



## Mantendo Canais Ativos e em Execução

Morag Hughson e Paul Clarke

[hughson@uk.ibm.com](mailto:hughson@uk.ibm.com), [paulg\\_clarke@uk.ibm.com](mailto:paulg_clarke@uk.ibm.com)

WebSphere. software



@business on demand software

© 2003 IBM Corporation

V1.0.1



## Prefácio

**N  
O  
T  
A  
S**

Este documento é fornecido pela IBM como recomendações, com base no vasto conhecimento da IBM sobre os produtos WebSphere MQ e MQSeries, além de sua experiência com clientes utilizando esses produtos e seus ambientes e objetivos. A aceitação e implementação dessas sugestões fica a critério do leitor. Você pode ouvir opiniões diferentes de outros especialistas. Existem várias áreas dentro do WebSphere MQ que estão sujeitas a diferentes opiniões e podem variar de acordo com o ambiente específico no qual o WebSphere MQ está implementado.

Após você, o leitor, ter se familiarizado com o WebSphere MQ, poderá determinar quais sugestões deseja seguir, ou poderá receber opiniões apropriadas para seu ambiente específico.

Este documento é destinado à utilização por Administradores de Sistema e por pessoas que darão suporte às redes do WebSphere MQ. Esperamos que ele ofereça os seguintes benefícios:

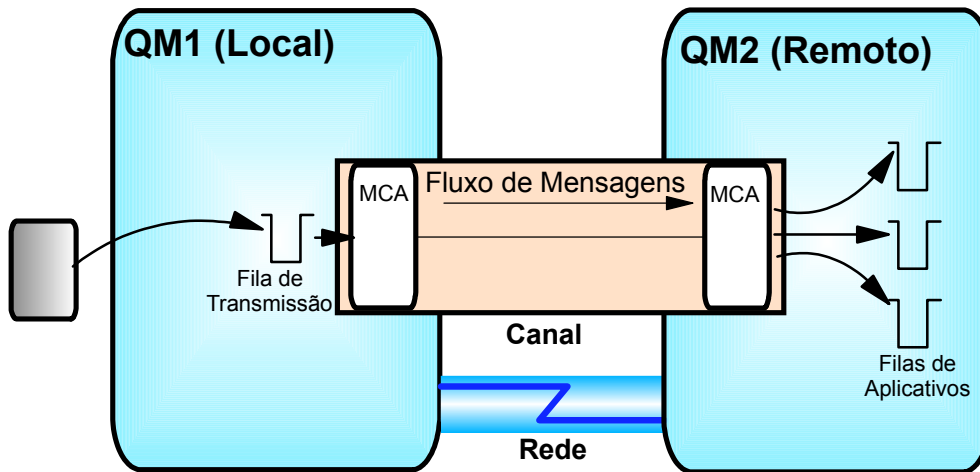
- Oferecer consistência aos processos de administração
  - Fornecer o máximo de disponibilidade aos aplicativos
  - Ajudar a evitar erros comuns cometidos por iniciantes
  - Ajudar a equipe nos estágios iniciais até se tornar especialista no WebSphere MQ
- Em geral, assegurar um início tranquilo e sucesso nos projetos do WebSphere MQ

A maioria das informações apresentadas aqui estão relacionadas às versões do MQSeries e do WebSphere MQ. Verifique a documentação para sua versão específica para assegurar-se de que os parâmetros sejam suportados.



© 2003 IBM Corporation

## Arquitetura de Canais



## Arquitetura de Canais - Notas

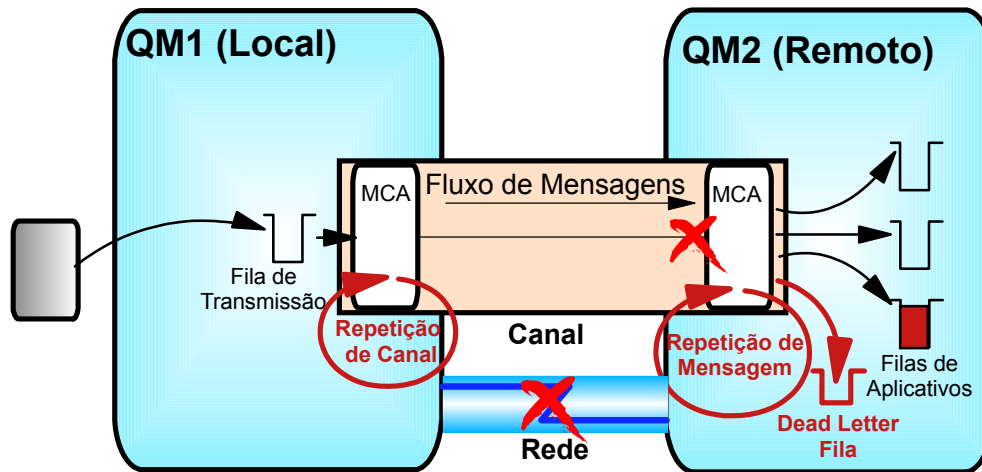
Este diagrama representa a arquitetura de canais do WebSphere MQ. Este diagrama será utilizado durante toda a apresentação para ilustrar os diferentes tópicos sobre canais.

**N** A parte principal do diagrama mostra que os aplicativos podem colocar mensagens (através de definições de fila remota ou por endereçamento explícito) que serão recebidas em uma fila de transmissão para envio a um sistema remoto. QM1 é nosso sistema local neste diagrama, e QM2 nosso sistema remoto. O MCA emissor lidando com a fila de transmissão obtém as mensagens fora da fila e as envia para o sistema remoto através do protocolo de comunicações especificado na definição do canal, onde as mensagens são recebidas pelo MCA receptor e colocadas na fila de aplicativos de destino apropriada. Este par de MCAs é conhecido como o canal.

Tudo está indo muito bem, o que pode dar errado?

**O**  
**T**  
**A**  
**S**

## O que Pode Dar Errado?



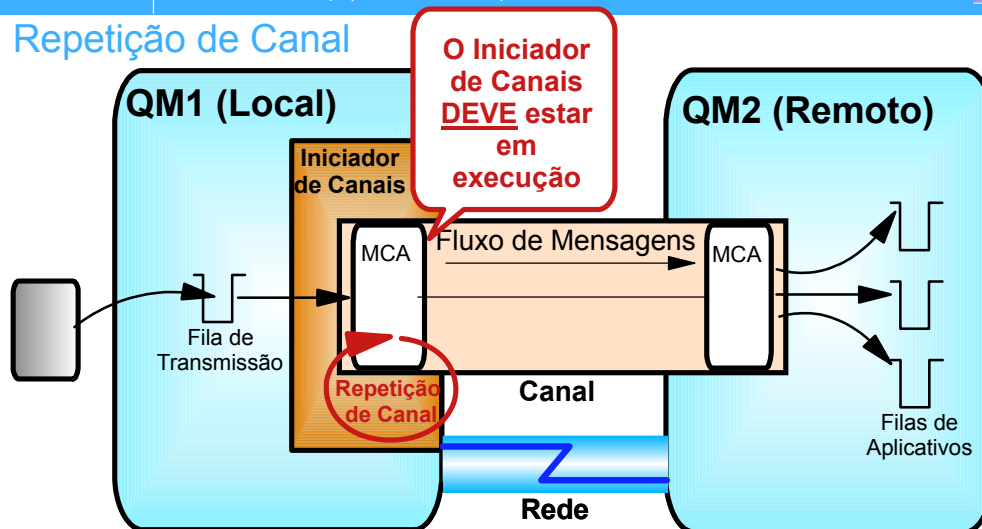
- Canal pode ser encerrado - finalização de QMgr remota
- Não foi possível enviar mensagem - Fila inválida ou cheia
- Falha na rede
- Desempenho

## O que Pode Dar Errado? - Notas

Esta seção atua como uma agenda do que iremos discutir hoje. Abordaremos brevemente diferentes itens que podem dar errado, e depois detalharemos cada um deles em seções posteriores.

- N** O canal pode encerrar. Por exemplo, devido ao encerramento do Gerenciador de Fila remoto. Para recuperar-se deste tipo de problema, você pode utilizar o iniciador de canais para tentar utilizar os canais novamente.
- O** Pode não ser possível enviar a mensagem. Isto pode ocorrer devido a um erro de aplicativo, por exemplo, a fila especificada ser inválida, ou devido a um problema de sistema, por exemplo, a fila estar cheia. Para continuar após este tipo de problema, você poderá utilizar a Dead-Letter Queue ou o recurso de Repetição de Mensagens.
- T** Você poderá sofrer uma falha na rede e, dependendo do tipo de falha, existe uma variedade de maneiras para que seu canal detecte isso e se recupere.
- A** O desempenho pode ser um problema, por isso também examinaremos algumas considerações sobre desempenho.
- S**

## Repetição de Canal



DEF CHL ....

SHORTRTY(10) SHORTTMR(60)

LONGRTY(999 999 999) LONGTMR(1200)

## Repetição de Canais - Notas

N  
O  
T  
A  
S

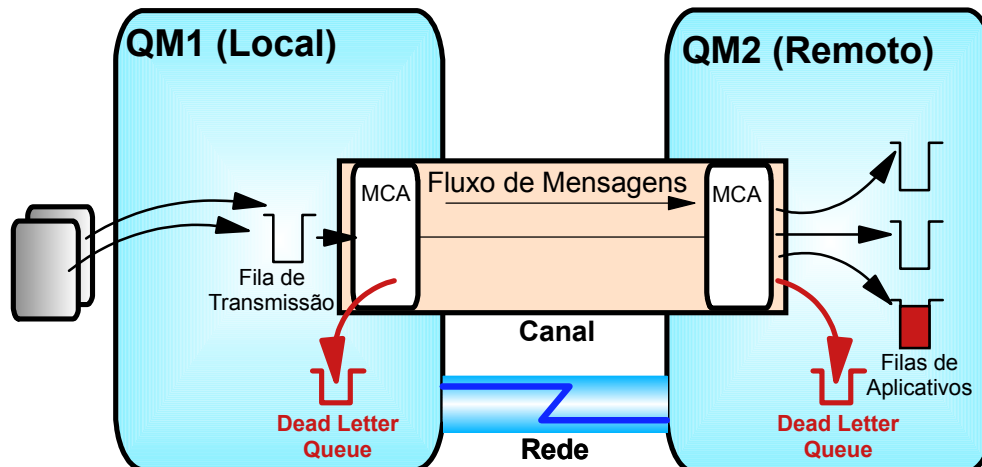
O mecanismo 'catchall' para assegurar que os canais estejam operando é para que o WebSphere MQ reinicie o canal após uma falha. Isto é conhecido como repetição de canal. A repetição de canais pode ser utilizada no envio de tipos de canais, ou seja, emissores, servidores ou emissores de cluster.

Todos os canais emissores devem ser configurados com valores de nova tentativa apropriados e o Iniciador de Canais deve estar em execução para utilizar o mecanismo de nova tentativa.

Valores de nova tentativa devem assegurar tempo suficiente para que os problemas de rede sejam resolvidos, por exemplo, fazer uma nova tentativa por 2 minutos provavelmente não será suficiente! Existem dois conjuntos diferentes de valores de nova tentativa para uso, uma nova tentativa curta e outra longa. Os valores da nova tentativa curta tendem a ser utilizados para um número pequeno de novas tentativas frequentes; em outras palavras, para tentar conectar-se novamente ao parceiro após algum blipe temporário. Os valores de nova tentativa longa tendem a ser utilizados para um número maior de novas tentativas com menor frequência; em outras palavras, se o motivo do encerramento do canal for um problema diferente de um simples blipe.

Saiba que os números especificados nos valores de repetição longa devem ser suficientes para cobrir qualquer tipo de queda pois, se a seqüência de nova tentativa for concluída sem uma transferência bem-sucedida das mensagens, o canal irá precisar de intervenção manual para reiniciá-lo.

## Dead-Letter Queue



▪ A área de armazenamento para mensagens que estão, por exemplo :-

- ▶ Endereçadas incorretamente
- ▶ Não autorizadas
- ▶ Muito grandes

## Dead-Letter Queue - Notas

É altamente recomendado que Gerenciadores de Fila tenham uma Dead-Letter Queue. Sem uma Dead-Letter Queue, um simples erro de programação em um aplicativo, ao especificar um Nome de Fila inválido, pode ser suficiente para derrubar o canal. Se vários aplicativos diferentes estiverem utilizando o mesmo canal, isto logicamente será inaceitável.

N

Observe que a DLQ pode ser utilizada no início ou fim do canal. Na extremidade emissora, o DLQ será utilizado caso uma determinada mensagem seja muito grande para ser enviada pelo canal. Isto ocorre caso o tamanho da mensagem ultrapasse o tamanho de mensagem definido para o canal, mesmo que ela possa ser acomodada pela fila de transmissão. Na extremidade receptora, o DLQ será utilizado caso a mensagem não possa ser colocada (PUT) na fila de destino. Isto pode ocorrer por vários motivos, alguns deles temporários. Os parâmetros de repetição de mensagem (definidos para um canal) e a saída da Repetição de Mensagem foram projetados para lidar com esses casos. Abordaremos a Repetição de Mensagem novamente em alguns momentos.

O

T

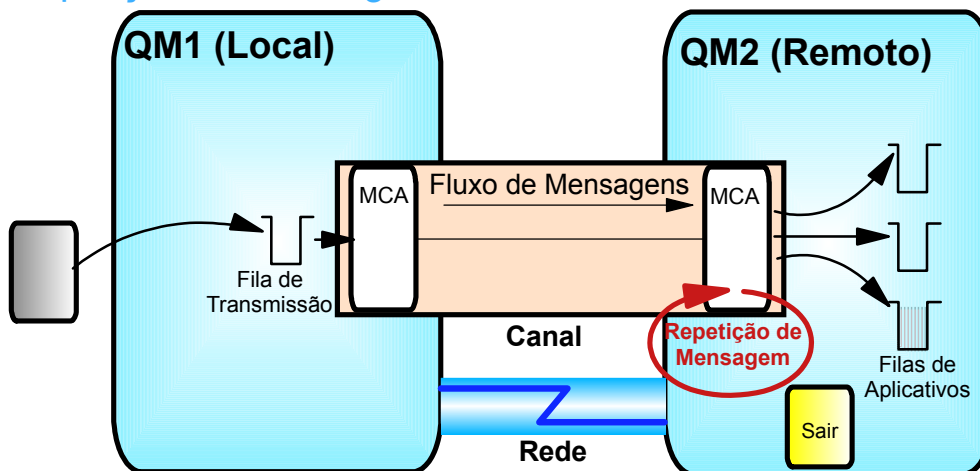
Deve ser observado que o uso da DLQ - em qualquer um dos lados de um canal - pode afetar a ordem de envio das mensagens. Caso exista algum problema com uma mensagem que faça parte de alguma seqüência, esta mensagem será colocada na DLQ e a seqüência de mensagens será interrompida.

A

S

Alguns aplicativos 'preguiçosos' requerem que as mensagens sejam sempre entregues na mesma seqüência em que foram enviadas. Esses aplicativos requerem que não exista nenhuma DLQ. A consequência de não ter uma DLQ é que uma mensagem 'inválida' em um canal irá encerrar o canal e, conseqüentemente, impedir que qualquer aplicativo receba suas mensagens. Por este motivo, é altamente recomendado que instalações possuam uma DLQ e que aplicativos sejam modificados para lidar com mensagens que sejam ocasionalmente enviadas fora da seqüência. Esta modificação não é difícil e existem várias técnicas que podem ser empregadas, como utilizar o ID de Correlação como contador, ter uma fila secundária ou utilizar agrupamento de mensagens.

## Repetição de Mensagem



**DEF CHL ....**

**MRRTY(10) MRTMR(1000)**

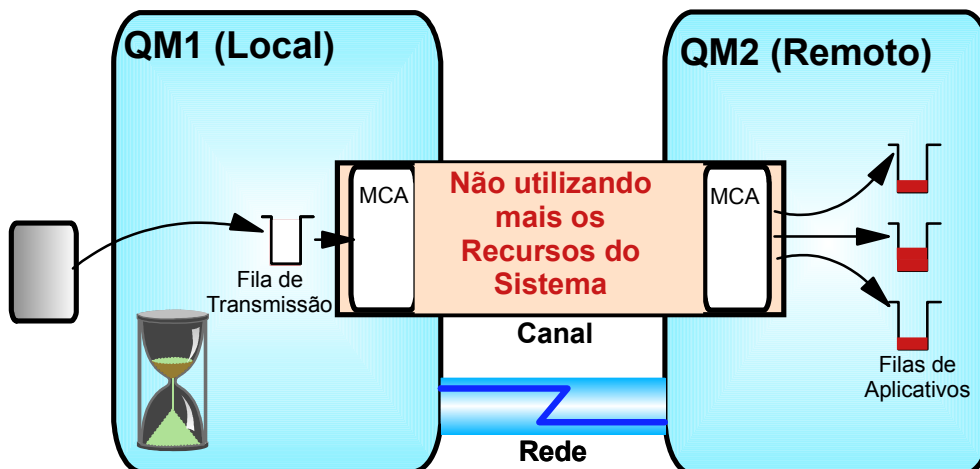
**MREXIT(msg-retry-exit-name) MRDATA()**

## Repetição de Mensagens - Notas

Considere utilizar a Repetição de Mensagens, especialmente se estiver executando sem uma Dead-Letter Queue.

- N** O mecanismo de repetição de mensagem pode ser utilizado com ou sem uma saída. Sem uma saída, o MCA de recebimento simplesmente repete o put com falha em cada intervalo. Os valores para o intervalo de tempo entre repetições e o número de repetições são retirados da definição do canal. Se utilizados com uma saída, esses números serão transmitidos pela saída, mas poderão ser alterados para qualquer valor. Utilizar uma saída com o mecanismo de repetição de mensagens permite um processamento mais inteligente da falha. Você pode, por exemplo, redirecionar mensagens, caso elas não possam ser colocadas na fila de destino.
- O** O uso de uma saída permite que a decisão de repetição seja feita por mensagem.
- T** A Repetição de Mensagens não é suportada atualmente no z/OS.
- A**
- S**

## Intervalo de Desconexão



**DEF CHL ... DISCINT(6000)**

- Combine com Acionadores

## Intervalo de Desconexão - Notas

**N  
O  
T  
A  
S**

O Intervalo de Desconexão ou DISCINT é o atributo de canal em um canal de tipo de envio utilizado para controlar a duração da execução de um canal, mesmo quando não houver mensagens a serem enviadas.

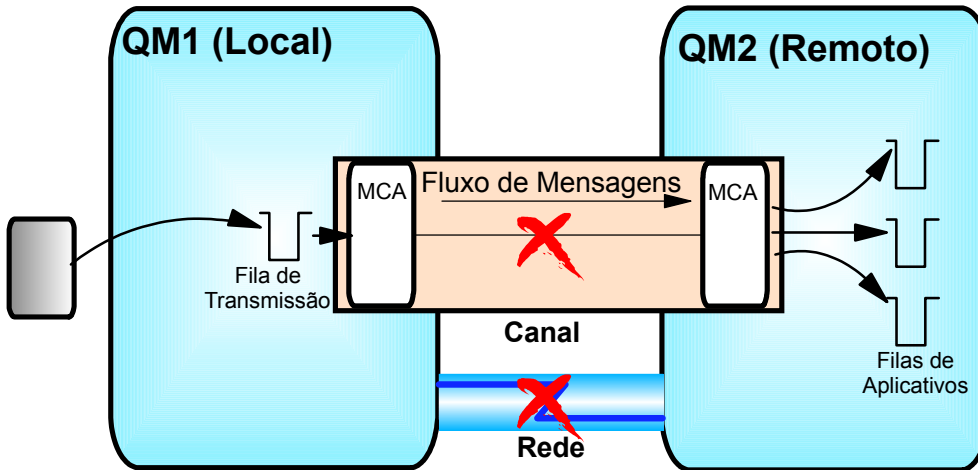
O valor definido depende de vários critérios.

- Quantos canais você possui e o quão crítico é o recurso do sistema
- Quão confiável é sua rede
- Qual é o gasto com a sessão de comunicação

O valor ideal deve ser maior que o tempo médio entre as mensagens para esse parceiro. Isto significa que o canal será desconectado quando não houver trabalho a ser enviado para o parceiro.

Combine o uso do intervalo de desconexão de seu canal com o acionador de canais para reiniciar o canal. Desta maneira, seu canal será iniciado automaticamente quando houver trabalho a ser feito, e será encerrado automaticamente quando não houver mais o que ser feito.

## Falha na Rede



- Heartbeats
- Keepalive de TCP/IP
- Adopt MCA

## Falha de Rede - Notas

Qualquer falha de rede relatada para o canal fará com que o canal seja encerrado. Caso a falha de rede não seja relatada para o canal, ele não será encerrado - ele poderá não detectar uma falha de rede.

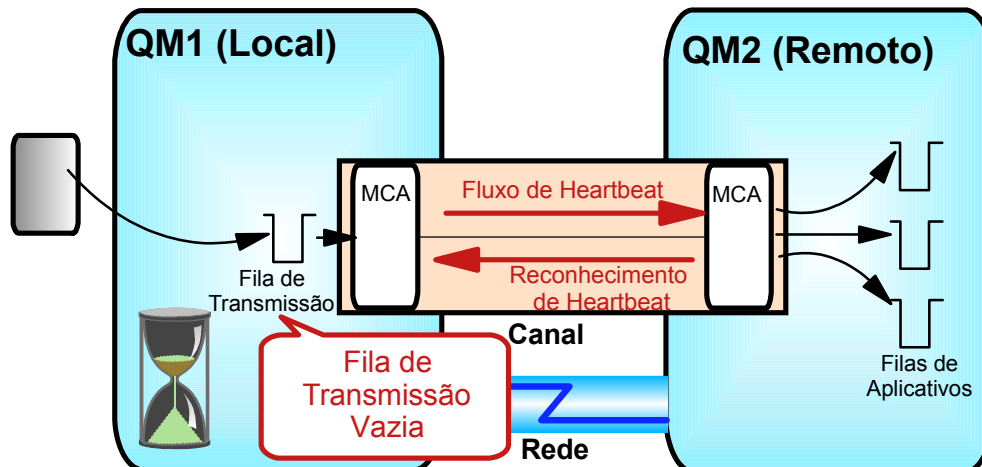
**N** O canal depende das funções de comunicação de menor nível para relatar as quedas da rede. Por exemplo, o estado de descanso de um canal RECEIVER está localizado em uma chamada recv. Caso a chamada recv não retorne, o Canal não terá como detectar que a conexão com o parceiro foi perdida. Uma das dificuldades com MCAs era o reconhecimento dentro de um tempo aceitável de falhas na rede, uma vez que os protocolos de comunicação nem sempre retornavam erros nos tempos corretos. Geralmente, LU6.2 tem obtido mais sucesso neste aspecto do que o TCP/IP (ele é um protocolo mais complexo).

**O** Vários aperfeiçoamentos feitos nos MCAs melhoraram a situação consideravelmente. Abordaremos alguns deles nas próximas seções.

**T**  
**A**  
**S**



## Heartbeats



**DEF CHL ... HBINT(300)**

**Buffers Livres** ✓

**Fechar Filas** ✓

## Heartbeats - Notas

**N**

O Intervalo de Heartbeats ou HBINT é o atributo de canal que controla a frequência com que a extremidade emissora do Canal deve verificar se tudo está correto com o receptor, quando não houver mensagens a serem enviadas. O valor é especificado em segundos em ambas as extremidade do canal e o valor apropriado é negociado para utilizar na inicialização do canal. Este valor é o intervalo com menor frequência entre os dois especificados. Se uma ou ambas as extremidades especificarem 0 (que significa a ausência de heartbeats), este valor, sendo o valor com a menor frequência possível, será escolhido.

**O**

Por padrão, os canais são definidos com intervalo de heartbeat de 300 segundos (5 minutos). Um valor relativamente alto foi escolhido como o padrão, para que houvesse um impacto baixíssimo na rede. Na prática, um valor menor que 60 segundos, por exemplo, é melhor em muitas circunstâncias.

**T**

Este heartbeat é na realidade uma mensagem breve, enviada para o MCA receptor (ou CLNTCONN durante MQGET) para verificar se ele ainda está disponível. Ele será enviado apenas quando não houver outra atividade no canal, pelo menos durante os segundos do Heartbeat. Essas mensagens resolvem dois problemas; MCAs emissores agora podem receber um retorno de erro imediato a partir da rede, em vez de esperarem seu MQGET; eles também podem permitir que MCAs receptores façam algum processamento, como procurar pelo encerramento de um gerenciador de filas antes de enviar suas respostas.

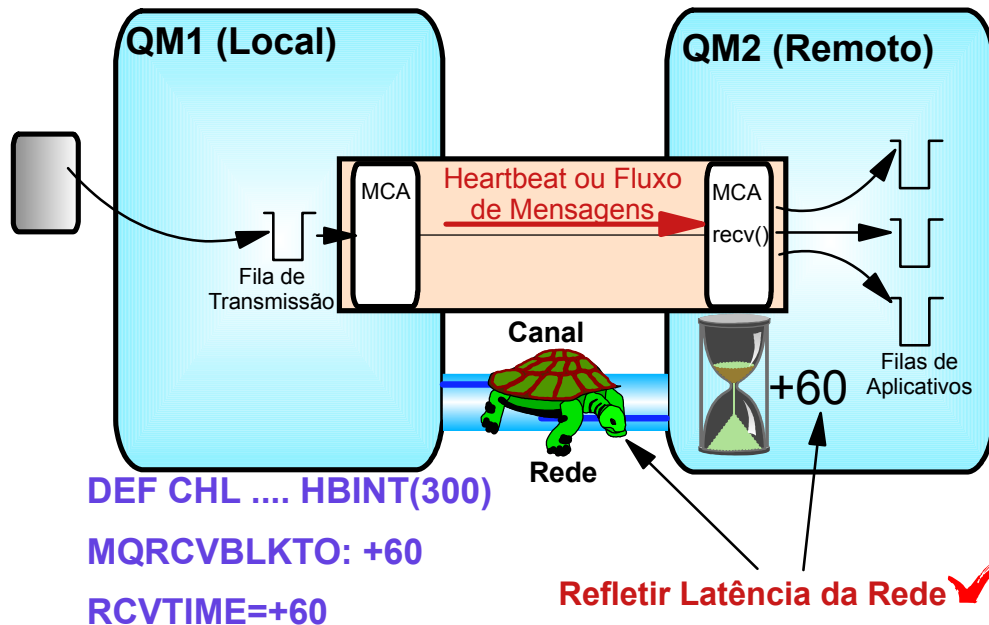
**A**

Além disso, como o canal não está transferindo mensagens regularmente, ele opta por não manter seu cache de armazenamento e Filas. Dessa maneira, no momento do heartbeat, um canal também irá limpar todos os buffers de armazenamento desnecessários e áreas de mensagem, além de fechar todas as filas armazenadas em cache.

**S**

Dessa maneira, para fins de desempenho, o valor do Heartbeat não deve ser muito pequeno. Mas observe que valores baixos de heartbeat não oferecerão tráfego adicional na rede, contanto que o canal permaneça ocupado. Leve em consideração um aplicativo enviando uma mensagem a cada dois segundos em um canal. Mesmo um valor baixo de heartbeat de 3 segundos nunca irá fazer com que o heartbeat seja enviado, uma vez que sempre haverá uma mensagem a ser enviada.

## Tempo Limite de Espera de Recebimento



## Tempo Limite de Espera de Recebimento - Notas

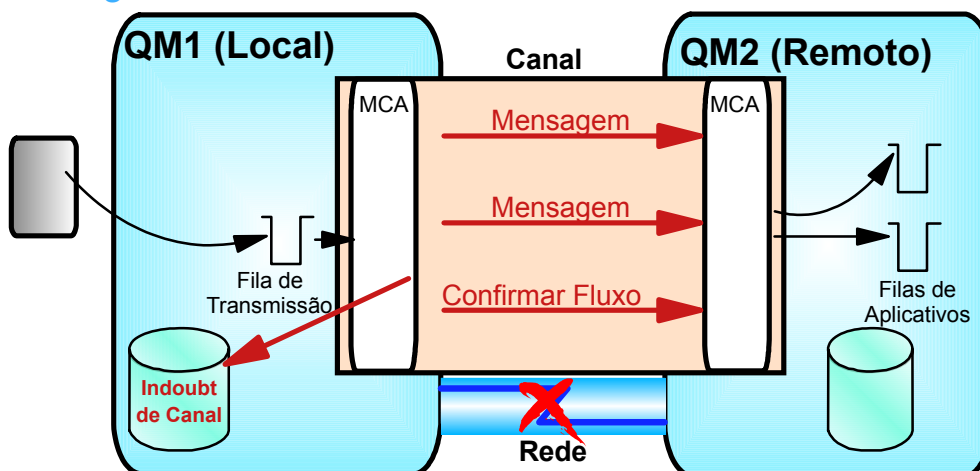
**N**  
**O**  
**T**  
**A**  
**S**

Como já mencionamos, o intervalo de heartbeats utilizado pela extremidade emissora do canal para enviar esta mensagem de heartbeat periodicamente é um valor negociado, dessa maneira, ambas as extremidades sabem qual é o valor. Sem um heartbeat, a extremidade receptora do canal não pode prever quando o próximo bloco de dados deve chegar através do canal, uma vez que o recebimento de mensagens não é determinista. No entanto, com um heartbeat, o receptor 'sabe' que dentro do intervalo do heartbeat uma mensagem ou um heartbeat será recebido. Caso ele não saiba, pode haver uma falha na sessão de comunicação. Isto é utilizado como base para o reconhecimento de falha de rede TCP/IP. Para canais CLNTCONN, este tempo limite também é utilizado enquanto aguarda por uma resposta para o comando MQI (exceto quando SVRCONN está no z/OS, que não suporta a variante cliente de heartbeats).

O receptor receberá um aviso de tempo limite caso nenhum dado seja recebido no dobro de tempo do Intervalo de Heartbeat se o Intervalo de Heartbeat for menor que 60 segundos, ou 60 segundos além do intervalo de heartbeat negociado, se ele for maior ou igual a 60 segundos, por padrão, antes de assumir que houve uma falha de comunicação. Este qualificador pode ser ajustado para refletir sua latência de rede em plataformas distribuídas, alterando a variável de ambiente MQRCVBLKTO. Existe também outra variável de ambiente, MQRCVBLKMIN, que permite definir um valor mínimo a ser utilizado ao calcular o tempo limite baseado em heartbeats. O comportamento pode ser definido no z/OS na V5.3.1 pelo uso de atributos RCVTIME de XPARM para qualificar o tempo limite e RCVTMIN para definir o valor mínimo.

Sendo assim, um pequeno valor irá detectar falhas mais rapidamente, mas também poderá ser mais dispendioso em termos de tráfego de rede e recursos do seu CPU, caso o canal esteja levemente carregado.

## Entrega Garantida



- **Dados de Sincronização gravados no disco em ambas as extremidades**
  - ▶ Permite a ressincronização após uma falha na rede
  - ▶ Mensagens em um batch em estado in-doubt não podem ser realocadas pelo algoritmo de armazenamento em cluster

## Entrega Garantida - Notas

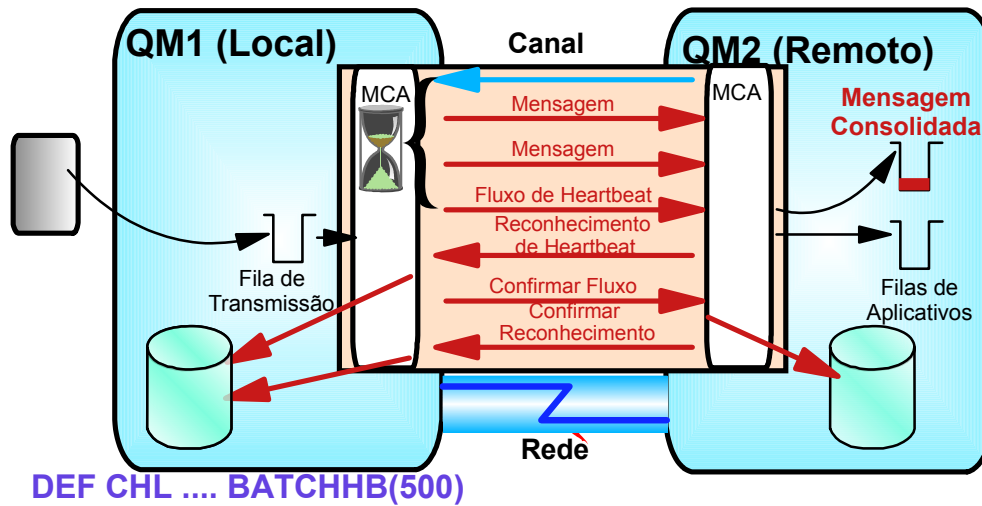
**N**  
**O**  
**T**  
**A**  
**S**

Aqui podemos ver as duas extremidades do canal transferindo um batch de mensagens. Observe como os MCAs gravam dados no disco no final do batch. Isto é feito somente para batches recuperáveis. Os dados gravados contêm o ID da transação utilizada na extremidade de envio para recuperar todas as mensagens. Assim que a extremidade emissora emitir um fluxo de "confirmação" para seu parceiro, ela estará no estado de "dúvida" até receber uma resposta. Em outras palavras, o canal emissor não sabe se a mensagem foi entregue com sucesso ou não. Caso exista uma falha de comunicação durante esta fase, o canal será encerrado em dúvida. Quando ele conectar-se novamente ao seu parceiro, irá perceber no armazenamento de dados que ele estava em dúvida com seu parceiro e irá perguntar ao outro canal se o último batch de mensagens foi entregue ou não. Utilizando a resposta enviada pelo parceiro, o canal poderá decidir se irá consolidar ou realizar o rollback das mensagens na fila de transmissão.

Estes dados de sincronização podem ser visualizados emitindo um comando DIS CHSTATUS(\*) SAVED. Os valores exibidos devem ser os mesmos em ambas as extremidades do canal.

Observe que, se o canal for reiniciando quando estiver no estado indoubt, ele resolverá o estado automaticamente. No entanto, ele poderá fazer isto apenas se estiver falando com o mesmo parceiro. Se os atributos de canal forem alterados ou se um Gerenciador de Fila diferente tomar o endereço IP ou um canal diferente servindo a mesma fila de transmissão for iniciado, o canal será encerrado imediatamente com uma mensagem dizendo que ele ainda está no estado indoubt com um Gerenciador de Fila diferente. O usuário deve iniciar o canal direcionando-o para o Gerenciador de Fila correto ou resolver o estado indoubt manualmente emitindo o comando RESOLVE CHANNEL. Observe que, neste caso, o usuário deve utilizar a saída de DIS CHS(\*) SAVED para assegurar-se de que a ação COMMIT ou BACKOUT correta seja selecionada.

## Heartbeats em Batch



- Se o tempo decorrido desde a última comunicação for maior que o Intervalo, o fluxo do heartbeat será enviado

Refletir Estabilidade de Rede ✓

## Heartbeats em Batch - Notas

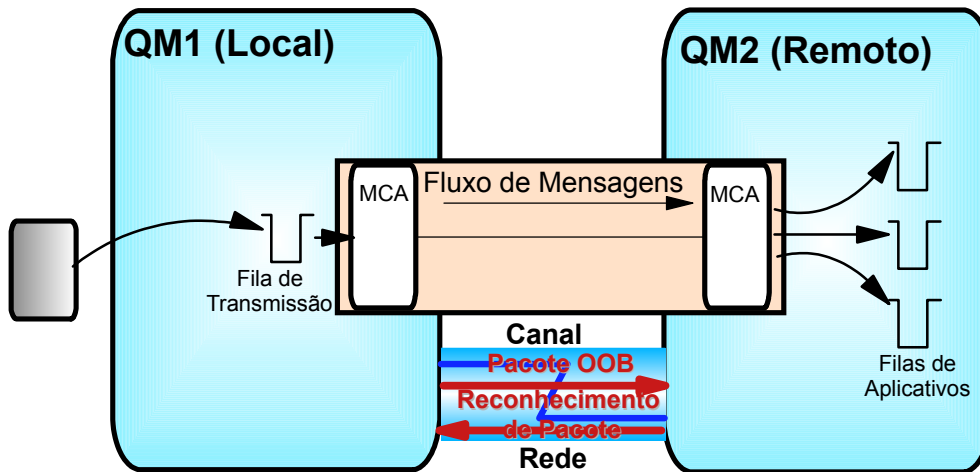
N  
O  
T  
A  
S

O problema com canais encerrados indoubt pode ser visto no algoritmo de armazenamento em cluster. Se um canal de armazenamento em cluster for encerrado de maneira anormal, todas as mensagens que deveriam ser entregues por este canal serão realocadas para outro canal e para outra instância da fila do cluster de destino. No entanto, elas não podem ser realocadas caso façam parte de um batch de mensagens sobre o qual o canal emissor esteja em dúvida se o canal receptor as recebeu.

Para reduzir a probabilidade de um canal ser encerrado no estado indoubt, um fluxo adicional foi incluído no protocolo do canal. Este fluxo é conhecido como um heartbeat em batch. Este fluxo será enviado após as mensagens para o batch terem sido enviadas, mas antes o canal marcará a si mesmo como indoubt e enviará o fluxo de confirmação. Após receber uma resposta da extremidade receptora do canal, pode-se ter praticamente certeza de que o batch pode ser concluído. Agora existe uma janela extremamente pequena em que o canal receptor pode ser encerrado e nos deixar no estado indoubt. Sem o fluxo de heartbeat em batch, esta janela será muito maior (até o tempo decorrido desde que o último fluxo foi recebido do parceiro).

Para determinar se este heartbeat em batch será enviado, observamos o tempo em que recebemos um fluxo de nosso parceiro pela última vez. Se este tempo for maior que o intervalo especificado no parâmetro BatchHeartbeat ou BATCHHB no canal, o fluxo será enviado, caso contrário, não. O intervalo especificado, portanto, é um reflexo da estabilidade da rede. Este intervalo é especificado em milissegundos.

## Keepalive de TCP/IP



▪ QM.INI **KEEPALIVE=YES**

▪ XPARM **TCPKEEP=YES**

**DEF CHL ... KAIN(360)(apenas z/OS)**

## Keepalive de TCP/IP - Notas

TCP/IP também possui um protocolo de heartbeat, chamado KeepAlive, permitindo que ele reconheça falhas de rede. O uso de KeepAlive é recomendado.

**N** Apenas no z/OS, isto pode ser configurado como parâmetro de canal, permitindo que diferentes canais tenham valores de tempo limite específicos ou que utilizem os valores do sistema. Este parâmetro de canal ainda requer que KeepAlive seja comutado no TCP/IP antes de ser utilizado.

**O** Em todas as plataformas, isto pode ser configurado por Gerenciador de Fila no WebSphere MQ, no arquivo qm.ini do equivalente. Os valores reais utilizados para os cronômetros e contadores, neste caso, são definidos para toda a pilha do TCP/IP na máquina. Valores típicos são para que a falha de rede seja detectada após um padrão de 2 horas. Esse valor provavelmente é muito alto para um ambiente do WebSphere MQ, mas outros aplicativos na mesma máquina também podem estar esperando para utilizar o KeepAlive, portanto, pode ser necessário que os valores sejam um meio-termo entre os requisitos desses diferentes aplicativos.

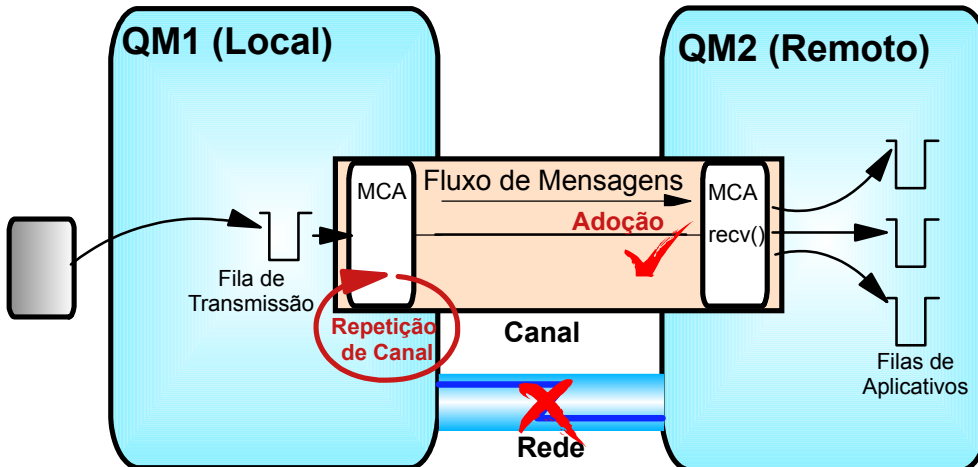
**T** Será necessário ler seus manuais sobre TCP/IP para descobrir como configurar o KeepAlive. Para o AIX, utilize o comando "no". Para HP-UX, utilize o comando "netttune". No Windows NT, será necessário editar o registro. No z/OS, atualize seu conjunto de dados TCP/IP PROFILE e inclua/altere o parâmetro INTERVAL na seção TCPCONFIG.

**A** Nota: No z/OS, recomenda-se utilizar APAR PQ75195 para TCP/IP

**S** O uso de heartbeats e tempo limite de espera de recebimento tornam o KeepAlive desnecessário quando ambas as extremidades suportam ambas as opções; no entanto, KeepAlive ainda pode ser utilizado. Ao comunicar-se com implementações do WebSphere MQ que não suportam heartbeats e leituras de não-bloqueamento, recomenda-se utilizar o KeepAlive.

Keepalive é altamente recomendado para canais SVRCONN pois, mesmo se os heartbeats clientes estiverem disponíveis, eles serão utilizados apenas durante os canais e MQGETs.

## Adopt MCA



- **QM.INI** **AdoptNewMCA, AdoptNewMCATimeout, AdoptNewMCACheck**
- **XPARM** **ADOPTMCA, ADOPTCHK**

## AdoptMCA - Notas

Se ocorrer um falha em um canal, por exemplo um erro de comunicação, não existe nenhuma garantia de que ambas as extremidades do canal irão detectar a falha ao mesmo tempo. Frequentemente, a extremidade emissora do canal detecta a falha antes, pois ela é a extremidade que está tentando enviar mensagens através do soquete.

Conseqüentemente, o emissor irá tentar reconectar-se ao Gerenciador de Fila de destino. Quando a conexão for recebida na máquina de destino, o Gerenciador de Fila verá que ele já possui um canal em execução com esse nome e a partir desse local. Conseqüentemente, a nova conexão será rejeitada.

Ao configurar AdoptNewMCA, você poderá dizer ao Gerenciador de Fila que se uma nova conexão for recebida e já existir um canal com esse nome em execução a partir do mesmo endereço de rede e a partir do mesmo Gerenciador de Fila, o canal existente deverá ser encerrado e substituído por esta nova instância. Em outras palavras, a instância do canal deve ser 'adotada' pela nova conexão.

Os vários parâmetros de configuração nas sub-rotinas dos canais de QM.INI ou equivalente (ou, no z/OS, utilize XPARMS) permitem que o comportamento de Adopt seja ligeiramente alterado. Defina a lista de tipos de canais que podem ser adotados desta maneira para ativar a adoção.

AdoptNewMCA=SVR,SDR,RCVR,CLUSRCVR,ALL,FASTPATH

ADOPTMCA=YES/NO (no z/OS)

Defina o intervalo de espera entre o encerramento do canal 'adotado' lentamente e o encerramento imediato.

AdoptNewMCATimeout=60

Claramente o nome do canal deve corresponder, mas além disso você pode especificar que o endereço de rede deve ser o mesmo, além do nome remoto do Gerenciador de Fila. Isto impede que uma conexão perdida de uma máquina diferente na rede adote canais que já estejam em execução. Defina quais atributos devem corresponder entre as conexões Adotada e nova.

AdoptNewMCACheck=QM,ADDRESS,NAME,ALL

ADOPTCHK=NONE/QMNAME/NETADDR/ALL (no z/OS)

## Desempenho

- **Algumas questões para se pensar**
  - ▶ O que é a taxa de recebimento em uma fila de transmissão?
  - ▶ O que são tempos de resposta requeridos?
  - ▶ Quanto tempo um batch leva para transmitir?
  - ▶ Qual é o tempo de inicialização de um canal?
- **As respostas dependem de seu ambiente**
  - ▶ Uso do Aplicativo
  - ▶ Sistema Operacional/Hardware
  - ▶ Largura da Banda da Rede
  - ▶ Tamanhos de Mensagens

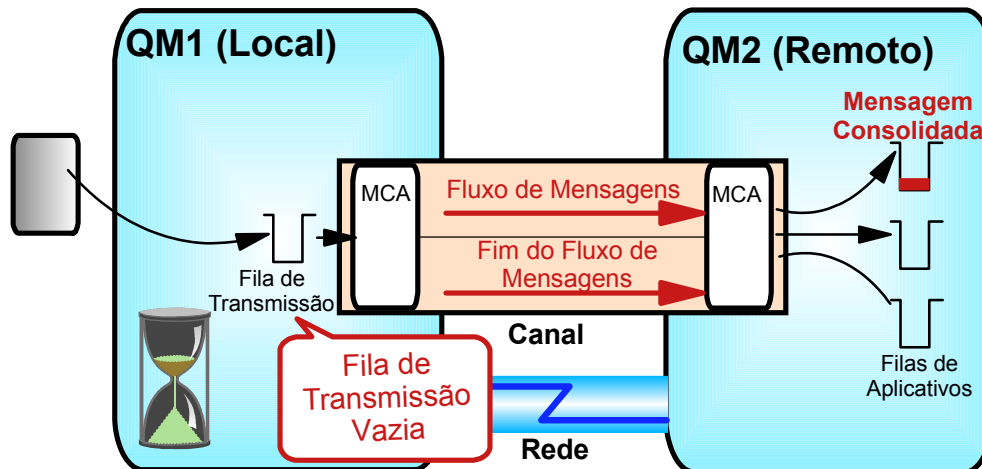


## Configuração do Desempenho

- **Utilize BatchInterval para manter os batches abertos**
  - ▶ caso a taxa de recebimento seja ligeiramente menor do que enviar o batch completo
  - ▶ pode aumentar a latência, mas melhora o rendimento total do processamento
- **Defina o valor de DisconnectInterval alto o suficiente para manter os canais ativos**
  - ▶ Reduza o número de vezes que um canal é iniciado/parado/reiniciado
  - ▶ A não ser que as cargas de rede sejam por "sessões ativas" ou que você esteja atingindo o limite de MaxChannels
- **Três cenários**
- **Utilizando Ligações Confiáveis**
- **Consulte Guias de Planejamento de Capacidades**
  - ▶ SupportPac MP00
  - ▶ SupportPac MP16



## Intervalo do Batch



- Tempo de vida mínimo do Batch de Mensagens

**DEF CHL .... BATCHINT(100)**

- ▶ Mensagens recuperáveis serão recebidas de acordo com o valor definido em BATCHINT

## Intervalo do Batch - Notas

**N**

Um batch recuperável de mensagens será encerrado assim que a fila de transmissão estiver vazia. Isto pode levar a um número maior de retornos em linha e atividade de disco do que necessário. Leve em consideração um aplicativo colocando 3 mensagens persistentes em uma fila de transmissão. Existe uma boa chance de que a primeira mensagem será recuperada pelo canal antes da segunda mensagem ser colocada. Conseqüentemente, o canal irá completar o batch e a segunda e terceira mensagens serão colocadas em um segundo batch.

**O**

Ao incluir um pequeno atraso no batch, como por exemplo BATCHINT(100), todas as 3 mensagens serão enviadas no primeiro batch, com apenas metade da atividade de disco. Observe que, apesar disso, as mensagens aparentemente demoram mais para serem enviadas, pois existe um atraso de no mínimo 100 ms. BATCHINT não tem nenhum efeito em mensagens não persistentes NPMSPEED(FAST), pois essas mensagens são colocadas fora de syncpoint. Geralmente, os aplicativos enviam consultas de bancos de dados como não-persistentes, que são atualizações críticas, e atualizações de bancos de dados, como mensagens persistentes, que não são tão críticas.

**T**

O efeito e valor para a rede de BATCHINT depende muito da rede e do perfil do sistema de mensagens do MQ. Quanto maior for o número de canais na máquina, maior será a economia potencial.

**A**

Uma maneira de ajudar a identificar se há muita vantagem em se utilizar BATCHINT é verificar o tamanho do batch efetivo. Emita DISPLAY CHSTATUS e divida o número de mensagens pelo número de batches. Caso este número seja muito baixo, (por exemplo, entre 1 e 2), o canal terá que consolidar a cada 1 ou 2 mensagens, o que não é eficiente e que pode ser melhorado por BATCHINT. Observe que isto é apenas uma indicação, pois, se todas as mensagens forem não-persistentes, BATCHINT não irá ajudar em nada.

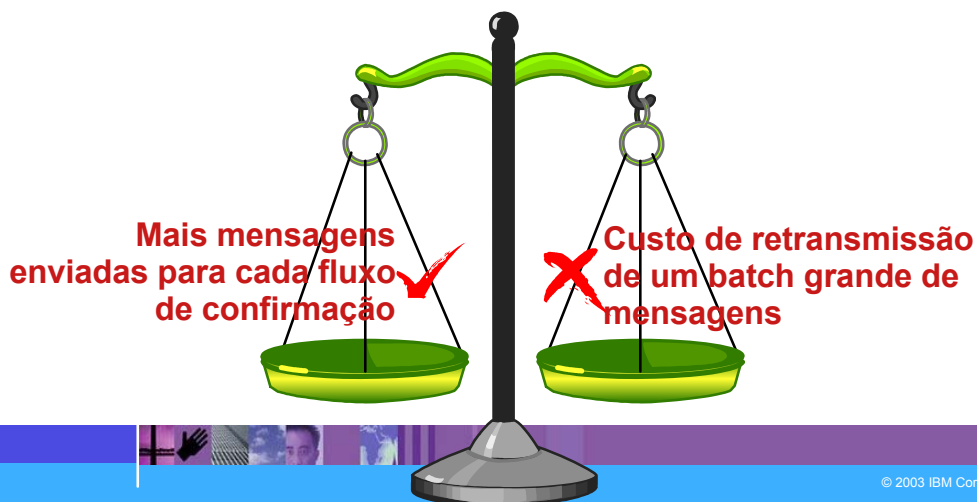
**S**

Outro fator a ser verificado é o uso de disco dos processos do canal. Se você constatar uma alta utilização de disco, existe uma boa chance de que BATCHINT reduza este custo e melhore o rendimento geral do processamento.



## Cenário 1

- **Transferência em massa de filas de transmissão pré-carregadas**
  - ▶ **Utilize um valor grande de BatchSize (o padrão de 50 é apropriado),**
    - a não ser que você esteja utilizando mensagens muito grandes (custo de retransmissão)
  - ▶ **BatchInt deve ser 0**



## Cenário 1 - Notas

Nosso primeiro cenário de ajuste de desempenho é uma transferência em massa das mensagens em uma fila de transmissão pré-carregada.

**N** O objetivo aqui é conseguir enviar as mensagens em batches grandes, dessa maneira, reduzindo o custo do processamento final em batch por mensagem. Assim, um Tamanho de Batch grande deve ser utilizado (o padrão de 50 é apropriado) e o Intervalo do Batch deve ser definido como 0. Não há necessidade de manter o batch aberto neste cenário, uma vez que todas as mensagens serão pré-carregadas na fila de transmissão.

**O** O equilíbrio do desempenho é de quantas mensagens você deve enviar em um único batch para obter um custo de processamento final em batch baixo por mensagem, comparado com o custo de retransmissão do batch inteiro caso a conectividade da rede seja perdida durante a conexão. Se estiver movendo mensagens muito grandes, esse equilíbrio deve ser atingido com um Tamanho de Batch menor que 50.

**T**

**A**

**S**

## Cenário 2

### ■ Transferência aos poucos para processamento adiado

- ▶ Os números exatos dependem da taxa de recebimento
- ▶ Tente atingir batches altos
- ▶ Defina BatchInt para um atraso aceitável antes da transmissão (por exemplo, 1 minuto)
- ▶ Mas não o defina muito alto: caso contrário, você terá uma Unidade de Trabalho de longa execução



## Cenário 2 - Notas

Nosso segundo cenário de ajuste de desempenho é uma transferência aos poucos de uma mensagem para processamento adiado.

**N** O objetivo aqui é conseguir atingir um tamanho de batch razoável para as mensagens, dessa maneira, reduzindo o custo do processamento final de batch por mensagem. Este é um uso clássico do Intervalo de Batch. As mensagens são recebidas em uma taxa ligeiramente mais devagar do que o canal pode processá-las. Manter o batch aberto por um pouco mais de tempo permite que mais mensagens sejam colocadas em batch.

**O** O equilíbrio de desempenho é de quantas mensagens você consegue enviar em um batch, comparado com o fato de que as mensagens recuperáveis aparentemente demorarão mais para serem enviadas, uma vez que não são consolidadas nas filas de destino até a conclusão do batch, ou seja, pelo menos após o Intervalo do Batch.

**T** Se o tempo de envio não for problema, tome cuidado para não definir o Intervalo de Batch para um valor muito alto, pois você estará correndo o risco de gerar Unidades de Trabalho de longa execução.

**A**

**S**

## Cenário 3

### ▪ Pedido/Resposta Síncronos

- ▶ Recomenda-se não utilizar BatchInt caso o tempo de resposta seja essencial
- ▶ Se o volume persistente for baixo, considere definir um valor menor para BatchSize
- ▶ Para fluxos de mensagem não-persistentes, utilize NPMSPEED(FAST) e um valor que não seja zero para BatchInt



## Cenário 3 - Notas

Nosso terceiro cenário de ajuste de desempenho é um modelo de pedido/resposta síncronos, onde o tempo de resposta importa.

Esta é uma situação semelhante à do cenário 2, exceto que desta vez o equilíbrio é ponderado pelo tempo mais rápido de resposta em vez de por um número maior de mensagens em um batch.

Se o volume de mensagens persistentes for baixo, considere utilizar um tamanho menor de batch e não utilizar um Intervalo de Batch, ou utilizar um valor relativamente mais baixo, de alguns segundos, para assegurar-se de que as mensagens estejam disponíveis durante o tempo certo.

Caso tenha mensagens não-persistentes, você deve utilizar uma Velocidade de Mensagem Não-persistente ou NPMSPEED de FAST e definir um Intervalo de Batch diferente de zero, pois mensagens não-persistentes enviadas através de um canal rápido estão disponíveis imediatamente. O valor do Tamanho do Batch ainda será utilizado, mas ele não afetará o tempo de resposta.

Mais alguns valores de exemplo:

Com uma taxa esperada de 30 mensagens não-persistentes por segundo, definidos BatchInt=2000 (2 segs) e BatchSize=50 para um tamanho efetivo de batch de 50.

N  
O  
T  
A  
S

## Ligação Confiável

- **Processo Único**
  - ▶ Para MCA e Agente de Gerenciador de Fila Local
  - ▶ Sem IPC superior para comandos MQI do MCA
    - significativo para mensagens não-persistentes
- **Configurado no arquivo ini do Gerenciador de Fila**
  - ▶ Sub-rotinas de Canais

### MQIBINDTYPE=FASTPATH

- Não utilize caso ocorra uma saída do Canal de teste
- Tenha atenção ao emitir

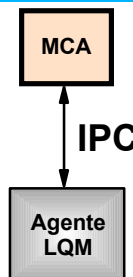
### STOP CHL() MODE(FORCE)

(pré-V5.3)

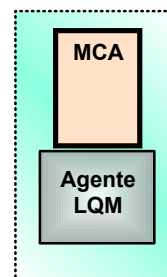
### STOP CHL() MODE(TERMINATE)

V5.3

#### Canal Comum



#### Canal Utilizando Ligação Confiável



© 2003 IBM Corporation

## Ligação Confiável - Notas

**N**  
**O**  
**T**  
**A**  
**S**

Por padrão, no conjunto distribuído de produtos, todas as operações do Gerenciador de Fila são na verdade executadas por um processo separado do aplicativo conhecido como agente. Cada chamada da API é enviada para o processo do agente através de algum mecanismo IPC. Isto é feito por motivos de isolamento. Um aplicativo não tem acesso direto aos recursos ou memória do Gerenciador de Fila. No entanto, como cada chamada de MQI é um comutador de processos, isso causa um impacto no desempenho. Para uma mensagem não-persistente, aproximadamente 60% do custo da chamada é somente a comutação do processo.

Todos os aplicativos podem optar por serem executados no modo 'confiável', que significa que as funções do Gerenciador de Fila são executadas diretamente alinhadas com o aplicativo. Recomenda-se que os aplicativos, em geral, não sejam executados neste modo. Canais, no entanto, são código do produto e, portanto, podem ser executados como aplicativos de atalho 'confiáveis'. Por padrão, eles serão executados como aplicativos normais, mas ao definir MQIBindType=FASTPATH na sub-rotina dos Canais no arquivo QM.INI, todos os canais serão executados de maneira confiável.

Deve-se tomar cuidado com saídas de canais. Como saídas de canais são executadas alinhadas com o MCA, elas agora também ocuparão o mesmo espaço de memória que o agente do Gerenciador de Fila. Saídas de canais devem ser testadas muitas vezes antes de serem executadas neste modo.

Também deve-se tomar cuidado ao utilizar o comando STOP MODE(FORCE) em releases anteriores à V5.3 e, na V5.3, deve-se tomar cuidado ao utilizar o comando STOP MODE(TERMINATE), pois eles podem danificar ou interromper o Gerenciador de Fila, já que a limpeza não pode ser executada completamente quando o canal é eliminado.

© 2003 IBM Corporation

## Referência de Parâmetros

Canal	Descrição	
HBINT	Intervalo de Heartbeat	
BATCHSZ	Número Máximo de Mensagens em um Batch	
BATCHINT	Tempo de Vida Mínimo do Batch de Mensagens	
DISCINT	Intervalo após o qual o canal será encerrado	
SHORTRTY, SHORTTMR LONGRTY, LONGTMR	Contagens e cronômetros de repetição de canal	
MRRTY, MRTMR MREXIT, MRDATA	Contagens e cronômetros de repetição de mensagens e saída (não z/OS)	
QM.INI - Sub-rotina de Canal	XPARMS	Descrição
AdoptNewMCA	ADOPTMCA	Adotar Tipos de Canal
AdoptNewMCATimeout		Adotar Tempo Limite de Quiesce
AdoptNewMCACheck	ADOPTCHK	Adotar Verificação de Parâmetro
KeepAlive	TCPKEEP	Utilizar Keepalive
MQBindType		Ligação de FASTPATH do Canal
Variáveis de Ambiente	XPARMS	Descrição
MQRCVBLKTO	RCVTIME	Tempo Limite de Espera de Recebimento
MQRCVBLKMIN	RCVTMIN	Tempo Limite Mínimo de Espera de Recebimento

## Referência de Parâmetros - Notas

**N  
O  
T  
A  
S**

Este quadro lista os atributos de um canal que afetam os recursos de desempenho e recuperação.

Alguns dos atributos fazem parte da definição verdadeira de canal e outros são definidos em outros lugares, como em um arquivo INI.

Detalhes completos dos parâmetros na definição do canal podem ser encontrados no manual InterCommunication.

Em plataformas distribuídas, os parâmetros presentes no arquivo QM.INI estão presentes nas sub-rotinas de Canais; no Windows, eles estão implementados no Registro

Key : HKEY\_LOCAL\_MACHINE/SOFTWARE/IBM/MQSeries/QueueManager/<QMNAME>  
ou utilize os serviços do MQSeries

Os parâmetros do Iniciador de Canais fornecem alguns dos mesmos parâmetros no z/OS que podem ser encontrados no arquivo QM.INI nas plataformas distribuídas. Os XPARMS são criados utilizando a macro CSQ6CHIP.