

# Protocolos de Comunicación

## Trabajo Práctico Especial

Ignacio Rivera 53029

Federico Tedin 53048

Teresa Di Tada 52354

Javier Fraire 53023

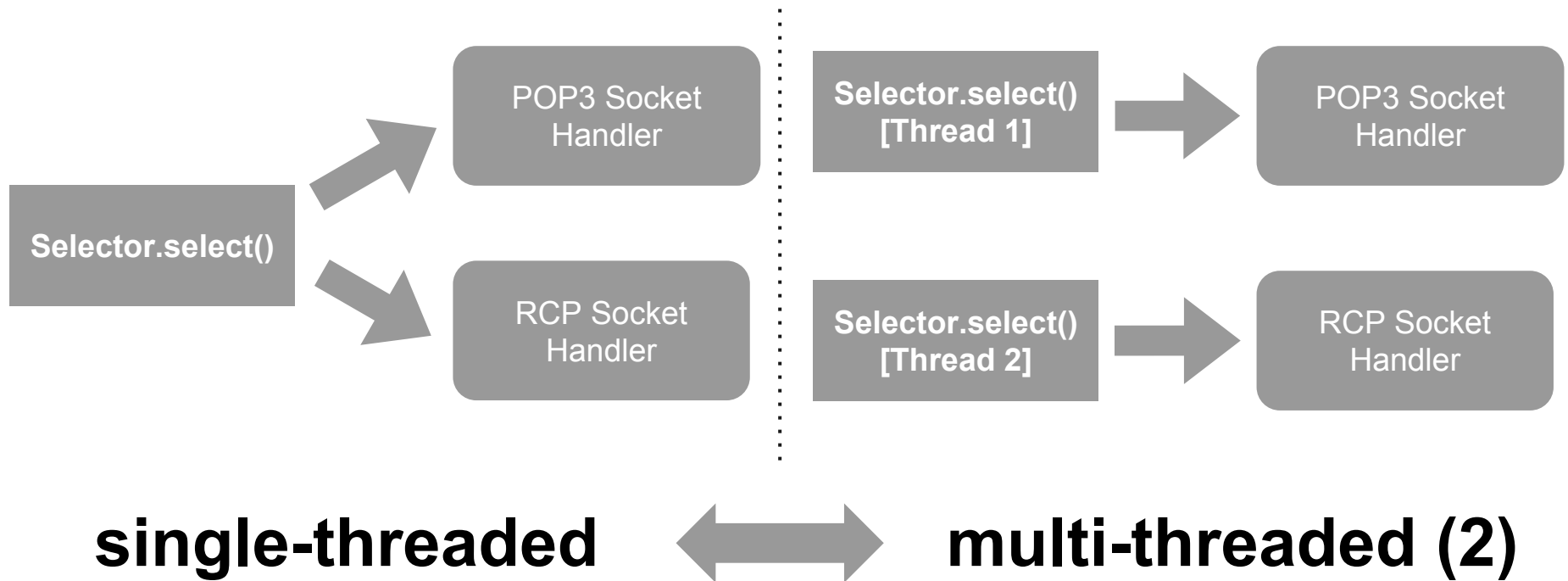
# Características del Proxy POP3

- Mediar una conexión POP3 entre un cliente y un servidor
- Modificar ciertos contenidos de mails
- Multiplexado de cuentas
- Registrar accesos y estadísticas
- Configuración remota mediante un nuevo protocolo

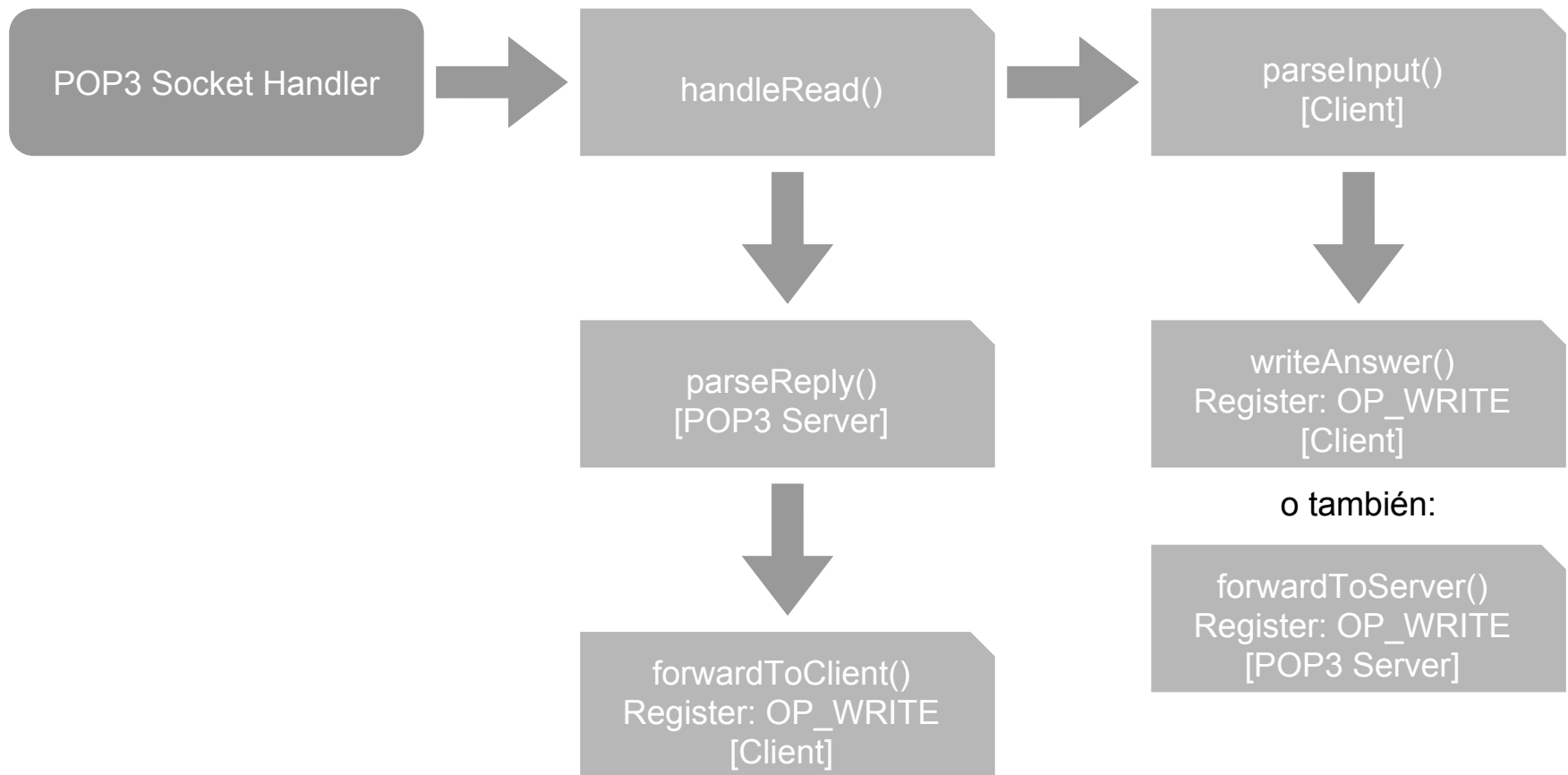
# Conexiones POP3 y RCP

- Todas las conexiones se manejan utilizando un Selector
- Se utilizan dos implementaciones de TCPProtocol para los dos protocolos
- Sockets de POP3 y RCP (Remote Control Protocol) conectados simultáneamente

# Conexiones POP3 y RCP



# Conexiones POP3 (read)

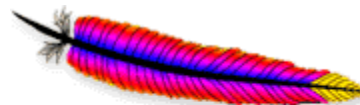


# Conexiones POP3 (write)



# Transformaciones

- Se modifican valores del campo “Subject” de los mensajes recibidos del servidor
- Transformaciones definidas en un .XML
- Se puede agregar y cambiar reglas en tiempo de ejecución



**Apache Commons**<sup>TM</sup>  
<http://commons.apache.org/>

# Multiplexado de Cuentas

- Se tiene un servidor al que el Proxy se conecta por *default*
- Se puede especificar otro servidor para cierto nombre de usuario
- Los valores se almacenan en un .XML:

```
<default>pop3.google.com</default>
<users>
  <user>mike@aol.com</user>
  <server>pop3.aol.com</server>
</users>
```



# *Logging*

- Registrar información sobre accesos, comandos y estadísticas en un archivo
- Mostrar información de usuarios (IP, username, etc.)
- Utilizar Log4j o Apache Commons para facilitar el proceso

# Remote Control Protocol (RCP)

- Parecido a POP3
- Request/Response
- Comandos básicos: PASS, GET, SET, DELETE y QUIT
- Cada comando termina con `\r\n`
- +OK si está todo bien
- -ERR (error code) [message]. El código de error permite identificar el error cometido

# Capacidades de RCP

- Obtener estadísticas y el estado del servidor
- Activar y desactivar y modificar el multiplexado y las transformaciones (e.g default server, binding de un usuario específico)
- Restablecer las estadísticas
- Configurar otras variables que alteran la performance (e.g tamaño de los buffers)

# Ejemplo de RCP

```
Server: +OK \r\n
Client: PASS password \r\n
S: +OK \r\n
C: SET BUFFER_SIZE 2048 \r\n
S: +OK \r\n
C: SET USER mike@aol.com pop3.aol.com \r\n
S: +OK \r\n
C: GET ALGO \r\n
S: -ERR 2 Invalid keyword \r\n
C: GET BYTES \r\n
S: +OK \r\n
S: 65500 \r\n
S: . \r\n
C: GET STATS \r\n
S: +OK \r\n
S: BYTES 65500 \r\n
S: ACCESS_COUNT 2543 \r\n
S: . \r\n
C: QUIT \r\n
S: +OK \r\n
```

# Reactor + Thread Pool

- Si se utiliza un solo thread, el programa no responde a nada mientras está en un handler
- Utilizar un pool de una cantidad pequeña y fija de “worker threads”
- Al llegar un evento, se corre el handler en un “worker thread” del pool
- Al inicializarlos al principio, no hay costo extra cada vez que se responde a un evento

# Reactor + Thread Pool

- Mejora la performance
- Usar pocos threads evita el consumo excesivo de recursos y facilita la concurrencia y organización
- Disminuye el congestionamiento del event handler

# Reactor + Thread Pool

