Кафедра: «Автоматика и информационные технологии»

**Лабораторная работа**

**"ИССЛЕДОВАНИЕ КАНАЛЬНЫХ ПРОТОКОЛОВ С ОБРАТНОЙ СВЯЗЬЮ"**

**Введение**

Уровневая архитектура и принцип функциональной модульности – основа построения современных сложных технических систем. Организация системы с разбиением ее на иерархически подчиненные функциональные уровни, каждый из которых не обязан знать детали внутреннего устройства других уровней, представляя их себе в виде «черного ящика» с известными лишь входными и выходными сигналами, зарекомендовала себя в мировой практике как рациональная и облегчающая процесс проектирования, реализации и эксплуатации систем. Известно, что применительно к сетям передачи данных, этот принцип получил статус международного стандарта, краеугольным камнем которого является «Эталонная модель взаимодействия открытых систем» (ЭМ\_ВОС, OSI – Open Systems Interconnection) (стандарт ISO‑7498, 1978 г.; стандарт МККТТ (ITU-T) Х.200)

**1. Предмет и задачи исследования**

Предметом исследования в данной лабораторной работе являются протоколы 2-го, канального уровня OSI‑модели, которые оперирует *кадрами*. От верхних функциональных уровней стороны‑отправителя на канальный уровень поступает сообщение, представляющее собой «пакет данных» источника, который подвергается кадрированию. Стандартная структура кадра – это заголовок, собственно данные и концевик.[[1]](#footnote-1)

На приемной стороне канальный уровень разбирает кадр и передает на более высокий уровень только само сообщение[[2]](#footnote-2).

Задачей исследования является изучение протоколов как таковых, оценка их эффективности использования в каналах с различными техническими характеристиками, сопоставление оценок и формулирование условий рационального применения тех или иных версий канальных протоколов.

На эффективность влияет множество факторов. Чтобы их учесть, оценивая работу того или иного протокола в различных условиях реальной сети, нужно затратить много времени. Кроме того, нужно располагать инструментарием для проведения длительных экспериментов, не говоря уже об административном доступе к каналам для этих целей. Этот способ исследования наиболее затратный и трудоемкий. Помочь в решении этой задачи может только компьютерное моделирование.

**2. Два подхода к организации каналов в сетевых системах передачи**

Первая стратегия характерна для систем с так называемым *предварительным установлением соединения*, когда между двумя корреспондирующими устройствами устанавливается (и поддерживается в течение сеанса обмена данными) постоянный тракт (преобразований сигналов и его продвижения в направлении к получателю), а все кадры в одной сессии обмена проходят в сети по этому тракту по одному маршруту последовательно друг за другом. В этой же естественной последовательности они оказываются и у получателя. При этом стандартные протоколы канального уровня, как правило, используют помехоустойчивое кодирование для обнаружения ошибок в кадре. Естественно, что задача состоит не только в обнаружении ошибок, но и в выполнении необходимых действий для ее нейтрализации. После обнаружения ошибок на приемной стороне может быть, по крайней мере, два варианта дальнейших протокольных действий:

1). Либо как-то дать знать об этом передающей стороне, тем самым инициировав повторную передачу кадра (как принято говорить, «обработав ошибку» *на канальном уровне*);

2). Либо пропустить кадр с ошибкой на более высокий уровень, предварительно «пометив» его. В этом случае обязанность обработать ошибку должен принять на себя более высокий уровень системы.

При этом, из-за возможных повторных (быть может, многократных) передач искаженных кадров, время доставки кадра, вообще говоря, становится величиной случайной. Протоколы данного класса, когда приемная сторона сообщает передающей, что видит ошибки в поступившем кадре, называются протоколами *с решающей обратной связью* (решение о верности кадра возлагается на приемную сторону).

Вторая стратегия свойственна так называемому *дэйтаграммному* режиму работы системы передачи данных, когда предварительное соединение корреспондентов не устанавливается, а передатчик отправляет кадры в сеть, не сообразуясь с ситуацией у получателя (свободен, занят, работоспособен или нет) и в системе в целом. При этом разные дэйтаграммы[[3]](#footnote-3) могут *идти разными маршрутами*, с разной временной задержкой в сети. Прядок поступления дэйтаграмм к получателю становится случайным («раньше отправлена – позже поступила»). Собрать все дэйтаграммы в одно упорядоченное целое является задачей 4‑го, транспортного уровня стороны‑получателя. В таком режиме канальный уровень не отвечает за организацию обратной связи и вообще за обеспечение верности кадра[[4]](#footnote-4).

**3. Протоколы канального уровня**

На протокол канального уровня, обычно, возлагаются функции *перемещения пакета данных* в точку расположения приемника и обеспечения его верности.[[5]](#footnote-5) Протоколы канального уровня могут быть расклассифицированы следующим образом.

1). Протоколы многократной передачи, т.е. многократного повторения одного и того же кадра стороной‑отправителем с мажоритарной системой (по принципу голосования) принятия решения о наличии ошибок на стороне получателя. Дополнительно они могут быть разделены на протоколы с голосованием по поводу каждого бита или по поводу каждого кадра (кодового слова) в целом. Эти протоколы не требуют обязательного использования сигнальной избыточности в форме помехоустойчивого кодирования.

2). Протоколы с использованием однократной передачи кадра, закодированного помехоустойчивым кодом, с прямым исправлением ошибок средствами декодера приемного пункта.

3). Протоколы с обратной связью. Различают *решающую* и *информационную* обратную связь. В протоколах с решающей обратной связью (РОС) используются кодированные сигналы, позволяющие обнаруживать ошибки средствами декодера на приемной стороне. В случае обнаружения ошибок передающему модулю по обратному каналу передается запрос на повторную передачу кадра. Это важнейший протокольный акт в деле «продвижения кадра» в сторону получателя и обеспечения верности приема этого кадра. Протоколы с РОС называют также протоколами ARQ (Automatic Repeat Request – автоматический запрос на повторение).

4). Четвертый метод называется протоколом с информационной обратной связью (ИОС), или с «эхо-сигналом». Согласно этому методу, передающий модуль сохраняет копию передаваемого пакета в своем буфере. Принимающий модуль помещает полученный кадр в буфер приемника и транслирует его обратно отправителю, который сравнивает полученный им обратно кадр с кадром, хранящимся в буфере передачи. Если кадры совпадают, то считается, что пакет, содержащийся в буфере приемника, не содержит ошибок. Протоколы ИОС не обязательно должны использовать помехоустойчивое кодирование для обеспечения верности, так как ответственность за верность лежит на передающей стороне, которая располагает достоверным знанием о передаваемых данных.

Естественно, что применение метода многократной передачи не исключает в принципе попутного использования и механизма обратных связей, но такие версии протокола, обычно, малоэффективны и не находят сколько-нибудь широкого применения.

В обоих подходах (с обратными связями и без них) протокол канального уровня, стремясь обнаружить и (или) исправить ошибки, требует внесения, кроме сигнальной (битовой) избыточности (при помехоустойчивом кодировании), еще и протокольной избыточности[[6]](#footnote-6).

В данной лабораторной работе исследуются варианты протоколов *с обратной связью*. Они должны быть рассмотрены здесь более подробно.

3.1 Канальные протоколы ARQ (РОС)

При рассмотрении протоколов ARQ будем полагать, что пакеты данных передаются от станции-отправителя A к станции-получателю B. Между этими пунктами существуют два канала – *прямой* A→B и *обратный* A←B. Кадры данных (A→B) и технологические (A←B) защищаются избыточным циклическим кодированием (CRC). На обеих сторонах производится обнаружение ошибок в пакетах.

**3.1.1 ARQ с остановкой и ожиданием (SAW – Stay And Wait)**

Самая простая версия протокола с повторной передачей называется протоколом *с остановкой и ожиданием*. Основная идея протокола состоит в том, что передающий модуль, отправив кадр, останавливается и ожидает реакции приемной стороны. Передатчик не отправит следующий пакет данных (кадр) до тех пар, пока в той или иной форме не состоится подтверждение верности. Используется термин «извещение», или «квитанция» ACK (acknowledge), если ошибки не зафиксированы или NAK (negative acknowledge), если кадр принят с ошибками.

Этот протокол требует минимального объема буферной памяти: для одного пакета на стороне передатчика A и для пакета на стороне приемника B.

Существенную роль играет тайм-аут *Т*ож, точнее сказать, те протокольные действия, которые предпринимаются в случае отсутствия реакции приемной стороны в течение расчетного времени. Типичное решение заключается в том, что по истечении тайм-аута ситуация приравнивается к поступлению NAK. Такое решение приемлемо, если время доставки пакетов в канале (в сети) – величина постоянная.

Протокол может повести себя некорректно, если пакеты данных или сообщения ACK/NAK задерживаются в каналах *на случайное время*. Как сам кадр, так и ACK или NAK могут быть потеряны. Такая потеря может произойти в результате сбоя, нештатной ситуации или просто длительного ожидания в очереди на обслуживание в памяти какого-либо коммутационного узла в разветвленной сети с маршрутизаторами. Если по истечении тайм-аута ситуация приравнивается к поступлению NAK, то повторная передача пакета данных может привести к *дублированию пакета* *данных* на приемной стороне. В поток пакетов происходит «вставка пакета», т.е. своеобразное искажение передаваемых данных.



Эти явления – следствие *обезличенности* сообщений ACK и NAK.

**3.1.2 ARQ с возвратом на n шагов назад (GBn – Go Back n)**

Для предупреждения подобных ситуаций должна использоваться нумерация пакетов. Вместе с каждым кадром передается уникальный порядковый номер (*ПН*) содержащегося в нем пакета.[[7]](#footnote-7) Станция-получатель B *хранит номер последнего пакета,* поступившего без ошибок и аннулирует все вновь поступающие кадры, если ранее она их уже получила без ошибок, т.е. игнорирует дубликаты.

Протокол отправляет очередные кадры данных сразу вслед за первым, не дожидаясь, пока придет подтверждение в надежде, что «все обойдется», ведь вероятность получить кадр с ошибками, обычно, меньше, чем без ошибок. Станция-отправитель производит посылку *серии из* ***n*** *кадров,* считая от последнего пакета, получившего подтверждение верности. Все посланные, но не подтвержденные кадры должны оставаться в буфере передатчика с тем, чтобы быть извлеченными для цели повторной передачи, если придет отрицательное подтверждение (или быть удаленными, если подтверждение будет положительным).

Естественно, как и прежде, сторона‑получатель должна тем или иным способом подтвердить верность *всех пакетов* потока*.* По-прежнему, если подтверждение на какой-либо кадр отрицательно, кадр должен быть повторно передан.

Существует две стратегии повтора:

а) повторить группу кадров, начиная с того, на который не поступило сообщение ACK;

б) повторить выборочно только те кадры из последовательности длиной *n*, в которых декодером были обнаружены ошибки.

Протокол ARQ с возвратом на ***n*** шагов назад является примером протокола, в котором используется метод непрерывной передачи с обработкой ошибочной передачи кадра по варианту (а), «с возвратом».

Использование обратного канала исключительно для посылки подтверждений – это частный случай. Более общий случай заключается в том, что данные между пунктами А и В передаются (или могут передаваться) в обоих направлениях. Тогда по обратному каналу[[8]](#footnote-8) также идет и поток пакетов данных от станции B к A. В этом случае нумерованное подтверждение посылается в специально выделенном битовом поле заголовка пакетов данных обратного направления (а не в форме специализированного кадра).

Сообщение о верности пакета приобретает новый смысл: любой из корреспондирующих пунктов (А и В), отправляя пакет данных, в заголовке кадра указывает номер пакета (*НП*) и номер *ожидаемого пакета данных* от противоположной стороны (не очень удачно называемый в стандартах «номером запроса» – *НЗ*). Нумерация пакетов в сеансе должна начинаться с нуля. Номер запроса – это свидетельство для удаленной стороны, что все *пакеты с меньшими номерами* успешно достигли пункта назначения (ошибки в них не зафиксированы). Если передающая сторона к моменту получения *НЗ* не исчерпала регламент на передачу подряд ***n*** пакетов, то она должна перенести начало отсчета регламента на точку (*НЗ‑1*) в потоке отправляемых пакетов и тем самым снова приобрести возможность передавать без подтверждения еще ***n*** пакетов подряд. Если к моменту получения *НЗ* передающая сторона исчерпала регламент, то протокол должен будет для повторной передачи пакетов совершить «откат назад», но на величину не более ***n*** пакетов. Так возникает понятие «скользящего окна из ***n*** пакетов», а протоколы ARQ в такой реализации называют протоколами ARQ со *скользящим окном пакетов*.

Если интенсивность искажения пакетов мала, то число безостановочно передаваемых пакетов может быть весьма большим. Эффективность использования пропускной способности каналов с таким протоколом может быть весьма высокой.

Отправленные, но еще неподтвержденные пакеты, остаются в буферной памяти станции-отправителя. Протокол ARQ с возвратом на ***n*** шагов назад требует буферной памяти не менее, чем на ***n*** пакетов на стороне отправителя A и на один пакет на стороне получателя B.[[9]](#footnote-9)

Теперь отметим, что требуемый номер (как пакета, так и подтверждения) при работе с окном из ***n*** пакетов может принадлежать ограниченному множеству чисел {0, 1, 2… (*n*‑1)}. На языке математики это может быть охарактеризовано как нумерация по mod *m*, причем *m>n*. Минимальный допустимый модуль равен *n+*1.

**3.1.3ARQ с выборочным повтором (SR – Selective Repeat)**

Протокол ARQ *с выборочным повтором* также использует метод непрерывной передачи с окном на ***n*** кадров, но обработка ошибочной передачи происходит по стратегии (б) (см. выше).

Протокол ориентирован на дэйтаграммный режим, который не гарантирует поступления пакетов на приемный пункт в той же последовательности, в которой они отправлены передающим модулем. В этих условиях протокол с выборочным повтором избегает излишних повторных передач, но в этом протоколе не могут применяться нумерованные подтверждения в виде *НЗ*, т. к. номер запроса служит *групповым* подтверждением для пакетов с номерами, меньшими текущего *НЗ*. В протоколе же с выборочным повтором каждое подтверждение должно быть строго индивидуально сопоставлено своему пакету. Пакеты должны нумероваться по модулю, но модуль ***m***должен здесь выбираться, исходя из условия ***m***>2***n***.

Для работы протокола как на передающей, так и на приемной стороне необходима буферная память не менее, чем на ***n*** пакетов. Буфер на приемной стороне необходим, т. к. пакеты, которые поступают к ней в случайной последовательности, могут быть предъявлены получателю (как нечто цельное) не раньше, чем будут приняты безошибочными все пакеты из ***n*** (включая и повторно присланные) и все пакеты будут *упорядочены* в памяти приемного пункта. Такая сортировка пакетов необходима, т. к. в общем случае набор пакетов составляет единое целое, и предъявление данных получателю с переставленными пакетами недопустимо.

# 3.2 Канальные протоколы ИОС

**3.2.1 «Эхо» с ретрансляцией по обратному каналу всего кадра (Эхо1)**

Пусть при передаче пакетов из пункта А в пункт Bк структуре кадра не предъявлено никаких требований в отношении обнаружения или исправления ошибок (нет избыточных бит). При отправлении пакета данных в прямой канал копия пакета остается в буфере передатчика А. Принимающий модуль В помещает пакет в буфер приемника (без какого-либо анализа) и тут же транслирует его по обратному каналу в сторону отправителя А.[[10]](#footnote-10) Сторона‑отправитель A производит сравнение пакета, хранимого в буфере передачи, с пакетом, поступившим по обратному каналу. Если сопоставление не выявило расхождений, то считается, что в памяти приемного пункта В находится неискаженный пакет данных и можно передавать следующий пакет. Если же сопоставление выявило расхождение, то считается, что пакет в пункте B ошибочный, и передача такого пакета дублируется.

Протокол «ошибается» (пропускает ошибку) только тогда, когда оба кадра, прямой и обратный, искажаются ошибкой *одинаковой конфигурации*. Во всех других случаях ошибка будет обнаружена.

**3.2.2 «Эхо» с ретрансляцией контрольных бит CRC (Эхо2)**

Чаще используется другая разновидность этого протокола: оправляемые в прямой канал пакеты данных в передающем модуле кодируются помехоустойчивым циклическим кодом, контрольные биты CRC сохраняются в буфере, а в канал отправляются только биты данных. На приемной стороне пакет данных сохраняется в буфере и тоже кодируется. Сформированные биты CRC отправляются по обратному каналу передающему модулю А. Производится сравнение контрольных бит. На основании этого сравнения делается вывод о верности пакета в буфере приемника пункта B.

Простейшим вариантом работы модуля-отправителя является отправление пакета по прямому каналу и ожидание, пока по обратному каналу не придут его контрольные биты (аналогично алгоритму ARQ с остановкой и ожиданием). Однако может использоваться, так же, как в протоколах с РОС, и метод непрерывной передачи пакетов с сопутствующей нумерацией, механизмом «окна» и всеми вытекающими протокольными действиями.[[11]](#footnote-11)

Незамеченные ошибки могут возникнуть в двух случаях:

– во-первых, при таком искажении кадра, которому не сопутствует изменение контрольной суммы, т.е. когда содержимое кадра исказилось, но таким образом, что вновь вычисленные контрольные биты совпадают с контрольными битами, вычисленными для прежнего (неискаженного) кадра.[[12]](#footnote-12)

– во-вторых, когда кадр искажен в прямом канале, на приемной стороне получено свое значение CRC, которое при передаче по обратному каналу подверглось такому искажению, что совпало с CRC передатчика, отображающим отправленный (неискаженный) пакет.

# 4. Эффективность протоколов с обратной связью

Как следует из рис. 1, в отсутствие ошибок в каналах время передачи пакета данных под управлением рассматриваемого протокола равно длительности протокольного цикла *Т*пц. Как уже было отмечено, в этом времени цикла можно выделить долю «полезного», «чистого» времени, связанного с транспортировкой бит данных, и есть время, которое должно быть отнесено к «накладным расходам», обусловленным особенностями протокола.

Если ошибки в сигналах на физическом уровне не исключены, то передача одного пакета данных может потребовать двух протокольных циклов или трех или еще большего их числа. Следовательно, в общем случае в рассматриваемом протоколе время доставки пакета данных (без замеченных на приемной стороне ошибок) есть величина случайная.

Принимая это во внимание, эффективность протокола обычно, предлагается оценивать показателем, похожим на понятие КПД в технике, который отражает влияние различных факторов, действующих в системе передачи



Если имеется возможность наблюдать процесс передачи данных достаточно длительное время, численное значение *U* можно найти экспериментально:

, (1)



где *Tпередачи* – суммарное время, в течение которого станция‑отправитель передает в канал кадры данных.

*Tобщее* – общее время, затраченное на передачу, включает *Tпередачи* плюс время, затраченное на обработку, ожидание, повторные передачи кадров и прочие накладные расходы.

**4.1 Эффективность протокола ARQ-SAW**

Проследим влияние на эффективность различных параметров.

Протокольный цикл в данном случае включает (см. рис. 1)

*T*пц=2*T*з+*T*пк+*T*пACK+*T*ок+*T*оACK (2)

Здесь:

*T*пк – время передачи кадра данных в прямой канал (пропорционально длине кадра и тактовому интервалу);

*T*пACK – время передачи извещения ACK или NAK (пропорционально длине извещения и тактовому интервалу в обратном канале);

*T*з – задержка из-за конечного времени распространения сигнала в физической среде (считается, что она одинакова в прямом и обратном каналах);

*T*ок – интервал «обработки» кадра декодером прямого канала;

*T*оACK – интервал обработки извещения декодером обратного канала.

Из формулы и рис. 1 видно, что накладные расходы времени в пределах одного протокольного цикла могут быть весьма различными в зависимости от «игры параметров» в тех или иных конкретных условиях. Например, если с *малой скоростью* передаются *длинные кадры* данных, возможно, что исследователь сочтет возможным пренебречь величинами *T*пACK, *T*ок и *T*оACK. Приняв это допущение и временно предположив отсутствие ошибок в кадрах (никакой кадр не передается дважды), получим эффективность

. (3)



Если снять допущение о безошибочных кадрах (пусть вероятность ошибки в кадре равна *Pк*), то каждый кадр будет передаваться в среднем не 1 раз, а *N*п раз (из-за повторных передач), причем

*N*п=. (4)



Поэтому эффективность снизится в *N*п раз:

, (5)



Одним из факторов, влияющим в данном случае на величину *U* является интенсивность ошибок в опознавании бит на физическом уровне. С увеличением вероятности происхождения ошибки снижается эффективность протокола, т. к. происходит больше повторных передач.

Из приведенных формул видно, что эффективность *U* зависит также от отношения (*Tз)*/*Tпк*. Если (*Tз)*/*Tпк*<<1 (задержка невелика по сравнению с длиной кадра, выраженной в единицах времени), то эффективность скорее всего будет близка к единице.

Вышесказанное относится к простейшему протоколу с автоматическим запросом на повторение.

**4.2 Эффективность протоколов ARQ-GBn**

Более сложны рассуждения для протокола с возвратом на *n* шагов назад. Его эффективность будет зависеть еще и от параметра *n*.

Допустим, ошибок в кадрах не происходит. Пусть также окно достаточно широкое, а время задержки достаточно мало:

2*Tз*<*nTпк* (6)

Допустим, что первое ответное подтверждение успеет прийти до того, как будет передан последний *n*-й кадр окна. Окно будет непрерывно скользить, а передатчик ни разу не возвратится назад для повторной передачи, т. к. все кадры будут своевременно подтверждены. В этом случае, если пренебречь временем обработки кадров на сторонах канала, эффективность равна 100%.

Если по каким-либо причинам неравенство (6) не выполняется (окно мало, либо слишком велика задержка распространения, либо передается слишком много бит в единицу времени), то даже при отсутствии ошибок в кадрах придется производить повторную передачу вследствие того, что подтверждения опаздывают.

Наличие ошибок в кадрах прямого направления вызывает возвращение назад и повторную передачу до *n* добавочных кадров для каждого ошибочного кадра. Например, даже при выполнении неравенства (6) эффективность уже не будет 100%-ной. Если всего надо передать *N* кадров, а среди них искажаются в среднем *NPк* кадров, то

. (7)



Как было отмечено выше, источником неэффективности протокола с возвратом на *n* шагов назад также могут быть наличие ошибок в обратном канале и длинные кадры обратного направления. Иногда потери эффективности из-за этих причин могут быть уменьшены путем выбора большей ширины окна *n*.

**4.3 Эффективность протокола ARQ-SR**

Как говорилось ранее, протокол с выборочным повтором отличается от протокола с возвратом на *n* шагов назад стратегией обработки ошибок в кадрах. Поэтому при отсутствии ошибок его эффективность зависит от тех же факторов.

Более действенно работает протокол с выборочным повтором при наличии ошибок, т. к. в этом случае повторно передаются только искаженные пакеты. Теоретически, при выборочном повторе может быть достигнута предельно возможная эффективность в условиях присутствия ошибок, равная *U* = 1‑*Pк*.

**4.4 Эффективность протоколов ИОС**

Рассмотрим кратко вопрос об эффективности протоколов с информационной обратной связью. Напомним, что при протоколе с ретрансляцией всего пакета каждый пакет вынужден передаваться не менее, чем дважды: один раз по прямому каналу и один раз по обратному. Поэтому его эффективность, как правило, меньше 50%. Для протокола с ретрансляцией CRC значение *U* будет определяться используемым циклическим кодом.[[13]](#footnote-13) Вообще говоря, эффективность протоколов с ИОС будет тоже определяться стратегией работы передатчика: останавливается тот или нет после посылки первого кадра, ожидая эхо-сигнала, который здесь выступает как аналог подтверждения. Характер зависимости эффективности от этих факторов аналогичен протоколам с РОС.

Приведенные формулы для величины *U* не отражают всей полноты протокольных нюансов и призваны лишь пояснить понятие эффективности. Более тщательный подход к делу должен учитывать ненулевые времена обработки кадров и извещений, нужно принимать во внимание то, что кадры могут иметь переменную длину, брать в расчет наличие тайм-аутов, а также возможность появления ошибок пачками и др.

Альтернативой использованию формул является моделирование передачи данных под управлением тех или иных протоколов в интересующих нас условиях.

Добавим, что общая оценка того или иного протокола канального уровня не должна ограничиваться лишь оценкой его эффективности. Важное место при этом занимает также вопрос, насколько помехоустойчив метод кодирования / декодирования. Если высокоэффективный в смысле величины *U* протокол использует для кодирования пакетов код со слабой помехоустойчивостью, то много ошибок не будет замечено и получателю будут переданы недостоверные данные.

Имеет смысл говорить об обеспечении эффективности *U* при обеспечении помехоустойчивости не менее заданной.

# 5. Моделирующая программа OSI\_2

Лабораторная работа проводится при помощи компьютерной программы, моделирующей работу протоколов канального и только канального уровня. Она имитирует те процессы с обратной связью, которые происходят после того, когда на 2‑й уровень одного устройства сети (станции-отправителя) с вышестоящего уровня поступает пакет данных, предназначенный к передаче на 2‑й уровень другого устройства сети (станции-получателя). Модель «не видит» процессы получения доступа к среде распространения сигналов. Предполагается, что постоянное соединение станций-абонентов уже заранее установлено, что процедуры «разборки» сообщений на пакеты на передающей стороне и «сборки» их в сообщения на приемной не отнимают времени и не создают проблем. Не моделируются вставки и выпадения кадров.

Это все явления, не относящиеся к канальному уровню по классификации OSI‑модели систем.

Имитируются процессы, протекающие на канальном уровне с момента поступления пакета данных от вышестоящего уровня: кадрирование, кодирование и отправление кадров, воздействие ошибок, возникающих на физическом уровне, прием и декодирование с использованием циклического кода с заданным порождающим многочленом.

В качестве единицы модельного времени принят *ВТ* (BitTime), т.е. один тактовый интервал, в течение которого передается один бит в точке порта пользовательского компьютера (или другого источника данных).

В процессе имитации передачи происходит учет общего времени передачи (множества пакетов), а также сбор статистических данных, которые позволяют оценить и сравнить эффективность различных протоколов канального уровня в зависимости от тех или иных характеристик канала, пакета, характера и интенсивности ошибок, определенных исследователем.

**5.1 Функциональные модули модели**

В каталоге программы **OSI\_2** находится исполняемый файл osi\_2.exe и два подкаталога:

* IN
* RESULTS

В каталоге IN хранятся файлы с входными данными для работы программы, имеющие расширение \*.in. В каталоге RESULTS – текстовые файлы с результатами моделирования программы, имеющие расширение \*.txt.

Для работы программы требуется компьютер IBM PC c ОС Windows’9X, Windows NT 4.0, Windows 2000 или ХР.

## 

## 5.2 Функциональная схема пользовательского интерфейса

Функциональная схема интерфейса с пользователем приведена на рис. 2.

После запуска программы на экране появляется главное окно программы. В верхней части окна находится *Главное меню* программы. Под главным меню расположена *Панель управления*, которая для удобства работы с программой содержит кнопки быстрого вызова операций. Под Панелью управления расположена *Область просмотра* результатов моделирования. Внизу главного окна расположена *Панель состояния* модели (подсказок).

Главное меню содержит следующие пункты:

* *Файл*
* *Параметризация*
* *Моделирование*

Позиция *Файл* главного меню содержит следующие пункты:

* *Создать* – очистить область просмотра главного окна и параметры моделирования, приготовив их для ввода спецификаций входных данных.
* *Открыть* – открыть текстовый файл для просмотра в главном окне программы. Предназначен для просмотра сохраненных результатов предыдущих прогонов модели.



Рис. 2. Функциональная схема пользовательского интерфейса

* *Сохранить* – сохранить результаты моделирования в текстовом файле.
* *Сохранить как*… – сохранить результаты моделирования в файле с другим именем.
* *Печать* – вывести результаты моделирования (содержимое области просмотра) на принтер.
* *Выход* – выход из программы.

При выборе пункта *Параметризация* появляется диалоговое окно задания параметров модели.

Позиция *Моделирование* главного меню содержит два пункта:

* *Начать моделирование*;
* *Результаты моделирования*;

Выбор пункта *Начать моделирование*запускает процесс моделирования передачи данных. При этом появляется окно с индикатором хода (процента выполнения) процесса моделирования. При нажатии на кнопку *Отмена* система просит подтвердить прекращение процесса моделирования. При завершении или прекращении моделирования в главное окно программы выводятся результаты моделирования.

Выбор пункта *Результаты моделирования* приводит к выводу в главное окно программы результатов последнего моделирования (Это могут быть и результаты предшествующих прогонов модели, если загрузить файл результатов другого процесса).

Позиция «*?»* главного меню содержит пункт *О программе*, при выборе которой выводится информация о программе.

Под главным меню расположена *Панель управления*, которая для удобства работы с программой содержит следующие кнопки:

* *Создать* (Очистить от старого и создать)
* *Открыть…*
* *Сохранить*
* *Сохранить как…*
* *Параметризация модели*
* *Начать моделирование*

## 5.3 Параметризация модели

При выборе пункта *Параметризация* появляется диалоговое окно задания параметров модели. Это окно содержит две закладки: *Протокол* и *Канал*, которые предназначены для ввода параметров протокола и каналов (прямого и обратного).

**5.3.1 Параметризация протокола**

При выборе закладки *Протокол* появляется окно, показанное на рис. 2. В данном окне задаются следующие параметры протокола.

1). Тип моделируемого протокола:

* ARQ с остановкой и ожиданием;
* ARQ c окном на N пакетов;
* ARQ c выборочным переспросом;
* «Эхо» с ретрансляцией кадра;
* «Эхо» с ретрансляцией CRC;

2). Порождающий полином циклического кода:

CRC‑12, CRC‑16, CCITT‑16, CRC‑32.[[14]](#footnote-14)

Каждый кадр (как прямой, так и обратный) имеет битовое поле CRC с размером, равным степени порождающего полинома, и поле информационных бит. И в прямом, и в обратном кадрах информационная часть несет собственно данные, которые передаются (в обратном кадре информационные биты пустые). Порядковые номера кадров и подтверждений, хотя передаются вместе с кадром, не занимают места в его длине, т. к. находятся в заголовке.

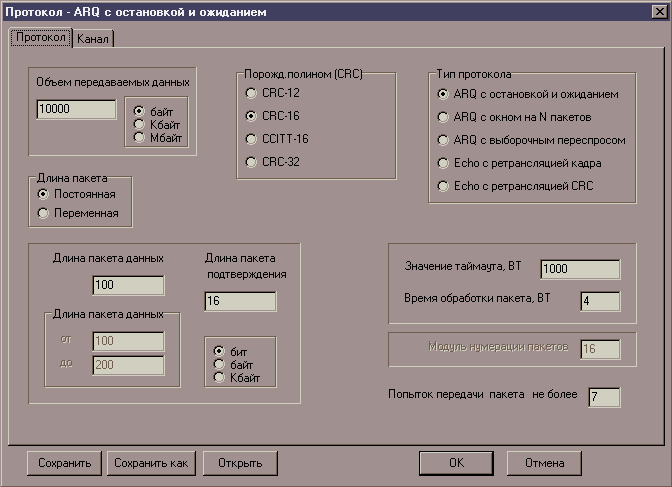


Рис. 2. Окно параметризации протокола

3). Тайм-аут на подтверждение пакета и время, затрачиваемое на обработку кадров (в том числе на кодирование и декодирование). Таймер начинает отсчет с момента окончания передачи кадра. Одна и та же величина тайм-аута действует как для станции – отправителя, так и получателя.

Т.к. модель не учитывает возможную потерю кадров в сети, то механизм тайм-аута отсутствует в протоколах без окна Значение тайм-аута в этом случае может быть любое, даже нулевое. В протоколах с окном тайм-аут может иметь (по умолчанию) нулевое значение, что допустимо, но нежелательно.

Время в модели измеряется в BT (bit-time). Один ВТ соответствует времени передачи одного бита в прямом канале. BT при необходимости может быть выражена в секундах, если названа пропускная способность (скорость) канала (бит/с). Тайм-аут задается *от момента окончания* передачи пакета.

4). Задается допустимое количество попыток передачи одного пакета. При превышении этого числа моделирование прекращается. Если это число задается равным нулю, то учет количества попыток передачи не производится.

5). Для протоколов ARQ с окном на N пакетов и ARQ с выборочным переспросом необходимо задать значение модуля нумерации пакетов. В зависимости от модуля нумерации и типа протокола модель вычисляет «ширину окна». Выбор модуля нумерации следует связывать со скорости передачи данных и задержки распространения сигнала в линии. Для протоколов ARQ с остановкой и ожиданием и протоколов с эхо-сигналом модуль нумерации пакетов принимается равным двум.

6). Длины кадров отдельно в прямом и обратном каналах. В этом же окне задается объем передаваемых данных (длина файла, который рассматривается как пользовательское сообщение).

При задании длин кадров и объема передаваемых данных имеется возможность выбора единицы измерения. Длина кадров прямого направления может быть постоянной или переменной. Можно выбрать кадры данных постоянной или переменной длины. При задании постоянной длины кадра указывается непосредственно эта длина. При задании переменной длины кадра указывается максимальная и минимальная длина.[[15]](#footnote-15) В последнем случае при моделировании генерируются кадры с длиной, равномерно распределенной в интервале от минимальной заданной до максимальной. Длина кадров в обратном направлении может быть только постоянной.

ВНИМАНИЕ: длины кадров прямого и обратного потока определяются по разным правилам. В поле с названием «Длина пакета данных» диалогового окна нужно ввести *полную длину кадра* прямого направления, включая контрольные биты. В поле «Длина пакета подтверждения» ожидается ввод длины *только информационной части* кадра подтверждения, не считая контрольных бит. Например, если в первом поле введено 32, а во втором 2 и используется код CRC‑16, то прямые кадры будут иметь общую длину 32 бита, из которых 16 контрольные, а обратные кадры будут иметь длину 18 бит, из которых 16 контрольные, а 2 информационные.

Для протоколов с эхо-сигналом поле длины обратного кадра не играет роли, т.к. длины обратных кадров определяются длинами прямых.

**5.3.2 Параметризация каналов**

При выборе закладки *Канал* появляется окно задания параметров прямого и обратного каналов. Для каждого канала можно задавать следующие параметры:

1). Скорость передачи (в бит/c и кратных величинах).[[16]](#footnote-16) Скорость обратного канала не должна быть больше, чем скорость прямого. Если она меньше, то в целое число раз.

2). Задержка распространения сигнала в канале (и, следовательно, неявно заданная длина);

3). Характер ошибок: независимые или ошибки типа «пачка».

При моделировании работы каналов с независимыми ошибками[[17]](#footnote-17) задается *р*б – вероятность ошибки в принятом бите на физическом уровне. При моделировании работы каналов с группированием ошибок задается вероятность появления пачки ошибок *р*пач, а также математическое ожидание и дисперсия длины пачки (длина пачки – случайная величина с нормальным распределением).

**5.4 Моделирование**

Меню *Моделирование* содержит два пункта:

* *Начать моделирование* – запуск процесса моделирования;
* *Результаты моделирования* – отображение результатов в области просмотра.

## Результаты моделирования и их интерпретация

После окончания моделирования на экран выводится (для справки) сводка входных параметров данного запуска модели, а затем как результаты отображаются следующие статистические данные, накопленные в процессе моделирования:

* Общее время передачи (в единицах BT и в секундах), засекается начиная с момента начала передачи первого пакета станцией-отправителем, вплоть до момента приема последнего пакета станцией-получателем;
* Размер переданного файла – общий объем данных, предназначавшихся к передаче и поступивших от вышестоящего уровня (в байтах). Каждый отправляемый в процессе передачи кадр прямого направления несет в своем информационном поле порцию бит из общего объема.
* Результирующие данные у стороны-отправителя:

«отправлено бит» – общее число отправленных в кадрах бит (в полном объеме, считая биты CRC);

«отправлено пакетов» – общее количество отправленных кадров, которые пришлось послать для передачи общего числа бит, включая повторные кадры;

«отправлено пакетов данных» – количество кадров, которые пришлось послать для передачи общего числа бит, не считая повторные кадры, т.е. число уникальных (не повторных, «полезных») отправленных кадров;

«получено пакетов с ошибками» – количество кадров, при передаче которых через линию в них возникли ошибки;

«обнаружено пакетов с ошибками» – количество кадров, в которых возникновение ошибок было обнаружено декодером приемной стороны;[[18]](#footnote-18)

* «суммарный вес ошибок» – общее число искаженных бит.
* Результирующие данные у стороны-получателя, которые интерпретируются аналогичным образом.

## Пример выходных данных моделирования. Размерность и обозначения величин

**Спецификации протокола**

*L*кд Длина кадра, бит 2000

*L*аск Длина кадра подтверждения, бит 16

*T*out Значение тайм-аута, ВТ 2400

*T*код Время на кодирование кадра, ВТ 0

Порождающий полином CRC‑16

*m* Модуль нумерации кадров 7

*B*m Ширина окна 6

**Прямой канал**

*C*пк Скорость передачи, бит/c 65536

*Т*зп,пк Запаздывание сигнала, с 0,00075

*р*б,пк Вероятность возникновения ошибки 0,001

*р*пач,пк Вероятность возникновения пачки ошибок

*L*пач,пк Мат. ожидание длины пачки, бит

*D*пач,пк Дисперсия длины пачки, бит

**Обратный канал**

*С*ок Скорость передачи, бит/c 65536

*Т*зп,ок Запаздывание сигнала, с 0,00075

*р*б,ок Вероятность возникновения ошибки 0,001

*р*пач,ок Вероятность возникновения пачки ошибок

*L*пач,ок Мат. ожидание длины пачки, бит

*D*пач,ок Дисперсия длины пачки, бит

*Т*мдл Время моделирования, ВТ 34466114

*Т*прд,ф Время передачи, сек 525,90

*М*ф Размер переданного файла, байт 512000

**Отправитель (А)**

Отправлено

*N*б,пк Бит 34466000

*N*к,пк Кадров 17233

*N*кд,пк Кадров данных 2065

Получено

*N*к,ош,окКадров c ошибками 288

*N*к,оош,окОбнаружено кадров c ошибками 288

*Е*w,ок Суммарный вес ошибок 290

**Получатель (В)**

Отправлено

*N*б,ок Бит 275728

*N*к,ок Кадров 17233

*N*кд,ок Кадров данных 0

Получено

*N*к,ош,пкКадров c ошибками 14879

*N*к,оош,пкОбнаружено кадров c ошибками 14879

*Е*w,пк Суммарный вес ошибок 34539

Эти данные могут быть использованы, для вычисления показателя эффективности протокола в тех или иных условиях применения, а также другие показатели, требуемые конкретным заданием.

**6. Программа лабораторных работ**

Модель протокола может быть применена для исследования влияния различных факторов на эффективность протокола в различных условиях его использования. Индивидуальное задание должно быть получено от преподавателя. Оно может быть ориентировано на различные аспекты дела, например:

Задача 1. Сопоставление эффективности различных протоколов с обратной связью в тех или иных условиях (ограничения на длину кадров, запаздывание в каналах и проч.)

Задача 2. Исследование эффективности протоколов с обратными связями в зависимости от интенсивности ошибок в каналах.

Исследовать, как влияет наличие ошибок по отдельности в прямом и обратном каналах на общую эффективность протоколов с ОС.

Рекомендуется следующий план работы: для каждого протокола сделать по 3 расчета эффективности U: при отсутствии ошибок, при ненулевой ошибке в прямом канале и нулевой в обратном и наоборот. Все остальные параметры протокола и канала (пропускная способность, длина кадра прямого и обратного направлений и др.) надо поддерживать постоянными. Заполнить таблицу (см. табл. 1).

На основе экспериментальных данных сделать выводы, ответив на вопросы:

* Как и почему меняется эффективность, если вероятность ошибки в одном из каналов становится ненулевой?
* Ошибки в каком канале (прямом или обратном) больше влияют на эффективность и почему так происходит?

Таблица 1. Эффективность протоколов в зависимости от интенсивности помех

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *P*б,пк | *P*б,ок | Эффективность протоколов | | | | |
| 1 | 2 | 3 | 4 | 5 |
| 0 | 0 |  |  |  |  |  |
| 0,001 | 0 |  |  |  |  |  |
| 0 | 0,001 |  |  |  |  |  |

* Сравните степень влияния ошибки на эффективность при разных протоколах и объясните результат.

Задача 3. Эффективность протокола в системе с идеальными (безошибочными) каналами

При работе протоколов без окна в условиях отсутствия ошибок исследовать, как влияют на эффективность следующие параметры: пропускная способность (бит/с), длина кадров, задержка распространения. Считать все эти величины для прямого и обратного каналов одинаковыми.

Рекомендуется следующий план работы: рассчитывать эффективность, изменяя один из трех вышеуказанных параметров, оставляя постоянными все остальные. Результаты свести в таблицу.

Таблица 2. Эффективность протоколов в каналах без ошибок

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| vпр= vобр | lпр= lобр, бит | Tз(пр)=Tз(обр), мс | 1 | 2 | 3 | 4 |
| 1 Кбит/с | 32 | 0,001 |  |  |  |  |
| 1 |  |  |  |  |
| 500 | 0,001 |  |  |  |  |
| 1 |  |  |  |  |
| 1Мбит/с | 32 | 0,001 |  |  |  |  |
| 1 |  |  |  |  |
| 500 | 0,001 |  |  |  |  |
| 1 |  |  |  |  |

**7. Порядок выполнения лабораторных работ. Отчет**

Лабораторная работа выполняется в соответствии с заданием, указанным преподавателем. Прогон модели и получение отчета с выходными данными – операция простая. Но из опыта известно, что рационально предварительно спланировать последовательность своих действий, особенно в отношении назначении значений тех или иных параметров, последующих вычислений, построения кривых.

Содержание отчета – это мотивированные формулировки и ответы на вопросы в соответствии с версией конкретного задания, с необходимыми иллюстрациями в форме цифрового материала, кривых, графиков Общий подход к построению отчета можно сформулировать следующим образом: «*иллюстрации к выводам и заключениям, а не наоборот*».

Известно из опыта, что для оформления отчета рационально привлекать возможности пакета EXCEL в части вычислений по наперед известным формулам, а также в части построения графиков по табличным данным.

Не следует приводить в отчете обширных извлечений из методического руководства. На него достаточно ссылаться, сохраняя обозначения величин.

Отчет должен быть сброшюрован, оформлен в соответствии со стандартом и представлен к очередной лабораторной работе для защиты.

**7.1 Завершение лабораторного сеанса**

По завершении лабораторной работы необходимо предъявить преподавателю полученные данные. Кроме того, необходимо *привести рабочие папки модели в исходный вид, т.е. удалить из папок IN и RESULTS все файлы, полученные в процессе работы,* и также предъявить их преподавателю. Папки IN и RESULTS должны оставаться свободными.

**Библиографический список**

1). Бертсекас Д., Галлагер Р. Сети передачи данных: Пер. с англ. – М.: Мир, 1989. – 544 с.

2). Халсалл Ф. Передача данных, сети компьютеров и взаимосвязь открытых систем: Пер. с англ. – М.: Радио и связь, 1995. – 408 с.

1. Концевик канального уровня, как правило, представляет собой контрольные биты CRC циклического кода (Cyclic Redundancy Check). [↑](#footnote-ref-1)
2. Терминология, вытекающая из OSI-модели, структурно не очень проста и не всегда тщательно соблюдается на практике. Поскольку целью передачи являются пакеты данных, их, обычно, и имеют в виду в какие бы кадры на разных этапах в процессе преобразований они ни облекались. Несмотря на то, что при исследовании протоколов корректнее пользоваться термином «кадр j-уровня», в ряде случаев приходится употреблять термин «пакет данных» (см. ниже, например, пользовательский интерфейс модели). [↑](#footnote-ref-2)
3. Дэйтаграмма (от английского datagram) – тот же кадр и пакет данных, который, однако, преодолевает сеть совсем иначе, под управлением других протоколов [↑](#footnote-ref-3)
4. Обратную связь в этом случае организует транспортный уровень приемной стороны. Таков, например широко используемы протокол ТСР (его неизменный спутник – протокол IP занимается передачей дэйтаграмм). [↑](#footnote-ref-4)
5. Можно отметить, что канальный уровень не является единственным, который несет ответственность за верность переданного пакета данных. На практике все уровни OSI‑модели вплоть до прикладного в той или иной мере занимаются обеспечением верности данных своими средствами.

   Другой группой функций канального уровня эталонной модели является *управление доступом к среде* в «многоточечных» соединениях Эти функции являются отдельной темой и не рассматриваются в данной лабораторной работе. Однако стоит отметить, что их выделяют в отдельный подуровень (УДС) канального уровня, а протоколы, рассматриваемые в лабораторной работе, относят к подуровню «управления линией передачи данных» (УЛПД). [↑](#footnote-ref-5)
6. Назовем неизбыточным протокол, который сводится к *однократной* передаче сигнала в *одном направлении*. [↑](#footnote-ref-6)
7. Нумерация (идентификация) пакетов в протокольном отношении оказывается непростой задачей. Простая, бесхитростная возрастающая (или убывающая) нумерация в потоке кадров оказывается неприемлемой. См., например, [1]. [↑](#footnote-ref-7)
8. Понятия «прямой» и «обратный канал» получают уже только протокольный смысл. Один и тот же физический тракт выступает то в роли прямого, то обратного канала. [↑](#footnote-ref-8)
9. Вспомним только, что каждый из пунктов может выступать в роли как отправителя, так и получателя данных.

   Можно заметить также, что протокол с остановкой и ожиданием похож на протокол с возвратом на 1 шаг назад, т.е. является частным вырожденным случаем рассматриваемого протокола, но с другой формой подтверждения верности (ACK и NAK). [↑](#footnote-ref-9)
10. Трансляция пакета по обратному каналу может начинаться (если позволяют технические средства) не дожидаясь завершения приема кадра по прямому каналу, т.е. не дожидаясь прихода последнего бита. Если кадры имеют достаточно большую длину, это может ощутимо уменьшить потери времени в протокольном цикле. [↑](#footnote-ref-10)
11. Это же относится и к предыдущему протоколу с ретрансляцией всего кадра данных. [↑](#footnote-ref-11)
12. Это возможно, т.к. мощность множества различных двоичных последовательностей данных (пакетов) много больше мощности множества последовательностей CRC и, следовательно, одна и та же последовательность CRC может быть порождена несколькими последовательностями данных. [↑](#footnote-ref-12)
13. Эффективность точно меньше 50%, если пункт-получатель начинает ретрансляцию пакета данных по обратному каналу не раньше, чем запишет в буферную память последний бит из прямого канала. Зависит от технической реализации, но возможны варианты протокола, в которых ретрансляция начинается вскоре после записи в буфер *первых бит* пакета. А вот передача CRC по обратному каналу не может начаться раньше окончания приема кадра по прямому. [↑](#footnote-ref-13)
14. Цифра в обозначении многочлена означает старшую степень многочлена. Конкретная структура многочлена в данной работе не имеет значения. [↑](#footnote-ref-14)
15. Длина кадра данных не может быть выбрана меньше длины CRC и больше, чем 2L-1, где L – степень многочлена порождающего кода. Длина кадра подтверждения считается постоянной и подчиняется тем же ограничениям. [↑](#footnote-ref-15)
16. При моделировании считается, что 1 Кбит/с=1024 бит/с; 1 Мбит/с=1024 Кбит/с. [↑](#footnote-ref-16)
17. В окне интерфейса они названы «одиночными» (в противовес «пачкам»), что не совсем корректно. [↑](#footnote-ref-17)
18. Поскольку моделирующая программа сама генерирует вектор ошибки, то она знает точно, случилась ошибка в отдельном кадре или нет, и может подсчитать, сколько всего кадров подверглось искажению при “прохождении через линию”. На самом деле реальный функционирующий модуль канального уровня системы передачи данных “узнает” о возникновении ошибки только по факту ее обнаружения декодером. Незамеченная им ошибка, по сути, на канальном уровне ошибкой не считается: протокол ее «не видит». [↑](#footnote-ref-18)