NETINF
LIVE VIDEO STREAMING
FOR
SKI WORLD CUP FALUN 2015
SEPTEMBER, 2014

BÖRJE OHLMAN (ERICSSON)
ADEEL MOHAMMAD (ERICSSON)
ANDERS LINDGREN (SICS)
Ericsson to deliver an experience “beyond skiing” in Falun2015
VIRTUAL ARENA PLATFORM
FUNCTIONAL VIEW

Visualization

WEB
- Static Data focus
- Static visualization
- Target audience:
  - All personas

Virtual Live arena
- Dynamic Data focus
- Dynamic visualization
- Target audience:
  - Off-site/on-line viewers

WC App
- Dynamic Data focus
- Static visualization
- Target audience:
  - On-site visitors
  - Off-site visitors

3PP apps linked to the live arena
- Personal Data Focus
- Dynamic/Static Visualisation
- Target Audience: Live Arena
  - WeMeMove
  - Traffic, TV ...

Service environment (Service packages)

Ericsson Virtual Arena Platform

Competitio
n and Event

Traffic and Location

Sponsor activation through Wellness

2nd Screen TV

Data integration APIs

Sveriges Radio
Trafikverket
Dala trafik
SWISS TIMING
FIS API
WeMeMove
Other...

Analytics

MONITORING TOOL

billing

DATA SOURCES

NetInf | Uppsala/Falun | 2014-04-03 | Page 3
What problems to solve:

› Give both on-site and off-site viewers an exciting 2nd screen experience while the event is ongoing
  - TV and other digital experiences offered in a gaming like virtual arena client optimized for Tablet/PC

› Ensure minimum delay between 2nd screen and real event to deliver information while it is relevant (especially for on-site)
  - Efraim project

› Minimize the impact on MBB consumer bucket with reversed volume charging
  - Zero rating of data traffic
  - Ad-funded virtual arena content with kick-back to operator for the volume consumed.

› Understanding consumer behaviour and network impact
  - Consumer consumption patterns
  - Network impact analytics
  - Social Media analysis, what events created what kind of reactions and sentiments in the viewership.
Goal with the project:
• Offer a live video streaming Android application for the FIS Nordic World Ski Championships 2015 in Falun as part of the Virtual Arena, see
  • http://falun2015.com
• Live video from any Android client should be distributed using Erlang NetInf routers. For information on NetInf, see
  • http://netinf.org
• Two key objectives are to take an existing prototype and evolve it so that it can:
  • Be distributed to real users via an Android market
  • Reduce the delay in the streaming service to become “near realtime”
HOW THE LIVE STREAMING SERVICE IS USED

Recording a video
• User starts the recording app and selects the publish option. The application will then publish the stream object with location and thumbnail. The user can optionally add human readable name and/or description of object to the metadata of the object.

Playing a streamed video
• User needs to get the name of the desired stream. Possible ways of getting the name includes:
  • Clicking on stream object (e.g. a thumbnail or location on map) in event browser (see next slide)
  • Clicking on name/thumbnail (from regular web page or from the Virtual Arena)
  • Scanning a QR code on a poster/sign or on a screen, e.g. a Jumbotron
EXISTING PROTOTYPES OF EVENT BROWSER

› In previous projects, similar event browsers have been developed
› SAIL Event With Large Crowds Directory
› MOSES opportunistic event sharing service
As an additional feature the virtual arena could potentially connect to the live streaming service as a user of the service (seen from the live streaming service perspective).

The virtual arena can then treat these streams as any other content stream.

This would require that there is a clear interface to the virtual arena (e.g. video format, GPS coordinates, …) which we can deliver our streams to. There should not need to be any modifications done to the virtual arena for this.
In this alternative no modifications are made to the local connectivity network (3G/WiFi). The NetInf clients in the Androids connect via the regular IP network to NetInf nodes in a cloud.

Requirements on the NetInf network
- NRS that can resolve IO name to streaming node locator
- NetInf “DHCP” for connecting NetInf clients to NetInf router or possibly just to the NetInf cloud (could then possibly be hardcoded in clients?)
- Possibility to scale up the number of NetInf routers according to user demand
- Need to auto configure the NetInf network when new nodes come alive
In this alternative an existing WiFi network is enhanced with NetInf functionality. This needs to be done in a way that it does not interfere with any existing WiFi service.

- WiFi access points are combined with NetInf routers. (Either the APs need to be able to connect to two different backhaul networks or the NetInf routers, which are transparent to IP traffic, needs to be powerful enough to not become bottlenecks.)
- The live streams can thus be served in the local WiFi network by nearby users without causing traffic on backhaul links.
- The NetInf clients in the Androids connect directly to these NetInf nodes.

Requirements on the NetInf network
- NRS that can resolve IO name to streaming node locator
- NetInf “DHCP” for connecting NetInf clients to NetInf router
In this alternative a separate NetInf WiFi networks are deployed where there is no existing WiFi service.

- WiFi access points are combined with NetInf routers.
- The live streams can thus be served in the local WiFi network by nearby users without causing traffic on backhaul links.
- The APs will be transparent to regular IP traffic.
- The NetInf clients in the Androids connect directly to these NetInf nodes.

Requirements on the NetInf network

- NRS that can resolve IO name to streaming node locator
- NetInf “DHCP” for connecting NetInf clients to NetInf router
EVENT BROWSER FOR USER GENERATED CONTENT

In addition to official content from the event organizer, users will also generate content (videos/photos/etc) to share with other users.

It is important to have an intuitive and user friendly way for the user to find interesting content in the network to view.

Event Browser
• When a user generates and publishes content, it is tagged with a timestamp and location where it was created
  • (Location can use GPS or other localization technology)
• The Event Browser provides a graphical representation of all content in the system through a map of the event area
• A time slider allows the user to browse through the recent history of the event and see on the map when and where content has been published
• There is also the option to set “triggers” so that the system notifies the user when new content that was published nearby is available

Design choices:
• Show all data in the browser, or only locally available (or differentiate in the view so the user knows expected delay)
• How to share data between users?