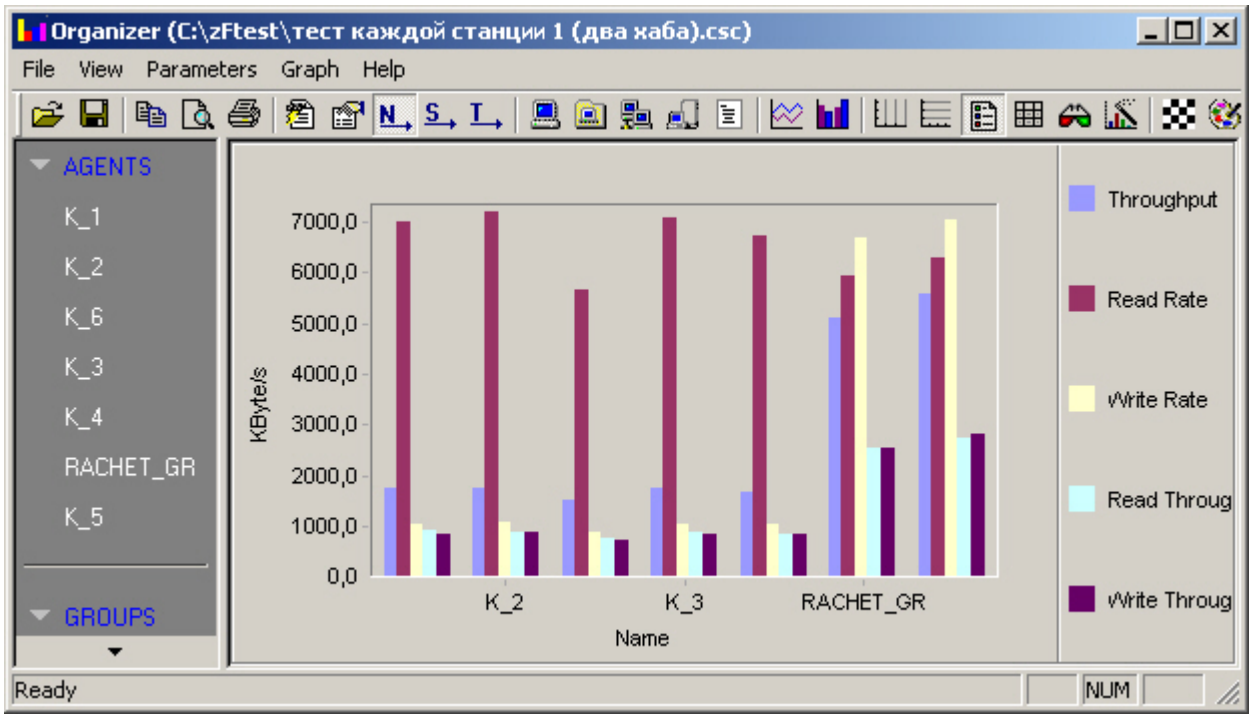

рис. 5.3 Результат калибровочного теста.

5.3 Диагностика потери данных

Данные при обработке в сетевом оборудовании могут пропадать в результате перегрузки или сбоя. При передаче данных с установлением соединения мы можем зафиксировать повтор транспортным или более высоким уровнем передачи пакета. В датаграммном режиме, естественно, никакого повтора не будет и отследить потери пакетов можно только с помощью статистики принимающего приложения.

Сам по себе процесс фиксирования повторов передачи прост, но практически все анализаторы выполняют его только при анализе уже захваченных пакетов. Видимо, Observer единственный, по крайней мере из программных анализаторов, проводит анализ в режиме реального времени, что является огромным преимуществом.

Экспертная система Observer (приведена на рис. 5.4) понимает потоки данных TCP, IPX, NetBIOS, UDP. Кроме определения повторов (Retransmission), эта система рассчитывает задержки передачи, фиксирует ответы типа Busy для IPX и нулевые окна для TCP.

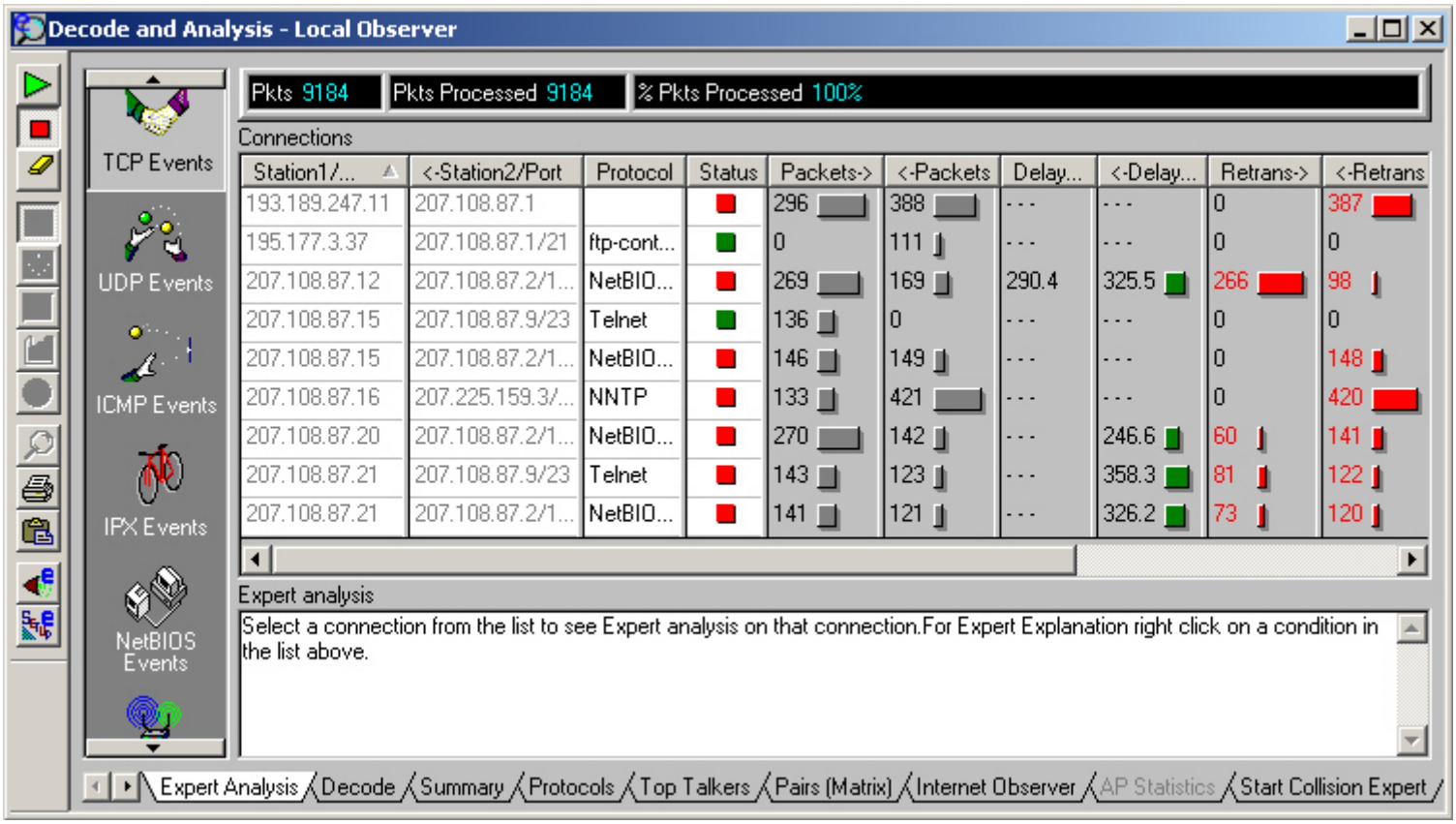

рис. 5.4 Экспертная система Observer.

5.4 Определение скорости передачи данных

Замедление передачи данных на уровне драйверов и сервисов ОС компьютера легче всего диагностировать с помощью FTest, тем более, что результаты сразу можно сравнить с аналогичными компьютерами. Так, на рис. 5.2 видно, что станции RACHET_GR и K_5 обладают гораздо большей пропускной способностью по сравнению с остальными станциями. Объясняется это очень просто и может служить прекрасным примером диагностики уровня сервиса. RACHET_GR и K_5 работают под управлением Windows 98, остальные машины работают под Windows 2000 и выполняют операции записи в транзакционном режиме. Т.е., записал – дождался подтверждения. Машины под Windows 98 производят запись и не ждут подтверждения, соответственно, скорость записи у них гораздо выше, что отражается на общем уровне пропускной способности (приведено на рис. 5.3).

Замедление скорости передачи при прохождении потока через сетевое устройство в основном вызывается перегрузкой устройства, и в конечном итоге вызовет потерю пакетов, что может быть зафиксировано экспертной системой Observer или с помощью SNMP статистики устройства. В других случаях измерить

скорость потока в сети можно с помощью функции Observer Pair Statistics (рис. 5.5). Заодно рассчитываются задержки передачи от каждой из сторон. Измерять скорость потока имеет смысл, только если известно, какой она должна быть или во время настройки приложений. В крайнем случае, можно провести измерения, меняя активное оборудование и настройки и сравнивая полученные данные. Результаты измерений наиболее используемого ПО на различном оборудовании и другие примеры по диагностике ИС можно посмотреть в базе знаний, созданной компанией Пролан на сервере http://kb.netconsulting.ru/index.asp.

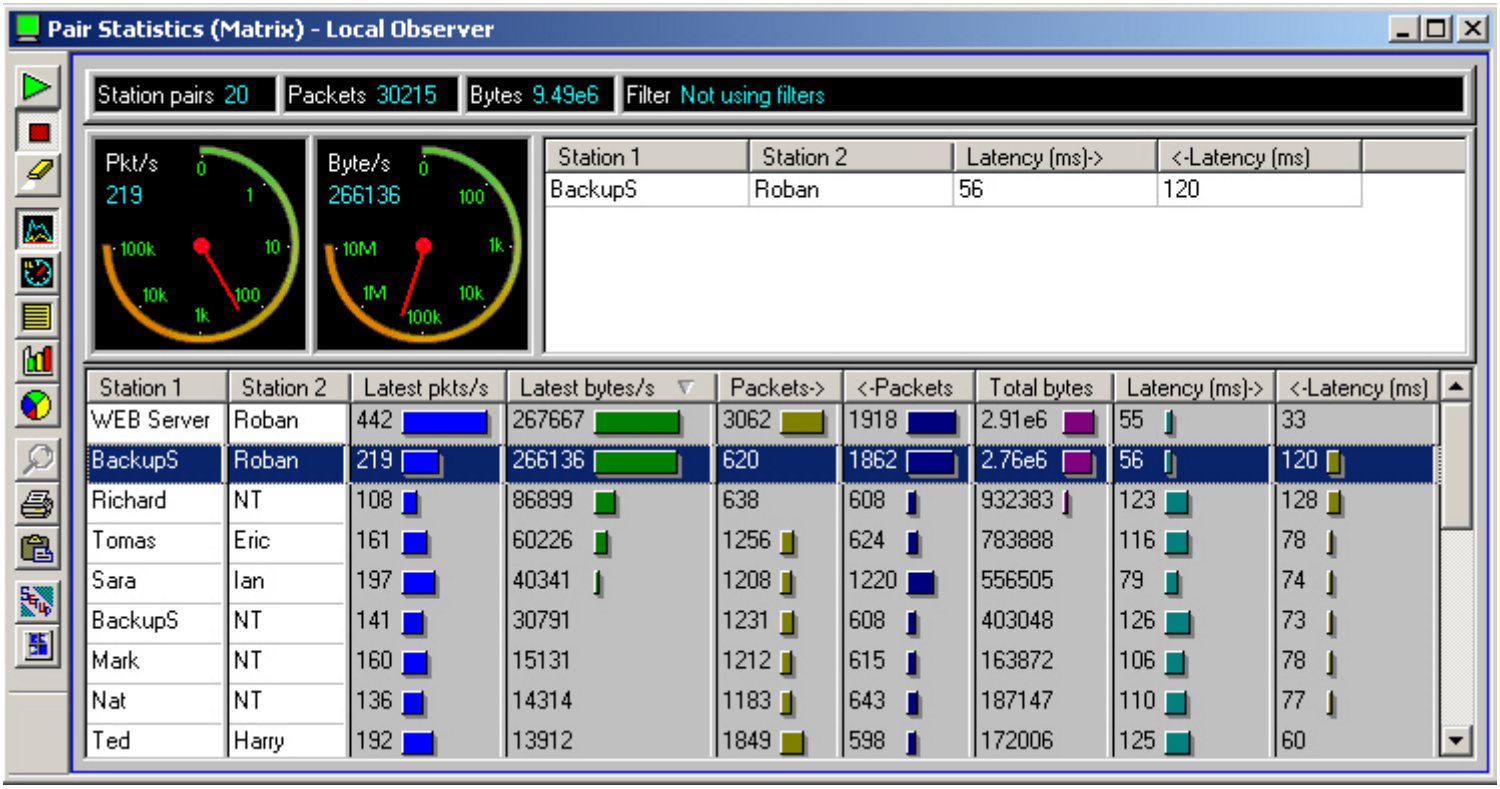

рис. 5.5 Статистика потоков

5.5 Обработка и передача информации сетевыми устройствами

С одной стороны разбираться с проблемами обработки данных сетевыми устройствами легко, а с другой – тяжело. Результат неверной обработки, скорее всего, проявится в сбое, отказе обслуживания, т.е. симптомы будут достаточно отчетливые. Однако для выяснения причин необходимо декодировать данные обмена и разобраться, почему сетевой сервис обработал их неверно. Для этого надо полностью понимать принципы функционирования сервиса.

Декодирование – основная функция любого сетевого анализатора. Различия между анализаторами могут состоять только в следующем: выполняется ли декодирование в режиме реального времени и какое количество протоколов декодируется.

Мощным помощником может служить экспертная система WinPharaoh. В режиме реального времени она отслеживает системные сообщения сетевых устройств, например, сообщение об отказе в доступе к файлу на сервере.

Сервисы и драйверы рабочих станций, серверов, сетевого оборудования по сути являются программами, т.е. неверная обработка вызывается ошибками, неверными настройками или отклонениями от стандартов в этом ПО.

5.6 Примеры

Пример 1. *Диагностика работоспособности ноутбука.*

Проблемы в сети:

После замены в ноутбуке беспроводного адаптера (2 Мбит/с) на сетевой адаптер (100 Мбит/с) и включения в коммутатор скорость работы в сети снизилась.

Методика и средства:

Диагностика проводилась с помощью экспертной системы и RMON модуля анализатора Observer.

Результаты:

На порту коммутатора, к которому был подключен ноутбук, ошибки не зафиксированы. Анализ работы ноутбука в сети показал повторы передач.

Вывод:

Т.к. по сети нет ошибок передачи данных, остается только одно предположение – данные теряются на уровне сервиса ОС ноутбука еще до передачи в сеть. Внимательное исследование настроек сетевой карты показало, что она была настроена на одно прерывание с другим устройством ноутбука, а именно это и вызывало периодическое уничтожение данных передаваемых сетевой карте от уровня приложения.

Примечание:

Стоит заметить, что в отличие от коллизии, повтором потерянных пакетов в этом случае занимался транспортный уровень, что привело к резкому падению работоспособности.

Пример 2. *Восстановление связи рабочей станции с Интернетом.*

Проблемы в сети:

Рабочая станция перестала получать доступ в Интернет. Никаких других проблем работы в сети не было.

Методика и средства:

Диагностика проводилась с помощью Observer.

Результаты:

На фазе получения ip-адреса Internet Explorer посылал запросы сразу на два, прописанных в настройках станции, DNS-сервера. Один из них являлся тестовым, внутрисетевым. Тестовый сервер тут же отвечал, что он не исправен. После этого Internet Explorer, не дожидаясь ответа от второго, сообщал, что не может установить ip-адрес.

Вывод:

В данном случае мы имеем дело с явной неверной обработкой данных. После удаления из настроек тестового сервера связь с Интернетом восстановилась.

6 Диагностика ПО

6.1 Вопросы диагностики ПО

После проведения полной диагностики ИС до уровня приложения и устранения всех проблем остается выяснить только следующее: как приложение должно работать, как оно работает и от чего зависит его работа. Проблемы со сбоями работы приложения рассматривать не будем, т.к. при отсутствии неполадок на более низких уровнях ясно, что они вызваны внутренними ошибками, а это уже полностью относится к компетенции программистов.

Как приложение должно работать, можно сказать только очень условно, т.к. работоспособность ПО зависит от работоспособности всех элементов ИС. Можно только измерить работоспособность на заведомо исправном тестовом стенде, соответствующем рекомендуемым требованиям для данного приложения, или во время сдачи-приемки. После этого можно контролировать работоспособность приложения и фиксировать замедление или ускорение его работы. Для измерений достаточно секундомера, чтобы зафиксировать время выполнения определенных операций.

Диагностика работы приложения является наиболее сложной задачей, поэтому в ней должны участвовать как минимум три специалиста с высоким уровнем квалификации: специалисты по сетям, серверам и программист (разработчик ПО). Работы являются очень трудоемкими, т.к. необходимо собрать, проанализировать и сделать выводы по большому количеству данных о функционировании ИС.

6.2 Средства и методика определения зависимости работы приложения от других элементов ИС

Для того, чтобы понять, от каких подсистем элементов ИС и на сколько зависит работоспособность приложения, нужно собрать данные о работе всех подсистем параллельно с протоколированием работы приложения. Причем для однозначных, не случайных результатов данные должны собираться достаточно продолжительное время (как минимум несколько дней), Но сразу возникает следующий вопрос – что делать с этой массой данных?

Элементы ИС состоят из множества подсистем, например дисковая подсистема в сервере и процессор в маршрутизаторе. Только в одном сервере Windows 2000 с помощью административной утилиты Performance Logs and Alerts можно протоколировать несколько сотен параметров работоспособности подсистем.

Для проведения анализа данных о работоспособности ИС предназначена программа Trend Analyst, компании Пролан (http://www.prolan.ru/solutions/diagnostics/trendanalyst/index.html).

Эффективность программы Trend Analyst основана на том, что с ее помощью можно привязать к единой временной шкале, наложить друг на друга и совместно обработать характеристики работы различных подсистем сети. Это могут быть характеристики работы каналов связи, серверов, пользовательских приложений и т.п.

Особенность программы Trend Analyst заключается в том, что с ее помощью можно отображать и аналитически обрабатывать данные, которые измеряются различными средствами. Например, программой NPM Probe, входящей в состав пакета NPM Analyst и такими программами как Open View NNM компании Hewlett Packard, Observer Suite компании Network Instruments, CiscoWorks2000 компании Cisco, Spectrum компании Aprisma, и многими другими.

Данными о работе приложения может являться время отклика на какую-нибудь операцию, например, бухгалтер нажимает “ввод” и получает через определенное время на экране выборку. Встроенная в программу функция может протоколировать данные о времени отклика.

Эмуляция работы приложения с файловым или SQL-сервером может выполняться агентами FTrend или NPM Probe.

Программа Trend Analyst позволяет не только объединить в единой базе данных разнородную информацию, но и провести ее вероятностный, корреляционный и регрессионный анализ.

Еще одним важным свойством Trend Analyst является возможность обработки данных, выраженных не только в процентном соотношении, но и в численном. Мы можем не знать и не узнать, сколько прерываний в секунду на сервере нормально для нашего приложения, а сколько много, но, увидев, что замедление работы приложения на 95% зависит от этого параметра – будем знать, где потенциальный источник проблем.

6.3 Функции анализа программы Trend Analyst

6.3.1 Вероятностный анализ.

Функция вероятностного анализа позволяет обрабатывать любые характеристики, которые содержатся в базе данных программы Trend Analyst .По каждой характеристике с ее помощью можно вычислить: минимальное значение за заданный интервал времени, максимальное значение, среднее значение, среднеквадратическое отклонение от среднего значения, а также строить выборочную плотность вероятности (гистограмму). Выборочная плотность вероятности позволяет определить, какие значения и с какой вероятностью принимала выбранная характеристика за заданный интервал времени.

Перед выполнением вероятностного анализа можно произвести предварительный отбор данных, по которым будут формироваться вероятностные оценки. При отборе задаются: дата начала, и дата окончания анализируемого интервала времени, дни недели и время суток, по которым должны выполняться обработка данных. Это позволяет исключить из рассмотрения периоды времени, когда сеть не эксплуатировались.

Пример вероятностного анализа одной из характеристик работы сети (в данном случае – скорости выполнения операций чтения; единица измерения – КБайт/с), приведен на рис. 6.1.

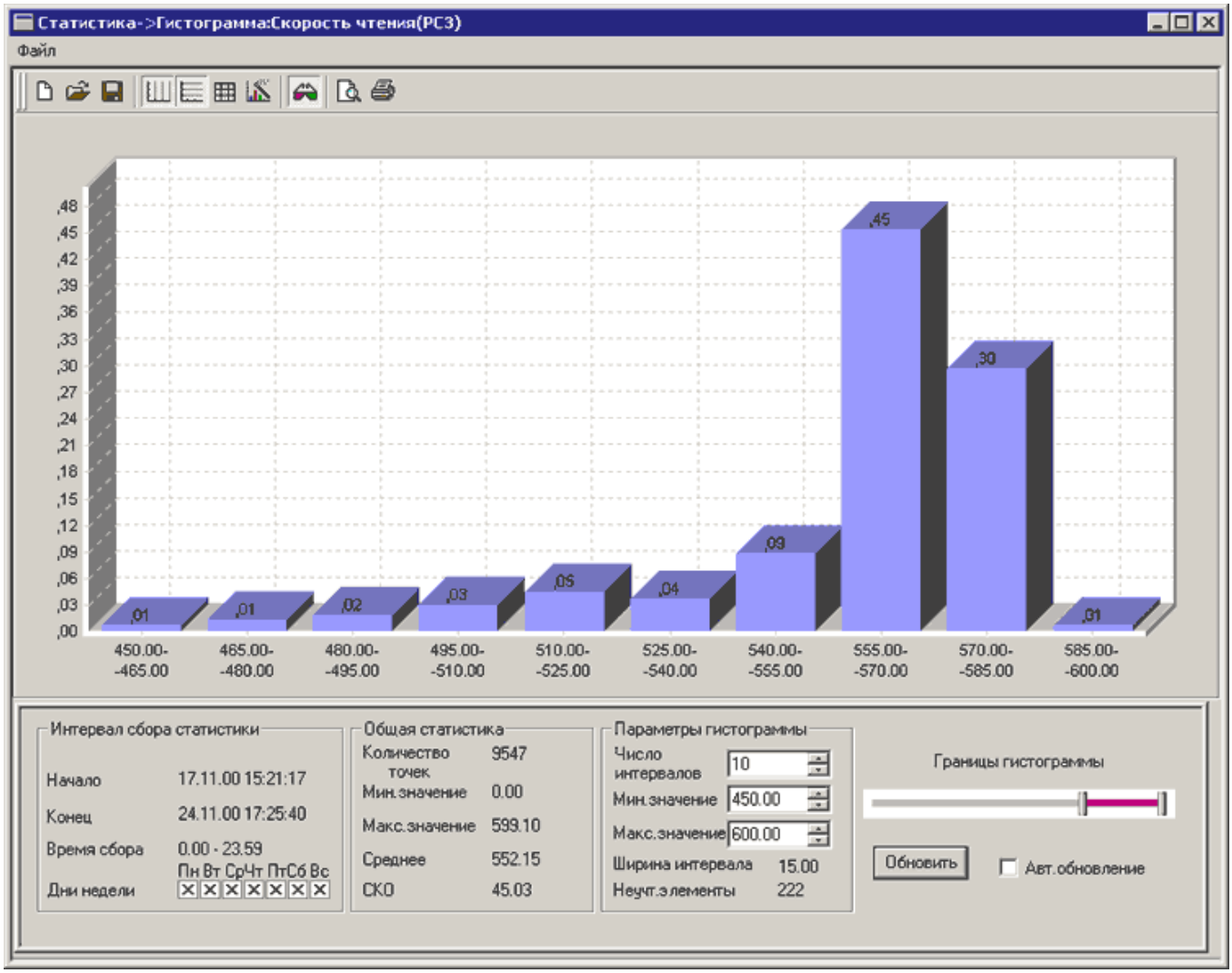

рис. 6.1 Пример вероятностного анализа.

6.3.2 Корреляционный анализ

Корреляционный анализ позволяет в процентном отношении показать, насколько зависит работа интересующей нас подсистемы от любой другой. Т.е. в результате анализа мы видим, что, например, время формирования отчета бухгалтерским приложением зависит на 95% от утилизации дисковой подсистемы сервера и на 40% от утилизации процессора рабочей станции.

Программа Trend Analyst позволяет вычислять два типа корреляционных отношений: парные и множественные. Парное корреляционное отношение – это отношение функции к одному аргументу. Множественное корреляционное отношение – это отношение некоторой функции к двум и более аргументам.

Множественный корреляционный анализ проводится в том случае, если предполагается, что интересующая нас подсистема может зависеть только от нескольких подсистем сразу. Например, замедление работы приложения может быть связано с перегрузкой коммутатора, т.е. функция скорости работы приложения будет сильно зависеть от одновременной загрузки всех портов коммутатора и слабо – от загрузки одного порта.

Результаты вычислений парных корреляционных отношений оформляются в виде таблицы. В каждой клетке таблицы выводится корреляционное отношение между характеристикой строки и столбца. Кроме этого, корреляционное отношение кодируется цветом. Наибольшему значению корреляционного отношения

соответствует красный цвет клетки, наименьшему – белый цвет. Примерная таблица с проведенным корреляционным анализом приведена на рис. 6.2.

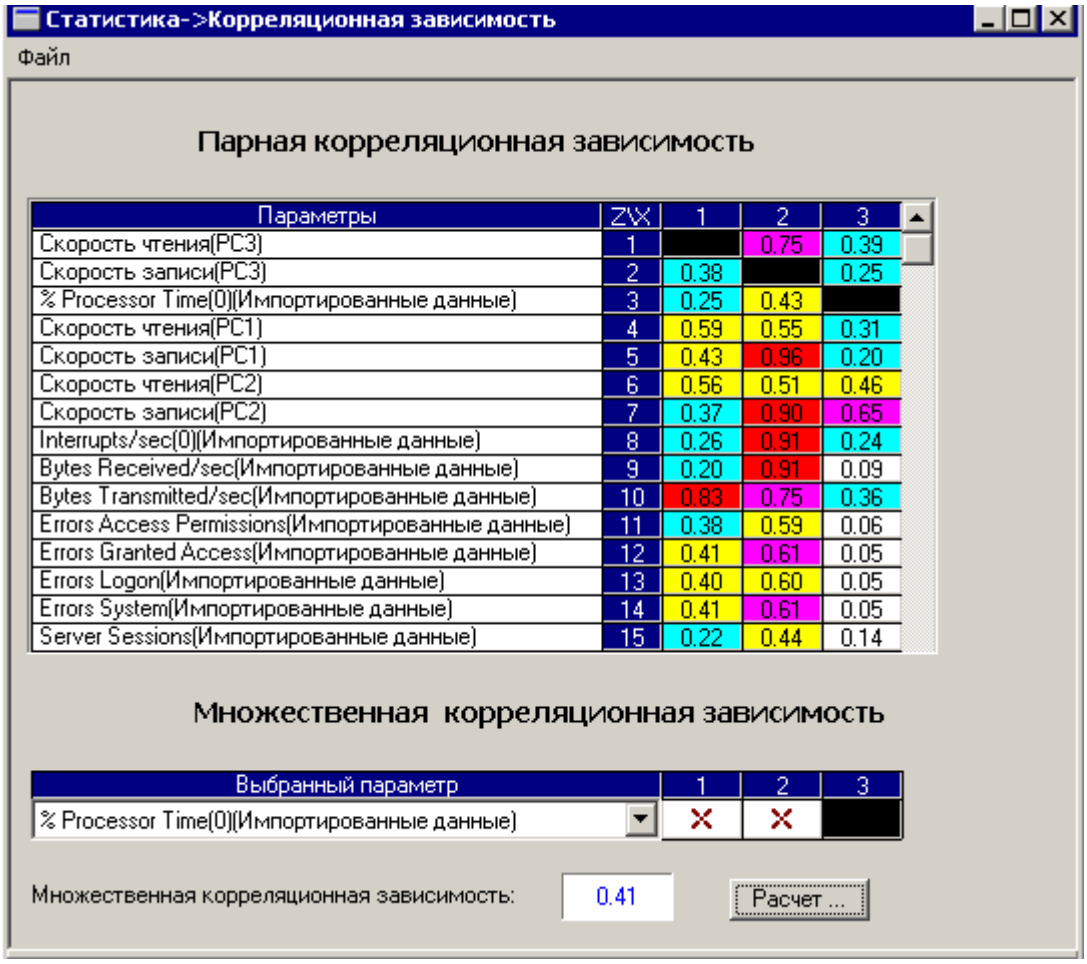

рис. 6.2 Корреляционная таблица

6.3.3 Регрессионный анализ

Если в процессе корреляционного анализа будет установлена высокая степень зависимости подсистемами, желательно установить, каков вид этой зависимости. Например, если в процессе корреляционного анализа было установлено, что скорость работы сети в наибольшей степени зависит от доли широковещательных пакетов, необходимо построить график этой зависимости. Такой график поможет правильно установить критичную долю широковещательных пакетов. Такими же характеристиками, влияющими на работу сети, могут быть: число ошибок передачи данных, утилизация процессора сервера, утилизация канала связи и т.п.

Важно отметить, что регрессионный анализ позволяет оценить пороговые значения характеристик работы подсистем, критичные именно для диагностируемой ИС. Эти пороговые значения, в дальнейшем, могут задаваться в средствах мониторинга для выдачи сигналов тревоги. Например, в пакетах Observer, MS Performance Monitor 2000, Novell ManageWise, HP Open View NNM и многих других. Таким образом, регрессионный анализ существенно увеличивает эффективность использования средств сетевого управления.

Пример работы функции регрессионного анализа пакета Trend Analyst приведен на рис. 6.3 (разрыв в графике обозначает отсутствие данных в точке разрыва).

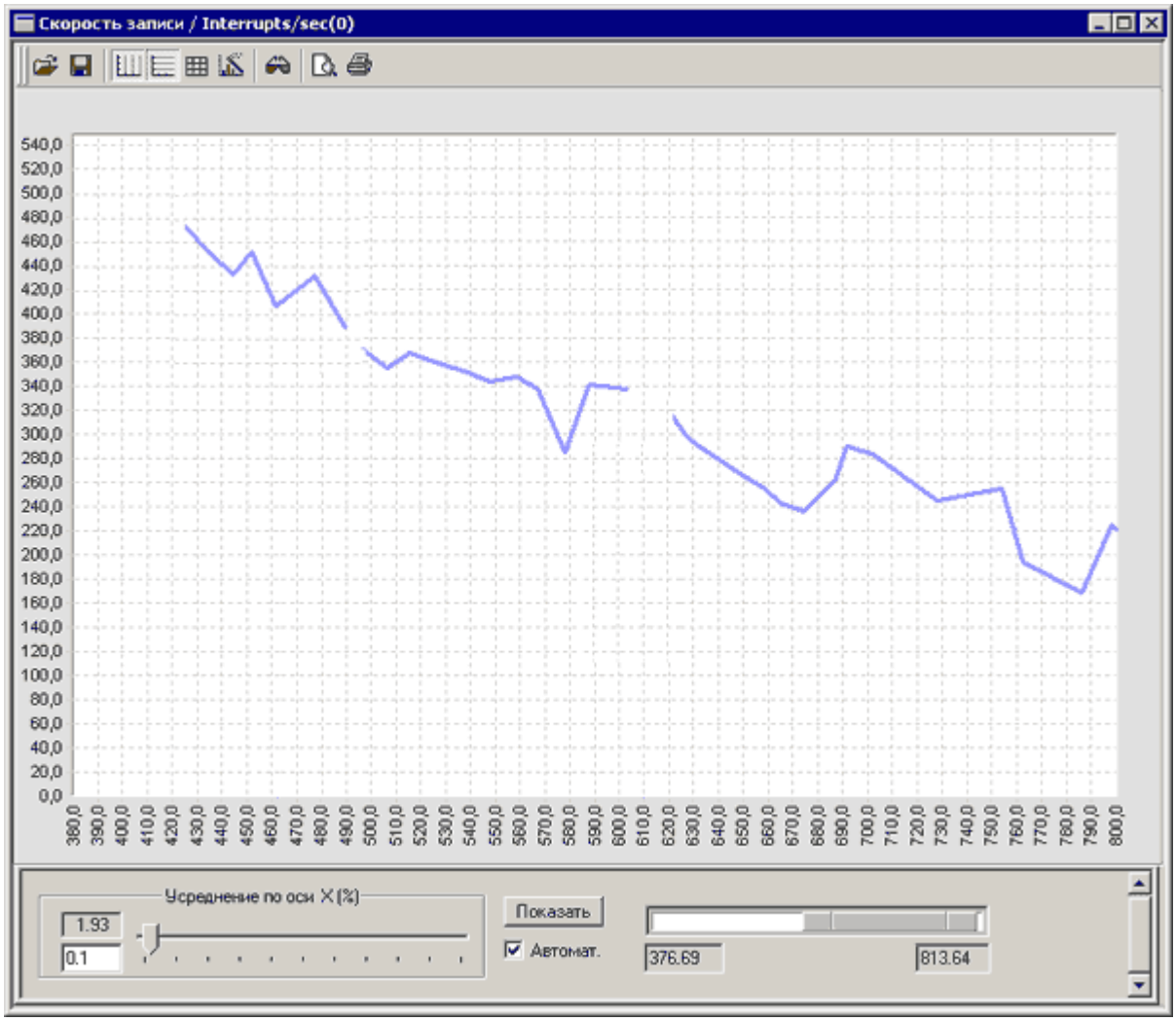

рис. 6.3 Пример работы функции регрессионного анализа.

6.4 Примеры

Пример 1. *Диагностика ИС. Пример взят из практики компании Пролан (http://www.prolan.ru/company/article/kb/2.html).*

Проблемы в сети:

Периодически замедляется скорость работы пользовательских приложений.

Методика и средства:

Для нахождения зависимости работы приложений от других подсистем ИС были запущены агенты пакета FTrend, параллельно протоколировались результаты работы других подсистем ИС. Полученные данные были проанализированы с помощью TrendAnalyst.

Результаты:

График скорости чтения и записи показал резкое дневное снижение скорости записи (рис. 6.4). Результат корреляционного анализа данных выявил, что скорость записи в наибольшей степени (коэффициент 0,99) зависит от параметра "% Disk Write Time" на сервере (рис. 6.5).

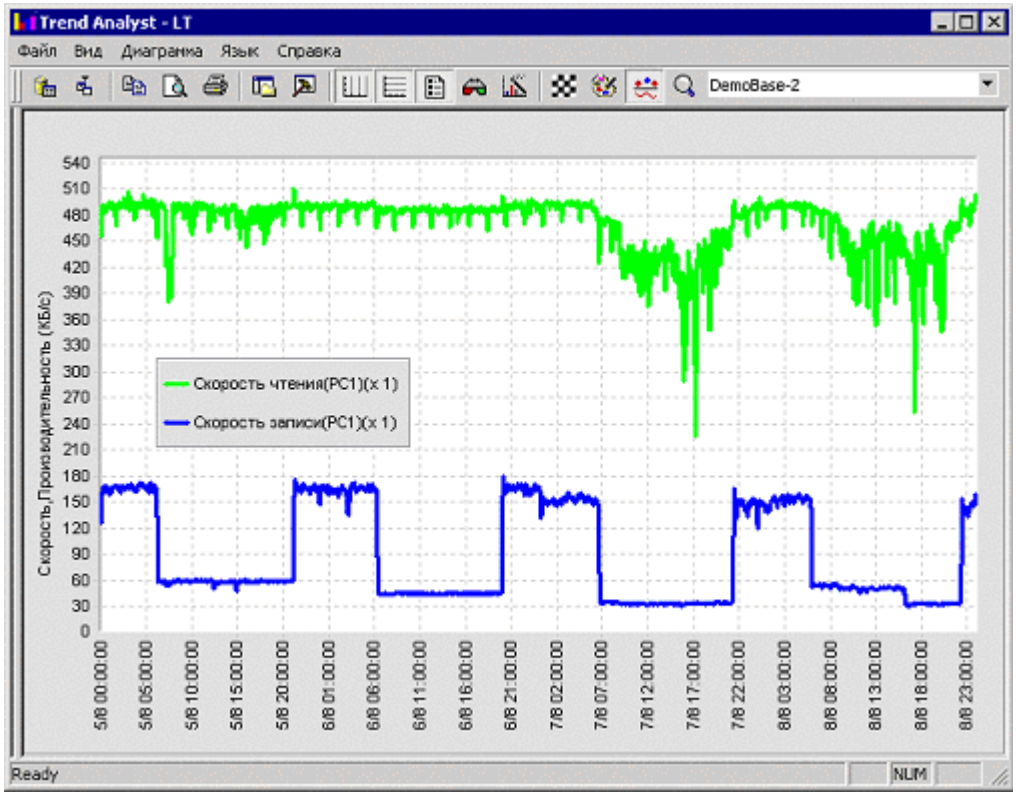

рис. 6.4 График скорости чтения и записи агента FTrend

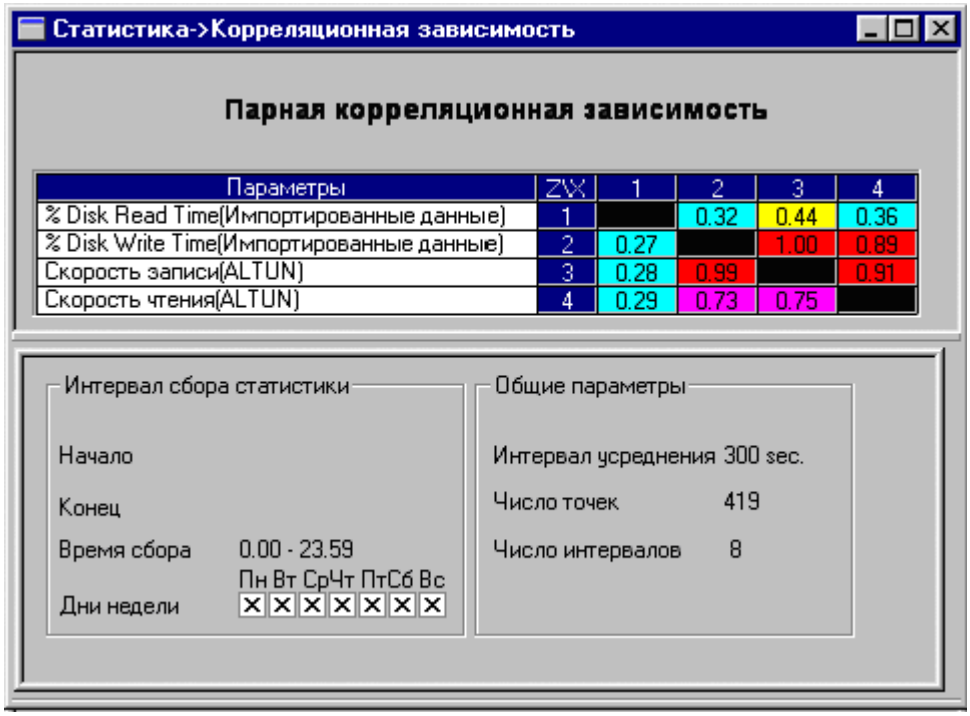

рис. 6.5 Фрагмент таблицы корреляционного анализа

Вывод:

Фактически результат корреляционного анализа и является выводом, т.е. замедление работы приложений вызывала низкая производительность дисковой подсистемы сервера, что и было подтверждено после ее замены.

Сергей Поповский **эксперт компании IBA IS, Минск, Беларусь**
popovsky@iba.com.by, z_7@mail.ru