Enhancing Multimedia QoE via More Effective Time Synchronisation over 802.11 Networks

Jonathan Shannon
National University of Ireland Galway
University Rd
Galway City
Ireland
shannon.jonathan@gmail.com

Padraig O'Flaithearta
National University of Ireland Galway
University Rd
Galway City
Ireland
p.oflaithearta1@nuigalway.ie

Yusuf Cinar
National University of Ireland Galway
University Rd
Galway City
Ireland
cinar.yusuf@gmail.com

Hugh Melvin
National University of Ireland Galway
University Rd
Galway City
Ireland
hugh.melvin@nuigalway.ie

ABSTRACT

Time synchronisation plays a critical role in time-sensitive distributed applications. While a variety of such applications exist across many domains, one particular set of applications where improved time synchronisation can lead to significant benefits, particularly with respect to QoE (Quality of Experience), is multimedia applications. While time synchronisation is not a new challenge, advances in wireless technologies have drastically transformed network infrastructures. 802.11 wireless networks increasingly represent the last hop within the ever expanding Internet and whilst users expect the same levels of multimedia QoE as exist over wired networks, the reality of moving back to contention based access leaves many disappointed. This transformation of networks has also proven problematic for time synchronisation protocols that were designed for wired infrastructures. Wireless networks, particularly contention based networks, can be the source of very significant non-deterministic packet latencies. In certain scenarios, such latencies can greatly degrade the performance of time synchronisation. This work details and validates a technique that can be used to determine the latency of time messages in real-time as they traverse an 802.11 wireless link. Knowledge of these latencies can be used to greatly reduce the error in a dataset employed by time synchronisation protocols such as NTP and, thus, improve their performance. Experimental results confirm error reductions of up to 90% in a dataset and prove that the use of this technique can deliver time accuracies akin to those achievable over wired networks. This in turn can greatly benefit users by enabling multimedia applications to benefit from the continued use of time synchronisation for QoE management. We outline two such scenarios, one where time synchronisation is used to prioritise VoIP traffic within an Access Point and a second where the aim is to use time synchronisation to optimise jitter buffer strategies for WebRTC.

MMSys'16, May 10-13, 2016, Klagenfurt, Austria

 ISBN 978-1-4503-4297-1/16/05...$15.00

DOI: http://dx.doi.org/10.1145/2910017.2910615

CCS Concepts

•**Networks** → **Time synchronization protocols; Wireless local area networks;** •**Software and its engineering** → *Synchronization;* •**Information systems** → Multimedia streaming; Massively multiplayer online games;

Keywords

Time-synchronisation, Multimedia, QoE, Asymmetric Delays, 802.11, NTP

1. INTRODUCTION

The presentation of multiple forms of media in an integrated way has for decades captured the attention of the masses. Before the technological revolution that is the internet, the majority of multimedia applications took the form of crude standalone arcade games. This was followed by standalone applications that ran on the first generation of personal computers. The advent of packet-switched networking allowed for the distribution of media in a new and interesting way and this coupled with continuous advances in both integrated circuitry design and size, as well as programming tools and techniques has lead to the countless multimedia applications that exist in modern times. Today, such applications have become synonymous with the internet and represent the core, if not sole, means by which the majority of the world's population interact over networks.

Distributed multimedia applications such as VoIP (Voice over IP), MMOG (Massive Multilayer On-line Games) and IPTV (Internet Protocol Television) represent a sub-set of a category of applications that rely heavily on the large datarates offered by current network infrastructures. This reliance is due to the fact that they, for the most part, operate in real-time and, as such, their performance is highly dependent on network throughput and delay. The performance of these applications is generally measured in terms of the quality of experience (QoE) of the user base. Whilst there has been significant work in quantifying the relative contributions of system, human and context factors, much more needs to be done. However, regarding system factors, there exist formal metrics that are used to quantify this performance; an example being the Conversational Mean Opinion Score (MOS) which can be used to measure the quality of service (QoS) of a VoIP call [10]. Given that these applications (especially RealTime Communications such as VoIP and MMOG) distribute live content in real-time and must, therefore, adhere to one or more temporal constraints in order to meet the demands of one or more performance metrics, it is reasonable to conclude that distributed instances of these applications, particularly those that interact, can benefit from a common notion of time, one that they can agree is accurate to a particular degree. The application of time synchronisation techniques, thus, has the potential to improve the QoE of these applications as detailed in subsequent section.

In recent years a growing body of research has analysed and verified the benefits that so-called Time Awareness can bring to a broad range of application domains [4], [12], [11], [2]. Of particular interest is the TAACCS Interest GRoup (Time Aware Applications, Computers and Comm Systems, that has published a NIST white paper [27]. Whilst the multimedia community have always understood the importance of timing, the extent to which it can radically improve QoS/QoE for existing and emerging applications is not yet fully realised. In [14] the authors investigate opportunities for improving VoIP jitter buffer performance through implementation of synchronised time via the Network Time Protocol (NTP). This essentially optimises buffer play-out delays based on the ITU-T recommendation on one-way delays for VoIP. In [16] the authors present a Clock Skew detection and compensation technique for VoIP. This research developed a patented mechanism [17] for detecting skew via NTP/RTCP timestamps, so as to compensate for same whilst minimising QoS degradations. Using the technique presented in this paper the authors of [7] developed the concept of an intelligent Access Point (iAP), that deploys a Software Defined Networking approach (SDN) to manage QoS/QoE across multiple concurrent RTC sessions. Essentially, the intelligent controller builds up knowledge of real-time delays for each RTC session, maps this to QoS scores, and dynamically alters MAC session parameters so as to prioritise certain RTC streams over others. The authors of [29] present research that successfully synchronises Hybrid Broadcast and Broadband Media streams (HBB). It focuses on live or near-live streaming media, typically sports events. The application scenario implemented in a prototype represents a use-case where an end-user can select media streams that are logically and temporally related e.g. a video stream of a live football match, and a separate audio stream of the same football match from an Internet radio feed, and have them integrated into a single media stream. In this case, synchronized time is used to precisely align the streams before integration and also to detect and compensate for clock skew. This differs from the previous two case studies where synchronised time is used to precisely measure end-to-end delays so as to better manage QoS. This latter research also examines the evolution of HTTP Adaptive Streaming (HAS), and the timing challenges and opportunities for near-live streaming.

It is clear from the aforementioned research that time synchronisation can, through differing approaches, enhance the QoE of various types of multimedia applications. In this respect it is important that the time synchronisation techniques employed by such applications provide the desired level of synchronisation to realise such benefits. The proliferation of wireless network in the last decade has resulted in the migration of applications from the wired to the wireless domain. While this has been extremely beneficial to users, by facilitating realtime multimedia consumption across WiFi, in the context of time synchronisation it has introduced new and interesting challenges. Wireless network protocols such as WiFi (IEEE 802.11) operate quite differently from wired networks such as switched Fast Ethernet (IEEE 802.3) that are essentially deterministic. This operational difference can under particular circumstances lead to a degradation in the performance of traditional time synchronisation protocols and, thus, their utilisation to enhance the QoE of multimedia applications over such networks can in fact cause more problems than good.

This research presents a technique that remedies huge challenges that WiFi networks present to time synchronisation protocols, with a particular focus on the Network Time Protocol (NTP). Section 2 presents some background information on time synchronisation protocols and highlights the issue with delivering precise time over 802.11 wireless networks. Section 3 presents a technique than can be used to remedy the issue. Section 4 presents the experiments that were performed to validate the technique and section 5 presents the results of the experiments. Section 6 details two application scenarios, where this much improved time synchronisation technique can yield benefits. The first prioritises VoIP traffic within an Access Point using an SDN-like approach whereas the second builds on previous research that uses time synchronisation to optimise jitter buffer strategies. In particular it outlines a proposed hybrid approach , currently being deployed for WebRTC that will combine existing packet scaling with fixed playout delays. Finally section 7 presents the conclusions and their relevance to RTC multimedia applications.

2. BACKGROUND

The system clock of a general purpose computer system plays a vital role in the proper functioning of the system as a whole. Computing software often requires a notion of time for such tasks as process scheduling, utilisation tracking, event generation, time-stamping, and other application dependent tasks. When two or more computing systems must interact and their interaction requires and/or can benefit from high levels of time synchronisation, factors such as clock stability, accuracy and granularity become very important. Distributed real-time applications necessitate a common notion of time amongst their distributed hosts. These hosts must incorporate techniques that ensure their real time clocks adhere to a common time-scale. Time synchronisation techniques and protocols fulfil this role.

Time synchronisation protocols such as the Network Time Protocol (NTP) [18, 19, 20] and the Precision Time Protocol (PTP) [9, 6, 5] prove effective synchronisation solutions particularly over wired links. Both employ the *round-trip synchronisation [24]* technique illustrated in fig. 1 which presumes largely symmetric delays over networks. In most cases, wired infrastructures provide the necessary resources required to accommodate large traffic loads and, thus, time protocol messages that traverse such networks are not subjected to significant asymmetric latencies. In comparison, wireless networks such as those based on the 802.11 standard [1, 8], have relatively low bandwidth and operate in a contention based manner. As a result they can subject time protocol messages to relatively large non-deterministic and asymmetric latencies [13] as illustrated in fig 2. As such the protocol's performance may be seriously degraded limiting its role as a time synchronisation solution for numerous time-sensitive applications. 802.11-based networks are increasingly providing last hop connectivity, and this trend is likely to accelerate as low power chipsets evolve.

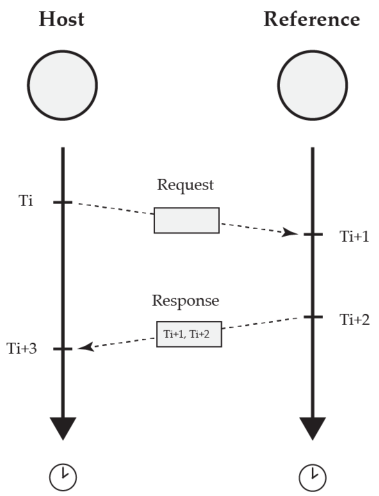


Figure 1: Round-trip synchronisation

Recent efforts to resolve the time synchronisation issues observed in WLANs include 802.11v and 802.1AS. The former incorporates the latter to deliver tight time synchronisation over wireless links in bridged LANs. 802.11v is an amendment to the 802.11 standard that specifies timing service primitives that allow 802.11 stations to determine their time offset from their associated AP's time. In the context of time synchronisation with respect to a time standard, this mechanism only proves useful in a scenario where the AP itself is synchronised to a time standard by some other means. 802.1AS can provide synchronisation with respect to a time standard within a WLAN by realising this. It incorporates a generic form of IEEE 1588 termed the general Precision Time Protocol (gPTP) which operates on an AP itself thus allowing the AP to synchronise to its designated master while providing a recognised time scale to associated wireless stations via 802.11v. The current concern regarding

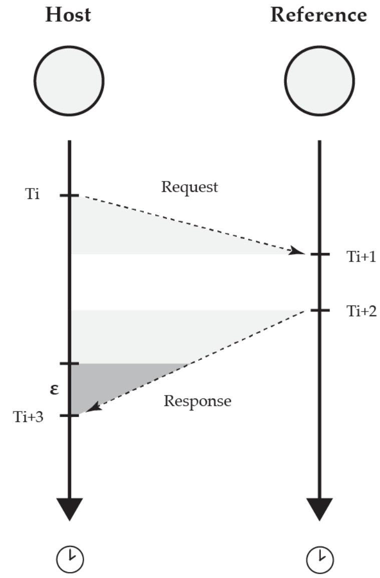


Figure 2: Asymmetric two-way latencies

the incorporation of 802.1AS into current network infrastructures is one related to cost. The expense of upgrading all of the wireless nodes within a WLAN may not justify the resulting benefits of tight time synchronisation except in certain domains such as industrial control or broadcasting environments. It is in this regard that a much more cost effective means of providing time synchronisation accuracies over WLANs akin to those currently achievable over wired LANs is potentially valuable.

This work proposes a practical, low-overhead, and transparent cross-layer technique that can be used to quantify and remove asymmetries, thus enabling legacy time protocols to provide the level of synchronisation typical of wired networks. The mechanism leverages the facilities provided by many modern wireless network interface cards (WNICs) and exploits the operation of 802.11 networks in order to determine/estimate the delays encountered by time protocol messages as they traverse an 802.11 link. The entire practical approach provides a realistic alternative to approaches that model network delays or employ complex signal processing techniques as summarised in [28]. It instead provides a solution that operates effectively regardless of the traffic profile.

3. DELAY DETERMINATION MECHANISM

The technique entails actively determining and compensating for, the up-link (Δ_u) and down-link (Δ_d) delays of time messages as they traverses the link between a client and its associated AP and vice versa. The procedure is discussed with respect to the operation of the NTP protocol.

To begin, it is necessary to determine the up-link delay of an NTP *request* message which represents the time interval between when it is generated by the NTP client process and when it gains access to the wireless link. To do this we exploit the 802.11 protocol which dictates that the transmission of a unicast data frame and the reception of its associated *acknowledgement* frame (ACK) is an autonomous

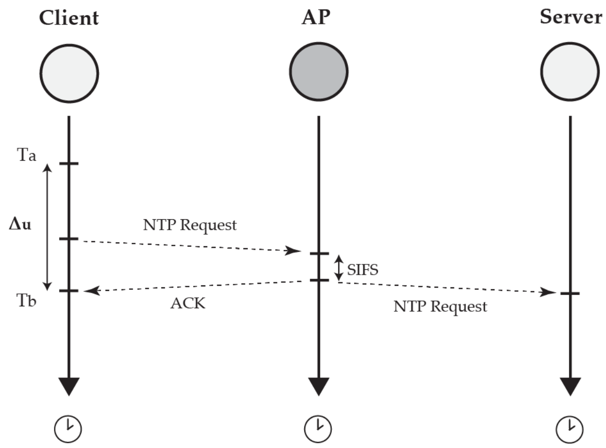


Figure 3: Up-link delay determination

operation [1]. Thus, every NTP *request* message will have an associated ACK frame. By recording the reception time of this ACK frame one can estimate the point in time that a time message gained access to the medium. In this case relatively minor propagation delays (typically nanoseconds-microseconds) associated with the *request* message and the ACK frame as well as the delays associated with the *short interframe space (SIFS)* are ignored. In application scenarios such as those described in [7], this is acceptable in that such delays are relatively negligible. The approach is illustrated in fig. 3 (note fig. 3 is not to scale) where T_a represents the timestamp recorded by the NTP client process before it transmits an NTP *request* message while T_b represents the timestamp recorded by the WNIC on reception of the *request* message's associated 802.11 ACK frame. The difference between T_a and T_b approximates the up-link delay (Δ_u) of the *request* message as shown in equation 1.

$$\Delta_u \approx T_b - T_a \tag{1}$$

In many 802.11 deployments however, the down-link delay will most likely dominate (e.g. Web traffic) and it is the relative difference between the down-link delay and up-link delay that is important as large differences will degrade the quality of the NTP dataset. Determining the down-link delay involves a more active approach since the AP rather than the client is the source of these delays. One must estimate the duration that an NTP response packet, en route to an NTP client from a time server, spends within the buffer of an AP. To do this, we leverage the *injection* facility provided by many modern WNICs. This involves the construction of a specific 802.11 data frame that when injected into the network will be returned to the client by its associated AP. This particular data frame has its *destination address* field set to the contents of its *source address* field. As a result, the AP, on receipt of the data frame, will append it to its outgoing transmission buffer and eventually transmit it back to the sender. The overall operation is illustrated in fig. 4.

Referring to fig. 4, on receipt of an NTP *response* packet, the client immediately injects a *'crafted'* packet into the network and records the reception time, T_c, of its associated ACK frame. The packet is eventually returned by the AP

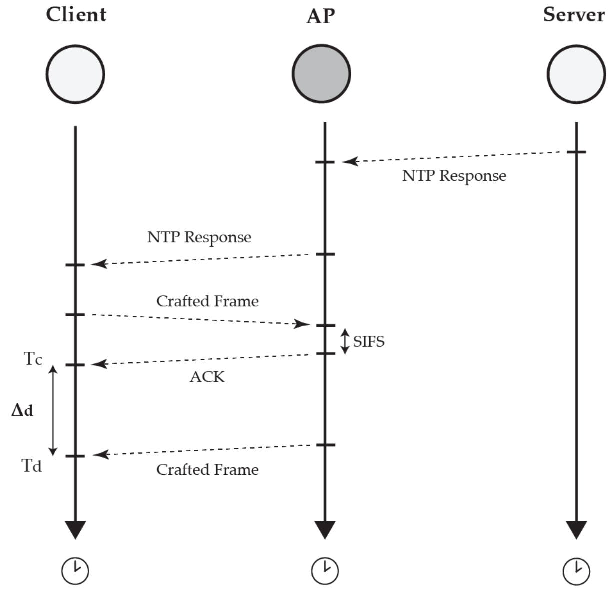


Figure 4: Down-link delay determination

and its reception, T_d, time recorded. The difference between these two timestamps will provide an estimate of the magnitude of the delay the NTP response packet was subjected to within the AP, that is, the down-link delay (Δ_d) as shown in equation 2.

$$\Delta_d \approx T_d - T_c \tag{2}$$

Equation 3 presents the formula used by NTP to calculate the offset of its host. This offset estimate is represented by the term θ_U and referred to here as the *un-corrected offset*. The terms T_i and T_{i+1} represent the transmission and reception times of an NTP *request* message at the client/sender and server/receiver respectively. The terms T_{i+2} and T_{i+3} represent the transmission and reception times of an NTP *response* message at the server/sender and client/receiver respectively.

$$\theta_U = \frac{(T_{i+1} - T_i) + (T_{i+2} - T_{i+3})}{2} \tag{3}$$

Equations 4 and 5 presents timestamps T_i and T_{i+3} in terms of their corresponding corrected timestamps components, $T_{i(C)}$ and $T_{i+3(C)}$, and their up-link and down-link delay components.

$$T_i = T_{i(C)} - \Delta_u \tag{4}$$

$$T_{i+3} = T_{i+3(C)} + \Delta_d \tag{5}$$

Plugging equations 4 and 5 into equation 3 yields the *corrected* or *modified offset*, θ_C as presented in equation 6.

$$\theta_C = \frac{(T_{i+1} - T_i) + (T_{i+2} - (T_{i+3} - \Delta_d + \Delta_u))}{2} \tag{6}$$

4. EXPERIMENTS

The effectiveness of the technique was evaluated in a real-world test-bed. The test-bed was designed so that time messages were subjected to large down-link delays relative to corresponding up-link delays. The reasoning for this was twofold. Firstly, in situations with high traffic loads, down-link delays will typically dominate, and secondly, as highlighted in the previous section, determining the down-link delay is far more challenging than determining the up-link delay and therefore better highlights the technique's effectiveness. Accordingly, the test-bed in fig. 5 was employed.

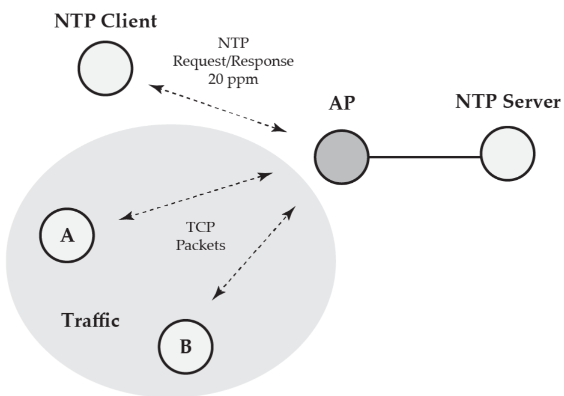


Figure 5: Experimental test-bed

The test-bed consisted of an NTP server connected to an AP via a 100Mbps wired link. A wireless NTP client, associated with the AP, polled the NTP server 20 times a minute. The NTP server responded to each client's NTP *request* message with a corresponding NTP *response* message. To induce large down-link delays two further wireless devices acting as traffic generators were introduced. Device A opened up a TCP connection to device B and transmitted raw TCP packets to B at a rate of 1500 packets per second. Device B responded to each received TCP packet with a similar TCP packet destined for device A.

All devices, excluding the AP, employed the *Ubuntu* distribution of Linux. Each wireless device contained an 802.11g compliant wireless NIC and operated at a data rate of 54Mbps. The NTP client's NIC contained an *atheros* chipset which together with the *ath5k* driver and *libpcap* library permitted packet *sniffing* and *injection*. The packet generator *hping3* was used to generate traffic. The interval between TCP packet transmissions was carefully chosen via trial and error so as to ensure the AP's buffer could cope with the traffic load. The experiment was run for a duration of 1 hour. Traffic was not introduced until the 16th minute. The traffic pattern produced by the traffic generator, *hping3*, is illustrated in Fig 6.

In order to be able to quantify the improvement gains of the technique it was necessary to track the offset of the NTP client from the NTP server between times $T0$ and $T1$ (see fig. 7) which represent the 1st and 16th minute respectively. Measurements taken within this interval allowed the *clock skew* of the client relative to the server to be determined. Knowledge of the skew allowed the offset between the client

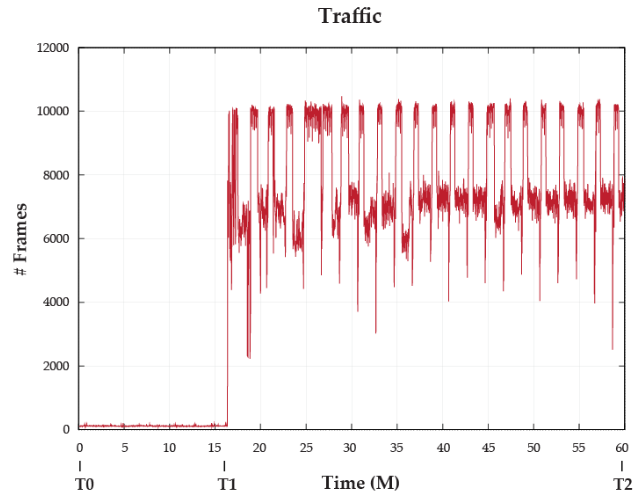


Figure 6: Number of frames detected by NTP client's NIC

and server to be determined at any point in time. Naturally, it was presumed that the skew did not change during this interval which is a valid assumption except under extreme temperature changing environments. A change in clock skew is referred to as *clock drift* and is more of an issue in *wireless sensor networks (WSN)* that are deployed in unstable temperature environments as detailed in [26]. In this experiment, the ambient temperature of the client and server remained constant over the short timeframe and, thus, it is reasonable to assume that the skew remained constant. The calculated offset based on the clock skew is referred to as the true offset and denoted by θ_T.

It is important to note that although the NTP protocol was employed in the experiments, the NTP algorithms were disabled and, thus, the offset of the client and server changed linearly. Once a value for the true offset (θ_T) was calculated, traffic was introduced into the 802.11 wireless network and persisted for the remainder of the experiment. The performance of the technique was subsequently evaluated by analysing the measurements obtained for the true offset, the corrected offset (θ_C) (using the technique) and raw un-corrected offset (θ_U) for the duration of the experiment.

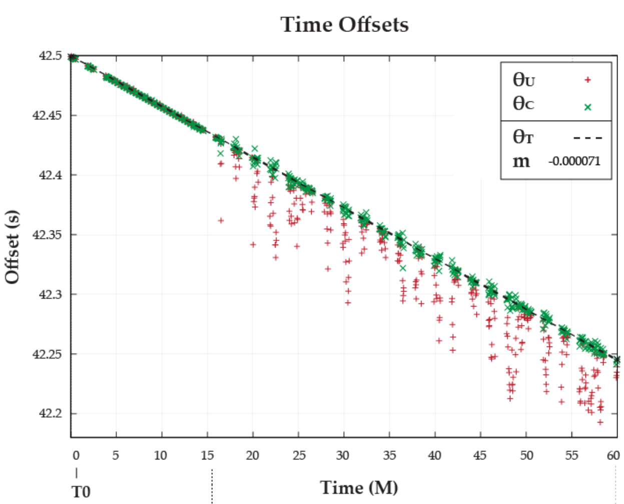
5. RESULTS

The absolute difference between the true offset, θ_T, and the corrected offset, θ_C, is represented by ϵ_C and presented in equation 7. ϵ_C can be compared to the error that results from not using the technique which is denoted by ϵ_U. The error ϵ_U, as shown in equation 8, is the absolute difference between the true offset, θ_T, and the un-corrected offset, θ_U.

$$\epsilon_C = |(\theta_T - \theta_C)| \tag{7}$$

$$\epsilon_U = |(\theta_T - \theta_U)| \tag{8}$$

The NTP client offsets (θ_U, θ_C and θ_T) obtained from the experimental data are presented in fig. 7. It indicates that the client's true offset, θ_T, from the server, as represented by the broken line, changes linearly during the course of the experiment at a rate of 6.13 seconds per day relative to the server which translates to a frequency error of 71 ppm, a value not atypical of computer clocks. Fig. 7 clearly illustrates the improvement gains that the technique provides. The determination of the up-link and down-link delays significantly improves the accuracy of the calculated offsets. On average, the error produced with the technique, ϵ_C, is 90% less than the error produced without the technique, ϵ_U, as the mean (μ) error is reduced from 13.6 ms to 1.5 ms. In addition, the maximum error is reduced from 82.5 ms to 23 ms and the standard deviation (σ) of the errors is reduced from 18.7 ms to 2 ms a 60% and 90% reduction respectively. The distribution of the errors with and without the use of the technique are illustrated in fig 8 and fig 9 respectively. These results suggest that if the technique described in this work is used in conjunction with the NTP protocol, then it is possible to achieve time accuracies similar to those achievable over wired networks. It is demonstrated in [19] that NTP can achieve sub-millisecond to single millisecond synchronisation over a typical LAN and less than a few tens of milliseconds over the global Internet.



ε (ms)	μ	MAX	σ
εC	1.5	23.04	2.06
εU	13.62	82.56	18.76

Figure 7: Calculated offset with and without the module

6. PRACTICAL APPLICATIONS

6.1 Network Delay Optimisation for QoS

As mentioned in section 1, the authors in[7] and [21] calculated each way delays for multiple VoIP calls over WiFi which have varying delays between endpoints. A dual approach of simulation and real-world experimentation is undertaken. The former is used to validate the core idea whereas the latter is used principally to assess the technical feasibility of the approach. As part of an SDN based proof-of-concept

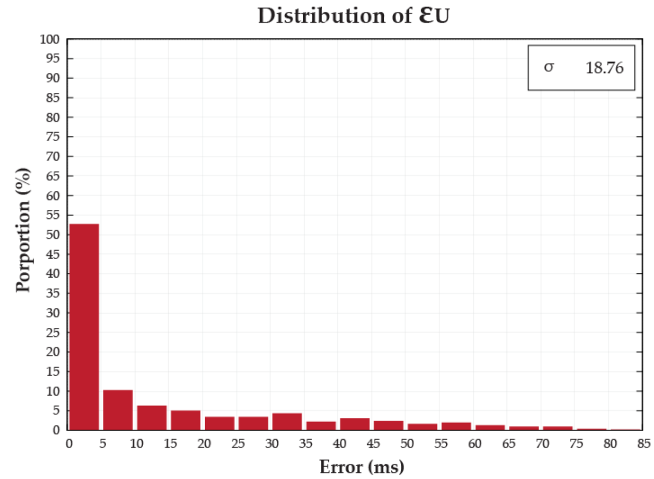


Figure 8: Error distribution without the module

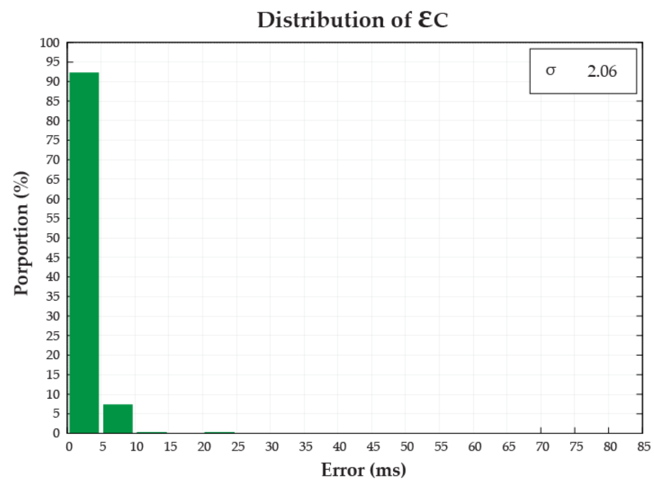


Figure 9: Error distribution with the module

(PoC) test-bed (Fig. 10), an intelligent Access Point (iAP - Fig. 11) is presented which integrates a number of key features that operate in real-time.

The iAP consists of a number of modules that assess the QoS of all VoIP sessions in a network in real-time and then use this information to optimize QoS/QoE. The calculation of accurate one-way delays requires the clocks of all end nodes in the test-bed being synchronized with millisecond accuracy. The iAP (illustrated in Fig. 10) dynamically and in real-time performs the following at fixed intervals:

1. Identify individual VoIP sessions
2. Calculate one-way delay for each session, including wired/wireless components of the de-lay, termed intra-one-way delay
3. Calculate each-way R-values (quality perceived by the end user estimated by the E-model [8] for each session using one-way delays from step 2).
4. Run a prioritization algorithm for VoIP sessions (based on R-values)

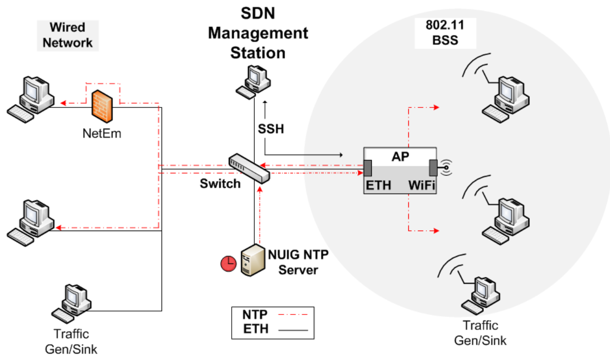


Figure 10: Proof-of-concept test-bed

5. Implement prioritization on AP downlink amongst sessions via modification of the IP DSCP field which is interpreted appropriately at the MAC layer. Note that the mechanism operates on the downlink only as much research has highlighted the so-called AP bottle-neck as the main source of delay/loss.

These actions can be controlled from a remote management device as shown in Fig. 10 where a remote station controls the iAP via SSH. As data packets enter the AP, a packet capture application scans for RTCP SR/RR packets. When RTCP packets are found, the AP timestamp is noted and they are passed up to a Delay Calculation module which identifies unique VoIP sessions and calculates the intra-one-way, and one-way delays for each session. The delay is calculated for both directions, namely uplink and downlink separately. These delay values are then passed to an E-Model based QoE module to generate an R-factor score for each session. The R-factor value derived for each session is then passed to the Prioritization module which runs a prioritization algorithm to decide which, if any, VoIP sessions should be prioritized.

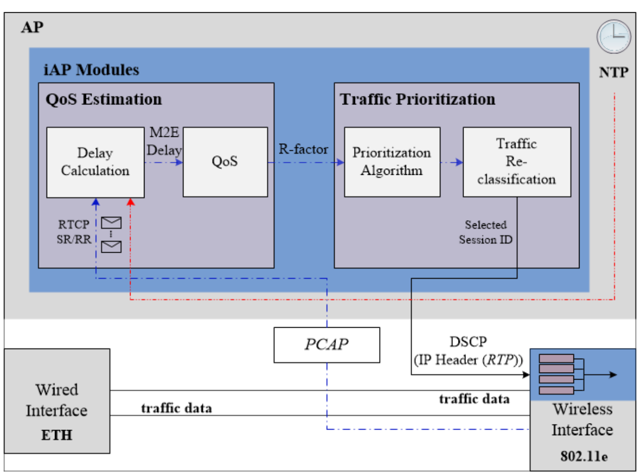


Figure 11: iAP Architecture

If a VoIP session is to be prioritized, the assigned ID of the session is passed to the Traffic Re-classification module which modifies (or mangles) the DSCP value in the IP header of VoIP data packets. This ensures that when these packets arrive at the MAC layer they are redirected appropriately to prioritize the session in question. In order to implement prioritization for concurrent VoIP sessions, a three-tiered categorization mechanism (Fig. 12) is used in the PoC whereby all sessions initially reside in the lowest category, and thereafter, certain sessions may be promoted (prioritized) to a higher priority category or demoted (deprioritized) where possible, based on their R-factor score. A background traffic category also exists in order to cater for non-VoIP traffic. The three Multimedia Categories are termed MC1, MC2, and MC3, where MC1 has the highest priority and MC3 has the lowest priority, along with a background access category (BK). All VoIP traffic resides within the three MCs. These four categories were chosen so that existing IEEE 802.11e equipment could be used in the PoC as IEEE 802.11e implements a four category system.

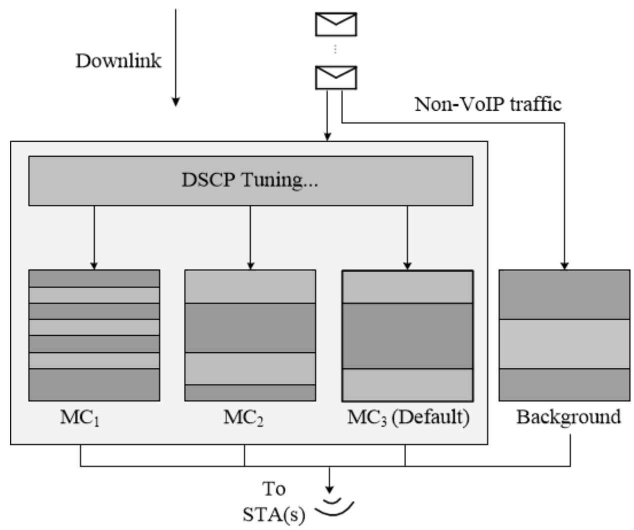


Figure 12: Multimedia Categorization System

When a VoIP session is chosen for prioritization or deprioritization, the iAP needs to implement this. The design approach utilizes the Netfilter framework to modify the Diffserv Code Point (DSCP) value in the IP header of all downlink packets (RTP) belonging to the session. A mapping exists between DSCP values and the IEEE 802.11e MAC traffic categories, and this process is utilized in the design to implement re-classification between MC1 - MC3. This has the effect of ensuring that all voice packets for a session will follow the categorization framework described in the previous section. Results show that the approach is both valid and feasible. The approach is also very much aligned with the emerging SDN paradigm described earlier. In the above example the iAP acts as a centralized controller that monitors all RTC traffic, and can dynamically optimize QoS for each RTC session in real-time.

However, we envisage that it can be extended to more complex mesh WiFi net-works and is scalable. As illustrated in Fig. 10, a remote management station can also be used to control the iAP. Whilst Fig. 11 realizes the intelligence within the AP, we also experimented with offloading the intelligence to the remote management station. In this approach, the delay calculation, QoS estimator and prioritization modules are remote and we simply pass down the prioritization decisions for each session ID to the AP which becomes a slave AP as illustrated in Fig. 10. Such a system could then be extended whereby the management station becomes an SDN controller for a larger scale mesh network. In this context, we have also examined scalability issues on AP hardware. In particular we have examined the extent of mangling that can be significant for APs that manage multiple concurrent RTC flows. Analysis shows that the processing latencies are negligible using conventional hardware, relative to the Mouth to Ear (M2E) latencies being considered. Furthermore we examined the scalability of the delay calculation mechanism using RTCP. Essentially, the sniffing and analysis of RTCP traffic is undertaken first/last hop of our small scale network (a single WiFi AP). However the mechanism described may operate on a full mesh network as long as the following two conditions are met; 1 - all session endpoints and intermediate iAPs are synchronised, and 2 - the session traffic is transported using RTP along with RTCP control.

6.2 Jitter Buffer Optimisation for QoS

In VoIP applications, the presence of network jitter arising from best-effort Internet protocol is compensated for via jitter buffer strategies that seek to maintain intra-media synchronization whilst also optimising both conversational and listening quality. Significant research [14] [23] [25] over recent years has yielded much insight into the challenges involved. One key differentiating factor is whether they adjust jitter buffer playout delay via the silence periods in a talkspurt or by applying time-scaling on a per packet basis. Both techniques have their advantages and disadvantages as described in [22] but a key aim is to optimise inherent speech listening quality whilst also minimising delay, and thus conversational quality. Regarding the latter, one key metric that is often not available is precise one way delays.

In the absence of precise time synchronisation, some approximation techniques, such as round trip time, are used to deduce a one-way transmission delay. However, due to asymmetries, this estimate can often be quite misleading leading to poor strategic decisions.

As shown in earlier research using the ITU-T E-model [15], precise calculation of a packet's transmission delay, which can be achieved with time synchronisation between the sender and receiver, can significantly improve overall VoIP quality by eliminating unnecessary late packet loss. In our proposed research, we build on this approach using WebRTC (webrtc.org), an open-source project for real time communications standardisation. By default, WebRTC uses a time-scaling technique to combat jitter [3]. Our hypothesis is that the NetEq algorithm within WebRTC, which controls the jitter buffer and playout strategy, can benefit from precise delay information and thus make more informed decisions regarding scaling versus silence period adjustments. We aim to use a hybrid QoS model that draws from the existing research on both of these strategies to ensure that a more satisfactory quality can be accomplished.

7. CONCLUSION

High precision time synchronisation can currently be used to enhance various categories of multimedia applications, particularity in terms of QoE. As the precision offered by time synchronisation techniques increases, it can spark new innovative solutions that can further benefit such applications as well as add other classes of multimedia applications to the current group of benefactors. However, as the proliferation of 802.11 networks continues, especially in the context of the aforementioned applications, poor time synchronisation will greatly hamper this progress. This applied research has proposed and validated a cross-layer technique that greatly improves the performance of time synchronisation protocols that operate in 802.11 environments. A detailed description of the technique has been presented along with experimental results that verify its effectiveness. The results indicate that its use can lead to a very significant (up to 90% in a test case) improvement in the quality of data presented to a time protocol. It is the authors' opinion that this research can go a long way towards mitigating those wireless related issues that negate the obvious benefits provided by time synchronisation techniques for multimedia applications as they continue to be migrated to the wireless domain.

8. REFERENCES

[1] Ieee draft standard for local and metropolitan area networks - specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications - amendment 3: Enhancements for very high throughput in the 60 ghz band. *IEEE P802.11ad/D8.0, May 2012 (Draft Amendment based on IEEE 802.11-2012)*, pages 1–667, 2012.

[2] I. M. Arntzen, N. T. Borch, F. Daoust, and D. Hazael-Massieux. Multi-device linear composition on the web, enabling multi-device linear media with htmltimingobject and shared motion. MediaSync2015 Workshop, 2015.

[3] Y. Cinar, H. Melvin, and P. Pocta. A black-box analysis of the extent of time-scale modification introduced by webrtc adaptive jitter buffer and its impact on listening speech quality. *Special Issue of Communications journal (Scopus) on Telecommunications Beyond*, 2016.

[4] M. Climent, A. Jansen, P. Cesar Garcia, F. Boronat, and H. Stokking. Review of media sync reference models: Advances and open issues. MediaSync2015 Workshop, 2015.

[5] J. C. Eidson. *Measurement, Control, and Communication Using IEEE 1588 (Advances in Industrial Control)*. Springer, 2006.

[6] J. C. Eidson and K. Lee. Ieee 1588 standard for a precision clock synchronization protocol for networked measurement and control systems. In *Sensors for Industry Conference, 2002. 2nd ISA/IEEE*, pages 98–105. IEEE, 2002.

[7] P. O. Flaithearta, H. Melvin, and P. Pocta. Time awareness in software defined networking. *European Conference on Networks and Communications*, 2015.

[8] M. Gast. *802.11 wireless networks: the definitive guide*. O'Reilly Media, Inc., 2002.

[9] IEEE. Standard for a precision clock synchronization protocol for networked measurement and control systems. *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pages c1–269, 2008.

[10] O. ITU. Series p: Telephone transmission quality, telephone installations, local line networks, methods for objective and subjective assessment of quality.

[11] A. Jesina, J. Martinez, R. Nachtrab-Ribback, M. Nalis, G. Linder, and P. Romero-Fresco. New approaches to improving the quality of media accessibility services on hbbtv. MediaSync2015 Workshop, 2015.

[12] J. Le Feuvre, V.-T.-T. NGuyen, W. Hammidouche, P. Marchal, R. Monnier, and P. Dupain. A test bed for hybrid broadcast broadband services. MediaSync2015 Workshop, 2015.

[13] T. Li, D. Leith, and D. Malone. Buffer sizing for 802.11-based networks. *IEEE/ACM Transactions on Networking (TON)*, 19(1):156–169, Feb 2011.

[14] H. Melvin and L. Murphy. Time synchronization for voip quality of service. *Internet Computing, IEEE*, 6(3):57–63, 2002.

[15] H. Melvin and L. Murphy. An evaluation of the potential of synchronized time to improve voice over ip quality. In *Communications, 2003. ICC'03. IEEE International Conference on*, volume 3, pages 1922–1926. IEEE, 2003.

[16] H. Melvin and L. Murphy. An integrated ntp-rtcp solution to audio skew detection and compensation for voip applications. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 2, pages II–537. IEEE, 2003.

[17] H. Melvin and L. Murphy. System and method for determining clock skew in a packet-based telephony session, Jan. 20 2005. WO Patent App. PCT/EP2004/006,808.

[18] D. L. Mills. Internet time synchronization: the network time protocol. *Communications, IEEE Transactions on*, 39(10):1482–1493, Oct 1991.

[19] D. L. Mills. Improved algorithms for synchronizing computer network clocks. *IEEE/ACM Transactions on Networking (TON)*, 3(3):245–254, 1995.

[20] D. L. Mills. *Computer Network Time Synchronization: The Network Time Protocol (1st Edition)*. CRC Press, 2006.

[21] P. O Flaithearta, H. Melvin, and M. Schukat. A qos enabled multimedia wifi access point. *International Journal of Network Management*, 25(4):205–222, 2015.

[22] P. Počta, H. Melvin, and A. Hines. An analysis of the impact of playout delay adjustments introduced by voip jitter buffers on listening speech quality. *Acta Acustica united with Acustica*, 101(3):616–631, 2015.

[23] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne. Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *INFOCOM'94. Networking for Global Communications., 13th Proceedings IEEE*, pages 680–688. IEEE, 1994.

[24] K. Römer, P. Blum, and L. Meier. Time synchronization and calibration in wireless sensor networks. *Handbook of Sensor Networks: Algorithms and Architectures*, pages 199–237, 2005.

[25] J. Rosenberg, L. Qiu, and H. Schulzrinne. Integrating packet fec into adaptive voice playout buffer algorithms on the internet. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1705–1714. IEEE, 2000.

[26] T. Schmid, Z. Charbiwala, R. Shea, and M. B. Srivastava. Temperature compensated time synchronization. *Embedded Systems Letters, IEEE*, 1(2):37 –41, Aug 2009.

[27] M. Weiss, J. Eidson, C. Barry, L. Goldin, B. Iannucc, E. Lee, and S. Kevin. The case for cross disciplinary research on time aware applications, computers and communication systems (taaccs). 2013.

[28] Y.-C. Wu, Q. Chaudhari, and E. Serpedin. Clock synchronization of wireless sensor networks. *Signal Processing Magazine, IEEE*, 28(1):124–138, 2011.

[29] L. B. Yuste and H. Melvin. Client-side multisource media streams multiplexing for hbbtv. In *Consumer Electronics-Berlin (ICCE-Berlin), 2012 IEEE International Conference on*, pages 1–5. IEEE, 2012.