Java ATM API

B. Yu, Z. Huang, S. Cranefield and M. Purvis
Department of Computer Science
University of Otago,
New Zealand
Introduction

- ATM network provides QoS for multimedia applications
- Java is good for network applications
- ATM-aware applications hard to implement
- Using Java ATM API easy to develop ATM-aware applications
- Java ATM API combines benefits of Java and ATM
Java ATM API
Objectives of the Java ATM API’s Design

- Make Java network applications communicate over ATM network
- Make the API follow the convention of Java network functions
- Provide flow and error control
Implementation of the Java ATM API

- First Java ATM API was implemented on the Window NT OS (2000) using Java Native Interface which calls ATM API
- We ported the Java ATM API on the Linux OS (2001)
- We implemented flow & error control for data transfer over ATM networks based on credit-based and selective repeat protocols
Credit-based Flow Control

When the credits reach zero, the server can’t send data until it receives credit from the client.

Whenever server sends data it decreases the credits.

When client application reads data, a buffer is freed so IT increments the credits unnoticed by the server.
Selective Repeat Error Control

OT: output thread  OB: output buffer  OSPB: outstanding packet buffer
IT: input thread  IB: input buffer  ICPB: incoming packet buffer
UOB: user output buffer  UIB: user input buffer  MT: main thread

Data flow  control relationship
Performance Evaluation (1)
(Maximum Throughput)

Mbps

<table>
<thead>
<tr>
<th></th>
<th>Java ATM 25</th>
<th>Java ATM 155</th>
<th>LANE 155</th>
<th>ttcp_atm</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>28</td>
<td>105</td>
<td>86.84</td>
<td></td>
</tr>
</tbody>
</table>
Performance Evaluation (2) (Return Trip Time)

- **Java ATM 155**
- **LANE 155**
- **Ethernet 100**

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Java ATM 155</th>
<th>LANE 155</th>
<th>Ethernet 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>0.27</td>
<td>0.108</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>0.425</td>
<td>0.324</td>
<td></td>
</tr>
<tr>
<td>2048</td>
<td>0.67</td>
<td>0.43</td>
<td></td>
</tr>
</tbody>
</table>

**Times (ms):**
- 3.05
- 3.4
- 3.4
Java Multimedia Application using the Java ATM API

- Multimedia Application
  - Video server and client system using Java Media Framework (JMF)
  - *atm protocol used e.g.*
    - `atm://00:20:2b:03:08:44/atm.mpg`

- File Transfer Application
  - `ATMServerSocket(PORT, 50, "00:20:2b:03:08:44")`
  - `ATMSocket("00:20:2b:03:08:39", 5678)`
Java Applet using the Java ATM API
Future Research

- Current Java ATM API did not include end-to-end QoS guarantee
- Design and implementation of ATM protocol for multimedia applications
- Solve the large latency problem
Question?