A³: Application-Aware Acceleration for Wireless Data Networks

Zhenyun Zhuang and Tae-Young Chang
GNAN Research Group, Georgia Tech, Atlanta, GA

Raghupathy Sivakumar and Aravind Velayutham
Asankya Networks, Inc., Atlanta, GA
Introduction

Wireless Environments
- High loss rate
- Large delay
- Low bandwidth

Works focused on developing better transport protocols for wireless environments
- TCP-ELN, WTCP, STP, etc.
- Novel design, deals with unique characteristics
- Improves throughput significantly

Evaluation of transport protocols
- FTP, or bulk of data
Analysis of Enterprise Traffic

**Figure 3:** Percent of Enterprise Traffic

- Client-server: CIFS
- Email: SMTP
- Web applications: HTTP

*Figure from Business Communications Review (April 2006)*
Motivation: Network Setup

- Traffic Generator
  - Ixia IxChariot (IxProfile, Console, Endpoint)

- Applications
  - FTP, CIFS, SMTP and HTTP

- Wireless Networks
  - Wireless LAN (WLAN)
  - Wireless WAN (WWAN)
  - Satellite Networks (SAT)

- Transport Protocols
  - NewReno, TCP-ELN, WTCP, STP

- Parameters
  - Varying loss rate (RTT, BDP)
Motivation Results: FTP

- Significant performance improvement...
  - Up to 120% in satellite networks
Motivation Results: Other Applications

- Less performance improvement in other applications...
  - 5%!

- CIFS
- SMTP
- HTTP
Application Traffic Patterns

FTP

Client

Command

Data

Server

Data

CIFS

Client

Establish NetBIOS Session

Negotiate CIFS Dialect

Choose CIFS Dialect

User Login

Connect to Resource

Open A File

Request Data Block

Server

Positive Session Ack

Negotiate CIFS Dialect

Choose CIFS Dialect

User ID

Tree ID

File ID

Data 1

Data 2

1

Data 1

Data 2

2
Application Traffic Patterns (cont.)

SMTP

Client
- Connect to server
- 200 smtp.receiver.com Ready
- HELO mail.sender.com
- 250 smtp.receiver.com
- MAIL FROM: david@sender.com
- 250 OK
- RCPT TO: bod@receiver.com
- 250 OK
- DATA
- End of Data
- 250 OK
- Quit
- 221 Service Closing

Server

HTTP

Client
- HTTP Request (GET)
- HTTP 200 OK
- HTTP Request (GET)
- HTTP Request (GET)
- HTTP Request (GET)
- HTTP Request (GET)

Server
**Typical Application Behaviors**

- **Thin Session Control Messages**
  - Sent before DATA; Small
  - Retransmission timer expires to recover loss

- **Batched Data Fetch**
  - Data transfer is performed in batches
  - Bandwidth Delay Product cannot be fully utilized

- **Flow Control Bottlenecked Operations**
  - When applications are slow, receive buffer fills up
  - Flow control can kick in and be the bottleneck

- **Non-prioritization of Data**
  - Data are given equal importance

- **Non-use of Data-reduction Techniques**
  - Application-specific and user-specific information
A³: Application-Aware Acceleration

- Application aware
  - Recognize applications
- Application transparent
  - No modifications to applications
- A set of design elements
  - Transaction Prediction (TP)
  - Redundant and Aggressive Retransmissions (RAR)
  - Prioritized Fetching (PF)
  - Infinite Buffering (IB)
  - Application-aware Encoding (AE)
**A³: Transaction Prediction (TP)**

**Transaction Prediction**
- Deterministically predict future requests
- Issue them ahead of time
- Designed for protocols that divide data into blocks
- Examples: CIFS, HTTP
Redundant and Aggressive Retransmission (RAR)

- Helps protect thin session control messages from losses
- Packet-level redundancy
- Aggressive retransmission
- Not applying to DATA
  - Loss recovery is masked by subsequent packets
  - High overhead

![SMTP Throughput Graph](image-url)
Prioritized Fetching (PF)

- Divide data into categories of different priorities
- Fetch them with different speeds
- Helps protocols that treat data with equal importance
  - Example, HTTP
Infinite Buffering (IB)

- Prevents flow control from throttling the transmissions
- Uses local storage to store data at the receiver
- Flow control never kicks in

CIFS: Throughput vs Application Rate

![Graph showing throughput vs application rate]

14 of 20
Application-aware Encoding (AE)

- Uses application and user specific information
- Better compress data
- Example, SMTP

### 10 Persons (100 emails)

<table>
<thead>
<tr>
<th>ID</th>
<th>Unique Word</th>
<th>Total Word</th>
<th>Char. per Word</th>
<th>Bits per Email of Binary Coding</th>
<th>Bits per Email of Simple Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1362</td>
<td>6383</td>
<td>6.22</td>
<td>3176</td>
<td>664.6</td>
</tr>
<tr>
<td>2</td>
<td>3554</td>
<td>19284</td>
<td>7.12</td>
<td>10984</td>
<td>2274.6</td>
</tr>
<tr>
<td>3</td>
<td>2645</td>
<td>12653</td>
<td>7.08</td>
<td>7167</td>
<td>1438.5</td>
</tr>
<tr>
<td>4</td>
<td>4536</td>
<td>25481</td>
<td>6.15</td>
<td>12537</td>
<td>3095.2</td>
</tr>
<tr>
<td>5</td>
<td>966</td>
<td>4728</td>
<td>11.46</td>
<td>4335</td>
<td>468.8</td>
</tr>
<tr>
<td>6</td>
<td>1205</td>
<td>6413</td>
<td>5.48</td>
<td>2811</td>
<td>656.4</td>
</tr>
<tr>
<td>7</td>
<td>798</td>
<td>3346</td>
<td>4.40</td>
<td>1178</td>
<td>322.6</td>
</tr>
<tr>
<td>8</td>
<td>1527</td>
<td>6836</td>
<td>5.72</td>
<td>3128</td>
<td>723.0</td>
</tr>
<tr>
<td>9</td>
<td>1758</td>
<td>9171</td>
<td>4.91</td>
<td>3602</td>
<td>988.6</td>
</tr>
<tr>
<td>10</td>
<td>1402</td>
<td>8320</td>
<td>7.3</td>
<td>4859</td>
<td>869.7</td>
</tr>
</tbody>
</table>

### Word Frequencies of a Person

![Graph showing cumulative percentage of usage vs. percentage of top words]
A³ Deployment Model

- Client side is a software module
- Server side can be software modules installed on application servers, or packet processing appliances
- Point solution is also possible
- Implementation with Netfilter for Linux Systems
Evaluation Setup

- Application Emulator (AppEm)
- A³ Emulator (A³Em)
- Wireless Network Emulator (WNetEm)

<table>
<thead>
<tr>
<th></th>
<th>WLAN</th>
<th>WWAN</th>
<th>SAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW (Mb/s)</td>
<td>5</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>RTT (ms)</td>
<td>5</td>
<td>200</td>
<td>1000</td>
</tr>
<tr>
<td>Loss (%)</td>
<td>1</td>
<td>8</td>
<td>3</td>
</tr>
</tbody>
</table>
Performance Evaluation

- CIFS
  - RAR, IB and TP
  - 70% improvement

- SMTP
  - RAR, IB and AE
  - 110% improvement

- HTTP
  - RAR, IB and PF
  - 30% improvement
Related Works

- WAP: Wireless Application Protocol
- PIE: Pocket Internet Explorer
- Odyssey [Noble 1997]
- Coda [Satyanarayanan 1990]
- “Upload” client-tasks to server side [Czerwinski 2001]
- Out-of-order HTTP objects transmitted in UDP [Mohomed 2006]
- Commercial WAN optimizers [Riverbed, etc]
Summary

- Use emulations to test performances of several popular used applications
- Identify five application behaviors, and analyze their impacts on performances
- Propose an application-aware acceleration solution
  - TP, RAR, PF, IB, and AE
- Evaluate its effectiveness

GNAN Research Group : http://www.ece.gatech.edu/research/GNAN/
Asankya Networks, Inc. : http://www.asankya.com/