

A Component-Based Active Network System for Satellite Platforms

Stefanos Zachariadis¹, Saleem Bhatti¹, Wolfgang
Fritsche², Gerhard Gessler², Peter Kirstein¹, Karl
Mayer² and Lionel Sacks³

<http://www.cs.ucl.ac.uk/staff/s.zachariadis>

s.zachariadis@cs.ucl.ac.uk

1. Department of Computer Science, University College London
2. IABG mbH
3. Department of Electronic & Electrical Engineering, University College London



Outline

- Introduction
- Active Networks and Satellite Active Network Systems
- The SATIN Component Based Platform
- Transcoding JPEG2000
- Conclusions



Introduction

- Internet Communication is Best of Effort
- Delivery of Multimedia Services Difficult
 - Strict requirements on latency and bandwidth
 - Violations result in artifacts / interruption
- Reducing Congestion in Terrestrial Networks:
 - More routers
 - Larger pipes
 - Installation / customisation of QoS software at key nodes



Satellite Networks?

- More and more streams over satellite
- IP over DVB-S(2)
- Satellite hardware becoming more advanced
- What to do about introduction of new services?
 - Given longevity, difficult to predict
- What to do about congestion?



Satellite Networks!

- The frequency bandwidth per transponder is limited
- More capacity == more satellites
- Installation / customisation of QoS software is tough
- Satellite software built with antiquated engineering methodology
 - Monolithic, deployed once, never modified
 - Deployment of new services difficult
 - Dynamic / intelligent customisation difficult



Active Networks

- Respond intelligently to changes in network environment and user requirements
- Programmable, allowing the assertion of policies (declarative) and evaluation of algorithms (algorithmic) on content and flow of data
- Fixed API
- Often realised using mobile code



Component Systems

- Adaptable component-based systems
- Component == dynamically deployable functionality
- Now suitable for systems with limited resources
- We require
 - Pipe flow through component
 - Deployment
 - Control



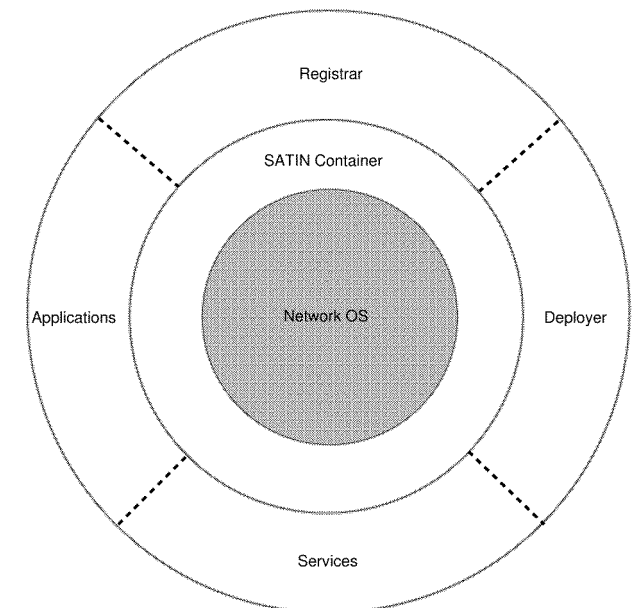
Satellite Active Networks

- Longevity of platform makes AN dynamicity very desirable, but:
- Security
 - Malicious components
 - Bad components
- Limited Resources



SATIN

- Component based system
 - Everything is a component
- Used in various projects
- J2ME, 150329 bytes
- Offers:
 - Reflection
 - Security
 - Logical Mobility
 - Advertising and Discovery
 - Dynamic Instantiation

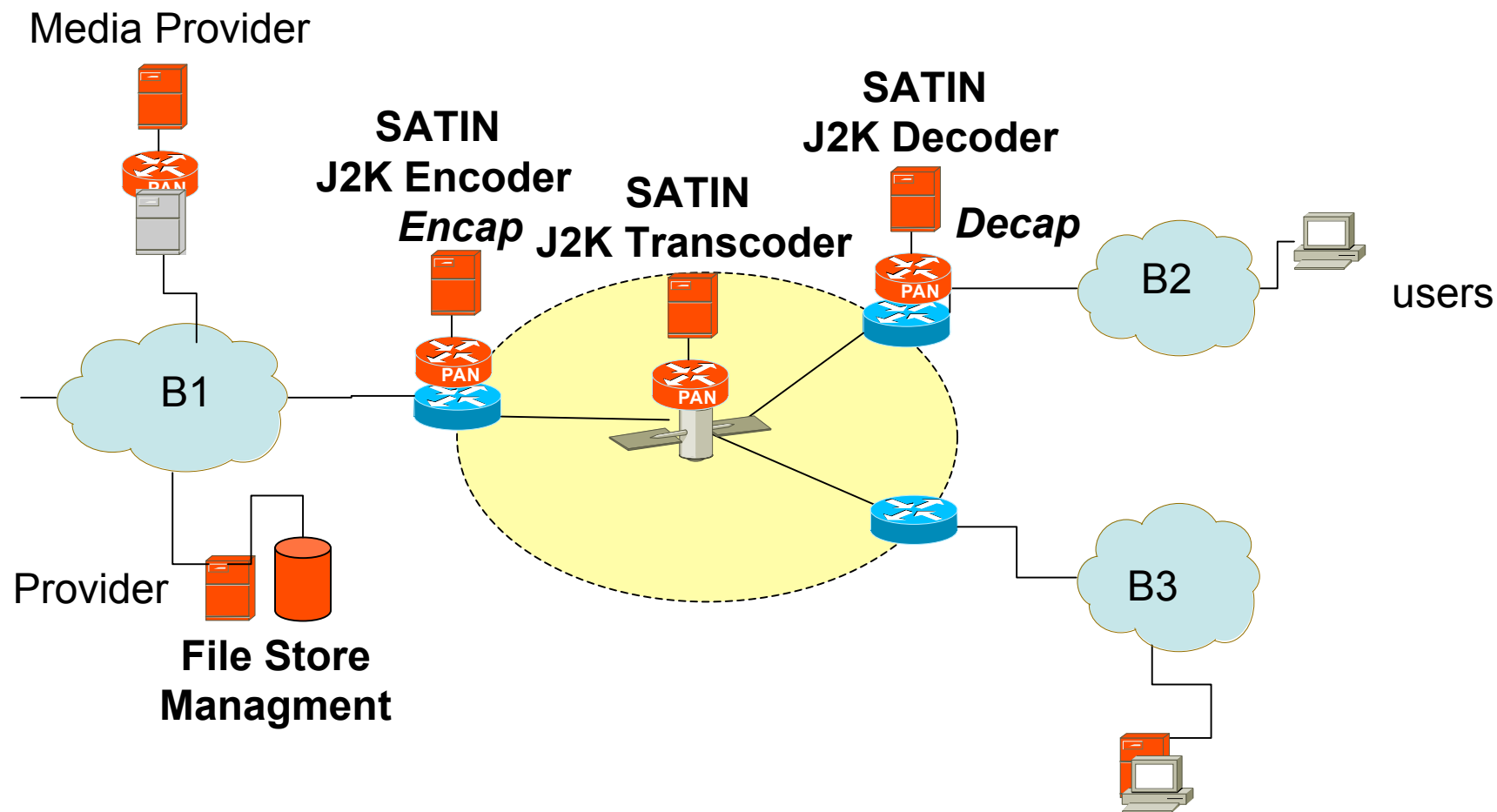


A Transcoding Service

- Change stream
 - Reduce size
- Why transcode?
 - Improve QoS
 - Conserve bandwidth
- How to transcode?
 - Drop
 - Reduce bpp
 - Reduce resolution



Transcoding JPEG2000



Satellite Platform Simulation

- Linux and Linux TC framework for Satellite
 - Latency (250ms) & different spot beams (512kbps, 64kbps)
- Apache for media provider
 - Still images and slideshow
- SATIN adaptable http proxy component for satellite access
- Adaptation of JPEG2000 Reference implementation in Java for transcoder
- Linux terminals (Mozilla) for clients
- Dynamic (runtime) deployment of everything



Results

- Deploy and reprogram the transcoder to offer same QoS to users (18 to 3 seconds)
 - Reduce bpp and/or
 - Reduce size



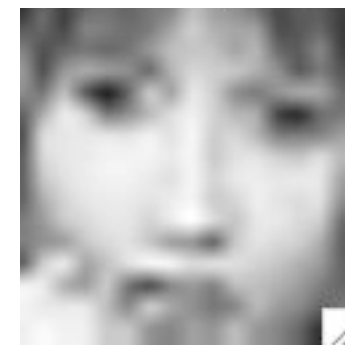
3.55 bpp



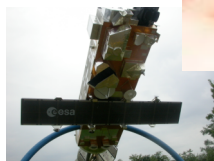
0.35 bpp



0.034 bpp



0.0091 bpp



Conclusions

- SATIN suitable for sending code/parameters/policies
- SATIN can expose & manipulate streams but not individual packets
- Need better management architecture





Thank You!

Any Questions?

Released (LGPL)

<http://satin.sourceforge.net>

