# Adaptando as Especificações FT-CORBA para Redes de Larga Escala

Lau Cheuk Lung, Joni da Silva Fraga, Ricardo Padilha, Luciana Souza
Laboratório de Controle e Microinformática (LCMI) – DAS – CTC – UFSC
Campus Universitário – Caixa Postal 476 – Trindade - CEP 88040-900 – Florianópolis – SC – Brasil
e-mail: {lau, fraga, padilha, luciana}@lcmi.ufsc.br

**Resumo**

Este trabalho apresenta uma proposta de extensão das especificações Fault-Tolerant CORBA [OMG00] para sistemas distribuídos de larga escala. A motivação deste trabalho foi à inadequação ou a falta de definições nas especificações FT-CORBA que permitissem atender a requisitos de tolerância a falta para sistemas de larga escala, tal como a Internet. Neste trabalho, é apresentado um conjunto de soluções, englobando detecção de falhas, *membership* e comunicação de grupo, que visam principalmente atender aspectos de escalabilidade, necessários quando são tratados sistemas desta natureza. A principal contribuição deste trabalho é a proposta de um modelo de hierarquia de domínios de tolerância a faltas que facilita o gerenciamento e comunicação de grupo interdomínio.

***Palavras chave: tolerância a faltas, sistemas distribuídos, membership, comunicação de grupo e CORBA .***

**Abstract**

This work presents a proposal to extend the Fault-Tolerant CORBA specifications [OMG00] to large-scale distributed systems. Our motivations for this proposal are that FT-CORBA does not assist, properly, to the requirements of fault-tolerant in large-scale networks, such as Internet. In this paper, we present a set of solutions (including failure detection, membership, group communication) that aim, mainly, to treat scalability aspects in large-scale network. The main contribution of this paper is the proposal of a model of hierarchy to fault tolerance domains, which facilitates the management and communication for inter-domain groups.

***keywords: fault tolerance, distributed systems, membership, group communication and CORBA .***

## 1. Introdução

Muitas aplicações distribuídas estão seguindo o paradigma de orientação a objetos e considerando o CORBA (*Common Object Request Broker Architecture*) [OMG96] como a melhor alternativa de se adequar a sistemas abertos. No entanto, as especificações iniciais do CORBA [OMG96] não contemplavam requisitos de tolerância a faltas, fundamentais em sistemas distribuídos. Isto motivou vários grupos de pesquisa no sentido de propor extensões para adicionar mecanismos de tolerância a faltas na arquitetura CORBA [Maffeis95, Felber98, Chung98, Moser98, Lau99]. O amadurecimento dessas pesquisas e a crescente demanda por aplicações confiáveis em sistemas distribuídos foram determinantes para que a OMG resolvesse então, lançar um edital convidando empresas e instituições de pesquisa a submeterem propostas no sentido de introduzir tolerância a faltas no CORBA. Como resultado disso, após alguns anos de trabalho, foi publicada então, as especificações do *Fault-Tolerant* CORBA (FT-CORBA) [OMG00]. Essas especificações, que ainda devem passar por varias revisões (e extensões), definem um conjunto de interfaces de serviços e facilidades úteis para a implementação de técnicas de replicação em ambientes distribuídos heterogêneos. As especificações atendem apenas aos requisitos básicos para tolerância a faltas, definindo algumas interfaces bastante genéricas, de fácil entendimento, e úteis em praticamente todas aplicações que requerem tolerância a faltas em sistemas distribuídos.

Fazendo uma análise detalhada das especificações FT-CORBA, é verificado que muitos requisitos necessários para tolerância a falta em sistemas de larga escala (tal como a Internet) ainda não foram abordados de forma precisa. A principal dificuldade é devido às características assíncronas[1] dos sistemas de larga escala – as especificações ainda não apresentam abstrações objetivas para esta classe de sistema [OMG00]. Especificamente, ainda não foram discutidas soluções para: detecção de falhas em sistemas assíncronos, gerenciamento de replicação (*membership*) e comunicação de grupo em um contexto de redes de larga escala e,

[1] Os sistemas de larga escala, inerentemente assíncronos, são caracterizados pela grande distribuição espacial, com um caráter aberto, integrando quantidades significativas de recursos computacionais, assinalados pela sua heterogeneidade. Não são deterministas no sentido em que os parâmetros temporais, tais como padrões de tráfego, taxas de transferência, atrasos de comunicação e diferentes velocidades de processamento dos processadores, são dependentes das condições de carga e evolução dinâmica desses sistemas e, portanto, não são conhecidos *a priori*.

principalmente, como estas soluções podem ser integradas ou construídas a partir da estrutura do FT-CORBA sem que isto represente modificar quaisquer interfaces já padronizadas pela OMG.

Neste trabalho é apresentada uma proposta de adequação das especificações dos serviços de detecção de falhas e gerenciamento de replicação do FT-CORBA para atender aos requisitos de redes de larga escala. Esta proposta está fundamentada em um estudo realizado que envolveu a definição de um modelo combinando o serviço de gerenciamento de replicação com um serviço de nomes hierarquizado. Este modelo nos permitiu, então, a definição de um protocolo de detecção de falhas assimétrico (serviço dedicado) e de um suporte de comunicação de grupo, próprios para sistemas de larga escala. A proposta visa apresentar soluções, baseadas nas especificações FT-CORBA, para o gerenciamento de replicação e a difusão atômica de mensagens – seriamente dificultados pelos custos e pela falta de escalabilidade das soluções usuais destes problemas quando tratamos com redes de larga escala.

Na literatura, são encontrados diversos trabalhos envolvendo a disponibilidade de serviço de comunicação de grupo na arquitetura CORBA. Estas experiências (classificadas em três abordagens: *integração* [Maffeis95, ISIS95], *serviço* [Felber98] e *interceptação* [Fraga97, Moser98, Lau00a]) se preocupam basicamente em manter características de interoperabilidade, portabilidade e de desempenho. Não existindo, portanto, propostas voltadas para um contexto de larga escala ou mesmo, uma discussão mais detalhada de adaptação dos conceitos da especificação FT-CORBA para estes ambientes.

O artigo apresenta na seção 2 uma descrição sucinta das especificações de tolerância a faltas no CORBA. As especificações FT-CORBA são discutidas considerando requisitos de escalabilidade na seção 3. Na seção 4, são apresentados o *GroupPac* e a sua proposta gerenciamento de replicação e comunicação de grupo em redes de larga escala. Aspectos de configuração no item 5. Considerações gerais são feitas na seção 6. Finalmente, na seção 7, são levantadas as conclusões deste trabalho.

## 2. A Especificação *Fault-Tolerant* CORBA

Recentemente, a OMG publicou uma especificação para introduzir tolerância a faltas na arquitetura CORBA (FT-CORBA) [OMG00]. Estas especificações definem um conjunto de serviços essenciais para o desenvolvimento de aplicações confiáveis em sistemas abertos. A tolerância a faltas no CORBA é obtida através da replicação de objetos, técnicas de detecção e recuperação de falhas. Nessa primeira especificação foi apresentado um conjunto de interfaces e protocolos que podem ser separados em três módulos: Gerenciamento de Replicação (SGR); Gerenciamento de Falhas (SGF) e Gerenciamento de Recuperação e *Logging* (SLR); além das definições para Interoperabilidade próprias da arquitetura CORBA. O *Serviço de Gerenciamento de Replicação* é constituído pelos serviços de gerenciamento de propriedades de tolerância a faltas, gerenciamento de grupo de objeto e fábrica genérica. O SGR é responsável pelo gerenciamento de grupo, exercendo um controle dinâmico, nas entradas e saídas (normal ou por falha) de objetos replicados. Para isso, o SGR tem o auxílio do *Serviço de Gerenciamento de Falhas*. No SGF são definidas as interfaces dos serviços de detecção de falhas, notificação de falhas e análise de falhas. Estes três serviços são responsáveis pelo monitoramento e notificação de falhas de objetos de grupo. Finalmente, no *Serviço de Gerenciamento de Recuperação e Logging* são definidos os mecanismos para transferência de estado de objeto e recuperação de réplicas faltosas.

A OMG, como é de sua característica, se limita em definir interfaces de serviço genéricas, tentando atender diferentes abordagens de tolerância a faltas. Por exemplo, protocolos de comunicação de grupo confiável, base fundamental para a implementação de técnicas de replicação em sistemas distribuídos, não são padronizados – como era de se esperar, pela complexidade dos mesmos e pela quantidade de algoritmos envolvidos. A OMG enfatiza, neste caso, o uso de soluções proprietárias. Todavia, para garantir um mínimo grau de

interoperabilidade, os sistemas de comunicação de grupo devem adotar a IOGR (*Interoperable Object Group Reference*): uma referência interoperável de grupo de objetos definida em [OMG00]. A IOGR corresponde a uma extensão da IOR (*Interoperable Object Reference*), referência a um objeto simples. Uma IOGR permite a um cliente referenciar um grupo de objetos como uma entidade única. De forma genérica, esta referência de grupo contém, em seus campos, a IOR de cada membro de um grupo de objetos.

## 3. Limitações de Escalabilidade nas Especificações FT-CORBA

O objetivo desta seção é verificar nos serviços de gerenciamento de replicação e de detecção de falhas, das especificações FT-CORBA, quais os pontos que ainda não atendem adequadamente a requisitos de escalabilidade. Além disso, são discutidos aspectos relacionados ao suporte de comunicação de grupo, que ainda não é tratado, explicitamente, nas especificações atuais do FT-CORBA. Neste caso, o enfoque sobre este aspecto será também para sistemas de larga escala.

### 3.1. Gerenciando Replicações em Domínios de Tolerância a faltas

Nas especificações FT-CORBA foi introduzido o conceito de *Domínios de Tolerância a faltas* para facilitar o gerenciamento de grupos de objetos distribuídos em uma rede. Cada domínio pode conter diversos grupos de objeto (figura 1), cada grupo possuindo suas próprias propriedades de tolerância a faltas. É definido, também, que cada domínio tem disponível um conjunto de serviços do FT-CORBA (gerenciamento de replicação, gerenciamento de falhas e gerenciamento de recuperação e *logging*), os quais atuam, exclusivamente, dentro deste domínio de tolerância a faltas. Portanto, um domínio é formado por grupos, que se apresentam dispostos em *hosts*, formados por processos e objetos. Segundo as especificações, um objeto não pode pertencer a mais de um grupo e domínio de tolerância a faltas.

Os objetos pertencentes a um domínio não são, necessariamente, limitados a uma rede local (LAN – *Local Area Network*), ao invés disso, podem estar distribuídos em uma rede metropolitana (MAN – *Metropolitan Area Network*) ou ainda, em uma rede de longa distância (WAN – *Wide Area Network*). Essas diferentes opções de rede em que um domínio pode se estender implica, certamente, na definição de diferentes propriedades de tolerância a falta.

Uma das funções do serviço de gerenciamento de replicação é criar as referências de grupos (IOGR - *Interoperable Object Group Reference*) do domínio de TF correspondente. No entanto, estas IOGRs não estão disponíveis a partir deste serviço. A alternativa plausível é o uso do serviço de nomes (*CosNaming*) padrão OMG [OMG97] para guardar todas as IOGRs do domínio geradas pelo serviço de gerenciamento de replicação – cada domínio de TF teria, então, o seu próprio serviço de nomes. A especificação do *CosNaming* foi recentemente revisada com o objetivo de tratar aspectos de interoperabilidade, definir um formato URL para nomes e estender as funcionalidades para ligar contextos de nomes. O serviço de nomes seria, portanto, um "portão de acesso" para todos objetos e grupos de objetos de um domínio.

As especificações FT-CORBA, ao limitar grupos a um domínio de TF específico, não se mostram, pelas suas abstrações, apropriadas para sistemas distribuídos de larga escala. É claro que seria sempre possível definir um domínio de tolerância a faltas que comporte todos as réplicas de um grupo em um ambiente de larga escala. No entanto, devido às características assíncronas nestes sistemas, envolvendo grandes distâncias geográficas, as dificuldades para gerenciar grupos seriam enormes.

### 3.2. Detecção de Falhas em um Domínio de TF

Segundo o FT-CORBA, todos os *hosts*, contendo objetos de grupos de um domínio de TF, devem ser monitorados por pelo menos um detector de falha. Os detectores de falhas de um domínio podem ser replicados atuando sobre os mesmos *hosts* ou os mesmos objetos. Mas situações diversas podem ser admitidas: detectores podem não monitorar, necessariamente, os

mesmos conjuntos de *hosts* ou de objetos do domínio. Um domínio de tolerância a faltas pode apresentar detectores monitorando diferentes subconjuntos de *hosts* do domínio. Neste último caso não há tolerância a possíveis falhas de detectores no domínio.

Além disso, detecção de falhas em sistemas assíncronos é uma classe de problemas onde só são permitidas soluções probabilistas [Fisher85, Ricciardi91, Chandra96]. As especificações FT-CORBA fornecem alguns recursos, que nas suas definições, não são bem adequados para esta classe de problemas. É o caso do monitoramento de um *host* a partir de um conjunto de detectores de falhas. Neste caso, o *host* é assumido como *crash* quando não responde, dentro de *timeouts* estipulados, a pelo menos um dos detectores do domínio. Se considerarmos as características assíncronas dos sistemas de larga escala este procedimento é extremamente penalizante. No entanto, se associarmos algumas semânticas aos detectores de falhas, as condições de detecção podem ser melhoradas na precisão[2].

### 3.3. Comunicação de grupo no modelo CORBA

As especificações atuais do FT-CORBA ainda não definem um serviço de comunicação de grupo, fundamental na implementação de qualquer técnica de replicação em sistemas distribuídos. O problema atual esta na grande variedade de protocolos de comunicação de grupo, cada um apresentando soluções específicas para atender aos requisitos de determinadas aplicações (ex: processamento replicado, base de dados distribuída, aplicações de *groupware*, etc). Se um padrão fosse definido, na atual especificação, para esse serviço, provavelmente não seria uma solução capaz de atender aos requisitos de toda gama de aplicações distribuídas que podem fazer uso de comunicação de grupo.

A OMG está tratando deste problema cuidadosamente, tanto que a maior parte dos membros do grupo responsável pela padronização do FT-CORBA está participando de outras RFPs que estão diretamente relacionadas com tolerância a faltas no CORBA. Por exemplo, recentemente, foi lançada uma RFP (*Request for Proposal*) no sentido de especificar um padrão para difusão não confiável de mensagem (*Unreliable Multicast*) [OMG00a] para ser incluída no CORBA. Considerando as experiências a serem obtidas nessa força tarefa, essa RFP seria, de certa forma, o primeiro passo para futuras RFPs, no sentido da concepção de um ou mais serviços de comunicação de grupo com garantias mais restritivas de acordo e ordenação.

Por outro lado, em sistemas de larga escala, grupos podem ser formados por uma quantidade considerável de objetos espalhados geograficamente. Um exemplo típico desse tipo de sistema seria uma base de dados de uma corporação com réplicas espalhadas por todo planeta no sentido de facilitar o acesso às suas informações. Um novo dado inserido em uma réplica da base de dado teria que ser difundida atomicamente para as outras réplicas do grupo para assegurar a consistência. Embora nas especificações atuais sejam recomendadas ferramentas proprietárias para a comunicação de grupo, o FT-CORBA, através das IOGRs, limitam grupos a um domínio de TF específico, dificultando o uso destas ferramentas em redes de larga escala. As ferramentas usuais de comunicação de grupo pecam pela falta de escalabilidade [Fritzke98].

## 4. *GroupPac*

O *GroupPac* [Lau99, Lau00a] consiste de um conjunto de serviços específicos para o desenvolvimento de aplicações tolerantes a falhas. Adere as recomendações da OMG como objetos de serviços, tais como as especificações COSS [OMG97]. Os serviços do *GroupPac* têm suas interfaces baseadas nas especificações FT-CORBA (ou seja, nas especificações dos serviços de gerenciamento de replicação, gerenciamento de falha, *logging* e recuperação). A

[2] Em [Chandra96], são definidas várias semânticas para detectores de falhas não confiáveis a partir de duas propriedades: *completitude* (“*completeness*”) e a *exatidão* (“*accuracy*”). A completitude especifica que um detector de falha terminará por suspeitar de todos processos que realmente estiverem falhos. E a exatidão restringe os enganos (suspeitas errôneas) que um detector possa cometer.

diferença em relação ao padrão é que apresenta um conjunto de extensões e adaptações no sentido de também atender a requisitos de tolerância a faltas em sistemas de larga escala.

As soluções, propostas neste trabalho, envolvem a definição de um modelo de estruturação de sistemas distribuídos onde a composição de domínios de TF e de serviços de nome aporta a escalabilidade necessária em sistemas de larga escala. A adoção deste modelo tem implicações nos serviços de gerenciamento de replicação e de gerenciamento de falhas e na comunicação de grupos. A comunicação de grupo é disponibilizada no *GroupPac* usando o conceito de interceptadores e encapsulando ferramentas proprietárias na forma de serviços de objeto[3] [Lau00a]. Propriedades de ordenação nas comunicações dependem da ferramenta usada – no caso foi usado o ISIS [Birman91].

Estas soluções são adaptadas ao CORBA de forma transparente para aplicação, e sem que isto represente qualquer modificação das interfaces do padrão. É também assumido, nestas soluções, em nível de transporte, canais de comunicação ponto-a-ponto confiáveis, servindo de suporte aos objetos se comunicando via CORBA por troca de mensagens. Por sua vez, como hipótese de falhas se assume objetos (*hosts*) falhando por *crash* e serviços de aplicação sendo restaurados através dos serviços recuperação do FT-CORBA.

### 4.1. Hierarquia de Domínios e a Escalabilidade

A literatura indica que a forma mais adequada para tratar a complexidade de sistemas de larga escala é a decomposição hierárquica do problema [Rodrigues96, Fritzke98]. Em um contexto de sistema de larga escala com vários grupos apresentando objetos replicas dispersos geograficamente, o *GroupPac* introduz então um modelo de gerenciamento e comunicação de grupo baseado em uma hierarquia de domínios de tolerância a faltas.

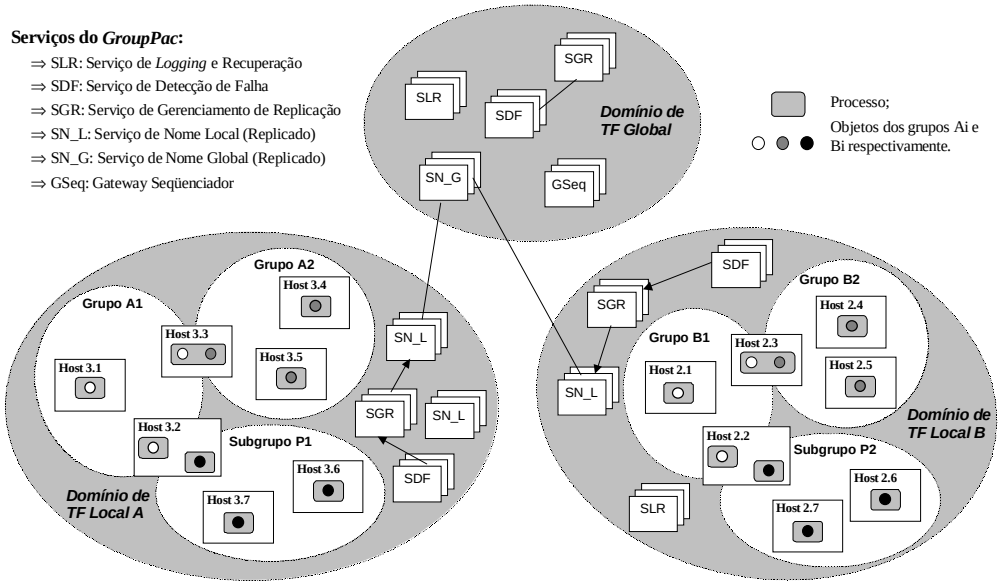


Figura 1. Modelo hierárquico para sistemas de larga escala.

Neste modelo, os domínios podem conter grupos onde todas as suas réplicas são gerenciadas e se comunicam dentro do próprio domínio. Estes grupos são identificados no *GroupPac* como *Grupos Intradomínios*. Um grupo, que tenha componentes dispersos em uma rede de larga escala, teria estes componentes compondo subgrupos distribuídos entre vários domínios. Isto é, ao invés de ter um único domínio para gerenciar um grupo largamente distribuído teremos então, um conjunto de domínios gerenciando partes deste grupo (seus

[3] Um interceptador é um mecanismo de ORB interposto entre um cliente e um objeto servidor (no nosso caso um grupo de objetos) cujo propósito é desviar de forma transparente a requisição de serviço. O interceptador então ativa o objeto que encapsula a ferramenta de comunicação de grupo, provocando a difusão da requisição de serviço no grupo de objetos servidores.

subgrupos) de forma separada. Estes grupos envolvendo membros em vários domínios são identificados como *Grupos Interdomínios*.

Na figura 1, os grupos *A1* e *A2* contidos no domínio *A* são exemplos de grupos intradomínios. O grupo *P*, formado pelos subgrupos *P1* (domínio *A*) e *P2* (domínio *B*), corresponde a um grupo interdomínios. Na hierarquia de domínios de tolerância a faltas da figura 1, são apresentados dois níveis de gerenciamento:

- Domínios de Tolerância a faltas Local ou inferior: pode conter e gerenciar grupos locais que não envolvam grandes distribuições físicas (grupos intradomínios) e também subgrupos de grupos interdomínios. Na figura 1, o domínio local A gerencia e contém o grupo A1 e o subgrupo P1;
- Domínio de TF Global: é o domínio de gerenciamento global. Comportam, exclusivamente, o conjunto de serviços responsável por garantir a interação entre os domínios de tolerância a faltas locais. Esse domínio não comporta grupos ou subgrupos de objetos de aplicação.

De forma sucinta, cada domínio possui sua própria infraestrutura para tolerância a faltas (constituído pelo SLR, SDF, SGR, SN_L, SN_G e GSeq, ver figura 1) suportada pelo *GroupPac*. Serviços *GroupPac* de um domínio operam somente sobre os grupos e subgrupos contidos no domínio de tolerância a faltas. O serviço de detecção de falhas (SDF) é o responsável por monitorar os *hosts* do seu domínio de tolerância a faltas. Eventuais falhas de *host* são detectadas pelo SDF que notifica ao serviço de gerenciamento de replicação (SGR) do domínio para que estabeleça uma nova IOGR (com uma nova lista de membros). As IOGRs dos grupos replicados do domínio são, então, enviadas ao serviço de nome local replicado (SN_L) para que este as disponibilize no sistema. O SN_L contém as IOGRs (e IORs) do domínio ao qual pertence. Os grupos SDF, SGR e SN_L podem ser replicados em qualquer número e estar localizados em qualquer *host* do domínio. Finalmente, o *Serviço de Nomes Global* (SN_G) faz a ligação de todos os SN_L – permitindo, por exemplo, que objetos de um domínio possam localizar grupos de objetos de outros domínios de tolerância a faltas. No decorrer desta seção, serão apresentados mais detalhes sobre cada um destes serviços.

Cada grupo de objeto pode ser uma aplicação replicada distinta, disponível como entidade lógica única, segundo o modelo de objetos CORBA (cliente/servidor). Os grupos contidos dentro de um domínio serão regidos de acordo com as propriedades de tolerância a faltas definidas no serviço de gerenciamento de propriedades de cada domínio de tolerância a faltas.

Grupos interdomínios, por apresentarem seus subgrupos dispostos em diferentes subgrupos, implicam em um gerenciamento mais simples e adequado às necessidades de escalabilidade. Esta separação permite, por exemplo, que cada domínio tenha seus próprios protocolos de gerenciamento e comunicação de grupo – estabelecidos de acordo com as características do ambiente sobre o qual está disposto. As mudanças de *membership* são mais fáceis de tratar com esta decomposição em domínios com subgrupos. No entanto, esta decomposição não é uma solução trivial, implicando na definição de um conjunto de suportes e extensões nas especificações FT-CORBA que mostraremos ao longo desta seção.

### 4.2. O Serviço de Detecção de Falhas do *GroupPac*

Conforme discutido, no item 3.2, a detecção de falhas do FT-CORBA, na forma como foi especificada, não é adequada para sistemas com características assíncronas. Para tratar com as dificuldades destes ambientes, estendemos a noção de detectores de falhas do FT-CORBA, sem que isto represente qualquer modificação das interfaces dessa especificação. Assumimos então que um *host* é considerado em *crash* se não responder a um certo número de detectores de falhas dentro de *timeouts* específicos. É necessário então um procedimento de consenso para ser executado por um conjunto de detectores na "determinação" de falhas. O monitoramento de um *host* a partir de um grupo de detectores minimiza a probabilidade de erros de detecção.

A solução que adotamos neste trabalho foi especificar um protocolo baseado em voto majoritário para alcançar consenso na decisão sobre as falhas de objetos [Lau00b]. Diferente de

[Chandra96], em que o módulo detector é agregado a cada processo da aplicação, a nossa solução, para melhor se adequar à especificação do serviço de detecção de falhas do FT-CORBA, propõem um conjunto de objetos detectores dedicados, dispostos no domínio, podendo ou não estar nos mesmos *hosts* de objetos de aplicação. No nosso esquema todos detectores realizam as mesmas funções, ou seja, monitoram todos os *hosts* dentro de seu domínio de TF. A figura 2 exemplifica este esquema de detecção. Os detectores de falhas que se utilizam do algoritmo definido em [Ricciardi91] para o consenso podem ser classificados como *Perfeitos* (classe *P* [Chandra96]). Assumimos, portanto, que os detectores são sempre completos, após um certo tempo suspeitará de todo processo permanentemente faltoso ou desconectado dentro do domínio de tolerância a faltas.

O serviço de detecção de falhas do *GroupPac* é concretizado em dois níveis de monitoramento: nível de detectores e nível de *hosts*. No primeiro, os detectores de *host* formam um grupo autogerenciável. Isto é, controlam as entradas e saídas (por falha) dos detectores membros do grupo. O monitoramento dos detectores é necessário para indicar o número de membros no grupo para, assim, definir a maioria de detectores durante um processo votação.

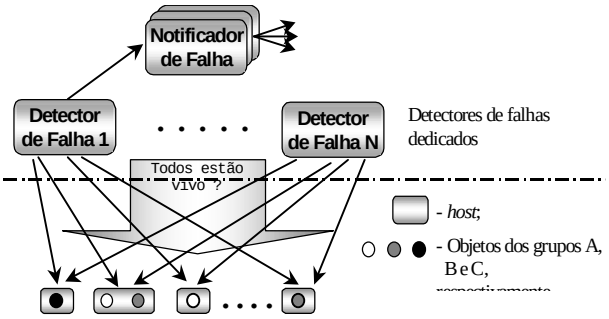


Figura 2. Monitoramento de falhas no GroupPac.

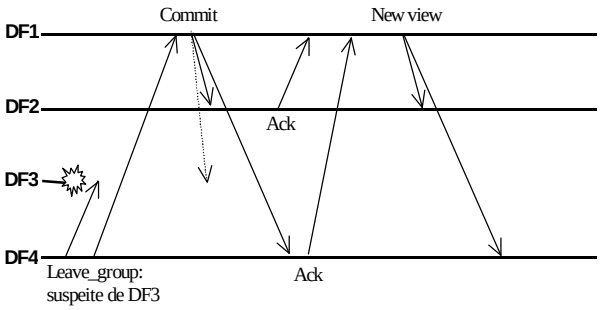


Figura 3. Situação: crash da réplica R2.

O monitoramento em nível de detectores usa um protocolo de *commit* centralizado (no detector primário), de duas fases (três fases no pior caso), para alcançar acordo (*agreement*) com os outros membros em relação à nova composição do grupo de detectores de falhas (DF). Este protocolo é baseado em [Ricciardi91] e foi implementado em [Lau99]. Os membros deste grupo compõem um anel virtual e, periodicamente, cada membro monitora o parceiro imediatamente anterior da seqüência do anel. A coordenação na obtenção de uma nova lista de membros (*new view*) é centralizada no objeto detector de falha primário DF1. A figura 3 mostra um exemplo de funcionamento desse protocolo. O objeto DF4 ao detectar um possível *crash* de DF3 envia a mensagem "suspeite de DF3" para o objeto primário DF1. Neste ponto, o protocolo de *membership* é ativado e o primário difunde uma mensagem *commit* para detectar quem ainda continua no grupo, os detectores que estão ativos devem dar um reconhecimento a esta mensagem (*Ack*). Após um prazo de espera pré-definido, o primário DF1 produz uma nova lista de membros a partir dos *Ack* recebidos (que deve constituir uma maioria em relação ao *view* anterior) [Ricciardi91]. Caso ocorra vários particionamento no grupo, a partição majoritária permanece ativa. As minoritárias são desativadas e seus membros reintegrados como novos na partição majoritária. Quando o detector de falhas primário (DF1) é suspeito de ter falhado; o candidato para substitui-lo segue a ordem implementada pelo anel virtual [Ricciardi91, Lau99]. Assumimos um canal de comunicação ponto-a-ponto confiável e não bloqueante entre os membros do grupo SDF. Portanto, a falta de uma mensagem esperada não pode ser devido à sua perda no meio de comunicação, mas pode, ao invés disso, ser um indicativo de que algo errado está ocorrendo com o emissor.

O outro nível de detecção de falhas no *GroupPac*, nível de *hosts*, tem então o grupo de detectores de falhas ($DF_1$, $DF_2$ e $DF_3$ da figura 3) atuando nos *hosts* do domínio considerado. Na falha de um *host* são considerados como falhos todos processos e objetos deste *host*. Os DFs monitoram periodicamente, dentro de um intervalo T(s), os mesmos *hosts* de um domínio de

tolerância a falta. São eles que decidem por maioria se um determinado *host* é faltoso (*crash*) ou não. Caso qualquer um dos detectores suspeite de uma falha de um *host*, o protocolo executa um procedimento, baseado em voto majoritário, para alcançar consenso. A coordenação deste procedimento de consenso é também centrada no detector *primário* na ordenação do anel.

A figura 4 apresenta uma instancia desse protocolo, o DF2 suspeita do *host* H2 e avisa ao DF1 (o primário) sobre esta suspeita. A partir daí é iniciado o protocolo, em três passos, para o consenso e a sinalização da falha de H2. No primeiro passo, o DF1 solicita aos outros detectores (DF2 e DF3 da figura 4) para que no próximo intervalo de monitoração (T(s)) o informe sobre o status (vivo ou falho) do *host* H2. No segundo passo, após a nova monitoração, cada detector informa ao DF1 sobre sua posição em relação ao status de H2. Finalmente, no passo três, o detector de falha primário (DF1), a partir dos resultados, decide sobre o status de H2. Então, o DF1, através do notificador de falhas (NF da figura 4), informa ao gerenciador de replicação para que este gere uma nova IOGR (item 2), removendo H2 da lista. A nova IOGR é enviada ao grupo de detectores para que estes atualizem suas listas de *hosts* a serem monitorados.

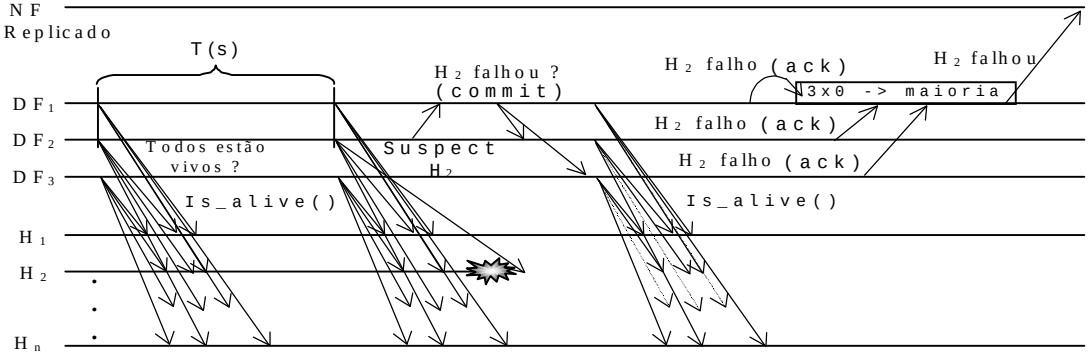


Figura 4. Uma instância do protocolo de detecção de falhas.

### 4.3. Localizando Grupos de Objetos no *GroupPac*

As soluções propostas no *GroupPac* são direcionadas no sentido de facilitar a escalabilidade. Buscamos uma forma de permitir que objetos, dos diferentes domínios de TF, pudessem ser localizados e se juntar ou sair de um grupo dinamicamente, durante o ciclo de vida da replicação. Isto é conseguido através da associação de domínios de TF com o serviço de nomes – é importante ressaltar que a infraestrutura do FT-CORBA não possui uma interface para deixar disponível IOGRs a objetos clientes [OMG00]. O serviço de nomes é então importante na localização destas IOGRs de grupos intradomínios (grupos *A1*, *A2*, *B1* e *B2* da figura 1) como de grupos interdomínios (subgrupos *P1* e *P2* do grupo *P*).

A composição do serviço de nomes mais domínios de TF, proposta no *GroupPac*, permite que objetos de um domínio localizem grupos e subgrupos de outros domínios de tolerância a faltas. O serviço de nomes no *GroupPac* é também baseado em uma hierarquia para facilitar o trato da complexidade de sistemas de larga escala. Esta hierarquia é formada com a introdução de um serviço de nomes global, replicado por questões de disponibilidade e de tolerância a faltas (grupo SN_G, na figura 1). Esta hierarquia, apresentada na figura 1, divide o serviço de nomes dividido em dois níveis:

- Serviço de Nomes Replicado Local (SN_L): responsável por gerenciar todas as IOGRs (e IORs) dos objetos da aplicação de um domínio de tolerância a faltas. O SN_L possui também, em seu contexto de nomes, uma cópia da IOGR do SN_G;
- Serviço de Nomes Replicado Global (SN_G): responsável por gerenciar a IOGR de cada SN_L de domínio – possui cópias atualizadas das IOGRs dos SN_L de todos os domínios registrados.

O grupo SN_G que faz parte do domínio de TF global permite o acesso a grupos ou subgrupos de um domínio de tolerância a faltas a partir de outros domínios. No modelo apresentado na figura 1, a principal função do serviço de nomes global (grupo SN_G) é registrar

as IOGRs dos SN_Ls de cada domínio. Além disso, sobre o serviço de nomes em si, tanto os SN_Ls como o SN_G seguem as especificações CosNaming [OMG97] definidas pela OMG. No modelo convencional quando um objeto CORBA necessita acesso a um outro objeto ou grupo é, primeiramente, necessário que este obtenha a IOR do serviço de nomes que contém a referência do serviço de aplicação desejado (IOR ou IOGR). Depois então, de posse desta última referência, o acesso desejado é efetuado. No *GroupPac*, em acessos interdomínios, há três níveis de resolução de nomes para que IORs de objetos de aplicação sejam obtidas.

### 4.3.1. Tolerância a faltas no Serviço de Nomes do *GroupPac*

O serviço de nomes, essencial na ligação entre objetos no sistema, deve atender a requisitos de tolerância a faltas. Devido à disposição e função dos SN_Ls e SN_G no sistema, temos diferentes propriedades de TF atribuídas a cada um destes grupos de serviços.

*Tolerância a faltas nos Serviços de Nomes Locais*

O serviço nomes local (SN_L) em cada domínio é implementado utilizando os serviços do *GroupPac* e a técnica de replicação primário/*backups* [Lau99]. Tal como os outros grupos da aplicação, cada SN_L replicado é gerenciado como um grupo de seu domínio de TF (item 4). O primário do grupo SN_L mantém atualizados os *backups* (através de atualizações periódicas), utilizando o serviço de recuperação e *Logging*. O SN_L primário é responsável por receber e disponibilizar todas as referencias do domínio. No entanto, no sentido de melhorar o desempenho, requisições de operação de obtenção de referência (uma operação do tipo *stateless*) são atendidas por qualquer réplica *backup* [Lau99]. Para melhor acessibilidade, as réplicas do SN_L devem estar espalhadas em diferentes pontos do domínio de TF. Todavia, é importante ressaltar que só o SN_L primário se registra no serviço de nomes global (SN_G). O SN_L primário tem também a sua IOGR disponível em servidores HTTP como forma de redundância e para facilitar o acesso a objetos do seu domínio.

Quando a falha do SN_L primário (*crash*) é detectada (através do protocolo apresentado no item 4.2), é definido então, um novo primário entre seus *backups* – o primeiro da ordem definida na IOGR – e homologado pelo SGR do domínio. Com isso, o novo SN_L primário deve se registrar como o novo representante do domínio, tornando disponível a nova IOGR do grupo, nos servidores HTTP e no SN_G.

*Tolerância a faltas no Serviço de Nomes Global*

Outras soluções foram adotadas no *GroupPac* na replicação do serviço de nomes global (SN_G). Por conter todas as IOGRs dos serviços contidos no domínio de TF Global e também dos grupos SN_Ls, o SN_G deve ser de fácil acesso para todos os objetos de cada domínio de TF Local. Portanto, as réplicas do SN_G, em termos geográficos e para melhor acessibilidade, devem ser bem distribuídas no sistema. A característica principal do SN_G é, portanto, de ser um grupo geograficamente distribuído com poucos elementos.

Adotar a técnica de replicação primário/*backup*, com um elemento primário centralizador, não é uma solução adequada por comprometer o acesso e o desempenho em um contexto de larga escala. A solução proposta indica que as réplicas do SN_G sejam distribuídas no sistema, com pelo menos uma réplica em um *host* de cada domínio de TF Local, sendo todas ativas, exercendo as mesmas funções (registrar e disponibilizar IOGRs). A solução proposta implica no uso de propriedades da técnica de replicação ativa onde as réplicas não faltosas recebem, executam e respondem a todas requisições dos clientes. Por ter todas as réplicas ativas executando o mesmo conjunto de requisições é necessário que o grupo seja determinista, isto é conseguido assegurando as regras de acordo e ordenação [Schneider90].

No entanto, se considerarmos as operações em um serviço de nomes global, uma requisição para obter a IOGR de um SN_L de um determinado domínio é, em essência, uma operação *stateless* – não altera o estado do SN_G. Além disso, as requisições de registro de

novas IOGRs, de diferentes grupos, podem ser tratadas de forma independente – não existe uma relação de ordem entre as mesmas – o que torna desnecessário estabelecer uma ordem total no conjunto de requisições de registro de IOGRs. É considerado, também, que a carga de acesso para registros de IOGRs dos SN_L no SN_G é bem menor que os registros em um SN_L de um domínio qualquer. Diante disso, podemos utilizar protocolos mais flexíveis, considerando que o único requisito necessário para dar suporte ao SN_G é a garantia na ordem do emissor nas atualizações das IOGRs dos SN_L. Portanto, para diminuir os custos e as complicações de um protocolo de difusão atômica no domínio de TF global que envolve grandes dimensões, propomos uma solução menos custosa. Um protocolo de difusão confiável com propriedade de ordenação FIFO pode ser usado, o que atende aos propósitos de operações de registro de IOGR. Operações de solicitação de IOGR não precisam ser difundidas para todas réplicas do SN_G. Uma das réplicas, preferivelmente a mais próxima, pode atender a essa re quisição.

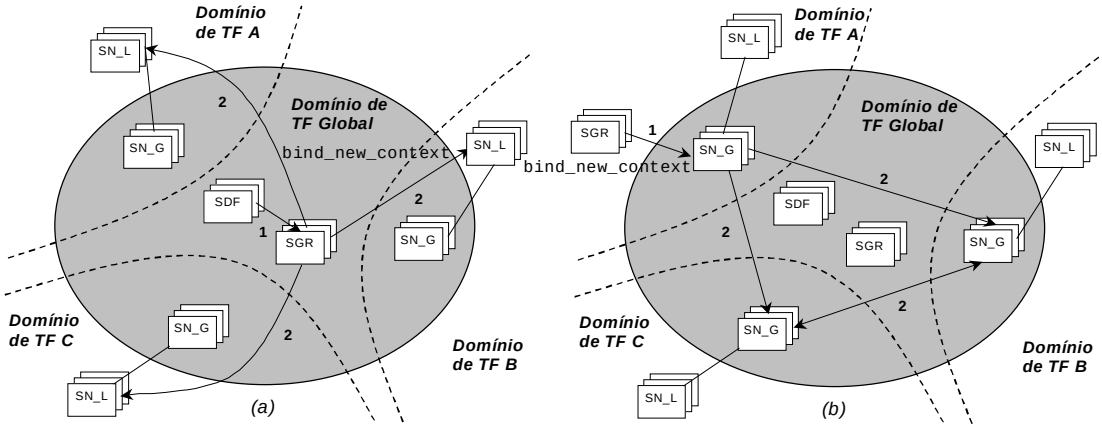


Figura 5. Serviço de nomes para suporte ao gerenciamento de grupo em sistema de larga escala.

Diante dessas considerações, apresentamos, na figura 5, o serviço de nomes global (SN_G). Diferentes de objetos, que segundo as especificações FT-CORBA não podem pertencer a mais de um domínio de tolerância a faltas, os *hosts*, ao contrário, não possuem tais restrições (item 3.1). Apesar das réplicas do SN_G fazerem parte, exclusivamente, do domínio de TF global, as mesmas podem ser instaladas em *hosts* de domínios locais, no sentido tornar o acesso dos objetos da aplicação mais eficiente.

As réplicas do SN_G formam um grupo que é, tal como SN_L nos domínios locais, gerenciado pelos serviços do *GroupPac* pertencentes ao domínio de tolerância a faltas global. Devido ao maior distanciamentos das réplicas do SN_G, intervalos de *timeout* mais longos ou protocolos de detecção de falhas [Chandra96, Ricciardi91] mais adequados às características do grupo devem ser considerados. O SDF e o SGR deste domínio realizam a tarefa de manter, através da IOGR, uma lista atualizada dos membros (*membership*) do SN_G. Em qualquer alteração na composição dos membros do grupo SN_G é gerada uma nova IOGR e enviada pelo SGR para os SN_L de cada domínio (as setas numeradas da figura 5a indicam o movimento destas informações). De forma similar, quando uma nova IOGR de grupo SN_L é gerada, o SGR do mesmo domínio local se encarrega em enviar uma cópia para uma réplica do SN_G, a mais próxima do domínio local. Esta réplica SN_G registra e difunde a IOGR (usando *multicast* FIFO) para as outras réplicas SN_G do grupo. As setas indicadas na figura 5b descrevem estas ações envolvendo o registro de uma nova IOGR de grupo SN_L.

### 4.4. Comunicação de Grupo no modelo hierárquico do *GroupPac*

A partir do modelo de hierarquia de domínios adotado pelo *GroupPac* são detectados três tipos de comunicações de grupos: comunicações com grupos intradomínios do mesmo domínio de TF do cliente; comunicações de cliente com grupos intradomínios de domínios de TF diferentes daquele ao qual pertence e, comunicações com grupos interdomínios.

O primeiro tipo, envolvendo comunicações entre clientes e grupos pertencentes ao mesmo domínio é bastante simples. Envolve a requisição do cliente no SN_L para liberar a IOGR e conseguir a ligação com o grupo desejado. A comunicação de grupo se dará usando interceptador e ferramenta proprietária disponível na forma de objeto de serviço do domínio considerado (ver introdução do item 4).

Quando um cliente e um grupo pertencem a domínios de TF diferentes, a comunicação entre ambos deve passar por uma resolução de nomes mais complicada. O objeto cliente, em um domínio *A*, ao desejar invocar um grupo de um domínio *B*, deve executar as seguintes ações:

1. Acessar o serviço de nomes local (SN_L) de seu domínio para obter a referência (IOGR) do serviço de nomes global (SN_G);
2. Acessar o SN_G para obter a IOGR do SN_L do domínio desejado, o qual contém o grupo que deseja invocar;
3. Finalmente, acessar o SN_L do domínio B para obter a referência de grupo servidor (IOGR) para a ligação desejada.

A comunicação de grupo deste caso também se dará usando interceptador e uma ferramenta proprietária disponível na forma de objeto de serviço. Só que este objeto de serviço e a correspondente ferramenta que implementa a comunicação de grupo pertencem ao domínio de TF do grupo servidor.

Os tipos de comunicações citados acima envolvem algumas trocas que depende essencialmente dos níveis de resolução necessários para estabelecer a ligação entre o objeto cliente e o grupo servidor. O terceiro tipo de comunicação citado acima envolve grupos interdomínios e é discutido nos itens subseqüentes.

### 4.4.1. Comunicação de grupos interdomínios

A comunicação com grupos escaláveis que tem suas réplicas distribuídas em diferentes domínios de tolerância a faltas (grupos interdomínios), necessitam de um suporte mais complexo do que os tipos apresentados anteriormente. Na literatura são encontradas diversas soluções algorítmicas para comunicação de grupo, inclusive com propriedades de difusão atômica, onde estes grupos se caracterizam pela dispersão de seus membros em redes de larga escala. Estes trabalhos normalmente resolvem a comunicação com estes grupos é através do particionamento em vários subgrupos onde as propriedades de comunicação seriam tratadas de maneira mais viável, localmente, em cada subgrupo [Rodrigues96, Fritzke98].

Se considerarmos as especificações FT-CORBA e as proposições no *GroupPac*, um grupo interdomínio teria, portanto, seus subgrupos distribuídos em diferentes domínios de tolerância a faltas locais. Cada domínio, que contém um subgrupo, poderia, portanto, ter um protocolo de comunicação de grupo distinto, segundo as características físicas da sua rede.

| Domínio de TF | Nome | Referência de Grupo (Subgrupos de *P*) |
|---|---|---|
| *A* | "HAL" | IOGR1 do subgrupo P1 |
| *B* | "HAL" | IOGR2 do subgrupo P2 |
| *C* | "HAL" | IOGR3 do subgrupo P3 |
| *D* | "HAL" | IOGR4 do subgrupo P4 |

Tabela 1. Registro do Grupo P particionado em quatros domínios de TF.

A figura 6 ilustra um exemplo destes grupos interdomínios que vai nos permitir entender melhor a dinâmica que envolve estes grupos. O grupo de objetos *P* desta figura é formado por um conjunto de subgrupos *P1*, *P2* e *P3*, distribuídos nos domínios locais *A*, *B* e *C*, respectivamente. Cada subgrupo é gerenciado e controlado, separadamente, pelos serviços do *GroupPac* de cada domínio (item 4.1). Além disso, quando uma nova réplica é inserida em um dos subgrupos de *P*, pelo SGR do domínio de TF local correspondente, esta nova réplica tem seu estado atualizado através do Serviço de *Logging* e Recuperação [OMG00] do *GroupPac*.

Portanto, a lista de membros (*view*) de um grupo particionado em vários domínios neste modelo pode ser dinâmica.

Para que o modelo da figura 6 possa ser plausível, estendemos a noção de nome e referencia de grupo de objetos (IOGR) do serviço de nomes para suportar grupos interdomínios. Todo grupo que for criado e que tiver em sua especificação a possibilidade de ter suas réplicas distribuídas em vários domínios deve assumir um mesmo nome para cada IOGR de subgrupo registrado no SN_L de cada domínio de TF. Isto é, as IOGRs de cada subgrupo (*P1*, *P2 e P2*) que compõe o grupo *P* deve ter o mesmo nome registrado nos SN_Ls de seu domínio local. Se um novo subgrupo de *P* for introduzido em um outro domínio (por exemplo, em um domínio de TF *D*), este subgrupo terá uma IOGR gerada pelo SGR e deverá, também, ser registrado no SN_L de domínio *D* com o mesmo nome definido para o grupo de objeto *P* (por exemplo, "HAL" na tabela 1). Logicamente, apesar de ter o mesmo nome, as três IOGRs da tabela são diferentes entre si. Uma vez que cada domínio tem o seu próprio SN_L não há problema em se registrar várias IOGRs diferentes com o mesmo nome.

No sentido de viabilizar a comunicação de grupos interdomínios no *GroupPac*, é proposto na seqüência: (i) duas novas propriedades de tolerância a faltas a serem introduzidas no FT-CORBA; (ii) uma extensão na estrutura das IOGRs; e (iii) um serviço que permite a comunicação com grupos interdomínios seguindo a hierarquia de domínios proposto.

#### 4.4.1.1. Propriedades de TF para dar Suporte a Escalabilidade

As propriedades de tolerância a faltas, definidas no FT-CORBA, são associadas para cada grupo de objeto em um domínio de tolerância a faltas. Estas propriedades são gerenciadas pelo serviço de *gerenciamento de propriedades* [OMG00] e podem ser estabelecidas como valor padrão (*default*), pela aplicação, ou dinamicamente quando o grupo está se executando. Dentre estas propriedades estão o tipo de replicação (*ReplicationStyle*), estilo de monitoramento de falhas (*FaultMonitoringStyle*), número inicial de réplicas (*InitialNumberReplicas*), intervalo de monitoramento de falhas (*FaultMonitoringInterval*) e intervalo de atualização de estado (*CheckpointInterval*). De acordo com as especificações, estas propriedades, e seus valores (parâmetros), podem ser estendidos de acordo com as necessidades da aplicação – as propriedades de TF são informações adicionais que, dependendo da aplicação, podem ser utilizadas ou não durante a execução do sistema, e não representam um ônus às especificações FT-CORBA. Portanto, para dar suporte a grupos envolvendo subgrupos gerenciados em diferentes domínios, introduzimos duas novas propriedades:

- *InterdomainGroup*: determina se um grupo contém membros em diferentes domínios (interdomínio) ou não;
- *OrderingType*: determina o tipo de ordenação a ser utilizada pelos *Gateway Seqüenciadores* (item 4.4.1.3), se *Unreliable*, *Reliable*, *Causal* ou *Total*.

Esta última propriedade depende do suporte à ordenação de mensagens provido pela ferramenta proprietária usada na comunicação de grupo no domínio.

#### 4.4.1.2. Extensão da IOGR

A especificação CORBA definiu que a referência de objeto ou de grupo de objetos, além de carregar informações sobre a localização do objeto, pode agregar algumas informações adicionais referentes ao tipo de aplicação. As informações adicionais são mapeadas através de uma estrutura de dados, conhecida como etiqueta (TAG). As etiquetas podem ser adicionadas ao final da estrutura de uma referência de objeto (IOR ou IOGR). Qualquer informação contida numa TAG pode, dependendo da aplicação, ser lida ou não durante a sua execução. No *GroupPac* estendemos a IOGR com a introdução de uma nova TAG, chamada de TAG_FTProperties, que conterá todas as propriedades de TF definidas para o grupo correspondente – inclusive as propostas no item anterior. É importante ressaltar que a inclusão

desta etiqueta é uma extensão da IOGR. Esta extensão objetiva permitir ao cliente (através do seu interceptador de mensagem), quando da obtenção de uma IOGR, conhecer, de antemão, todas as propriedades de tolerância a faltas do grupo – inclusive se este é ou não interdomínio.

#### 4.4.1.3. GSeq – *Gateway* Seqüenciador

No modelo da figura 6 é introduzido um mecanismo de ordenação de mensagem, chamado de *Gateway Seqüenciador* (GSeq), que é parte importante na comunicação de grupo interdomínios. O Gseq que é um grupo de objetos, tal como o SN_G, faz parte do domínio de TF Global – regido, portanto, pelos mesmos serviços e protocolos definidos nesse domínio (item 4.1). Se uma nova réplica do GSeq é adicionada no grupo pelo serviço de gerenciamento de replicação (SGR), o serviço de *logging* e recuperação se encarrega de atualizar esta nova réplica, tornando-a uma cópia idêntica. O GSeq tem uma única IOGR gerada pelo SGR do domínio global. Cada réplica do GSeq pode se comunicar com as outras réplicas de seu grupo. Qualquer alteração na lista de membros do grupo GSeq é gerenciado pelo SGR do domínio global, o qual realiza a função de gerar e atualizar a lista (a IOGR) das réplicas do GSeq.

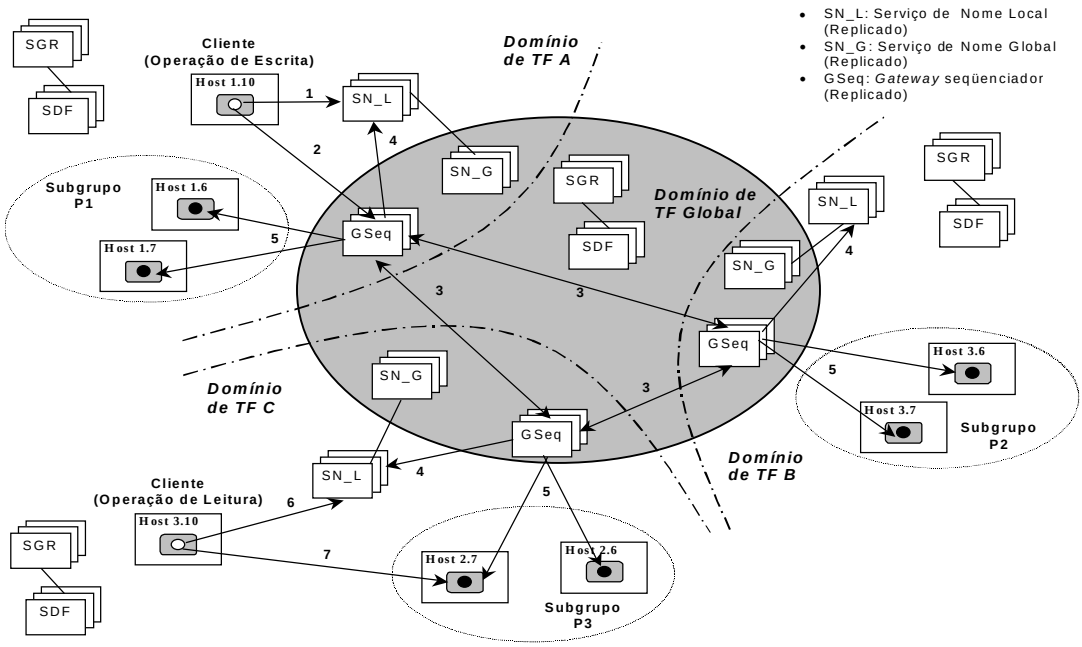


Figura 6. Comunicação de grupo em sistemas de larga escala usando o GSeq.

O grupo GSeq tem como principal função: servir de ponte para a difusão de uma mensagem entre os diferentes subgrupos de um grupo interdomínio. Qualquer domínio de TF local deve estar sempre associado a uma ou mais réplicas do Gseq. Quanto mais réplicas do GSeq estiverem associadas a um domínio de TF local, maior é o grau de tolerância a faltas do sistema. Apesar de fazerem parte apenas do domínio de TF global, as réplicas do GSeq podem ser posicionadas em *hosts* de diferentes domínios locais de TF no sentido de aumentar a disponibilidade e o desempenho no sistema. Neste modelo, o grupo GSeq fornece seus serviços para qualquer grupo interdomínio do sistema.

A figura 6 apresenta a seqüência de passos na difusão de uma requisição de cliente em um grupo interdomínio. Quando um cliente obtém a IOGR de um grupo (seta 1 da figura 6), no SN_L de um domínio de TF qualquer, o mecanismo de interceptação de mensagens [Lau00a] no cliente verifica, primeiramente, através da TAG_FTProperties da IOGR (item 4.4.1.2), se o grupo é interdomínio ou não. Caso não seja, é feita a invocação direta para o grupo (intradomínio), de acordo com os tipos de comunicação apresentados anteriormente no início do item 4.4. Em caso afirmativo, o interceptador do cliente desvia a requisição, incluindo o nome do grupo (ex: "HAL"), para uma das réplicas do GSeq (seta 2 da figura 6). Neste ponto, as

réplicas do GSeq executam um protocolo para a difusão da requisição segundo a ordem definida pelo parâmetro *OrderingType* (seta 3 da figura 6). Uma vez executado este protocolo, as réplicas do GSeq entram em contato com o SN_L (seta 4 da figura 6), enviando o nome do grupo ("HAL") e obtendo a IOGR do subgrupo de *P* do domínio ao qual pertence. Após isto, as réplicas do GSeq enviam a requisição para o subgrupo do domínio (seta 5 da figura 6). Neste passo, é importante observar que a entrega da mensagem *m*, do GSeq para o subgrupo do domínio, está vinculada ao estilo de replicação definida na etiqueta TAG_FTProperties. Se o estilo for replicação ativa, a mensagem *m* é enviada para todos os membros do subgrupo, com garantia de ordem FIFO – mensagens repetidas são descartadas pelos interceptadores de mensagem. Se for definida uma técnica de replicação passiva (fria ou quente), onde existe a figura de um elemento primário, a entrega da mensagem *m* é direcionada para esta réplica, de acordo com a especificação definida no FT-CORBA.

## 5. Aspectos de Configuração de Domínios de Tolerância a faltas

Aspectos de implementação desse sistema é apresentado em detalhes em [Lau01]. O protótipo do *GroupPac* foi testado em uma aplicação de sistema bancário executando sobre duas redes locais de 10Mbs *Ethernet* (lcmi.ufsc.br e nurcad.ufsc.br) aqui na UFSC. O sistema bancário é formado por um grupo em que suas réplicas formam dois subgrupos, uma na rede do LCMI e outra no NURCAD. Foram utilizadas no total seis máquinas *Pentium*, com *Linux* ou *Windows98* instalado, sendo três em cada rede local. Na figura 7, é mostrada uma possível composição de serviços do *GroupPac* em cada máquina das duas redes locais. Os serviços que estão dentro do circulo cinza fazem parte do domínio de TF global.

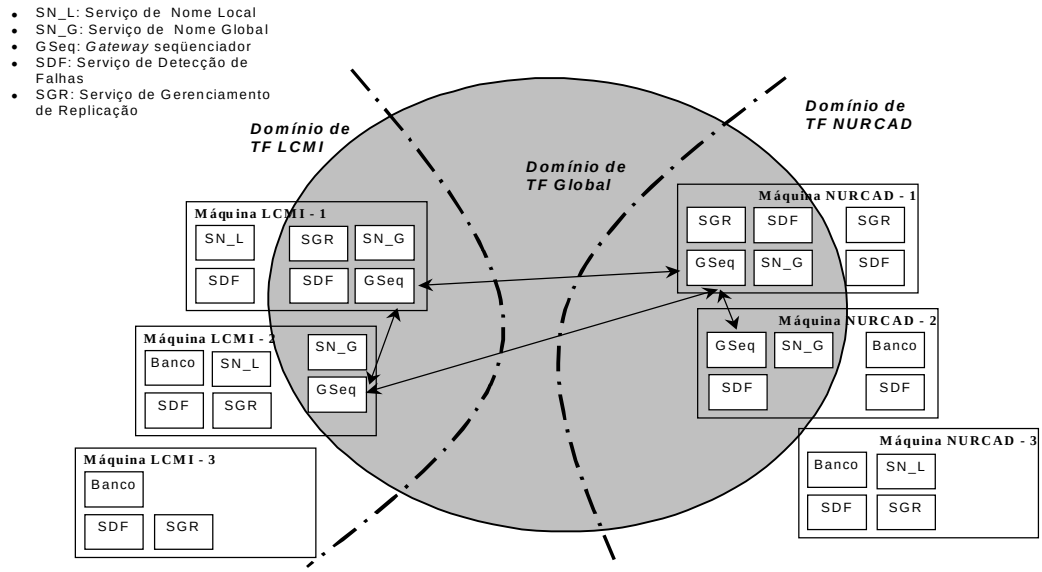


Figura 7. Exemplo de configuração para suporte a grupo interdomínio.

O grupo GSeq fornece suporte de comunicação de grupo através do ISIS. Em operações de escrita, o grupo GSeq se encarrega de difundir a atualização em cada membro do serviço bancário. No entanto, para operações de leitura o cliente pode localizar, através do SN_L, e acessar uma das réplicas do serviço bancário que esteja no seu domínio de TF, a mais próxima.

## 6. Considerações Gerais

O modelo de hierarquia de domínios de tolerância a faltas, introduzido neste trabalho, implicou, conforme apresentado, em um conjunto de adaptações e extensões nas especificações FT-CORBA. Essas mudanças tiveram maior impacto na forma como o SGR e o SDF, das especificações FT-CORBA, fornecem suporte para gerenciamento de grupos. O SN_L e o SN_G refletem os resultados deste gerenciamento nas IOGRs registradas. A hierarquia de

proposta define dois níveis de domínios de TF. Hierarquias com mais níveis poderiam ser definidas ao custo de uma complexidade maior.

As soluções do *GroupPac* foram especificadas de modo que várias soluções algorítmicas podem ser usadas nos serviços de detecção de falhas e de comunicação de grupo. Por exemplo, para a detecção de falhas no grupo de detectores, o algoritmo envolvendo detectores não confiáveis de [Chandra96] poderia ser usado. Por outro lado, o modelo de detecção de falhas dos objetos da aplicação tem que ser assimétrico, devido à semântica definida para estes detectores nas especificações FT-CORBA. O modelo hierárquico de domínios adotado e os conceitos de detectores de falhas e de domínio de TF das especificações FT-CORBA colaboram para diminuir e confinar as mensagens envolvidas na detecção de falhas em sistemas de larga escala. Em termos de trabalhos relacionados, envolvendo o uso das especificações CORBA na detecção de falhas em sistemas distribuídos, a solução apresentada em [Chung98], embora não use os conceitos definidos pelo FT_CORBA, define um conjunto de detectores, chamados de *WatchDog*, para monitorar os objetos da aplicação de forma similar aos detectores introduzidos pelo FT-CORBA. Este trabalho se limita a redes locais e não trata do problema da falha dos detectores. No OGS [Felber98] a detecção de falhas é, também, similar à especificação, sendo que suas interfaces são diferentes do FT-CORBA. No entanto, o OGS apresenta um objeto de serviço de consenso que poderia ser adaptado para a detecção de falhas. Ambas propostas não consideram aspectos de escalabilidade.

O conjunto de extensões introduzidas basicamente se deve ao suporte necessário para a comunicação de grupos interdomínios. A necessidade de adicionar duas novas propriedades de TF (item 4.4.2) é devido a IOGR [OMG00], conforme a especificação no FT-CORBA, sempre limitar um grupo em um único domínio de TF. A proposta de definir um mesmo nome para cada IOGR de um subgrupo, exemplificada na tabela 1, permite resolver o problema tanto de gerenciamento e comunicação de grupo interdomínio.

Uma vez que os grupos GSeq e SN_G se apresentam como compostos de poucas réplicas, isto nos abstrai um pouco de um modelo de sistema de larga escala. Portanto, o modelo hierárquico proposto permite que diferentes protocolos de comunicação de grupo possam ser usados pelos grupos Gseq e SN_G. O suporte de comunicação presente nos domínios envolve objetos de serviços que encapsulam as funcionalidades de uma ferramenta de comunicação de grupo. Nos nossos desenvolvimentos foi usado o Isis [Birman91]. As nossas primeiras experiências em implementar estes objetos encapsuladores e o GSeq foram apresentadas em [Lau00a]. Comparações de abordagens e medidas de desempenho foram expostas nesses trabalhos. Outras ferramentas como o Horus, Newtop, Totem, etc, poderiam ser usadas. O sistema também pode conviver com objetos de serviço de diferentes domínios encapsulando diferentes ferramentas de comunicação de grupo. Mesmo o uso de um protocolo particular, implementado na forma de um objeto de serviço CORBA é perfeitamente aceitável no nosso modelo. Em [Lau01] é apresentado, como exemplo, no item 4.4.1.3.1, um protocolo de difusão atômica, implementado como objeto de serviço CORBA, para ser usado no grupo GSeq para fornecer ordenação nas mensagens enviadas aos grupos interdomínio.

## 7. Conclusão

Fornecer suporte para o desenvolvimento de aplicações em redes de larga tem sido, atualmente, uma das principais preocupações da comunidade cientifica de sistemas distribuídos. Neste trabalho não objetivamos propor, especificamente, novos algoritmos distribuídos para sistemas de larga escala, uma vez que existem diversos na literatura, para uma larga faixa de aplicações. Nossas preocupações, sim, foram definir modelos que adequassem conceitos do FT_CORBA e soluções usuais de tolerância a faltas às necessidades de escalabilidade em sistemas de larga escala.

Nesse documento, são explicados e analisados os meios como um modelo proposto pode ser implementado no CORBA sem causar maiores impactos nas especificações originais do padrão. Este artigo mostra também um conjunto de extensões às especificações FT-CORBA que foram introduzidas para viabilizar este modelo. As soluções adotadas tiveram como princípio não modificar as interfaces do padrão. Devido à falta de espaço, disponibilizamos uma versão mais estendida deste texto em [Lau01], com uma descrição mais detalhada de cada serviço proposto e aspectos de implementação e de configuração do sistema. Este trabalho faz parte do projeto *GroupPac* que visa propor soluções para tolerância a falta às aplicações CORBA em um contexto de sistema de larga escala. As implementações atuais do *GroupPac* podem ser obtidas na WEB, no seguinte endereço (www.lcmi.ufsc.br/grouppac).

## Bibliografia

[Birman91] K. P. BIRMAN, **"The Process Group Approach to Reliable Distributed Computing"**, Tr91-1216, Cornell University Computer Science Department, Ithaca, N.Y., July 1991.

[Chandra96] T. Chandra, S. Toueg, **"Unreliable Failure Detectors for Reliable Distributed Systems"**, JACM, 43(1): 225-267, March 1996.

[Chung98] P. E. Chung, Y. Huang, S. Yajnik, D. Liang, J. Shih, **"DOORS: Providing Fault Tolerance for CORBA Applications"**, in poster session of Middleware '98, Sept. 1998.

[Felber98] P. Felber, **"The CORBA Object Group Service – A Service Approach to Object Groups in CORBA"**, PhD. Thesis, École Polytechnique Fédérale de Lausanne, Lausanne, 1998.

[Fischer 85] M. J. Fischer, N. A. Lynch and M. S. Paterson, "Impossibility of Distributed Consensus with One Faulty Process", Journal of the ACM, 32(2):374-382, Apr 1985.

[Fraga97] J. Fraga, C. Maziero, Lau L. and O. Loques, **"Implementing Replicated Services in Open Systems Using a Reflective Approach"**, Proceedings of the 3th IEEE International Symposium on Autonomous Descentralized Systems - ISADS 97, Berlin - Germany, April 1997.

[Fritzke 98] U. Fritzke, P. Ingels, A. Mostefaoui, M. Raynal, **"Fault-Tolerant Total Order Multicast to Asynchronous Group"**, In Proceedings of the 17th International Symposium on Reliable Distributed Systems - SRDS'98, IEEE Computer Society, 1998.

[Isis95] Isis Distributed Systems Inc, IONA Technologies, Ltd. **"Orbix+Isis Programmer's Guide"**, Document D070-00, 1995.

[Lau99] Lau L., J. Fraga, J. Farines, M. Ogg, A. Ricciardi**, "CosNamingFT – A Fault-Tolerant CORBA Naming Service"**, Proceeding of the 18th IEEE Symp. on Reliable Distributed Systems - SRDS'99, Lausanne, Suiça, October 1999.

[Lau00a] Lau L., J. Fraga, J. Farines, J. R. Oliveira, **"Experiências com Comunicação de Grupo nas Especificações Fault Tolerant CORBA"**, XVIII Simpósio Brasileiro de Redes de Computadores - SBRC'2000, SBC, Belo Horizonte – MG. Maio de 2000.

[Lau00b] Lau L., J. Fraga, **"Detecção de Falha para Redes de Larga Escala no Fault-Tolerant CORBA"**, II Workshop de Testes e Tolerância a faltas - WTF'2000, SBC, Curitiba – PR. Julho de 2000.

[Lau01] Lau L., J. Fraga, L. Souza, R. Padilha,**"Extensões nas Especificações FT-CORBA para Redes de Larga Escala"**, Disponível via e-mail (contate lau@lcmi.ufsc.br).

[Maffeis95] S. Maffeis, **"Run-Time Support for Object-Oriented Distributed Programming"**, Ph.D. Thesis University of Zurich. Zurich, 1995.

[Moser98] L. E. Moser, P. M. P. Melliar-Smith, P. Narasimhan, **"Consistent Object Replication in the Eternal System"**, Theory and Practice of Object Systems, 4(2): 81-92, 1998.

[OMG96] Object Management Group, **"CORBA 2.0 spec"**, Document 96-08-04. March 1996. www.omg.org.

[OMG97] Object Management Group, **"CORBAservices: Common Object Services Specification"**, OMG Document. March 1997. www.omg.org.

[OMG00] Object Management Group, **"Fault-Tolerant CORBA Specification V1.0"**, OMG document: ptc/2000-04-04, April, 2000. www.omg.org.

[OMG00a] Object Management Group, **"Unreliable Multicast Inter-ORB Protocol RFP"**, OMG document: orbos/99-11-14, Feb, 2000. www.omg.org.

[Renesse95] Robbert V. Renesse and Kenneth P. Birman, **"Protocol Composition in Horus"**, Dept. of Computer Science of the Cornell University, Mar 1995.

[Ricciardi91] A. Ricciardi and K. Birman**, "Using Process Groups to Implement Failure Detection in Asynchronous Systems"**. In 10th Symp on the Principles of Distributed Computing . August, 1991.

[Rodrigues96] L. Rodrigues, H. Fonseca and P. Verissimo, **"Totally Ordered Multicast in Large-Scale Systems"**, In Proceedings of the 16th Int. Conf. on Distributed Computing System, IEEE, 1996.

[Schneider90] F. B. Schneider, **"Implementing Fault-Tolerant Service Using the State Machine Approach: A Tutorial"**, ACM Computing Survey, 22(4):299-319, Dec 1990.