



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

**МСЭ-Т**

СЕКТОР СТАНДАРТИЗАЦИИ  
ЭЛЕКТРОСВЯЗИ МСЭ

**X.606.1**

(02/2003)

СЕРИЯ X: СЕТИ ПЕРЕДАЧИ ДАННЫХ И  
ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ

Организация сети ВОС и системные аспекты –  
Организация сети

---

**Информационные технологии – Расширенный  
транспортный протокол связи: Спецификация  
управления QoS для симплексной групповой  
передачи**

Рекомендация МСЭ-Т X.606.1

---

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ X  
СЕТИ ПЕРЕДАЧИ ДАННЫХ И ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ

<b>СЕТИ ДАННЫХ ОБЩЕГО ПОЛЬЗОВАНИЯ</b>	
Службы и услуги	X.1–X.19
Интерфейсы	X.20–X.49
Передача, сигнализация и коммутация	X.50–X.89
Сетевые аспекты	X.90–X.149
Техническое обслуживание	X.150–X.179
Административные предписания	X.180–X.199
<b>ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ</b>	
Модель и обозначение	X.200–X.209
Определения служб	X.210–X.219
Спецификации протоколов в режиме с установлением соединений	X.220–X.229
Спецификации протоколов в режиме без установления соединений	X.230–X.239
Проформы PICS	X.240–X.259
Идентификация протоколов	X.260–X.269
Протоколы обеспечения безопасности	X.270–X.279
Управляемые объекты уровня	X.280–X.289
Испытание на соответствие	X.290–X.299
<b>ВЗАИМОДЕЙСТВИЕ МЕЖДУ СЕТЯМИ</b>	
Общие положения	X.300–X.349
Спутниковые системы передачи данных	X.350–X.369
IP-сети	X.370–X.399
<b>СИСТЕМЫ ОБРАБОТКИ СООБЩЕНИЙ</b>	X.400–X.499
<b>СПРАВОЧНЫЕ СИСТЕМЫ</b>	X.500–X.599
<b>ОРГАНИЗАЦИЯ СЕТИ ВОС И СИСТЕМНЫЕ АСПЕКТЫ</b>	
<b>Организация сети</b>	<b>X.600–X.629</b>
Эффективность	X.630–X.639
Качество обслуживания	X.640–X.649
Наименование, адресация и регистрация	X.650–X.679
Абстрактно-синтаксическая нотация версии 1 (ASN.1)	X.680–X.699
<b>УПРАВЛЕНИЕ В ВОС</b>	
Структура и архитектура управления системами	X.700–X.709
Служба и протокол связи для управления	X.710–X.719
Структура управляющей информации	X.720–X.729
Функции управления и функции ODMA	X.730–X.799
<b>БЕЗОПАСНОСТЬ</b>	X.800–X.849
<b>ПРИЛОЖЕНИЯ ВОС</b>	
Фиксация, параллельность и восстановление	X.850–X.859
Обработка транзакций	X.860–X.879
Удаленные операции	X.880–X.899
<b>ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА</b>	X.900–X.999

Для получения более подробной информации просьба обращаться к Перечню Рекомендаций МСЭ-Т

## Информационные технологии - Расширенный транспортный протокол связи: Спецификация управления QoS для симплексной групповой передачи

### Резюме

В настоящей Рекомендации | Международном стандарте (вторая часть ЕСТР) определяется сквозной групповой транспортный протокол для согласования QoS, контроля и поддержания функций в симплексном (один-ко-многим) групповом подключении. Этот протокол может использоваться групповыми приложениями типа мультимедиа в реальном масштабе времени, распределяя службы для поддержки их QoS требований.

Если управление QoS допускается, то согласование параметра QoS может быть выполнено в любое время в течение фазы создания подключения. Посылающий контроллер затем выбирает значения параметров из предложенных получателями. Если согласование не допускается, то отправитель задает предопределенные значения.

Уровень состояния QoS проверяется в течение фазы передачи данных. Каждый получатель измеряет реально полученные значения QoS и сообщает о состоянии каждого параметра своему родительскому контроллеру, используя модифицированные АСК пакеты. Отправитель объединяет значения состояния параметра, сообщенные получателями, чтобы составить представление о состоянии подключения QoS. Отправитель поддерживает операции QoS, такие как настройка скорости передачи данных, на основе анализа состояния подключения QoS.

### Источник

Рекомендация МСЭ-Т Х.606.1 была подготовлена 17-й Исследовательской комиссией МСЭ-Т (2001–2004 гг.) и утверждена 13 февраля 2003 года. Идентичный текст опубликован также в качестве Международного стандарта ИСО/МЭК 14476-2

### Ключевые слова

ЕСТР, транспортный групповой протокол, QoS, симплексная групповая передача .

## ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи. Сектор стандартизации электросвязи МСЭ (МСЭ-Т) является постоянным органом МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним с целью стандартизации электросвязи на всемирной основе.

Всемирная ассамблея по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяет темы для изучения Исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, вырабатывают Рекомендации по этим темам.

Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ.

В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

## ПРИМЕЧАНИЕ

В настоящей Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

## ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на то, что практическое применение или реализация этой Рекомендации может включать в себя использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, обоснованности или применимости заявленных прав интеллектуальной собственности, независимо от того, отстаиваются ли они членами МСЭ или другими сторонами вне процесса подготовки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ не получил извещения об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для реализации этой Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что это может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ.

© МСЭ 2004

Все права сохранены. Никакая часть данной публикации не может быть воспроизведена с помощью каких-либо средств без письменного разрешения МСЭ.

## СОДЕРЖАНИЕ

Стр.

1	Область применения .....	1
2	Нормативные ссылки .....	1
3	Определения .....	2
3.1	Термины, определенные в Рекомендации МСЭ Х.605   ИСО/МЭК 13252 .....	2
3.2	Термины, определенные в Рекомендации МСЭ Х.606   ИСО/МЭК 14476-1 .....	2
3.3	Термины, определенные в настоящей Рекомендации   Международном стандарте.....	2
4	Сокращения.....	2
4.1	Типы пакетов .....	2
4.2	Прочие сокращения.....	3
5	Соглашения .....	3
6	Общий обзор .....	3
7	Компоненты для управления QoS.....	6
7.1	Информационный элемент подключения .....	6
7.2	Параметры QoS .....	7
7.3	Элемент расширения QoS .....	7
7.4	Элемент подтверждения .....	9
7.5	Пакеты, используемые для управления QoS .....	10
8	Процедуры управления QoS.....	10
8.1	Согласование QoS .....	10
8.1.1	Процедуры согласования.....	11
8.1.2	Согласование QoS в иерархическом дереве .....	12
8.1.3	Согласование MSS .....	12
8.1.4	Резервирование ресурса.....	12
8.2	Контроль QoS .....	13
8.2.1	Генерация АСК .....	13
8.2.2	Измерение значений параметров QoS.....	13
8.2.3	Отображение на значение состояния параметра .....	14
8.2.4	Сообщение поставщику услуги .....	14
8.3	Поддержание QoS .....	15
8.3.1	Настройка скорости передачи данных .....	16
8.3.2	Приостановка и возобновление соединения.....	16
8.3.3	Отключение источника неисправности .....	17
8.3.4	Завершение подключения .....	17
9	Таймеры и переменные .....	17
9.1	Таймеры .....	17
9.2	Переменные операций .....	17
Приложение А - Межсетевое взаимодействие между ECTP и RSVP для резервирования ресурса .....		19
A.1	ECTP параметры QoS .....	19
A.2	Обзор RSVP .....	19
A.2.1	RSVP SENDER_TSPEC .....	19
A.2.2	RSVP ADSPEC .....	20
A.2.3	RSVP FLOWSPEC .....	20
A.2.4	RSVP API .....	20
A.3	Пример отображения параметра между RSVP и ECTP .....	21
A.4	Сценарий меж сетевого взаимодействия между ECTP и RSVP.....	21
Приложение В - Прикладные программные интерфейсы.....		24
B.1	Краткий обзор.....	24
B.1.1	Функции API .....	24
B.1.2	Использование функций API ECTP.....	24
B.2	Функции API ECTP .....	25
B.2.1	msocket().....	25
B.2.2	mbind().....	26
B.2.3	mascept().....	27
B.2.4	mconnect() .....	28

	<b>Стр.</b>
В.2.5 msend() .....	28
В.2.6 mrecv().....	29
В.2.7 mclose().....	30
В.2.8 mgetsockopt() и msetsockopt().....	30
В.3 Пример файла заголовка msocket.h .....	32
Справочная литература.....	36

## Введение

В настоящей Рекомендации | Международном стандарте определяется Усовершенствованный Транспортный Протокол Связи (ЕСТР), который является транспортным протоколом, разработанным для поддержки групповых приложений Интернета через сети, обеспечивающие поддержку групповых приложений. ЕСТР работает в сетях IPv4/IPv6, которые имеют IP возможность группового распределения с помощью групповых протоколов маршрутизации IGMP и IP, как показано на рисунке 1. ЕСТР может быть предоставлен посредством UDP.

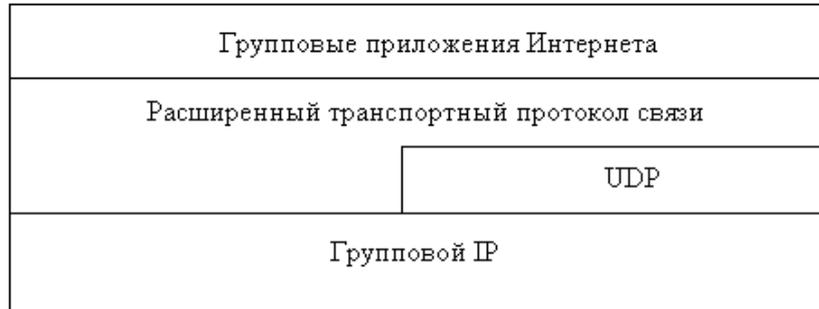


Рисунок 1 – Модель ЕСТР

ЕСТР используется для поддержки плотно контролируемых групповых подключений в симплексных, дуплексных и мультиплексных приложениях. Эта часть ЕСТР (часть 2) определяет функции управления QoS для устойчивого управления QoS подключения пользователей в симплексном групповом соединении. Функциональные возможности управления QoS состоят из согласования QoS, контроля QoS и операций поддерживания QoS. Процедуры протокола для управления надежностью в симплексном групповом транспорте определены в части 1 ЕСТР (Рекомендация МСЭ-Т X.606 | ИСО/МЭК 14476-1), которая составляет неотъемлемую часть данной Рекомендации | Международного стандарта. Другие разделы стандарта определяют процедуры управления и связанные функции управления QoS соответственно для дуплексного случая (X.ectp-3 | ИСО/МЭК 14476-3 и X.ectp-4 | ИСО/МЭК 14476-4) и для мультиплексного случая (X.ectp-5 | ИСО/МЭК 14476-5 и X.ectp-6 | ИСО/МЭК 14476-6).

В ЕСТР все предполагаемые элементы зарегистрированы в мультиабонентскую группу до создания подключения или сеанса. Эти элементы определяют зарегистрированную группу. Каждый получатель в зарегистрированной группе описан как зарегистрированный получатель. В процессе набора каждый элемент будет аутентифицирован. Информация группы, включая ключ группы, групповые IP адреса и номера портов, будет распределена среди зарегистрированных элементов в течение процесса набора. Для этих зарегистрированных элементов группы формируется подключение ЕСТР.

Отправитель - это основа для соединений мультиабонентских групп. Один отправитель на симплексном групповом подключении исполняет роль владельца подключения и обозначается как владелец с максимальным приоритетом (ВМП) в данной спецификации. Владелец подключения отвечает за полный контроль подключения, управляя созданием подключения и его завершением, прерыванием подключения и его возобновлением, операциями позднего связывания и разъединения.

Отправитель запускает процесс создания подключения. Некоторые или все зарегистрированные получатели будут участвовать в подключении, становясь "активными получателями". Получатели, активные на этой стадии, способны участвовать в согласовании желательного качества службы для сеанса. Любой зарегистрированный получатель, который не активен на этой стадии, может участвовать в подключении как присоединившийся позже, но он должен будет принять установленный QoS. Активный получатель может выйти из соединения.

После того как подключение создано, отправитель начинает передавать групповые данные. В период активности подключения отправитель контролирует состояние сеанса через пакеты управления с обратной связью от активных получателей.

Отправитель может предпринять ряд действий, если обнаружатся сетевые проблемы (например, сильная перегрузка) посредством обратной связи от активных получателей. Эти действия включают корректировку скорости передачи данных, временную приостановку групповой передачи данных или, как последний ресурс, завершение подключения.

Эти технические спецификации управления QoS могут использоваться в групповых приложениях для поддержки различных требований QoS и соответствующих моделей оплаты.



## МЕЖДУНАРОДНЫЙ СТАНДАРТ РЕКОМЕНДАЦИЯ МСЭ-Т

### Информационные технологии – Расширенный транспортный протокол связи: Спецификация управления QoS для симплексной групповой передачи

#### 1 Область применения

Настоящая Рекомендация | Международный стандарт является неотъемлемой частью Рекомендаций МСЭ-Т X.606.x | ИСО/МЭК 14476 "ЕСТР: Расширенный Транспортный Протокол Связи", которые являются семейством спецификаций протокола, предназначенного для поддержки групповых транспортных служб.

Рекомендация МСЭ-Т X.606 | ИСО/МЭК 14476-1 обеспечивает технические требования различных операций протокола для симплексной групповой передачи. Эти операции протокола включают такое управление подключением, как создание / завершение подключения и приостановку/возобновление подключения; управление составом пользователей, в том числе посредством таких операций, как позднее присоединение, отключение пользователя и контроль состава пользователей; а также защиту от ошибок для групповой передачи данных, т.е. обнаружение ошибок и восстановление.

В данной части Рекомендации | Международного стандарта даются технические требования управления QoS для осуществления желательного качества службы при симплексном групповом транспортном подключении.

Данная спецификация описывает следующие операции управления QoS:

##### а) QoS согласование

Для QoS согласования в данной спецификации предполагается, что желательный QoS уровень для групповой прикладной службы может быть выражен в терминах набора параметров QoS. QoS согласование выполняется через обмен управляющими пакетами между отправителем и получателями. Отправитель предлагает базовые значения параметров QoS, полученные от требований приложений, а затем каждый получатель может предлагать измененные значения, основанные на его системе и-или пропускной способности сети. Отправитель выносит решение относительно модифицированных значений, предложенных получателями. Базовые значения параметров QoS могут использоваться как входные параметры для резервирования сетевых ресурсов.

##### б) QoS контроль

Управление QoS в ЕСТР основано на обратной связи посредством управляющих пакетов получателей. Сообщения обратной связи от получателей дают возможность отправителю следить за числом активных получателей, а также контролировать состояние подключения для групповой передачи данных. Контроль QoS разработан для того, чтобы позволить отправителю диагностировать состояние подключения в терминах значений параметра QoS и, таким образом, предпринимать необходимые действия для поддержания состояния подключения на желательном уровне QoS. Проверенное состояние подключения будет сообщено приложению на стороне отправителя. Переданная информация может обеспечивать полезную статистику, например для целей составления счетов.

##### в) QoS поддержка

Основываясь на информации, получаемой по обратной связи от получателей, отправитель предпринимает действие(я) для поддержки состояния подключения на желательном уровне QoS. Эти действия поддержки QoS включают настройку скорости передачи данных, приостановку и возобновление подключения, обнаружение нарушений и операции завершения подключения. Эти функции контроля и поддержки QoS, основанные на контролируемом состоянии параметра, обеспечивают управление перегрузкой на основе скорости передачи.

Настоящая Рекомендация | Международный стандарт является неотъемлемой частью Рекомендаций МСЭ-Т X.606.x | ИСО/МЭК 14476, которые состоят из 6 частей. Все компоненты протокола, включая форматы пакета и процедуры протокола, указанные в Рекомендации МСЭ-Т X.606 | ИСО/МЭК 14476-1, также являются допустимыми в данной Рекомендации | Международном стандарте.

#### 2 Нормативные ссылки

Следующие Рекомендации и Международные стандарты содержат положения, которые, через ссылку в этом тексте, составляют положения настоящей Рекомендации | Международного стандарта. Во время публикации обозначенные издания были действительны. Все Рекомендации и Стандарты могут пересматриваться, и стороны соглашений, основанных на настоящей Рекомендации | Международном стандарте, должны изучать возможность применения самого современного издания Рекомендаций и Стандартов, перечисленных ниже. Члены МЭК и ИСО поддерживают регистры действующих в настоящее время Международных стандартов. Бюро Стандартизации Электросвязи МСЭ поддерживает список действующих в настоящее время Рекомендаций МСЭ-Т.

– Рекомендация МСЭ-Т X.601 (2000), Многоточечная одноранговая структура связи.

- Рекомендация МСЭ-Т X.605 (1998) | ИСО/МЭК 13252:1999, *Информационная технология – Определение Службы расширенного транспортного протокола связи.*
- Рекомендация МСЭ-Т X.606 (2001) | ИСО/МЭК 14476-1:2002, *Информационная технология – Расширенный транспортный протокол связи: Спецификация симплексной групповой передачи.*

### 3 Определения

#### 3.1 Термины, определенные в Рекомендации МСЭ X.605 | ИСО/МЭК 13252

Настоящая Рекомендация | Международный стандарт основывается на положениях, изложенных в Расширенном транспортном протоколе связи (Рекомендация МСЭ-Т X.605 | ИСО/МЭК 13252).

- a) параметры QoS;
- b) согласование QoS; и
- c) арбитраж QoS

#### 3.2 Термины, определенные в Рекомендации МСЭ X.606 | ИСО/МЭК 14476-1

Настоящая Рекомендация | Международный стандарт основывается на положениях, изложенных в спецификации симплексной групповой передачи в ЕСТР (Рекомендация МСЭ-Т X.606 | ИСО/МЭК 14476-1).

- a) приложение;
- b) пакет;
- c) поставщик услуги;
- d) пользователь услуги;
- e) дерево;
- f) родитель; и
- g) потомок.

#### 3.3 Термины, определенные в настоящей Рекомендации | Международном стандарте

Для целей настоящей Рекомендации | Международного стандарта применяются следующие определения:

- a) **Контроль QoS:** является операцией протокола, которая используется для диагностики текущего состояния подключения. Для контроля QoS от каждого пользователя услуги требуется измерять действующие значения параметров и сообщать о них поставщику услуги. Поставщик услуги объединяет информацию о состоянии, полученную от пользователей услуги.
- b) **Поддержание QoS:** является операцией протокола, которая используется для поддержания состояния подключения на желательном уровне QoS. Поставщик услуги предпринимает действия по поддержанию QoS на основе контролируемой информации о состоянии.

### 4 Сокращения

Для целей настоящей Рекомендации | Международного стандарта применяются следующие сокращения.

#### 4.1 Типы пакетов

ACK	Подтверждение
CC	Подтверждение о создании подключения
CR	Запрос на создание подключения
CT	Завершение подключения
DT	Данные
HB	Подтверждение работоспособности
JC	Подтверждение позднего присоединения
JR	Запрос на позднее присоединение
LR	Запрос на отключение
ND	Отсутствие данных
RD	Данные повторной передачи

## 4.2 Прочие сокращения

API	Прикладные программные интерфейсы
CHQ	Управляемое нивысшее качество
Diffserv	Дифференцированные услуги
ECTP	Расширенный транспортный протокол связи
ECTS	Расширенная транспортная служба связи
IP	Протокол Интернета
LQA	Самое низкое приемлемое качество
MSS	Максимальный размер сегмента
OT	Операционный адресат
QoS	Качество обслуживания
RSVP	Протокол резервирования ресурса

## 5 Соглашения

В настоящей Рекомендации | Международном стандарте ключевые слова "ДОЛЖЕН", "ТРЕБУЕТСЯ", "БУДЕТ", "НЕ ДОЛЖЕН", "НЕ БУДЕТ", "СЛЕДУЕТ", "НЕ СЛЕДУЕТ", "МОЖЕТ" и "НЕОБЯЗАТЕЛЬНО" указывают на требуемые уровни для различных реализаций ECTP.

## 6 Общий обзор

В настоящей Рекомендации | Международном стандарте приведены спецификации управления QoS для групповых (симплексных) транспортных подключений "один ко многим". Данная спецификация описывает следующие операции управления QoS:

- 1) Согласование QoS, включая резервирование сетевых ресурсов;
- 2) Контроль QoS; и
- 3) Поддержание QoS.

В фазе создания подключения поставщик услуги сообщает пользователям услуги: допускается ли управление QoS. Если управление QoS допускается, то поставщик услуги должен также определить, действительно ли согласование QoS будет выполнено при подключении. Операции контроля и поддержания QoS выполняются только в том случае, если допускается управление QoS.

На рисунке 2 показаны эти операции управления QoS для симплексного группового подключения. На рисунке операции протокола, отмеченные пунктирными линиями, определены в Рекомендации МСЭ-Т X.600 | ИСО/МЭК 14476-1.



Рисунок 2 – Управление QoS в ЕСТР

В общем случае QoS - это качество услуги, требуемое для удовлетворительного приема данных прикладной программы пользователем услуги, например для достижения желательного качества звука и изображения дисплея. В настоящей спецификации принимается, что требования QoS приложения выражены в терминах одного или большего количества параметров QoS, таких как производительность, задержка передачи, флуктуация задержки и коэффициент потери информации. В зависимости от требований приложения некоторые из этих параметров QoS не могут использоваться при подключении. Например, служба, работающая не в режиме реального времени, не может установить требование на время задержки передачи.

В соответствии с требованиями приложений поставщик услуги определит заданные значения для каждого параметра QoS. Способ отображения требования приложения на значения параметров находится вне области применения данной спецификации. Для выполнения таких отображений могут использоваться прикладные программы.

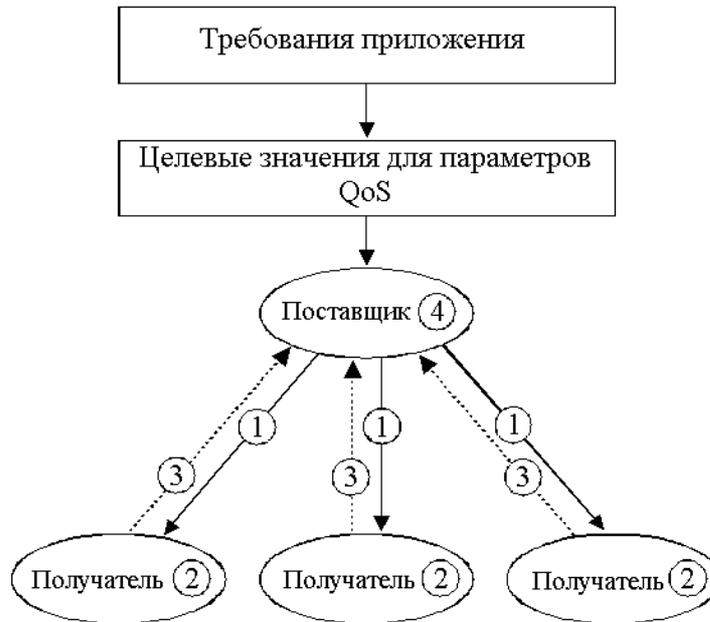
Согласование QoS выполняется в фазе создания подключения. Поставщик услуги предлагает желательные значения каждого параметра QoS для всех пользователей услуги групповой рассылкой. Для обеспечения производительности определены три значения: CHQ (контролируемое самое высокое качество), OT (рабочее значение) и LQA (самое низкое допустимое качество). Для других параметров, таких как задержка передачи, флуктуация задержки и коэффициент потери информации, задаются только два значения: OT и LQA.

Если допускается согласование QoS, тогда каждый пользователь услуги может предлагать модификации предложенных значений параметра поставщика услуги. Эти измененные значения будут определены, исходя из пропускной способности системы на стороне пользователя услуги и в сети. Следующие ограничения наложены для модификации значений параметра пользователями услуги:

- 1) значения OT не должны быть изменены пользователями услуги;
- 2) значения, измененные пользователями услуги, должны быть в пределах значений LQA и CHQ, предложенных поставщиком услуги.

Значения параметра, измененные пользователями услуги, доставляются поставщику услуги через сообщения АСК. Поставщик услуги рассматривает различные значения параметра для различных пользователей услуги, принимая согласованный диапазон значений.

На рисунке 3 показана общая схема согласования QoS, которая может быть реализована в ЕСТР. Согласно требованиям приложения набор значений параметров QoS будет конфигурирован поставщиком услуги. Поставщик услуги сообщает пользователям услуги предустановленные значения (шаг 1). Базируясь на этих значениях, каждый пользователь услуги начинает резервирование ресурса с помощью RSVP или Diffserv (шаг 2). Если согласование QoS допускается при подключении, то каждый пользователь услуги может предлагать модифицированные значения для параметров QoS (шаг 3). Исходя из модифицированных значений параметра, поставщик услуги определяет согласованные значения (шаг 4). Эти согласованные значения направляются пользователю услуги последовательностью пакетов NB или JC и будут использоваться для контроля и поддержания QoS.



X.606.1\_F03

- ① Целевые значения, предложенные поставщиком услуг
- ② Резервирование сетевых ресурсов получателем услуг
- ③ Значения, измененные получателем услуг
- ④ Согласование измененных значений поставщиком услуг

**Рисунок 3 – Согласование QoS**

После того как создано ЕСТР подключение и если допускается управление QoS, то выполняются операции контроля и поддержания QoS для групповой передачи данных. Для контроля QoS от каждого пользователя услуги требуется, чтобы он измерял значения параметра QoS. Базируясь на измеренных и согласованных значениях, пользователь услуги определяет значение состояния параметра для каждого параметра как целое число: нормальное (0), приемлемое (1), возможно аварийное (2) или аварийное (3). Эти значения состояния будут направлены поставщику услуги посредством пакетов АСК.

Поставщик услуги объединяет значения состояния параметра, полученные от пользователей услуги. Если используется дерево управления, то каждый родитель узла LO объединяет измеренные значения, сообщенные от его дочерних записей, и перенаправляет далее агрегированное значение(я) своему родителю, используя пакеты АСК.

На рисунке 4 показаны операции контроля и поддержания QoS, описанные в данной спецификации.

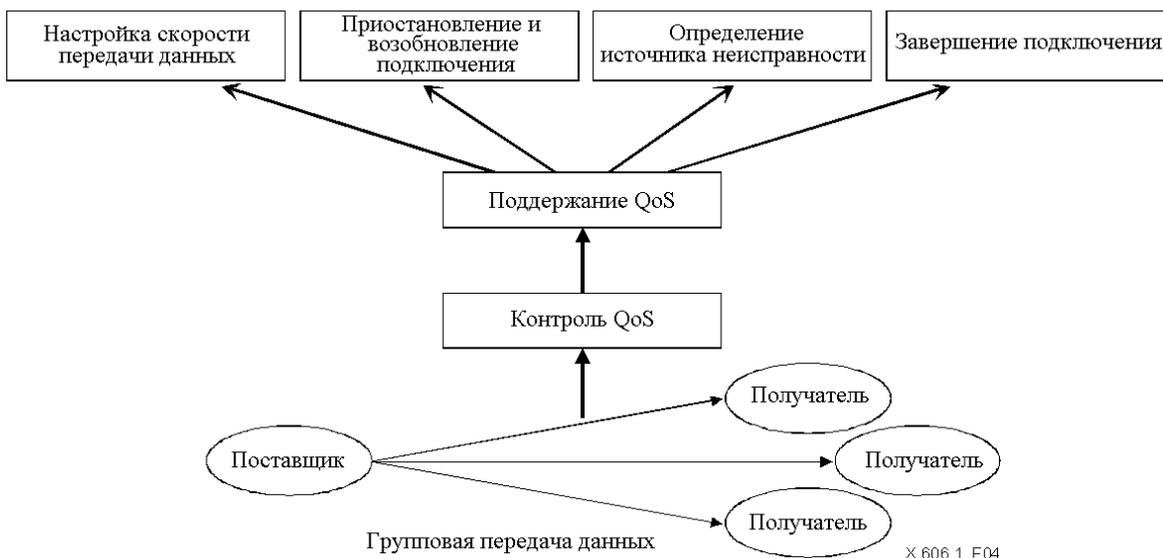


Рисунок 4 - Общая схема контроля и поддержания QoS

Поставщик услуги осуществляет операции поддержания QoS, чтобы удерживать состояние подключения на желательном уровне QoS на основе контролируемых значений состояния. Заранее определяются правила запуска операций поддержания QoS, такие как настройка скорости передачи данных, приостановление и возобновление подключения, определение источника неисправности и завершение подключения. Те или иные соответствующие действия будут предприняты в зависимости от того, сколько пользователей услуги находятся в неисправном или в возможно неисправном состоянии.

## 7 Компоненты для управления QoS

В данном разделе описываются ЕСТР компоненты протокола, требуемые для операций управления QoS. Все компоненты являются расширениями тех, которые уже определены в Рекомендации МСЭ-Т X.606 | ИСО/МЭК 14476-1.

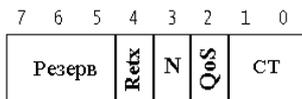
### 7.1 Информационный элемент подключения

На рисунке 5 показан информационный элемент подключения, определенный в Рекомендации МСЭ-Т X.606 | ИСО/МЭК 14476-1.

0	8	16	24	31
Следующий элемент	Версия	Флаги	Опции конфигурации дерева	Максимальный уровень дерева
Интервал создания подключения		Битовый размер АСК		Зарезервировано

Рисунок 5 – Информационный элемент подключения

Для управления QoS ЕСТР поставщик услуги определяет следующие три поля в октете 'Флаги':



- 1) *QoS* - это бит флага для указания того, допускается ли управление QoS (1) или нет (0) при подключении. Если этот бит установлен в '1', то вызываются все процедуры для управления QoS. Значение по умолчанию - '0'.
- 2) *N (согласование)* – это бит флага для указания того, допускается ли согласование QoS (1) или нет (0) при подключении. Если этот бит установлен в '1', то каждой пользователь услуги может предложить свои собственные значения параметра. Значение по умолчанию - '0'.
- 3) *Retx (повторная передача)* - это бит флага для указания того, выполняются ли повторные передачи родителем (0) или нет (1). Если этот бит установлен в '1', то поставщик услуги или родители не должны ретранслировать RD пакеты, даже если получены запросы повторной передачи посредством пакетов АСК. Значение по умолчанию - '0'.

Бит QoS должен быть установлен в "1" (допустим QoS) прежде чем установлен N бит. Существуют три возможных варианта.

- a) Бит QoS установлен в "1", а N бит - в "0", т.е. QoS должен использоваться при подключении, и значения QoS будут предоставлены поставщиком услуги. Пользователи услуги не могут согласовывать эти значения.
- b) Оба бита установлены в "1", т.е. QoS должен использоваться при подключении, и значения параметра QoS могут быть согласованы между пользователями услуги и поставщиком услуги.
- c) Бит QoS установлен в "0", т.е. QoS не должен использоваться при подключении. В этом случае N бит не используется.

Установка Retx (бит повторной передачи) может быть сделана независимо от установки бита QoS. Ожидается, что потоковые мультимедийные приложения в реальном масштабе времени не будут нуждаться в восстановлении ошибок на основе повторных передач, но они нуждаются в функциях управления QoS. Даже в этом случае пакеты АСК все еще используются для передачи информации о состоянии подключения.

## 7.2 Параметры QoS

В данной спецификации определены следующие четыре параметра QoS:

- 1) пропускная способность (в октетах в секунду);
- 2) транзитная задержка (в миллисекундах);
- 3) флуктуация транзитной задержки (в миллисекундах);
- 4) коэффициент потери информации (в процентах).

Пропускная способность - это объем вывода данных прикладной программы за заданный период времени. Требуемая пропускная способность - это значение пропускной способности, необходимое для отображения данных прикладной программы. Приложения генерируют групповые данные, а ЕСТР поставщик услуги передаст их на основе требуемого значения(ий) пропускной способности. Фактическая скорость приема данных на стороне пользователя услуги будет зависеть от скорости передачи данных, состояния сети, пропускной способности оконечной системы и т.д.

Для пропускной способности поставщик услуги должен конфигурировать следующие значения:

- 1) CHQ пропускная способность;
- 2) OT пропускная способность;
- 3) LQA пропускная способность.

Между ними должны соблюдаться следующие неравенства: LQA пропускная способность  $\leq$  OT пропускная способность  $\leq$  CHQ пропускная способность.

Транзитная задержка - это время передачи от поставщика услуги до пользователя услуги. Для требуемого отображения групповых данных поставщик услуги может конфигурировать следующие значения:

- 1) OT транзитная задержка;
- 2) LQA транзитная задержка.

Между ними должны соблюдаться следующие неравенства: OT транзитная задержка  $\leq$  LQA транзитная задержка.

Флуктуация транзитной задержки - это вариация значений транзитной задержки. Для требуемого отображения данных поставщик услуги может конфигурировать следующие значения:

- 1) OT флуктуация транзитной задержки;
- 2) LQA флуктуация транзитной задержки.

Между ними должны соблюдаться следующие неравенства: OT флуктуация транзитной задержки  $\leq$  LQA флуктуация транзитной задержки.

Коэффициент потери информации определен как отношение количества потерянных данных к количеству полностью переданных данных. Для требуемого отображения данных поставщик услуги может конфигурировать следующие значения:

- 1) OT коэффициент потерь;
- 2) LQA коэффициент потерь.

Между ними должны соблюдаться следующие неравенства: OT коэффициент потерь  $\leq$  LQA коэффициент потерь.

## 7.3 Элемент расширения QoS

Для управления QoS в настоящей спецификации определен элемент расширения QoS. Все добавочные элементы, используемые в ЕСТР, перечислены ниже.

Таблица 1 - Таблица кодирования элементов расширения ЕСТР

Элемент	Код
Информация подключения	0001
Уведомление	0010
Дерево участников	0011
Временная метка	0100
QoS	0101
Отсутствие элемента	0000

Элемент расширения QoS определяет Максимальный Размер Сегмента (MSS) и начальные значения ЕСТР для параметров QoS, описанных в п. 7.2. Как показано на рисунке 6, элемент QoS имеет длину '28' октетов:

0	8	16	24	31
следующий элемент	версия	флаги QoS	Максимальный размер сегмента	
CHQ пропускная способность				
OT пропускная способность				
LQA пропускная способность				
OT транзитная задержка			LQA транзитная задержка	
OT флуктуация транзитной задержки			LQA флуктуация транзитной задержки	
OT коэффициент потерь		LQA коэффициент потерь	зарезервировано	

Рисунок 6 – Элемент расширения QoS

Определены следующие параметры :

- a) *Следующий элемент* - указывает тип следующего элемента после данного элемента QoS;
- b) *Версия* - определяет текущую версию этого элемента, начиная с '1';
- c) *QoS флаги* - это октет флага, определяющий используется ли каждый из параметров QoS и MSS при подключении. Кодирование этого октета показано на следующем рисунке. Если бит установлен в '1', то соответствующий параметр QoS или MSS будет использоваться. Значение по умолчанию равно '0' для каждого бита.

7	6	5	4	3	2	1	0
Резерв	Е	Д	С	В	А		

- 1) А - пропускная способность;
- 2) В - транзитная задержка;
- 3) С - флуктуация транзитной задержки;
- 4) D - коэффициент потери информации;
- 5) Е - максимальный размер сегмента (MSS);
- 6) *Зарезервировано* - зарезервировано для будущего использования.
- d) *Максимальный Размер Сегмента (MSS)* - максимальный размер сегмента ЕСТР или пакета в модуле октетов. Если 'Е' бит QoS флагов установлен в '1', то MSS согласуется (см. п. 8.1.3). В противном случае будет использоваться значение MSS - 1024 , заданное по умолчанию.
- e) *Значения пропускной способности* - значения, выраженные в октетах в секунду в виде 32-разрядного целого числа без знака. Следующие значения допустимы только, если 'А' бит QoS флагов установлен в '1':
  - 1) CHQ пропускная способность - верхний предел пропускной способности;
  - 2) OT пропускная способность - требуемая пропускная способность, необходимая для отображения групповых данных;
  - 3) LQA пропускная способность - нижний предел пропускной способности.
- f) *Значения транзитной задержки* - значения, выраженные в миллисекундах в виде 16-разрядного целого числа без знака. Следующие значения допустимы только, если 'В' бит QoS флагов установлен в '1':

- 1) ОТ транзитная задержка - заданная транзитная задержка для требуемого отображения групповых данных;
- 2) LQA транзитная задержка - максимально допустимая транзитная задержка.
- g) *Флуктуации значений транзитной задержки* - значения, выраженные в миллисекундах в виде 16-разрядного целого числа без знака. Следующие значения допустимы только, если бит 'C' QoS флагов установлен в '1':
  - 1) ОТ флуктуация транзитной задержки - заданная флуктуация транзитной задержки для требуемого отображения групповых данных;
  - 2) LQA флуктуация транзитной задержки - максимально допустимая флуктуация транзитной задержки.
- h) *Значения коэффициента потери информации* - значения, выраженные в процентах в виде 8-разрядного целого числа без знака в диапазоне от 0 до 100. Следующие значения допустимы только, если 'D' бит QoS флагов установлен в '1':
  - 1) ОТ коэффициент потерь - заданный коэффициент потерь для требуемого отображения групповых данных;
  - 2) LQA коэффициент потерь - максимально допустимый коэффициент потерь.
- i) *Зарезервировано* - зарезервировано для будущего использования.

Элемент QoS используется поставщиком услуги для сообщения пользователям услуги значения параметров QoS пакетом CR в фазе создания подключения. При согласовании QoS элемент QoS также используется, когда пользователь услуги предлагает его собственные измененные значения поставщику услуги. Согласованные значения QoS будут сообщены пользователям, которые подключатся позднее, посредством пакета JC, а существующим пользователям услуг посредством пакетов HB.

Эти значения QoS также используются в операциях контроля и поддержания QoS.

#### 7.4 Элемент подтверждения

Для контроля QoS от каждого пользователя услуги требуется, чтобы он измерял значения параметров в месте своего подключения. Измеренное значение параметра отображается на значение состояния параметра. Значение состояния - это целое число, например 0, 1, 2, или 3. Большее значение состояния указывает на худшее состояние подключения.

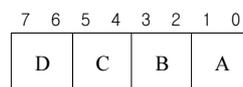
Значения состояния направляются поставщику услуги посредством пакетов AСК. Элемент подтверждения пакета AСК содержит значения состояния параметров QoS, используемых при подключении.

Элемент подтверждения, определенный в Рекомендации МСЭ-Т X.606 | ИСО/МЭК 14476-1, показан ниже. На рисунке октет '*состояние параметра*' ранее определен в настоящей спецификации.



Рисунок 7 – Элемент подтверждения

Октет '*состояние параметра*' имеет следующую структуру:



- a) A - два бита, указывающих на значение состояния для измеренной пропускной способности;
- b) B - два бита, указывающих на значение состояния для измеренной транзитной задержки;
- c) C - два бита, указывающих на значение состояния для измеренной флуктуации транзитной задержки;
- d) D - два бита, указывающих на значение состояния для измеренного коэффициента потерь пакета.

Значение состояния, состоящее из двух битов, имеет одно из следующих значений:

- a) 00 - определяет '0' как значение состояния;
- b) 01 - определяет '1' как значение состояния;
- c) 10 - определяет '2' как значение состояния;
- d) 11 - определяет '3' как значение состояния.

Детальные схемы отображения измеренного значения параметра к значению состояния описаны в п. 8.2.3.

## 7.5 Пакеты, используемые для управления QoS

В таблице 2 дан список ЕСТР пакетов, используемых для управления QoS.

Таблица 2 – ЕСТР пакеты, используемые для управления QoS

Тип пакета	Элемент расширения				
	Информация о подключении	Дерево участников	Подтверждение	Временная метка	QoS
CR	О				О
CC		О			О
HB		О		О	О
АСК		О	О	О	
IC	О				О

Пакет CR содержит элемент QoS. Это используется поставщиком услуги, чтобы предложить (или установить) значения параметров QoS, которые используются при подключении. Эти значения могут использоваться механизмами резервирования ресурса и протоколами, например RSVP, если они допустимы в сети. Если допускается согласование QoS, то каждый пользователь услуги отвечает поставщику услуги со своими предложенными значениями для параметров QoS посредством пакета CC. Поставщик услуги рассмотрит поступившие предложения, и выбранные значения параметров QoS будут направлены пользователям услуги посредством пакетов HB. При последующих подключениях пользователям будет направляться заданный параметр QoS, используемый в действующем подключении (первоначально или выбранный, или установленный), посредством пакета IC (см. п. 8.1).

Заданные или выбранные значения будут использоваться в операциях контроля и поддержания QoS. Пакеты АСК используются для передачи значения состояния параметров QoS, полученных на стороне пользователя услуги (см. п. 8.2).

## 8 Процедуры управления QoS

В ЕСТР управление QoS включает следующие операции:

- 1) согласование QoS возможно с резервированием сетевых ресурсов;
- 2) контроль QoS;
- 3) поддержание QoS.

Согласование QoS выполняется в фазе создания подключения, в то время как операции контроля и поддержания QoS будут выполняться в фазе передачи данных.

Если управление QoS допускается при подключении, то операции контроля и поддержания QoS будут выполнены по умолчанию. С другой стороны, согласование QoS допускается только тогда, когда бит *N* в октете 'Флаги' элемента информации подключения установлен в '1'.

### 8.1 Согласование QoS

Поставщик услуги ЕСТР передает пакет CR всем пользователям услуги, чтобы начать фазу создания подключения. Пакет CR содержит предложенное (или установленное) значение каждого параметра QoS, например CHQ, OT и LQA. Каждый пользователь услуги может обратиться к этим значениям для резервирования ресурса (см. п. 8.1.4). Если согласование QoS допускается при подключении, то процедуры согласования активизированы (см. п. 8.1.1). Установленные или согласованные значения впоследствии используются при контроле и поддержании QoS (см. п. 8.2 и п. 8.3).

Если согласование QoS допускается при подключении, то каждый пользователь услуги может предлагать новое измененное значение в ответ на заданное значение параметра, предложенное поставщиком услуги. Чтобы предложить новое значение, пользователь услуги должен иметь возможность идентифицировать системные или сетевые ресурсы, которые будут использованы. Например, измененное значение пропускной способности может быть оценено относительно пропускной способности линий передачи, доступных в узле пользователя услуги (например, DSL, кабельный модем, беспроводные сети и т.д.). Измененное значение может также быть определено рассмотрением требования конечного пользователя в узле пользователя услуги. Для оконечного компьютера хоста существует возможность использования программы, чтобы определить измененное значение параметра для согласования на основе сетевых и системных ресурсов, а также на требованиях конечного пользователя. Однако в реальных ситуациях не просто точно идентифицировать емкость ресурса сетей, связанных с пользователем услуги. Соответственно, по крайней мере в ближайшем будущем, согласование QoS будет делаться на основе требований конечного пользователя на прикладном уровне или на пропускной способности системы конечного компьютера хоста.

В данной спецификации от поставщика услуги требуется посредством элемента расширения QoS определить, является ли каждый параметр QoS предметом согласования (см. п. 7.3). Для параметров, которые являются согласуемыми, пользователь услуги может предлагать измененные значения. Если пользователь услуги не пожелает модифицировать параметр QoS, то он возвратит тот же самый элемент QoS, полученный от поставщика услуги.

### 8.1.1 Процедуры согласования

Если допускается согласование QoS при подключении, то каждый пользователь услуги отвечает поставщику услуги посредством пакета CC, содержащего измененные значения для соответствующих параметров QoS.

Этот подраздел описывает процедуры согласования QoS для параметра пропускной способности, который имеет три значения: LQA, OT и CHQ. Процедуры согласования для других параметров, таких как задержка, флуктуация и коэффициент потерь, проводятся таким же образом, за исключением того, что эти параметры не имеют значений CHQ.

Во время согласования QoS пользователи услуги не должны модифицировать значение OT для каждого параметра. Детальные процедуры согласования QoS описаны ниже и показаны на рисунке 8.

- 1) Поставщик услуги предлагает заданные значения параметра:  
Исходя из требований приложений, поставщик услуги определяет значения параметра:  $LQA_o$ ,  $OT_o$ ,  $CHQ_o$ , где  $LQA_o < OT_o < CHQ_o$ , а затем передает пакет CR с элементом расширения QoS всем пользователям услуги.
- 2) Пользователи услуги модифицируют значения параметра:  
В ответ на значения, предложенные поставщиком услуги, каждый  $i$ -й пользователь услуги может предложить измененные значения:  $LQA_i$  и  $CHQ_i$ . Значение  $OT_o$  не должно изменяться. Таким образом, заданы следующие неравенства:  $LQA_o < LQA_i < OT_o < CHQ_i < CHQ_o$  для каждого  $i$ -го пользователя услуги. Каждый пользователь услуги направляет измененные значения поставщику услуги посредством пакета CC.
- 3) Поставщик услуги выбирает измененные значения параметра:  
Поставщик услуги выбирает измененные значения параметра, предложенные пользователями услуги, следующим образом:  
 $CHQ_{min}$  = минимальное  $CHQ_i$ , для каждого  $i$ -го пользователя услуги  
 $LQA_{max}$  = максимальное  $LQA_i$ , для каждого  $i$ -го пользователя услуги  
 $LQA_{max}$  и  $CHQ_{min}$  - согласованные значения параметра, которые следуют из согласования QoS.
- 4) Поставщик услуги сообщает выбранные значения параметра:  
Поставщик услуги сообщает  $LQA_{max}$ ,  $CHQ_{min}$  и значения OT пользователям услуги посредством пакетов NB и JC.

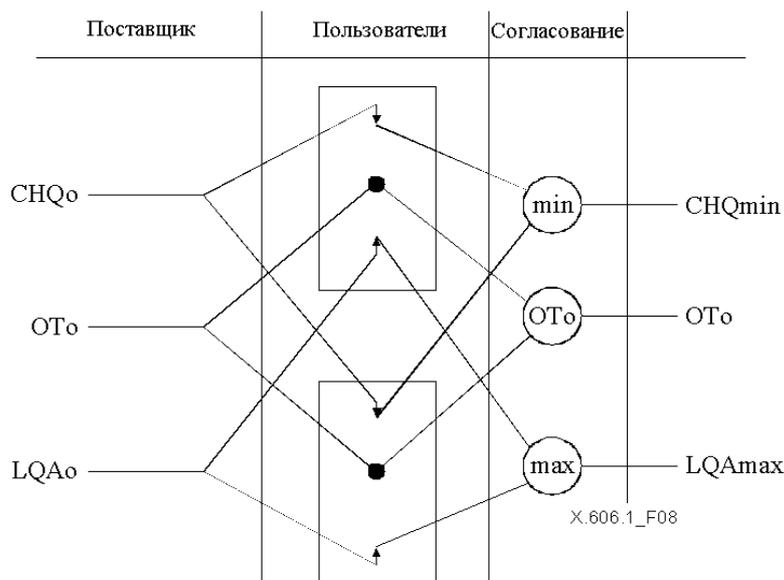


Рисунок 8 – Процедуры согласования QoS

Для параметров задержки, флуктуации и коэффициента потерь вместо  $LQA_{max}$  будет получено значение  $LQA_{min}$ , так как значения CHQ не используются и значение  $OT <$  значения LQA для этих параметров. Таким образом:  
 $LQA_{min}$  = минимальное  $LQA_i$ , для каждого  $i$ -го пользователя услуги

Согласование QoS не выполняется для пользователей услуги, которые присоединятся позднее. Поставщик услуги только уведомляет такого пользователя услуги о согласованных значениях параметра посредством пакета JC.

### 8.1.2 Согласование QoS в иерархическом дереве

Если используется дерево управления с более чем двумя уровнями, то каждый родитель LO должен выполнить процедуры арбитража QoS для измененных значений, предложенных его потомками (шаг 3 в п. 8.1.1). В этом случае правило арбитража состоит в следующем:

$CHQ_{min} ' =$  минимальное  $CHQ_i$ , для каждого  $i$ -го потомка

$LQA_{max} ' =$  максимальное  $LQA_i$ , для каждого  $i$ -го потомка

Родитель направляет арбитражные значения  $CHQ_{min} '$  и  $LQA_{max} '$  своему родителю посредством пакета CC. Таким образом, поставщик услуги принимает пакеты CC от всех своих потомков, а затем получает арбитражные значения параметра для всех пользователей услуги.

### 8.1.3 Согласование MSS

*MSS* представляет максимальный размер пакета, который зависит от *Максимального Блока Передачи данных (MTU)* для передач уровня соединения. Типичные значения *MTU* - 1500 октетов для Ethernet, 1492 октета для IEEE 802, 4352 октета для FDDI, 576 октетов для X.25 и так далее. *MTU* определяет формат кадра для передач уровня соединения и, следовательно, значение *MSS*.

Значение *MSS* по умолчанию для ECTP равно 1024 октетам, которое поддерживает большинство типов соединения, кроме X.25. Если поставщик услуги и/или пользователь услуги не могут идентифицировать свое значение *MSS*, то берется значение *MSS* по умолчанию - 1024 октетов.

В ECTP, если указано согласование *MSS* (см. п. 7.3), выполняются следующие процедуры :

- 1) Поставщик услуги записывает свое значение *MSS* в пакет CR, а затем передает его всем пользователям услуги.
- 2) Каждый пользователь услуги предлагает свое собственное значение *MSS* через пакет CC. На *MSS* будет воздействовать *MTU* местной сети, которой принадлежит пользователь услуги . Если *MSS* пользователя услуги больше, чем *MSS* поставщика услуги, то пользователь услуги выберет *MSS* поставщика услуги.
- 3) Поставщик выбирает *MSS* подключения, исходя из минимального значения *MSS* для всех пользователей услуги.

### 8.1.4 Резервирование ресурса

Сам ECTP не может гарантировать уровни QoS, требуемые приложениями. Однако значения параметров QoS могут использоваться при резервировании сетевых ресурсов, таких как интегрированные службы с RSVP и дифференцированные службы (Diffserv).

Модель RSVP хорошо соответствует ECTP протоколу, так как резервирование ресурса сделано сквозным способом. В сетях, где допускается RSVP , периодические PATH сообщения RSVP будут направлены пользователям услуги. В ответ на PATH сообщение каждый пользователь услуги посылает периодические RSVP RESV сообщения поставщику услуги через групповой информационный канал.

Если RSVP используется с ECTP, то значения ECTP параметров QoS будут вызываться в конфигурации RSVP дескрипторов трафика для RSVP TSPEC (или FLOWSPEC), содержащихся в PATH сообщении RSVP. Эти PATH сообщения будут направлены в соответствии с CR пакетом и NB пакетами, переданными поставщиком услуги. Следует отметить, что RSVP модули управляются отдельно от ECTP в конечных системах. Это означает, что резервирование ресурса и функции управления в RSVP будут выполнены только теми RSVP демонами, которых это касается.

В Приложении А описывается пример межсетевого взаимодействия между ECTP и RSVP, наряду с возможной схемой отображения между связанным параметрами QoS и параметрами трафика.

Дифференцированная служебная модель обеспечивает дифференцированный класс обслуживания для IP трафика с целью поддержки различных типов приложений и конкретных деловых требований. Небольшая комбинация битов в каждом пакете в октете IPv4 TOS (или DSCP) или в октете IPv6 Класс Трафика используется для метки пакета, чтобы принять заданную ускоряющую интерпретацию, или режим "прыжка", в каждом сетевом узле. Обычное понимание использования и интерпретации этой комбинации битов требуется для междоменного использования, взаимодействия между многими поставщиками и для непротиворечивого рассуждения относительно ожидаемых служебных характеристик в сети. До настоящего времени конкретная схема межсетевого взаимодействия между ECTP и Diffserv не была идентифицирована.

ECTP должен использовать механизмы QoS основной сети, такие как RSVP и Diffserv, чтобы установить сетевые подключения, поставляющие требуемые уровни QoS. Однако потенциальное использование ECTP не ограничено методами QoS существующей сети. ECTP разработан для использования усовершенствований QoS в ряде различных моделей QoS в будущем.

## 8.2 Контроль QoS

Функция контроля QoS обеспечивает поставщика услуги информацией относительно того, насколько хорошо работает подключение. Чтобы осуществить это, каждый пользователь услуги должен измерить у себя значения параметров и сообщить об этих значениях поставщику услуги.

Для параметров QoS, используемых при подключении, каждый пользователь услуги измеряет у себя значения параметров. Измеренное значение отображается на значение состояния для каждого параметра. Состояние параметра - это целое число, имеющее значение 0, 1, 2 или 3. Это значение состояния записывается в элементе подтверждения и передается поставщику услуги через пакет АСК (см. п. 7.4). Поставщик услуги объединяет значения состояния параметров от всех пользователей услуги.

Цель контроля QoS состоит в том, чтобы обеспечить поставщика услуги информацией относительно состояния QoS для данного подключения. На базе контролируемой информации состояния поставщик услуги может предпринимать любые необходимые действия для поддержания QoS.

### 8.2.1 Генерация АСК

Каждый пользователь услуги сообщает о значениях состояния параметра своему родителю, генерируя пакеты АСК. В ЕСТР-2 правило генерации АСК несколько отличается от правила, определенного в ЕСТР-1. В ЕСТР-2 значение *AGT* установлено равным значению *AGN*. Более точно значение *AGT* установлено равным значению '*AGN* × 1 секунд, т.е. *AGN* секунд'. Соответственно, каждый пользователь услуги генерирует периодические пакеты АСК каждую *AGT* секунду. Установка *AGN* - это вопрос реализации, но в ЕСТР-2 значение *AGN* установлено по умолчанию равным '8'.

Чтобы генерировать пакет АСК, каждый пользователь услуги должен хранить таймер, время контроля QoS (*QMT*), в секундах. *QMT* таймер начинает отсчет, как только пользователь услуги завершает установку подключений, т.е. после приема пакета СС или пакета JS от поставщика услуги. *QMT* таймер нарастает монотонно по мере развития подключения. Таким образом, он не будет обновлен в течение подключения.

Каждый пользователь услуги передает пакет АСК его родителю, если:

$$QMT \% AGN = \text{Дочерний ID} \% AGN$$

Эта схема используется, чтобы свернуть имплозии АСК на родительской стороне в максимально возможной степени. С помощью этого механизма каждый пользователь услуги продолжит генерировать пакеты АСК каждую *AGN* секунду, после того как *QMT* таймер стартует. Например, если *AGN* установлен в 8, то пользователь услуги с дочерним ID, равным 3 или 11, будет генерировать пакеты АСК в *QMT* интервалы, равные 3, 11, 19, 27 секунд и т.д. Как показано в примере первый АСК будет сгенерирован для пакетов данных, переданных только в течение 3 секунд, а не 8 секунд. Однако другие последующие пакеты АСК генерируются для интервала времени *AGN* секунд. Таким способом каждый пользователь услуги генерирует пакеты АСК каждую *AGN* секунду, за исключением первого пакета АСК.

Каждый пакет АСК передает измеренное значение состояния параметра, как описано в п. 8.2.2.

### 8.2.2 Измерение значений параметров QoS

Для групповой передачи данных каждый пользователь услуги измеряет у себя значения каждого параметра QoS. Все значения параметров измеряются, записываются и рассчитываются, пока новый пакет АСК не сгенерирован согласно правилу генерации АСК, описанному выше. Когда приходит время послать пакет АСК, пользователь услуги вычисляет значение состояния параметра для пакетов данных, полученных и собранных до этого момента. После передачи пакета АСК собранные данные стираются, а затем новые данные будут собраны и записаны для генерации нового пакета АСК.

#### 8.2.2.1 Пропускная способность

Пропускная способность определяется как скорость приема данных в единицах октетов в секунду. Скорость приема данных рассчитывается согласно выражению:

$$\text{Количество полученных пакетов данных в октетах в течение } AGN \text{ секунд.}$$

Однако первый пакет АСК может быть сгенерирован прежде, чем пройдет *AGN* секунд. Для измерения значения пропускной способности пользователь услуги должен сохранять информацию о том, сколько пакетов данных (октетов) было получено в течение заданного времени.

Каждый раз по мере получения нового значения пропускной способности значение отображается на значение состояния параметра, равное целому числу 0, 1, 2 или 3, согласно правилу отображения, которое будет описано в п. 8.2.3.

#### 8.2.2.2 Коэффициент потери данных

Коэффициент потери информации представляет коэффициент потери пакета, выраженный в процентах. Коэффициент потери пакета рассчитывается согласно выражению:

$$\text{Число потерянных пакетов относительно Числа пакетов данных, полученных в течение } AGN \text{ секунд.}$$

Первый пакет АСК может быть сгенерирован прежде окончания интервала длиной *AGN* секунд. Для измерения значения пропускной способности пользователь услуги должен сохранять информацию о количестве потерянных пакетов данных. Когда пользователь услуги генерирует пакет АСК, текущее измеренное значение коэффициента потери отображается на *значение состояния параметра*.

### 8.2.2.3 Задержка передачи и ее флуктуации

Для измерения сквозной транзитной задержки поставщик услуги должен передавать пакеты данных с временной меткой. Другое требование - это синхронизация часов между поставщиком услуги и пользователями услуги. Без удовлетворения этих требований сложно получить точную информацию относительно транзитной задержки и флуктуации. В этом подразделе считается, что каждый пользователь услуги может измерять сквозную транзитную задержку пакета данных, направленного поставщиком услуги.

Транзитная задержка измеряется для каждого полученного пакета данных. Эти значения транзитной задержки усреднены по пакетам данных, полученным в течение *AGN* секунд. Флуктуация транзитной задержки измерена как разность между максимальными и минимальными значениями транзитной задержки этих полученных пакетов данных.

В течение *AGN* секунд, как только приходит новый пакет данных, рассчитывается транзитная задержка, а усредненная задержка и значения флуктуации изменяются. Непосредственно перед тем, как пользователь услуги генерирует пакет АСК, текущее измеренное значение будет отображено на значение состояния параметра.

### 8.2.3 Отображение на значение состояния параметра

Измеренное значение параметра отображается на значение состояния параметра для каждого параметра QoS. Последующий пакет АСК будет содержать значение состояния. Снова следует отметить, что измерение параметра QoS активируется только тогда, когда указывается на использование параметра при подключении (см. п. 7.3).

На рисунке 9 показано отображение измеренного значения на значение состояния.

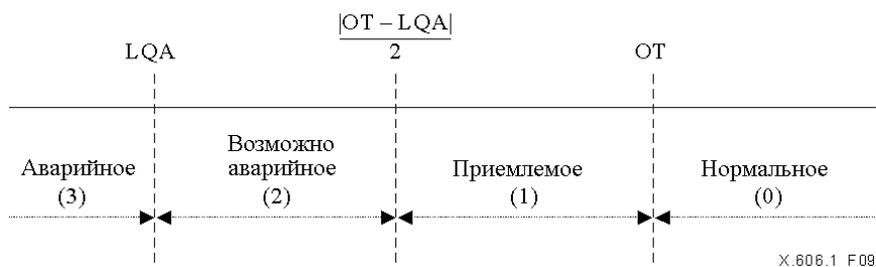


Рисунок 9 – Отображение измеренного значения на значение состояния

Как показано на рисунке, отображение измеренного значения параметра на значение состояния основано на значениях параметров *OT* и *LQA*. ЕСТР также использует пороговое значение, чтобы классифицировать состояние как нормальное (0), приемлемое (1), возможно аварийное (2) и аварийное (3). Пороговое значение установлено в среднее значение между значениями параметров *OT* и *LQA* (т.е.  $|LQA - OT| / 2$ ), как показано на рисунке.

Начальное значение состояния параметра установлено в '0'. Как только получено измеренное значение, отображение на значение состояния параметра выполняется следующим образом:

- ЕСЛИ " измеренное значение параметра > *OT*, " тогда состояние = 0;
- ИЛИ ЕСЛИ "порог < измеренное значение параметра ≤ *OT*," тогда состояние = 1;
- ИЛИ ЕСЛИ "*LQA* < измеренное значение параметра ≤ порог, " тогда состояние = 2;
- ИЛИ ЕСЛИ " измеренное значения ≤ *LQA*," тогда состояние = 3.

В вышеуказанных правилах отображения неравенства верны только для параметра пропускной способности. Для других параметров, таких как задержка, флуктуация и коэффициент потерь, эти неравенства должны быть реверсированы, потому что значения  $OT \leq$  значений *LQA*.

### 8.2.4 Сообщение поставщику услуги

Каждый пользователь услуги сообщает о полученных значениях состояния параметра своему родителю посредством пакетов АСК. Пакеты АСК генерируются на базе *дочернего ID* и в течение *AGN* секунд. Таким образом, вся информация относительно значений состояния параметра будет направлена поставщику услуги согласно иерархическому дереву.

#### 8.2.4.1 Агрегирование по LOв иерархическом дереве

В иерархическом дереве каждый родитель LO объединяет пакеты АСК своих дочерних записей. Это агрегирование проводится непосредственно перед тем, как родитель генерирует свой собственный пакет АСК. LO также генерирует свой собственный пакет АСК каждую *AGN* секунду, как это сделано пользователем услуги.

Родитель просто берет среднее по значениям состояния параметра, сообщенным потомками, вместе с его собственным измеренным значением непосредственно перед тем, как он генерирует пакеты АСК.

Усредненное значение рассчитывается для каждого параметра QoS:

*Суммирование значений состояния параметров отвечающих потомков по CCN.*

В многоуровневой древовидной структуре число потомков дается *Активным Номером Пользователя* услуги (*ARN*), зарегистрированным в списке элементов дерева. В этом случае значения состояния параметра, сообщенные потомками, взвешиваются по их значениям *ARN* следующим образом:

Средневзвешенное значение =  $\sum_{i=1, \dots, CCN} \{ARN(i) \times \text{значение состояния параметра (i)}\}$  по  $\sum_{i=1, \dots, CCN} ARN(i)$

Родитель LO округляет это среднее значение до 0, 1, 2 или 3 для каждого параметра QoS и формирует свой пакет АСК.

#### 8.2.4.2 Агрегирование поставщиком услуги

Агрегирование значений состояния параметра поставщиком услуги происходит так же как по LO. Поставщик услуги также выполняет агрегирование пакетов АСК, полученных от его потомков, через интервал, длительностью *AGN* секунд.

После агрегирования пакетов АСК поставщик услуги просто берет средневзвешенное *ARN* каждого потомка. Таким образом:

*Агрегированное значение состояния = взвешенная сумма сообщенных состояний, взятая по ARN подключения.*

Поставщик услуги получает агрегированное значение состояния для каждого параметра QoS. Более точно поставщик услуги будет иметь следующие агрегированные значения состояния (если каждый из параметров используется при подключении):

- a) агрегированное состояние для пропускной способности, обозначаемое как *Tvalue*;
- b) агрегированное состояние для транзитной задержки, обозначаемое как *Dvalue*;
- c) агрегированное состояние для флуктуации транзитной задержки, обозначаемое как *Jvalue*;
- d) агрегированное состояние коэффициента потери информации, обозначаемое как *Lvalue*.

Каждое из агрегированных значений также находится между 0 и 3.

Поставщик услуги может отправлять контрольную информацию состояния приложению. Контрольная информация полезна для посылающего приложения в части диагностики, качества работы подключения в терминах QoS, что может быть далее полезно для создания модели составления счетов/оплаты.

Контрольная информация также используется для поддержания QoS. Среди контрольных значений *Lvalue* используется для настройки скорости канала передачи (см. п. 8.3.1). Взвешенное значение суммы для всех значений состояния *Tvalue*, *Dvalue*, *Jvalue* и *Lvalue* может использоваться для запуска приостановки подключения, обнаружения источника неисправности и для завершения подключения.

### 8.3 Поддержание QoS

Поддержание QoS выполняется для поддержания качества подключения на желательном уровне и для предотвращения снижения качества подключения ниже согласованного уровня QoS.

Базируясь на контрольных значениях состояния параметра, поставщик услуги будет выполнять следующие операции для поддержания QoS:

- 1) настройка скорости передачи данных;
- 2) приостановка и возобновление подключения;
- 3) обнаружение источника нарушения;
- 4) завершение подключения.

Настройка скорости передачи данных связана со скоростью потока и управлением перегрузкой. Приостановка и возобновление подключения, а также завершение подключения являются операциями, которые могут быть предприняты для управления подключением. Эти события будут объявлены всем пользователям услуги через ND и СТ пакеты, переданные поставщиком услуги.

Для запуска этих действия поддержания QoS поставщик услуги должен конфигурировать следующие пороговые значения:

- 1) *порог\_увеличения\_скорости* и *порог\_уменьшения\_скорости* - для настройки скорости передачи данных;
- 2) *порог\_приостановки\_подключения*.

Все пороговые значения - это числа между 0 и 3.

### 8.3.1 Настройка скорости передачи данных

В ЕСТР используется управление потоком данных на основе окон фиксированного размера. Заданный по умолчанию *размер окна* - тот же самый, что и *размер битового массива АСК (ABS)*, т.е. 32. Поставщик услуги может наиболее быстро передать пакеты данных *размера окна со скоростью передачи данных (DTR)*. В ЕСТР управление перегрузкой осуществляется динамически посредством настройки *DTR* на базе значений состояния коэффициента потери *Lvalue* (см. п. 8.2.4.2).

Настройка скорости передачи данных основана на значениях *порога увеличения скорости* и *порога уменьшения скорости*, которые заданы поставщиком услуги, исходя из требований приложения. Эти значения находятся в следующем соответствии:

$$0 \leq \text{порог\_увеличения\_скорости} \leq \text{порог\_уменьшения\_скорости} \leq 3$$

Значения по умолчанию: для *порога увеличения скорости* = 1,0 для *порога уменьшения скорости* = 2,0.

В фазе передачи данных поставщик услуги начинает с *DTR* = пропускная способность *LQA*, а затем значение *DTR* может быть откорректировано следующим образом:

$$\text{пропускная способность } LQA \leq DTR \leq \text{пропускная способность } CHQ.$$

Каждые *AGN* секунд поставщик услуги настраивает *DTR* на базе значений *порога увеличения скорости*, *порога уменьшения скорости* и контролирует значение *Lvalue* следующим образом:

ЕСЛИ  $Lvalue < \text{порог\_увеличения\_скорости}$

ТОГДА  $DTR = \text{Минимум} \{CHQ, DTR + \text{Увеличение скорости передачи (TRI)}\}$

ИЛИ ЕСЛИ  $\text{порог\_увеличения\_скорости} \leq Lvalue \leq \text{порог\_уменьшения\_скорости}$

ТОГДА *DTR* не меняется

ИЛИ ЕСЛИ  $Lvalue > \text{порог\_уменьшения\_скорости}$

ТОГДА  $DTR = \text{Max} \{LQA, DTR - \text{Уменьшение скорости передачи (TRD)}\}$

Переменные настройки скорости, такие как *TRI* и *TRD*, могут быть установлены на базе значений пропускной способности *CHQ* и *LQA*. Например,

$$TRI = (CHQ - LQA) \times 1/20$$

$$TRD = (CHQ - LQA) \times 1/5$$

### 8.3.2 Приостановка и возобновление соединения

Приостановка подключения может быть выполнена поставщиком услуги, чтобы временно остановить групповые передачи данных для предотвращения ухудшения качества подключения.

Приостановка подключения и его возобновление могут быть выполнены согласно запросу приложения. В этом случае посылающее приложение инициирует приостановку подключения на основе контролируемых значений состояния параметров, таких как *Tvalue*, *Dvalue*, *Jvalue* и *Lvalue*. Если приостановка подключения инициирована, то поставщик услуги передает периодические пакеты *ND* с битом *F*, установленным в '1' в заголовке (см. Рекомендацию МСЭ-Т X.606 | ИСО/МЭК 14476-1). Поставщик услуги не должен передавать никакого нового пакета *DT*, в то время когда могут быть посланы управляющие пакеты, включая пакеты *NB*. Каждый пользователь услуги может также посылать управляющие пакеты *АСК*.

Приостановка подключения может также быть инициирована на основе заданного *порога приостановки подключения*. В этом случае, если контролируемое значение состояния подключения больше, чем *порог приостановки подключения*, тогда ЕСТР поставщик услуги инициирует приостановку подключения. Заданное значение *порога приостановки подключения* равно 2,5.

Для этой цели значение *Состояния подключения* рассчитывается для всех контролируемых значений состояния параметра следующим образом:

$$\text{Состояние подключения} = Tweight \times Tvalue + Dweight \times Dvalue + Jweight \times Jvalue + Lweight \times Lvalue.$$

Каждое из значений веса также должно быть задано наряду со значением *порога приостановки подключения*, при этом налагаются следующие ограничения:

$$0 \leq Tweight, Dweight, Jweight, Lweight \leq 1$$

$$Tweight + Dweight + Jweight + Lweight = 1$$

Значение веса установлено в '0', если соответствующий параметр *QoS* не доступен при подключении.

Приостановка подключения инициируется, если:

$$\text{Состояние подключения} \geq \text{порог\_приостановки\_подключения}$$

После того как приостановка подключения была обозначена и если *время приостановки подключения (CPT)* истекло, то инициируется возобновление подключения и поставщик услуги начинает передавать групповые данные при скорости передачи *LQA*. Если возобновление подключения обозначено, то пакеты ND установят бит *F* заголовка в значение '0'.

### 8.3.3 Отключение источника неисправности

Поставщик услуги или LO могут инициировать отключение источника неисправности, чтобы поддержать состояние QoS на желательном уровне, а также предотвратить ухудшение состояния подключения. Детальная схема отключения источника неисправности может быть реализована различными способами на основе значений состояния параметров, описанных в настоящей спецификации.

Например, пользователь услуги может быть отключен его родителем, если он сообщил о *значении состояния параметра*, превышающем значение *порог\_приостановки\_подключения* большее число раз, чем это предусмотрено заданным порогом. Проектирование и реализация схемы отключения источника неисправности должны быть выполнены с учетом всех деталей, так как операция отключения может иметь существенное воздействие на весь режим протокола ECTP.

### 8.3.4 Завершение подключения

Естественная опция для завершения подключения состоит в завершении подключения, когда все групповые данные были переданы. В операциях управления QoS завершение подключения также инициируется, если состояние подключения воспринято как 'невосстанавливаемое'.

Завершение подключения может быть выполнено согласно запросу приложения. Если завершение подключения инициировано, то поставщик услуги передает пакет СТ всем пользователям услуги и завершает подключение.

Завершение подключения может также быть инициировано на базе заданного *времени завершения подключения (CTT)*. В этом случае завершение подключения инициируется если:

Последующая приостановка подключения происходит в пределах *CTT* от возобновления подключения.

*CTT* таймер активируется когда обозначено возобновление подключения. Завершение подключения может не поддерживаться некоторыми приложениями.

## 9 Таймеры и переменные

Далее описаны таймеры и переменные, используемые для управления QoS.

### 9.1 Таймеры

- a) *Время генерации ACK (AGT)* в секундах: Каждый пользователь услуги периодически генерирует пакет ACK каждые *AGT* секунд, за исключением первого пакета ACK. В ECTP-2 значение *AGT* установлено равным *AGN* секунд (см. п. 8.2).
- b) *Время приостановки подключения (CPT)* в секундах: Как только приостановка подключения обозначена, подключение делает паузу в течение интервала *CPT* (см. п. 8.3.2).
- c) *Время завершения подключения (CTT)* в секундах: После того как возобновление подключения обозначено и если приостановка подключения происходит снова в пределах *CTT* интервала, то инициируется завершение подключения (см. п. 8.3.4).
- d) *Время контроля QoS (QMT)* в секундах: Каждый пользователь услуги измеряет у себя значения параметров QoS каждый интервал *QMT* (см. п. 8.2.2).

### 9.2 Переменные операций

- a) *Агрегированные значения состояния параметров*: Поставщик услуги объединяет значения состояния параметров, сообщенные пользователями услуги, что дает значения: *Tvalue*, *Dvalue*, *Jvalue* и *Lvalue* (см. п. 8.2).
- b) *Номер генерации ACK (AGN)*: Каждый пользователь услуги периодически генерирует пакет ACK каждые *AGT* секунд. В ECTP-2 значение *AGT* установлено равным *AGN* секунд (см. п. 8.2). По умолчанию значение *AGN* установлено равным 8.
- c) *Состояние подключения*: Для запуска приостановки/возобновления подключения все агрегированные значения состояния параметра могут быть усреднены с заданными весами параметров, такими как *Tweight*, *Dweight*, *Jweight* и *Lweight*. Это приводит к *состоянию подключения*, которое представляет полное состояние подключения (см. п. 8.3.2).
- d) *Скорость передачи данных (DTR)*: Поставщик услуги передает групповые данные со скоростью *DTR* (см. п. 8.3.1).
- e) *Измеренное значение параметра*: Каждый пользователь услуги должен измерять у себя значение параметров QoS, используемых при подключении. Это дает измеренное значение параметра (см. п. 8.2).

- f) Значение состояния параметра: *Измеренное значение параметра* отображается на *значение состояния параметра*, которое является целым числом 0, 1, 2 или 3 (см.п. 8.2).
- g) Уменьшение скорости передачи (*TRD*): Значение *DTR* уменьшено на *TRD* (см. п. 8.3.1).
- h) Увеличение скорости передачи (*TRI*): Значение *DTR* уменьшено на *TRI* (см. п. 8.3.1).
- i) *Порог\_увеличения\_скорости*: Пороговое значение для увеличения скорости передачи данных (см. п. 8.3.1).
- j) *Порог\_уменьшения\_скорости*: Пороговое значение для уменьшения скорости передачи данных (см. п. 8.3.1).
- k) *Порог\_приостановки\_подключения*: Пороговое значение для инициации приостановки подключения (см. п. 8.3.2).

## Приложение А

### Межсетевое взаимодействие между ЕСТР и RSVP для резервирования ресурса

(Настоящее приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

В ЕСТР QoS можно гарантировать межсетевым взаимодействием с RSVP/Intserv для сетевых протоколов резервирования ресурса. Базируясь на значениях параметров QoS, RSVP может конфигурировать свои дескрипторы трафика, если это допускается в сети.

В настоящем приложении описывается как использовать сигналы RSVP наряду с ЕСТР для резервирования ресурса и как отобразить параметры QoS в ЕСТР для RSVP дескрипторов трафика. Все схемы, показанные ниже, предполагают режим не-согласования (см. п.п. 7.1 и 8.1), в котором значения параметров QoS, предложенные поставщиком услуги, предписаны для всех пользователей услуги без согласования. Использование RSVP в режиме согласования является вопросом дальнейшего изучения. Однако даже в режиме согласования параметр, отображающий отношения между RSVP и ЕСТР, все еще допустим. В этом случае каждое из 'согласованных' ЕСТР значений параметров QoS будет отображено на RSVP дескрипторы трафика.

#### А.1 ЕСТР параметры QoS

ЕСТР параметры QoS и их заданные значения в итоге определены следующим образом:

- a) пропускная способность: CHQ, OT и LQA;
- b) транзитная задержка: OT и LQA;
- c) флуктуация транзитной задержки: OT и LQA;
- d) коэффициент потери информации: OT и LQA.

В заключение, по крайней мере с точки зрения современной RSVP технологии, транзитная задержка, флуктуация задержки и коэффициент потери не могут быть явно поддержаны RSVP. RSVP только обеспечивает строгую гарантию требования к задержке на сетевом уровне, используя резервирование пропускной способности и работая с планировщиком трафика в промежуточных маршрутизаторах пути. Термин 'задержка', определенный в RSVP (задержка очереди сетевого уровня), весьма отличается от определения в ЕСТР (сквозная транзитная задержка). Следующий подраздел дает краткий обзор операций RSVP.

#### А.2 Обзор RSVP

RSVP (IETF RFC 2205, 2210, 2212) включает следующие объекты RSVP:

- a) RSVP SENDER\_TSPEC;
- b) RSVP ADSPEC;
- c) RSVP RECEIVER\_FLOWSPEC.

##### А.2.1 RSVP SENDER\_TSPEC

Технические требования трафика (TSPEC) RSVP - это объект, который содержит дескрипторы трафика, сгенерированные поставщиком услуги в пределах сеанса RSVP. Этот объект включает следующие дескрипторы трафика:

- a) спецификации маркеров памяти  $r$  и  $b$ , где  $r$  - маркер генерации или скорость утечки, а  $b$  - объем области памяти;
- b) пиковая скорость  $p$ ;
- c) минимальная сохраняемая единица  $m$ ;
- d) максимальный размер пакета  $M$ .

$R$  и  $p$  измеряются в октетах в секунду, а  $b$ ,  $m$  и  $M$  измерен в октетах. Скорости  $r$  и  $p$  - это средняя и максимальная скорости потока информации, соответственно, а  $b$  - параметр, который ограничивает изменчивость генерации трафика.

Другие параметры обеспечивают границы распределения длины пакета в пределах потока.  $M$  - это максимальный размер соответствующего пакета, а  $m$  - это такой размер, что любой пакет, с размером, меньшим  $m$ , обрабатывается контроллером сети, как будто он имеет размер, равный  $m$ .  $M$  и  $m$  непосредственно не связаны с ЕСТР.

### A.2.2 RSVP ADSPEC

Спецификация извещения (ADSPEC) RSVP - это объект, который содержит информацию, сгенерированную или в источниках данных, или в промежуточных элементах сети. RSVP ADSEPC передается следующим пользователям услуги, а также может использоваться и модифицироваться внутри сети перед доставкой принимающему приложению. Эта информация включает как параметры, описывающие свойства информационного канала, включая доступность определенных служб управления QoS, так и параметры, требуемые конкретными службами управления QoS для их правильной работы. Этот объект включает следующие дескрипторы трафика:

- a) Пропускная способность соединения с минимальной доступной пропускной способностью на сквозном пути.
- b) Сквозные задержки кроме задержки очереди; эти задержки могут быть получены измерением двух параметров ошибки *C* и *D*. Параметр ошибки *C* - это параметр ошибки, зависящий от скорости. Его величина дает задержку пакета в потоке, которая может быть обусловлена параметрами скорости потока. Параметр ошибки *D* не зависит от скорости, это параметр ошибки элемента. Его величина представляет собой наихудший случай изменения времени передачи через служебный элемент, независимо от скорости потока. Он обычно определяется или устанавливается при загрузке или во время конфигурации.

### A.2.3 RSVP FLOWSPEC

Этот объект содержит запрос на резервирование информации, сгенерированный пользователями услуги, и состоит из объектов RECEIVER\_TSPEC и RECEIVER\_RSPEC. Информация в FLOWSPEC направляется от приемников данных к источникам данных. Этот объект может использоваться или модифицироваться в промежуточных элементах сети до получения посылающего приложения.

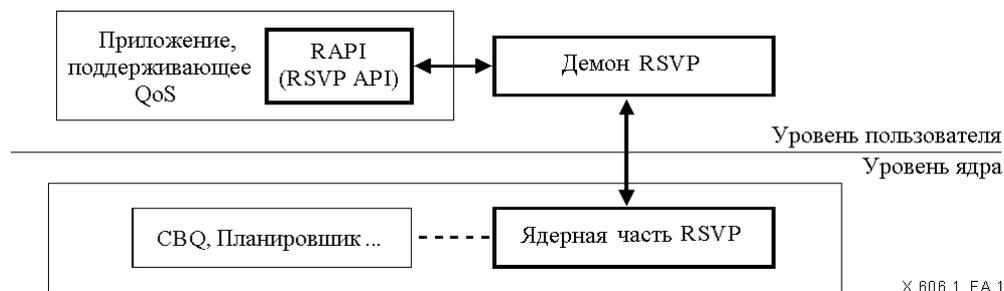
RECEIVER\_TSPEC идентичен SENDER\_TSPEC, кроме MTU сквозного пути. С другой стороны, RECEIVER\_RSPEC включает следующие параметры:

- a) *R*: зарезервированная пропускная способность;
- b) *S*: параметр резерва времени.

Значение *R* должно быть больше или равно значению *r* объекта TSPEC. Параметр резерва времени *S* определяется в микросекундах. Параметр резерва времени показывает разность между желательной задержкой и задержкой, полученной при использовании уровня резервирования *R*. Данный параметр резерва времени может использоваться элементом сети, чтобы уменьшить его резервирование ресурса для соответствующего потока.

### A.2.4 RSVP API

На рисунке A.1 показана структура реализации RSVP в системе.



X.606.1\_FA.1

**Рисунок А.1 – Структура реализации RSVP**

В системе RSVP работает с тремя модулями: ядро RSVP, демон RSVPи (РАPI) RSVP. Ядро RSVP управляет организацией очереди данных и планированием согласно требованиям более высокого уровня RSVP. Демон RSVP обрабатывает передачу сигналов между РАPI и ядром RSVP. РАPI обеспечивает интерфейс с приложениями, воспринимающими параметры QoS, что гарантирует поддержку операциями RSVP требований QoS со стороны приложений. ЕСТР характеристики QoS будут интегрированы и предписаны к выполнению в сетях с поддержкой RSVP через РАPI.

Функции РАPI могут быть классифицированы на две группы: функции, называемые приложением, и асинхронные подпрограммы, поставленные приложению демоном RSVP.

Основные функции RSVP следующие:

- `rapi_session ()`, которая используется, чтобы установить RSVP сеанс;
- `rapi_sender ()`, которая используется поставщиком услуги, чтобы послать RSVP сообщения PATH;
- `rapi_reserve ()`, которая используется для резервирования приемником ресурсов сети;

- `gapi_release ()`, которая используется, чтобы закончить RSVP сеанс;
- `gapi_getfd ()` и `gapi_dispatch ()`, которые используются приложением, чтобы получить информацию состояния о RSVP через асинхронные подпрограммы.

### А.3 Пример отображения параметра между RSVP и ЕСТР

В таблице А.1 приведен пример отображения параметра между RSVP и ЕСТР, который может использоваться для поддержки QoS наряду с RSVP.

ПРИМЕЧАНИЕ - Другое отображение также может использоваться в зависимости от реализации.

Таблица А.1 – Отображение параметра от ЕСТР на RSVP

Параметры RSVP	ЕСТР параметр QoS	Описание
$p$ (TSPEC)	Пропускная способность CHQ	Прямое отображение на $p$
$r$ (TSPEC)	Пропускная способность OT	Прямое отображение на $r$
$b$ (TSPEC)	(пропускная способность CHQ – пропускная способность OT) $\times$ произвольный интервал (1 ~ 3 секунды)	Размер маркированного объема памяти устанавливается согласованно на базе параметров пропускной способности ЕСТР
$m$ (TSPEC)	IP+UDP+ЕСТР заданный заголовок (20 + 8 + 16 = 44 октета)	Прямое отображение
$M$ (TSPEC)	ЕСТР MSS (по умолчанию 1024 октета)	Прямое отображение
$R$ (RSPEC)	(пропускная способность CHQ + пропускная способность OT) / 2	Резервирование установлено как среднее значение пропускных способностей CHQ и OT
$S$ (RSPEC)	0	Параметр резерва времени не устанавливается

Когда демон ЕСТР вызывает демона RSVP, ЕСТР параметры QoS отображаются на дескрипторы трафика RSVP TSPEC, такие как  $p$ ,  $r$ ,  $b$ ,  $m$ ,  $M$ , как показано в таблице. Эти значения параметра TSPEC передаются поставщиком услуги пользователям услуги через RSVP сообщение PATH.

По прибытии RSVP сообщения PATH в приемник, информация в объектах SENDER\_TSPEC и ADSPEC будет передаваться через RSVP API к приложению ЕСТР. Приложение интерпретирует получаемую информацию и использует ее, чтобы выбрать параметры резервирования ресурса. Эти параметры встраиваются в объект RSVP FLOWSPEC и будут переданы поставщику услуги через RSVP RESV сообщения. Резервирование пропускной способности  $R$  и параметр резервирования времени  $S$ , которые содержатся в объекте RSVP RSPEC, будут отображены с использованием параметров ЕСТР, как показано в таблице А.1.

### А.4 Сценарий межсетевое взаимодействие между ЕСТР и RSVP

Заданные значения для каждого ЕСТР параметра QoS могут использоваться для создания объекта RSVP SENDER\_TSPEC. Для этого требуется такой интерфейс между ЕСТР и RSVP, чтобы ЕСТР значения параметра QoS были значимы для процессора RSVP. Объект SENDER\_TSPEC будет направлен пользователям услуги через RSVP сообщения PATH.

Время, когда ЕСТР вызывает RSVP, зависит от реализации, но рекомендуется, чтобы ЕСТР поставщик услуги вызвал RSVP после отправки пакета CR и чтобы пользователь услуги работал после приема пакета CR. На рисунке А.2 показана модель межсетевое взаимодействие между ЕСТР и RSVP.

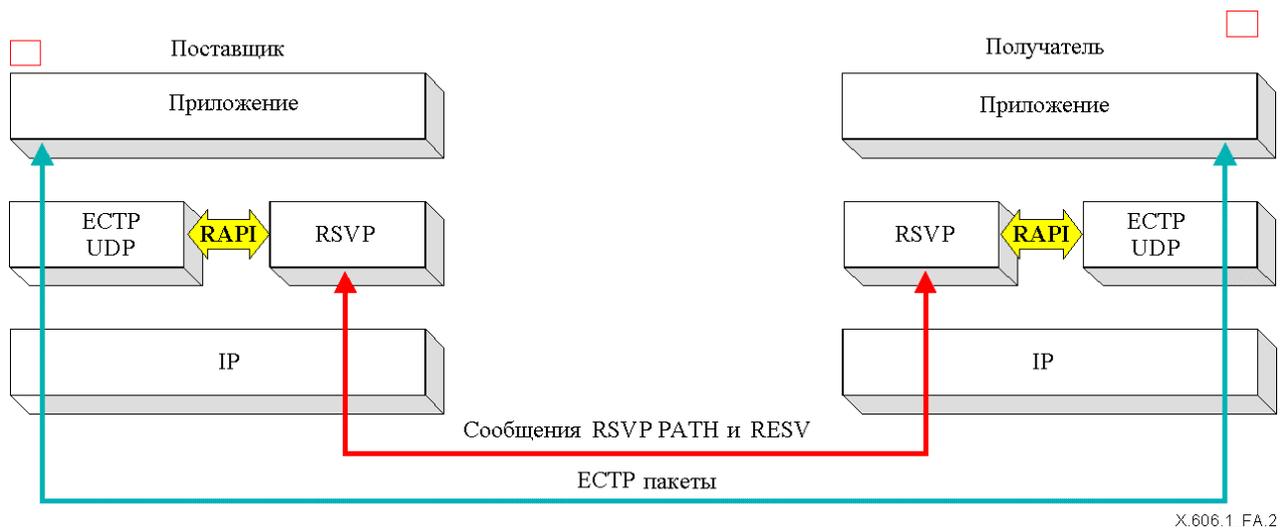


Рисунок А.2 – Модель межсетевое взаимодействия между ECTP и RSVP.

Как показано на рисунке связь между ECTP и RSVP осуществляется через RSVP API (RAPI). ECTP поставщик услуги передает информацию о параметрах QoS для RSVP через RAPI, когда вызывает демона RSVP. Если RSVP уже запущен ECTP, то RSVP связь будет установлена между посылающим демоном RSVP и принимающим демоном RSVP. Таким образом, равноправные отношения RSVP будут установлены между поставщиком услуги и пользователями услуги.

В течение ECTP подключения каждый модуль RSVP сообщает о состоянии сетевого резервирования ресурса связанному с ним демону ECTP через RAPI. Исходя из сообщения о состоянии, можно определить было ли резервирование сделано успешно или произошла аварийная ситуация. Набор примеров кодов состояния RSVP:

- Код Состояния 0: события PATH обозначены (используется принимаемым RSVP).
- Код Состояния 1: события ошибки PATH обозначены (используется посылающим RSVP).
- Код Состояния 2: события RESV обозначены (используется посылающим RSVP).
- Код Состояния 3: события ошибки RESV обозначены (используется принимаемым RSVP).
- Код Состояния 4: события RESV\_CONFIRM обозначены (используется принимаемым RSVP).

Ядро протокола ECTP может получить эти коды состояния, используя асинхронные функции RSVP от демона RSVP.

Процедуры, выполненные модулями ECTP и RSVP для межсетевое взаимодействия между ними, могут быть получены в итоге следующим образом:

- (1) ECTP поставщик услуги передает пакет CR пользователям услуги. Пакет CR содержит элемент QoS, указывающий на значения параметра QoS, требуемые приложением.
- (2) ECTP поставщик услуги вызывает связанного с ним демона RSVP.
- (3) RSVP поставщик услуги создает сообщение PATH, которое содержит параметры TSPEC, основанные на ECTP параметрах QoS.
- (4) RSVP поставщик услуги периодически передает сообщения PATH пользователям услуги.
- (5) Если обозначена RSVP ошибка PATH, то RSVP поставщик услуги сообщает о соответствующем коде состояния ECTP поставщику услуги.
- (6) Как только ECTP пользователь услуги принимает пакет CR, то вызывается связанный с ним демон RSVP пользователя услуги.
- (7) На базе сообщения PATH, полученного от поставщика услуги, RSVP пользователь услуги организует прием передачи RESV сообщения, включая параметры RSPEC. Часть информации относительно RSPEC, такой как *R* (пропускная способность) и *S* (параметр резерва времени), может быть конфигурирована при взаимодействии RSVP пользователя услуги и ECTP пользователя услуги.
- (8) RSVP пользователь услуги передает соответствующие RESV сообщения поставщику услуги.
- (9) Когда обозначена RSVP RESV ошибка, тогда RSVP пользователь услуги сообщает о соответствующем коде состояния ECTP пользователю услуги.

Если ECTP принимает код состояния от связанного с ним RSVP, то он может информировать об этом его приложение. Конкретное использование кодов состояния зависит от реализации.

На рисунке А.3 показаны потоки сообщений ЕСТР и RSVP между поставщиком услуги и пользователями услуги. После передачи пакета НВ, ЕСТР поставщик услуги может вызывать начальное RSVP сообщение PATH. Соответствующие RSVP RESV сообщения поступают от ЕСТР пользователей услуги. Последующие RSVP сообщения PATH будут инициированы и будут повторяться наряду с периодическими ЕСТР сообщениями с пакетами НВ.

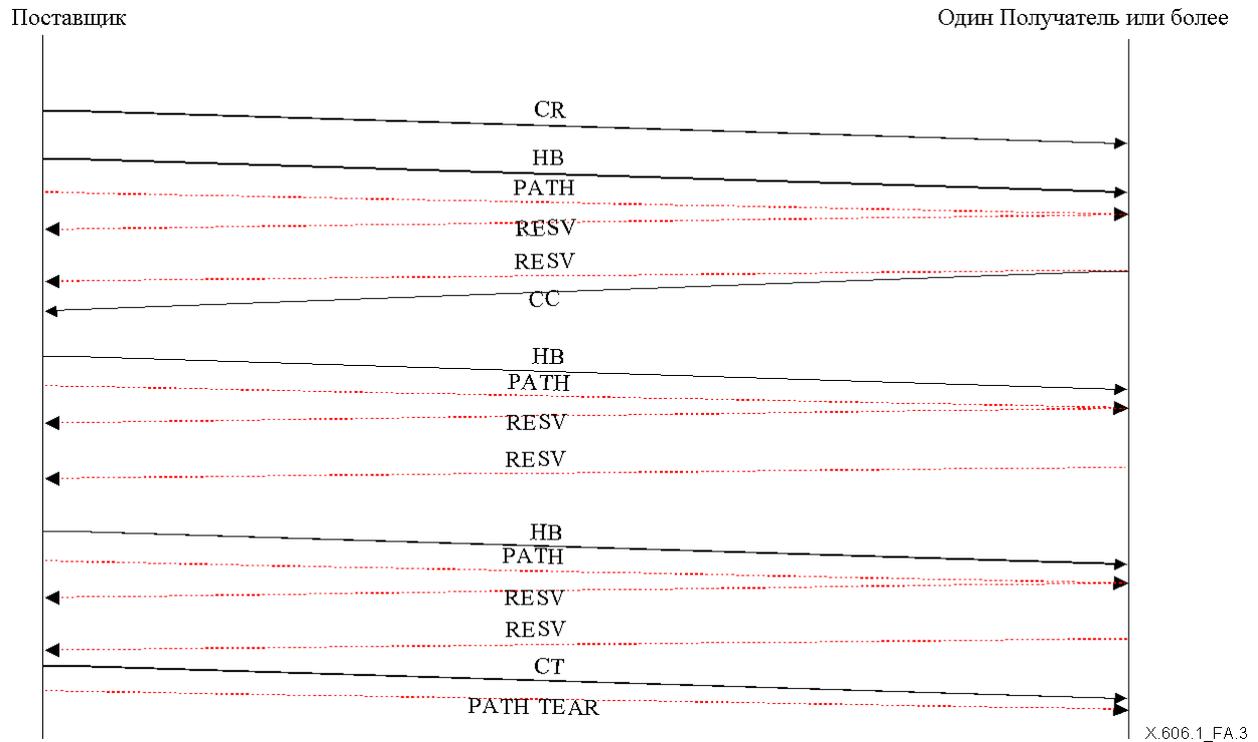


Рисунок А.3 – Модель межсетевое взаимодействия между ЕСТР и RSVP

## Приложение В

### Прикладные программные интерфейсы

(Настоящее приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

В настоящем приложении определяются прикладные программные интерфейсы (API) для ЕСТР. API ЕСТР, описанный в настоящей Рекомендации | Международном стандарте, может применяться приложениями, которые используют транспортные возможности ЕСТР часть 1 (Рекомендация МСЭ-Т X.606 | ИСО/МЭК 14476-1) и ЕСТР часть 2 (Рекомендация МСЭ-Т X.606.1 | ИСО/МЭК 14476-2).

Этот API был разработан на базе функций API сокета Беркли. Однако, чтобы отличать API ЕСТР от существующих функций сокета Беркли, функциям API ЕСТР присваиваются имена с префиксом 'm' (например, msocket).

#### В.1 Краткий обзор

##### В.1.1 Функции API

В таблице В.1 приведены функции API, используемые в ЕСТР.

Таблица В.1 – Функции API ЕСТР

Наименование функции	Описание
msocket ()	Создает новый групповой сокет в ЕСТР домене связи.
Mbind ()	Сопоставляет набор местных и адреса / порты группы с сокетом.
Mconnect ()	Поставщик услуги инициализирует создание подключения к указанному внешнему адресу. Пользователь услуги, который присоединяется позже, инициализирует процесс присоединения.
maccept ()	Предполагаемые пользователи услуги присоединяются к ЕСТР подключению, принимая сигнал создания подключения от поставщика услуги.
Msend ()	Посылает данные прикладной программы группе адресатов.
Mrecv ()	Поставляет полученные данные приложению. Передает некоторые индикативные сообщения управления приложению в течение фазы передачи данных.
mclose ()	Заканчивает подключение и освобождает сокет.
mgetsockopt ()	Получает сокет и параметры протокола от ядра.
msetsockopt ()	Устанавливает сокет и параметры протокола для ядра.

##### В.1.2 Использование функций API ЕСТР

На рисунке В.1 показан пример использования функций API ЕСТР. Последовательности вызванных функций API показываются в терминах поставщика услуги, пользователя услуги, а также пользователя услуги, присоединяющегося позднее.

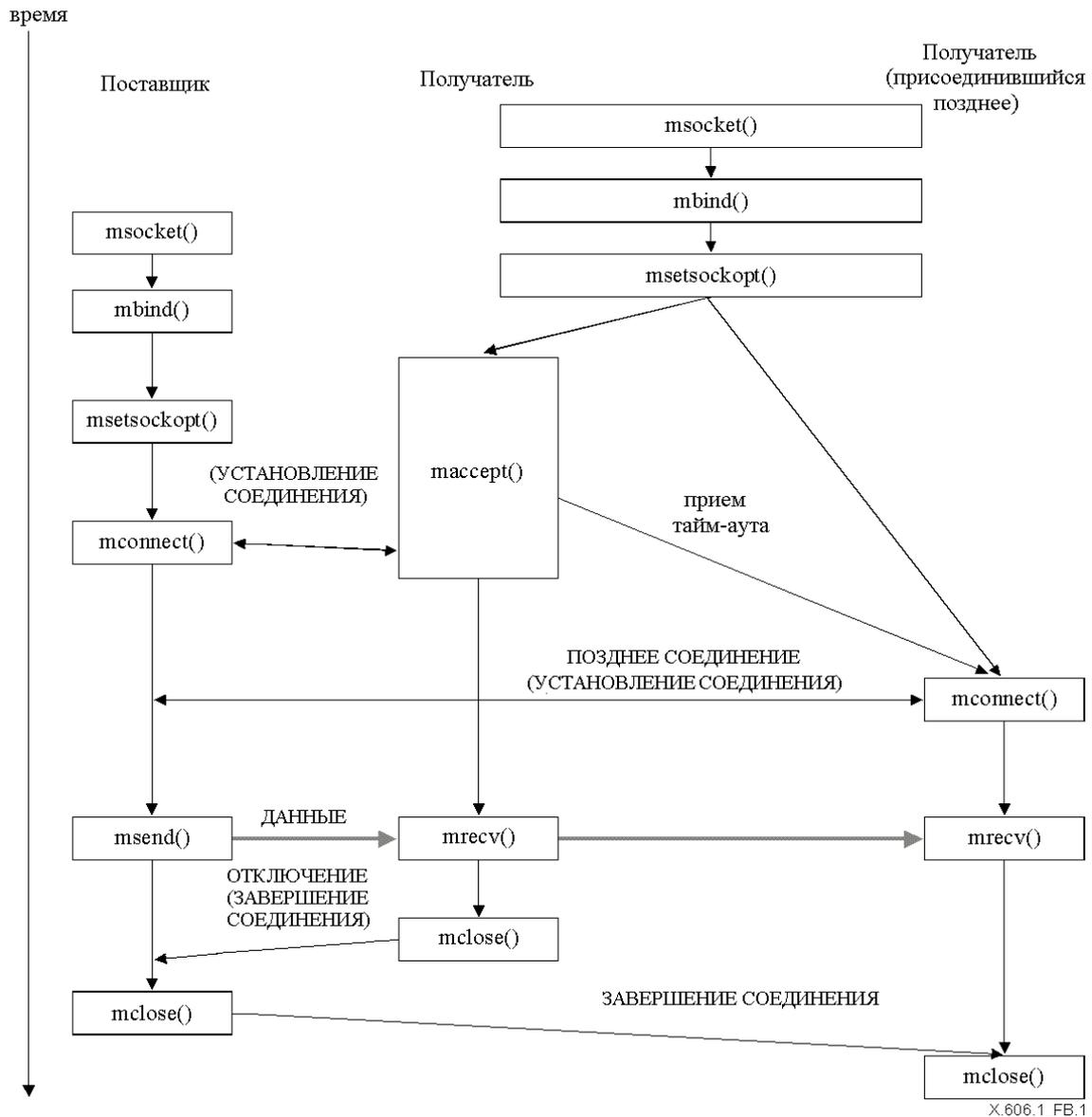


Рисунок В.1 – Использование API ЕСТР

Как показано на рисунке, пользователь услуги выполняет различные операции в соответствии с типом его узла: раннее присоединение или позднее присоединение. Пользователь с ранним присоединением входит в режим `maccept` после `msetsockopt`, в то время как пользователь с поздним присоединением вызывает функцию `mconnect` непосредственно. Пользователь с ранним присоединением может войти в режим `mconnect` после заданной продолжительности блокировки времени приема, если он не может принимать любой сигнал индикации создания подключения.

## В.2 Функции API ЕСТР

### В.2.1 `msocket()`

Для использования ЕСТР приложение ДОЛЖНО сначала вызывать функцию `msocket`, которая определяет тип протокола связи желательного типа, такого как ЕСТР с использованием IPv4, ЕСТР с использованием IPv6, ЕСТР с использованием протоколов ВОС и т.д.

```
int msocket(int family, int type, int protocol);
```

Описание параметров:

- *family*: определяет семейство протокола и равен одной из констант, приведенных в таблице В.2;
- *type*: определяет тип сокета и равен одной из констант, приведенных в таблице В.3;
- *protocol*: установлен в 0.

Таблица В.2 – Константы протокола *family*, используемые в функции *msocket*

family	Описание
AF_INET	Протоколы IPv4
AF_INET6	Протоколы IPv6
AF_ISO	Протоколы домена ВОС

Таблица В.3 – Константы сокета *type*, используемые в функции *msocket*

type	Описание
SOCK_ESTRP1	сокет для ESTRP часть 1
SOCK_ESTRP2	сокет для ESTRP часть 2

В таблице В.4 показаны допустимые комбинации, наряду с фактическим протоколом, который задан парой констант.

Таблица В.4 – Комбинации констант *family* и *type* для функции *msocket*

type	AF_INET	AF_INET6	AF_ISO
SOCK_ESTRP1	ESTRP1	ESTRP1	ESTRP1
SOCK_ESTRP2	ESTRP2	ESTRP2	ESTRP2

Запрос *msocket* возвращает неотрицательный дескриптор в случае успеха или -1 в следующих случаях, перечисленных в таблице В.5.

Таблица В.5 – Коды ошибок вызова *msocket*

Код ошибки	Описание
EPROTONOSUPPORT	Тип протокола или указанный протокол не поддерживаются в пределах этого домена.
EMFILE	Дескрипторная таблица процесса заполнена.
ENFILE	Таблица системного файла заполнена.
EACCESS	Разрешение создавать сокет указанного типа и/или протокола отклонено.
ENOBUFS	Недостаточный доступный размер буфера.

### В.2.2 mbind()

Функция *mbind* назначает сокету набор локальных, групповых адресов, адреса управления и роль узла в сеансе. Для протокола Интернета адрес протокола - это комбинация или 32-разрядного адреса IPv4, или 128-разрядного адреса IPv6, наряду с 16-разрядным номером порта.

```
int mbind(int msockfd, const struct sockaddr *laddr, socklen_t laddrlen, const struct
sockaddr *gaddr, socklen_t gaddrlen, struct sockaddr *caddr, socklen_t caddrlen, int role);
```

Описание параметров:

- *msockfd*: дескриптор сокета, который был возвращен функцией *msocket*;
- *laddr*: указатель на абсолютный протокольный адрес, чтобы связать локальный адрес с вышеупомянутым сокетом;
- *laddrlen*: размер вышеупомянутой адресной структуры;
- *gaddr*: указатель на абсолютный протокольный адрес, чтобы связать адрес целевой группы с сокетом;
- *gaddrlen*: размер структуры адреса группы;
- *caddr*: указатель на абсолютный протокольный адрес, чтобы связать адрес управления с сокетом;
- *caddrlen*: размер структуры адреса управления; и
- *role*: определяет роль этого инициатора запроса, например TO, LO или LE

Таблица В.6 – Значения *role* сокета для использования в функции *mbind*

<b>role</b>	<b>Описание</b>
TO	Создатель подключения и поставщик услуги как владелец в ЕСТР связи.
LO	Пользователь услуги, ответственный за повторные передачи в иерархическом дереве.
LE	Пользователь услуги, который не обозначен как LO.

Приложение может посредством *mbind* связать конкретный IP адрес и сетевой адрес группы со своим сокетом. Адреса источника и группы должны принадлежать интерфейсу узлового компьютера.

Вызов *mbind* возвращает нуль в случае успеха или -1 в следующих случаях, перечисленных в таблице В.7.

Таблица В.7 – Коды ошибок вызова *mbind*

<b>Код ошибки</b>	<b>Описание</b>
EAGAIN	Ресурсы ядра, необходимые для завершения запроса, временно недоступны.
EBADF	<i>msockfd</i> - не допустимый дескриптор.
ENOTSOCK	<i>msockfd</i> - не сокет.
EADDRNOTAVAIL	Указанный адрес не доступен для локального компьютера.
EADDRINUSE	Указанный адрес уже используется.
EACCES	Требуемый адрес защищен, а текущий пользователь не имеет разрешения для обращения к нему.
EFAULT	Адресный параметр не находится в допустимой части адресного пространства пользователя.
EROLE	Требуемая роль не допустима.
ПРИМЕЧАНИЕ - Выделенный код ошибки определен для ЕСТР недавно.	

### В.2.3 *maccept()*

Только пассивный элемент сеанса, подобный LE или LO, может вызывать эту функцию. Она может ожидать инициирования поставщика услуги для заданной блокировки времени через *msetsockopt* и сообщает, установлено ли групповое подключение или нет.

```
int maccept(int msockfd, struct sockaddr *raddr, socklen_t *raddrlen);
```

Описание параметров:

- *msockfd*: дескриптор сокета, который был возвращен функцией *msocket*;
- *raddr*: адрес протокола отдаленного инициатора подключения (поставщик услуги или TO); и
- *raddrlen*: указатель на размер структуры адреса сокета, указываемой значением *raddr*.

Если функция *maccept* выполнена успешно, то она возвращает значение первого параметра, *msockfd*. После этого данное возвращаемое значение называется дескриптором *присоединенного сокета*.

Запрос *maccept* возвращает неотрицательный дескриптор в случае успеха или -1 в следующих случаях, перечисленных в таблице В.8.

Таблица В.8 – Коды ошибок вызова `massept`

Код ошибки	Описание
EBAADF	Дескриптор недопустим.
EINTR	Операция <code>massept</code> была прервана.
EMFILE	Дескрипторная таблица процесса заполнена.
ENFILE	Таблица системного файла заполнена.
ENOTSOCK	Дескриптор ссылается на файл, а не на сокет.
EFAULT	Параметр <code>addr</code> находится не в перезаписываемой части адресного пространства пользователя.
EWOULDBLOCK	Сокет отмечен как не блокируемый, и отсутствуют подключения для приема.
ECONNABORTED	Подключение открыто, но оно затем было закрыто при ожидании слушающей очереди.
ECRTIMEOUT	Указывает, что время ожидания CR истекло.
ПРИМЕЧАНИЕ - Выделенный код ошибки определен для ЕСТР недавно.	

### В.2.4 `mconnect()`

Функция `mconnect` используется ТО или поздно соединяющимся LE для подключения.

```
int mconnect(int msockfd, const struct sockaddr *daddr, socklen_t daddrlen);
```

Описание параметров:

- `msockfd`: дескриптор сокета, который был возвращен функцией `msocket`;
- `daddr`: указатель на абсолютный протокольный адрес назначения. Этот адрес может быть адресом группы (в случае, если вызывающий инициатор является ТО) или адресом поставщика услуги (в случае, если это вызывающий инициатор - LO или LE);
- `daddrlen`: размер `daddr`.

Вызов `mconnect` возвращает нуль в случае успеха или -1 в следующих случаях, перечисленных в таблице В.9.

Таблица В.9 – Коды ошибок функции `mconnect`

Код ошибки	Описание
EBAADF	<code>msockfd</code> - недопустимый дескриптор.
ENOTSOCK	<code>msockfd</code> - дескриптор файла, а не сокета.
EADDRNOTAVAIL	Указанный адрес не доступен для этого компьютера.
EAFNOSUPPORT	Адреса в указанном адресном семействе не могут использоваться с этим сокетом.
EISCONN	Сокет уже присоединен.
ECONNREFUSED	Попытка соединения была полностью отклонена.
ENETUNREACH	Сеть недоступна для этого компьютера узла.
EADDRINUSE	Адрес уже используется.
EFAULT	Параметр имени определяет область вне адресного пространства процесса.
EALREADY	Сокет неблокируемый, а предыдущая попытка подключения еще не была закончена.
EDENIED	Указывает, что ТО отклонил запрос присоединения со стороны LE или LO.
ETIMEOUT	Время установления подключения истекло без установления подключения. Указывает, что от ТО нет ответа.
ПРИМЕЧАНИЕ - Выделенные коды ошибок определены для ЕСТР недавно.	

### В.2.5 `msend()`

Эта функция `msend` записывает данные из буфера в присоединенный сокет.

```
ssize_t msend (int msockfd, const void *buf, size_t buflen, int *flags);
```

Описание параметров:

- *msockfd*: дескриптор сокета, который был возвращен функцией `msocket`;
- *buf*: указатель на буфер, из которого записываются данные;
- *buflen*: размер *buf*; и
- *flags*: еще не определены.

Функция `msend` возвращает число записанных октетов в случае успеха или -1 в следующих случаях, перечисленных в таблице В.10.

**Таблица В.10 – Коды ошибок функции `msend`**

Код ошибки	Описание
EBADF	Был определен недопустимый дескриптор.
EACCES	Адрес назначения является ширококвещательным адресом, но <code>SO_BROADCAST</code> не был установлен на сокете.
ENOTSOCK	Параметр <i>msockfd</i> не является сокетом.
EFAULT	Недопустимый адрес пространства пользователя был определен для параметра.
EMSGSIZE	Сокет требует, чтобы сообщение было послано целиком, а размер сообщения, которое будет послано, делает это невозможным.
EAGAIN	Сокет отмечен как неблокируемый, а требуемая операция блокировала бы его.
ENOBUFS	Система не смогла распределить внутренний буфер. Операция может завершиться успешно, когда буферы станут доступными.
ENOBUFS	Очередь вывода для сетевого интерфейса была заполнена. Это обычно указывает на то, что интерфейс прекратил передачу, но ошибка также может быть вызвана транзитной перегрузкой.
EPARTITIONED	Указывает, что сеанс был разбит на разделы.
ПРИМЕЧАНИЕ - Выделенный код ошибки определен для ЕСТР недавно.	

### В.2.6 `mrecv()`

Функция `mrecv` используется, чтобы принимать групповые данные и сигналы индикации для целей управления.

```
ssize_t mrecv (int msockfd, void *buf, size_t buflen, int *flags, struct sockaddr *fromaddr, socklen_t *fromaddrlen);
```

Описание параметров:

- *msockfd*: дескриптор сокета, который был возвращен функцией `msocket`;
- *buf*: указатель на буфер, в котором читаются данные;
- *buflen*: размер *buf*;
- *flags*: еще не определены;
- *fromaddr*: указатель на абсолютный протокольный адрес для определения поставщика услуги;
- *fromaddrlen*: размер *fromaddr*.

Когда приложение получает данные из буфера, оно может идентифицировать соответствующего поставщика услуги, используя *fromaddr*.

Функция `mrecv` возвращает число полученных октетов в случае успеха. В противном случае она возвращает -1, если происходит ошибка или имеется управляющее сообщение, которое будет доставлено приложению. Коды ошибок перечислены в таблице В.11.

Таблица В.11 – Коды ошибок функции `mresv`

Код ошибки	Описание
EBADF	Параметр <code>msockfd</code> является недопустимым дескриптором.
ENOTCONN	Сокет ассоциируется с ориентированным на соединение протоколом и не был связан (см. <code>mconnect</code> и <code>maccept</code> ).
ENOTSOCK	Параметр <code>msockfd</code> не ссылается на сокет.
EAGAIN	Сокет отмечен как неблокируемый, а операция приема заблокировала бы его или была установлена блокировка времени приема и она завершилась, прежде чем данные были получены.
EINTR	Получатель был прерван сигналом до того как какие-либо данные были доступны.
EFAULT	Указатель(ли) буфера получателя указывает(ют) на адрес вне адресного пространства процесса.
ETOTERM	ТО закончил сеанс.
ETOEXPEL	ТО исключил LE или LO.
EPARTITIONED	Указывает, что сеанс был разбит на разделы.
ПРИМЕЧАНИЕ - Выделенные коды ошибок определены для ЕСТР недавно.	

### В.2.7 `mclose()`

Функция `mclose` используется, чтобы оставить или завершить ЕСТР подключение закрытием сокета. Действие по умолчанию `mclose` с ЕСТР сокетом - пометка сокета как закрытого и немедленное возвращение к процессу. Дескриптор сокета больше не пригоден для использования процессом.

```
int mclose (int msockfd);
```

Описание параметров:

- *msockfd*: дескриптор сокета, который был возвращен функцией `msocket`.

Функция `mclose` возвращает нуль в случае успеха или -1 в следующих случаях, перечисленных в таблице В.12.

Таблица В.12 – Коды ошибок функции `mclose`

Код ошибки	Описание
EBADF	<i>msockfd</i> не является активным дескриптором.
EINTR	Было принято прерывание.

### В.2.8 `mgetsockopt()` и `msetsockopt()`

Функция `mgetsockopt` используется для получения параметров и действующих характеристик подключения, которые затрагивают сокет.

```
int mgetsockopt(int msockfd, int level, int optname, void *optval, socklen_t *optlen);
```

Описание параметров:

- *msockfd*: дескриптор сокета, который был возвращен функцией `msocket`;
- *level*: определяет код в системе, чтобы интерпретировать опцию: общий код сокета или некоторый конкретный код протокола (например, IPv4, IPv6 или ЕСТР);
- *optname*: тип опции. Каждый уровень может определять несколько имен опции;
- *optval*: указатель на переменную, в которую текущее значение опции сохраняется функцией `mgetsockopt`; и
- *optlen*: размер *optval* как результат-значение для `mgetsockopt`.

Функция `msetsockopt` используется для установки параметров, которые затрагивают сокет.

```
int msetsockopt (int msockfd, int level, int optname, const void *optval, socklen_t *optlen);
```

Описание параметров:

- *msockfd*: дескриптор сокета, который был возвращен функцией *msocket*;
- *level*: определяет код в системе, чтобы интерпретировать опцию: общий код сокета или некоторый конкретный код протокола (например, IPv4, IPv6 или ECTP);
- *optname*: тип опции. Каждый уровень может определять несколько имен опции;
- *optval*: указатель на переменную, из которой новое значение опции извлекается функцией *msetsockopt*; и
- *optlen*: размер *optval* как результат-значение для *msetsockopt*.

В таблице В.13 приведены параметры, с которыми могут делать запрос функции *mgetsockopt* и *msetsockopt*.

**Таблица В.13 – Параметры сокета для функций *mgetsockopt* и *msetsockopt* с ECTP**

Level	optname	mget	mset	Описание	Флаг	Тип данных
IPPROTO_ECTP	ECTP_QOS	•	•	Получение и установка параметров QoS		QoS
	ECTP_OPPAR		•	Получение и установка адреса управляющего родителя		OP_par
	ECTP_OPVAR1	•	•	Получение и установка значений операции для ECTP части I		OP_var1
	ECTP_OPVAR2	•	•	Получение и установка значений операции для ECTP части II		OP_var2
	ECTP_OPLJ	•	•	Возможность позднего соединения	•	int
	ECTP_OPWCR	•	•	Установка времени ожидания CR		u_long

Функции *mgetsockopt* и *msetsockopt* возвращают нуль в случае успеха или -1 в следующих случаях, перечисленных в таблице В.14.

**Таблица В.14 – Коды ошибок функций *mgetsockopt* и *msetsockopt***

Код ошибки	Описание
EBADF	Параметр <i>msockfd</i> - не допустимый дескриптор.
ENOTSOCK	Параметр <i>msockfd</i> - файл, а не сокет.
ENOPROTOOPT	Опция неизвестна на указанном уровне.
EFAULT	Адрес, указанный значением <i>optval</i> находится в недопустимой части адресного пространства процесса. Для функции <i>mgetsockopt</i> эта ошибка также может возвращаться, если значение <i>optlen</i> находится в недопустимой части адресного пространства процесса.

Следующие структуры данных показаны в качестве информации для использования функцией *mgetsockopt* или *msetsockopt*.

```

typedef struct _op_var1 {
    int agn; /* Номер генерации пакета ACK */
    int mcn; /* Максимальный номер потомка*/
    int mrn; /* Максимальный номер повторной передачи */
    int mtl; /* Максимальный уровень дерева */
    int nft; /* Порог отказа узла */
    int arn; /* Номер активного получателя */
    int csn; /* Текущий номер потомка */
    int crn; /* Текущий номер получателя */
    int ctl; /* Текущий уровень дерева */
    int ctr; /* Текущая скорость передачи */
    int cct; /* Время создания подключения */
}OP_var1;

```

---

```
typedef struct _op_var2{
    int mintr; /* Минимальная скорость передачи */
    int ctr; /* Текущая скорость передачи */
    int maxtr; /* Максимальная скорость передачи */
}OP_var2;
```

---

```
typedef struct _op_par{
    u_long ctladdr; /* Групповой адрес управления локальной группы */
    u_short ctlport; /* Управляющий порт локальной группы */
}OP_par;
```

---

```
typedef struct _QoS {
    u_shortthreshold; /* уровень порога */
    u_long th_CHQ_TQA:2,th_CHQ_N:1,th_CHQ:29; /* пропускная способность CHQ */
    u_long th_OT_TQA:2,th_OT_N:1,th_OT:29; /* пропускная способность OT */
    u_long th_LQA_TQA:2,th_LQA_N:1,th_LQA:29; /* пропускная способность LQA */
    u_long td_OT_TQA:2,td_OT_N:1,td_OT:29; /*транзитная задержка OT */
    u_long td_LQA_TQA:2,td_LQA_N:1,td_LA:29; /* транзитная задержка LQA */
    u_long tdj_OT_TQA:2,tdj_OT_N:1,tdj_OT:29; /* флуктуация транзитной задержки OT */
    u_long tdj_LQA_TQA:2,tdj_LQA_N:1,tdj_LQA:29; /* флуктуация транзитной задержки LQA */
    u_long cr_TSDU_TQA:2,cr_TSDU_N:1,cr_TSDU:29; /* коэффициент искажения данных */
    u_long l_TSDU_TQA:2,l_TSDU_N:1,l_TSDU:29; /* коэффициент потери данных */
}QoS;
```

---

### В.3 Пример файла заголовка msocket.h

Далее описывается файл заголовка 'msocket.h', который используется при выполнении ЕСТР для информации.

---

```
#ifndef __MSOCKET_H_
#define __MSOCKET_H_
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/un.h>
#include "ectpcb.h"
#include "var.h"
#define SIZE_MSETSOCKOPT 524
struct _common{
    u_char command;
    u_short len;
    u_long error;
};
```

```
enum{ MCONNECT = 101,
      MBIND,
      MSEND,
      MRECV,
      MACCEPT,
      MLISTEN,
      MCLOSE,
      MSETSOCKOPT,
      MGETSOCKOPT,
      DETACH};
```

```
/* значения, возвращаемые msocket */
```

```
#define ECONNECT_OK      0x00
#define ECONNECT_ROLE  0x01
#define ESEND_OK        0x0000
#define ESEND_RESEND    0x1004
#define ESEND_NOSEND    0x1001
#define ESEND_NOONE     0x1002
#define EBIND_OK        0x00
#define EBIND_ADDR      0x01
#define EBIND_ROLE      0x02

#define ECLOSE_OK       0x00
#define ECLOSE_TO       0x01 // Когда вызвана ТО
#define ECTP_QOS        0x4001
#define ECTP_OPVAR1     0x4002
#define ECTP_OPVAR2     0x4003
#define ECTP_OPLJ       0x4004
#define ECTP_OPWCR      0x4005
#define ECTP_OPTREE     0x4006
#define ECTP_OPTIME     0x4007
#define ECTP_OPFLOW     0x4008
#define ECTP_OPPAR      0x4009
```

```
/*типы сообщений msocket*/
```

```
#define FIXED_SIZE      24 // размер фиксированного_заголовка
struct _mbind_req{
    struct _common c;
    int    role;
    struct sockaddr_in ctrl;
    struct sockaddr_in local;
    struct sockaddr_in grp;
};
struct _mconnect_req{
```

```

        struct _common c;
};
struct _maccept_req{
        struct _common c;
        struct sockaddr_in to;
};
struct _msend_rep{
        struct _common c;
};
struct _mrecv_req{
        struct _common c;
};
struct _msend_req{
        struct _common c;
        u_char rsvd[FIXED_SIZE - sizeof(struct _common)];
        u_char data[MAXDATA];
};
struct _mrecv_rep{
        struct _common c;
        u_char rsvd[FIXED_SIZE - sizeof(struct _common)];
        u_char data[MAXDATA];
};
struct _mclose_req{
        struct _common c;
};
struct _mopt_op{
        int agn;
        int arn;
        int ccn;
        int ctl;
        int mcn;
        int mrn;
        int mtl;
        int nft;
        int lrtd;
};
struct _mopt_time{
        long agt; // Время генерации пакета ACK
        long cct; // Время создания подключения
        long hgt; // Время генерации пакета HB
        long iat; // Время неактивности
        long ndt; // Время генерации пакета ND
        long rbt; // Время паузы повторной передачи
        long rxt; // Время повторной передачи
        long tct; // Время создания дерева
};

```

```
struct _mopt_flow{
    int cit;
    int cmn;
    int crn;
    int mintr;
    int ctr;
    int maxtr;
    int iri;
    int cri;
    int crd;
};

struct _mopt_tree{
    int tree_ct; // тип подключения
    int tree_conf; // Конфигурация уровня дерева
    int tree_mtl; // Максимальный уровень дерева (действителен только, если tree_conf ==2)
    int tree_mcn; // Максимальный номер потомка
    int bitmapsizе; // размер битового поля АСК
};

struct _mgetsockopt_req{
    struct _common c;
    int optname;
    int optlen;
    u_char optval[MAXDATA/2];
};

struct _msetsockopt_req{
    struct _common c;
    int optname;
    int optlen;
    u_char optval[MAXDATA/2];
};

char cbreq[128];
char dbreq[4096];
#endif
```

---

### Справочная литература

Следующие серии документов IETF RFC будут полезны для понимания или реализации настоящей спецификации:

- IETF RFC 768, *Протокол пользовательских датаграмм, Стандарт Интернета*, август 1980.
- IETF RFC 791, *Протокол Интернета, Программа Интернета DARPA, спецификация Протокола, Стандарт Интернета*, сентябрь 1981.
- IETF RFC 793, *Протокол управления передачей, Программа Интернета DARPA, спецификация Протокола, Стандарт Интернета*, сентябрь 1981.
- IETF RFC 1112, *Узловые расширения для группового IP, Стандарт Интернета*, август 1989.
- IETF RFC 1119, *Сетевой Протокол Времени, Стандарт Интернета*, май 1990.
- IETF RFC 2119, *Ключевые слова для использования в документах RFC для указания уровня требований, Лучшая современная практика*, март 1997.
- IETF RFC 2205, *Протокол резервирования ресурса (RSVP) - Версия 1 Функциональная Спецификация, Предложенный стандарт*, сентябрь 1997.
- IETF RFC 2210, *Использование RSVP с интегрированными службами IETF, Предложенный стандарт*, сентябрь 1997.
- IETF RFC 2236, *Протокол Интернета для управления групп , Версия 2, Предложенный стандарт*, ноябрь 1997.
- IETF RFC 2460, *Протокол Интернета, Спецификация Версии 6 (IPv6), Проект стандарта*, декабрь 1998.
- IETF RFC 2474, *Определение дифференцированного поля служб (Поле DS) в IPv4 и IPv6 Заголовках, Предложенный стандарт*, декабрь 1998.
- IETF RFC 2597, *Гарантированная отправка PNH Группы, Предложенный стандарт*, июнь 1999.
- IETF RFC 2598, *Срочная отправка PNH, Предложенный стандарт*, июнь 1999.
- IETF RFC 2750, *RSVP Расширения для политики управления, Предложенный стандарт*, январь 2000.
- IETF RFC 2836, *Идентификационные коды режима пересылки, Предложенный стандарт*, май 2000.



## СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия А	Организация работы МСЭ-Т
Серия В	Средства выражения: определения, символы, классификация
Серия С	Общая статистика электросвязи
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и средств передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
Серия J	Передача сигналов телевизионных, звуковых программ и других мультимедийных сигналов
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	TMN и техническое обслуживание сетей: международные системы передачи, телефонные, телеграфные, факсимильные и арендованные каналы
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Качество телефонной передачи, телефонные установки, сети местных линий
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
<b>Серия X</b>	<b>Сети передачи данных и взаимосвязь открытых систем</b>
Серия Y	Глобальная информационная инфраструктура и аспекты межсетевого протокола (IP)
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи

\*25385\*