ACE: A Robust and Efficient IP/UDP/RTP Header Compression Scheme

March 31, 2000

Khiem Le, Christopher Clanton, Zhigang Liu, Haihong Zheng

Nokia Research Center - Dallas, USA

Outline

- Introduction
- Basic Concepts of ACE
- Details of ACE
- Compression Performance Results

Introduction

- Background: Foreseen introduction of Voice/multimedia over IP in future cellular systems is the primary driver for the need of an error-robust and overhead-efficient IP/UDP/RTP header compression scheme
- Two distinguishing characteristics of cellular links: scarce and expensive bandwidth and error-prone link
- RTP/UDP/IP packets contain relatively large headers (at least 40 bytes in IPv4 or 60 bytes in IPv6), and small payloads (about 20 bytes for voice).
- Spectrum efficiency requirement translates into header compression efficiency requirement; must meet existing baseline for voice
- Aim at generic framework that can be parameterized for deeper optimization in specific cases and technologies

NOKIA

Basic Concepts of ACE (1/2)

- ACE: Adaptive headeromprEssion
- Internet draft at **`http://38.197.106.103/draft-ace-robust-hc-01.txt`**.
- Compressor starts from no compression state and progressively transitions to higher compression states
- Controlled transition: compressor transitions to higher compression state only when it has enough confidence the decompressor has acquired the information to decompress in the higher state
- Confidence is achieved by e.g. acknowledgments from the decompressor
- Compressor states:
  - **FH**: **Compressor essentially sends full headers; operates in this state only at initialization or reinitialization; no compression state**
  - **FO**: **Compressor sends a sequence number + additional information; operates in this state if there is no string; higher compression state**
  - **SO**: **Compressor sends only a short sequence number; operates in this state only if there is a string; highest compression state**

NOKIA

Basic Concepts of ACE (2/2)

- In RFC2508, behavior is reactive
  - **When the decompressor detect a loss of synchronization between compressor and decompressor, it requests the compressor to send refresh information; problems with this approach include**
    - **Refresh information negatively affects compression efficiency, because it is a full header (40 or 60 bytes), or a large size header;**
    - **Surge in bandwidth demand, which is not well handled by some radio technologies**
    - **While waiting for the refresh, the decompressor has to discard all incoming compressed headers, even if they are not corrupted (error propagation caused by loss of synchronization)**
- ACE robustness is based on <u>proactive</u> behavior: Avoid loss of synchronization so don't have to recover from it
- The most robust way to avoid loss of synchronization is through proactive feedback (acknowledgments), but some systems may not have a reverse link

NOKIA

3 Modes in ACE

- **ACE can operate in 3 modes, depending on the environment**
- **Bidirectional Deterministic**
  - **A feedback channel is available and has predictible performance behavior**
- **Bidirectional Opportunistic**
  - **A feedback channel exists, but does not have predictible performance behavior**
- **Unidirectional**
  - **No feedback channel**
- **Conversational applications belong to mode 1 or 2**

ACE Assumptions

- **<u>General</u>**
- **Packets transferred on the forward channel can be lost and corrupted. *No particular pattern of packet errors is assumed*.**
- **Reasonably good error detection exists so corrupted information is not delivered; error detection is provided by the link.**
- **The order of packets is maintained between the compressor and decompressor, i.e., a receiver always receives the packets in the order they were sent by the sender.**
- **<u>When a feedback channel exists</u>**
- **No strict delay or error requirements on feedback channel. Acks can be lost or delayed. Delay and error characteristics can fluctuate over time.**
- **The order is maintained on the feedback channel**
- **Reasonably good error detection exists so corrupted information is not delivered**
- ***ACE does not assume any particular distribution of round trip time (RTT)* and can adapt dynamically to the change of RTT.**

NOKIA

Robustness of ACE

- Resilient to large number of packets lost
  - **Between compressor and decompressor**
  - **Before the compressor**
- Resilient to large scale packet misordering before the compressor
- Average overhead/compression efficiency is nearly constant, even for very high error rates
- Handovers are not a problem - possible to resume operation in most optimal state immediately after HO completion
- **<u>Bidirectional modes</u>**
- Extremely resilient to loss or delay of Acks, as they do not result in error propagation and do not affect the correctness of decompression

NOKIA

Variable Length Encoding, VLE (1/2)

- Compressor sends k least significant bits of original value
- k is smallest value to ensure correct decompression; k is chosen from a set of values, e.g. {k1, k2, k3}
- Robustness against errors is ensured by the compressor maintaining values of previous packets in window W = {p1, p2, p3, p4}; W is such that compressor knows that at least one packet in W was received.
  - Feedback channel: W consists of packets sent since and including the last acked packet
  - No feedback channel: W consists of last L packets
- Decompressor chooses as decompressed value the one that is closest to the reference and whose k LSBs match the received value; reference is last decompressed value
- Compressor chooses k so that no matter what value in W is chosen as reference by the decompressor, decompression is correct

Variable Length Encoding, VLE (2/2)

- Advantages of the scheme
  - Can handle case of positive jump (e.g. packet loss before compressor or IP-ID jump) or negative jump (e.g. packet misordering before compressor)
  - Dynamically adapts in real-time to magnitude of jump; no need to guess beforehand the packet loss and packet reordering statistics of the environment
  - Efficient, since k only grows logarithmically with the magnitude of the jump
  - Robustness against errors by means of window W
  - Has been implemented and proven to work
  - Broadly applicable: Compression of RTP SN, IP-ID and RTP TS

NOKIA

Timer-based (1/3)

- Used to compress RTP TS
- Exploits fact that
  - RTP TS is of the form TS0 + *n* * TS_stride ----> can use *n* (packed RTP TS) as a more concise form of RTP TS, without loss of information
  - RTP TS can be approximated by wall-clock ---> can use local clock at the decompressor to obtain an approximation; for voice, approximation error only caused by jitter between RTP source and decompressor
- Compressor sends k least significant bits of packed RTP TS
- For each packet, compressor estimates upper bound of jitter, and determines k as smallest value to ensure correct decompression, given the jitter; k is chosen from a set of values, e.g. {k1, k2, k3}

NOKIA

Timer-based (2/3)

- Robustness against errors is ensured by the compressor maintaining jitter information for previous packets in window W = {p1, p2, p3}; W is such that compressor knows that at least one packet in W was received.
  - Feedback channel: W consists of packets sent since and including the last acked packet
  - No feedback channel: W consists of last L packets
- Decompressor
  - Approximates the current packed RTP TS = packed RTP TS of reference packet + time elapsed since reference packet; reference packet is last received packet
  - Refines approximation by choosing the value closest to the approximation whose k LSBs match the received value;
- Compressor chooses k so that no matter what packet in W is chosen as reference by the decompressor, decompression is correct

NOKIA

Timer-based (3/3)

- Advantages of the scheme
  - Media such as voice have silence intervals which cause a jump in the RTP TS; with other approaches, size of compressed RTP TS depends on the RTP TS jump magnitude;
    - **assuming scheme based on sending LSB, a silence of 10 seconds would require 10 bits, even with packed RTP TS encoding**
  - Extremely well suited for conversational voice; size of compressed RTP TS is small and practically constant: 4 bits can handle a jitter of 320 msec
  - For non conversational, jitter could be higher: e.g. 8 bits can handle a jitter of 5 seconds
  - Robustness against errors by means of window W
  - Works even when RTP TS decreases, at lower efficiency (larger k required) ---> works for video, but efficiency compared to VLE needs to be determined
  - Requires only a low granularity timer (Time spacing between packets)
  - Has been implemented and proven to work

Handover (1/2)

- Handover is a procedure necessary in cellular systems, where the mobile terminal moves to a new radio cell
- Handover requires the mobile terminal to resynchronize to the new radio cell ---> temporary disruption of communications ----> Loss of multiple packets in case of real-time traffic. E.g. 100 msec disruption equates to loss of five 20 msec packets
- When the terminal moves out of the area served by a compressor/decompressor entity, compressor/decompressor function has to be relocated to another entity ---> Cause for additional disruption
- Disruption can be minimized by transferring compression/decompression context information from old compressor/decompressor entity to new entity
- Transfer of context information is not a trivial process, because it takes non-zero time, and the context may evolve in the meantime ---> New entity may have stale context information

NOKIA

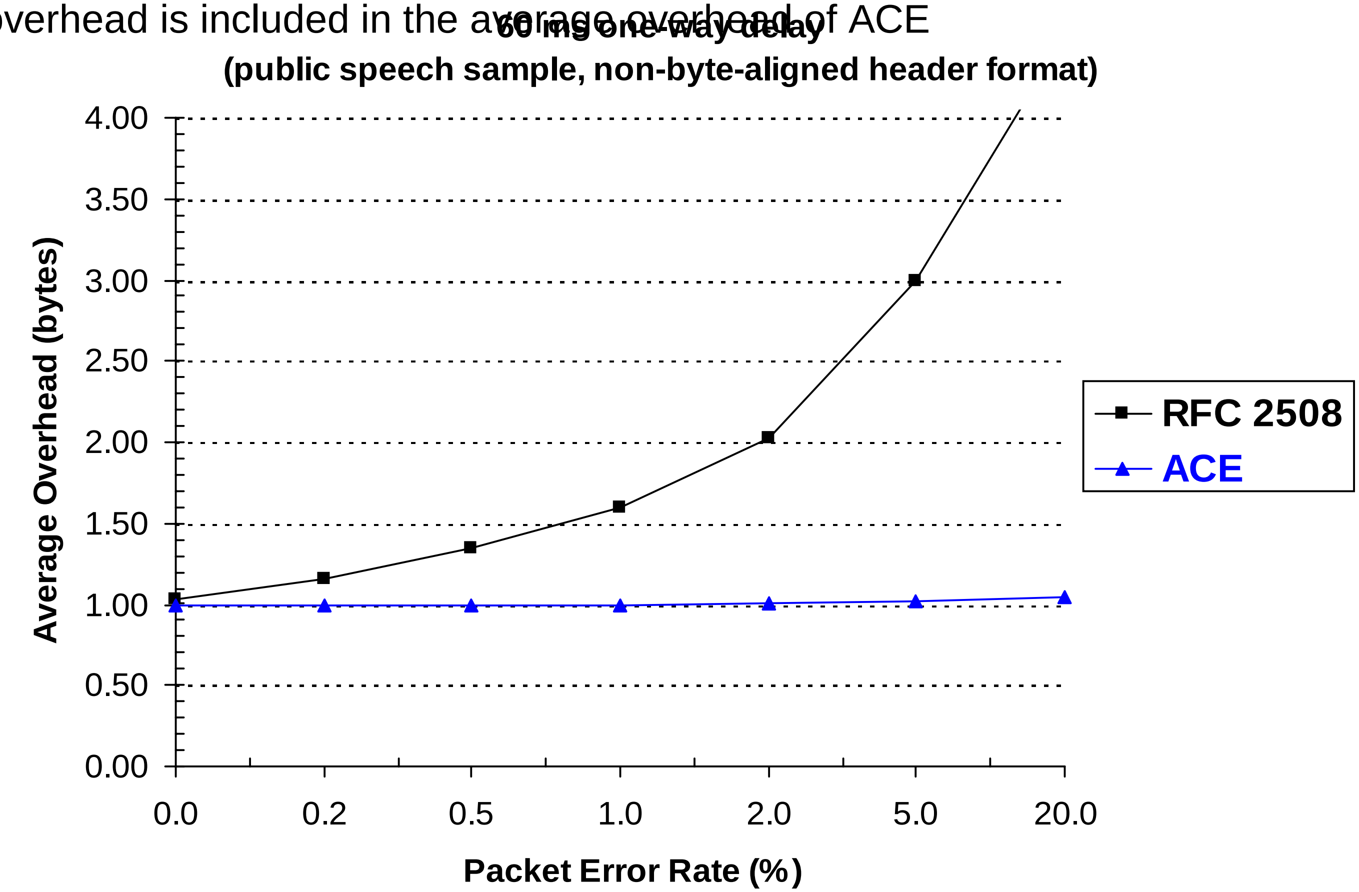
Handover (2/2)

- ACE meets all requirements for cellular handover
  - Resilient to multiple packet loss
  - Can seamlessly run across compressor/decompressor entity relocation; mobile terminal does not see any disruption caused by relocation

NOKIA

Compression Performance of ACE v.s. RFC2508

- Note: For fair comparison, CID (1 byte) is not counted in for both schemes. Random error model is used. Use VLE encoding for RTP-SN & IP-ID, timer-based scheme for TS.
- IP-ID non sequential
- Ack overhead is included in the average overhead of ACE



Conclusions

- ACE is very well suited for header compression in cellular environments
- Very high compression efficiency: average overhead at or sslightly above1 byte per packet for actual samples of speech and conversational voice, across a wide range of error rates
- Extreme robustness to
  - **Packet loss and misordering before the compressor**
  - **Packet loss between compressor and decompressor**
- Loss of synchronization avoided by controlled transition from lower compression state to higher compression state ---> No error propagation
- Can seamlessly handover, even when compression/decompression function is relocated from one network entity to another
  - **No degradation of compression efficiency, in particular no need to reinitialize with full headers**

