



# Relatório Técnico

**Núcleo de  
Computação Eletrônica**

## Estudo das Propostas de Padronização para Serviços Multicast

Andrea Yumi Igarashi  
Luiz Fernando Rust da Costa Carmo

NCE - 03/97

Universidade Federal do Rio de Janeiro

# Estudo das Propostas de Padronização para Serviços Multicast \*

Andrea Yumi Igarashi  
Luiz Fernando Rust da Costa Carmo

## Parte I - Introdução

Os objetivos principais deste trabalho compreenderam: (i) inicialmente estabelecer um panorama geral do estágio atual de padronização dos novos serviços de transporte que vão servir como suporte para as novas aplicações distribuídas, tais como os sistemas multimídias, sendo basicamente uma atividade de busca aos mais recentes documentos referentes as atuais propostas em andamento; e (ii) realizar um estudo detalhado destas principais propostas de padronização, buscando destacar as novas funcionalidades impostas pelas atuais aplicações, e com isso possibilitar uma posterior caracterização da necessidade do desenvolvimento de novos mecanismos protocolares.

Como resultado deste trabalho elaborou-se este relatório, que esta organizado da seguinte forma: na segunda parte apresenta-se a proposta de padronização da ISO de uma taxonomia para a especificação de comunicação multipartes (*multi-peer*); na terceira parte é feita a descrição das duas propostas de padronização para serviço de transporte multicast (ISO e ITU-T) que atualmente estão sendo discutidas pela comunidade; na quarta parte deste documento é analisado o documento gerado pelo PROJECT COS 127 com o intuito de reforçar as definições da taxonomia multicast proposta através de exemplos de aplicações em cenários tipicamente *multipeer*, mostrando as relações entre as definições e aplicações reais; na quinta e última parte são descritas algumas conclusões sobre o trabalho realizado.

## Resumo das Referências

- ISO/IEC JTC1/SC6 - Draft text on the subject of “Multi-peer Taxonomy”, de 17 de março de 1995;
- ISO/IEC JTC1/SC6 M 9476 - First Draft of Enhanced Communications Transport Service Definition, de 23 de março de 1995;
- ITU-T Recommendation: Information Technology - Open System Interconnection - Multicast Transport Service Definition, de 1994;
- Project COST 237 - Working Group on New Transport Service Specification - Multipeer scenario, de dezembro de 1994.

## 1. ISO/IEC JTC1/SC6 - *Taxonomia Multi-peer*

Nesse documento, são feitas definições de termos empregados em especificação de comunicação *multi-peer*. Define-se como *multi-peer*, um modo de operação de um serviço OSI, que permite troca de informações entre mais de dois usuários. Transmissão multicast é a transmissão de uma mesma unidade de dados de um emissor para vários receptores, num único serviço.

## 2. ISO/IEC JTC1/SC6 - *Definição de Enhanced Communications Transport Service - ECTS*

Nesse documento é proposto uma descrição de um padrão internacional baseado no modelo de camadas OSI, definindo o serviço de transporte necessário para numa comunicação multicast. A definição proposta se refere apenas a uma comunicação *n-plex multicast*, onde um usuário pode enviar e receber dados de, e para, todos os outros.

## 3. ITU-T - *Definição de Serviço de Transporte Multicast*

As recomendações ISO e ITU-T definem basicamente as mesmas características e procedimentos para o serviço de transporte. Existem, porém, algumas diferenças entre essas recomendações. Por exemplo, a recomendação ITU-T não prevê primitivas que o provedor peça uma conexão ao usuário. As diferenças entre a duas propostas serão analisadas na próxima fase do projeto.

## 4. Project COST 237 - *Working Group on New Transport Service Specification - Multipeer scenario*

O propósito desse documento é reforçar as definições da taxonomia multicast proposta, com exemplos de aplicações em cenários tipicamente *multipeer*, mostrando as relações entre as definições e aplicações reais, feito pelo grupo de trabalho da ISO (*working group*) com base na especificação ISO/IEC JTC1/SC6 N9161- "Second Draft on Multipeer Taxonomy". Como analisamos o "Forth Draft on Multipeer Taxonomy" algumas definições da taxonomia foram modificadas. Na avaliação aqui realizada, essas definições foram vistas de acordo com a especificação mais recente.

### 1. Introdução

Nesta seção, são feitas definições de termos empregados em especificação de comunicação multipartes (*multi-peer*). Define-se como *multi-peer*, um modo de operação de um serviço OSI, que permite troca de informações entre mais de dois usuários de serviço OSI. Transmissão multicast é definida como a transmissão de uma mesma unidade de dados de um emissor para vários receptores, num único serviço.

### 2. Conceitos iniciais

#### a) Grupo

O conceito de grupo permite que várias entidades sejam definidas como se fossem uma só, pois têm um nome e endereço únicos denominados respectivamente de nome do grupo e de endereço do grupo. A existência de um grupo é independente de qualquer instância de comunicação, ou seja, o grupo existe, independente de haver troca de informação. Esse grupo é chamado *enrollment group*.

Uma instância de um *enrollment group*, é chamada de *enrolled group* e é composta de  $n$  entidades, chamadas membros. A criação de um *enrolled group* é necessária para identificar o subconjunto de membros que estejam participando de uma comunicação. Essas entidades podem ser alcançadas através do endereço ou nome do grupo, após terem passado pelo processo de registro, explicado mais adiante. Abaixo são listadas algumas características que definem os grupos:

- dinâmico - a sua população (membros) muda, varia dinamicamente;
- estático - a população do grupo não muda;
- definido - quando é possível definir a lista completa de participantes; um grupo definido é dito completamente conhecido se todos os participantes se conhecem, caso contrário é dito parcialmente conhecido;
- indefinido - quando não é possível determinar uma lista completa de participantes.

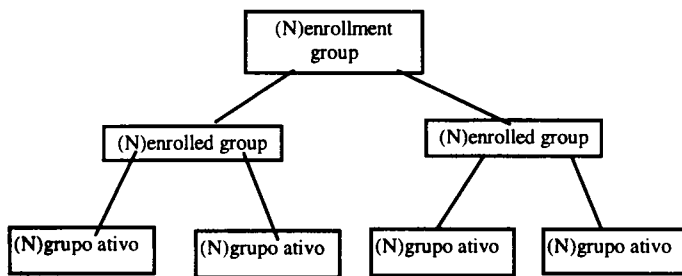
Obs : Dentro de um grupo indefinido, um participante pode ser conhecido por apenas alguns dos outros participantes. Nesse caso, os membros são chamados de *membros conhecidos*.

## b) Associação de grupo (Group Association - GA)

A partir do estabelecimento de um *enrolled group* pode-se fazer associações entre membros deste *enrolled group*, denominadas de *associações de grupo*, com o propósito de transferir dados. O conceito de associação de grupo engloba ambos os modos de transmissão com ou sem conexão. Os membros que aceitam participar de uma associação de grupo são denominados de *participantes* desta associação.

## c) Grupo ativo (Active Group - AG)

Conjunto de membros do *enrolled group* que participam de uma associação de grupo (GA).



## d) Integridade do Grupo Ativo (AGI)

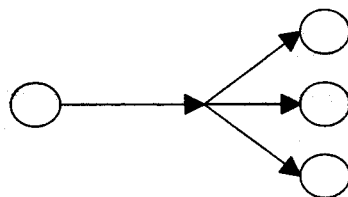
A AGI (Active Group Integrity) especifica as condições de estabelecimento e manutenção de um grupo ativo de uma GA. São os requisitos mínimos e máximos do grupo ativo. Quando uma AGI não é mais satisfeita, a associação é desfeita ou suspensa até que essas condições sejam novamente satisfeitas. Exemplos de possíveis condições de AGI:

- quorum - número mínimo de membros que devem pertencer ao grupo ativo;
- máximo - número máximo de membros que podem pertencer ao grupo ativo;
- membro chave - membro necessário para a existência do grupo ativo.

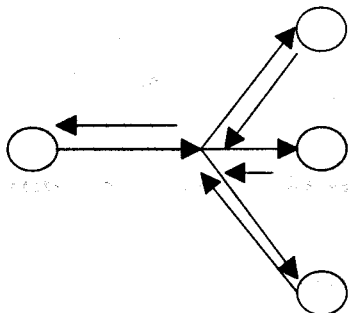
## e) Conversação

São efetivamente os canais criados numa GA para que haja troca de informações entre membros. Uma GA tem várias conversações ocorrendo simultaneamente, e cada membro pode também realizar diferentes conversações ao mesmo tempo. A seguir são definidos os diversos tipos de conversação:

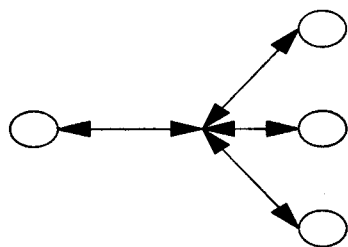
- simplex multicast - um membro manda informações para todos os outros, que só recebem dados;



- duplex multicast - um único membro pode mandar mensagens para todos os outros membros da conversação, e esses membros podem enviar dados de volta somente para esse membro central;



- N-plex multicast - todos os membros podem mandar e receber dados para os outros membros; todo mundo se comunica com todo mundo;



- unicast - conversação de um membro com outro, um a um.

### 3. Topologia de Associações de Grupo

Uma associação pode ser vista como composta de várias sub-associações, cada qual composta de  $n$  conversações. Para que uma associação seja estabelecida, as condições de integridade são verificadas no estabelecimento do grupo, e periodicamente checadas, durante o seu tempo de duração.

#### 3.1. Condições de integridade (IC)

Existem condições de integridade da associação (ATI - Integridade da Topologia de uma Associação de grupo) e de cada grupo ativo (AGI), que devem ser periodicamente checadas. No momento de estabelecimento da associação, são checadas as condições de ATI. No estabelecimento do Grupo ativo, e no decorrer de sua vida útil, as

condições de AGI (Integridade do Grupo Ativo) são testadas, e continuarão tendo que ser satisfeitas durante todo o tempo de existência do grupo. Caso essas condições sejam violadas após o estabelecimento da conexão, tanto da associação quanto do grupo, a associação pode ser desfeita ou suspensa até que as condições voltem a ser válidas. São previstas duas diferentes políticas de gerenciamento das condições de integridade:

- *hard IC management policy* - associação desfeita quando não são mais satisfeitas as condições de integridade;
- *soft IC management policy* - associação suspensa quando não são mais satisfeitas as condições de integridade.

Essas políticas podem ser definidas individualmente para as condições de integridade de uma associação, do grupo ativo e do grupo ativo de conversação, mas as condições analisadas devem no entanto serem as mesmas.

## **4. Modelo de serviço de uma operação de grupo**

Este modelo define as operações necessárias para a constituição (e as respectivas operações inversas) de um grupo (*enrolled group*).

### **4.1. Estabelecimento do Enrollment Group**

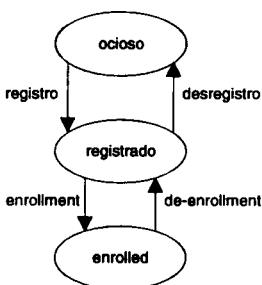
Esta primeira operação refere-se à fase de criação de um enrollment group, sendo realizada através da criação de estruturas de grupo. Este procedimento permite a um criador de grupo especificar um conjunto de regras que definem o perfil dos futuros membros de um *enrolled group*.

### **4.2. Registro do grupo**

Procedimento de inicialização que permite a uma entidade vir a ser instanciada como membro de um atual *enrolled group*. A entidade se torna conhecida pela gerência do grupo, o que a permite depois passar por uma operação de *enrollment* e se tornar um membro deste grupo. Antes de ser registrada, a entidade tem que se submeter às regras definindo o grupo. Para isso, a entidade precisa apresentar suas características ao grupo, por exemplo fornecendo o endereço para ser incluído nos diretórios, os parâmetros que caracterizem suas capacidades e as regras de controle de acesso para que sejam aplicadas a ela em comunicações posteriores.

### **4.3. Group Enrollment**

Torna uma entidade já registrada, membro de um *enrolled group*. Pode-se fazer uma analogia entre a operação de enrollment com a facilidade de um recurso passar de on-line a off-line (e vice-versa), sem que o sistema ache que o recurso deixou de existir. Um *enrolled group* é portanto composto de entidades que passaram pela fase de registro e enrollment.



## 5. Modelo de serviço da Operação da Associação de Grupo(GA)

### 5.1. Estabelecimento e liberação

Na operação estabelecimento um membro de um *enrolled group* cria uma GA entre os membros desse mesmo grupo. Nessa operação são definidas todas as características da GA, como Integridade(IC) e Qualidade de Serviço(QoS). O membro que inicia o estabelecimento da GA é chamado de **iniciador**. Ele pode expressar características obrigatórias ou negociáveis para a futura GA. A operação de estabelecimento do grupo é feita através de um pedido aos membros do *enrolled group*, podendo estes aceitarem ou não.

Define-se como *condições de estabelecimento*, as condições mínimas especificadas pelo **iniciador** para que o grupo seja criado, tendo que ser coerente com as características da GA. Se as condições mínimas forem atingidas, uma nova GA é criada e os participantes dessa GA formam um grupo ativo(AG).

A operação de liberação é utilizada para encerrar uma GA, podendo ser iniciada pelo usuário ou pelo provedor (quando as condições de integridade não forem mais satisfeitas).

### 5.2. Join e Leave

A operação de *Join* permite que uma entidade se junte a uma GA existente de acordo com as suas regras de elasticidade (estática ou dinâmica). Existem dois tipos de *join*:

- operação *join-invitation* - onde os participantes podem convidar uma entidade a se juntar à GA;
- operação *join-calling* - a entidade requisita a participação numa GA existente, essa operação assume que a entidade sabe que a GA foi previamente estabelecida.

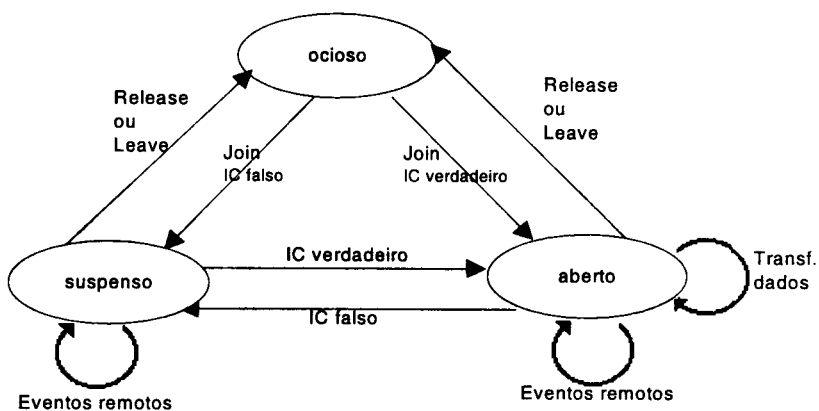
Para que o *join* seja bem sucedido, as características de GA são sempre levadas em conta: (i) para uma GA dinâmica e aberta a entidade não precisa pertencer ao grupo e (ii) no caso de uma GA dinâmica e fechada a entidade precisa pertencer ao grupo.



A operação de *Leave* permite que o participante deixe a GA, podendo ser iniciada pelo participante que quer deixar a GA ou por outro participante. Dependendo das condições de AGI da GA no momento, o *leave* pode desfazer o grupo, iniciando uma liberação. A operação de *Leave* pode também ser iniciado pelo provedor de serviço, quando ele não puder mais satisfazer à qualidade de serviço(QoS).

### 5.3. Transferência de Dados

Transferência de (N-1) SDUs entre os participantes de uma (N) GA. A transferência de dados só é permitida se as condições de integridade da associação (IC) são satisfeitas. O diagrama de estados a seguir descreve o comportamento de um participante quando este realiza uma operação de *join* à uma *associação de grupo* (GA), habilitando a transferência de dados.



## 6. Características de uma Associação de grupo (GA)

### 6.1. Topologia da GA

A topologia de uma associação engloba a definição das seguintes características:

- descrição do grupo ativo, incluindo as características de população e, se necessário, sua lista de participantes conhecidos;
- para cada conversação, o seu tipo (simplex multicast, duplex multicast, unicast), o grupo ativo da conversação, e as características de população da conversação, e ainda, se necessário, a lista de participantes conhecidos e a definição do papel de cada um (se emissor ou receptor).

### 6.2. Regras de controle de membros

Controlam o modo como as GAs são definidas, estabelecidas e mantidas. Essas regras definem :

- a elasticidade da GA;
- as condições de integridade da GA;

- as políticas de gerência das condições de integridade.

### **6.3. Modelo de transmissão**

Define o modo como os dados serão transmitidos entre os participantes. Do ponto de vista do usuário, o modelo de transmissão define a ordem de entrega das SDUs.

#### **6.3.1. Ordem de envio e recepção de PDUs**

É aplicada à nível de serviço e de protocolo. À nível de serviço, o provedor de serviço pode ser requisitado a garantir a ordem em que as SDUs são entregues ao usuário. À nível de protocolo, as PDUs são ordenadas ou reordenadas de acordo com a propriedade de ordem requerida pelo serviço.

##### *a) Sem ordem*

O provedor de serviço não garante nenhuma relação entre unidades de dados enviadas por uma mesma fonte ou por fontes diferentes.

##### *b) Ordem local*

Numa ordenação local, as unidades de dados enviadas por uma determinada fonte, são entregues aos seus destinos na mesma ordem em que foram enviadas; mas não é estabelecida uma ordem entre unidades de dados enviadas por fontes diferentes, mesmo que na mesma associação.

##### *c) Ordem Parcial*

As unidades de dados enviadas, são entregues a seu destino, seguindo uma regra arbitrária definida. Se a regra de entrega for aplicável a todos os receptores, todos os destinos receberão as unidades numa mesma ordem. Se a regra de entrega for diferente para cada receptor, cada usuário receberá suas unidades de dados numa ordem diferente.

##### *d) Ordem causal*

Baseia-se numa dependência causal entre as unidades de dados geradas. Os dados são entregues de acordo com a relação causa - consequência entre as unidades geradas.

##### *e) Ordem total*

As unidades de dados geradas por todas as fontes, são entregues para cada usuário destino, na mesma ordem em que foram enviadas.

### **6.4. Sincronização de diferentes conversações**

Em sistemas multimídia é necessária a sincronização entre unidades de dado geradas por diversas fontes. Por isso foi definida uma nova facilidade chamada orquestração que permite sincronizar unidades de dados transmitidas entre diferentes conversações.

## 6.5. Qualidade de Serviço (QoS)

Numa emissão de dados peer-to-peer (entre duas entidades), a qualidade de serviço está associada à cada parâmetro em cada direção da transferência de dados. Numa transmissão multi-peer, os parâmetros de qualidade de serviço são diferentes, pois têm que controlar a transferência de dados entre vários emissores e receptores.

Para controlar comunicações multi-peer, temos os parâmetros de QoS associados à cada conversaç o de uma GA. No caso de conversaç es multicast, existem duas categorias de parâmetros de QoS: (i) coletivo, abrangendo todos os emissores e receptores de uma conversaç o e (ii) individual, que s o diz respeito a um  nico receptor.

Obs: Os parâmetros de QoS est o definidos na norma ISO/IEC 9476.

## 7. Ger ncia do grupo

### 7.1. Estrutura do grupo

A estrutura do grupo pode ser encarada como sendo o conjunto de regras que definem as condiç es para se tornar membro desse determinado grupo. Pode ser definida como est tica ou din mica. Se for est tica, suas condiç es n o evoluem no decorrer da "vida" do grupo; e se for din mica, sim. Essas regras englobam os seguintes elementos:

- regra de identificaç o de membros;
- regra especificando o modo de se realizar o registro e enrollment;
- requisitos m nimos para o registro de uma entidade;
- regras para pol tica de ger ncia de grupo, a identificaç o de entidades que podem realizar operaç es de ger ncia de grupo (chamados gerentes de grupo) e suas atribuiç es.

As operaç es de estrutura de grupo consistem em criaç o, modificaç o e cancelamento da estrutura do grupo. A organizaç o da estrutura de grupo diz respeito   organizaç o dos grupos em termos de hierarquia e ainda n o foi definida.

### 7.2. Operaç es de estrutura de grupo

#### 7.2.1. Criaç o da estrutura de grupo

A criaç o de uma estrutura de grupo   iniciada por um *criador-de-grupo*, e define uma fase chamada de *criaç o de estrutura de grupo*.

### **7.2.2. Modificação da estrutura de grupo**

As modificações nas regras de estrutura do grupo são feitas pelo gerente de grupo na fase de *gerência de estrutura de grupo*. Só podem ocorrer modificações se o grupo estiver definido como dinâmico. Cada regra de estrutura de grupo, tem que ser definida como modificável ou não, e em caso afirmativo, definidos os grupos autorizados a fazê-la.

### **7.2.3. Cancelamento da estrutura de grupo**

É realizada pelo gerente do grupo e define a fase de *cancelamento de estrutura de grupo*.

## **8. Mecanismos e políticas de controle**

Um protocolo tem uma série de requisitos básicos, escolhidos de acordo com o serviço desejado para determinadas operações. Cada função de protocolo divide-se em duas partes: o mecanismo, que é determinado uma vez na definição do protocolo e não pode ser mudado; e as políticas, que podem ser escolhidas e utilizadas em cada instância da comunicação em que o protocolo é usado. A escolha da política afeta o modo em que o mecanismo será utilizado em determinadas circunstâncias. Os mecanismos e políticas não foram definidos, e são questões que estão em aberto, para estudos posteriores.

## Parte III - Definição de Serviços de Transporte Multicast

ISO - Enhanced Communications Transport Service Definition  
ITU-T - Multicast Transport Service Definition.

### 1. Introdução

Nesta seção é feita a descrição das proposta de padronização para serviço de transporte multicast baseado no modelo de camadas OSI. Atualmente existem duas novas propostas de recomendações que estão em discussão:

- ISO/IEC JTC1/SC6 M 9476 - First Draft of Enhanced Communications Transport Service Definition;
- ITU-T Recommendation: Information Technology - Open System Interconnection - Multicast Transport Service Definition.

As recomendações ISO e ITU-T definem basicamente as mesmas características e procedimentos para o serviço de transporte. Nesta seção seguiremos principalmente a proposta da ISO (por ser mais completa) e destacaremos quando necessário as eventuais diferenças que existam entre essas recomendações.

A seguir é feita uma lista das principais abreviações que serão utilizadas:

AGI Active Group Integrity  
OSIE Open System Interconnection Environment  
TA Transport Association  
TAEP Transport Association Endpoint  
TC Transport Conversation  
TCEP Transport Communication Endpoint  
TCEPI Transport Communication endpoint Identifier  
TS Transport Service  
TSAP Transport -service-access-point  
TSDU Transport -service-data-unit  
TPDU Transport-protocol-data-unit  
QoS Quality of Service

### 2. Características do serviço de transporte multicast

Um provedor de serviços de transporte multicast deve oferecer funcionalidades para que um usuário se ligue a uma entidade de transporte com o intuito de participar de uma TA (associação de transporte). Este tipo de operação envolve as seguintes características:

- o QoS deve ser especificado na fase de alocação, durante uma operação de *bind*;
- a AGI (integridade de grupo ativo) deve ser negociada na fase de enrollment e aplicada na fase de transferência de dados;

- a transferência de TSDUs numa TC (Transport Conversation) deve ser transparente; o fato de TSDUs serem corretamente entregues pode ou não ser conhecido;
- os usuários que estão recebendo os dados são responsáveis pelo controle de fluxo das TSDUs que estão sendo enviadas;
- o usuário pode deixar a associação por iniciativa própria ou sob certas condições de AGI e QoS;
- a entidade de transporte pode fazer o usuário deixar a associação.

### 3. Modelo do Serviço de Transporte *Enhanced* - ECTS

O ECTS utiliza os conceitos de conversação, associação, *enrolled group*, *registered group* e *enrollment group* descritos na sessão sobre taxonomia multipartes. Uma conversação é aqui referenciada por TC - Conversação de Transporte, e a Associação de Grupo (GA) é descrita como uma associação de transporte.

No caso da recomendação proposta pela ITU-T, uma **associação de grupo** é conhecida como **conexão de grupo**, sendo que as definições de *enrollment group*, *enrolled group* e grupo ativo são as mesmas.

#### 3.1. Modelo de uma Conversação de Transporte (TC)

Na figura 1 é apresentado o modelo abstrato de um tipo de TC denominado de T-GC (*Transport Group Conversation* - conversação de grupo de transporte). Numa T-GC cada participante pode enviar e receber mensagens. A qualquer momento algum participante pode enviar uma mensagem, e se o fizer, todos os outros participantes podem recebê-la.

Essa T-GC pode ser descrita como uma conversação *duplex descentralizada multipartes* (N x N), que pode ser considerada como uma conversação genérica. Outros tipos de TC seriam então degenerações da T-GC, referenciada como *N-plex multicast* no documento sobre taxonomia. Um exemplo de outro tipo de conversação seria quando um emissor, chamado mestre, somente envia mensagens e os outros participantes do grupo apenas recebem, compondo uma conversação *outward simplex multicast*, chamada *simplex multicast* na taxonomia.

Cada TC tem um TC-id associado e cada terminal de uma TC é denominado TC-endpoint (TCEP), tendo cada TCEP um identificador associado o TCEP-id (TCEPI). Numa mesma T-GC cada participante tem apenas um TCEP, através do qual envia e recebe os seus dados. Um TCEP é um ponto extremidade duplex, tipo soquete.

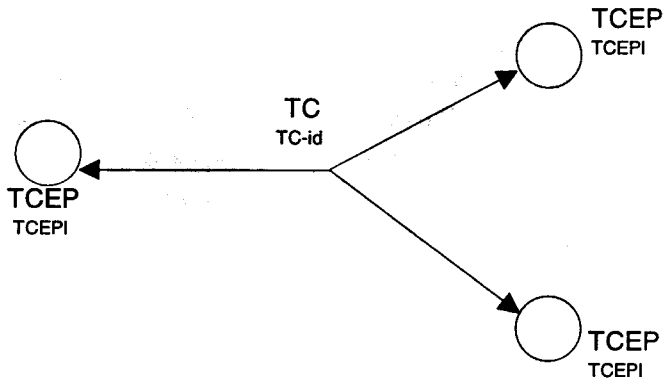
Efetivamente, as transmissões numa associação se realizam através das conversações. Cada TC é responsável por um único tipo de tráfego de dados, cujo QoS é determinado na fase de alocação. Todas as transmissões de dados, qualquer que seja a sua direção numa dada TC, são regidas pelas mesmas condições de QoS.

### *Tipos de Conversação de Transporte (TC)*

Os tipos de conversação apresentados na taxonomia multipartes são definidos aqui com outros nomes. Segue uma lista de correspondência entre os tipos de conversação nos dois documentos:

| Taxonomia         | ISO Multicast                           |
|-------------------|---|
| N-plex multicast  | Transport Group Conversation (T-GC)     |
| simplex multicast | Transport Multicast Conversation (T-MC) |
| duplex multicast  | Transport Polling Conversation (T-PC)   |
| unicast           | Transport Dualcast Conversation (T-DC)  |

Foram definidos ainda no padrão ISO, mais dois tipos de conversação de transporte: o *Transport Concentration Conversation (T-CC)*, onde o participante mestre apenas recebe mensagens e os outros participantes enviam mensagens; e o *Transport Unicast Conversation (T-UC)*, onde a transmissão de dados é de um participante para o outro, sendo que um apenas envia dados e o outro apenas recebe.



*Fig. 1 - Conversação de Transporte genérica*

### **3.2. Modelo de uma Associação de transporte**

Uma associação de transporte (TA) é feita entre membros de um *enrolled group* de transporte, com o propósito de transferir dados, sendo que a transmissão de dados em si, ocorre na conversação. Os *enrolled groups* definem grupos de usuários dispostos a iniciar comunicação multicast. Dentro de cada *enrolled group*, são estabelecidas associações, as quais o usuário pode se unir ou não, dependendo do seu interesse. Cada usuário pode participar de mais de uma associação. Dentro de cada associação, são estabelecidas as conversações, onde são efetivamente realizada a transferência de dados. Essa disposição pode ser observada na fig.2.

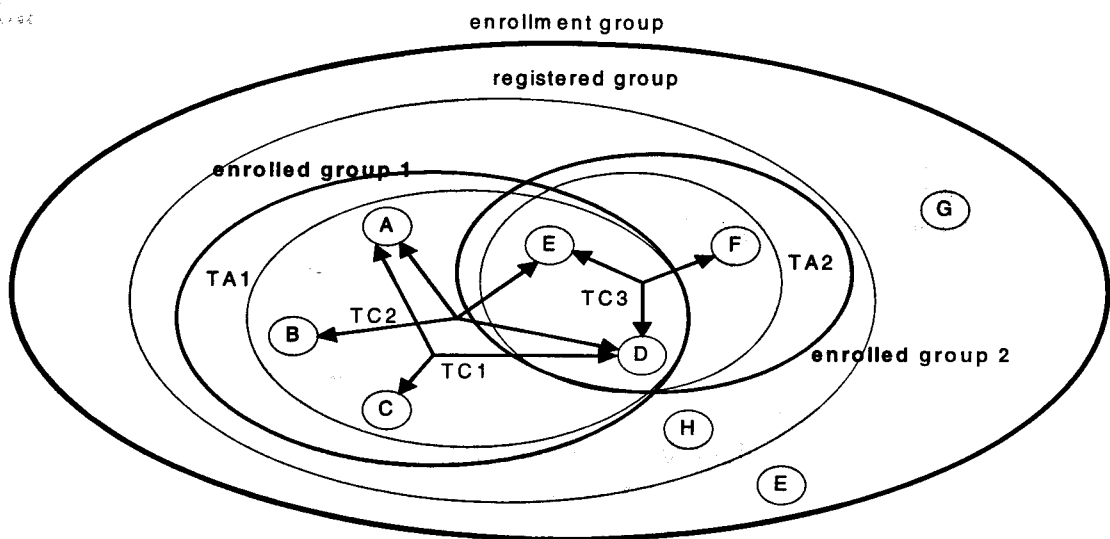


Fig 2 - Relação entre conversação, associação, enrolled groups.

Uma TA pode conter uma ou mais conversações, cada membro dessa TA pode participar de mais de uma associação e manter diversas conversações. Cada TA é identificada pelo seu TA-id. Como uma TA tem vários participantes, tem múltiplos endpoints (TAEPs), cada um deles residindo no TSAP (ponto de acesso ao serviço de transporte) do respectivo participante. O subconjunto de participantes de uma TA que estão no estado de transferência de dados é chamado T-grupo ativo. A figura 3 mostra um modelo típico de associação de transporte, mostrando a relação entre TSAPs, TCEPs e Tcs.

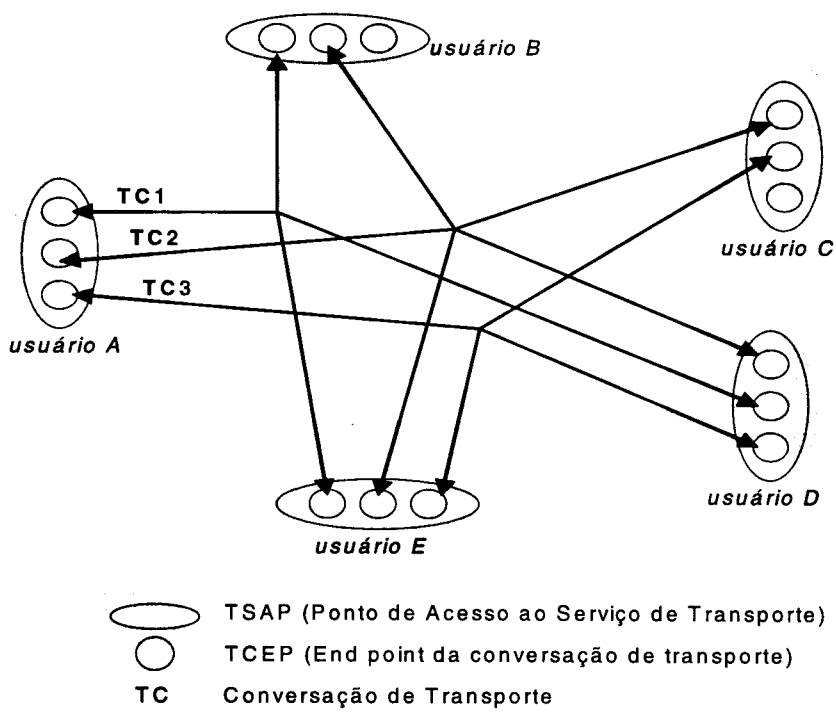


Fig. 3: Exemplo de uma TA



## 4. Procedimentos para o estabelecimento de uma comunicação multicast

Na comunicação multicast, um usuário manda dados para um determinado grupo de usuários, e também pode receber dados dos mesmos. Para isso, são formados os denominados *enrolled groups*. Neste item são expostos os procedimentos necessários para transferência de dados multicast considerando que o *enrolled group* já foi criado.

### 4.1. Diagrama de Estados

Para que uma entidade possa participar de uma sessão multicast, é necessário que se associe ao grupo em que ela deseja trocar dados, sendo que o grupo já tem que estar formado. Considerando que o grupo já esteja formado e registrado, temos duas fases no diagrama de estados: (i) a fase de alocação (onde a entidade se une a determinado grupo) e (ii) a fase de transferência de dados.

O primeiro estado, no início da fase de alocação, é o estado *enrolled*. Nesse estado, assume-se que :

- o nome da TA (associação) foi determinado;
- a associação tem um TA-id;
- as condições de AGI estão determinadas e não são negociáveis;
- existe uma lista de conversações com o TC-id de cada conversação e o tipo de cada uma;
- existe uma lista de QoS, descrevendo a qualidade de serviço de cada conversação em curso; a qualidade de serviço é negociada para cada conversação.



## 4.2. Tabela de primitivas e parâmetros

| Fase      | Primitiva de Serviço       | Parâmetros  |
|-----------|----------------------------|---|
| Alocação  | T-BIND.requestor.submit    | endereço chamador, endereço chamado, condições de TA, QoS |
|           | T-BIND.requestor.deliver   | endereço chamador, endereço chamado, condições de TA, QoS |
|           | T-BIND.acceptor.submit     | endereço chamador, endereço chamado, condições de TA, QoS |
|           | T-BIND.acceptor.deliver    | endereço chamador, endereço chamado, condições de TA, QoS |
|           | T-JOIN.requestor.submit    | endereço chamador, endereço chamado, condições de TA, QoS |
|           | T-JOIN.requestor.deliver   | endereço chamador, endereço chamado, condições de TA, QoS |
|           | T-JOIN.acceptor.submit     | endereço chamador, endereço chamado, condições de TA, QoS |
|           | T-JOIN.acceptor.deliver    | endereço chamador, endereço chamado, condições de TA, QoS |
| Dado      | T-DATA.requestor.submit    | referência chamador, referência chamado, dados do usuário |
|           | T-DATA.acceptor.deliver    | referência chamador, referência chamado, dados do usuário |
|           | T-PAUSE.requestor.submit   | endereço chamado, razão                                   |
|           | T-PAUSE.acceptor.deliver   | endereço chamado, razão                                   |
|           | T-READY.acceptor.submit    | endereço chamado  |
|           | T-REPORT.acceptor.deliver  | endereço chamado  |
| Desalocar | T-LEAVE.requestor.submit   | endereço chamador, endereço chamado, razão                |
|           | T-LEAVE.requestor.deliver  | endereço chamador, endereço chamado, razão                |
|           | T-LEAVE.acceptor.submit    | endereço chamador, endereço chamado, razão                |
|           | T-LEAVE.acceptor.deliver   | endereço chamador, endereço chamado, razão                |
|           | T-UNBIND.requestor.submit  | endereço chamado, endereço chamador                       |
|           | T-UNBIND.requestor.deliver | endereço chamado, endereço chamador                       |
|           | T-UNBIND.acceptor.submit   | endereço chamado, endereço chamador                       |
|           | T-UNBIND.acceptor.deliver  | endereço chamado, endereço chamador                       |

As primitivas *...requestor...* são iniciadas pelo usuário, e as primitivas *...acceptor...* são iniciadas pelo provedor (provedor do *enrolled group*). Existem quatro estados principais: *enrolled*, *bound*, *alocado* e *transferência*.

### 4.3. Descrição dos procedimentos do diagrama de estados

#### 4.3.1. Fase de Alocação

##### *BIND*

No caso de um usuário iniciar o pedido de entrada na associação

1. Quando um usuário deseja iniciar ou se juntar a uma TA, ele manda ao provedor um T-BIND.requestor.submit (endereço chamador, endereço chamado, condições de TA, QoS), e entra no estado *espera\_bindOUT*. A TA é identificada pelo endereço chamado (passado como parâmetro da primitiva). Essa primitiva também manda a QoS, pois ela pode ter sido modificada pelo usuário em algum momento depois que ele se tornou integrante do grupo.
2. Em resposta a um T-BIND.requestor.submit, o provedor obtém informações sobre essa TA através de troca de TPDUs internas com os participantes da associação. Em seguida o provedor responde ao usuário com um T-BIND.requestor.deliver, levando como parâmetros as condições de

integridade e QoS mais recentes na TA, e fazendo com que o usuário passe para o estado *bound*.

No caso do provedor pedir ao usuário que se associe

1. O provedor manda ao usuário2 uma primitiva T-BIND.acceptor.deliver (ao iniciar uma TA após um pedido feito por um usuário denominado usuário1). Essa primitiva contém os parâmetros de QoS que o usuário1 mandou ao fazer o pedido, ou a QoS modificada de acordo com a capacidade do provedor.
2. O usuário2 vai então para o estado *espera\_bind IN*. Se o usuário2 aceita as condições, ele manda ao provedor uma primitiva T-BIND.acceptor.submit com o QoS que ele pode manter e entra no estado *espera\_bind OUT*. Nesse estágio, o usuário2 ainda não sabe qual será a QoS final negociada.
3. O provedor passa ao usuário2 a QoS final, conhecida através de negociações como os outros usuários da TA, através da primitiva T-BIND.requestor.deliver e o usuário2 passa ao estado *bound*.

#### **JOIN:**

No caso do usuário iniciar o pedido :

1. Quando o usuário está no estado *bound* e aceita as condições de integridade e QoS definidas pelo provedor, ele se une à TA mandando ao provedor um T-JOIN.requestor.submit, sem mais negociações de QoS e passa para o estado *espera\_join OUT*.
2. O provedor troca PDUs internas com os componentes da TA, para saber das condições de AGI correntes na TA. Caso sejam satisfeitas, ele responde ao usuário com uma primitiva T-JOIN.requestor.deliver, contendo o valor de QoS final negociado. O usuário passa então para o estado *alocado*.

No caso do pedido iniciado pelo provedor :

1. O provedor convida um usuário a entrar na TA mandando um T-JOIN.acceptor.deliver e o usuário entra no estado *espera\_join IN*.
2. Se o usuário aceita as condições, manda uma primitiva T-JOIN.acceptor.submit e passa para o estado *alocado*.

A negociação de QoS é feita durante o BIND, e a negociação de AGI feita durante o JOIN.

O estado final da fase de alocação é o estado *alocado*. Desse estado são iniciadas as tentativas de transferência de dados. Nesse estado podem ou não existir conversações (TC) abertas. Nesse caso, o provedor é quem decide estabelecer conexões multiponto para uma TC dependendo das condições de QoS.

### 4.3.2. Fase de Liberação

#### UNBIND

Pode ocorrer mediante as seguintes condições: (i) no caso do usuário querer remover o seu *binding*; (ii) no caso do provedor querer remover o *binding* do usuário ou (iii) no caso do provedor indicar ao usuário que seu *binding* à associação de grupo não é possível.

##### *Procedimentos para a condição (i)*

1. Caso o usuário queira remover o seu *binding* (após ter enviado a primitiva T-BIND.requestor.submit) e passado ao estado *espera\_bindOUT*, ele envia ao provedor a primitiva T-UNBIND.requestor.submit e retorna ao estado *enrolled*.
2. Caso o usuário não queira fazer o *binding* após receber do provedor o T-BIND.acceptor.deliver, no estado *espera\_bindIN*, ele envia ao provedor o T-UNBIND.acceptor.submit e retorna ao estado *enrolled*.
3. Caso o usuário queira fazer um UNBIND quando se encontra no estado *bound*, ele envia uma primitiva T-UNBIND.requestor.submit ao provedor, e retorna ao estado *enrolled*.

##### *Procedimentos para a condição (ii)*

1. Se o usuário estiver no estado *espera\_bindOUT*, tanto devido a uma iniciativa do usuário através do envio da primitiva T-BIND.requestor.submit, tanto por iniciativa do provedor através do envio da primitiva T-BIND.acceptor.deliver, (e confirmado pelo usuário através de um T-BIND.acceptor.submit), o provedor pode enviar uma primitiva T-UNBIND.requestor.deliver, indicando que não foi possível o *bind* à associação, provocando um retorno ao estado *enrolled*.

##### *Procedimentos para a condição (iii)*

1. Se o usuário estiver no estado *bound*, o provedor pode solicitar o seu UNBIND enviando a primitiva T-UNBIND.acceptor.deliver, provocando uma transição ao estado *enrolled*.

#### LEAVE

Pode ocorrer de acordo com as seguintes condições: (i) quando o usuário desejar sair da associação, (ii) no caso do provedor querer remover algum usuário e (iii) quando o provedor avisa ao usuário que não foi possível realizar o JOIN à associação.

##### *Procedimentos para a condição (i)*

1. Caso o usuário queira remover o seu a partir do estado *espera\_joinOUT* (alcançado após o envio da primitiva T-JOIN.requestor.submit), ele envia ao provedor a primitiva T-LEAVE.requestor.submit e retorna ao estado *bound*.
2. Caso o usuário não queira fazer o JOIN após receber do provedor o T-JOIN.acceptor.deliver (estando no estado *espera\_joinIN*), ele envia ao provedor o T-LEAVE.acceptor.submit e retorna ao estado *bound*.
3. Caso o usuário queira fazer um LEAVE quando se encontra no estado *alocado*, ele envia uma primitiva T-LEAVE.requestor.submit ao provedor, e retorna ao estado *bound*.

#### *Procedimentos para a condição (ii)*

1. Se o usuário estiver no estado *espera\_joinOUT*, alcançado depois de enviar a primitiva T-JOIN.requestor.submit ao provedor, o mesmo pode enviar uma primitiva T-LEAVE.requestor.deliver, indicando que não foi possível o *join* à associação, retornando o usuário ao estado *bound*.

#### *Procedimentos para a condição (iii)*

1. Se o usuário estiver no estado *alocado*, o provedor pode solicitar o seu LEAVE enviando e ele a primitiva T-LEAVE.acceptor.deliver, passando então o usuário ao estado *bound*.

### **4.3.3. Fase de Transferência de Dados**

1. Quando o provedor determina que a transferência de dados pode ser feita de uma forma segura e as condições de AGI são satisfeitas, ele manda ao usuário uma primitiva T-READY.acceptor.deliver e o usuário entra no estado *transferência*.
2. No estado *transferência*, o usuário requisita transferência de dados através da primitiva T-DATA.requestor.submit.
3. O provedor então envia os dados para o destino usando T-DATA.requestor.deliver.
4. Quando ocorre uma falha nas condições de AGI ou QoS, o provedor envia aos usuários uma primitiva T-PAUSE.acceptor.deliver e suspende a transferência de dados. O usuário volta ao estado *alocado*.
5. O usuário pode suspender a transferência de dados enviando ao provedor uma primitiva T-PAUSE.requestor.submit. O usuário volta ao estado *alocado*.
6. Se os valores de *threshold* de um ou mais parâmetros de QoS são alcançados, o provedor avisa o usuário através da primitiva T-REPORT.acceptor.deliver.

### **4.4 Particularidades da recomendação ITU-T**

Uma primeira diferença consiste no formato adotado para descrição dos procedimentos envolvidos numa comunicação multicast: a ITU-T faz uso de fluxogramas ao invés de máquinas de estados. As fases descritas no ITU-T são as mesmas do padrão ISO: fase de *enrollment*, alocação, transferência de dados e desalocação. As primitivas de algumas fases é que diferem, pois, no padrão da ITU-T é utilizado um número menor de primitivas.

A Fase de enrollment foi definida num outro documento, anexado ao documento estudado, e a sua descrição é idêntica à feita no padrão ISO.

A fase de alocação utiliza apenas as primitivas T-BIND.requestor.submit, T-BIND.requestor.deliver, T-JOIN.requestor.submit e T-JOIN.requestor.deliver. Não há, portanto, a possibilidade do provedor pedir ao usuário que faça um *bind* ao grupo.

Na fase de Transferência de Dados existe uma primitiva T-RESUME.acceptor.deliver que funciona como o T-READY.acceptor.submit, indicando que a transmissão de dados pode ter início. Não é previsto a existência da primitiva T-REPORT.acceptor.deliver.

Durante a fase de liberação, utiliza-se apenas as primitivas T-LEAVE.requestor.submit, T-LEAVE.acceptor.deliver, T-UNBIND.requestor.submit e T-UNBIND.acceptor.deliver.

No que diz respeito aos parâmetros de qualidade de serviço (QoS) previstos na proposta da ITU-T, pode-se destacar a inclusão das seguintes opções:

- Comportamento dos terminadores e originadores do grupo em relação à transferência de dados: *originate only, terminate only, originate/terminate*;
- Comportamento dos terminadores do grupo em relação ao envio de dados de volta ao originador: *response, no response*;

## 5. Qualidade de Serviço (QoS)

### 5.1. Parâmetros de QoS

| Categoria                      | Parâmetros                      | Valores associados ou Opções                                 |
|--------------------------------|---------------------------------|--|
| Performance<br>Conversação(TC) | <i>Throughput</i>               | Valor mínimo compulsório                                     |
|                                |                                 | Valor de <i>threshold</i>                                    |
|                                |                                 | Valor máximo   |
|                                | Indicador de tipo de tráfego    | Opção de tipo de tráfego                                     |
|                                | Atraso de trânsito              | Valor máximo compulsório                                     |
|                                |                                 | Valor de <i>threshold</i>                                    |
| Jitter no atraso de trânsito   | Valor máximo compulsório        |  |
|                                | Valor de <i>threshold</i>       |  |
| Controle de Erro<br>na TC      | Política de controle de erro    | Opção TSDU corrompida  |
|                                |                                 | Opção TSDU perdida   |
|                                | Taxa de erro de TSDU corrompida | Taxa máxima compulsória<br>Valor de taxa de <i>threshold</i> |
|                                | Taxa de erro de TSDU perdida    | Taxa máxima compulsória<br>Valor de taxa de <i>threshold</i> |
| Outras características         | Proteção de TC                  | Opção de proteção de TC                                      |

## 5.2. Definição dos parâmetros de QoS

### 1) *Throughput*

Para cada TC, o *throughput* é definido como a razão entre o tamanho de um TSDU enviado via T-DATA.requestor.submit e o intervalo de tempo entre duas primitivas T-DATA.requestor.submit consecutivas.

O *throughput* é negociado entre o usuário e o provedor. Depois da negociação inicial de QoS na operação BIND, os valores relevantes do *throughput* de uma TC são :

- Valor mínimo compulsório de *throughput* - intervalo máximo entre duas primitivas T-DATA.requestor.submit consecutivas. Se for alcançado, a TC tem que ser congelada, através da primitiva T-Pause.acceptor.deliver.
- Valor de *threshold* - limiar de alerta para o *throughput*. Quando alcançado (o provedor não consegue manter o valor de QoS), o provedor avisa ao usuário através da primitiva T-REPORT.acceptor.deliver.
- Valor máximo negociado - Corresponde ao intervalo de tempo mínimo entre duas primitivas T-DATA.requestor.submit consecutivas. Nunca pode ser excedido, enquanto as condições de AGI se aplicarem. Esse valor máximo do *throughput* é utilizado para controle de taxas de transmissão.

### 2) Indicador de tipo de tráfego

Para cada TC, o usuário informa ao provedor (não é negociável) o tipo de tráfego que ele pretende enviar, se isócrono ou não-isócrono.

### 3) Atraso de trânsito

É o tempo entre a ocorrência de um T-DATA.requestor.submit no TCEP origem até a ocorrência de um T-DATA.requestor.deliver nos destinos para cada TC, ou seja, o tempo que o pedido de dados leva para ser aceito pelo provedor.

Depois da negociação inicial de QoS no BIND, os valores relevantes são :

- *threshold* do atraso de trânsito - quando alcançado, o provedor mantém a TC aberta, mas avisa o fato ao usuário mandando um T-REPORT.acceptor.deliver.
- Valor máximo compulsório - se alcançado, o provedor tem que congelar a TC, mandando um T-PAUSE.acceptor.deliver aos usuários.

### 4) Jitter do atraso de trânsito

Diferença entre o mais longo e o menor atraso de trânsito observado numa mesma TC, depois da negociação de QoS após a operação JOIN. Valores relevantes :

- *threshold* - quando alcançado, envia aviso aos usuários através de um T-REPORT.acceptor.deliver.
- Valor máximo compulsório - se for ultrapassado em qualquer transferência, ele tem que congelar a TC com um T-PAUSE.acceptor.deliver.



### 5.3. Política de controle de erro

Para cada TC, a política de controle de erro consiste numa combinação entre a política de tratamento das TSDUs corrompidos e das TSDUs perdidos. A política de controle de erro é negociada apenas entre os usuários. Define-se como:

- perda - quando não chega ao destino o mesmo número de octetos.
- corrupção - quando o valor de um octeto, ou parte dele, está corrompido, diferente.

Tipos de políticas de TSDUs perdidos :

- a) Perda de (parte de) TSDUs não é aceita.
- b) Perda de (parte de) TSDUs aceita, mas indicada.

Tipos de políticas de TSDUs corrompidos :

- a) Corrupção de (parte de) conteúdo não é aceita.
- b) Corrupção de (parte de) conteúdo aceita mas indicada.
- c) Corrupção de (parte de) conteúdo aceita e não indicada.

| Política de Controle de Erro |                       | Perdas                            |   |
|------------------------------|-----------------------|-----------------------------------|---|
|                              |                       | Não aceitas                       | Aceitas e indicadas   |
| Corrupção                    | Não aceita            |                                   | Taxa de erro de TSDUs perdidas                                      |
|                              | Aceita, mas indicada  | Taxa de erro de TSDUs corrompidas | Taxa de erro de TSDUs corrompidas<br>Taxa de erro de TSDUs perdidas |
|                              | Aceita e não indicada |                                   | Taxa de erro de TSDUs perdidas                                      |

*Tabela 2 : As seis políticas de controle de erro possíveis e as taxas correspondentes*

#### 5.3.1. Taxa de erro de TSDUs corrompidas

A realização bem-sucedida de uma negociação de QoS provoca à ocorrência de um T-BIND.requestor.deliver na fonte e no destino. Para que uma taxa de erro seja mensurada, são feitas algumas definições :

- as TSDUs enviadas por um usuário durante a vida útil de uma TC formam um frame de TSDUs;
- as TSDUs são numeradas crescentemente começando em 0;
- um frame pode ser dividido em pedaços de N TSDUs que pertencem ao mesmo frame, no qual o número do primeiro TSDU do conjunto é múltiplo do inteiro N.

A taxa de TSDUs corrompidas pode ser definida como a razão entre  $N_1$  e  $N$  ( $N_1$  e  $N$  são inteiros, sendo  $N_1 < N$ ,  $N > 1$ ) onde  $N_1$  é o número de TSDUs, de um conjunto de  $N$  TSDUs submetidos por um usuário ao provedor, que são entregues ao usuário destino com uma indicação de que seu conteúdo (ou parte dele) foi corrompido. Essa taxa só é significativa se na política de controle de erro, as corrupções de TSDUs são aceitas e indicadas. O número inteiro  $N$  é escolhido durante as negociações de QoS, pelos usuários.

Valores significativos estabelecidos depois da negociação inicial:

- *threshold* - se alcançado, o provedor avisa os usuários através de um T-REPORT.acceptor.deliver e mantém a TC aberta;
- valor máximo - se alcançado, o provedor congela a TC usando um T-PAUSE.acceptor.deliver.

Se o provedor detectar que o valor de *threshold* ou o valor máximo foi alcançado, durante a entrega das TSDUs, ele pode avisar os usuários ou congelar a TC, respectivamente, sem que tenha que enviar todas as TSDUs que compõem o conjunto de N TSDUs. Se o número de TSDUs corrompidas for menor que o de *threshold*, o provedor não precisa tomar nenhuma ação especial.

### 5.3.2. Taxa de erro de TSDUs perdidas

Do mesmo modo descrito no item anterior, a taxa de erro de TSDUs perdidas é expressa pela razão entre M1 e M (M1 e M são inteiros,  $M1 < M$  e  $M > 1$ ) onde M1 é o número de TSDUs que são entregues ao usuário destino com uma indicação de que (parte de) seu conteúdo foi perdido num conjunto de M TSDUs submetidas por um usuário ao provedor. Essa taxa só é significativa, se na política de controle de erro negociada para a TC, as perdas são aceitas e indicadas. O número inteiro M é escolhido durante as negociações de QoS, apenas pelos usuários.

Valores significativos estabelecidos depois da negociação inicial:

- *threshold* - quando alcançado, o provedor avisa os usuários através de um T-REPORT.acceptor.deliver e mantém a TC aberta.
- Valor máximo - se alcançado, o provedor congela a TC usando um T-PAUSE.acceptor.deliver.

Da mesma forma que no tratamento de TSDUs corrompidas, se o provedor detectar que o valor de *threshold* ou o valor máximo foi alcançado, ele pode avisar aos usuários ou congelar a TC sem que tenha que reenviar todas as TSDUs que compõem o conjunto de M TSDUs. Caso o número de TSDUs perdidas for menor que o de *threshold*, o provedor não precisa tomar nenhuma ação especial.

### 5.3.3. Indicador de *Dummy replacement*

Esse indicador é válido apenas para TCs onde a política de controle de erros negociada prevê aceitar perdas de TSDUs desde que estas sejam indicadas. O usuário que vai receber os dados pode selecionar entre duas opções :

- substituição com caracter *dummy* - em todos os TSDUs recebidos através da primitiva de serviço T-DATA.acceptor.deliver (com um indicador de perda), os octetos perdidos serão substituídos por um valor padrão (denominado caracter dummy) pelo provedor;
- não substituição com caracter dummy - os octetos perdidos podem assumir qualquer valor.

#### **5.3.4. Character Dummy**

Pode ser escolhido numa faixa de valores entre 0 e 255 (representação em 8 bits). Não é um valor negociável, mas sim imposto pelo usuário que envia os dados.

#### **5.3.5. Proteção da TC**

Através desta funcionalidade o provedor pode evitar a monitoração ou a manipulação não autorizada de informações originadas pelos usuários. É negociada apenas entre os usuários da TS, sendo disponíveis quatro diferentes opções :

- sem características de proteção;
- proteção contra monitoração passiva;
- proteção contra modificação, repetição, adição ou remoção;
- as duas proteções anteriores juntas.

#### **5.4. Negociação de parâmetros de QoS na operação *BIND***

- Negociado entre provedor e usuários - *throughput*, atraso de trânsito e prioridades da TC.
- Negociado apenas entre usuários - taxa de erro de TSDUs (corrompidos ou perdidos), e proteção da TC.
- Não negociados - indicador de tipo de tráfego, indicador de substituição *dummy* e caracter *dummy*.

## Part IV - Exemplos de cenários de aplicações multicast

Working Group on New Transport Service Specification  
COST 237 - Multipeer scenario

### 1. Introdução

O propósito desse documento é reforçar as definições da taxonomia multicast proposta, com exemplos de aplicações em cenários tipicamente *multipeer*, mostrando as relações entre as definições e aplicações reais.

O grupo de trabalho da ISO (*working group*) utilizou como documento de base a especificação ISO/IEC JTC1/SC6 N9161- “Second Draft on Multipeer Taxonomy”. Como a especificação analisada foi o “Forth Draft on Multipeer Taxonomy”, algumas definições de taxonomia foram modificadas. Na avaliação aqui realizada, essas definições foram analisadas de acordo com a especificação mais recente.

Esse documento ilustra como a taxonomia pode ser usada para descrever uma coleção de sistemas distribuídos reais, onde estão presentes grupos de comunicação. Para uma melhor visualização dos termos da taxonomia, eles são escritos em negrito no decorrer do texto.

São apresentados exemplos e feita uma breve introdução. Basicamente procura-se identificar os grupos de entidades relevantes, interpretar o grupo no contexto do exemplo, definir o que são associações de grupo e conversações, etc. Existem várias formas de se descrever os exemplos existentes de acordo com a taxonomia atual. Por isso, optou-se por descrever duas interpretações diferentes para cada exemplo, em relação à taxonomia, o que seria equivalente a apresentar dois tipos de implementação para um dado fim a ser alcançado.

### 2. Aplicações

Foram escolhidas duas aplicações presentes no relatório gerado pelo projeto COST 237: Tele-palestra e Mail exploder.

#### 2.1. Tele-palestra

Um palestrante e um certo número de assistentes enviam tele-palestras a um grupo de estudantes. O curso se prolonga durante um período, e as aulas tem horário determinado. Podem ser feitas palestras, tutoriais, ou sessões de pergunta/resposta. Toda a interação entre os estudantes e o professor é feita através de tele-interação. Esse serviço é desenvolvido levando em conta que as estações de trabalho dos estudantes podem ser diferentes. Algumas podem aceitar apenas texto e áudio, outras texto, áudio, gráficos e vídeo.

### 2.1.1. Grupos

Inicialmente o grupo é composto do palestrante e seus assistentes, bem como de todos os alunos aptos a se inscrever no curso.

No início do período letivo, todos os estudantes qualificados a participar, juntamente com o palestrante e seus assistentes, fazem um *enroll* ao grupo. Alguns estudantes podem escolher não fazer o *enroll* ao grupo, se não estiverem interessados em participar das palestras. Nesse caso, o *enrolled group* é composto dos alunos interessados, o palestrante e seus assistentes.

As regras para participação no grupo podem ser descritas assim :

- O palestrante e os assistentes relacionados a esse curso na Universidade XYZ;
- todos os estudantes do 3o ano dessa Universidade.

Os critérios que definem o grupo não mudam, mas os membros do grupo podem mudar, havendo saída de membros ou entrada de novos participantes, desde que satisfaçam as regras de participação no grupo. Por exemplo, um novo professor assistente pode se tornar membro do grupo. Essa instância do grupo é então o *enrolled group*.

- Nome do grupo : “palestra cs96”; endereço do grupo : “123.xx.696”

A população do grupo é uma **população parcialmente conhecida**. O sub-grupo de membros que podem saber o endereço e nomes dos outros membros do *enrolled group* será a equipe de professores.

Obs: Nesse cenário também seria útil se pudessem ser formados subgrupos do grupo principal, correspondendo a grupos de tutoriais dos professores assistentes. Embora a taxonomia permita que os grupos sejam membros de outros grupos, seria realmente interessante que existisse uma maneira de decompor grupos em subgrupos, ao invés de compor grupos existentes em grupos agregados.

### 2.1.2. Associações de grupo

Depois de fazer o *enroll* ao grupo, o palestrante e os assistentes podem criar **associações de grupo (GAs)** de acordo com as suas necessidades.

Ao fazer um pedido de criação da GA (que será direcionada ao endereço do grupo), cada membro do *enrolled group* será notificado de que o pedido foi realizado, e pode escolher participar ou não na GA proposta. Por exemplo, apenas os estudantes pertencentes ao grupo do tutorial de determinado professor assistente escolheriam participar da GA criada por esse professor. Por outro lado, todos os estudantes, e

talvez os professores assistentes escolheriam participar da GA iniciada pelo palestrante com o propósito de enviar os horários das palestras.

A criação da GA só será efetivada se as várias **condições de estabelecimento e condições de integridade (ICs)** forem satisfeitas. Serão definidas as condições de **integridade do grupo ativo (AGI)**, e as condições de **integridade de topologia ativa (ATI)**. Uma AGI pode especificar, por exemplo, que mais de 5 estudantes concordam em participar da GA, enquanto a ATI pode especificar que um certo número de conversações com um nível mínimo de participação deve estar presente (incluindo obrigatoriamente o palestrante!).

#### *Interpretação 1*

As GAs são entidades persistentes, que duram um semestre inteiro. Todos os estudantes, o palestrante e os assistentes são **participantes** e estão no **grupo ativo**. Quando as ICs não forem satisfeitas, a GA será suspensa - não serão possíveis transferência de dados. Nesse caso está sendo utilizada a **soft IC management policy**. Quando a GA é suspensa, todas as suas conversações também são suspensas e os recursos que elas utilizam são liberados. Quando as ICs forem novamente satisfeitas (no início da próxima palestra), o estado anterior da GA será reestabelecido e as transferências de dados podem ocorrer novamente.

#### *Interpretação 2*

Uma GA é estabelecida a cada palestra. Apenas membros que queiram participar dessa palestra em particular respondem positivamente ao pedido de formação da GA. Eles são os **participantes** e formam o **grupo ativo**.

Novos grupos (*enrolled groups*) têm que ser criados para cada tutorial. O processo é equivalente ao descrito para as palestras.

### 2.1.2.1. Conversações

#### *Interpretação 1*

Em horários previamente estabelecidos para palestras ou tutoriais, o palestrante ou o assistente cria **conversações** na GA apropriada, para realizar a comunicação. O pedido será dirigido ao endereço da GA e não ao endereço global do grupo. Como na criação da GA, o pedido de conversação será enviado a todos os participantes dessa GA e aqueles que quiserem se unir a essa determinada conversação respondem afirmativamente ao pedido. Se o número de conversações e o número de estudantes satisfizer as condições de integridade - **AGI e ATI**, o dado pode ser transferido nas conversações, ou seja, a palestra pode começar. No final da palestra, seria suficiente (dependendo das condições mínimas de AGI a ATI) ao palestrante, simplesmente fechar as suas conversações para suspender a GA.

## Interpretação 2

As conversações são estabelecidas juntamente com a GA, no início de cada palestra. Conversações são realmente responsáveis pela transferência de dados, de acordo com cada topologia. Podem haver múltiplas conversações para diferentes tipos de mídia. Para que os estudantes possam participar ativamente, devem existir conversações deles para o palestrante e para seus colegas estudantes. Todas as conversações pertencem a uma mesma GA e a maioria delas é estabelecida e extinguida juntamente com a GA a que está associada. Uma GA não existe sem nenhuma conversação.

### 2.1.2.2. Elasticidade da Associação de Grupo

Pode ser necessário, num certo estágio do curso, que uma palestra seja dada por uma pessoa que não é membro do *enrolled group*. Para isso, o palestrante original deve fazer uma operação de *join invitation*, especificando o endereço do palestrante convidado (assumindo que o grupo ativo da GA é um grupo ativo **aberto** e não um grupo ativo **fechado**). Depois de realizada essa operação, o palestrante convidado poderá receber notificações de pedidos de conversação de outros participantes da GA.

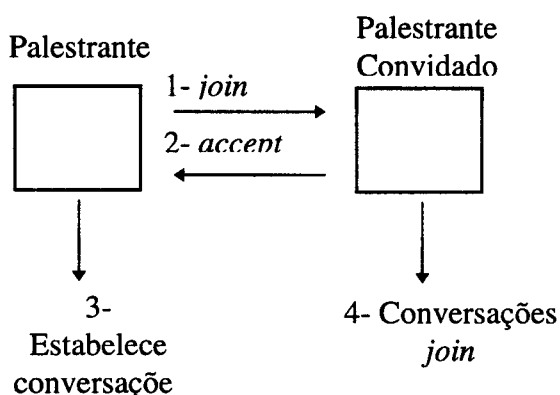


Figura 1 : *Join Invitation*

É possível aos participantes da GA fazer o *join* e o *leave* às várias atividades à nível do grupo e da GA durante a evolução do grupo. Por exemplo, estudantes que chegarem atrasados no semestre acadêmico, estudantes atrasados para uma determinada palestra já em andamento, etc.

## 2.2. Mail Exploder

Nesse caso, é mostrado como pode ser aplicada a taxonomia *multicast* num mail exploder. Pode ser entendido considerando-se um *mailing list* (quando uma mesma mensagem é enviada para uma lista de pessoas) como uma transmissão *multicast*, sendo o grupo as pessoas a que se pode enviar a mensagem, e a GA, as pessoas envolvidas no envio de uma mensagem em particular.

### 2.2.1. Grupo

O grupo define as pessoas que podem ser incluídas no *mailing list*. Pode ser definido como “todos”. Para que as pessoas que pertencem a esse grupo possam receber as mensagens do *mail exploder*, elas devem fazer um *enroll* ao grupo, formando o *enrolled group*.

O grupo é estático.

O grupo tem uma população parcialmente conhecida, pois o *mail exploder* sabe o endereço eletrônico de cada membro do *enrolled group*, mas cada membro, não sabe o endereço individual dos outros.

### 2.2.2. Associação de grupo

Um *mail exploder* transfere cada mensagem independentemente umas das outras. Isso significa que a ordem inicial das mensagens pode não ser mantida no recebimento e que cada mensagem é tem um identificador diferente.

Quando uma mensagem é enviada usando o endereço do grupo (o emissor pertencendo ou não ao grupo), não é dada nenhuma garantia quanto a quais entidades *enrolled* receberão a mensagem.

Nota: No estado atual, a taxonomia não permite que alguém que não pertence ao grupo envie uma mensagem. Portanto, o tipo de serviço de um *mail exploder* é tipicamente “sem conexão”.

Então, uma nova Associação de Grupo é utilizada para transferir cada mensagem. Essa GA não tem uma fase de *establishment e release* explícita e consiste apenas de uma mensagem. Não existem condições AGI ou ATI. O grupo ativo da GA é composto do emissor e dos membros do *enrolled group* que receberem a mensagem. O grupo ativo é então “aberto”, pois o emissor da mensagem não precisa estar necessariamente *enrolled*. Essa GA existe, portanto, apenas durante a transferência da mensagem, com um tempo de vida muito curto.



### 2.2.2.1. Topologia

Na hora em que a mensagem é enviada, a GA é composta de várias conversações: uma conversação entre o emissor e o *mail exploder*, e uma conversação entre o *mail exploder* e cada receptor.

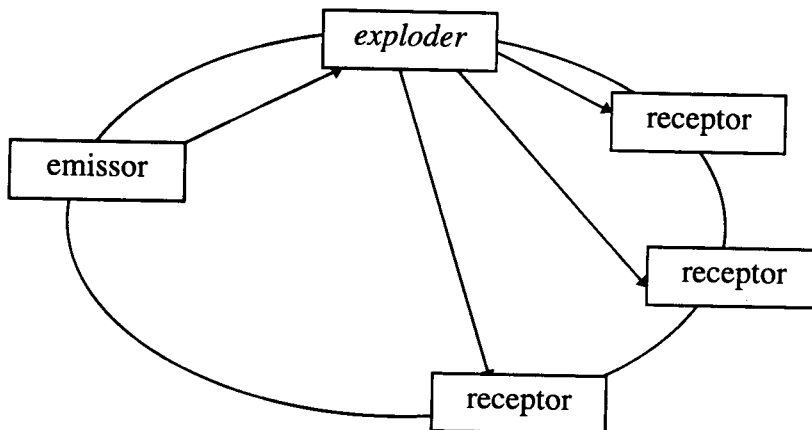


Figura 2: Topologia da Associação de Grupo

Quando é enviada a resposta a uma mensagem, ela pode ser enviada ao grupo e ao endereço individual do emissor (supondo que o emissor da mensagem não era *enrolled*). Nesse caso, a GA seria como descrito na figura.

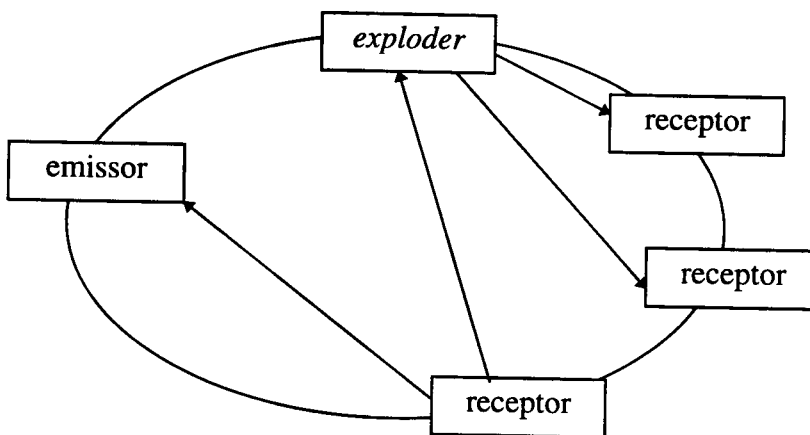


Figura 3: Topologia da Associação de Grupo usada na resposta

### 2.2.3. Outro conceito de um sistema de mensagens

No sistema de mensagens descrito anteriormente, é utilizado um *exploder*. Os mecanismos utilizados durante a distribuição da mensagem são ponto-a-ponto, não há comunicação direta entre o emissor e os receptores.

Porém, se um mecanismo multicast for incorporado a um sistema de mensagens, a distribuição de mensagens poderia ser feita sem a necessidade de um agente intermediário (o *exploder*).

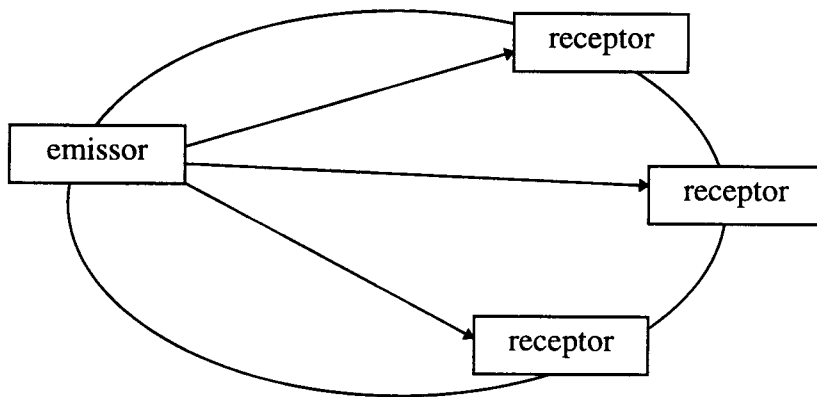


Figura 4: Associação de Grupo futura para *multicasting* de mensagens.

Além disso, a participação numa lista de mensagens poderia ser decomposta em uma fase de registro (em que seriam fornecidas informações relevantes) e uma fase de ativação, na qual o participante poderia decidir a temporariamente receber ou não suas mensagens (sem apagar as informações relativas a ele). Nesse caso, o *enrolled group* seria composto dos membros "ativados".

## Parte V - Conclusões

A integração das novas aplicações multimídias em um sistema de comunicação implica na definição de novos serviços adaptados as suas necessidades específicas. Essa integração deve ser realizada dentro de um contexto de trabalho que considere por um lado as novas características das aplicações, e por outro lado, as novas potencialidades dos sistemas de comunicações emergentes. Este trabalho visou principalmente investigar o atual estágio de desenvolvimento das novas propostas de serviços de transporte e sua real adequação ao binômio *novos requisitos de QoS & protocolos existentes*.

As principais tendências arquiteturais voltadas aos sistemas de comunicação multimídias, propõem uma distinção entre funcionalidades de transporte e funcionalidades de sincronização. Por outro lado, faz-se igualmente a distinção entre nível de sincronização e nível de orquestração. O primeiro disponibiliza funções de controle das relações de ordem, causalidade e temporal que vão possibilitar uma restituição coordenada das informações, durante a transferências de vários fluxos de dados. O nível de orquestração agrupa as funções de gerência de conexões e de associações mutipartes (coordenação de uma sessão multimídia), bem como as funções de controle dinâmico de grupo hierárquicos.

O nível de sincronização repousa sobre o que é normalmente denominado de subsistema de comunicação, sendo representado pelo nível de transporte. O papel de uma camada de transporte é de aproveitar as novas propriedades das informações multimídias (por exemplo a perda de alguns quadros durante a transferência de um fluxo de vídeo não implica necessariamente em conseqüências prejudiciais a qualidade da restituição) para construir mecanismos protocolares mais eficientes que suportem tanto os requisitos de alta vazão como os requisitos de atrasos limitados. Por sua vez, a sincronização e a orquestração levam em consideração os novos requisitos dos usuários tais como a sincronização intra- e inter-fluxo, bem como as manipulações temporais. De acordo com os documentos analisados, verifica-se que os organismos de padronização não seguiram esta tendência arquitetural, aglutinando todos estes novos requisitos funcionais na camada de transporte. Por exemplo, a taxonomia proposta pela ISO cita a necessidade de criar uma nova funcionalidade de transporte que cuidaria da sincronização entre diversas conversações, sendo esta funcionalidade denominada de orquestração.

De uma maneira geral, pode-se dizer que existe uma preocupação nítida em contemplar alguns requisitos de qualidade de serviço impostos pelas novas aplicações, mas sem que haja um comprometimento explícito com a viabilidade de desenvolvimento dos novos mecanismos necessários (entende-se por mecanismos, o conjunto de algoritmos, protocolos e recursos necessários a uma transferência de informações que seja capaz de atender a qualidade de serviço almejada). Muitas das funcionalidades definidas pelas propostas de padronização ainda não encontram soluções devidamente maduras. Para exemplificar esta questão, observa-se que no item 6.3 da parte II do documento (Modelo de transmissão), onde é abordada a questão da ordem de entrega das SDUS, já existe a preocupação em considerar aspectos de causalidade e ordem parcial. Por outro lado alguns requisitos funcionais mais complexos, tais como relações temporais entre os participantes de uma

comunicação em grupo (exemplo: um fluxo de dados pode ser transferido isocronamente para vários receptores sem que haja nenhuma espécie de sincronismo entre os receptores) não são explorados nas normas.

A conclusão fundamental deste estudo das propostas de padronização é que a maior contribuição refere-se às propostas de serviços para a criação e gerência de associações multipartes. Neste sentido foi desenvolvido um novo modelo bastante aprimorado, que deve influenciar substancialmente a definição de novos serviços de transporte.