Qualcomm Research

RaptorQ™

RaptorQ Forward Error Correction Technology Overview and Use Cases

October 2012

QUALCOMM®

- Presentation Contents
  - Raptor Forward Error Correction Technology
  - RaptorQ File Delivery Service Overview diagram
  - RaptorQ FEC Solves Digital Media Challenges
  - FEC Implementation Considerations for File Delivery using RaptorQ chart
  - RaptorQ File Delivery: FLUTE example
  - RaptorQ Streaming Video/Media Protection Process Overview diagram
  - FEC Implementation Considerations for Streaming Video/Media using RaptorQ chart
  - RaptorQ Streaming Implementation: FEC Framework
  - Challenges of adding FEC and RaptorQ Advantages chart
  - Use Case: File Delivery Services over Satellite Networks
  - Use Case: Mobile Live Broadcasting
  - Use Case: Streaming Video Conferencing within the Enterprise
  - Use Case: Mobile File Delivery Services over Cellular Networks
  - Use Case: IP Set-Top-Box Streaming Video Service
  - Raptor Technology Solutions Summary
  - Products and Business Model

Raptor Forward Error Correction Technology

Raptor is generically known as Application Layer-FEC (AL-FEC) and can be integrated into software to improve a wide range of content delivery solutions. FEC technologies solve network packet loss by sending repair data in addition to the original source data.

RaptorQ is the latest Raptor version and has the best performance of any AL-FEC code. Raptor Technology encodes original data at the sender and then decodes it at the receiver from an amount of received encoded data only slightly greater than the original source data, regardless of which specific data has been received.

Key Benefits:

- Improves customer QoS
- Reduces operating costs
- Eliminates HW expense
- Improves timeliness
- Superior for mobile apps

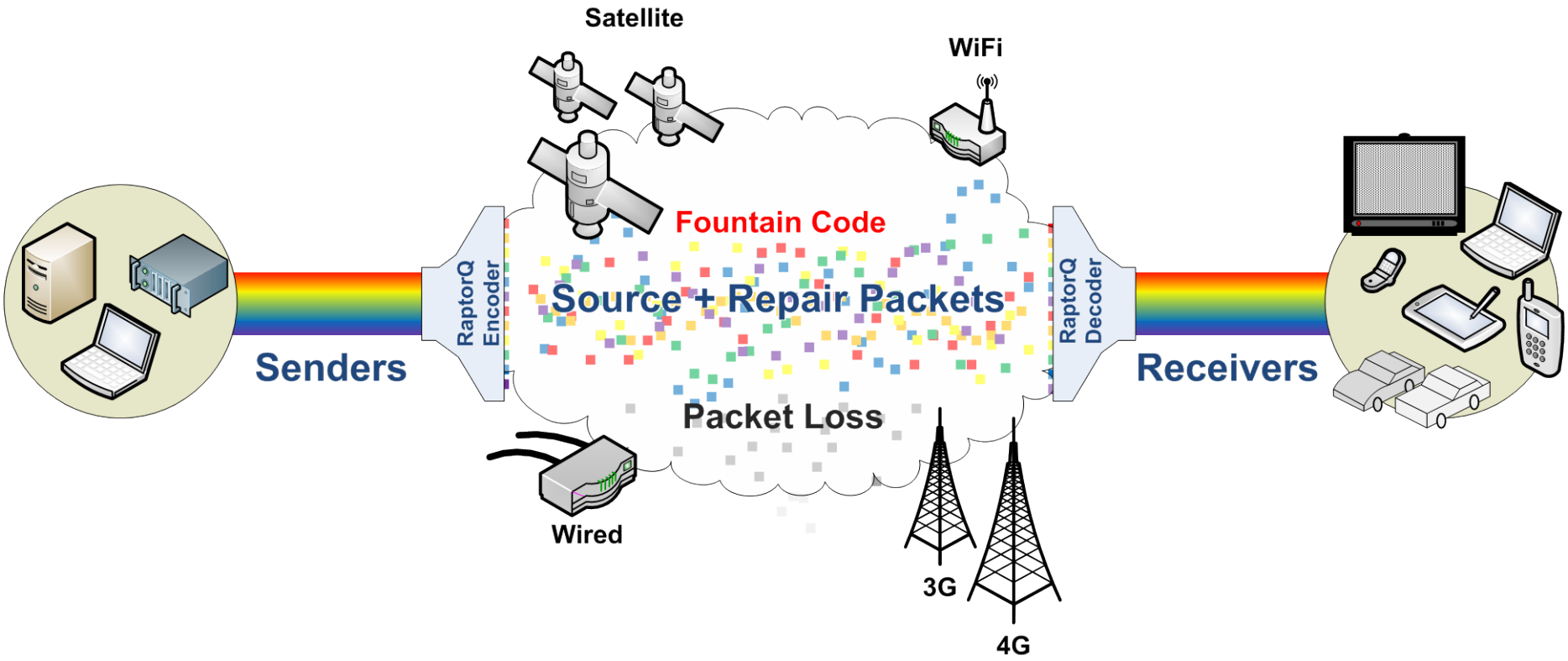
Why is Raptor the best FEC choice:

- Efficiency - very fast encode/decode, low cost/power device
- Superior Performance – HD video, high-speed data transfer
- Flexibility - low latency streaming to large file transfer
- Versatile - device/network/operating system agnostic
- Cost Effective - software only solution for easy integration

Where is the biggest Benefit?

- Networks with high/bursty packet loss, large latencies, intermittent connectivity
- Sending very large files efficiently over the network – using less bandwidth and memory
- Receiving large files (movies) on memory constrained devices (handsets/STB)
- Enabling HD streaming with existing hardware
- Services without a backchannel (i.e. over UDP) t require high QoS content delivery

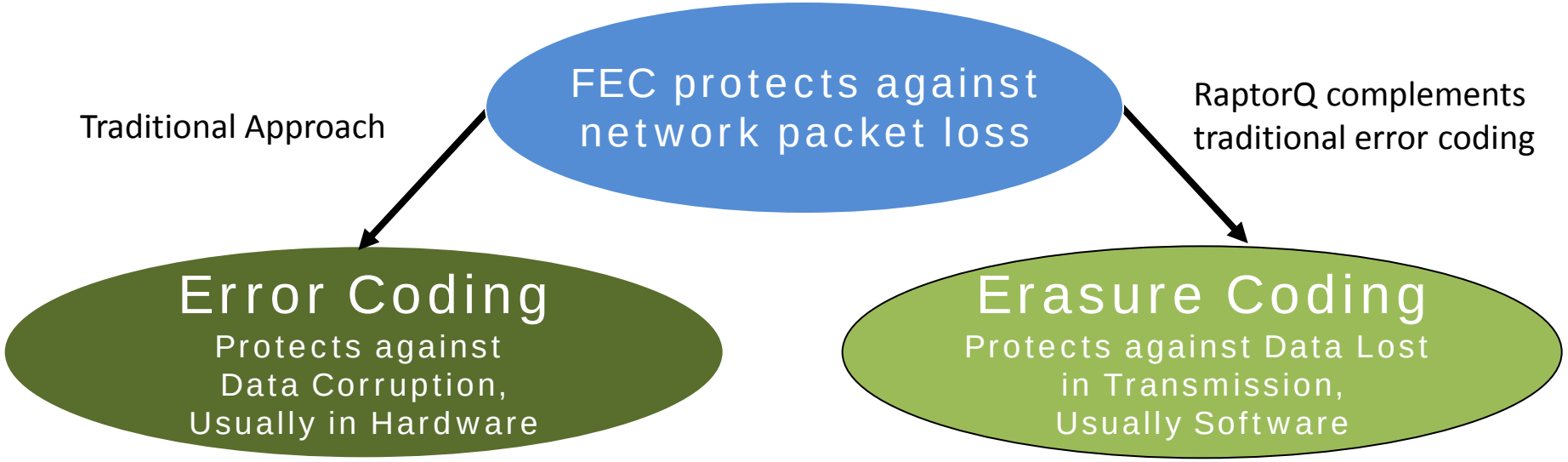
RaptorQ File Delivery Service Overview



Reliable and Efficient File Delivery Over any Network to any Device

Forward Error Correction (FEC) Technology

RaptorQ solves digital media delivery challenges



RaptorQ is a systematic fountain code

- Encoding = source + repair generated from the source
- Efficiently generate as much repair as desired, on the fly
- Recovery of source is possible if enough encoding is received
  - Independent of what is lost
- Can dynamically change the amount of repair based on loss

RaptorQ - the most advanced software FEC

- Highly efficient – allows excellent data recovery
- Optimized for mobile
- Wide range of applications

RaptorQ is used in an application specific manner

- Software only
- File delivery – encode the entire file as a block or a set of sub-blocks if receiver memory is limited
- Streaming – encode blocks of the stream
- Can be used at MAC layer to protect all data

RaptorQ is IETF standards RFC 6330 & 6681

FEC Implementation Considerations for File Delivery using RaptorQ

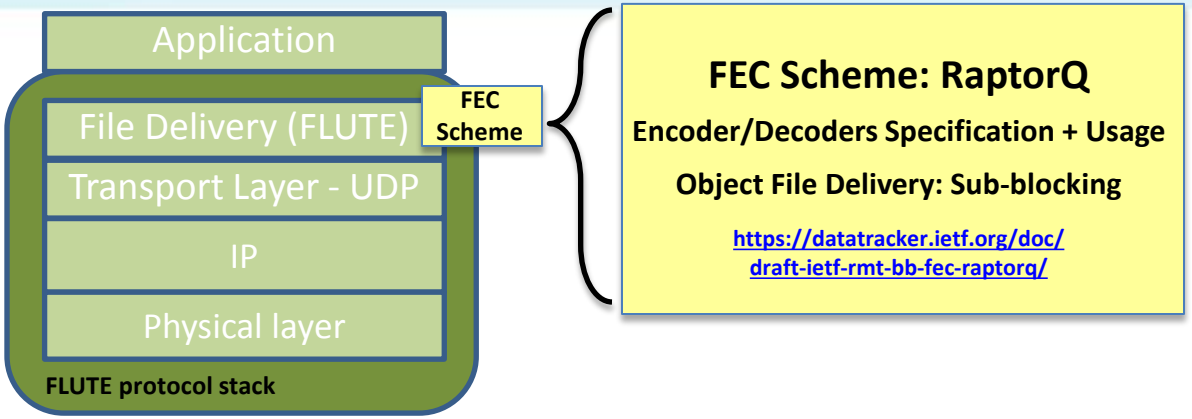
Application layer FEC is used in file delivery solutions to protect against data loss. Technical and business challenges require trade-offs decisions when choosing and implementing a file delivery solution using FEC.

Challenge	Solution	RaptorQ Advantage	How to implement
Maximize file protection in the most efficient way.	Sending the file as a single source block maximizes decoding efficiency and protection due to time diversity (spreading the protection across the whole file).	Very CPU efficient encoding and decoding algorithm, particularly for sending source blocks.	Object files up to 3.4GB can be sent as a single source block.
Recovery of file using limited available receiver memory.	Recover small portions of the file at a time.	Sub-blocking algorithm preserves the time diversity advantage of using larger source blocks, while providing efficient memory decoding of one sub-block at a time.	Partion large source blocks into multiple sub-blocks based on the size of available receiver memory.

RaptorQ has been optimized to be the best performing and most flexible FEC for file delivery services, particularly those with very large files.

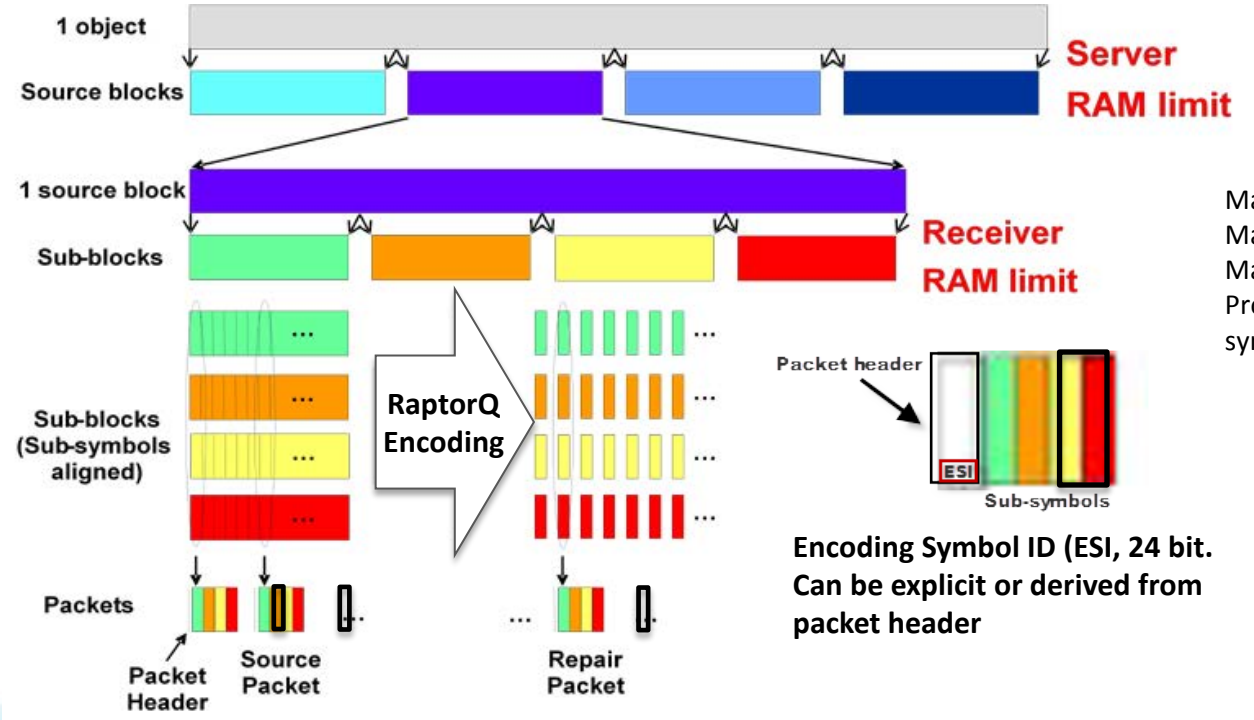
RaptorQ is RFC 6330, IETF FEC standard for object delivery over WWAN.

RaptorQ File Delivery: FLUTE Example



Properties of Qualcomm implementation of RaptorQ FEC Scheme codecs:

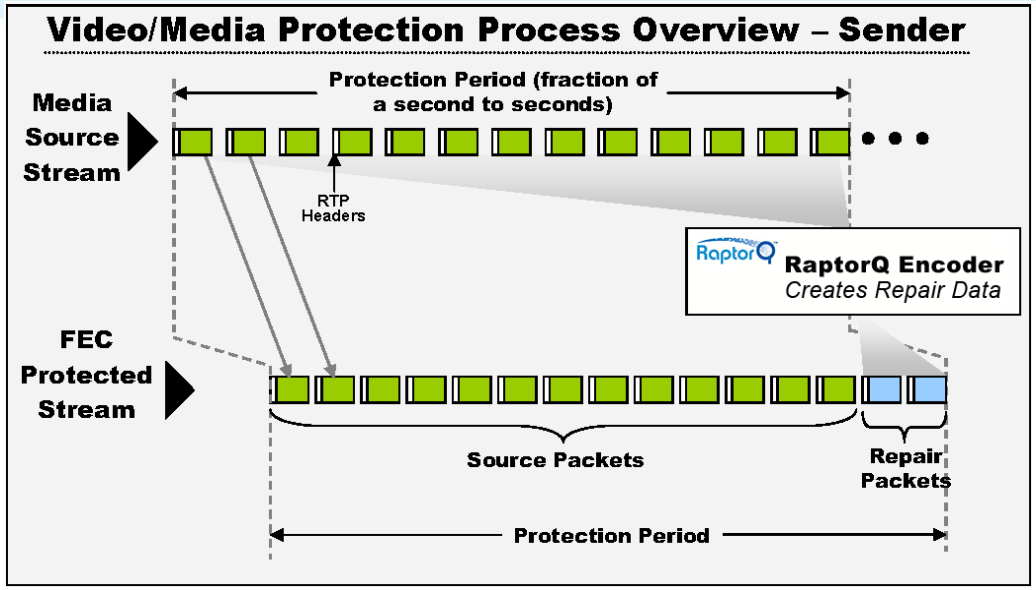
- Highly portable C code
- Very fast, patented algorithms and data structures
- Excellent decoder CPU performance
- Verified to be compliant to RaptorQ specification

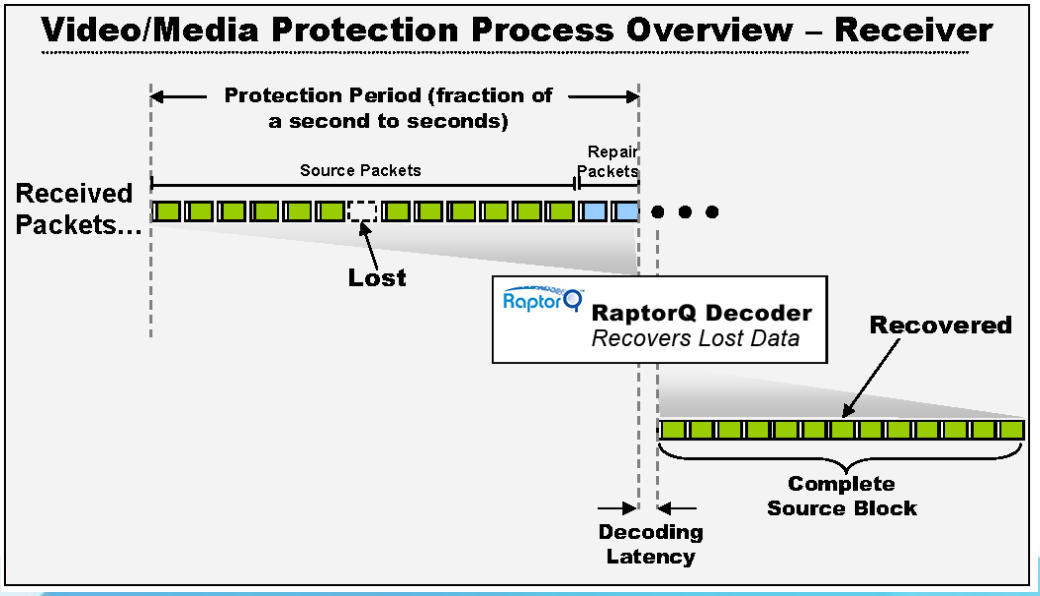


Encoding Symbol ID (ESI, 24 bit. Can be explicit or derived from packet header

Max symbol size in octets: 65535 (T)
Max number of source symbols: 56,403 (K)
Max source block size in octets: 3.4 GB
Probability decoding failure: receive K+2 symbols is 1e-6

RaptorQ Streaming Video/Media Protection Process Overview





FEC Implementation Considerations for Streaming Video/Media using RaptorQ

Application layer FEC is used in streaming solutions to increase Quality-of-Service (QoS). Technical and business challenges require trade-offs decisions when choosing and implementing a streaming solution using FEC.

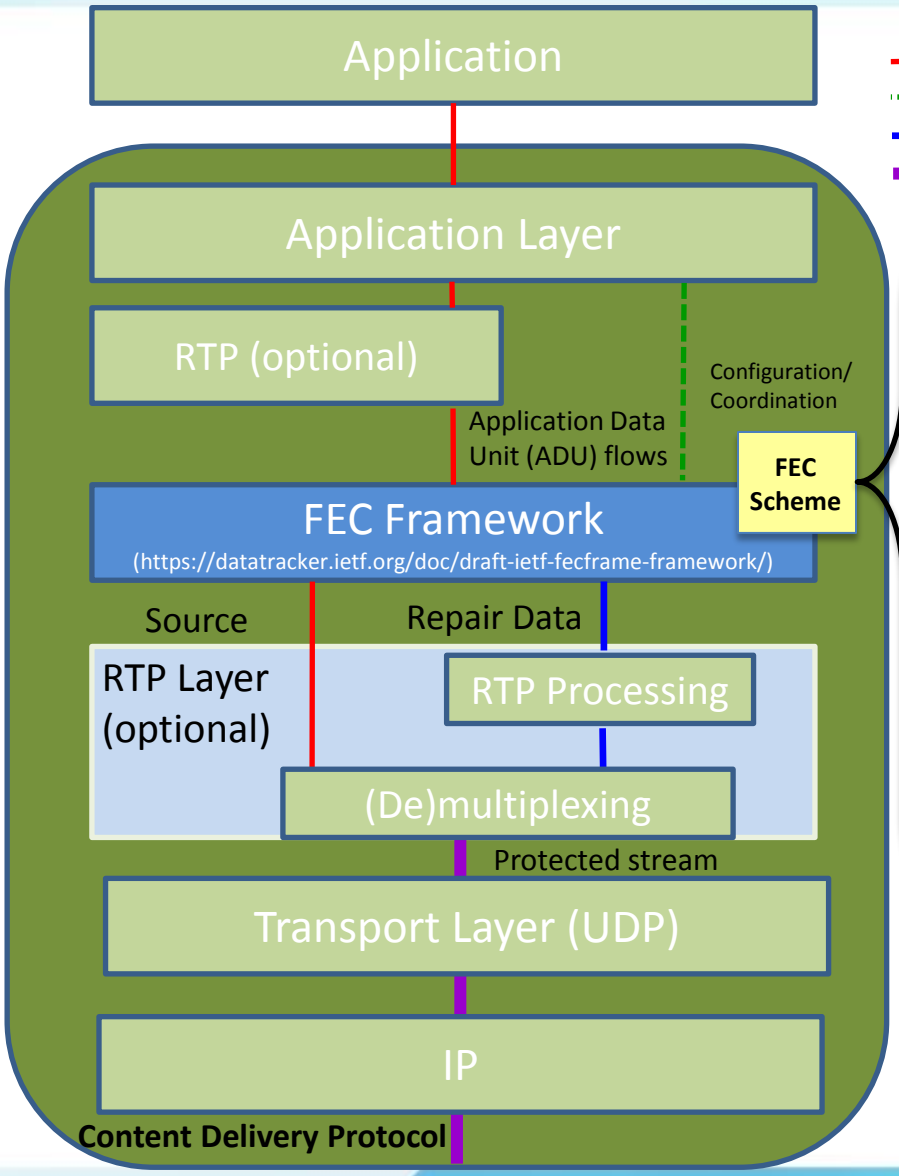
Challenge	Solution	RaptorQ Advantage	How to implement
Maximize Quality-of-Service (QoS) while minimizing latency.	Determine how delay-sensitive your application/service is (e.g. video conferencing is more sensitive than video broadcast) and use the largest acceptable source block size.	Flexible block size parameter range and fountain property provides best FEC protection due to time diversity and low CPU required for decoding. Allows flexible QoS trade-off between increasing protection amount vs. increasing latency.	Split the video/audio stream into consecutive source blocks and add RaptorQ repair data to the blocks to protect against packet loss. Send the source + repair data of a given block in the same time period slot that would be used to send just the source block with no FEC added (additional bandwidth needed is directly proportional to amount of repair data).
Ensuring QoS in a mobile environment while optimizing network power requirements.	Maximize coverage area and minimize transmit power needs to achieve target QoS.	Fountain code property and low code overhead reduces bandwidth usage, resulting in lower power needs for a given coverage area and QoS target.	

RaptorQ has been optimized to maximize transmit power management and minimize latency in streaming solutions, ensuring the best possible QoS.

RaptorQ is RFC 6681, IETF FEC standard for media streams for FECFRAME

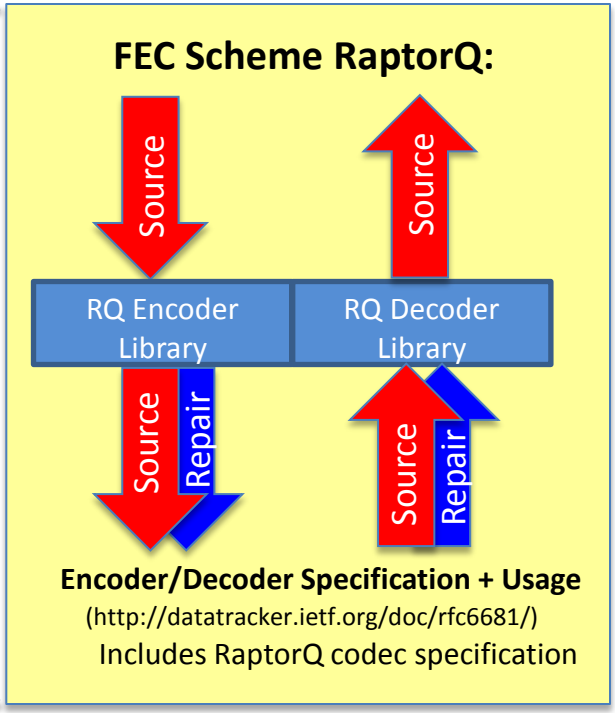
RaptorQ Streaming Implementation: FEC Framework

How to Protect a Video Stream



Legend:

- : data flow, unidirectional or bidirectional
- : control flow
- : repair data stream
- : protected stream = source+repair data



Properties of Qualcomm implementation of RaptorQ FEC Scheme codecs:

- Highly portable C code
- Very fast, patented algorithms and data structures
- Excellent decoder CPU performance
- Verified to be compliant to RaptorQ specification

Challenges of Adding FEC and RaptorQ Advantages

Challenges of adding FEC	RaptorQ Advantage
Application	
Cost to add FEC	Software only - no additional hardware costs
Very large files (digital cinema) that require efficient delivery	Large block sizes decode faster, more bandwidth efficient to send
Service/application runs on a custom device (STB, vehicle navigation)	Easily portable for platforms with C compiliers; ready-to-go for Intel x86 and ARM; operating system independent.
Service/application requires very fast encode/decode	Performance data available to verify requirements
Device/Receiver	
Small receiver memory - not enough to decode whole file in memory increases decoding time	Sub-blocking feature allows large files to be divided into smaller chunks that fit the reciever memory; great for handsets, STBs, low-cost devices
Receiver device has limited CPU available for decoding	Low CPU requirement

Challenges of adding FEC	RaptorQ Advantage
Network	
Highly intermittent reception	Fountain code property allows for a long transmission period
Given network loss is not consistent, how to maximize Quality-of-Service (QoS) by adding the minimum amount of FEC	Dynamically set protection amount (if backchannel is available) - allows optimal use of bandwidth with best QoS
Bandwidth contstraints	Low code overhead (2 additional symbols) allows virtually perfect delivery (99.9999%)
No backchannel available for network feedback	Receiver does not require notification about the amount of repair - simplifies architecture
Some networks have very high loss and need large amounts of repair data	Fountain code property allows unlimited amount of repair data to be sent
Network experiences bursty loss	File delivery with sub-blocking makes it immune to bursty loss

File Delivery Services Over Satellite Networks

Challenge: Digital Cinema

Need fast, reliable and efficient multicast digital cinema file delivery that can be cost effectively delivered

Receivers

Fixed: Hundreds of movie theater locations nationwide

Integration: Runs directly on enterprise's digital cinema servers

Challenge: Vehicle Navigation

How to reliably update map databases to moving vehicles that are not always connected to the network

Servers

Receivers

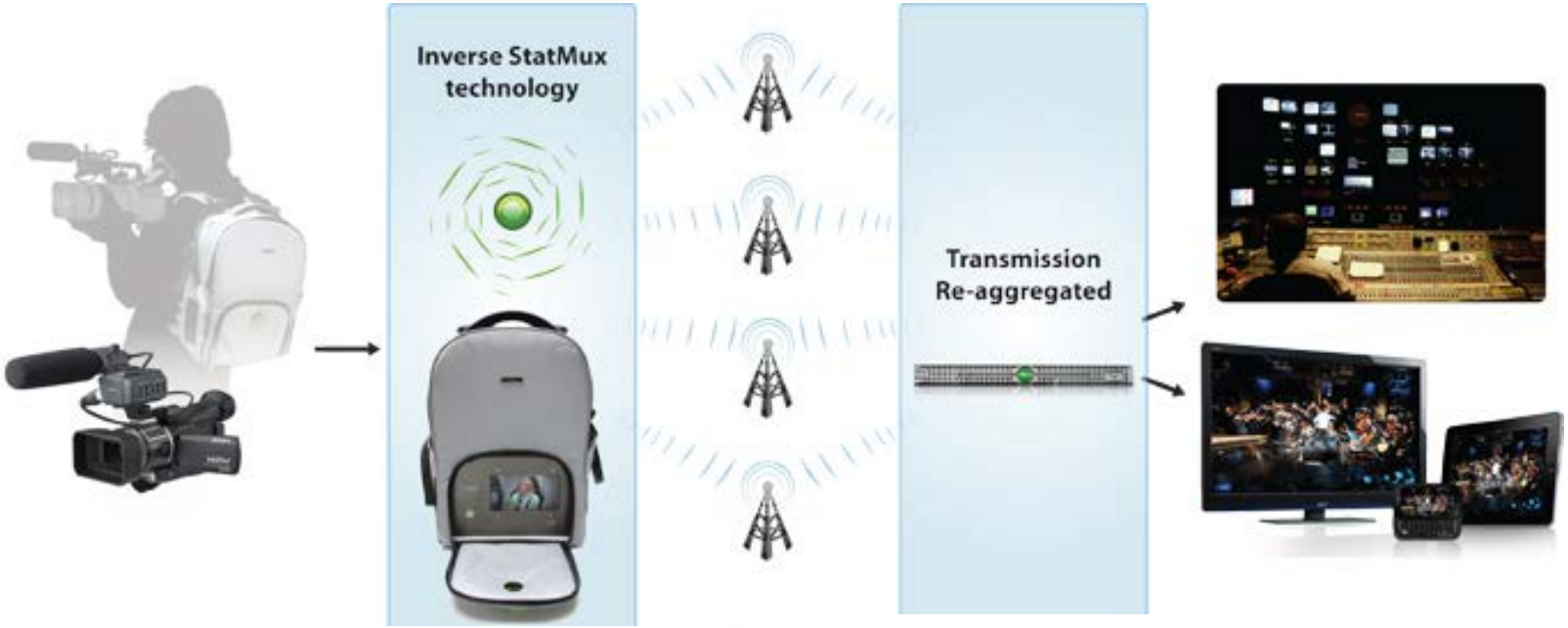
Mobile: GPS vehicle navigation systems

Integration: embedded platform with onboard GPS software

Mobile Live Broadcasting

Challenge:

- **Streaming HD video over cellular networks**
- **Requires low latency with high QoS**



Application Features:

- Broadcast quality live video streaming
- Mobility allows broadcasts from anywhere
- Dynamically segment live video and transmit via multiple cellular connections

Video Conferencing System

Challenge:

- **Vivid real-time video conferencing**
- **Requires low bandwidth with HD video quality**
- **Requires low latency with high audio QoS**

Enterprise LAN/WAN

Receivers

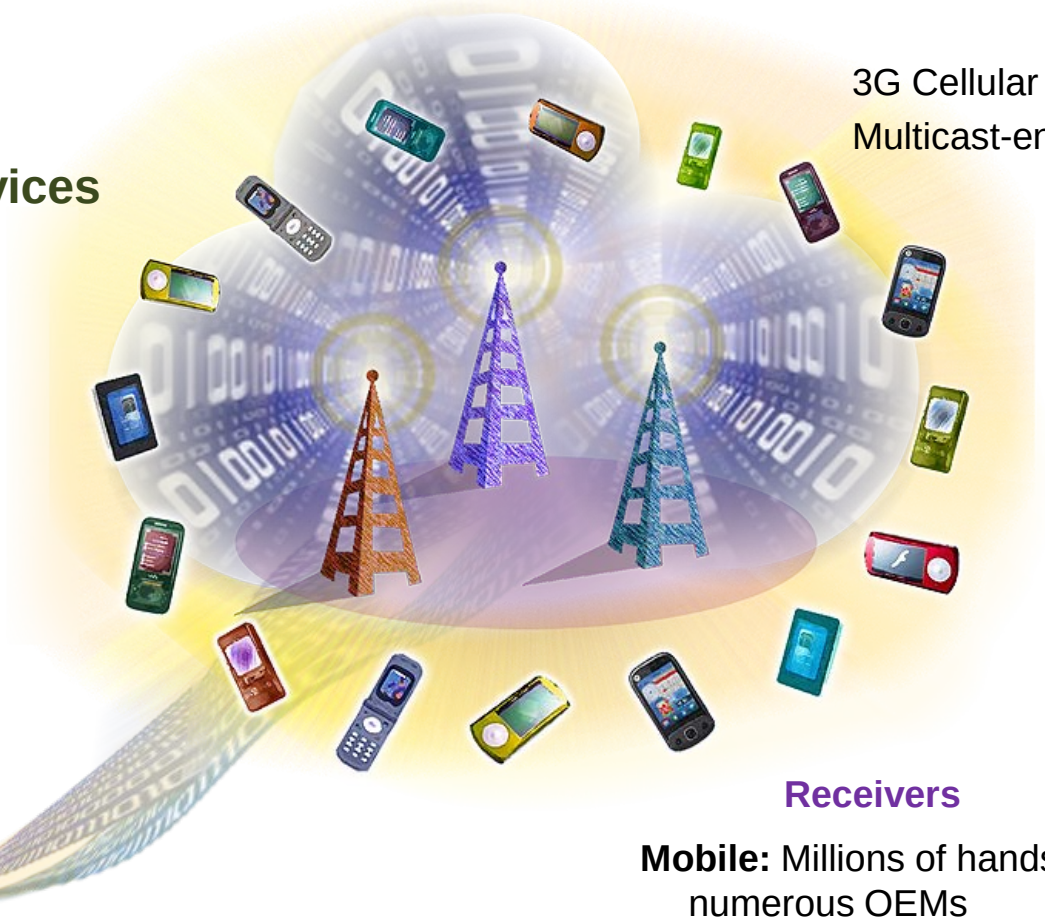
Fixed: Within the Enterprise
Integration: Within VCS product

Application Features:

- HD video streaming
- High quality audio streaming
- Content sharing during streaming
- Multi-way call (up to 4 sites)

Mobile File Delivery Services Over Cellular Networks

Challenge: reliable file delivery to mobile devices



3G Cellular
Multicast-enabled

Applications:

- Movie clip file casting
- News ticker
- Roaming list updates

Servers

Receivers

Mobile: Millions of handsets from numerous OEMs

Integration: Within handset flash image

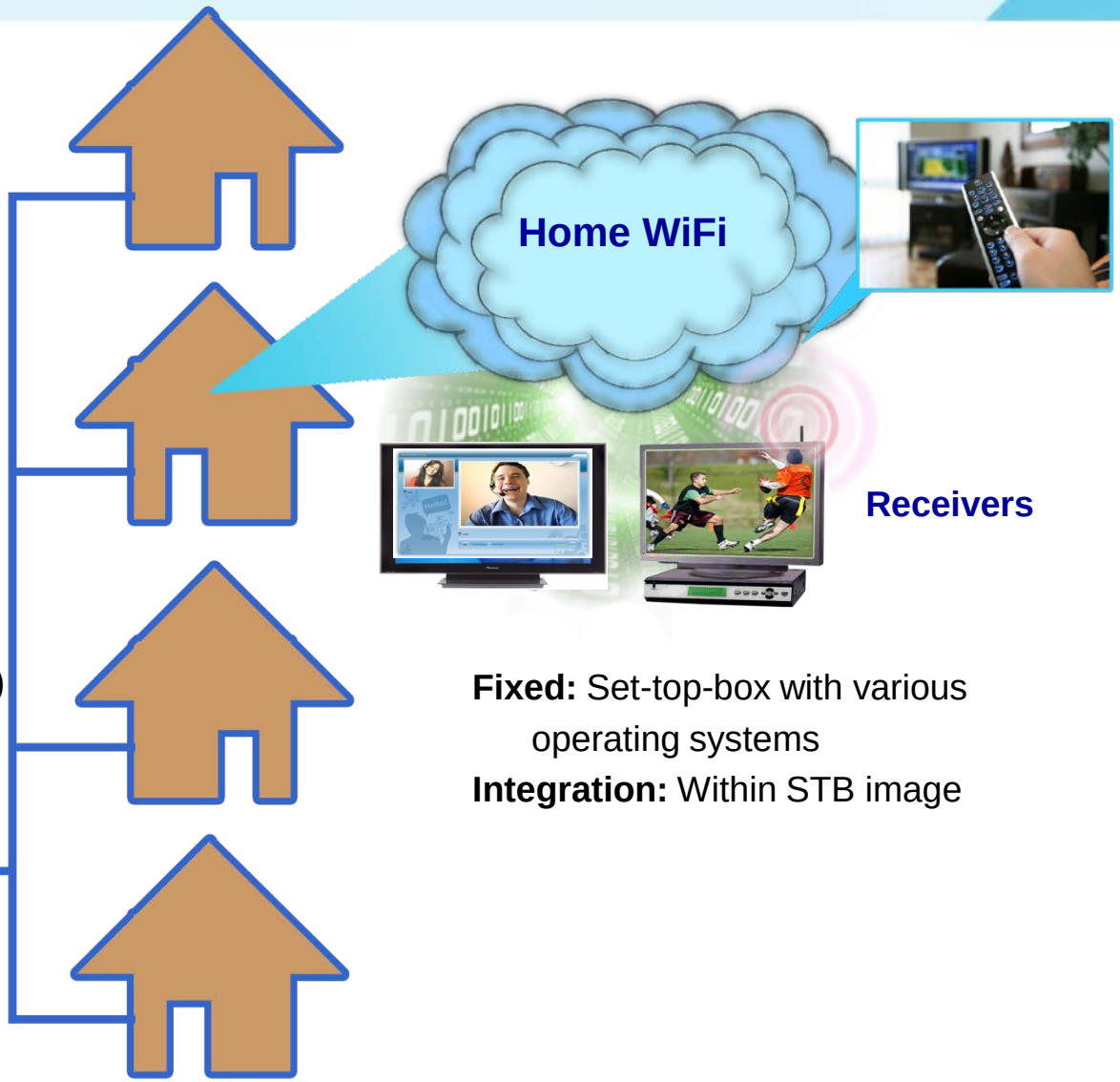
IP Set-Top-Box Streaming Video Service

Challenge: IPTV

- **Network loss disrupts video viewing**
- **Requires low latency with high QoS for chat**

Applications:

- Streaming Video-on-Demand
- Video Chat (Peer2Peer Duplex)



Fixed: Set-top-box with various operating systems

Integration: Within STB image

Raptor FEC Technology is Proven across a Broad Range of Solutions

- Proven in Content Delivery Services
  - Enterprises send large database updates and digital cinema files daily to hundreds of sites via satellite multi-cast network
  - Service Providers deliver VOD movies over various networks to subscribers
  - Cellular operators broadcast or multi-cast news flashes and video clips
  - Navigation service providers deliver map updates to fleets of vehicles
  - Military broadcasts image data via private mobile communications

- Proven in Streaming Solutions
  - Enterprises use multi-site HD video conferencing over public networks
  - TV stations broadcast mobile live news in HD formats
  - IPTV deployments in Asia and Europe on set-top-boxes, cable and Wi-Fi
  - Video surveillance streams near-real-time video to control centers
  - Military airborne operations stream live video feeds to ground troops
  - Ground sensor networks transmit data to central command centers

Our experienced engineering team understands embedded integration and low latency solutions

Products and Business Model

- **RaptorQ Platforms:**
  - Windows / Linux: x86 (32 bit and 64 bit)
  - Android / Linux: ARM
  - Apple IOS: ARM
  - Mac OS X: Intel
  - Other platforms can be ported for a fee

- **RaptorQ Software Development Kit (SDK):**
  - Encoder and Decoder library for target platform
  - Header file for library (same for all platforms)
  - Documentation: Developer's Guide, Release Notes, Installation instructions, README
  - Source code for file delivery and streaming sample applications
  - Test application code: SelfTest (sanity check), SysCheck (encode/decode and verify) and PerfTest (performance test)
  - Supporting files for sample code: makefile or project files (e.g. on Windows)

- **Licenses:**
  - Evaluation Kit (Free): sample file delivery and streaming applications and demos
  - Commercial: SDK, 6 month no obligation free development period

For more information on Raptor products or to request your free RaptorQ Evaluation Kit please go to www.qualcomm.com/raptor

For questions please email <raptorsupport@qti.qualcomm.com>

© 2012 Qualcomm Technologies, Inc. All rights reserved.