Achieving Viewing Time Scalability in Mobile Video Streaming Using Scalable Video Coding

Cheng-Hsin Hsu

Senior Research Scientist

Deutsche Telekom R&D Lab USA

Los Altos, CA

Joint Work with Mohamed Hefeeda

February 22, 2010

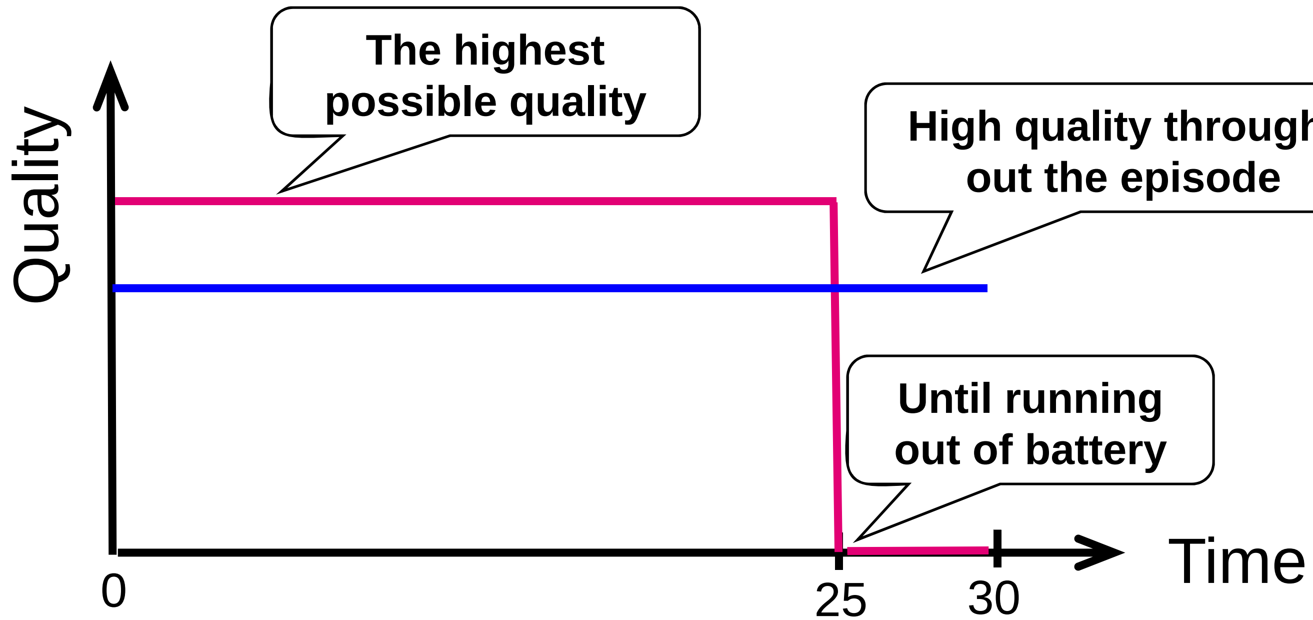
Mobile Video Streaming

- Recent studies show that many users watch videos using their mobile devices even when stationary devices, such as TVs, are available [Vorbau *et al*., HotMobile'07]

- Mobile devices have strict power constraint
  - Battery technologies do not grow as fast as CPU speed, memory size, and disk capacity

Motivation

- Mobile users must consider **battery lifetme** as a new dimension of viewing quality
- For example: a user watches a 30-min episode



Streaming at a *lower* quality, however, requires transcoding, which is computationally intensive

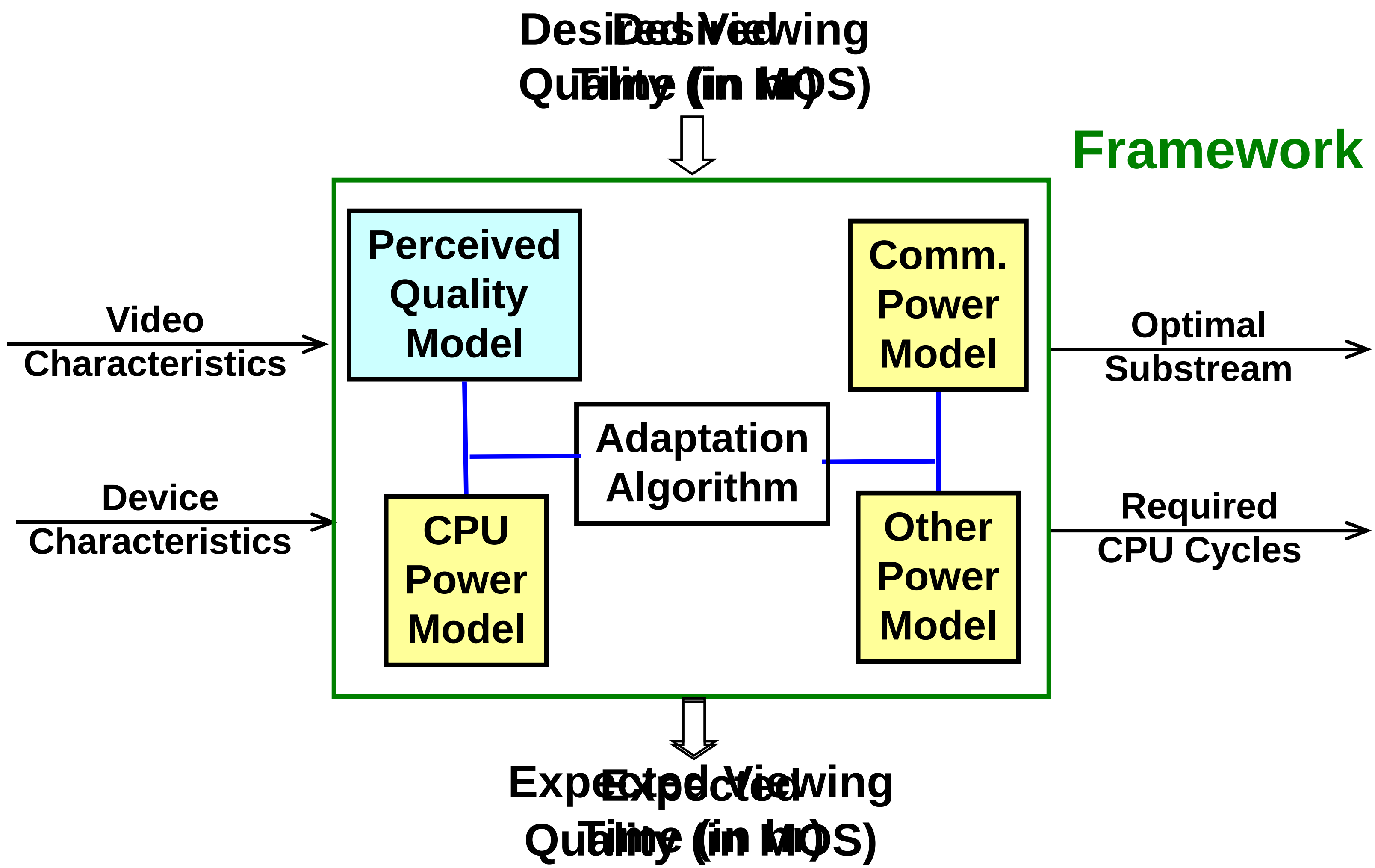
Scalable Video Coding (SVC)

- Multiple substreams can be efficiently extracted from a scalable stream for support different resolutions, frame rates, and fidelity levels
- Often used for devices with heterogeneous resources
  - Such as bandwidth, display resolution, and decoder capability
- We highlight another benefit of SVC: to enable **viewing time scalability**

Problem Statement

- **Goal**: allow users to choose desired video quality and viewing time of a streaming video
- We first develop a **framework** to predict battery lifetime and perceived quality of individual substreams extracted from a scalable stream
- We then propose an algorithm and provide users a **control knob** to trade off perceived quality and viewing time

Quality-Power Adaptation Framework



Power Model

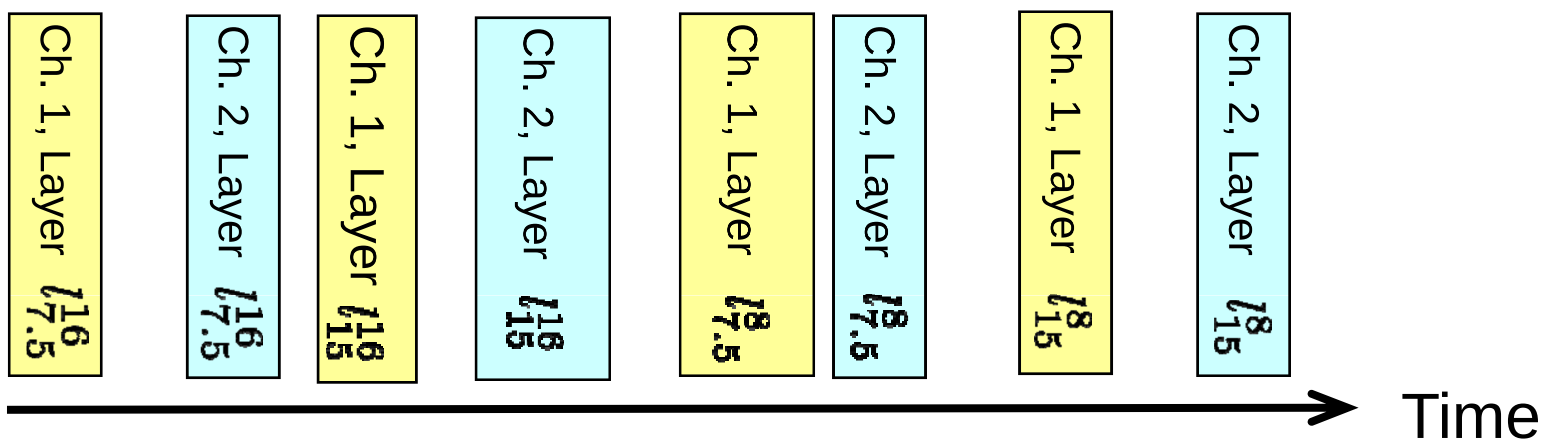
- **Divide power consumption into**
  - **CPU: for video decoding, written as $p_c(y)$, where y is number of cycles per second**
  - **Comm.: for receiving video stream, written as $p_n(\gamma)$, where γ is the sleep period ratio**
  - **Background: accounts for others, such as display backlight, written as a constant p_b**

Communication Power Model

- Depends on types of wireless networks
  - Such as WLANs, WiMAX, and 3G Cellular
- We focus on mobile TV networks and propose a broadcast scheme, **FMVB**, for scalable streams
- **FMVB** (Flexible Mobile Video Broadcast) supports substream extractions at mobile receivers
  - Extractions must be done at receivers because of the broadcast nature
  - Enables both temporal (specified by frame rate *t*) and fidelity (specified by quan. step δ) scalability

Communication Power Model (cont.)

- FMVB determines time and size of transmission bursts for individual layers
  - Formulas are in the paper



- *[Theorem]* FMVB produces feasible schemes with bursts that are long enough to sustain smooth playout

Communication Power Model (cont.)

- *[Theorem]* The sleep period ratio γ of mobile devices receiving the substream at frame rate t and quan. step δ is:

$$\gamma_t^\delta = 1 - \frac{\sum_{l_i^j \in \boldsymbol{d}_t^\delta} r_i^j}{rS} - \frac{\left|\boldsymbol{d}_t^\delta\right| T_o r_{t_{\max}}^{\delta_{\min}} R}{brS}$$

- We write $p_n(\gamma) = \lambda(1-\gamma)$, where λ mW is the power consumption of the network chip

Our Comm. Model

CPU Power Model

- Complexity model: estimate $y(\delta, t)$, which is the number of cycles required by decoding a substream at frame rate t and quan. step δ

$$y(\delta,t) = \frac{t_{\max}}{G} M\{\theta Y_I + (1-\theta)Y_P + (t/t_{\min} - 1)Y_B + t/t_{\min}\log_2(\delta_{\max}/\delta)Y_Q\}$$

[Ma and Wang, ICASSP'08].

- We can estimate how many cycles needed for each substream, and then?

CPU Power Model (cont.)

- Modern CPUs can run at a lower frequency to save energy
- We match produce (CPU frequency) with demand (number of cycles)
- We adopt the energy scaling model:

$$p_c(y) = \psi y^3,$$ [He *et al*., CSVT'05].

CPU power efficiency factor

- Combining $p_c(y)$ and $y(\delta, t)$ functions gives our CPU power model

Total Power Consumption

- **Total Power Consumption**

Total **CPU** **Comm.** **Bg.**

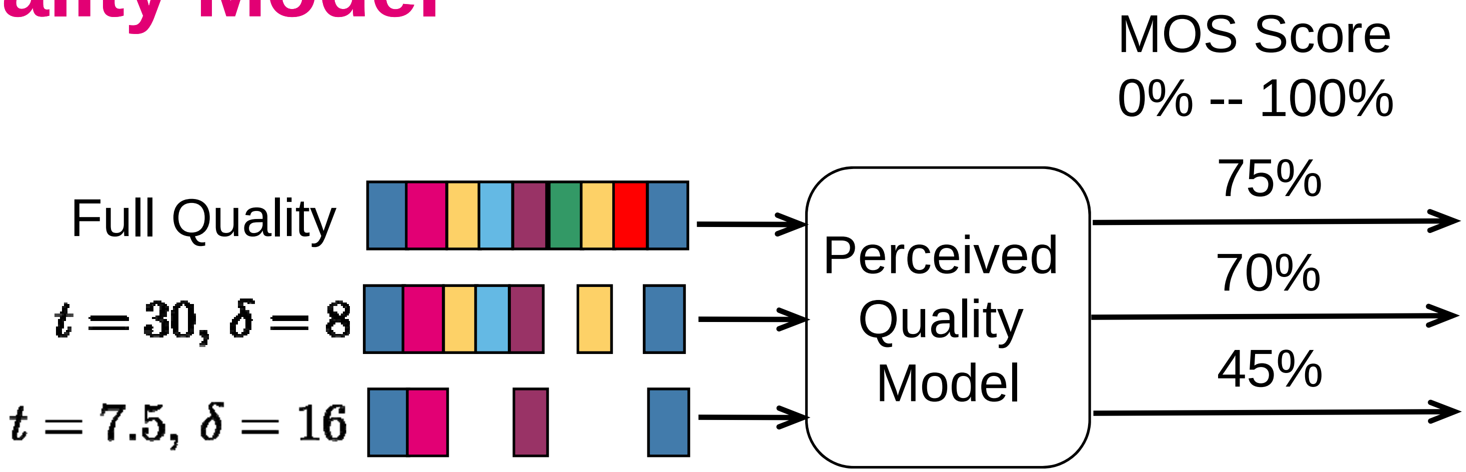
$$p(y,\gamma) = p_c(y) + p_n(\gamma) + p_b$$

Number of cycles per second

Interface sleep period ratio

- **Battery capacity ω is often measured in mAh using built-in circuits**
  - **Viewing time l can be written as $l = \dfrac{\omega v}{p(y,\gamma)}$, where v is the battery voltage**

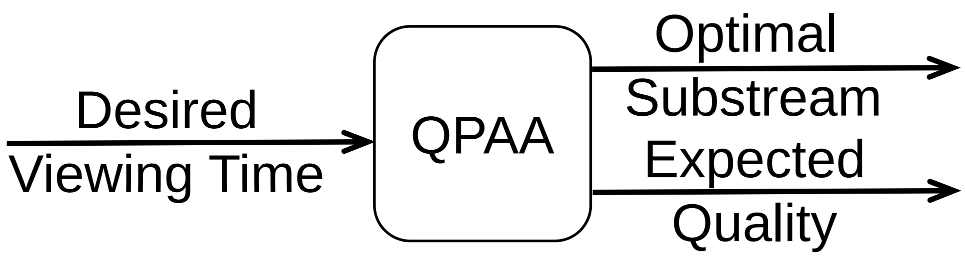
Quality Model

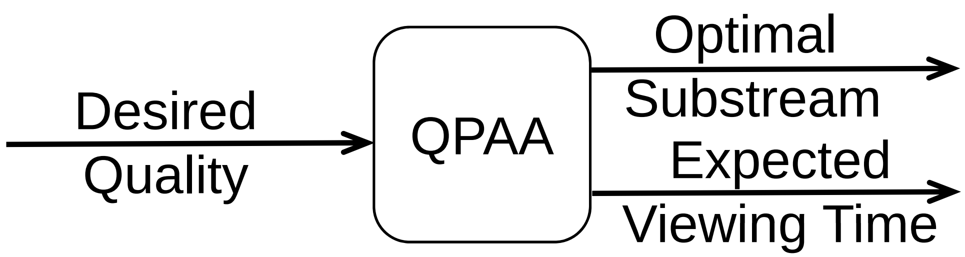


- The perceived quality in MOS is given as:

$$q(\delta, t) = q_{\max} \frac{e^{\alpha \frac{\delta}{\delta_{\min}}}}{e^{\alpha}} \frac{1 - e^{\beta \frac{t}{t_{\max}}}}{1 - e^{\beta}},$$ [Wang *et al*., PV'09].

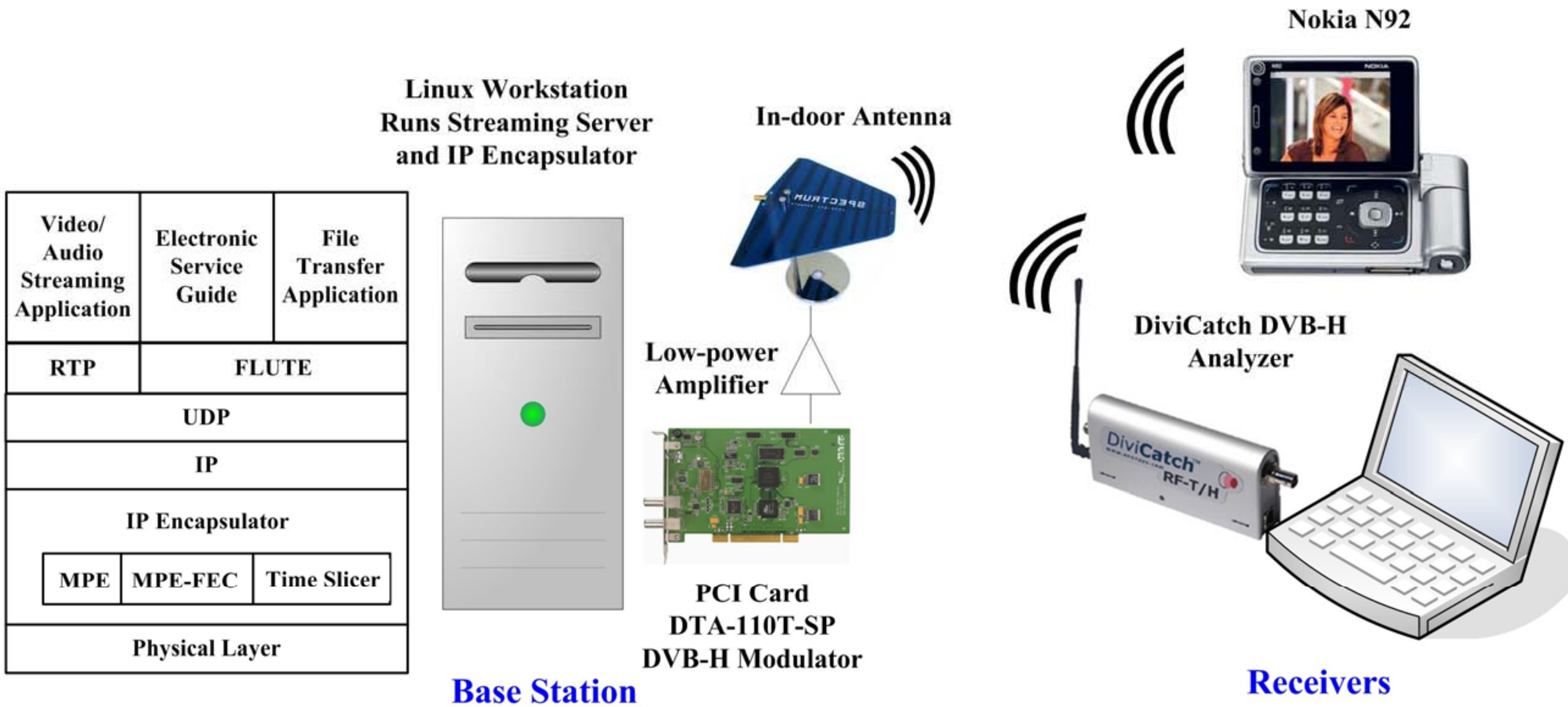
Quality-Power Adaptation Algorithms





- Leverages on the models to compute expected viewing time (or quality) for each substream
- Returns the **best** substream while satisfying users' requirements

Energy Saving on Mobile TV Receiver



- Use off-the-shelf hardware
- Build software in the base station
- Transmits DVB-H compliant signals [MM08'Demo]

Device Dependent Parameter Estimation

- Four model parameters are device dependent
  - λ : network interface power consumption
  - T_o : network interface transition time
  - p_b : background power consumption
  - ψ : CPU power efficiency factor
- We demonstrate how to derive the parameters using a Nokia N96 phone

Network Interface Power Consumption and Transition Time

- We encode a 10-min news into a stream at video rate 450 kbps and audio rate 32 kbps
- We configure our base station to broadcast this stream with different inter-burst periods: 250, 500, 1000, 2000, and 3000 msec
- We use an phone to watch each broadcast for 3.5 hours, and we measure power consumption 4 times each second
- We use the measurements to derive $\lambda = 325$ mW and $T_o = 166$ msec

Background Power Consumption and CPU Efficiency Factor

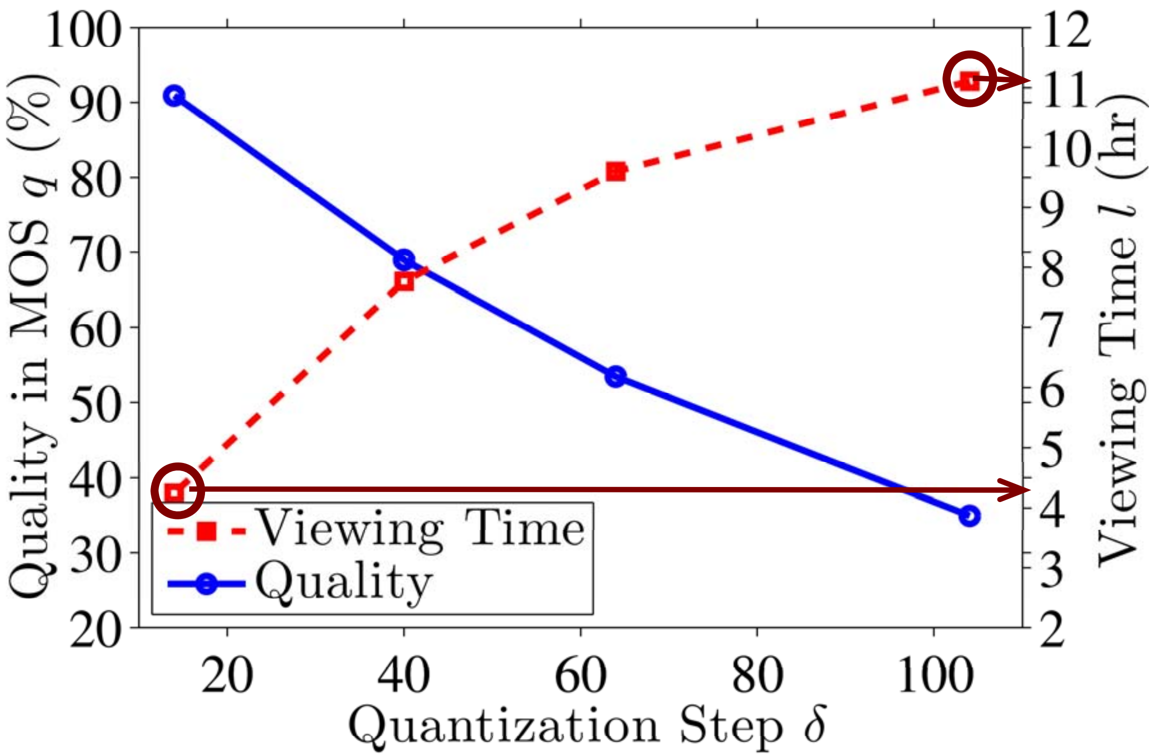
- To measure background power consumption, we configure an N96 phone to display pictures in slide show mode for 1.5 hr
  - Isolate comm. and CPU power consumption
  - Average bg. power consumption is p_b = 290.38 mW

- With all other model parameters, we can compute the CPU power consumption, and thus average CPU Efficiency Factor: $\psi = 3.1 \times 10^{-27}$

Tradeoff between Video Quality and Power Consumption

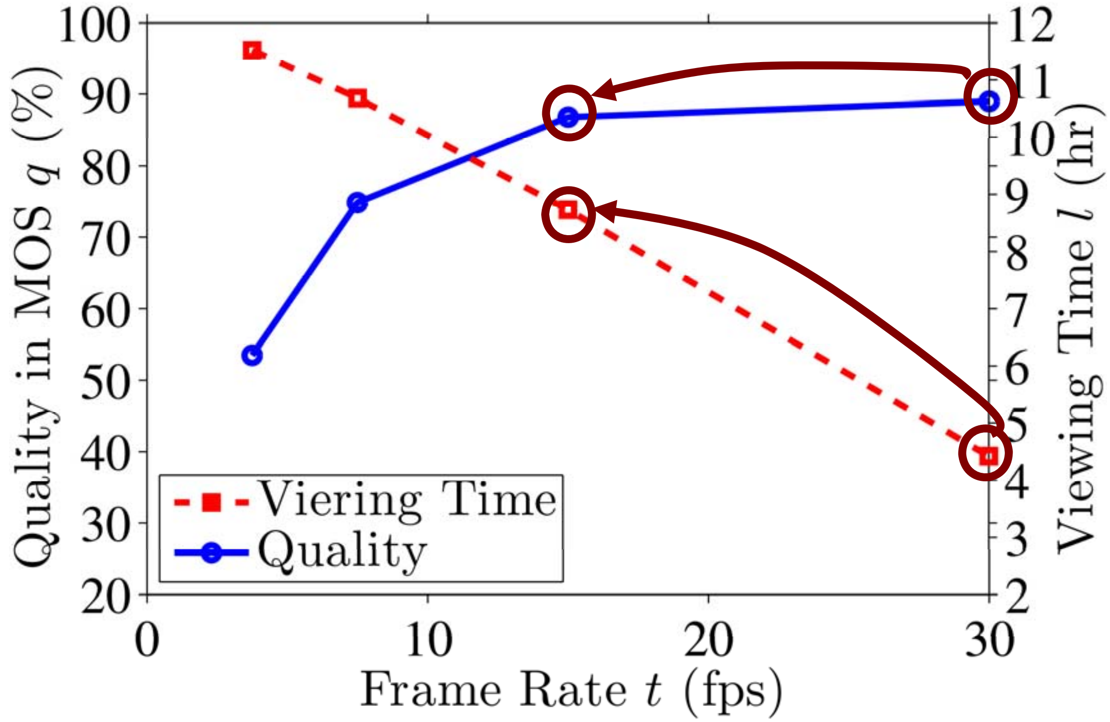
- Use an N96 phone with 950 mAh, 3.7 V battery
- Configure base station to broadcast at 8.289 Mbps
- Consider Crew sequence with frame rates 3.75, 7.5, 15, and 30 fps and quan. steps 16, 40, 64, and 104
- Three broadcast services using the proposed FMVB scheme
  - Temporal only, quality only, and combined

Sample Results



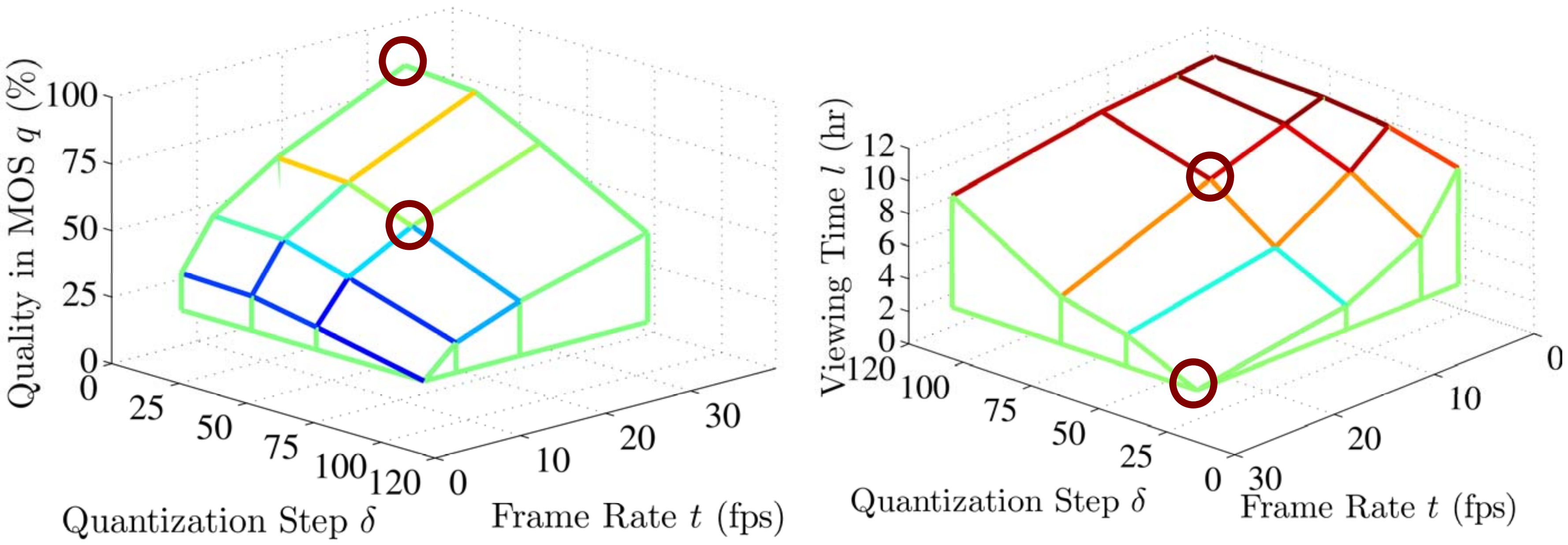
- Wide range of viewing time scalability is possible: 4.3 to 11.1 hrs

Sample Results (cont.)



- Quality degradation is not linear to viewing time increase
  - Frame rate 30 → 15 doubles viewing time, but only reduces quality by 5%
- Highlights the importance of our framework

Sample Results (cont.)



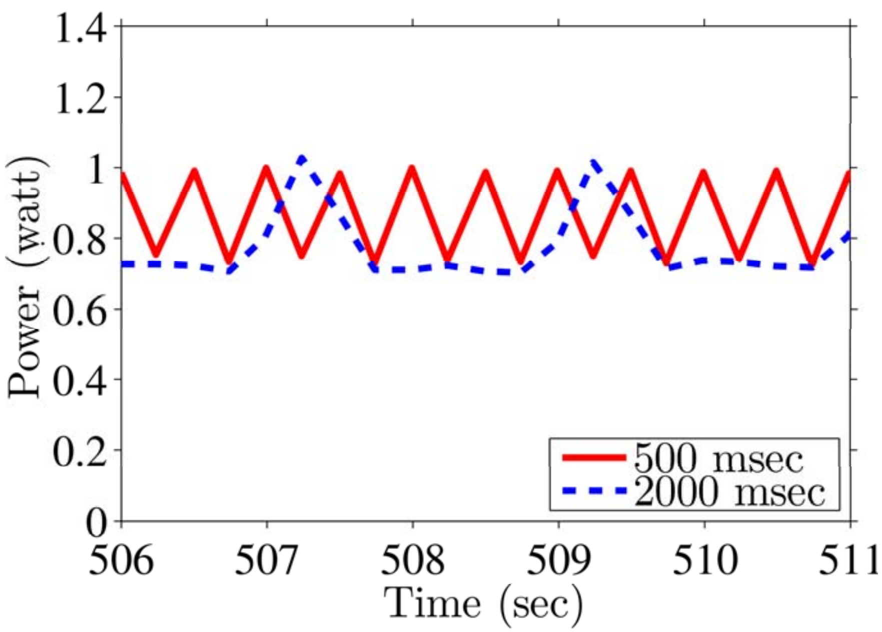
- Combined scalability: each point represents a substream
- Our framework helps users to pick the best substream

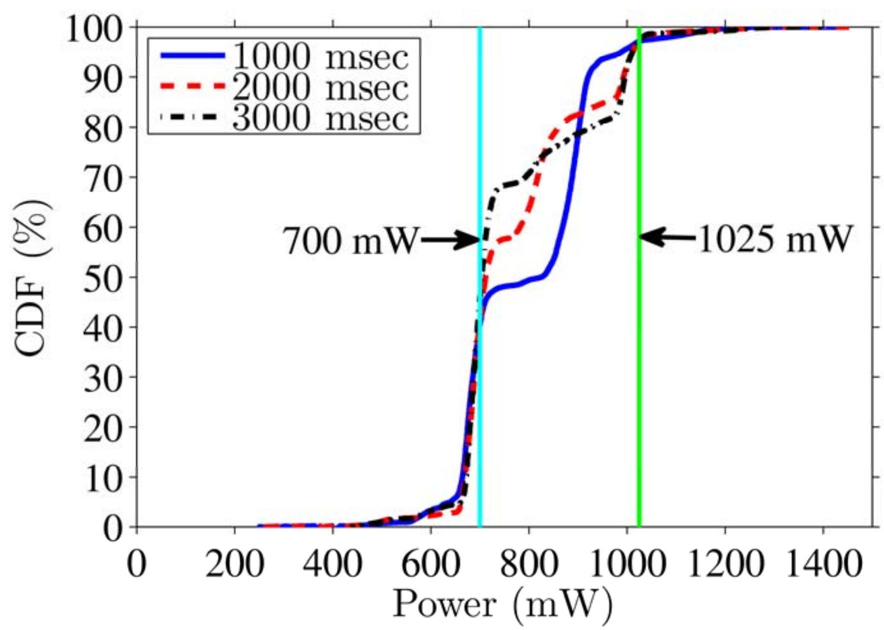
Conclusions

- Proposed a quality-power adaptation framework to systematically control the tradeoff between video quality and viewing time
- Proposed a video broadcast scheme (FMVB) to broadcast scalable streams for viewing time scalability
- Presented the estimation of device dependent parameters, and demonstrated the effectiveness of the proposed adaptation framework
- Framework can be extended to other wireless networks and applications

Questions and Comments

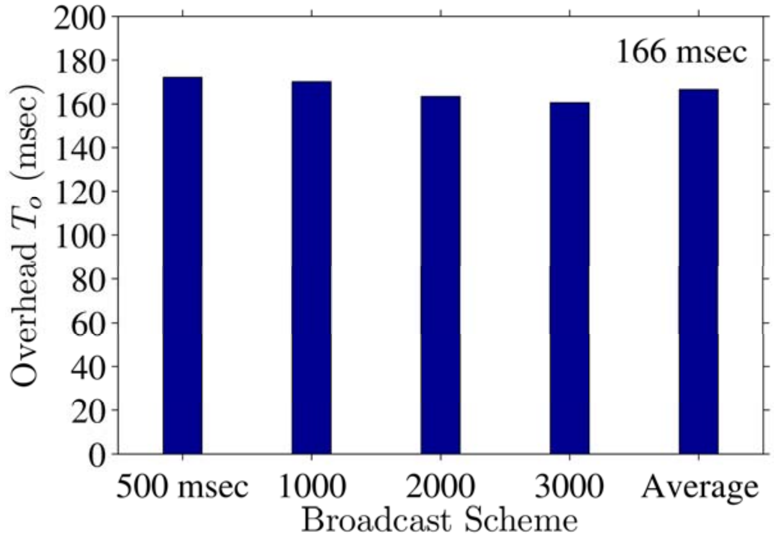
Network Interface Power Consumption and Transition Time (cont.)





- Longer inter-burst periods → fewer spikes
- CDFs indicate that the difference between chip sleep/active modes is $\lambda = 1025 - 700 = 325$ mW

Network Interface Power Consumption and Transition Time (cont.)



- Broadcast schemes with smaller inter-burst periods → more T_o instances
- Comparing any two schemes gives us a T_o value
- $T_o = 166$ msec, which is aligned with 150 msec reported on a recent DVB-H chip datasheet [Philips]