

3.4 consultas e comandos do Servidor

O grupo de consulta de servidor de comandos foi projetado para retornar informações sobre qualquer servidor que está ligado à rede.

Nessas consultas, quando um parâmetro aparece como <meta>, wildcard máscaras são geralmente válido. Para cada parâmetro, no entanto, apenas uma consulta e conjunto de respostas deve ser gerado. Na maioria dos casos, se um apelido é dada, isso significará que o servidor para o qual o usuário está conectado.

Essas mensagens normalmente têm pouco valor para os serviços, é Portanto, recomenda proibir serviços de usá-los.

3.4.1 mensagem MOTD

Comando: MOTD

Parâmetros: [<meta>]

O comando MOTD é usado para obter a "Mensagem do Dia" do dado servidor ou servidor atual <destino> é omitido.

Curingas são permitidos no parâmetro <target>.

As respostas numéricas:

RPL_MOTDSTART RPL_MOTD

RPL_ENDOFMOTD ERR_NOMOTD

3.4.2 mensagem lusers

Comando: lusers

Parâmetros: [<máscara> [<meta>]]

O comando lusers é usado para obter estatísticas sobre o tamanho do rede de IRC. Se nenhum parâmetro é dado, a resposta será sobre a net todo. Se a <máscara> for especificado, então a resposta só será

Kalt Informativa [Página 25]

RFC 2812 Internet Relay Chat: Cliente Protocolo abril 2000

referem-se à parte da rede formada pelos servidores correspondentes a máscara. Finalmente, se o parâmetro <target> for especificado, o pedido

é encaminhado para o servidor que irá gerar a resposta.

Curingas são permitidos no parâmetro <target>.

As respostas numéricas:

```
RPL_LUSERCLIENT RPL_LUSEROP  
RPL_LUSERUNKNOWN RPL_LUSERCHANNELS  
RPL_LUSERME ERR_NOSUCHSERVER
```

mensagem 3.4.3 Versão

Comando: VERSÃO

Parâmetros: [<meta>]

O comando versão é usado para consultar a versão do servidor programa. Um parâmetro opcional <destino> é usado para consultar a versão do programa de servidor que um cliente não está diretamente conectado.

Curingas são permitidos no parâmetro <target>.

As respostas numéricas:

```
ERR_NOSUCHSERVER RPL_VERSION
```

Exemplos:

VERSÃO tolsun.oulu.fi; Comando para verificar a versão do server "tofsun.oulu.fi".

mensagem 3.4.4 Stats

Comando: STATS

Parâmetros: [<query> [<meta>]]

O comando de estatísticas for usada para consultar as estatísticas de determinado servidor. E se <Query> parâmetro for omitido, apenas o fim de estatísticas resposta é enviada costas.

A consulta pode ser dada para qualquer única letra que só é verificada por o servidor de destino e é de outro modo transmitido por intermédio servidores, ignorados e inalterados.

Curingas são permitidos no parâmetro <target>.

RFC 2812 Internet Relay Chat: Cliente Protocolo abril 2000

Exceto para aqueles abaixo, a lista de consultas válidas é dependente de implementação. As consultas padrão abaixo devem ser suportado pelo servidor:

- l - retorna uma lista de conexões do servidor, mostrando como tempo cada conexão foi estabelecida ea tráfego sobre essa conexão em Kbytes e mensagens para cada sentido;
- m - retorna a contagem de uso para cada um dos comandos suportados por parte do servidor; Comandos para as quais a contagem de uso é zero pode ser omitido;
- o - retorna uma lista de usuários privilegiados configurados, operadores;
- u - retorna uma seqüência que mostra quanto tempo o servidor tem sido acima.

Recomenda-se também que a configuração do cliente e acesso ao servidor ser publicado este caminho.

As respostas numéricas:

```
ERR_NOSUCHSERVER
RPL_STATSLINKINFO RPL_STATSUPTIME
RPL_STATSCOMMANDS RPL_STATSOLINE
RPL_ENDOFSTATS
```

Exemplos:

ESTATÍSTICAS m; Comando para verificar o uso de comando para o servidor que você está conectado a

mensagem 3.4.5 Ligações

Comando: LINKS

Parâmetros: [[<servidor remoto>] <máscara servidor>]

Com links, o usuário pode listar todos os servernames, que são conhecidos pela

servidor responder a consulta. A lista retornada de servidores deve corresponder a máscara, ou se nenhuma máscara é dado, a lista completa é retornado.

Se <servidor remoto> é dada, além de <máscara do servidor>, os LINKS comando é encaminhado para o primeiro servidor descobriu que corresponde ao nome (Se algum), e é então necessário que o servidor para responder a consulta.

As respostas numéricas:

```
ERR_NOSUCHSERVER
RPL_LINKS RPL_ENDOFLINKS
```

Kalt Informativa [Página 27]

RFC 2812 Internet Relay Chat: Cliente Protocolo abril 2000

Exemplos:

LINKS * .au; Comando para listar todos os servidores que tem um nome que corresponda * .au;

LINKS * .edu * .bu.edu; Comando para servidores de listas de correspondência * .bu.edu Como visto pelo primeiro servidor correspondência * .edu.

3.4.6 mensagem Tempo

Command: TIME

Parâmetros: [<meta>]

O comando de tempo é usado para consultar a hora local do especificado servidor. Se o parâmetro <target> não é dado, o servidor receber o comando deve responder à consulta.

Curingas são permitidos no parâmetro <target>.

As respostas numéricas:

```
ERR_NOSUCHSERVER RPL_TIME
```

Exemplos:

TEMPO tolsun.oulu.fi; verificar o tempo no servidor
"Tolson.oulu.fi"

3.4.7 mensagem Conectar

Comandos: connect

Parâmetros: <servidor de destino> <porta> [<servidor remoto>]

O comando CONNECT pode ser usado para solicitar um servidor para tentar estabelecer uma nova ligação para outro servidor imediatamente. CONNECT é um comando privilegiada e deve estar disponível apenas para Operadores de IRC. Se a <servidor remoto> é dado e sua máscara não coincidir com o nome do análise de servidor, a tentativa CONNECT é enviada para o primeiro jogo da servidor remoto. Caso contrário, a tentativa de conexão é feito pelo servidor processar o pedido.

O servidor que recebe um comando remoto CONNECT deve gerar uma mensagem WALLOPS descrevendo a origem eo destino do pedido.

As respostas numéricas:

ERR_NOSUCHSERVER ERR_NOPRIVILEGES
ERR_NEEDMOREPARAMS

Kalt Informativa [Página 28]

RFC 2812 Internet Relay Chat: Cliente Protocolo abril 2000

Exemplos:

CONNECT tolsun.oulu.fi 6667; Comando para tentar conectar locais
servidor para tolsun.oulu.fi na porta 6667

mensagem 3.4.8 Traço

Comando: TRACE

Parâmetros: [<meta>]

comando TRACE é usado para encontrar o caminho para o servidor específico e informações sobre seus pares. Cada servidor que processa este comando DEVE informar ao remetente sobre ele. As respostas dos pass-through ligações formam uma cadeia, o que demonstra rota para o destino. depois de enviar esta resposta de volta, a consulta deve ser enviada para o próximo servidor até dada <meta> servidor é atingido.

comando TRACE é usado para encontrar o caminho para o servidor específico. Cada servidor que processa esta mensagem deve dizer o remetente sobre ele, enviar uma resposta indicando que é uma ligação pass-through, formando uma cadeia de respostas. Depois de enviar esta resposta de volta, ele deve, em seguida, enviar o mensagem TRACE para o próximo servidor até que determinado servidor é atingido. E se o <meta> parâmetro for omitido, recomenda-se que TRACE comando envia uma mensagem para o revelador remetente qual servidor os locais servidor tem conexão direta com.

Se o destino dado por <destino> é um servidor real, o servidor de destino deve informar todos os servidores, serviços e operadores que estão ligados a ele; se o comando foi emitido por uma operador, o servidor também pode relatar todos os usuários que estão ligados a isto. Se o destino dado por <destino> é um apelido, em seguida, apenas uma responder por esse apelido é dado. Se o <meta> parâmetro é omitido, recomenda-se que o comando TRACE é analisado como alvo para o servidor de processamento.

Curingas são permitidos no parâmetro <target>.

As respostas numéricas:

ERR_NOSUCHSERVER

Se a mensagem TRACE é destinado a outro servidor, todos servidores intermediários deve retornar uma resposta RPL_TRACELINK para indicar que o traço passou por ele e onde ele está indo em seguida.

RPL_TRACELINK

Kalt Informativa [Página 29]

RFC 2812 Internet Relay Chat: Cliente Protocolo abril 2000

Uma resposta TRACE pode ser constituído por um número do seguinte respostas numéricas.

RPL_TRACECONNECTING RPL_TRACEHANDSHAKE
RPL_TRACEUNKNOWN RPL_TRACEOPERATOR
RPL_TRACEUSER RPL_TRACESERVER
RPL_TRACENEWTYPE RPL_TRACESERVICE

RPL_TRACECLASS RPL_TRACELOG
RPL_TRACEEND

Exemplos:

TRACE * .oulu.fi; TRACE a uma correspondência de servidor
* .oulu.fi

3.4.9 comando admin

Comando: ADMIN

Parâmetros: [<meta>]

O comando admin é usado para encontrar informações sobre o administrador do servidor dado, ou servidor atual <meta> parâmetro é omitido. Cada servidor deve ter a capacidade de transmitir mensagens ADMIN para outros servidores.

Curingas são permitidos no parâmetro <target>.

As respostas numéricas:

ERR_NOSUCHSERVER
RPL_ADMINME RPL_ADMINLOC1
RPL_ADMINLOC2 RPL_ADMINEMAIL

Exemplos:

tolsun.oulu.fi ADMIN; solicitar uma resposta admin
tolsun.oulu.fi

syk ADMIN; pedido ADMIN para que o servidor
syk qual o utilizador está conectado

3.4.10 comando Informações

Comando: INFO

Parâmetros: [<meta>]

O comando INFO é necessário para retornar informações que descrevem o servidor: sua versão, quando foi compilado, o patchlevel, quando se foi iniciado, e qualquer outra informação que pode ser variado considerado relevante.

Curingas são permitidos no parâmetro <target>.

As respostas numéricas:

```
ERR_NOSUCHSERVER
RPL_INFO RPL_ENDOFINFO
```

Exemplos:

INFO csd.bu.edu; solicitar uma resposta INFO de
csd.bu.edu

INFO Anjo; pedir informação do servidor que
Angel é conectado.