Enhancing Multimedia QoE via More Effective Time Synchronisation over 802.11 Networks

Jonathan Shannon
Padraig O’Flaithearta
Yusaf Cinar
Hugh Melvin

Disc. of Information Technology, National University of Ireland, Galway.
Outline

- Multimedia & Synchronisation
- Computer Clocks, Synchronisation Terminology & Techniques
- 802.11, NTP & Issues
- Simulations
- Delay Determination Technique
- Experiments & Results
- Questions
Growing Role of Synchronisation

- **VoIP** (Voice over IP)
Growing Role of Synchronisation

- **MMOG** (Massive Multiplayer Online Game)
WebRTC

- **WebRTC**
  - Open source project released by Google in 2011
  - Aim to equip browsers with RTC capabilities
  - W3C standardising ECMAscript API’s
  - IETF standardising underlying RTC protocols (Rtcweb WG)

- **WebRTC & VoIP**
  - Techniques to cope with variable packet latencies (Jitter buffer and play-out strategy algorithms)
  - WebRTC NetEQ component
  - NetEQ can benefit from packet delay information – improve QoS
  - Synchronisation of wireless nodes important
System Clock Operation

Oscillator

Holding Register

0 1 1 1 1 1 1

Clock Register

0 0 0 0 1 1 1 1

7 bits
1 tick = 32 μs

Main Memory

Interrupt Handler

System Time

Execute

CPU

Operating System

Request Time

Precision - Latency in reading the time
Coverted to UTC format

Application

Query

Increment (1 tick)

Tick interrupt (Register empty)

Resolution 32 μs

Time epoch: 01/01/1970

Quartz Crystal

Frequency 4 MHz
Resolution 0.25 μs
Synchronisation Terminology

- **Host & Reference**

- **Offset**  
  Time difference between a host time and a reference time  
  - Indication of clock accuracy  
  - Denoted by $\theta$

- **Skew**  
  Rate of change of host’s time with respect to reference’s time  
  - Influenced by oscillator precision/accuracy  
  - Denoted by $\lambda$

- **Drift**  
  Rate of change of host’s skew with respect to reference’s time  
  - Influenced by oscillator stability  
  - Denoted by $\varphi$
Sources of Synchronisation Error

- **Send Time**
  - Timestamp, **construct** message & send to **NIC** (system load, system call latencies)

- **Access Time**
  - Access communication medium
  - (MAC rules)

- **Propagation Time**
  - **Traverse link** between sender and receiver

- **Receive Time**
  - NIC **receive** & **decode** time and time interval before timestamp
Synchronisation Techniques

- **Uni-directional Synchronisation**
  - **Host** sets its time to the value received in a time message from a **reference**

- **Round-trip Synchronisation**
  - Two-way message exchange
  - Host obtains timestamps $T_i, T_{i+1}, T_{i+2}$ and $T_{i+3}$
  - Determines **round-trip delay** ($\delta$) and **offset** ($\theta$)
  - $\delta = (T_{i+3} - T_i) - (T_{i+2} - T_{i+1})$
  - $\theta = (T_{i+1} - T_i) + (T_{i+2} - T_{i+3})/2$
Synchronisation in WiFi Networks

- **Offset**
  - \( A = T_{i+1} - T_i \)
  - \( B = T_{i+3} - T_{i+2} \)
  - \( \theta = (T_{i+1} - T_i) - (T_{i+3} - T_{i+2})/2 \)

- **WiFi/802.11**
  - Access & buffer delays

- **NTP**
  - Degrade performance

\[ \theta = \frac{A - B}{2} \quad \epsilon = \frac{-\Delta}{2} \]
Up-link & Down-link Delays
Simulations – NS3
Simulations – NS3

Average Up-Link & Down-Link Delays

\[ \epsilon = (\text{uplink}) - (\text{downlink})/2 \]
Simulations – NS3

![Graph showing Time Error vs AP buffer size for 801.11a and 802.11b]
Solution
Up-link Delay ($\Delta u$) Determination
Down-link Delay ($\Delta d$) Estimation
Down-Link Delay Estimation

NTP Client

AP

Injected Data Packet

Packet Returned

\[ T_d - T_c = A + B \]
Experiment – Real Test-Bed

- How effective is it?

- **NTP client** sends 20 NTP packets per minute

- A & B transmit TCP packets to each other via AP

- Create load at AP - induce large **buffer delays**

- Duration 60 mins
Traffic

[Graph showing traffic over time with peaks and troughs indicating traffic generation intervals.]
Up-Link Delays
Down-Link Delays

![Graph showing Down-Link Delays with time (T0 to T2) and delay (ms) on the y-axis. The graph indicates a rise in delays during the Traffic Generation phase.](image-url)
Results – Offsets ($\theta$) & Errors ($\varepsilon$)

\[ \Delta_u \text{ up-link delay} \]
\[ \Delta_d \text{ down-link delay} \]
\[ \theta_T \text{ true offset} \]
\[ \theta_U \text{ un-corrected offset} \]
\[ \theta_C \text{ corrected offset} \]
\[ \varepsilon_U \text{ un-corrected offset error} \]
\[ \varepsilon_C \text{ corrected offset error} \]

\[ \theta_T = -0.000071(T_i) + 42.547466 \]
\[ \theta_U = \frac{(T_i - T_i) + (T_{i+2} - T_{i+3})}{2} \]
\[ \theta_C = \frac{(T_i - T_i) + (T_{i+2} - (T_{i+3} - \Delta_d + \Delta_u))}{2} \]

\[ \varepsilon_U = |(\theta_T - \theta_U)| \]
\[ \varepsilon_C = |(\theta_T - \theta_U)| \]
Error Distribution ($\varepsilon_U$) (un-corrected)
Error Distribution ($\varepsilon_c$) (corrected)
Outcome

- **Module/Technique** – reduces synchronisation errors in 802.11 networks

- Can be used on any host with
  - Protocol that uses **uni-directional** or **round-trip** synchronisation
  - NIC that supports **packet injection**

- Results indicate up to **90% reduction** in average offset errors

- Improve quality of dataset provided to time protocols
References


Questions