

Haggle: a Networking Architecture Designed Around Mobile Users

Agustí Pellicer

- Hagggle: a Networking Architecture Designed Around Mobile Users by James Scott, Pan Hui, Jon Crowcroft and Christophe Diot. (WONS, 2006)
- Also for background on the idea of Hagggle: <http://vimeo.com/36560033> (Pursuit Summer School, August 2011)

Index

- Introduction
- Pocket Switched Networking
- Problems with current architecture
- New architecture principles
- Hagggle architecture
- Pros & Cons

Introduction

- Mobile world
- Different networks
- Dependent
- Mobile applications

Pocket Switched Networking

- Scenario:
 - Users with one or more mobile devices
 - Islands of connectivity
 - Devices within wireless range

Pocket Switched Networking

- Data can be transferred:
 - Namely neighborhood connectivity
 - Infrastructure connectivity to the Internet
 - User mobility which can physically carry data

Pocket Switched Networking

- Known-sender:
 - Destination another user, all users with a certain role, etc.
 - Network endpoints specified by user level naming
- Known recipient:
 - Device requests some data
 - Source of the data has to be reachable using the previous transfer methods

Pocket Switched Networking

- Resource management:
 - Limited resources in mobile
 - Processor, storage, memory always rising

Problems with current architecture

- Typically designed around infrastructure that is not always available (offline mode)
- Neighborhood connectivity not widely used
- Situation-context for a single task

Problems with current architecture

- Synchronous
- Data kept by applications
- Applications have no control on prioritization of mobile resources

New architecture principles

- Forward using application layer information
 - Endpoints specified using higher layer information
- Asynchronous operation
 - Next hop node can be left as late as possible
- Empower intermediate nodes
 - Intermediate nodes may also be valid destinations for data

New architecture principles

- Message switching
 - Application level messages should be exchanged by neighboring Hagggle nodes
- All user data kept network-visible
 - Data tagged with metadata
- Build request-response into the network

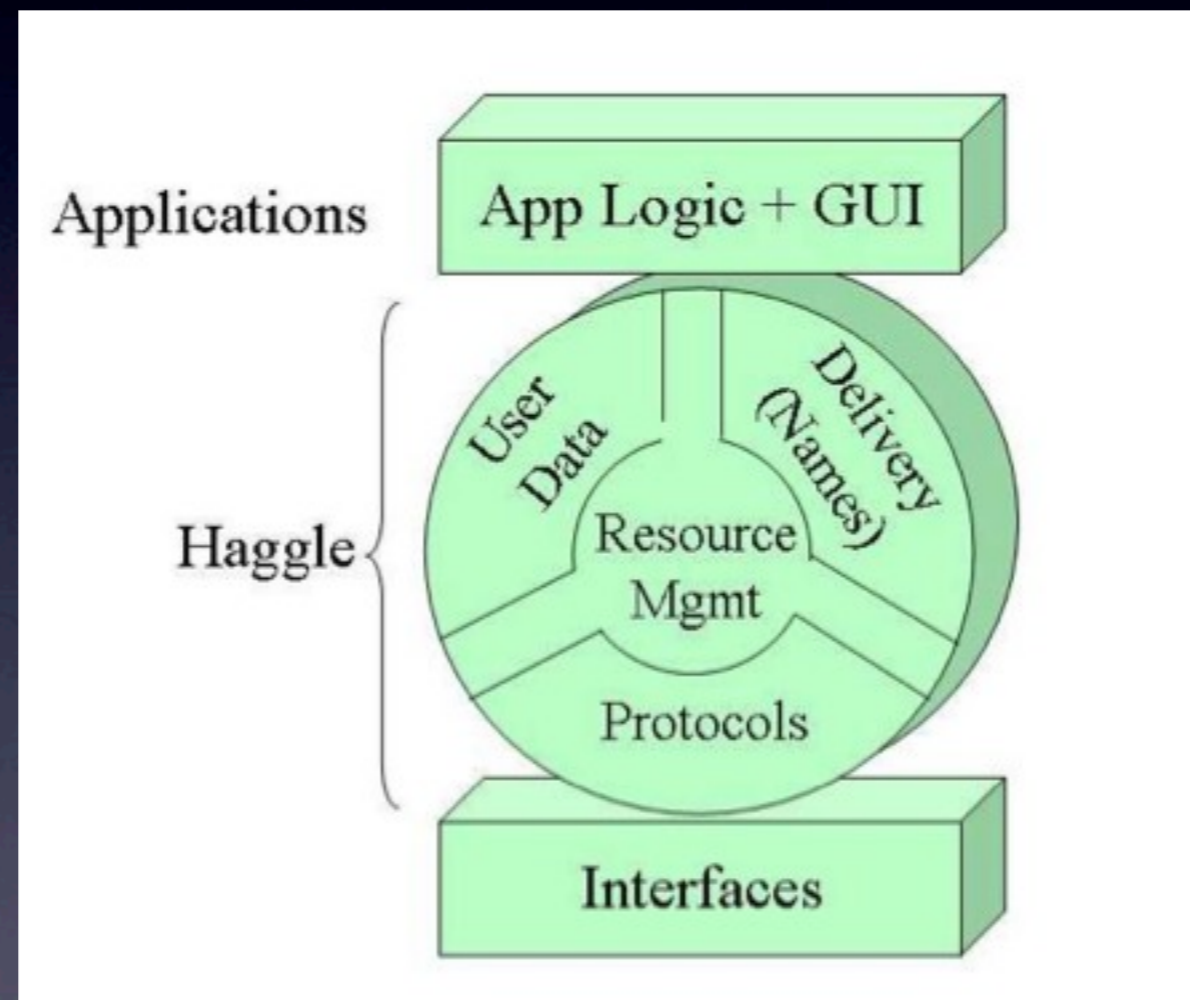
New architecture principles

- Exploit all data transfer methods
 - Mapping of interfaces to nodes, used in different contexts
- Take advantage of brief connection opportunities
- Neighbor discovery is critical

New architecture principles

- Empowered and informed resource management
 - Huggle manages the storage
 - Network resources determined by user-level priorities
 - Battery management using context-awareness
- Use and integrate with existing application infrastructure when possible
 - Make use of current technology

Haggle architecture



Haggle architecture

- User data
 - ADU's
 - Not isolated from the network
 - Often linked to form compound data
 - Attributes (value-pair)
 - Forwarding

Haggle architecture

ID	12345
Type	Photo
Filename	DSC10027.jpg
Mime-Type	image/jpeg
Creation-date	1/1/2006 17:32
Created-by	James Scott
Security-group	Public
Keywords	Athens, Greece, seashore, sunset
Data	[binary jpeg data]

ID	23456
Type	Message
Mime-type	text/plain
Date	1/1/2006 17:40
Data	"Wish you were here!"

ID	34567
Date	1/1/2006 17:40
From	"James Scott"; james.w.scott@intel.com
To	"Jon Crowcroft"; +447123456789; jon.crowcroft@cl.cam.ac.uk
To	"Pan Hui"; pan.hui@cl.cam.ac.uk; BT 0F:CC:3E:C9:87:21
Priority	5
Claim-ID	12345; 23456
TTL-hops	100
TTL-deadline	2/1/2006 17:40

Haggle architecture

- Protocols and naming
 - User-level naming schemes
 - Address is any name that is available in the Protocols module
 - An ADU can contain mappings to addresses

Haggle architecture

- Neighbors and forwarding
 - Haggle performs neighbor discovery
 - Marks ‘nearby’ nodes depending on the infrastructure
 - Haggle estimates the ‘benefit’ of performing a transfer

Haggle architecture

- Resource management
 - Cost/Benefit analysis
 - Applications cooperate in sharing resources
- Priority levels

Haggle architecture

- Interacting with applications
- Provides a new abstraction layer for mobile applications

Pros of Hagggle

- Interesting for certain scenarios
- Open source
- Implementation available for mobile platforms
- Developing mobile applications is easier

Cons of Hagggle

- Security (tracking, private information, etc.)
- Some aspects are not well defined (forwarding)
- Incentives to cooperate an issue
- Is it a valid idea nowadays?

Questions & Comments?

Thanks for listening!