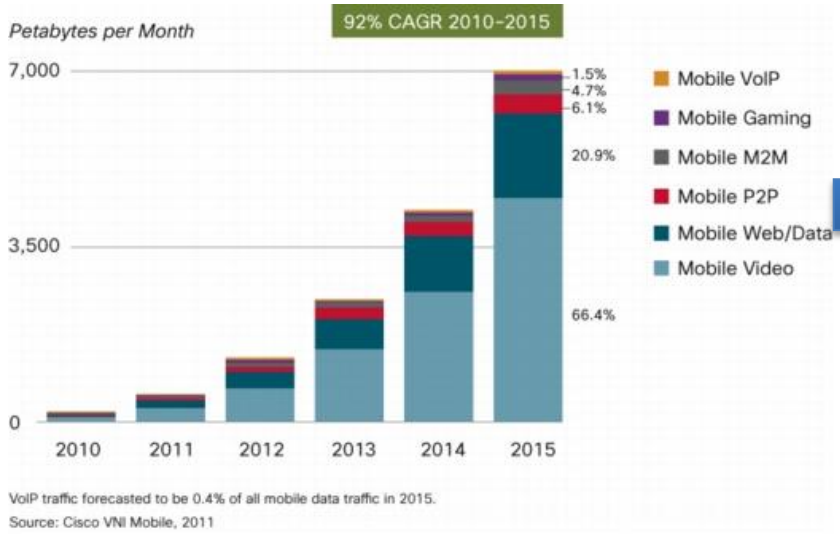
FlexCast: Graceful Wireless Video Streaming

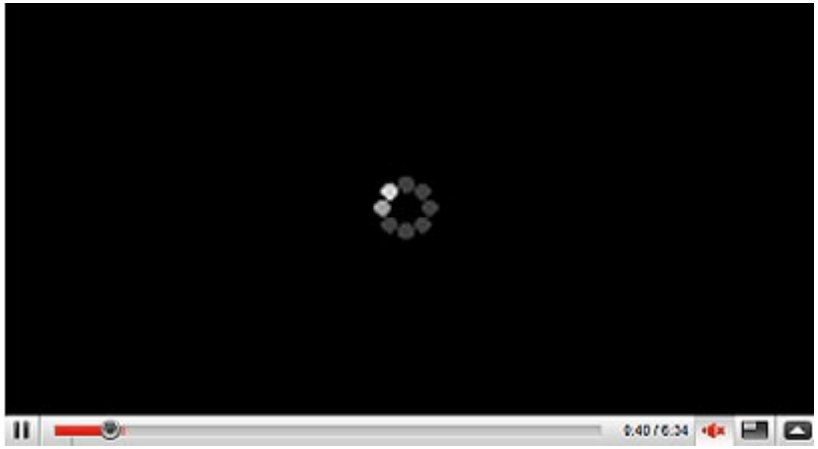
S. Aditya & Sachin Katti

Stanford University

Mobile Video Streaming



Exponential Video Traffic Growth



User experience is poor/choppy

Constant buffering/stuttering, lost frames are quite common in wireless video streaming

Why is Performance Choppy?

Current video streaming (MPEG4..)

- Estimate network path quality over long timescales (mins)
- Encode video at a specific bitrate (e.g 1mbps+ for HD)
- Expect wireless network to deliver that reliable bitrate

Wireless Networks

- Channel strength varies rapidly (on the order of millisecs)
- Do not guarantee a specific minimum reliable bitrate

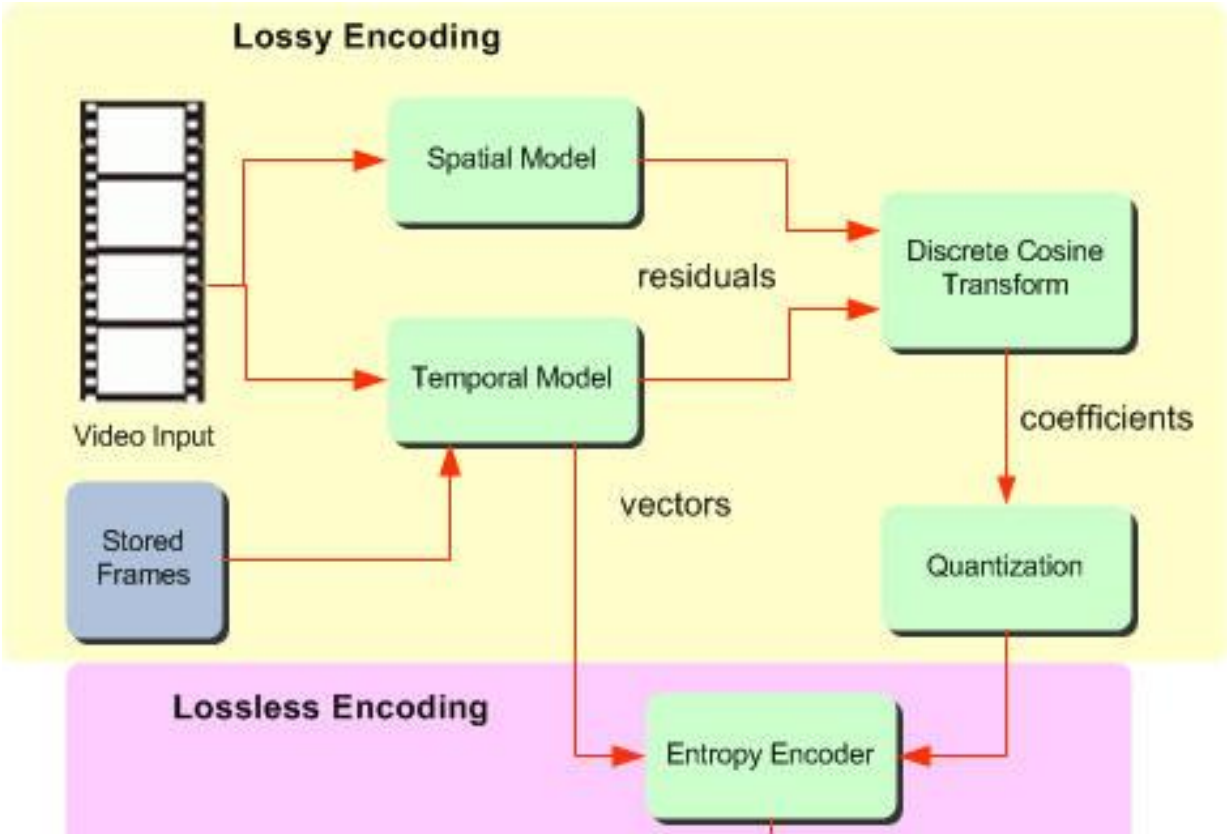
If link bitrate < video bitrate, video stalls or frames skipped
If link bitrate > video bitrate, video quality doesn't improve

This Talk

FlexCast: Novel video codec that allows a receiver to obtain a video reconstruction commensurate with instantaneous network quality

- Rateless video codec (no adaptation needed)
- Modular (no video specific changes to network or PHY/MAC layers)
- Practical (linear encoding/decoding complexity)

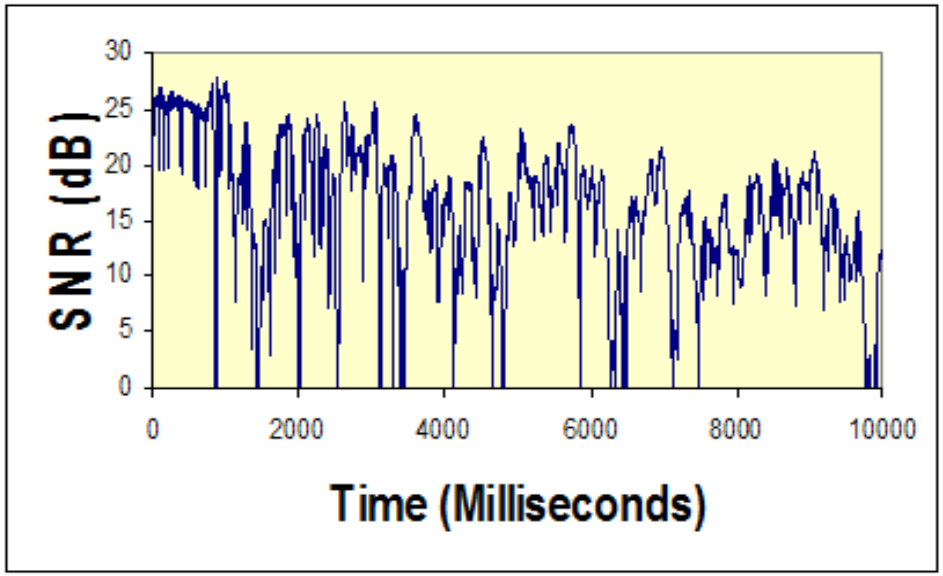
How does traditional video work?



- Output is a compressed constant bitrate stream
- Network is expected to provide a reliable bitrate link greater than the video bitrate to the receiver

Wireless Channels

Channels and Load vary continuously



Varying channels

→ Bit errors

To cope, retransmissions

→ Varying throughput

Why not use packets with errors in decoding video?

Entropy Coding + Wireless Bit Errors → All or Nothing Behavior

Decoding a compressed bitstream that has a few bit errors produces a large number of errors

- Bit error rate gets amplified

Consequently, current video streaming uses a conservative approach

- Picks low video encoding bitrates
- Ensures some minimum video quality
- Cannot take advantage of improved network conditions

FlexCast

Reconstructs a video even from erroneous packets with quality commensurate with current network quality

Key High Level Principle: Bit Errors in packets should translate proportionally into distortion errors in video

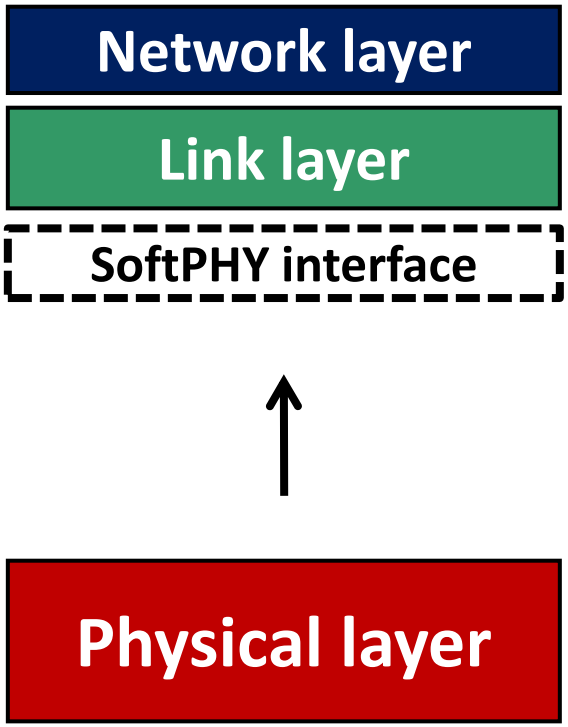
- Entropy Coding does not have that property, a few bit errors can completely distort decoded video
- Flexcast eliminates entropy coding, uses soft reconstruction and proportional representation to achieve proportionality

Key Insight 1: Soft Reconstruction

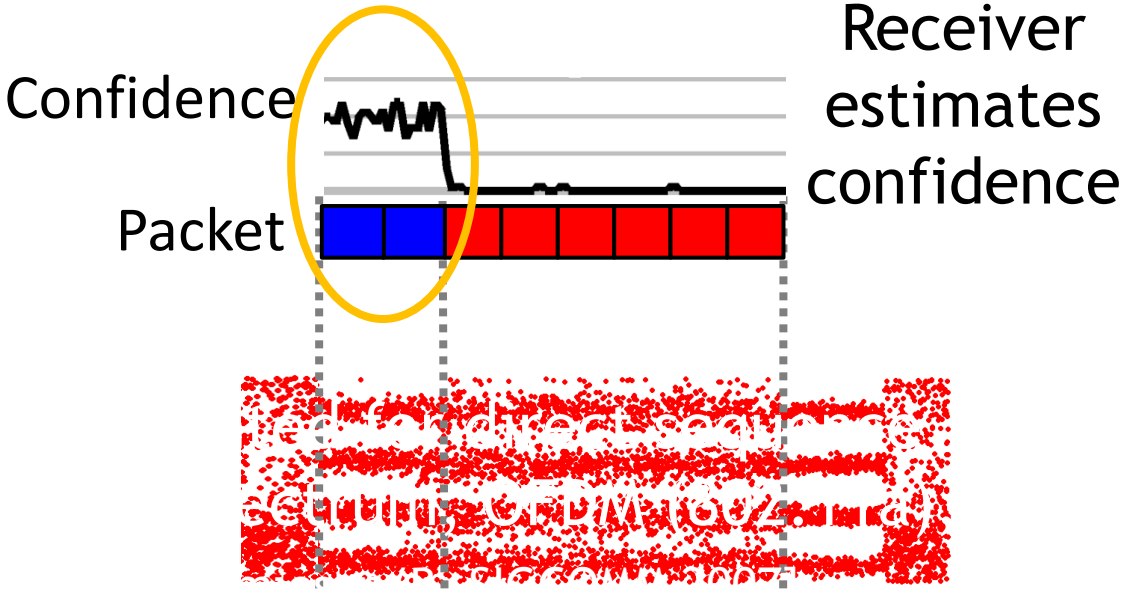
- PHYs compute soft estimates of decoded bits → probability bit is “1” or “0”
  - Soft Output Viterbi Decoder, SoftPHY
- Leverage soft information to compute expected value of DCT coefficient

Soft Reconstruction: SoftPHY

- Cross-layer information flow from PHY up
- Extract and use soft estimates from PHY



- Maintain layered architecture (PHY-independent use)



Soft Reconstruction

Video Sender

DCT Coeff: 13

Binary: 1101

Soft Reconstruction

DCT Coeff: 0.5*8+0.9*4+0.6*2+0.3*1 = **9.1**

Traditional Reconstruction

DCT Coeff = 0+4+2+0 = **6**

Binary: 0110

Soft: 0.5,0.9,0.6,0.7

SoftPHY

Soft Reconstructed DCT coefficient (9.1) is much closer to the transmitted value than traditional (6)

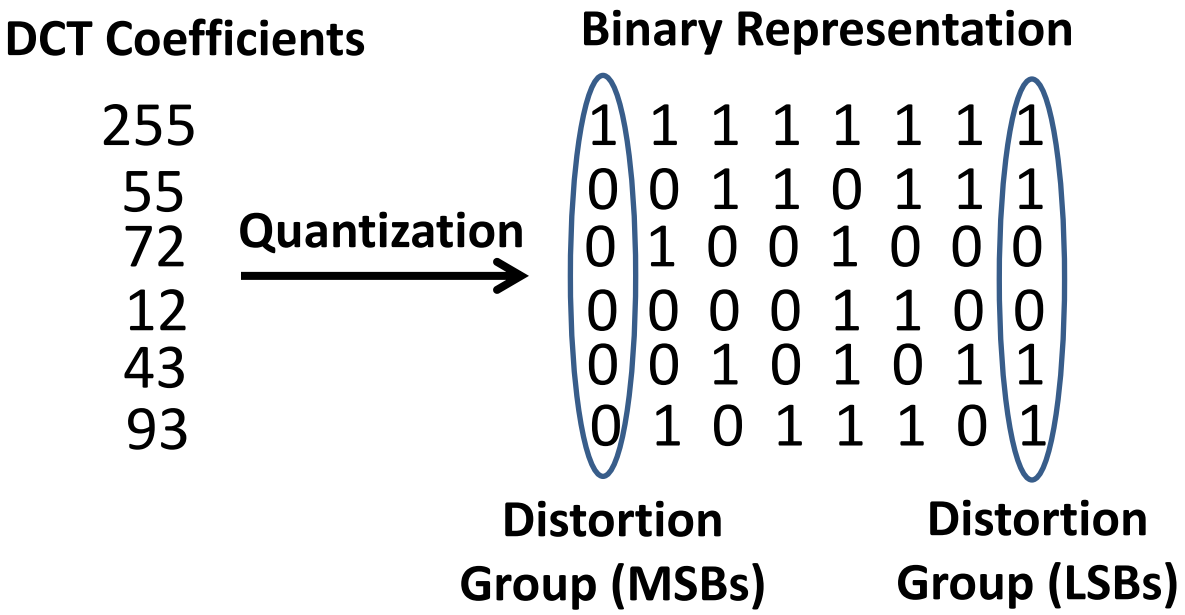
Key Insight 2: Proportional Representation

- Soft Reconstruction is not sufficient since some bits are more important than others
  - MSB of the low frequency DCT coefficient, single bit error translates to large video distortion

- Second Principle: Design a technique that allows sender to provide unequal error protection (UEP) without modifying the PHY

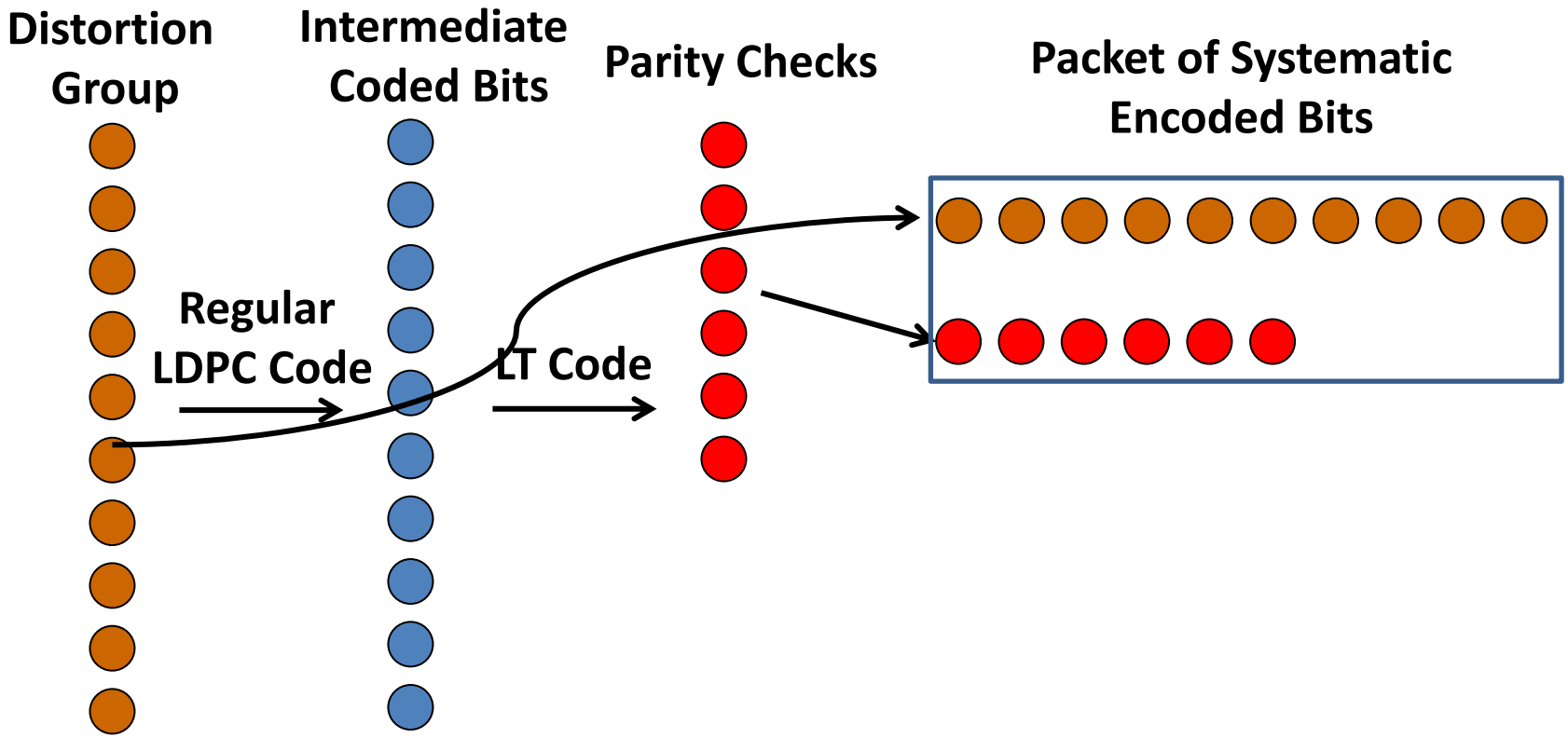
Distortion Grouping

Identify how important groups of bits are by estimating the amount of distortion they would cause if decoded in error



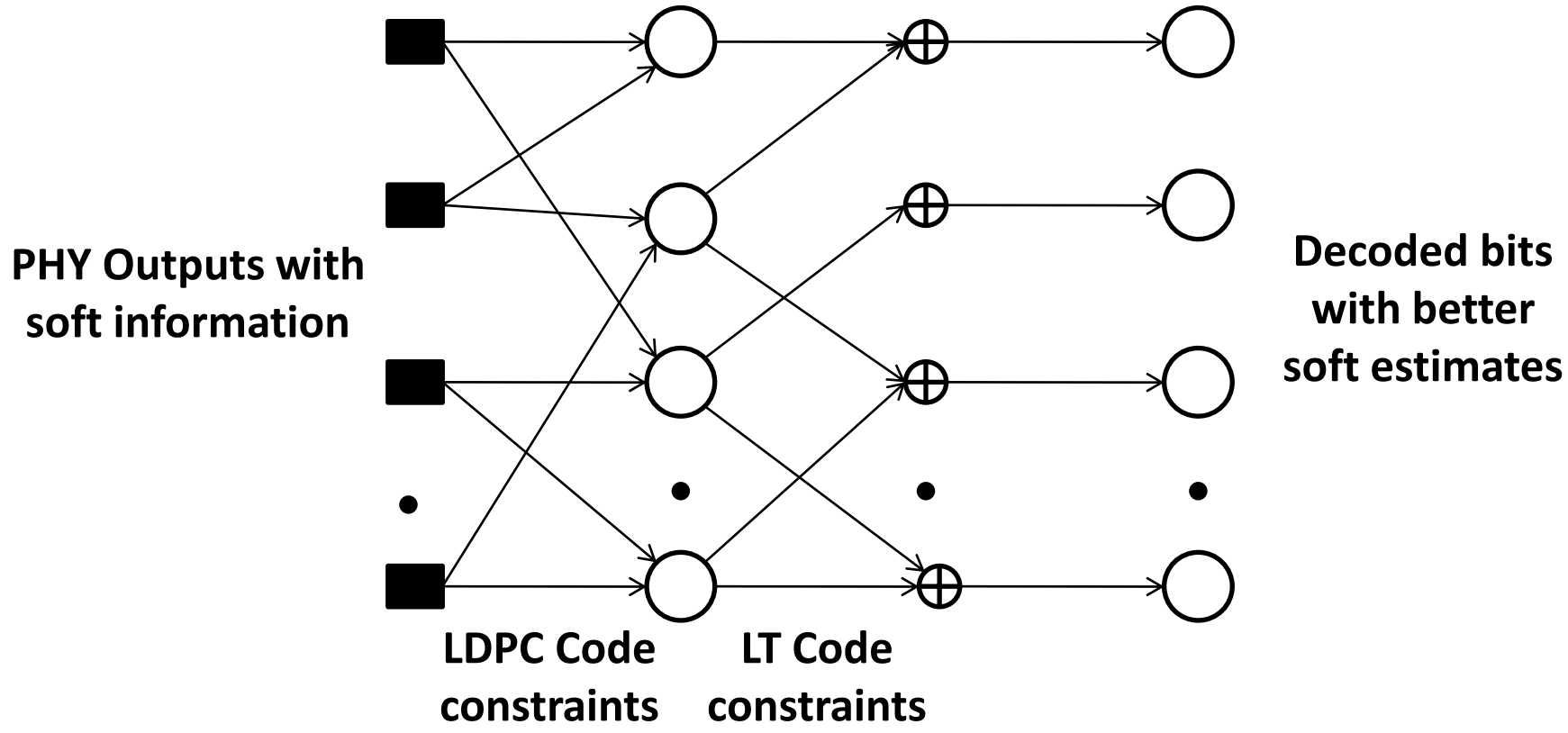
Rateless Code for UEP

To provide UEP, sender encodes each distortion group with a rateless Raptor code



Soft Raptor Decoder

- PHY passes demodulated bits with soft information
- Apply Belief Propagation to decode Raptor Code and compute soft estimate for the DCT coefficient bits



Rateless Raptor Code for UEP

Key Property: Soft estimate of decoded bit improves with every received rateless bit

- Standard Belief Propagation Decoder
- Intuitively, confidence in decoding decision improves with every extra coded bit

  → **By controlling the number of rateless bits allocated to each distortion group we can obtain proportional representation or UEP**

Putting it All Together: Video Sender

- Group DCT coefficient bits into distortion groups ranked according to importance
- Encode each distortion group with rateless code
- Create packets with bits from all groups
  - No of coded bits allocated in each packet for each distortion group is proportional to importance

No video bitrate is ever picked and no path quality estimation is required

Putting it All Togther: Network

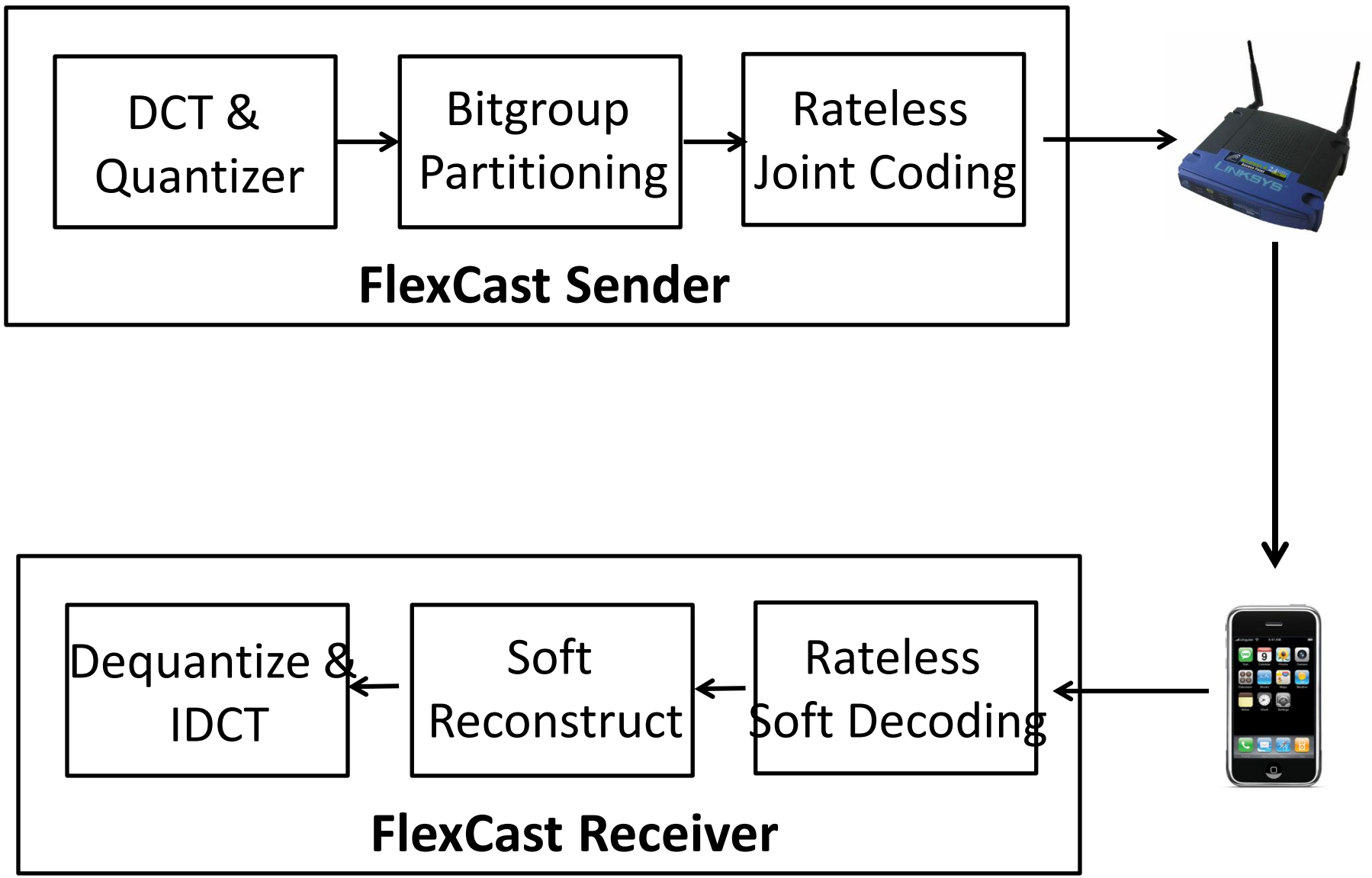
Nothing to do!

Network behaves exactly as before, no changes needed

Putting it All Together: Receiver

- PHY passes demodulated bits with soft information
- Apply Belief Propagation to decode Raptor Code and compute soft estimate for the DCT coefficient bits
- Apply soft reconstruction to estimate the expected value of the DCT coefficients
- Compute Inverse DCT to get original pixels!

FlexCast Architecture



Implementation

- FlexCast is implemented by modifying standard MPEG4 implementation
- Algorithms have linear complexity and are practical to implement
- Microbenchmarks (on Core i7 980x) for 720p Video

Channel SNR	12dB	10dB	8dB	6dB
CPU Time/Actual Video Time	0.3	0.37	0.44	0.51

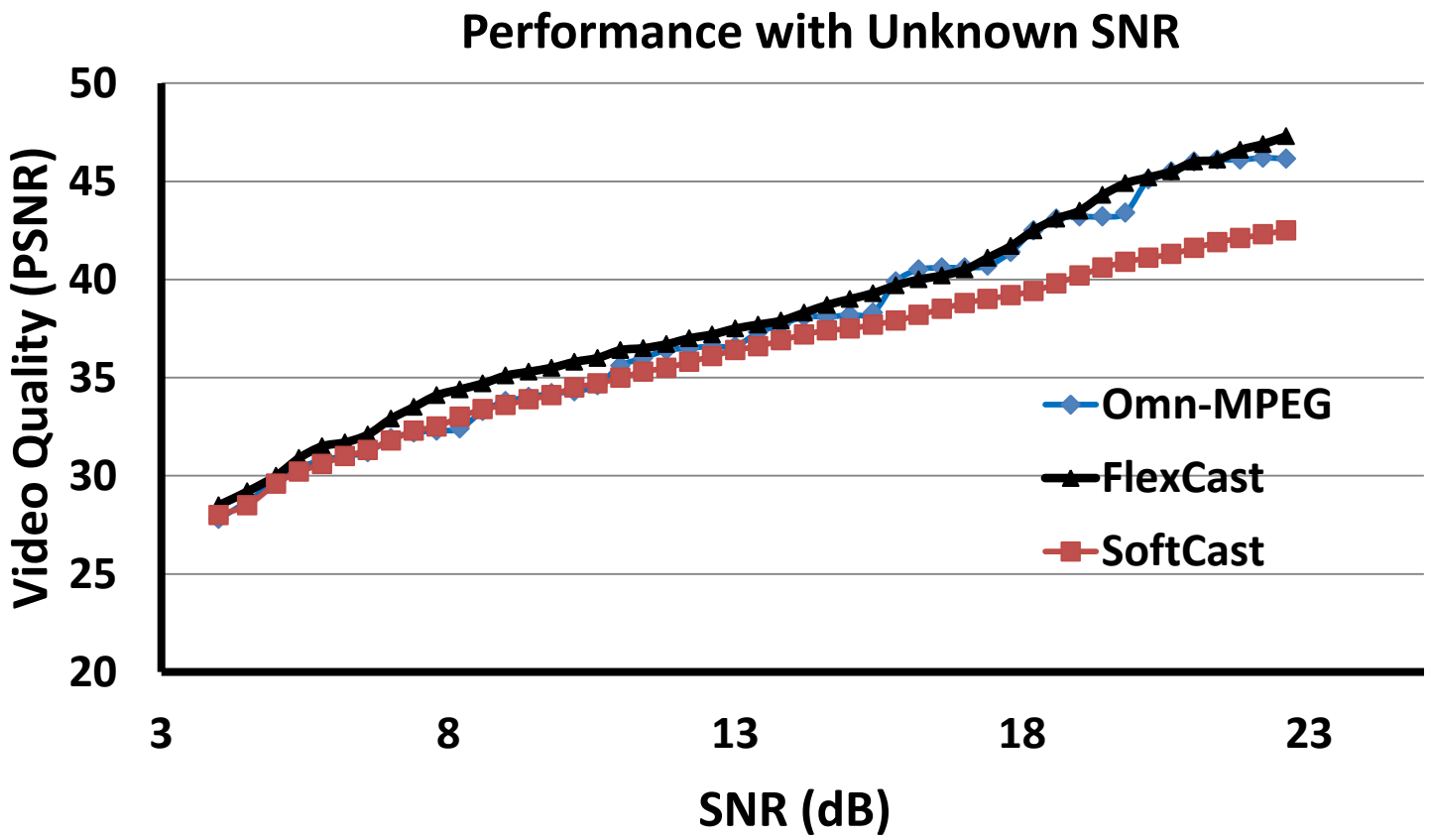
Evaluation Setup

- Deployed in an 10 node indoor USRP2 testbed
- PHY: WiFi style OFDM, 6.25 MHz channel
- Standard WiFi convolutional coding rates

Compared Approaches

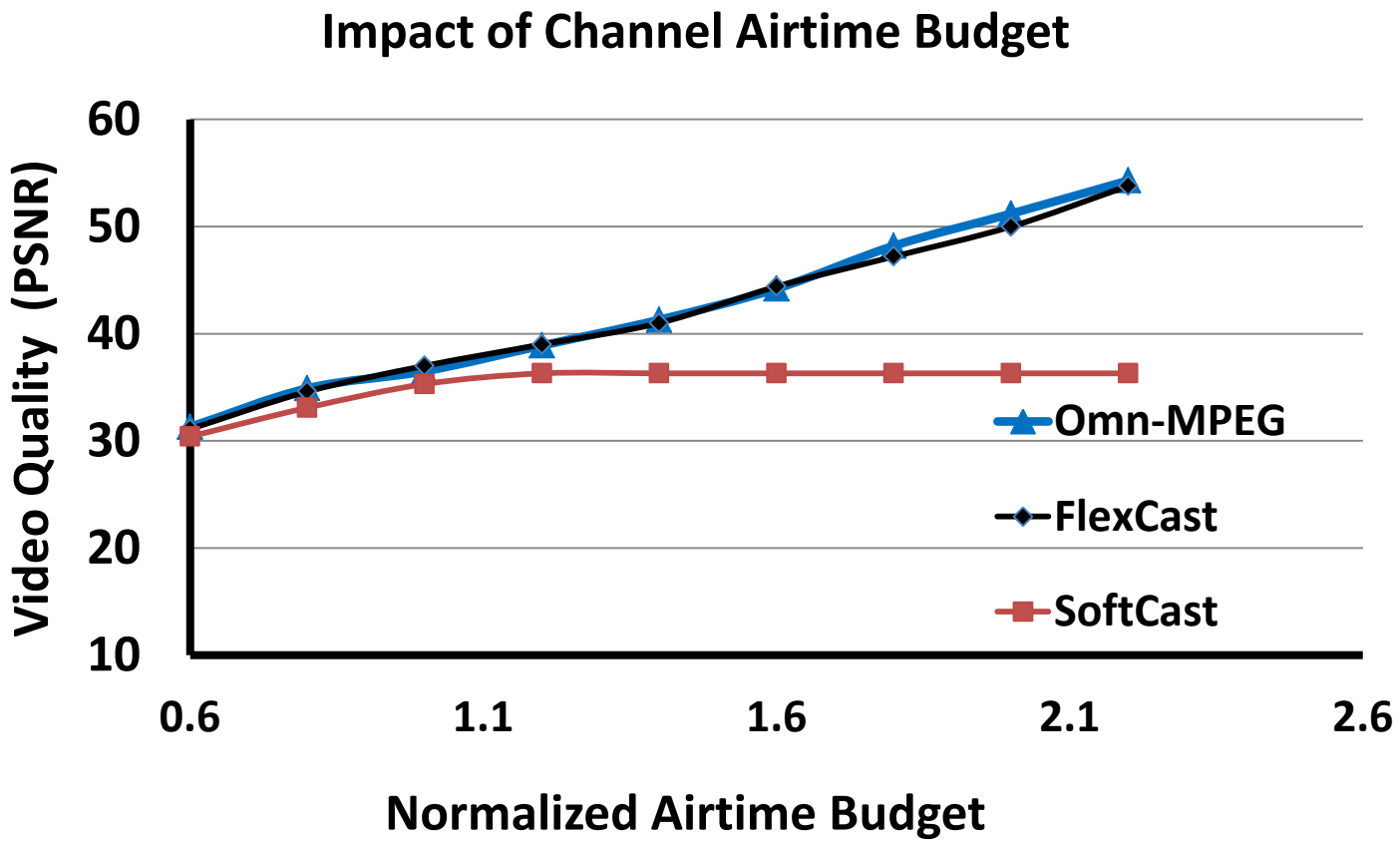
- **Omniscient Scheme**
  - Perfect advance channel knowledge
  - Picks best Wifi bitrate and video encoding bitrate that maximizes PSNR of received video
- **SoftCast**
  - Clean slate mobile video design (simpler version of next paper!)
- **Apex** (Sigcomm 2010)
  - UEP at the PHY layer

How Graceful is FlexCast?



FlexCast performs as well as the omniscient scheme without requiring any channel state knowledge

Can FlexCast automatically exploit additional capacity?

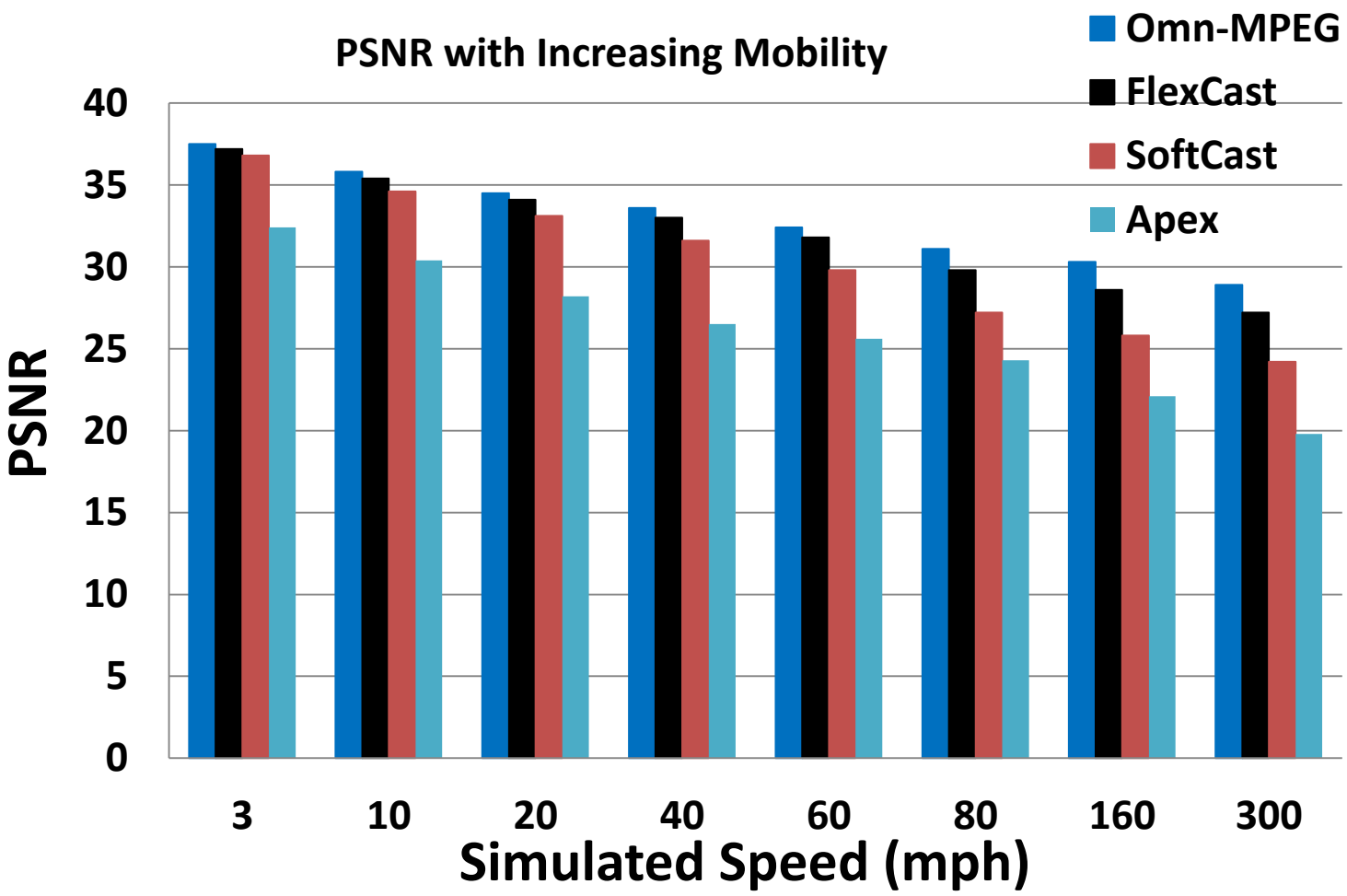


FlexCast automatically provides higher video quality if link provides additional capacity

Trace Driven Emulation

- Stanford RUSK channel sounder
  - High precision channel measurement
  - Continuous channel state information
- 2.426 GHz to 2.448 GHz
- Each trace: 100000 measurements over 100 sec
- 10 mobility traces at walking speed ~ 3 mph
- Simulate mobility by playing trace at increasing speeds

Performance with Increasing Mobility



FlexCast provides near optimal performance at high mobility

Conclusion

FlexCast provides graceful video streaming for dynamically varying wireless networks

- Rateless, Modular and Practical

Espouses a design philosophy that takes into account wireless channel properties to build modular robust protocols and systems