

FTP Tolerante a Intrusões

José Pascoal Tiago Jorge
Miguel Correia Nuno F. Neves Paulo Veríssimo

Faculdade de Ciências da Universidade de Lisboa
LASIGE, grupo Navigators

Sumário

- Tolerância a intrusões e Wormholes
- Concretização do WOO
- O serviço de FTP tolerante a intrusões
- Conclusões

1. Tolerância a intrusões e Wormholes

Tolerância a Intrusões

- *aplicar o paradigma da tolerância a faltas no domínio da segurança*
 - ☞ assumir e aceitar que o sistema permanece sempre vulnerável
 - ☞ assumir e aceitar que os componentes do sistema podem ser atacados e que alguns desses ataques terão sucesso
 - ☞ garantir que o sistema como um todo permanece seguro e operacional, ou seja, que não falha

Serviços Distribuídos TI

SERVIÇO DISTRIBUÍDO TI

Objectivos:

- integridade
- disponibilidade



Replicação de Máq. de Estados

- solução genérica para a concretização de *serviços* tolerantes a faltas... e intrusões
- cada servidor é uma máquina de estados definida por variáveis de estado; comandos atômicos
- mínimo **$N=3f+1$** réplicas para tolerar **f** réplicas maliciosas
- qq n^o de clientes maliciosos

SERVIÇO DISTRIBUÍDO TI



PEDIDO

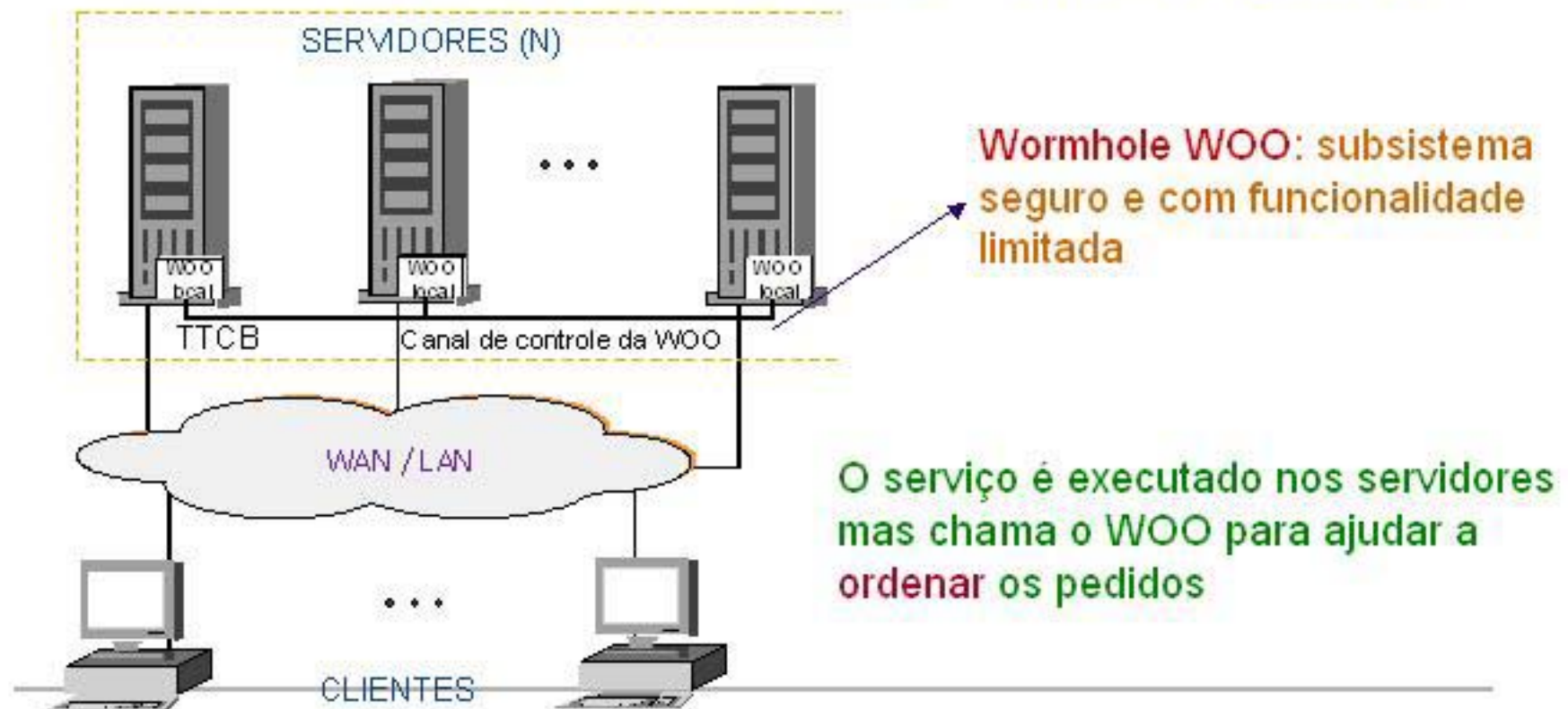
RESPOSTA



CLIENTES

Sistema com Wormhole

- componente privilegiada distribuída que pretende lidar com algum tipo de incerteza – aqui *insegurança*



Wormholes e TI

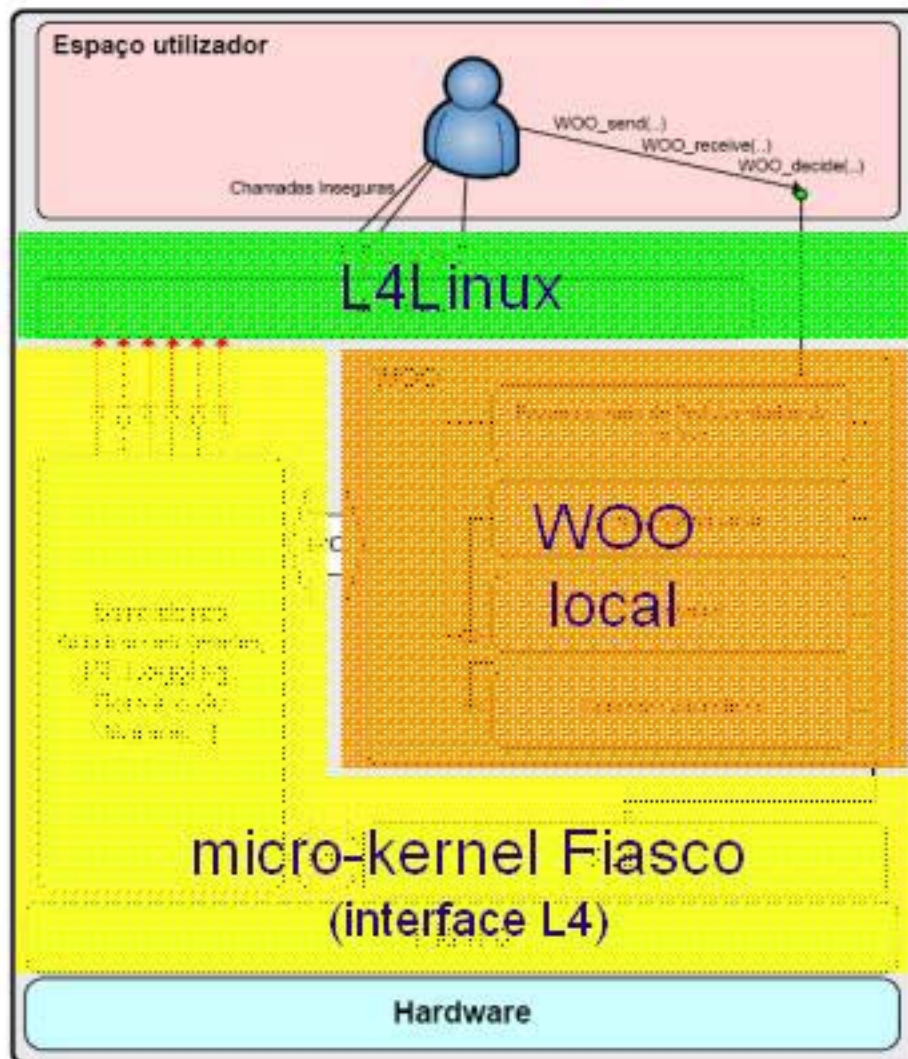
- Contribuições:

- ☞ protocolos simples e eficientes
- ☞ dispensar hipóteses temporais sobre o sistema "normal"
- ☞ diminuir número mínimo de máquinas de estados replicadas de $3f+1$ para $2f+1$
- ☞ contornar impossibilidade de fazer recuperação proactiva em sistemas assíncronos
 - 3 em vez de 4 para tolerar 1 réplica maliciosa
 - 5 em vez de 7 para 2...
 - poupa-se de 25% a 33% do nº de réplicas!

2. Concretização do WOO

Wormhole Ordering Oracle

Arquitectura local

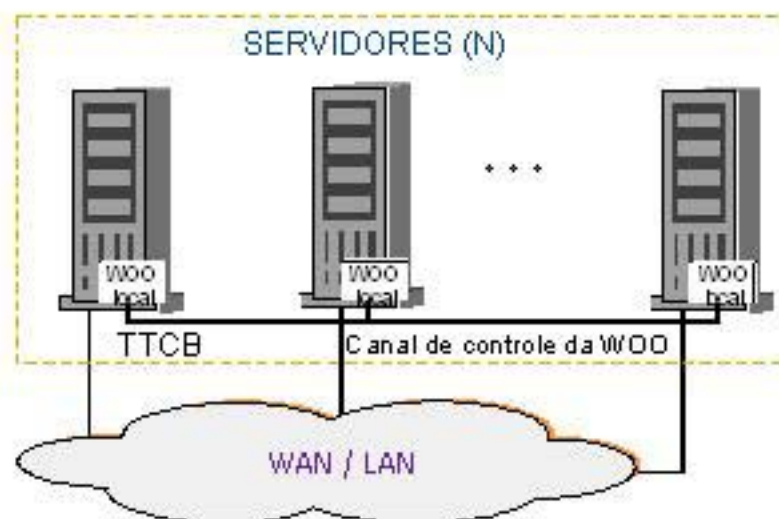


Micro-kernel:

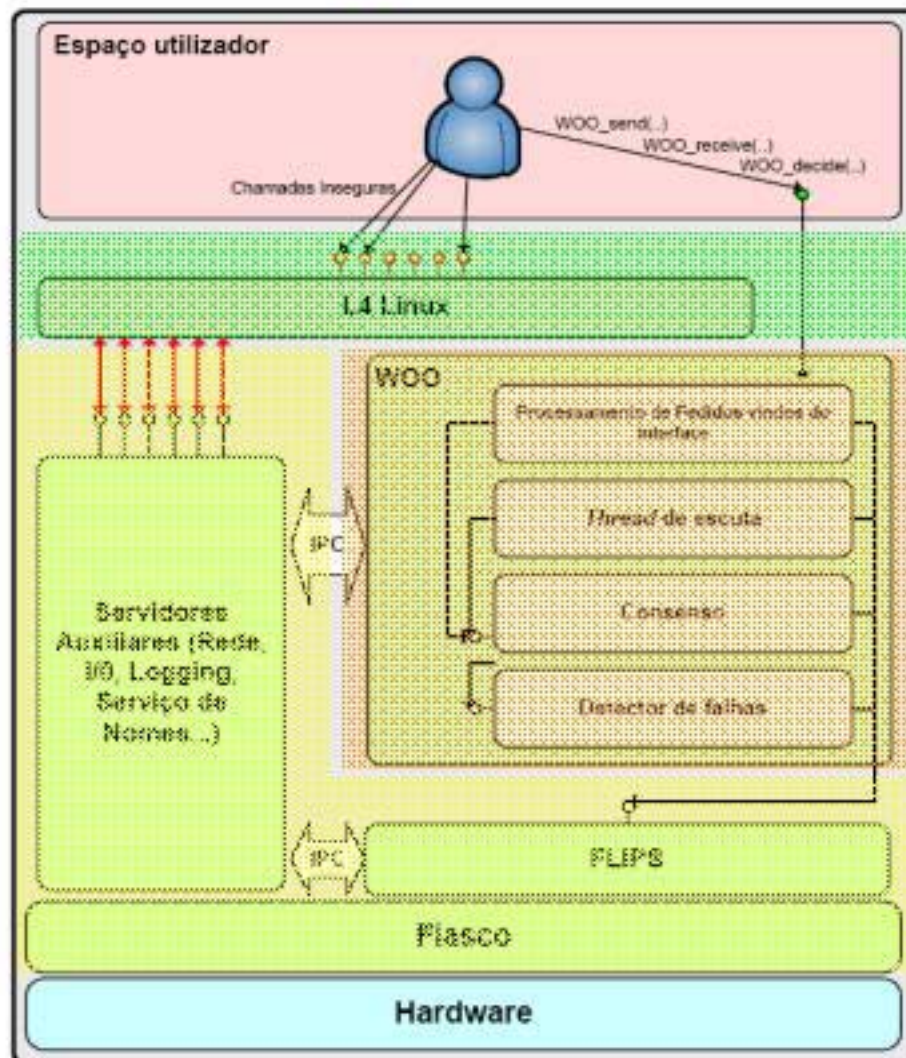
- concretiza monitor de referência
- permite isolar o WOO local do espaço utilizador e do SO

Segurança do WOO

- isolamento dentro do Fiasco
- análise estática do código
 - ☞ Flawfinder, RATS
 - ☞ 128 potenciais vulnerabil. (analisadas manualm/)
- canal de controle (LAN)
 - ☞ segurança física assumida
 - ☞ acesso nas máquinas reservadas ao Fiasco/WOO
- controle de admissão para evitar ataques DoS



Arquitectura numa máquina

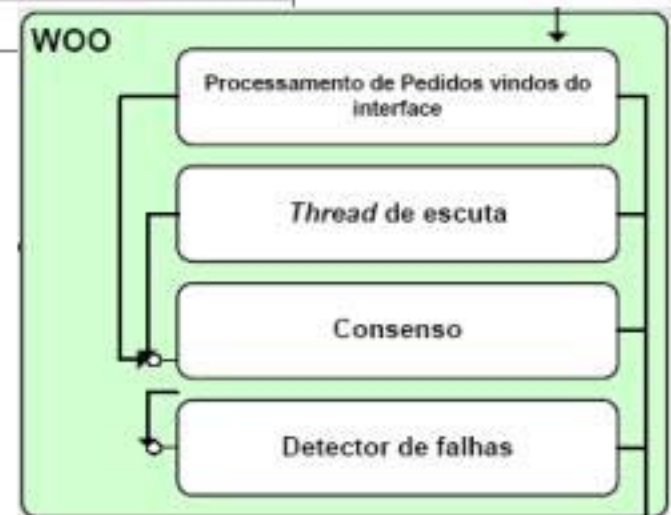


- L4Env – ambiente básico (I/O, locks, servidor de nomes,...)
- l4vfs – suporte de ficheiros
- FLIPS – suporte de rede

Trusted Multicast Ordering

- único serviço disponibilizado pelo WOO

Função	Assinatura
WOO Send	WOO_Send_retval WOO_send (eid, elist, threshold, msg_id, msg_hash)
WOO Receive	WOO_Receive_retval WOO_receive (eid, elist, threshold, msg_id, msg_hash, sender_eid)
WOO Decide	WOO_Decide_retval WOO_decide (tag)



3. O serviço de FTP tolerante a intrusões

Replicação de Máq. de Estados

- todos os servidores seguem a mesma sequência de estados sse:
- Estado inicial. Todos os servidores começam no mesmo estado.
- Acordo. Todos os servidores executam os mesmos comandos.
- Ordem total. Todos os servidores executam os comandos pela mesma ordem.
- Determinismo. O mesmo comando executado no mesmo estado inicial gera o mesmo estado final.

protocolo de
difusão atômica

SERVIÇO DISTRIBUÍDO TI



PEDIDO

RESPOSTA



CLIENTES

Protocolo básico

- Cliente:

- ✎ envia pedido para um servidor protegido com vector de MACs
- ✎ espera por $f+1$ respostas idênticas
- ✎ se após T_{resend} isso não suceder, reenvia pedido para outros f servidores

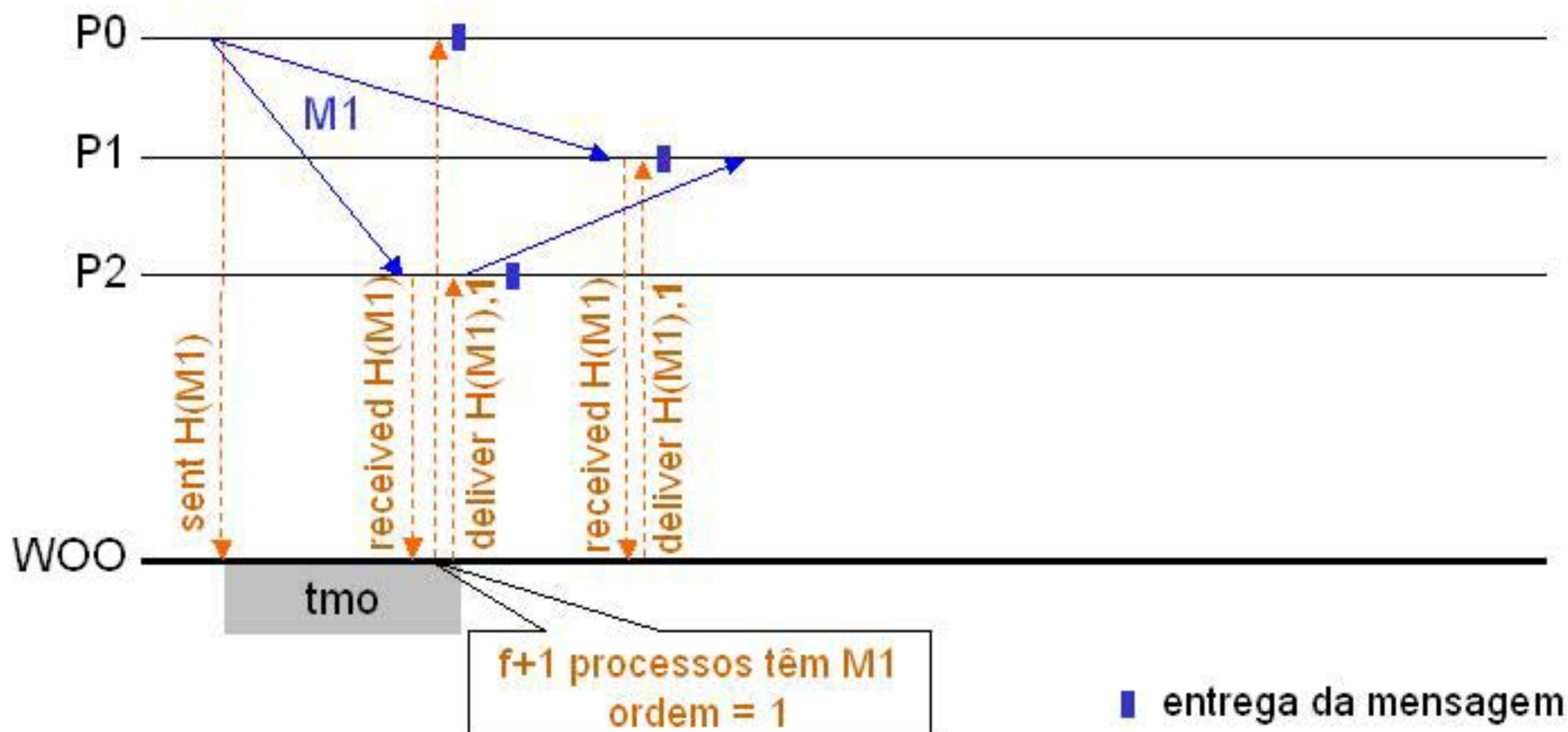
- Servidor:

- ✎ se receber um pedido com o seu MAC correcto, faz **difusão atômica** para todos os servidores
- ✎ quando difusão atômica entrega um pedido (por ordem), processa-o e responde

eficiente e $2f+1$ pois suportada pelo WOO

Execução do protocolo

$n=3$ $f=1$



Serviço de FTP TI

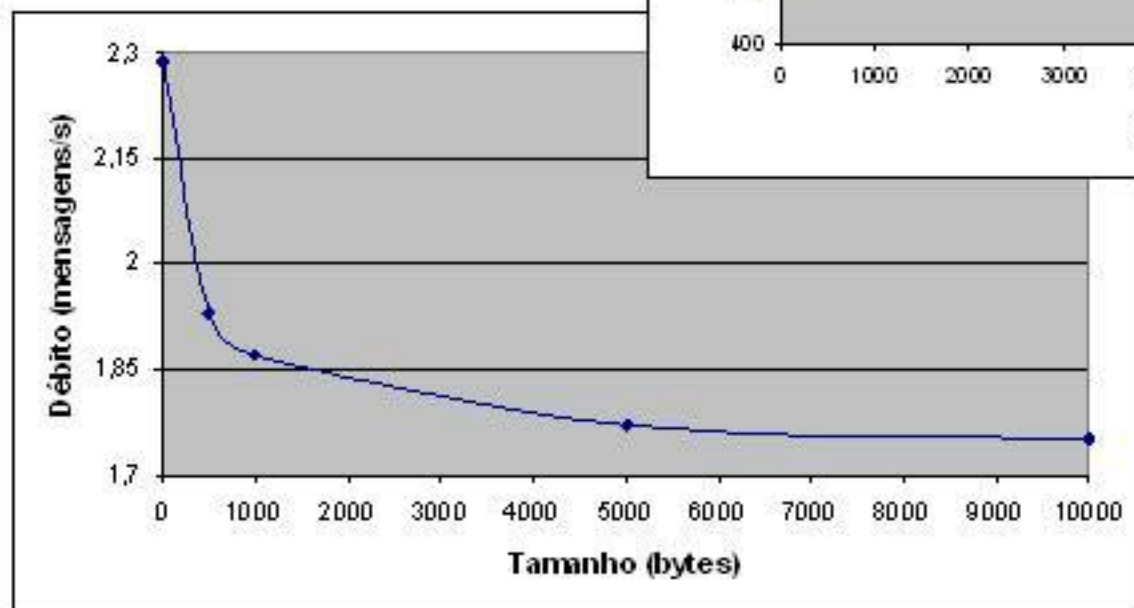
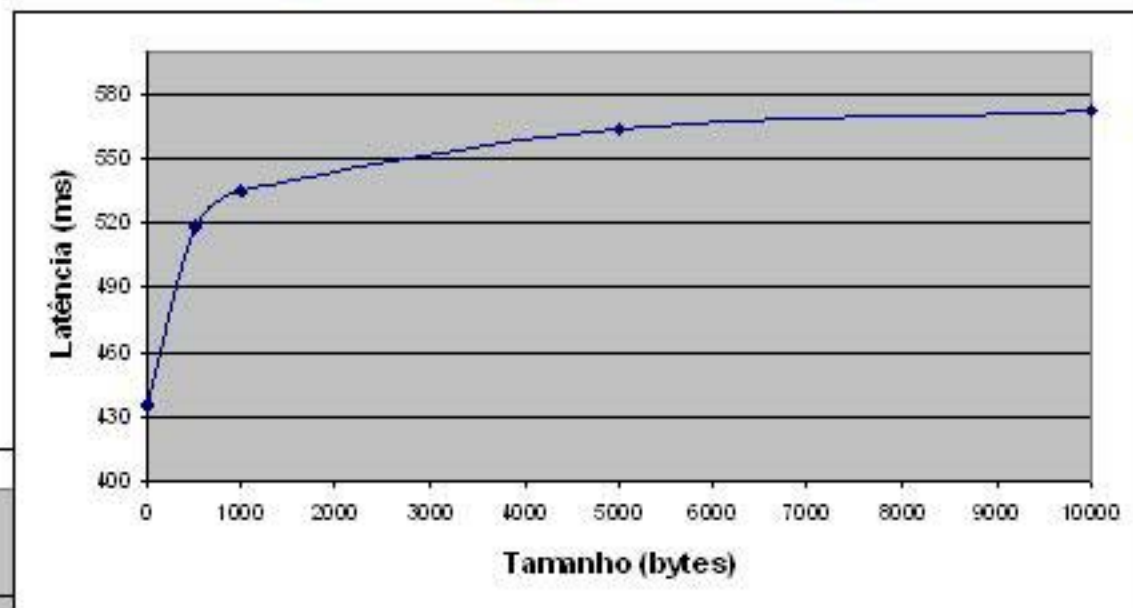
Apenas três tipos de pedidos:

- stor – guardar ficheiro no serviço
 - retr – ler ficheiro do serviço
 - list – listar ficheiros no serviço
-
- Ficheiros são fragmentados em pedaços de 16K

Ambiente das experiências

- 4 PCs Pentium III
 - ☞ 450Mhz com 256 MB SDRAM PC133
- Duas placas de rede Ethernet 10/100 por PC
 - ☞ Uma para rede normal
 - ☞ Outra para canal de controlo do WOO
- 2 switches
 - ☞ 3COM SuperStack II 10/100

Desempenho (carregamento)



Desempenho e L4Linux

- versão do sistema com chamadas ao WOO substituídas por *dummies*
- carregamento de ficheiro pequeno
- latência sobre Linux: 45 ms
- sobre L4Linux e Fiasco: 450 ms !

4. Conclusão

Conclusões

- Wormholes (como o WOO) permitem concretizar serviços distribuídos tolerantes a intrusões com características interessantes:
 - ☞ protocolos simples e eficientes
 - ☞ dispensar hipóteses temporais sobre o sistema “normal”
 - ☞ diminuir número mínimo de réplicas de $3f+1$ para $2f+1$
- Concretização de wormhole com micro-kernel é viável mas prejudica o desempenho face a SO convencional. Solução: hardware.

Perguntas?

- Grupo Navigators:
<http://www.navigators.di.fc.ul.pt/>
- Tolerância a intrusões:
<http://www.navigators.di.fc.ul.pt/it/>

