Melhoramentos

Não foram feitos nenhuns melhoramentos ao protocolo do projeto.

Concorrência

Para garantir que existia concorrência entre cada um dos subprotocolos do projeto, durante o desenvolvimento, o grupo decidiu implementar a interface *Runnable* na classe *DefaultChannel*, que através de herança foi implementado em cada um dos canais de comunicação. Segue-se um excerto do código desenvolvido na classe *DefaultChannel* e uma das classe que a estende (*ControlChannel*):

Classe *DefaultChannel:*

```
public abstract class DefaultChannel implements Runnable {

	private InetAddress ip;
	private int port;
	private Server server;
	private MulticastSocket socket;
```

Classe *ControlChannel:*

```
public class ControlChannel extends DefaultChannel {

	public ControlChannel(String ip, String port, Server
server) throws IOException {
		super(ip, port, server);
	}

	@Override
	public void run() {
		// TODO Auto-generated method stub
		System.out.println("Control Thread initiated!");

		while(true) {
			byte[] buffer = new byte[65535];
			DatagramPacket packet = new
	DatagramPacket(buffer, buffer.length);
```

Como é possível observar, a classe *ControlChannel*, como estende a classe *DefaultChannel*, dá *override* à função *run()*, resultando assim numa implementação concorrente, pois em cada *peer* são criadas *threads* que "correm" as classes de canal. Esta implementação permite que o programa funcione concorrentemente.

Segue-se o código do *main* da classe *Server*, responsável por inicializar cada peer.

Classe *Server*:

```
	public static void main(String[] args) throws
RemoteException, AlreadyBoundException {

		try {
			Server server = new Server(args);
			ClientInterface client = (ClientInterface)
	UnicastRemoteObject.exportObject(server, 0);
			Registry registry =
	LocateRegistry.getRegistry();
			registry.rebind(server.accessPoint, client);
			System.out.println("Server initiated!");

			//start threads
			Thread controlChannelThread = new
	Thread(server.MC);
			Thread backupChannelThread = new
	Thread(server.MDB);
			Thread restoreChannelThread = new
	Thread(server.MDR);

			controlChannelThread.start();
			backupChannelThread.start();
			restoreChannelThread.start();
		}
		catch(Exception e){
			System.out.println(e.toString());
			e.printStackTrace();
		}
	}
```