Utilização do Appia

Tolerância a Faltas Distribuída 2003/04

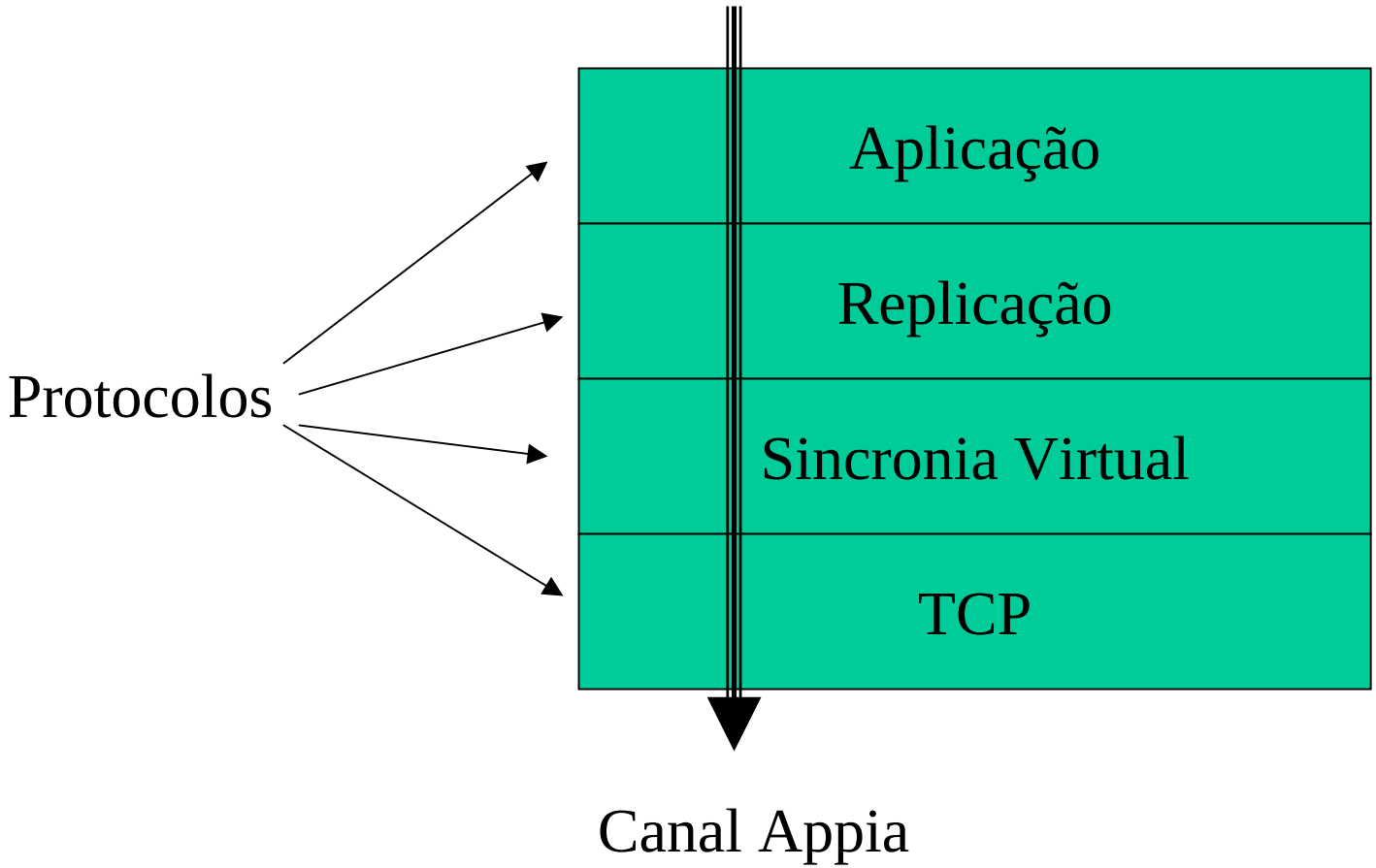
Nuno Carvalho

nunomrc@di.fc.ul.pt

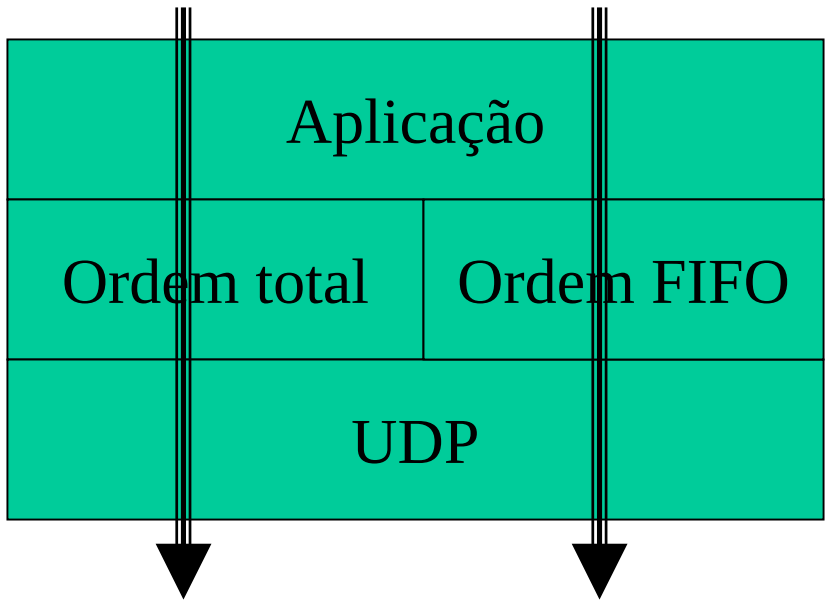
Appia

- “Framework” para execução e composição de protocolos.
- Comunicação efectuada por eventos
  - Entre camadas e entre processos
- Já existem protocolos que podem ser simplesmente usados

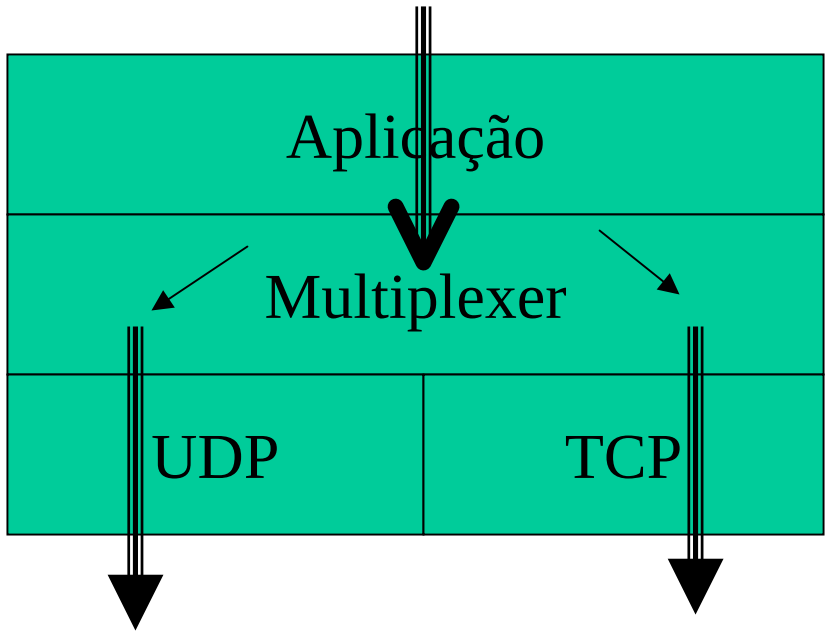
Exemplos de Canais Appia



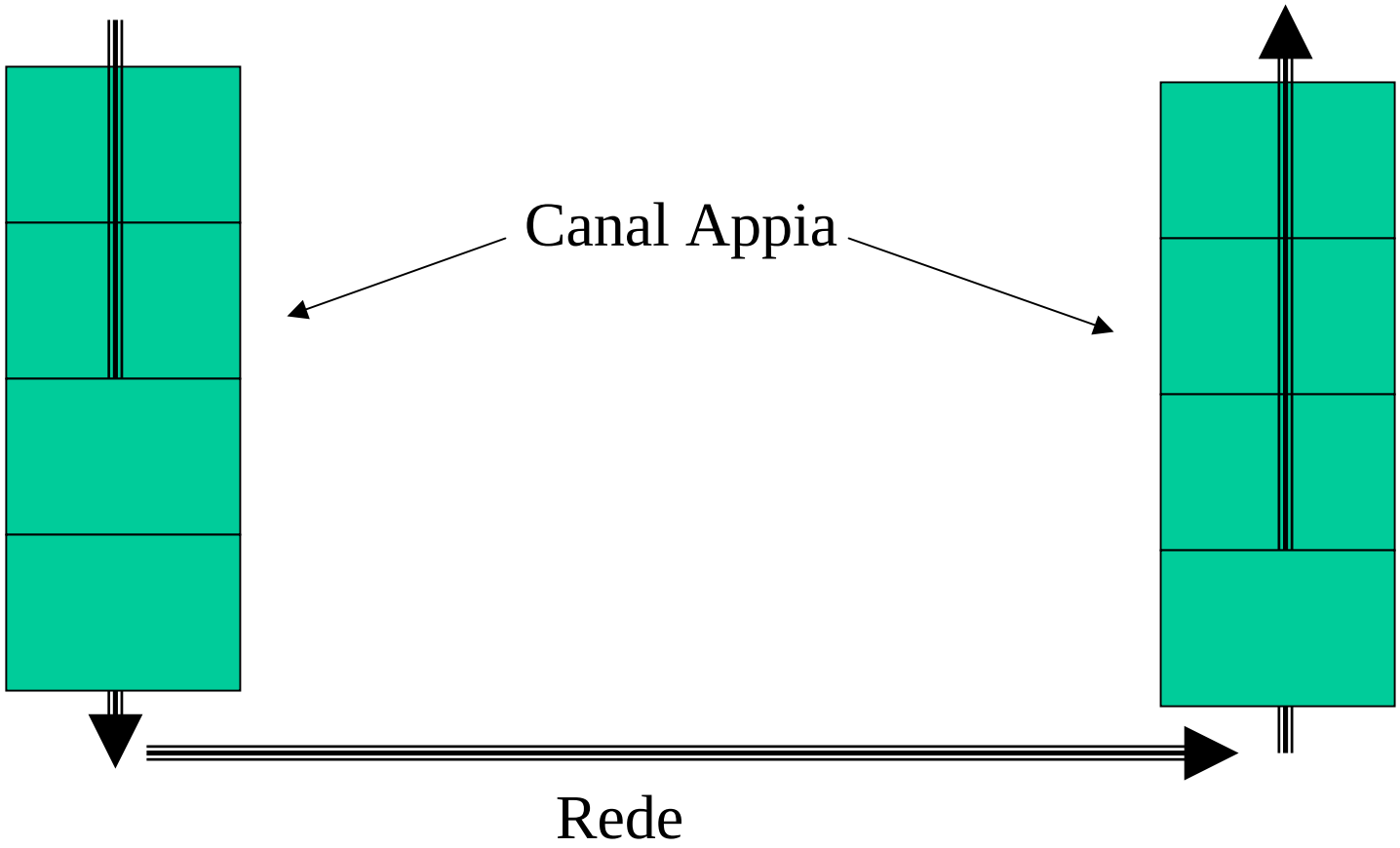
Exemplos de Canais Appia



Exemplos de Canais Appia



Exemplos de Canais Appia



Protocolos existentes no Appia

- **Interface entre um canal Appia e a rede**
  - TCP; UDP; SSL
- **Ordenação**
  - Fifo; Causal; Total
- Fiabilidade nas mensagens (Fifo)
- Consensus
- **Sincronia Virtual**
- Fragmentação de mensagens
- Token Bucket
- RPC’s replicados
- **Aplicações de teste**

Tipos de eventos existentes

- **Event** – Todos os eventos descendem deste
- **ChannelEvent** – Eventos de sinalização interna no canal
  - Timers; ChannelInit; ChannelClose
- **SendableEvent** – Evento a ser enviado pela rede
- **GroupSendableEvent** – descende do SendableEvent e é usado em conjunto com os protocolos de sincronia virtual

Eventos do canal

- **ChannelInit –** é **sempre** o primeiro evento a ser recebido por uma camada;
  - **Nunca** se deve enviar outro evento antes de enviar este.
- **ChannelClose –** Último evento a circular no canal. Indica que o canal vai ser fechado

Eventos com Informação a circular pela rede

- **SendableEvent –** Estende o Event com dois campos (source e dest) e com uma classe do tipo Message
  - InetWithPort – Objecto que contém um IP e um porto, a ser colocado nos campos source e dest.
  - Message – Objecto “serializavel”! É a única coisa que é passada pela rede.

Eventos com Informação a circular pela rede

- **GroupSendableEvent –** tem uma semântica diferente.
  - É usado com os grupos
  - Usa ObjectsMessage ao invés de Message

Message e ObjectsMessage

- Dispõe de métodos para inserir e remover informação
- Funciona como uma pilha
  - push() adiciona informação
  - pop() retira informação
  - peek() devolve a informação sem a retirar da mensagem

Construção de protocolos

- **Layer -** Informação estática
  - Provided – eventos que a camada gera
  - Accepted – eventos aceites pela camada
  - Required – eventos que a camada precisa para o seu bom funcionamento
    - Sub conjunto dos accepted
- **Session -** Instância de um protocolo
  - Código do protocolo
  - Mantém o seu estado
  - Eventos são entregues no método handle(Event e)
  - Podem ser partilhadas por vários canais Appia

Session

- Para fazer um protocolo, basta concretizar o método handle(Event e)
  - Usar o instanceof
  - Nunca bloquear à espera de algo (e.g. teclado)
  - O handle() recebe o evento, trata-o (envia outros, se necessário) e retorna.

Inicialização de um processo Appia

- Array de Layers
- QoS
- Canal (ou canais)
- Criar sessões (se necessário)
- Inicializar o canal (ou canais)
- Appia.run()
  - Já está!!! ☺

Camada aplicação

- Recebe eventos que são gerados fora do canal
- É a camada que inicializa outras camadas
  - RegisterSocketEvent
  - GroupInit
  - Etc...
- É a camada que tem a instância da interface com o utilizador

Exemplo de uma Layer

```
public class BEBLayer extends Layer {

	public BEBLayer(){
		evProvide = new Class[0];

		evRequire = new Class[3];
		evRequire[0] = SendableEvent.class;
		evRequire[1] = ChannelInit.class;
		evRequire[2] = ProcessInitEvent.class;

		evAccept = new Class[4];
		evAccept[0] = SendableEvent.class;
		evAccept[1] = ChannelInit.class;
		evAccept[2] = ChannelClose.class;
		evAccept[3] = ProcessInitEvent.class;
	}

	public Session createSession() {
		return new BEBSession(this);
	}
}
```

Exemplo de uma Session (1)

```
public class BEBSession extends Session {
    private ProcessSet processes;

    public BEBSession(Layer layer) {
        super(layer);
    }

    public void handle(Event event){
        if(event instanceof ChannelInit)
            handleChannelInit((ChannelInit)event);
        else if(event instanceof ProcessInitEvent)
            handleProcessInitEvent((ProcessInitEvent) event);
        else if(event instanceof SendableEvent){
            if(event.getDir()==Direction.DOWN)
                bebBroadcast((SendableEvent) event);
            else
                pp2pDeliver((SendableEvent) event);
        }
    }

    private void handleProcessInitEvent(ProcessInitEvent event) {
        processes = event.getProcessSet();
        try {
            event.go();
        } catch (AppiaEventException e) {
            e.printStackTrace();
        }
    }

    private void handleChannelInit(ChannelInit init) {
        try {
            init.go();
        } catch (AppiaEventException e) {
            e.printStackTrace();
        }
    }
```

Exemplo de uma Session (2)

```
    private void bebBroadcast(SendableEvent event) {
        SampleProcess[] processArray = this.processes.getAllProcesses();
        SendableEvent sendingEvent = null;
        for(int i=0 ; i<processArray.length ; i++){
           try {
                sendingEvent = (SendableEvent) event.cloneEvent();

                // set source and destination of event message
                sendingEvent.source = processes.getSelfProcess().getInetWithPort();
                sendingEvent.dest = processArray[i].getInetWithPort();

                // sets the session that created the event.
                // this is important when this session is sending a cloned event
                sendingEvent.setSource(this);

                // initializes and sends the message event
                sendingEvent.init();
                sendingEvent.go();
           } catch (CloneNotSupportedException e) {
                        e.printStackTrace();
                        return;
           } catch (AppiaEventException e) {
                        e.printStackTrace();
                        return;
           }
        }
    }

    private void pp2pDeliver(SendableEvent event) {
        // just sends the message event up
        try {
                event.go();
        } catch (AppiaEventException e) {
                e.printStackTrace();
        }
    }
}
```

Exemplo de Inicialização de um Processo Appia

```
public class SampleAppl {

    public static void main(String[] args) {
        /* Create layers and put them on a array */
        Layer[] qos = {
            new TcpCompleteLayer(),
            new BEBLayer(),
            new SampleApplLayer()
        };
        /* Create a QoS */
        QoS myQoS = null;
        try {
            myQoS = new QoS("Best Effort Broadcast QoS", qos);
        } catch (AppiaInvalidQoSException ex) {
            System.exit(1);
        }
        /* Create a channel. */
        Channel channel = myQoS.createUnboundChannel("BeB Channel");
        SampleApplSession sas =
            (SampleApplSession) qos[qos.length-1].createSession();
        sas.init(processes);
        ChannelCursor cc=channel.getCursor();
        try {
            cc.top();
            cc.setSession(sas);
        } catch(AppiaCursorException ex) {
            System.exit(1);
        }
        try {
            channel.start();
        } catch(AppiaDuplicatedSessionsException ex) {
            System.exit(1);
        }
        /* All set. Appia main class will handle the rest */
        Appia.run();
    }
}
```

Comecem com coisas simples!!!

- Camadas que concretizam protocolos de difusão fiável
  - Texto explicativo
  - Baseadas no capítulo de difusão fiável do livro

Código disponibilizado para o projecto

- Aplicações de teste
  - Apenas suporta dois jogadores
  - Não usa sincronia virtual
  - Tem comentários importantes ☺
- Javadoc existente em:
  - Gaivota.alunos.di.fc.ul.pt/ ~tfd000

Duvidas...?

- news:informatica.disciplinas.osc.tfd
- nunomrc@di.fc.ul.pt
  - só em caso de extrema necessidade!!!
- http://gaivota.alunos.di.fc.ul.pt/~tfd000
  - Tem exemplos e apontadores importantes para o projecto
  - Inclui uma FAQ!!!
- http://appia.di.fc.ul.pt - informação importante sobre o Appia