automatic testing and measurement of QoE in IPTV

using image and video comparison

Orange Labs / Spain

Antonio González (antonio.gonzalezm@orange-ftgroup.com)

Valencia, April 20th 2009

contents

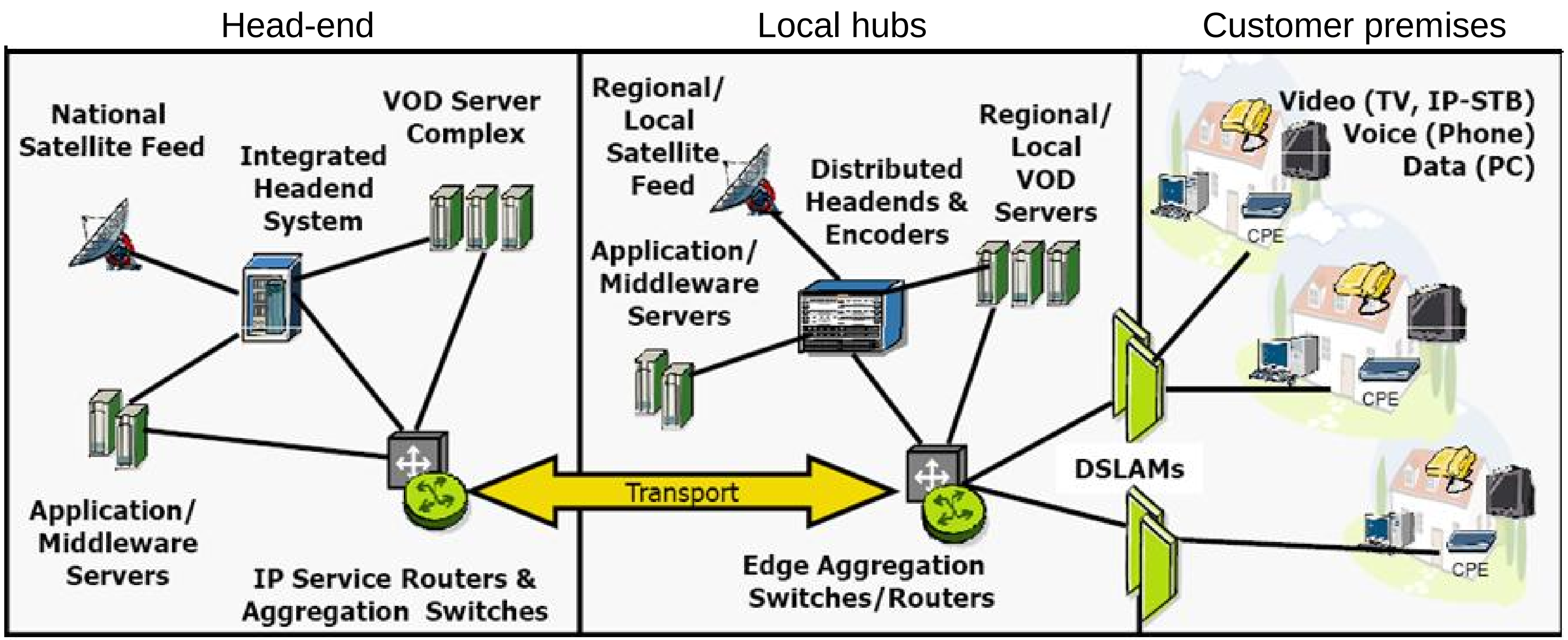
introduction

- Orange iLab Spain, as a major contributor to OrangeTV, has developed a PoC in order to make possible the automatic execution of tests, especially regression, with no human intervention.
- Testing is a critical phase of the development cycle of IPTV-related services, as lengthy and resource-consuming process.
- The developed prototypes open the possibility of an effective measurement of QoE in IPTV scenarios

introduction: OrangeTV

- IPTV (Internet Protocol Television) is the delivery of digital television services using Internet Protocol over a residential network infrastructure (ADSL, FTTH)
- OrangeTV is the Orange IPTV service that includes:
  - Home page: settings, personalization, mosaic
  - Live: free channels, subscription, PPV, EPG, PVR
  - VoD or “24/24 videoclub”: PPV, subscription, series, rewind TV
  - Interactive services
  - Bundles

introduction: architecture of IPTV platform

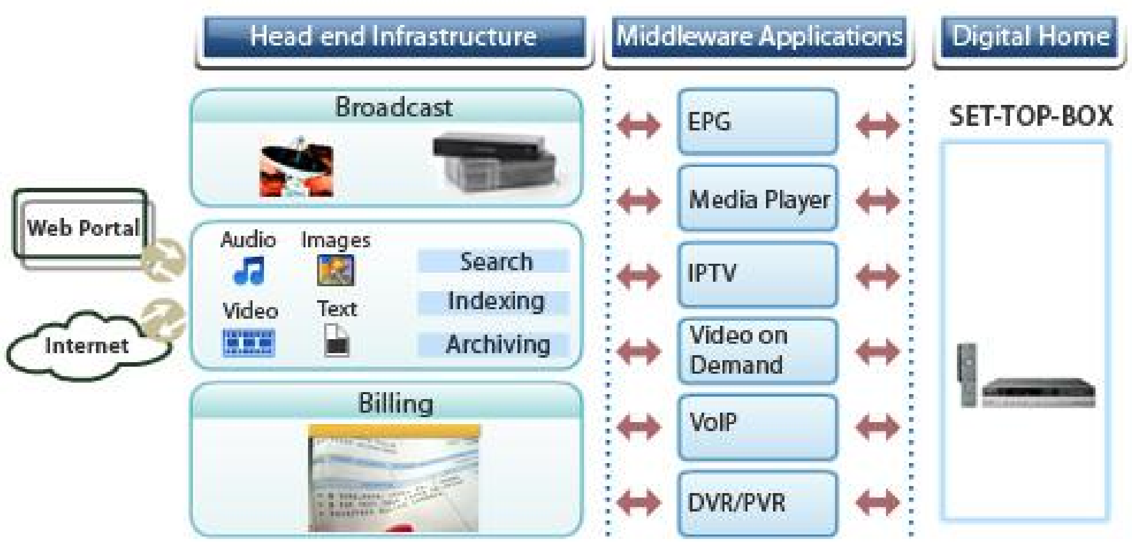


- National content acquisition
- Content encoding, DRM, IP encapsulation
- Multicast services
- Central VoD management
- OSS/BSS integration
- Subscriber management
- Middleware/Content/Application servers

- Regional content acquisition
- Regional content encoding
- DRM, IP encapsulation
- Unicast services, distributed VoD servers
- Access control
- Distributed Middleware/Content/ Application servers

- Set-top box: IPTV client, DRM, video decoder
- Subscriber identification
- Settings and personalization
- Premise storage, PVR
- Other triple-play services

introduction: architecture of IPTV applications



- The STB has a built-in HTML browser and a MPEG-4 player
- For content services, such as the VoD portal or interactive services, the STB behaves like a web client.
- HTML pages sent to the STB are dynamic, running Javascript.
- Client-server interactions are Ajax-like

contents

test automation: the testing process

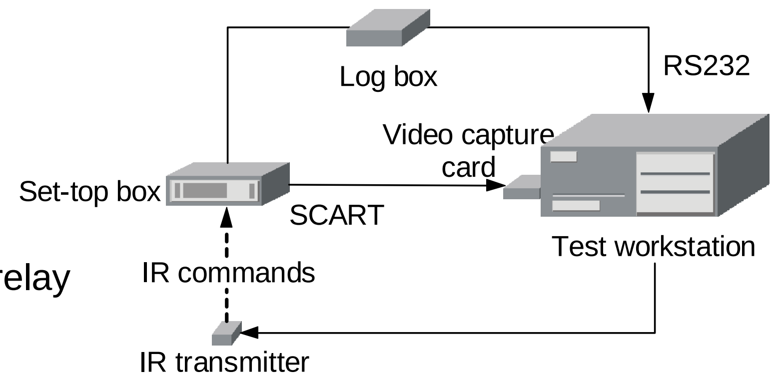
- The service evolution cycle:

Analysis → Specification → Development → Testing → Rollout

- Tests are manual: a human tester follows a test plan document, by commanding the STB via IR as indicated, and checking that the resulting screens are correct
- Test phases:
  - Development and assembly tests
  - Integration testing
  - Smoke tests
  - Qualification testing: functional, non-regression and operational
  - Performance, stress, endurance, an non-funcional testing
  - Acceptance of new STB and firmware
- How to automate the testing?

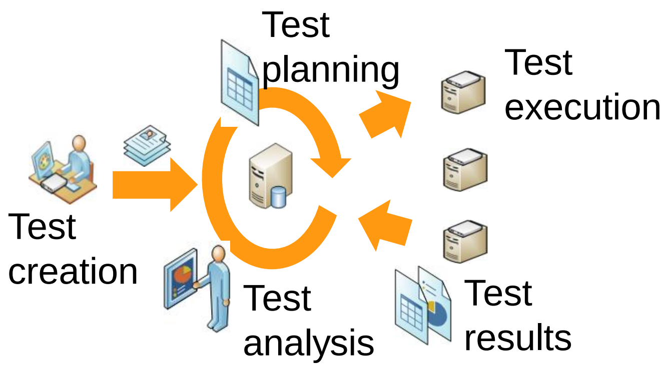
test automation: technical decissions

- Design constraints:
  - Black box. The test subject is the real STBs. No use of emulators or simulators.
  - HTML interpretation and video decoding done by the real STB
  - Tests methods non-intrusive, not requiring modifications to the application, nor scripting
  - The system should not rely on analysis of sniffed logs or network traffic. Most of the screen transitions are made inside the STB, with no traffic generation at all
  - No need for calibration, since calibration is a source of indetermination. Tests should result in an absolute Pass or Fail
- Solution: to simulate the human tester himself, keeping everything else on the loop
  - An execution engine simulates the navigation through the service
  - IR commands sent through a IR transmitter
  - STB SCART directed to a video capture card
  - Logs are taken from the STB to ACK IR commands
  - A/C power and Ethernet flow managed via a relay
- The resulting screen transitions are compared against previously recorded ones. An image comparison algorithm has been developed for this purpose.

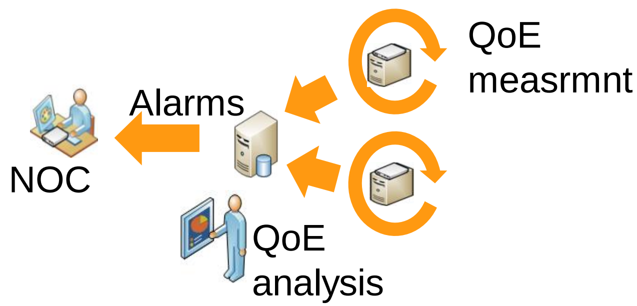


test automation: use cases

- Automatic regression testing:
  - Functional tests are defined and stored at a central server
  - Definition means the translation from a test document into a set of IR navigation commands and screen matches
  - Using a planning tool, tests are scheduled and assigned to the available probe automatons
  - The automatons execute the tests automatically, and generate the results back to the central server
  - The results are analyzed. Failed tests can be repeated, or sent to the defect/bug-tracking tools

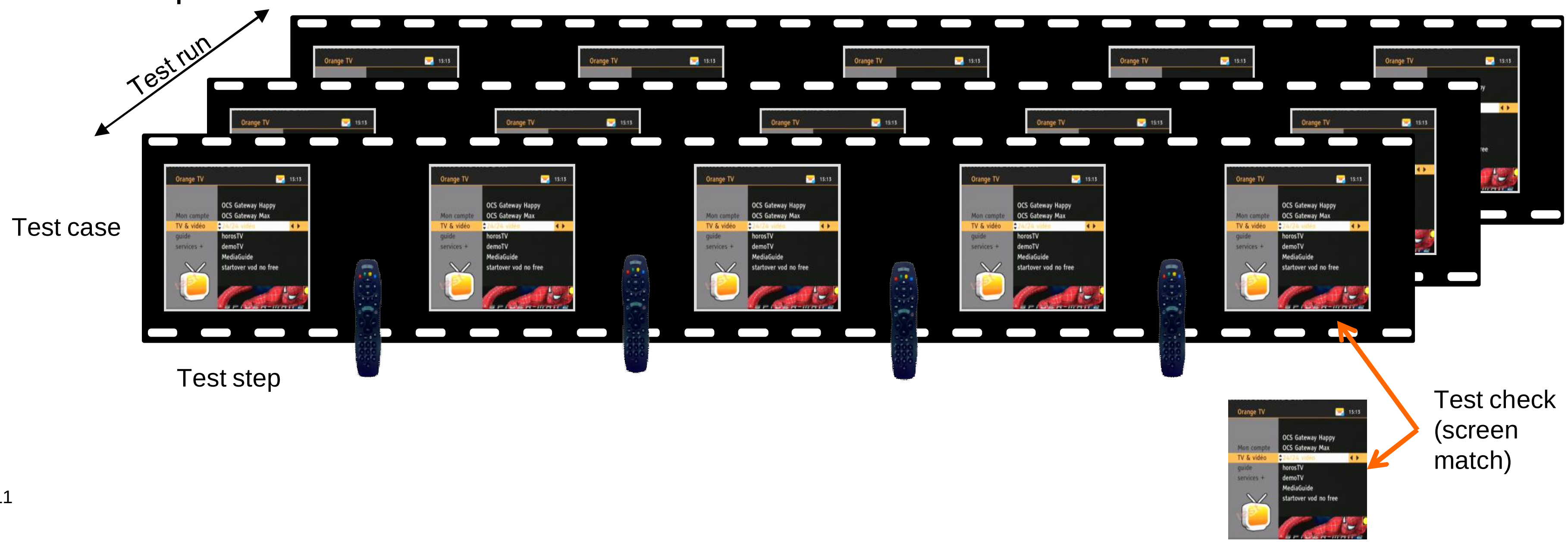


- QoE measurement
  - Monitoring tests are downloaded to the QoE probes, distributed geographically along the access network
  - The probes execute the actions in continuous loop, and report back the results to the central server
  - Reporting tools evaluate the KQIs
  - Repeated test failures can generate service alarms to the NOC



test automation: definition of test cases

- The following concepts are introduced:
  - A test run is a collection of test cases
  - A test case is a collection of steps in a test flow
  - A test step is the smallest action in a test case, and spans from an IR interaction until a change in the screen happens
  - A test flow can include loops and conditional branches
  - A check is a test step which includes at least one comparison and one exit criteria
  - Test steps can be grouped in test navigations, by means of a macro language of orders
- Example:



test automation: definition of test cases

- The test definition process (i.e. the translation from a test plan document into a computer representation) is assisted: no scripting language is required
- Process:
  - The tester executes the individual test plans in one of the Test workstations, interacting with the STB through the IR transmitter and video card, which are displayed on the screen
  - Since the workstation is receiving logs from the STB, the specific template being used is known. Associated with the template the system shows a suggestion of which is the more likely area for image comparison
  - The user can override this suggestion, defining other comparison areas at will
  - After some navigation, the user defines a “step”, which is the smallest test element. A step must include at least one comparison and one exit criteria
- Macros are a set of commands which position the STB at a predefined initial condition, as “shortcuts”, without making any comparison

STEP ID = 1

REMOTE COMMANDS ({COMMAND,SLEEP})

IR command = { move_down, 2000}
IR command = { ok, 3000}
IR command = { vod, 15000}
IR command = { move_down, 1500}
IR command = { move_down, 1500}
IR command = { move_down, 1500}
IR command = { move_down, 1500}
IR command = { move_down, 1500}
IR command = { move_down, 0}

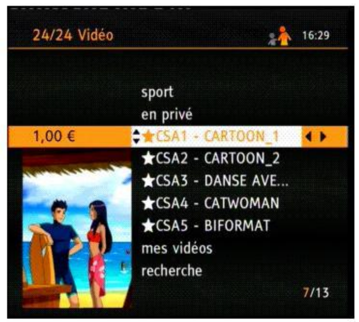
REGIONS

Region = 0
Rectangle: [x=51,y=29,width=551,height=41]
Precision: PRECISION_DEFAULT
Threshold: 0.6

LocalThreshold: 20.0

COMPARATION 1 ====> Next State: 2

COMPARATIONS



test automation: use of application logs

- Application logs are taken from the STB RS232 port. These logs are used:
  - To identify the screen template at test definition time, in order to suggest the most likely comparison area
  - To find acknowledges for the IR keys that have been sent, providing for key losses, at test execution time
- The tests confirmed the issue of spurious IR losses:
  - The miss rate is approximately 0.4% of the issued infrared commands.

```
[9:43:33.269][BROWSER][LOG] writeJSLoader::<script language="javascript" src="/SED/VOD/js/list.js"></script>
[9:56:44.806][BROWSER][LOG] writeJSLoader::<script language="javascript" src="/SED/VOD/js/content.js"></script>
……
[9:46:23.873][BROWSER][LOG] CSF SDK Trace -> RemoteControl('DOWN') Begin - CIS VOD
[9:46:23.874][BROWSER][LOG] RemoteControl::code = DOWN
[9:46:23.923][BROWSER][LOG] CSF SDK Trace -> RemoteControl('DOWN') End - CIS VOD
```

Template: Video presentation screen

ACK od Key DOWN

Suggested areas to check for this tempate screen

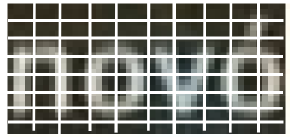
test automation: image comparison algorithm

- Design objectives:
  - To detect punctuation marks
  - To manage semi-transparent backgrounds

N

M

I



J

- Each original and captured screens are divided into NxM regions, each one IxJ pixels. For each region, a mean (R,G,B) vector is obtained

$$\overrightarrow{RGB}_{nm} = \left(\frac{\sum R_{ij}}{I \times J}, \frac{\sum G_{ij}}{I \times J}, \frac{\sum B_{ij}}{I \times J} \right)$$

- A Fuzz factor ff (max allowed difference) is used to filter out noise (transparency) effects. If $\left| R_{ij\,org} - R_{ij\,cap} \right| \leq ff$, then the original values are used
- The RGB distance between each captured and original regions is obtained

$$d_{nm} = \sqrt{(R_{cap} - R_{org})^2 + (G_{cap} - G_{org})^2 + (B_{cap} - B_{org})^2}$$

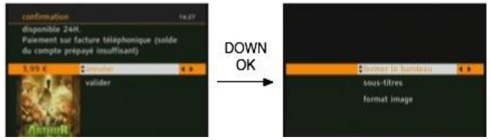
- A Local threshold LTh is defined as max allowed RGB distance between regions. If $d_{nm} \geq LTh$, the comparison is set as no-match.
- The global RGB distance between the full captured and original screens is set at

$$d = \frac{\sum d_{nm}}{N \times M}$$

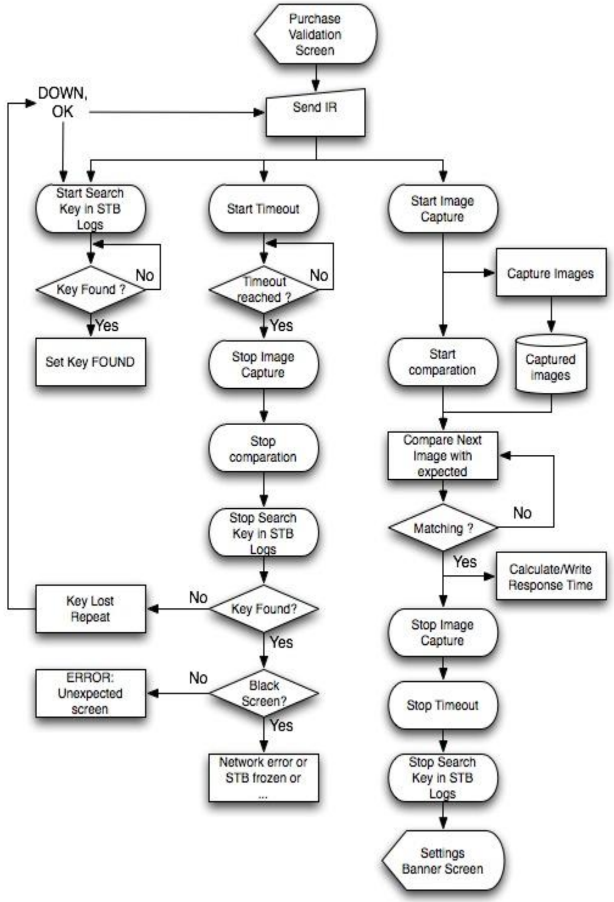
- A Global threshold GTh is used in order to determine if both images match. If d $\geq$ GTh, the comparison is set as no-match

test automation: test match logic

- Design objectives:
  - ✓ To de-couple video recording (synchronous) from image comparison (asynchronous)
  - ✓ To detect and measure response times in the threshold of 100 ms.
  - ✓ To recover from spurious IR transmission errors



- After sending IR commands, 4 processes are launched in parallel:
  1. Timeout: global timeout
  2. Grabber: it captures the video stream, as a sequence of frames, at the maximum available speed
  3. Key finder: it checks STB logs, in order to find acknowledges for the IR keys
  4. Checker: it compares each captured frame against the reference region
- Event control logic:
  - ✓ If the sent keys are acknowledged by the STB logs, the KeyFinder stops
  - ✓ If a match is found, the Checker process writes the log entry with the timestamp, saves the captured video stream, all processes are stopped, and the flow continues with the next step
  - ✓ If the global timeout is reached, and the sent keys are not yet found, it is assumed that they are lost, therefore the IR command is repeated and all processes are restarted
  - ✓ If the global timeout is reached, the sent keys are ack'ed, but no image match has been found, it is assumed a test failure



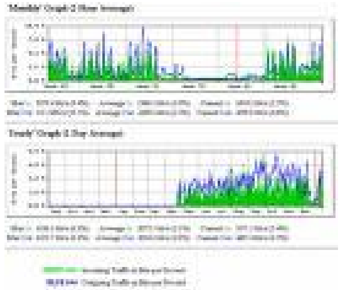
test automation: results and conclusions

- We have demonstrated an effective way to reduce development time and costs:
  - Testing is non-intrusive. Both STB and application are black boxes.
  - Tests results are reliable, and uncertainty reduced to a minimum
  - The test definition process is easy and assisted
  - Automation extends to the whole testing process. For instance, failed tests collect all necessary information for defining a bug.
- In the PoC we found 2 issues to be further worked on:
  - Spurious IR losses require a double-check taken from STB logs, but these logs are not available in production environments
  - We have not found an easy way to detect video start events. Some alternatives are:
    - To provision ad-hoc VoDs which have controlled frames, based on pixels, colors or numbers
    - To sniff the Ethernet flow directed towards the STB and to analyze the protocol

Our test automatons can be used as monitors to measure the quality of the service

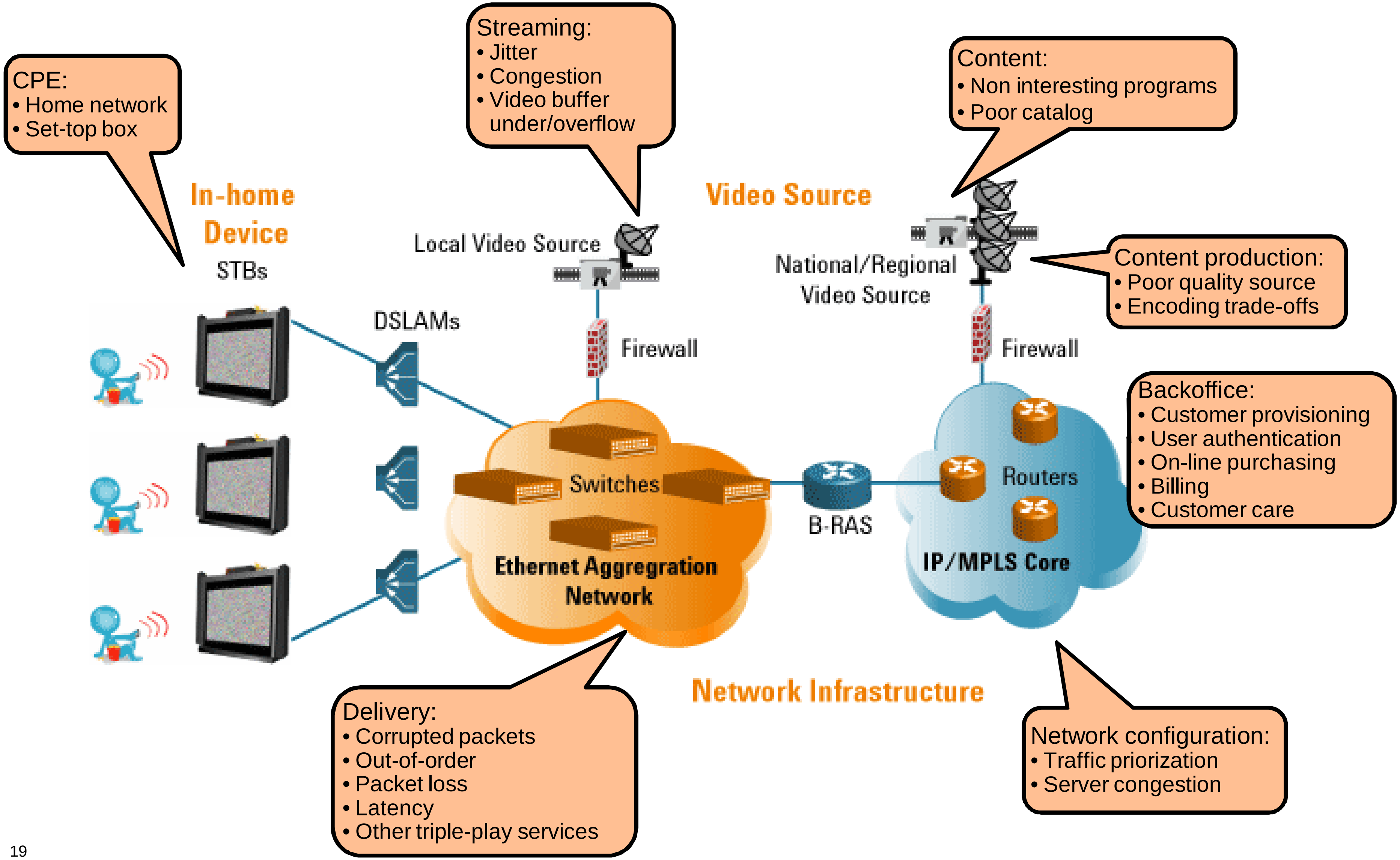
contents

IPTV QoE: concepts



- The operator perspective: QoS
  - QoS is the measurement of the difference between optimal and current delivery status (network, servers)
  - Measurements are based on networking parameters such as packet loss, latency…
  - QoS is averaged.
  - It stays at the network layer, and spans the access network only.
  - *"Quantitative and qualitative characteristics of a distributed multimedia system, which are necessary in order to achieve the required functionality of an application"*
- The customer perspective: QoE
  - QoE is the assessment of the level of customer satisfaction
  - QoE measurements are both subjective and objective
  - QoE is individual: each one customer matters
  - QoE is end-to-end, and stays at the application layer
  - *"The overall acceptability of an application or service, as perceived subjectively by the end-user"*
- QoE effects are caused by QoS problems. But we don't know very well the cause-effect relationship. Traditional measurement of quality is based on QoS.

IPTV QoE: the IPTV quality chain



IPTV QoE: artifacts and impairments

- Packet loss: missing blocks within the decoded image, causing "blockiness“, and frame freeze if a large proportion of blocks are missing
- Jitter: caused by variability in time of latency, or non-uniform packet arrivals
- Blockiness: result of Low bit rate coding and coarse quantization
- Blurriness: reduced sharpness of edges and spatial detail, because of compression to reduce resolution
- Jerkiness: Motion that was originally smooth and continuous is perceived as a series of distinct 'snapshots‘
- Audio impairments: noise, lip synch errors, …

Packet loss

Jitter

Out of order

Blockiness

Blurriness

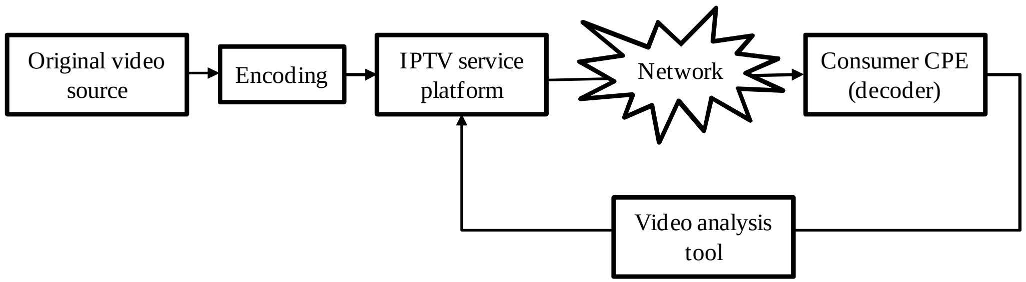
IPTV QoE: measurement

- QoE challenges :
  - To quantify the customer satisfaction: How responsive is the remote control? How fast is zapping? Are audio and video quality good? Always?
  - To measure what degrades customer satisfaction
- QoE measurement methods:
  - Subjective centric:
    - MOS (Mean opinion Score): subjective measurement indication, taken from a panel of participants, used to assess and rank the video quality
  - Objective centric:
    - PSNR (Peak Signal to Noise Ratio): root mean square value of the differences (errors) of the original and the received video frames
    - MQPM (Moving Pictures Quality Metric): it replicates the experience of a human observer and rates the IPTV stream on a scale from 1 to 5.
    - MDI (Media Delivery Index): scoring mechanism that indicates video quality levels and also identifies the originating network components. This is achieved by measuring instances of jitter levels and packet loss. The MDI metric is displayed as two values: a Delay Factor (DF) and a Media Loss Rate (MLR).

contents

QoE measuring method: our approach

- Most approaches to IPTV Quality Assurance focus on familiar subject: the network
  - Have all packets arrived across network? In order? With delays or jitter?
- QoE is usually inferred from QoS.
  - **Assuming network health is perfect, is quality to customers assured?**
- Following our black box approach for testing, we suggest to detect or measure the difference between what is delivered by the servers an what is seen at the customers' TV screen
- We propose to use image and video comparison: capturing the output from the STB, and comparing it against the provisioned reference



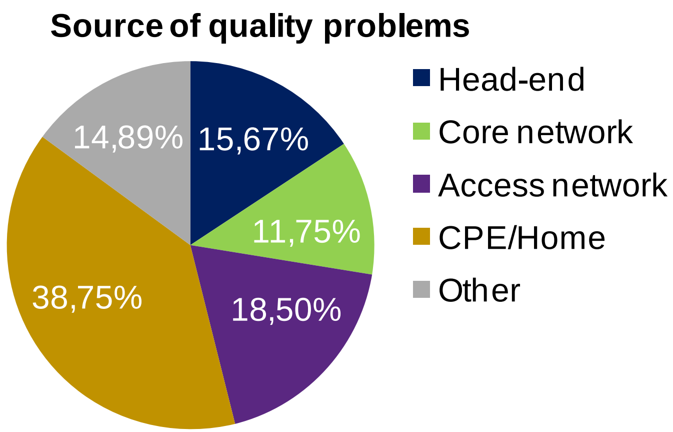
- Thus, we would filter out conditions that do not take place at the delivery moment, such as content production, encoding artifacts, etc

QoE measuring method: video quality analysis

- 3 types of models:
  - Full Reference (FR): this model samples the video signal near the network source and the STB. The samples are compared, and the result is measured in a range from 1 to 5. The Full Reference model involves a large file capture, and is generally considered to be the most accurate method.
  - Partial Reference (PR): this model also takes a sample near the source and destination, but a smaller reference sample is required. Again, results are measured in a range of 1 to 5. This model is considered to be at least as accurate as the full reference model.
  - Zero Reference (ZR): this model uses only the video stream at the destination. And since the model does not require any transfer, alignment or comparison, the calculations can be performed in near real time. However, because the ZR model does not compare source video data, it is not considered to be as accurate as the first two models.

QoE measuring method: indicators

- Key Quality Indicators:
  - Navigation response time
  - Service availability
  - Response time to transactions
    - Channel zapping
    - Trick latency
  - Audio/video quality (live/VoD):
    - Video Impairment Detection
    - Blackout
    - Freeze frame
    - Audio/video synchronization, lip synch
  - STB boot time



QoE measuring method: navigation response time

- The probes necessary for this use case are the same already developed for the proof of concept
- The test case consists in a flow of IR commands and the comparison of the captured screen output against the reference.
- The probes are constantly repeating the same tests in continuous loop, with a wait time between them.
- Raw data generated as logs are transformed into aggregated KQI, such as the average response time to the defined navigation steps

QoE measuring method: service availability

- Similar to the previous use case
- Examples of availability checks:
  - Transition from Live to VoD and backwards
  - EPG visualization
  - VoD search query
  - VoD playback start
- Raw data generated as logs by the probes are transformed into aggregated KQI:
  - Availability, as the 1's complement of the percentage of steps that generated a "STB not responding" or a "Service unavailable" condition

QoE measuring method: zapping time

- On an IPTV network, the process of channel changing takes place on a server instead of on the set-top box.
- Since the set-top box interacts with the network during a channel change, the viewing experience may be affected by delays from the home equipment, from the network and from the data center.
- The automatic test probes can be configured for executing a continuous process of zapping through a carrousel of predefined channels. There is no need for video comparison in this case, since the small corner logo present on most of the channels can be used instead as image comparison.

QoE measuring method: application transactions

- Transaction measurement deal with the monitoring of actions originated in the STB that end in the execution of a business function in the middleware of the IPTV service platform.
- Examples of transactions are:
  - introduction and recognition of PIN code or adult code
  - purchasing of a VoD
  - verification of the available credit
  - access platform messages
- In most cases, only the image comparison is needed.
- If a test step ends in a modification of a parameter in the customers database, the corresponding rollback transaction may be needed in order to restore the test case to its original state.

QoE measuring method: audio/video quality (VoD)

- VoD:
  - We look for differences in the video output generated by the STB and the original file delivered by the IPTV service platform.
  - FR video comparison algorithm
  - The original files can be previously downloaded to the probe and streamed to the STB.
  - Specific samples of videos may be prepared for this purpose, available to the STB as VoDs.
  - The recording of the VoD is made in real time, but the comparisons can be made in batch.
- Comparison algorithm able to detect the response time to the start of the video playback, as well as impairments such as:
  - pixelized video (tiling)
  - skewed video
  - random bars across the image
  - unsynchronized audio, lip synch errors
  - loss of audio with video
  - noise, degradation of audio
  - frozen video frame
  - gaps in video image
  - no video at all

QoE measuring method: audio/video quality (live)

- Live:
  - Most demanding use case, because of the on-line nature of a live broadcast
  - The probes record programs of particular channels, and analyze the recordings using a zero-reference algorithm.
  - The target is to produce the results in real-time.
  - In very specific cases, such as the transmission in PPV of an important event, it could be arranged as the comparison of a VoD in batch
- Similar measurements than for VoD

QoE measuring method: STB boot time

- The power-on cycle of the STB represents a significant chain of transactions; the longer they take, the less pleased becomes the customer.
- Power-on transactions performed by the STB:
  - OS start-up
  - start the IP stack and connection
  - get an IP address from the DHCP servers
  - receive SNMP traps signaling its configuration
  - check the firmware revision
  - load the stored user's settings
  - load the application framework and client middleware
  - check the subscriber's rights against the platform
  - receive the home page of the service, typically consisting on a mosaic of channels.
- The probes are commanded to reboot, acting on the power relay, and the image comparison algorithm seeks for the welcome page.

QoE measuring method: types

Indicator	Check method
Availability	Image comparison
Response time	Image comparison
Channel zapping	Image comparison
Transactions	Image comparison
Audio/video quality (VoD)	FR Video comparison
Audio/video (live)	ZR Video analysis
STB boot time	Image comparison

summary

- IPTV services represent a challenge for quality assurance, not only because they need a complex infrastructure, but also because the user satisfaction includes a subjective component.
- In our work we have reached two main conclusions:
  - IPTV interactive services can be tested and monitored the same way it has been done with web for a long time, using robot-like probes that reproduce the users' actions.
  - The quality impairments that affect QoE in stream broadcasts and video on demand can be objectively assessed and measured. However, we need further work on producing the most efficient video analysis and comparison algorithms.

thanks - gracias - merci