Research Article

Performance Evaluation of Concurrent Multipath Video Streaming in Multihomed Mobile Networks

James Nightingale, Qi Wang, and Christos Grecos

Centre for Audio-Visual Communications and Networks, School of Computing, University of the West of Scotland, Paisley PA1 2BE, UK

Correspondence should be addressed to James Nightingale; james.nightingale@uws.ac.uk

High-quality real-time video streaming to users in mobile networks is challenging due to the dynamically changing nature of the network paths, particularly the limited bandwidth and varying end-to-end delay. In this paper, we empirically investigate the performance of multipath streaming in the context of multihomed mobile networks. Existing schemes that make use of the aggregated bandwidth of multiple paths can overcome bandwidth limitations on a single path but suffer an efficiency penalty caused by retransmission of lost packets in reliable transport schemes or path switching overheads in unreliable transport schemes. This work focuses on the evaluation of schemes to permit concurrent use of multiple paths to deliver video streams. A comprehensive streaming framework for concurrent multipath video streaming is proposed and experimentally evaluated, using current state-of-the-art H.264 Scalable Video Coding (H.264/SVC) and the next generation High Efficiency Video Coding (HEVC) standards. It provides a valuable insight into the benefit of using such schemes in conjunction with encoder specific packet prioritisation mechanisms for quality-aware packet scheduling and scalable streaming. The remaining obstacles to deployment of concurrent multipath schemes are identified, and the challenges in realising HEVC based concurrent multipath streaming are highlighted.

1. Introduction

One of the main challenges in video streaming is to intelligently adapt the stream in response to dynamically changing network conditions in a way that attempts to minimise the distortion effect on the received video of adverse network conditions. H.264 Scalable Video Coding (H.264/SVC) [1], the scalable extension to the H.264 Advanced Video Coding standard (H.264/AVC) [2], has emerged as a promising means of adapting a video stream to prevailing network conditions but has yet to be adopted on a large scale for delivery of streamed video content. In H.264/SVC, a video stream consists of a number of scalable layers, each of which can be dropped either partially or in its entirety to adapt to changing network path conditions. Adapting an H.264/SVC stream to a change in the available bandwidth on a single wired network path is a well-understood problem; solutions that drop entire scalable layers [3] or individual packets from a layer [4] have both been previously proposed.

H.264/AVC is the current, widely deployed, video encoding standard, and its extensions such as H.264/SVC represent the current state of the art. However, work on a replacement for the H.264 family of encoding standards has reached the international standard stage with the next generation High Efficiency Video Coding (HEVC) standard published by the ITU as H.265 [5] in June 2013. HEVC offers the same visual quality as H.264/AVC while reducing the bandwidth requirement by 50%. The initial HEVC standard permits in-network adaptation of a video stream by employing temporal scalability by means of temporal prediction sublayers. A scalable extension to HEVC is also planned with work on the development having begun in December 2012 [6].

For situations where a single network path does not provide sufficient bandwidth to deliver a video stream, schemes that use the aggregated bandwidth of multiple paths [7–12] have been proposed. Nomadic users of streamed video content face the additional challenges associated with delivery over wireless mobile networks. A mobile network connected

to multiple potential access networks' paths (referred to as a multihomed mobile network) between the server and client presents a challenging environment for the delivery of delay-sensitive traffic such as video streaming. Of the multipath streaming schemes proposed in the literature, to the best of our knowledge, only [12, 13] are specifically designed for the use with H.264/SVC scalable video streams in the NEMO [14] based multihomed mobile networks environment. It is noted that any scheme that can deliver high-quality video content such as streaming in (or near to) real time in the downlink direction could potentially be explored in the reverse uplink direction, for example, to provide real-time passenger safety monitoring over the aggregated bandwidth of all available network paths in public transportation scenarios. The possibility of such use is considered in the design of the proposed system. The work in [12] is based on the "Always Best Connected" (ABC) model for multihomed mobile networks proposed by Wang et al. in [15]. In the ABC model, an application flow for example, a video stream is "switched" from one network path to another based on a policy-driven mechanism that considers both the application flow characteristics and the current network path conditions. The path selection component of the scheme proposed in [12] is an H.264/SVC-specific instance of the general model described in [15]. Therefore, although all available network paths can be used simultaneously (for the delivery of different application flows), any given application flow is in fact *sequentially* switched between paths by the path selection and packet scheduling algorithm operating at the streaming server.

The Stream Control Transmission Protocol (SCTP) [16] is an alternative networking protocol that supports multihoming. SCTP, following a host-to-host approach, differs from NEMO in that it only supports individual multihomed hosts. NEMO in contrast is able to support multihomed mobile routers that in turn provide mobility support for all of the mobile hosts within the mobile network in a much more efficient fashion. Mobile hosts in NEMO networks need not be mobility aware or multihomed themselves, whereas SCTP multihomed mobile hosts manage their own mobility as well as multihoming. A proposed extension [17] to the SCTP standard introduces a concurrent multipath transfer mechanism, cmt-SCTP. In cmt-SCTP, an application flow is split into a number of substreams, each of which is concurrently transmitted on a different end-to-end network path within an SCTP association. Due to the reliable nature of the SCTP protocol, the concurrent use of multiple paths leads to out-of-sequence packet delivery, and the "receiver buffer blocking" problem [18, 19] occurs.

In [13] path switching overhead was established as being the largest single limiting factor imposed upon the viability of video streaming in multihomed mobile networks environment, and the CMT-NEMO scheme was proposed to overcome this limitation. In this paper we significantly extend the work in [13] by providing a comprehensive analysis of concurrent multipath streaming in multihomed mobile networks which includes an investigation of the effects of background traffic on the CMT-NEMO scheme introduced in [13]. This work also extends the evaluation of concurrent multipath

streaming to include additional metrics of jitter and delay which have not previously been evaluated for concurrent multipath transmission in mobile networks. Additionally, looking to the future, this work provides a timely empirical investigation of the use of the newly standardised video encoding technology HEVC in concurrent multipath video streaming. We also propose a simple, HEVC specific packet weighting scheme for priority-based packet scheduling and scalable packet delivery (via selective packet dropping to meet network paths' bandwidth constraint). The results of this pilot HEVC implementation provide valuable insights into the challenges that will face researchers seeking to further investigate HEVC streaming in both single and multipath environments.

The rest of this paper is organized as follows. In Section 2, related work on multihomed mobile networks, H.264/SVC, HEVC, and multipath transmission is reviewed. The testbed implementation of our framework is described in Section 3 and numerical results of empirical experimentation are reported in Section 4. Section 5 concludes the paper.

2. Related Work

Due to the multidisciplinary nature of this work that combines networking, communications is and video signal processing, the state of the art of a number of related research areas are reviewed as follows.

2.1. Multihomed Mobile Networks. Working groups within the IETF have developed standards such as Mobile IPv4 [20] and Mobile IPv6 [21] for managing the mobility of mobile nodes. Both Mobile IPv4 and Mobile IPv6 address the mobility of an individual mobile host. The emerging wireless networking paradigm of mobile networks is based on the IETF Network Mobility (NEMO) standard [14]. In NEMO networks, groups of users travel together in unison (e.g., on a bus or train) and attach to the wider infrastructure via a mobile router serving as a gateway for the mobile network. The NEMO protocol [14] further considers the mobility requirements of an entire moving network by extending the Mobile IPv6 protocol to include its use on a new entity, the mobile router (MR), which manages mobility on behalf of all the individual mobile network nodes (MNNs) within the mobile network.

The mobile router is referred to as being multihomed when it is equipped with multiple network interfaces and is able to make simultaneous use of multiple network paths. The components of a multihomed mobile network are shown in Figure 1. In this typical scenario, a mobile network node away from its home link is contacted by a correspondent node (CN) via the NEMO-enabled home agent (HA). The home agent maintains a binding cache table entry linking the mobile router's home address (HoA) to its current care of address (CoA). In the case of multihomed mobile networks where the MR is equipped with multiple network interfaces, a third identifier called the binding identifier (BID) is used to differentiate between the different (HoA, CoA) bindings in the binding cache. Each BID is associated with a network interface on the MR.
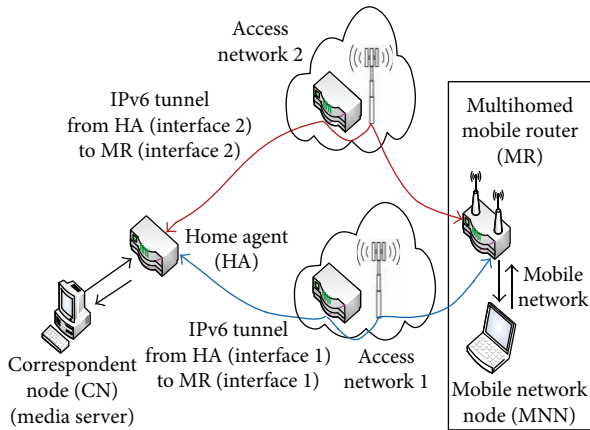
FIGURE 1: Topology of a two-path multihomed mobile network.

FIGURE 2: Extracting an H.264/SVC scalable bit stream.

When a mobile network node joins the mobile network, all traffic from a CN destined for the MNN is firstly directed to the HA of the MR. The packet is encapsulated in an IPv6 packet giving the destination address as the MR's CoA. This packet is then transmitted over an IPv6 tunnel between the HA and the MR. When the packet arrives at the MR, the encapsulation is removed and the packet is forwarded to the MNN. In multihomed mobile networks, each interface on the HA is connected to its corresponding interface on the MR by a separate IPv6 tunnel. Figure 1 shows a typical multihomed mobile network where the media server is the CN, and there are two distinct paths between the HA and the MR. It can been seen that the IPv6 tunnels connect the HA to the MR.

With ABC implementations of multihomed mobile networks, such as the one proposed in the MULTINET project [15], an application flow is firstly described as an association between the two end points (CN and MNN) using the (protocol, source IP address, source port, destination IP address, and destination port) quintuple to uniquely identify the flow. Consequently, by associating this flow quintuple with a NEMO BID, the flow is bound to a particular path between the HA and the MR. A policy-driven network selection algorithm uses metrics from the application flow (its bandwidth, delay, or other quality of service requirements) and the current network path conditions to decide the most appropriate path between the HA and the MR for any given application flow. An application flow is "switched" to another network path by changing the association (at the HA) between its unique quintuple flow identifier and the (HoA, CoA, BID) tuple that uniquely identifies a path from HA to MR. In the ABC model, a single flow can only be associated with one network path at any given time, with the flow being switched among paths by the network selection algorithm.

2.2. Scalable Video Coding. Video streams encoded using the H.264/SVC [1] format comprise a number of substreams (layers). An H.264/AVC [2] compliant base layer provides a minimum quality of video, while successive enhancement layers improve the quality of the video stream. H.264/SVC streams offer a three-dimensional scalability. Picture resolution, frame rate, and signal-to-noise ratio enhancement layers provide spatial, temporal, and quality scalability, respectively.
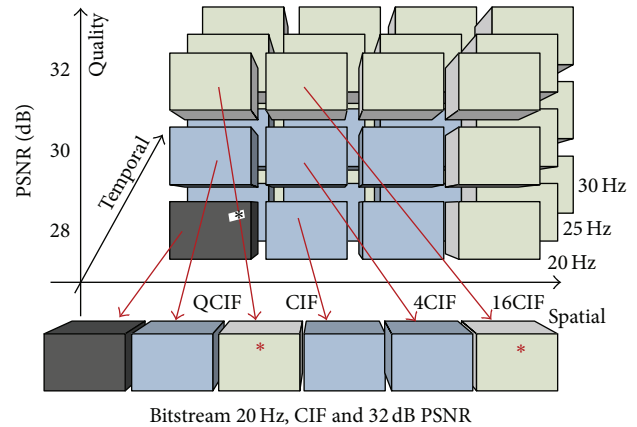
Figure 2 shows the extraction of a scalable substream matching a specified spatial (CIF), temporal (20 Hz), and quality (32 dB PSNR) tuple. H.264/SVC streams are readily adapted to meet terminal requirements or in response to changing network path conditions. For example, the sender will only send those layers that the receiver is capable of processing. Where there is insufficient bandwidth to transmit the entire stream, a Media-Aware Network Entity (MANE) may drop higher enhancement layers thus ensuring the delivery of the more important base layer and lower enhancement layers.

Meanwhile, the user is still provided with an acceptable (albeit lower) quality of received video. The user's quality of experience (QoE) is therefore managed by providing a graceful degradation of the stream rather than its disruption. As an example, the quality enhancement layer blocks (denoted by ∗) in the extracted bit stream in Figure 2 could be dropped to reduce the bandwidth requirement while maintaining an acceptable quality of received video with the desired spatial and temporal characteristics; alternatively the stream could have been adapted by dropping either temporal or spatial layer, or indeed some combination of the different scalable layers.

2.3. High Efficiency Video Coding. HEVC has been published in June 2013 by the ITU-T as H.265, the next generation video coding standard [5]. In common with the H.264 family of standards, HEVC consists of two separate layers, the video coding layer (VCL) and the Network Abstraction Layer (NAL). The VCL layer provides coded representations of each picture in a video sequence, while the NAL layer provides the basic data unit (NAL unit) in which VCL data is encapsulated for storage or transmission.

Compression efficiency in HEVC is improved (when compared to H.264/AVC) by around 50% which is achieved by the inclusion of a number of new encoding tools and the adoption of larger coding block sizes with an adaptive quadtree structure. The only form of scalability offered in the current HEVC specification [5] is temporal scalability. Three temporal prediction modes are used, the first of which (*Intraprediction Only*) does not permit temporal scalability. The *Low Delay* encoding mode has an Instantaneous Decoder Refresh (IDR) picture as its first picture with all subsequent

pictures being generalized P and B pictures. These generalized P and B pictures are only able to use pictures that are prior to themselves in output order (have lower picture order count (POC)) as references pictures. In the third (*Random Access*) encoding mode a hierarchical B picture structure similar to that employed in H.264/SVC is employed. The stream and IDR picture begin with intraencoded pictures being added approximately one per second; these Clean Random Access (CRA) pictures provide random access points within the stream. Pictures that follow a CRA picture in decoding order but precede it in output order may use pictures that come before the CRA for reference.

The experiments undertaken for this paper were conducted using version 6 (HM6.0) of the HEVC reference software. JCT-VC regularly update the HEVC reference software; the current version [22] is version 11.0 (June 2013).

2.4. Multipath Streaming Algorithms. Using the aggregated bandwidth of multiple network paths between sender and receiver has been proposed as a method of delivering high-quality video streams over links with a limited bandwidth. Chebrolu and Rao proposed the Earliest Delivery Path First (EDPF) scheme [7], where the arrival time of a packet is estimated for each available network path and the packet transmitted on the path offering the earliest delivery time. In [8], the same authors extended their work by including a frame-based selective drop mechanism. Fernandez et al. [9] further adapted EDPF to Time Division Multiple Access (TDMA) systems by inclusion of a time-slot policy mechanism. More recently, Jurca and Frossard [10] proposed a scheme that, while still using an earliest path first approach, considered the dependencies between packets in a video stream and only transmitted those packets that can be successfully decoded at the receiver.

This was achieved by dropping any packets, which rely on a previously dropped packet for decoding purposes, thus preventing the wastage of network resources on packets that are not viable. The authors of [11] identified a potential problem in [10], which can lead to out-of-sequence packet delivery to the client and proposed a new algorithm to overcome this limitation. The same observation is true for all the earliest path-type schemes. Our scheme in [12] included a mechanism to prevent the out-of-sequence packet delivery identified in [11]. In this work, we incorporate an additional component at the client to manage any out-of-sequence arrival that may still occur due to imprecise bandwidth measurements or transient interferences on the wireless links in our testbed.

Whilst the authors of [10] considered a generic scalable video stream, none of these schemes have addressed the higher levels of granularity and complex packet dependencies found in H.264/SVC-encoded streams. Our previous work [12, 23] considered the specific challenges associated with delivering H.264/SVC over multiple paths in multihomed mobile networks. The HEVC standard [5] is a very recent development, and no multipath streaming algorithms specifically targeted at HEVC have, as yet, been proposed in the literature.

The additional mobility-related overheads of NEMO and the path switching cost of ABC-based NEMO networks were mitigated in our optimised path selection and scheduling algorithm (OPSSA) [23]. A further review [13] of the performance of H.264/SVC streaming in multihomed NEMO environments identified the ABC-related path switching overhead as being a dominant limiting factor in the delivery of video streams in this context. The CMT-NEMO algorithm described in Section 2.5 was proposed in [13] to overcome the limitations placed on ABC-NEMO by path switching overheads.

2.5. Concurrent Multipath Transmission. The majority of existing concurrent multipath transfer (CMT) schemes are based on the SCTP transport-layer protocol. SCTP is a reliable transport protocol, which retransmits packets failing to reach the client and incorporates a congestion control mechanism [16]. An end-to-end SCTP association is made between two multihomed SCTP hosts. A number of independent paths exist within this association. In SCTP multihoming, only one of these paths may be used as the primary path for data transfer with the other being utilized for retransmissions and primary path failure redundancy.

Iyengar et al. [17] proposed the cmt-SCTP scheme where SCTP data chunks from the same application flow can be concurrently transmitted using all of the available paths within the single end-to-end SCTP association. Mechanisms are incorporated to reduce sender-initiated out-of-sequence packet delivery, to limit fast retransmissions and to manage the congestion-window update frequency. However, the reliable, sequence number driven nature of the SCTP protocol still leads to the "receiver buffer blocking" problem [18] as, when a client has to wait for retransmission of a missing data chunk, it is unable to pass the other chunks in the acknowledgement window to the application.

The authors of [24, 25] proposed the use of multiple sending and receiving buffers to resolve the issue of receiver buffer blocking in cmt-SCTP. Yuan et al. [24] also introduced the facility to differentiate subflows within an SCTP association in terms of quality of service (QoS) requirements, whilst in [25, 26] the use of cmt-SCTP in wireless environments was investigated. mCMT [26], a cmt-SCTP variant for use in wireless networks, employed multiple sending and receiving buffers in a path-orientated multistreaming scheme, which also incorporated the Media Independent Handover (MIH) scheme [27] for individual host mobility support.

In addition to the many other cmt-SCTP based schemes such as [28, 29] for concurrent multipath transmission, a number of generic CMT schemes have also been proposed. Tsai et al. [30] proposed a CMT scheme incorporating forward error correction (FEC) and path interleaving with an adaptive FEC block size. Liao et al. in [19] proposed SMOS, a sender-based multipath out-of-order scheduler for TCP-based multipath streaming, whilst in [31] the authors further considered the path correlation problem of shared bottlenecks on end-to-end paths and proposed a new generic multihoming sublayer.

Although all of the above schemes, whether SCTP or TCP based, offer solutions to some of the problems associated

with CMT, they all only considered the case of individual multihomed hosts. There is a significant difference between that environment and the one found in NEMO-based mobile networks. In NEMO networks it is the mobility agents (HA and MR) that are multihomed rather than the end nodes (CN and MNN). In SCTP, the end-to-end association is conducted over paths directly linking the network interfaces of the multihomed end hosts, where the IP address of each interface that is used as the endpoint of the individual paths in the association is known. However, in NEMO while the application flow is between the end nodes, the independent paths only exist between the interfaces of the HA and the MR. In CMT-NEMO [13] a mechanism is provided to enable the application to be split into subflows that are to be correctly associated with the interfaces at the HA and the MR to ensure that they travel over the desired network path. Each of the other CMT proposals described above was implemented at the transport layer or in a cross-layer manner between the transport layer and the application layer, whilst the CMT-NEMO scheme operates at the RTP/UDP/NEMO protocol stack and is able to support multihomed mobile networks.

In this work the components of CMT-NEMO are incorporated into, and become an integral part of, our comprehensive multipath video streaming framework for multihomed NEMO environments. Details of how the CMT components interact with the other components of our framework are provided in Section 3.

3. Framework Implementation

In this work, we introduce our comprehensive multipath streaming framework for H.264/SVC and HEVC over multihomed mobile networks. The framework is a substantial further development of the scheme previously proposed in [12]. Figure 3 gives an overview of the framework (the H.264/SVC-specific implementation is shown). H.264/SVC and HEVC specific implementations of the CMT-NEMO scheme [13] are combined with a modified version of the scheduling algorithm from [12], which no longer needs to mitigate the NEMO path switching delay. We also introduce a quality-layers-based packet weighting scheme for H.264/SVC, a simple packet weighting scheme for HEVC and an improved ancestor checking scheme which fully considers the granularity of H.264/SVC scalability. In the framework, the data plane processes and transmits video packets whilst the control plane provides supporting functionality regarding network path conditions, session setup, and so forth. Whilst the full framework is shown in Figure 3 and has been implemented for H.264/SVC, the current state of development of the HEVC standard does not, as yet, permit full implementation of components that leverage multidimensional scalability such as those responsible for target-rate derivation and extraction of substreams to a target bandwidth.

Our framework includes CMT-NEMO-based components of a session setup control structure to determine the number of potential paths from the streamer to the client and establish one application subflow (bound to a single network path), for each available path. A flow disaggregation mechanism overcomes the path switching limitation observed in previous schemes and facilitates concurrent multipath transmission. Preprocessing modules use historical data from a target-bit-rate derivation subsystem for NEMO-based vehicular networks to determine target bit rates that are used by the encoder and bit stream extractor, while a rate-distortion optimised stream is extracted using quality layers based on [32].

Encoding and extraction functions employ the JSVM reference software [33], which is written in C++. The quality level assigned to each NAL unit is also used to provide a packet prioritisation scheme that is specific to H.264/SVC and is used in our selective packet dropping mechanism in the case when there is insufficient aggregated bandwidth available.

Each NAL unit in the stream is checked for decoding viability using a new, more practical implementation of the ancestor checking scheme used in [10]. This mechanism is better suited to be used in real-time media-aware streaming situations by replacing the time-consuming recursive searching method in [10]. NAL units that pass the decoding viability test are packetized for RTP transport over UDP. A path selection and packet scheduling algorithm then determines if there is a viable path to the client and, if more than one path is found, selects the path offering the earliest delivery. Out-of-sequence packet delivery to the client is mitigated at both the streamer and the client in a manner that minimizes added delay required to prevent out-of-sequence packet arrival at the client. Finally, successfully scheduled packets are transmitted using the selected subflow, with the subflows being aggregated back to a single flow at the client prior to passing to the H.264/SVC decoder.

3.1. Framework Design Considerations. Placement of the stream adaptation, path selection, and packet scheduling agents within the network topology has a substantial effect on the viability of a multipath transmission scheme. In [7–9] the agents were placed at a network proxy deployed at the point of path divergence (e.g., the HA in multihomed mobile networks) whilst in [10–12] the agents resided at the streaming server (CN). Both [10, 11] only considered the case of multihomed hosts directly connected over independent paths via their multiple network interfaces. A signaling scheme in [12] sent control messages from the scheduling and path selection agents at the CN to the actual point of divergence at the HA where the path switching module directed the stream onto the desired path. In this work, we make the reasonable assumption that commercial multimedia content servers are unlikely to be equipped with interfaces that directly connect them to multiple heterogeneous network technologies. The point of path divergence is more likely to be at a router within the network infrastructure. In a multihomed mobile network, the logical point of path divergence is the HA, and the point of path convergence is the MR.

By considering the potential bidirectional use of the proposed streaming framework, further constraints are imposed on the possible deployment of these agents. Placement of the stream adaptation, path selection, and scheduling agents at the point of divergence would put a substantial burden on both the HA and, for bidirectional use, the MR. Both of
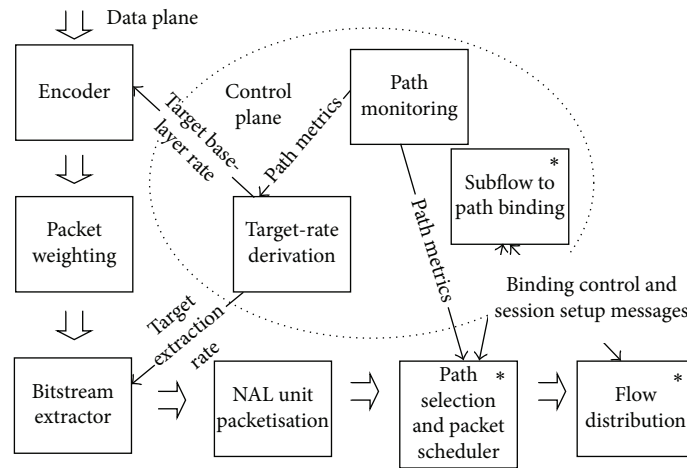
* Concurrent multipath transfer components

FIGURE 3: Overview of comprehensive streaming framework.

these devices are already handling the mobility needs of the entire mobile network. In the proposed scheme, the agents are placed at the streaming server. All of stream adaptation, path selection and packet scheduling tasks are performed at the CN or, in the case of the potential bidirectional use, at an appropriately resourced local fixed node (LFN) with a wired connection to the MR within the mobile network.

3.2. Session Setup and Control. The *subflow to path binding* subsystem is responsible for session setup and associating subflows with available paths. The initial session setup consists of a series of control messages, which are shown in Figure 4. A session setup agent at the MNN cooperates with its counterpart at the CN to establish a streaming session. New streamlining sessions are initiated by the mobile network node which sends a "stream initiation request" message (step 1 in Figure 4) to the CN. The CN responds by sending a "home agent discovery" message addressed to the MNN (step 2 in Figure 4). This message is intercepted by the HA that manages the mobility for the MR. The HA replies to the CN with a "client route identity" message containing the number of paths between the HA and the MR and the BIDs identifying each of its interfaces to the MR (step 3 in Figure 4). A reception "port negotiation" message is then undertaken jointly by the CN and MNN. The agreed reception port numbers are passed from the session setup agent to the subflow aggregation agent at the MNN (step 4 in Figure 4), which opens listening ports to the CN and then replies with a client ready message (step 5 in Figure 4). Upon the receipt of the "client ready" message, the CN sends a "subflow binding" message for each available path (from HA to MR) to the HA (step 6 in Figure 4).

This message contains the (protocol, source IP address, source port, destination IP address, and destination port) quintuple identifying a subflow and the NEMO BID with which the subflow should be associated. The HA then updates the binding cache and ensures that packets of the subflow described by the quintuple in the "subflow binding" message
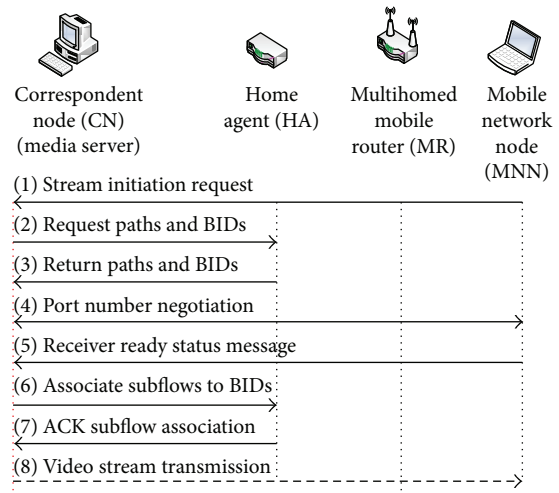

FIGURE 4: Session setup.

are delivered via the correct interface (BID) to the MR. When the subflow associations are established, the HA sends a "subflow acknowledgment" message to the CN (step 7 in Figure 4), which begins streaming to the client that is already in the listening state. The CN maintains a subflow to BID binding table for each streaming session. Figure 5 shows that application flow A is transmitted from the CN to the MNN. Application subflows (A, 1) and (A, 2) belonging to the same H.264/SVC, HEVC, or other application-type flows are associated with NEMO BIDs 100 and 200, respectively, at the HA.

3.3. Packet Weighting and Ancestor Checking (PWAC)

3.3.1. H.264/SVC PWAC Scheme. An H.264/SVC-encoded flow comprises a stream of logical data units called Network Abstraction Layer (NAL) units, each of which has a one-byte H.264/AVC [2] compliant header. While base-layer NAL units only comprise this one-byte header and payload, all
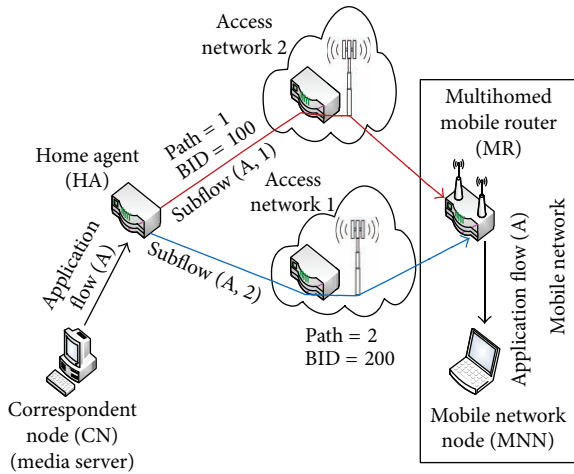
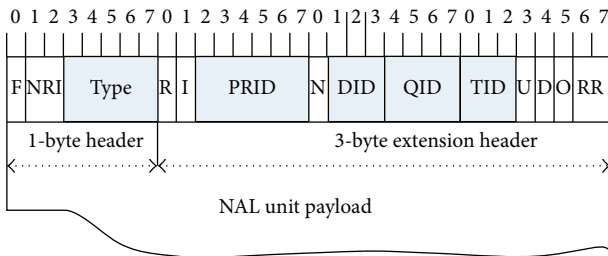FIGURE 5: Application subflow to NEMO BID binding.

FIGURE 6: Internal structure of an H.264/SVC NAL unit.

enhancement layer NAL units and supplementary enhancement information (SEI) messages also carry a three-byte scalability extension header. The extension header carries scalability information in three high-level syntax elements (shown in Figure 6). These syntax elements (*dependency_id*), (*temporal_id*), and (*quality_id*) respectively, describe the spatial, temporal, and quality layer characteristics of the NAL unit.

To assign different weights to the packets for selective dropping in the case of insufficient bandwidth, we employ a bitstream extraction method for H.264/SVC based on [32], where the rate-distortion impact of each NAL unit on the bitstream is calculated and then utilized to assign a quality level (ranging from 0 to 63) to the NAL unit. The quality level information is carried in a fourth high-level syntax element (*simple_priority_id*) within the extension header (or optionally in accompanying SEI messages). After a quality level is assigned to each NAL unit in the H.264/SVC stream, a scalable bitstream is extracted at the target bit rate by the bitstream extractor (which also receives a target extraction bit rate from the *target-rate derivation* subsystem).

The extracted stream is then passed from the JSVM bit stream extractor to the *path selection and packet scheduling* subsystem where stream adaptation takes place. In our quality-layers-based packet prioritisation scheme, we use the *simple_priority_id* field to weight the importance of NAL units when deciding which packets should be dropped at the MANE. Previous schemes for multipath transmission [10, 12]

used a less advanced, frame-based, and non-H.264/SVC-specific weighting approach where the I frames were given a higher weighting than the P and B frames and so forth. In our quality-layers-based scheme, NAL units with the lowest distortion impact are dropped to adapt to changes in network path conditions.

On a group of pictures (GOP) basis, NAL units are sorted in priority order. The NAL units are then presented to the ancestor checking component within the *path selection and packet scheduling* subsystem where packets that rely on a previously dropped packet for decoding are dropped from the stream. This preserves bandwidth by not sending packets that cannot be successfully decoded by the client. Those NAL units that pass the ancestor checking are then packetized based on RFC 6190 [34]. A "one NAL unit per RTP packet" strategy is applied to enhancement layer NAL units (Type 20), whilst a "single time aggregation packet (STAP)" strategy is employed for an AVC base-layer NAL unit (Type 5) packetized together with an NAL unit (Type 14) carrying the scalability extension header associated with the base-layer NAL unit.

From an example shown in Figure 7 it can be observed that packets numbered 6 and 13 are dropped since the estimated delivery time at the client was later than required by the decoding process. Because of these ancestor packets being dropped, their descendant packets numbered 9 and 15, which are dependent on them, are also dropped since they cannot successfully be decoded without packets 6 and 13. Sending packets 9 and 15 would have been a waste of bandwidth.

The ancestor checking scheme proposed in [10] used a time-consuming recursive searching method requiring that a search of all path queues is made for every new packet arriving at the scheduler in order to determine the status of a packet's ancestors. In this work, we propose a significantly less computation-intensive method of ancestor checking that is better suited to be used in a real-time environment by leveraging H.264/SVC scalability information. A prefetch window of variable size is employed, although in the experiments conducted for this paper the size was fixed at one GOP. Based on the scalable layer structure of the H.264/SVC stream, we determine which packets are dependent (for decoding) on others within the GOP, based on the frame number and scalability data contained in the NAL unit extension header (Figure 7). A record is maintained for the frame number and scalability data of dropped packets for the current prefetch window. This record is reset at the start of a new prefetch window. For each packet arriving at the ancestor checking component, a simple comparison is made of the NAL unit's frame number and scalability information (*dependency_id*, *temporal_id*, and *quality_id*) to the known failures within the current window. Given that the number of NAL units in a prefetch window is relatively small, our scheme provides a considerably faster and more efficient method of ancestor checking for H.264/SVC streams compared with [10]. Ancestor checking in this work is limited to NAL units dropped by the scheduler, the possibility of including an out-of-band feedback mechanism to deliver timely information on packets dropped in transit to the ancestor checking component will be considered in future work.
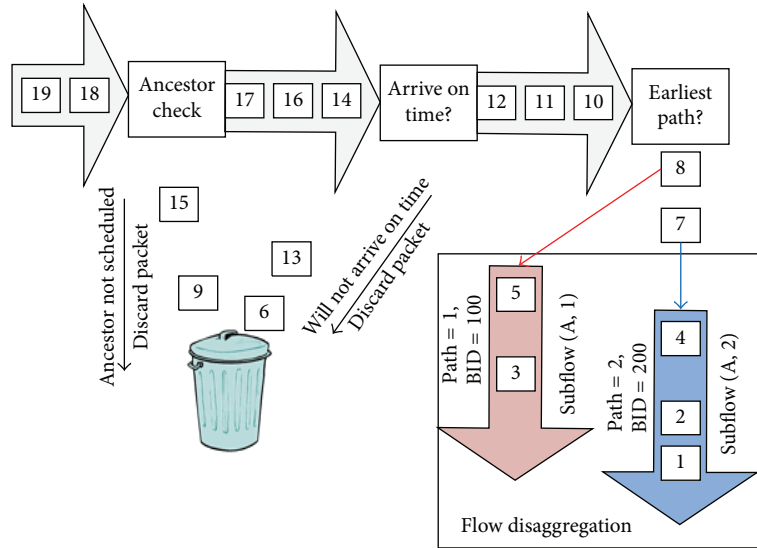
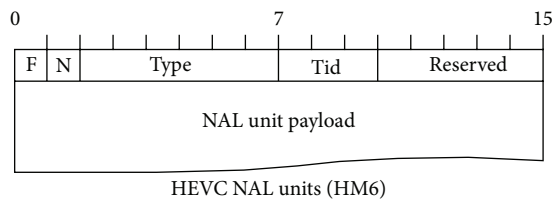FIGURE 7: The flow of packets through the path selection and packet scheduling subsystem.

FIGURE 8: Internal structure of an HEVC NAL unit.

3.3.2. HEVC PWAC Scheme. HEVC NAL units (in version HM6) consist of a two-byte header followed by the payload. Three fields within an HEVC header influence the manner in which packets can be prioritised. These fields could be employed by a selective packet dropping mechanism such as those employed by an MANE. The *nal_ref_flag* (N in Figure 8) is a one-bit flag denoting whether the picture to which this NAL unit belongs is used as a reference picture for other pictures in the sequence.

The 6-bit *nal_type* field indicates the NAL unit type in much the same way as for H.264 variants; however, the field is extended from 5 in H.264 to 6 bits in HEVC to accommodate the increased number of NAL unit types both currently used by HEVC and expected to be required for the envisaged extensions to HEVC. The *Temporal Layer_ID* field identifies the temporal prediction layer to which the picture contained in the NAL unit payload belongs. Substreams representing temporal layers of HEVC encoded streams can be successfully decoded when NAL units from higher temporal layers are missing or removed from the stream.

In this work we propose and employ a simple packet weighting scheme for HEVC which provides the highest level of importance to NAL units of pictures that are Instantaneous Decoder Refresh (IDR) pictures, Clean Random Access Pictures (CRA) or are marked as used for reference by the *nal_ref_flag* header field. NAL units are then further prioritised according to the temporal layer to which they belong with the highest level of importance being attached to the lowest temporal prediction layer.

Ancestor checking in this pilot implementation of concurrent multipath HEVC streaming is restricted to ensuring that, within a group of pictures (GOP), NAL units of higher temporal layers are only transmitted if the NAL units of the lower temporal layers from which they are predicted have been successfully transmitted.

3.4. Path Selection and Packet Scheduling. A *path state monitoring* subsystem, previously described in [12], provides path conditions to the path selection component. This component is also provided with the full network size of the packet including all NEMO-related tunneling overheads. The estimated arrival time on each available network path is calculated, and, if no path is able to deliver the packet to the client on time to be of use in the decoding process, it is dropped. Where a single viable path is found, the packet is passed to the *flow disaggregation* subsystem, which transmits it on the subflow associated with the chosen network path in the subflow to BID binding table. Where more than one viable path is found, the packet is transmitted on the path offering the earliest estimated arrival time at the client.

Figure 7 shows the flow of packets through the *path selection and packet scheduling* subsystem to the *flow disaggregation* subsystem. Using the NEMO protocol results in an additional network overhead due to IPv6 tunneling between the HA and the MR. In our streaming framework, we take account of all networking overheads when estimating the packet arrival time at the client for path selection purposes.

Network overheads of 100 bytes are added for every NAL unit (or fragment of an NAL unit) sent in a single level NEMO based mobile network (i.e., no other mobile network such as a personal area network nested within the mobile network). These network overheads consist of the initial IPv6 header containing the destination address of the MNN (40 bytes), the tunnelling IPv6 header (40 bytes) containing the

care of address (CoA) of the mobile router, an 8-byte UDP header, and a 12-byte (minimum) RTP header. On average the bandwidth required to transmit an HEVC encoded bitstream is increased by 9% in the NEMO environment due to these network overheads. This observation is based on a strategy of one NAL unit per RTP packet.

3.5. Out-of-Sequence Packet Handling. Out-of-sequence delivery of packets to the client is a significant problem in multipath delivery systems. It is especially prevalent in heterogeneous networks where each path has different characteristics such as available bandwidth and end-to-end delay. Resource constrained mobile devices may be limited in the amount of memory available for allocation to the receiver buffer of the client application. It is important to note that the nature of packet-switched networks often means that some level of out-of-sequence delivery to the client is often unavoidable.

Where a router has the choice of more than one path to a destination, it may choose to send some packets of the same application flow over different links to, for example, avoid congestion on a particular link or for load balancing purposes. Especially in busy wireless environments, it is possible that there will be sudden changes in the available bandwidth or end-to-end delay due to nodes either leaving or joining the network or contention in the wireless network that requires retransmissions. Since the sender-based approach has been proved to yield better performance in handling out-of-sequence packets [19, 35], we employ a sender-based out-of-sequence mitigation scheme that is supplemented by additional practical measures at the client.

At the CN, the estimated arrival time t_i^a of a packet p_i on each available path is calculated using the full network size of the packet and the end-to-end delay. Each packet has a decoding deadline t_i^d or time by which it must arrive at the client in order to be of use in the decoding process. The decoding deadline of a packet is calculated from the frame rate of the video and is relative to decoding time of the first packet in the stream. The decoding window of a packet opens at the decoding deadline (relative to the first packet) and closes at $t_i^d + \Delta$, where Δ is the playback delay at the client. If no available path can provide an estimated arrival time where $t_i^a \geq (t_i^d + \Delta)$, the packet is dropped at the streamer, and the failure points for the current prefetch window are updated to ensure that any subsequent packet relying on this packet for decoding is also dropped.

If only one viable path is found, the packet is passed to the *flow disaggregation* subsystem and placed on the subflow associated with the viable path found. Where more than one path can deliver the packet within the decoding window, the path offering the earliest arrival time is used. In order to prevent out-of-sequence delivery of packets at the client, two factors are considered in the *delay packet* function described in [13].

Firstly, if $t_i^a < t_i^d$, the packet will arrive before the opening time of its decoding window and will occupy the decoder buffer until its decoding deadline arrives. Dependent on the size of the playback buffer at the client, packets arriving before

the decoding window opens may lead to out-of-sequence delivery. Preventing a packet from arriving before the start of its decoding window ($t_i^a = t_i^d$) may not prevent out-of-sequence delivery, which can only be ensured when $t_i^a > t_{i-1}^a$. Therefore, when considering the estimated arrival time of a packet on any given path, if $t_i^a < t_{i-1}^a$, the packet would need to be delayed by $d_i = (t_{i-1}^a - t_i^a)$ on that path to ensure that it would not arrive before the previous packet.

When it is necessary to delay a packet at the streamer (CN) to prevent out-of-sequence packet delivery to the client (MNN), the path with the lowest added delay (d_i) is chosen. The added delay is taken into account when calculating the estimated arrival time of the next packet on the path where the delay was added, thus preventing any temporal drift in the estimated arrival time calculations.

3.6. Flow Disaggregation. The ABC approach to switching an application flow among network paths in NEMO-based mobile networks has previously been used in [12, 15]. It switches the application flow between the interfaces of the HA, thereby changing the path used between HA and MR.

An average delay of 137 ms is introduced for each path switching operation. The delay consists of the time for the CN to send a path switch message to the HA, the implementation of the path switch at the HA, and the waiting time until the CN receives a path switch acknowledgement from the HA. The overall delay measured in our testbed is likely to be higher in real implementations where the path between CN and HA may have a higher delay.

Our interpretation of the results in [15] indicates path switching between 200 and 300 ms. The CMT-NEMO based framework in this paper splits the application flow into a number of subflows at the CN. Each subflow is associated with a separate listening socket at the MNN and with a specific network interface at the HA.

Subflows travel across the path to which they are bound, with all of the mobility-related issues being handled by the existing NEMO protocol running at both HA and MR.

Table 1 compares CMT streaming over NEMO (CMT-NEMO) with both Always Best Connected streaming over NEMO (ABC-NEMO) and cmt-SCTP.

The flow disaggregation scheme consists of software agents at the CN, HA, and MNN. The agents at the CN would be replicated at the LFN and those at the HA replicated at the MR for bidirectional streaming. An application flow is split into subflows by creating a listening socket for each available path at the MNN and providing a subflow aggregation and out-of-sequence mitigation agent at the MNN. Such a design is applicable to applications beyond video streaming using H.264/SVC or HEVC.

3.7. Algorithm Summary. Algorithms 1–3 highlight the main algorithms implemented in the streaming framework. At the streamer, CMT-NEMO from [13] is fully integrated with updated *path switching and packet scheduling* modules from [12], which now include a revised ancestor checking scheme, enhanced out-of-sequence mitigation, and CMT-NEMO packet to subflow distribution.

TABLE 1: Features of the three schemes.

	Always Best Connected (ABC) NEMO	Concurrent multipath transfer (CMT) NEMO	cmt-SCTP
Use of paths	Switched sequential transmission.	Concurrent transmission.	Concurrent transmission.
Bandwidth	A trade-off against path switching overhead.	Effective bandwidth aggregation is achieved.	Effective bandwidth aggregation.
Aggregation	Higher levels of aggregation lead to increased switching overhead and lower QoE for the user.	Aggregation is suboptimal due to the relatively small overhead required to prevent out-of-sequence packet delivery (average 14 ms).	The reliable nature of SCTP can reduce throughput due to both control messages and retransmissions.
Path switching overheads	Average 137 ms [12].	No path switching delay incurred.	No path switching delay incurred.
Control plane overheads	Initial session setup control messages. Path switching REQ and ACK between CN and HA for every path switching operation.	Initial session setup control messages. No path switching messages during session.	Messaging for acknowledgement of delivery, retransmissions, and congestion control throughout streaming session.
Multihoming requirement	The HA and the MR in the infrastructure are multihomed.	The HA and the MR in the infrastructure are multihomed.	All end hosts are multihomed.
Mobility support	All mobile hosts in the whole mobile network.	All mobile hosts in the whole mobile network.	An individual mobile host only.

```
10    #STREAM PRE-PROCESSOR (H.264/SVC)
15    Get number of available paths m from Home Agent
20     For each available network path n
30         Get current bandwidth b_n^c from path monitoring sub-system
40         Get max bandwidth b_n^h min bandwidth b_n^l from route history file
50         If b_n^c < b_n^l lowest known bandwidth b_n^{lk} = b_n^c, else b_n^{lk} = b_n^l
60         If b_n^c > b_n^h highest known bandwidth b_n^{hk} = b_n^c, else b_n^{hk} = b_n^h
65     Next path
70     Base-layer target rate R^{bl} = ∑_{n=1}^{n=m} (b_n^{lk})
80     Extraction target rate R^{ex} = ∑_{n=1}^{n=m} (b_n^{hk})
90     If real time streaming encode scalable bit stream using R^{bl}
100     Perform rate distortion calculations to each NAL unit
110     Assign quality level for each NAL unit to simple_priority_id in NAL header as defined in [32]
120     Extract scalable bit stream at target R^{ex}
```

$$b_n^c, \quad b_n^h, \quad b_n^l, \quad b_n^{lk} = b_n^c, \quad b_n^{lk} = b_n^l, \quad b_n^{hk} = b_n^c, \quad b_n^{hk} = b_n^h, \quad R^{bl} = \sum_{n=1}^{n=m} (b_n^{lk}), \quad R^{ex} = \sum_{n=1}^{n=m} (b_n^{hk})$$

ALGORITHM 1: Stream preprocessing algorithm (H.264/SVC version).

```
570 #CLIENT SUB-FLOW AGGREGATOR
580 Repeat
590   For each available path
600     Read path received buffer
610     Next path
620     Sort received packets by RTP timestamp
630     Deliver aggregated flow to decoder
640 Until stream finished or receiver time out
```

ALGORITHM 2: Client side aggregation of subflows.

```
130    # STREAMING SESSION SETUP (H.264/SVC)
140    For each available path n
150       Get BID(n) from Home Agent
160       Create sub-flow S_n
170    Create sub-flow binding table entry at streamer for S_n ⟶ BID(n) association
180       Signal S_n ⟶ BID(n) association to Home Agent
190    Next n
200    #SCHEDULER
210    Sort each packet in pre-fetch window by simple_priority_id and t_i^d
220    If start of new pre-fetch window
230    Reset ancestor fail points
240    End If
250    For each packet in pre-fetch window
260       #ancestor_check_routine
270       For each fail point in pre-fetch window
280          If fail point is ancestor of this packet
290           Drop packet
300            Update fail points
310            Break
320       End If
330    Next fail point
340    #Estimate_arrival_time
350    Calculate mobility overheads as per [12]
360    Add mobility overheads to packet size
370    For each available path n
380       Calculate t_i^a(n) as per [12]
390       If t_i^a(n) ≤ (t_i^d + Δ)
400          If t_i^a(n) < t_{i-1}^a(n)
410             d_i(n) = (t_{i-1}^a(n) − t_i^a(n))
420          End If
440       End If
450    Next n
460       # INTEGRATED SELECTION AND PACKET SCHEDULING
470       If viable path found
480        Use path with earliest t_i^a
490        If multiple paths with same t_i^a use path with lowest d_i
500        Lookup sub-flow to BID binding table
510        Add packet to sub-flow queue associated with chosen path
520       Else
530        Drop packet
540         Update fail points
550       End If
560    Next packet
```

ALGORITHM 3: Main algorithm of the streaming framework incorporating CMT-NEMO, path selection, and packet scheduling (H.264/SVC version).

The preprocessing steps are detailed in Algorithm 1 using H.264/SVC as an example, where streams are preprocessed using path metrics from the *path monitoring* and *target rate derivation* subsystems. The stream is matched to the prevailing network conditions, and packets are prioritized, as described in Algorithm 3. Subflows are aggregated at the client, where additional out-of-sequence mitigation is also performed as shown in Algorithm 2. For the HEVC pilot implementation where the ability to extract a video stream to a target bandwidth is not yet available, the bandwidths of the network paths are set to match the requirements of the HEVC bitstream.

3.8. Control Plane Subsystems. Our framework contains three subsystems within the control plane. The *path monitoring* subsystem provides path state information on available bandwidth and delay to the scheduling algorithm.

Network emulation software running at a core router on each network path between the home agent and the mobile router changes the bandwidth and end-to-end delay
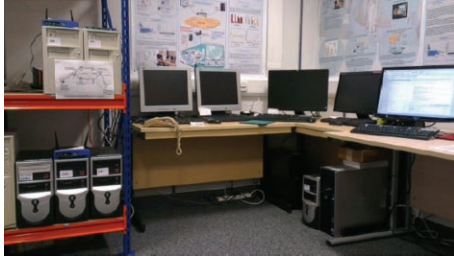
FIGURE 9: Testbed.

on each path autonomously. These changes are reported to the streamer using a series of low overhead control messages. A default message frequency of one per second is used; however, when the network emulator makes changes to a path (e.g., by reducing available bandwidth) to emulate the insertion of background traffic, a control message is triggered immediately. Although not implemented in our testing scenario, an out-of-band feedback mechanism for congestion control is considered for future work. For instance, a scheme for H.264/SVC congestion control in UDP was developed in [36], which is compatible with and could be integrated with our framework (an HEVC specific congestion control scheme is yet to be designed though). The *subflow to path binding* subsystem is described in Section 3.2. A *target-rate derivation* subsystem uses a combination of current and historical path metrics to make an informed decision on the bit rates at which a scalable stream should be both encoded and extracted to best match the anticipated network conditions. These control plane subsystems and the data plane subsystems described before constitute a practical and fully functional streaming system for concurrent multipath delivery of H.264/SVC or HEVC video to mobile network users.

3.9. Physical Testbed Implementation. Our framework has been implemented on a realistic, hardware-based multi-homed mobile networks testbed for testing and evaluation purposes. Figure 9 shows the actual testbed. The basic topology of the testbed is the same as that depicted in Figure 1 with the addition of core routers providing WAN emulation on the access network paths between HA and MR. The access routers are modified Linksys WRT54GL wireless routers running open source software OpenWRT. With the exception of the CN, which is an Intel Core i5 PC with 4 GB RAM and a solid state drive, all remaining nodes in the testbed are standard Intel Pentium 4 PCs with 1 GB RAM. All PCs run the Ubuntu Linux operating system with open source Network Mobility (NEMO) software installed at the HA and the MR.

The components of our framework are implemented in Linux user space using C++ with Python for pre- and post-processing tasks. The core routers run wide-area-network (WAN) emulation software and are configurable to change the bandwidth or delay on each path.

Options are available to run both static tests, where the characteristics of a path remain unchanged during a streaming session or to dynamically and independently change the nature of each path (within defined upper and lower limits).

4. Experimental Evaluation

For H.264/SVC evaluation four well-known video test sequences (*Bus*, *Foreman*, *Paris,* and *Soccer*) are encoded with the JSVM reference software using a range of spatial resolutions (QCIF, CIF, and 4CIF) and temporal resolutions (15, 30, and 60 fps). As a stream preprocessing step, the quality level data is generated using the *QualityLevelAssignerStatic* tool in the JSVM reference software. The bit stream is then extracted at a bit rate equal to the highest available aggregated bandwidth that will be configured within the testbed environment for the testing of that particular sequence.

Aggregated bandwidths in a range from 64 kbps to 3 Mbps and path delays of 20 ms to 250 ms are used during testing. Some tests are conducted using a static (for the duration of a streaming session) bandwidth and delay whilst in others the effects of competing background traffic (from other nodes in the network) are emulated when bandwidth and delay change dynamically during a session.

For HEVC evaluation two standard compliant HEVC test sequences (*Racehorses*, *BQSquare*) are encoded using version 6 of the HEVC reference software [22]. Each sequence is encoded using the standard HEVC conformance configuration files for *Intraprediction* Only, *Low Delay* and *Random Access* temporal prediction encoding modes. Two spatial resolutions (832×480, 416×240) and two temporal resolutions (30 fps, 60 fps) are employed. Encoded video bitrates are not manipulated by any means other than selection of different temporal prediction modes.

In the HEVC experiments the bitrate at which the video sequence is encoded using the standard conformance configuration is assumed to be the target bitrate, and the WAN emulation software is provided with these values.

Data from the CN, HA, CR1, CR2, and the MNN is collected in log files, and the network analysis tool Wireshark [37] is running at the HA, CR1, CR2, and the MR to allow collection and inspection of packets "in flight." The log file at the CN details the scheduling decision made for each packet including the data used to make that decision (packet size, priority weighting, scalability and frame number data, and estimated arrival time on each available path). The log at the MNN records all packets received and their arrival time.

The HA log shows subflow to BID binding data, and the CR logs contain details of path state changes and path state updates that have been transmitted as control messages to the CN. In the results highlighted in this paper, comparisons are made between Always Best Connected NEMO (ABC-NEMO) [12] and the video streaming framework which incorporates CMT-NEMO [13] with the described supporting subsystems of quality-layers-based prioritization, streamlined ancestor checking, and improved out-of-sequence packet handling.

4.1. Picture Quality Comparison Using H.264/SVC. In each figure (Figures 10, 11, 12, 13, and 14), schemes are compared when the available bandwidth has been allocated in a 50 : 50 split between the paths (equal paths) and also with an 80 : 20 split between the paths (differential or diff. paths). In each figure the legend denoting CMT-NEMO is used for brevity
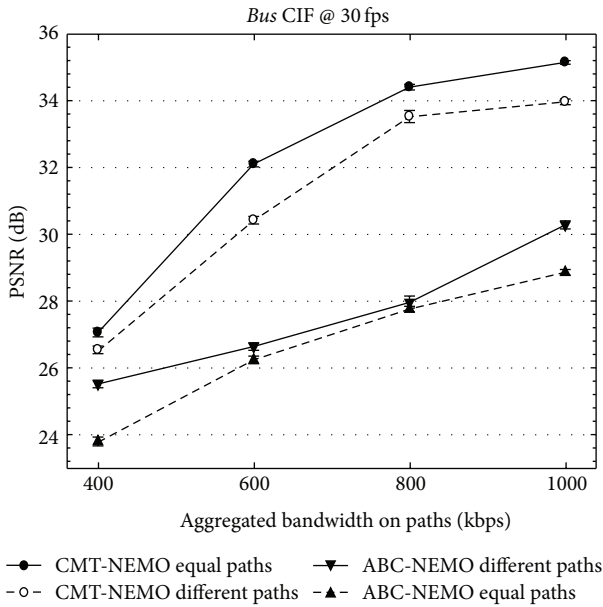
Figure 10: Comparison of schemes for the *Bus* sequence at 30 fps.
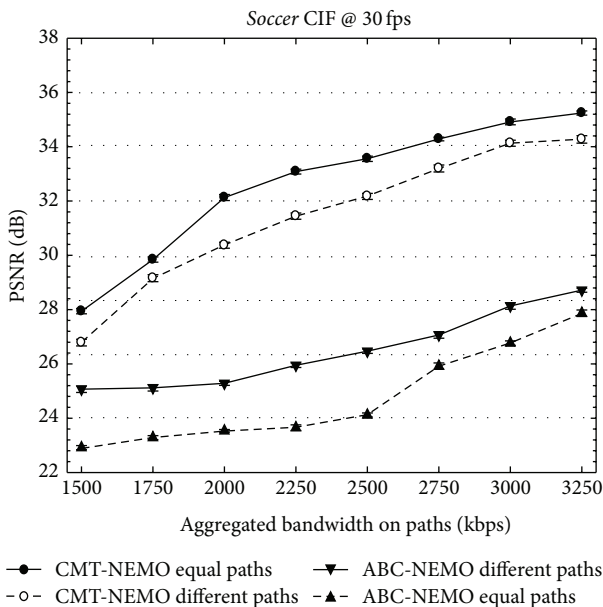

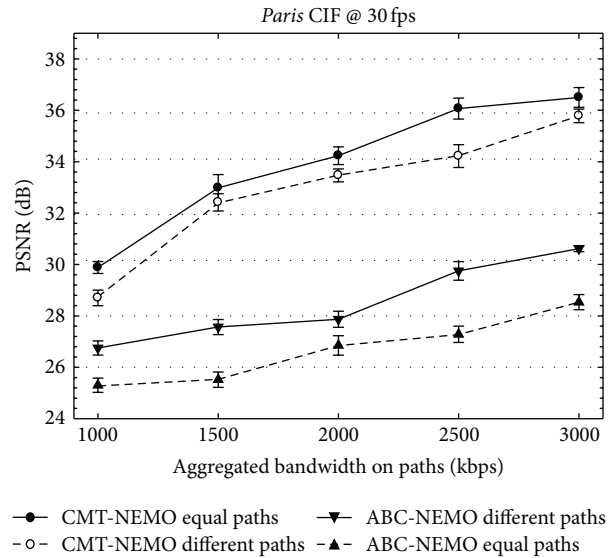Figure 11: Comparison of schemes for the *Soccer* sequence at 60 fps.

Figure 12: Comparison of schemes for the *Paris* sequence at 30 fps.
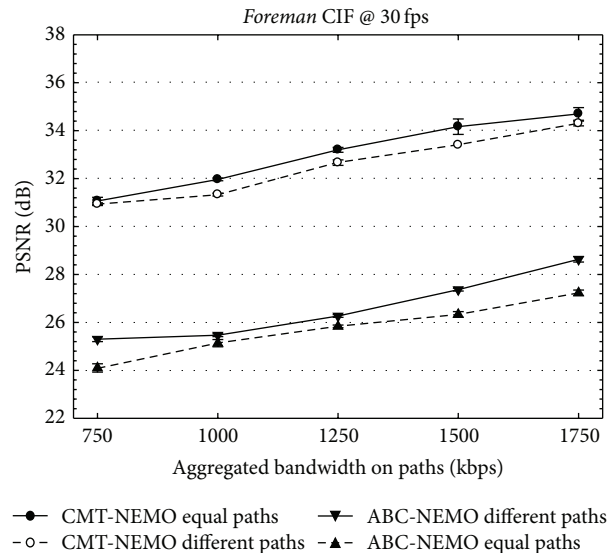
Figure 13: Comparison of schemes for the *Foreman* sequence at 30 fps.

and identifies the results obtained using the comprehensive streaming framework which incorporates CMT-NEMO.

Concurrent multipath transmission performs best in equal path scenarios for CMT-NEMO because of the reduced incidence of out-of-sequence packet delivery when paths are equal. In CMT-NEMO based framework, the sender-based out-of-sequence packet mitigation mechanism "holds back" packets that would arrive before the previously sent packet.

In addition, Figure 14 shows the results of streaming the *Paris* sequence at lower bit rates using an encoder target base-layer rate of 64 kbps. Overall, concurrent multipath transmission performs best on equal paths, providing a remarkable average PSNR improvement of 6.08 dB across all

testing scenarios with a maximum improvement of up to 8.72 dB being achieved in some tests.

Where the paths are differential, CMT-NEMO still outperforms ABC-NEMO considerably by an average of 4.20 dB and a maximum of 8.33 dB in some test sequences.

As the delay caused by out-of-sequence mitigation is less in equal path scenarios, it is possible to better utilize the available paths leading to an improvement in the number of packets arriving at the client on time and thus the resultant PSNR. Figure 15 shows both the number of packets arriving at the client and those that are usable in the decoding process for each scheme.

It can be seen from Figure 15 that more packets are delivered over equal paths using the CMT-NEMO based framework, with the opposite being true for ABC-NEMO.
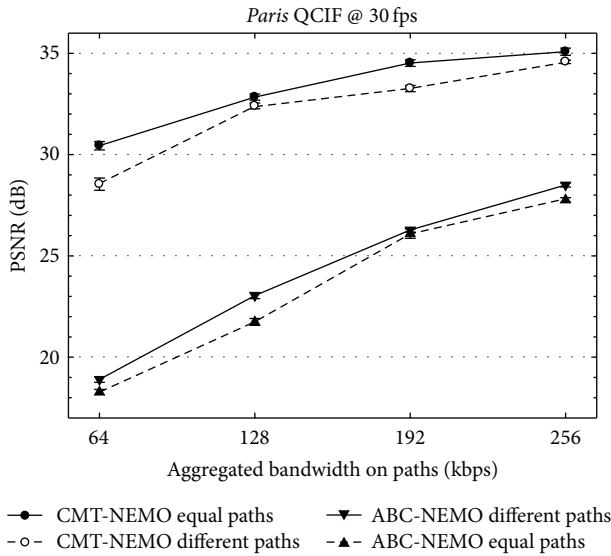
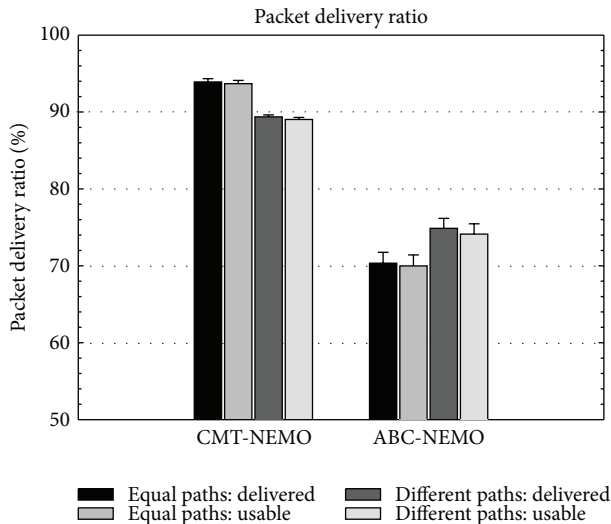FIGURE 14: Comparison using the *Paris* sequence at low bandwidths.

FIGURE 15: A comparison of packet delivery ratios at the client.

In ABC-NEMO, the path switching frequency is better controlled for differential path situations than for equal path scenarios. Path switching only takes place if the current (last used path) can no longer deliver packets within their decoding deadline.

Where the characteristics (delay and bandwidth) of the available paths are equal, ABC-NEMO is less effective as packets are distributed more evenly across the paths, which creates a higher path switching frequency. Each path switching adds a switching overhead, thereby reducing the number of packets that can be delivered in a given time. Therefore, ABC-NEMO performs better in terms of the number of packets delivered and the resultant PSNR in differential path situations. In concurrent multipath transfer, the opposite applies in that performance is better on equal paths.

Out-of-sequence packet delivery to the application running at the client is low for both the concurrent multipath
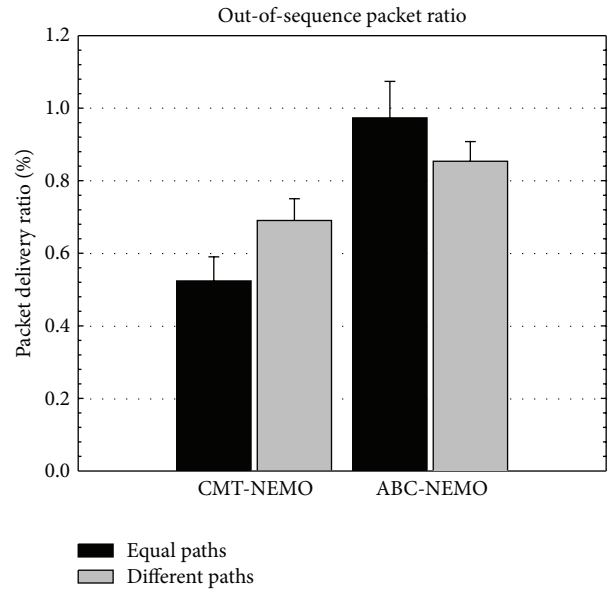

FIGURE 16: Comparison of out-of-sequence packet delivery rates.

streaming framework and ABC-NEMO schemes at less than 1%. However, it is noted that in ABC-NEMO out-of-sequence packets delivery mainly takes place during path switching operations only due to the sequential transmission style, and thus the out-of-sequence packet delivery ratio is low in nature. In contrast, due to concurrent transmission in CMT-NEMO based streaming framework, out-of-sequence packet delivery could occur all the time, and thus it must be tackled more rigorously through both sender and receiver mitigation mechanisms.

Due to the robust mitigation scheme, the number of packets sent out-of-sequence to the client in CMT-NEMO is minimized, even lower than that of ABC-NEMO, as shown in Figure 16.

The delay imposed by the out-of-sequence mitigation scheme at the sender is the primary cause of the reduction in throughput in differential path situations. Our framework gives a packet delivery ratio at the client (Figure 15) that is up to 24% higher than ABC-NEMO. The quality-layers-based weightings are employed in selective packet dropping to ensure that packets are dropped in a rate-distortion optimised manner. The higher number of packets arriving at the client contributes to a substantially improved PSNR value for all sequences. The PSNR results for concurrent multipath streaming represent a reduction of up to 68% in the "performance gap" identified in [38] for equal paths and up to 55% for differential paths.

The results obtained are valid across the whole range of video sequences and testing scenarios used in our experiments. Table 2 provides results of additional testing using alternative combinations of spatial and temporal resolutions for each of the test sequences.

4.2. Picture Quality Comparison Using HEVC. The HM6 version of the HEVC reference software [22] deployed in our framework and experiments does not offer any inherent

Table 2: Average PSNR and packets delivered for additional sequences.

	Bus QCIF 30 Hz	Soccer 4CIF 30 Hz	Foreman QCIF 15 Hz	Paris CIF 60 Hz
ABC-NEMO Avg. PSNR (dB)	27.22	25.35	26.97	25.82
CMT-NEMO Avg. PSNR (dB)	32.43	31.49	33.16	32.99
ABC-NEMO Usable packet delivery ratio	76.32	72.17	73.58	74.65
CMT-NEMO Usable packet delivery ratio	94.16	92.21	94.37	93.94

resilience to packet loss; consequently we adopt a "frame copy" based error concealment method in which a missing picture due to lost NAL units is copied either from the immediately previous picture (in the output order) or the nearest reference picture in the HEVC short term reference picture list if available in the reference picture list. The packet priority weighting scheme employed prioritises those pictures that are used for reference. Generally, in the *Low Delay* and the *Random Access* configurations of HEVC, which are both interpicture prediction modes, our experiments have shown that pictures of the highest temporal layer are unused for reference and may be safely discarded to meet a bandwidth constraint. In the *Intraprediction* Only configuration mode all pictures are intraencoded with temporal prediction or dependencies.

Figure 17 compares the PSNR achieved by ABC-NEMO and the CMT-NEMO based streaming framework when the *Racehorses* (832×480 @ 30 fps) sequence was streamed over multiple paths. The aggregated bandwidth was set to the encoded bandwidth of 2.253 Mbps which was achieved by encoding using the *Low Delay* main profile configuration with a quantisation parameter value of 27. The PSNR achieved when no packet loss is encountered is shown as the "encoded bitrate." In Figure 18 the *BQSquare* sequence is shown (416×240 @ 60 fps). The HEVC examples shown in Figures 17 and 18 were conducted using equal path settings where the available bandwidth was equally split between the two paths. Packet loss ratios and out-of-sequence delivery ratios for HEVC encoded sequences were consistent with those for the H.264/SVC experiment showing that the framework behaved consistently while delivering application flows are encoded under the two video encoding standards. Overall results from a small test sample of five test runs for each HEVC encoder configuration and test sequence combination showed that the PSNR losses relative to those of the encoded sequence before transmission were slightly higher for HEVC than for H.264/SVC.

Losses when comparing ABC-NEMO to the original sequence were, on average, 0.32 dB higher for HEVC than H.264/SVC and, on average, 0.28 dB when comparing concurrent multipath transmission. We attribute this performance issue to the relatively unsophisticated packet weighting scheme and error concealment mechanisms used in this
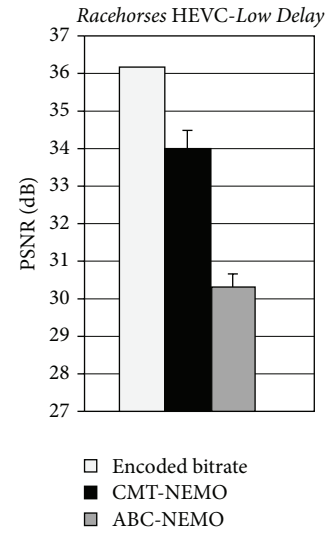

Figure 17: Comparison using HEVC at 2252 Kbps.

pilot implementation for HEVC. The results of our HEVC experiments show that our comprehensive video streaming framework for use in multihomed mobile networks can be readily adapted from its initial implementation for H.264/SVC to other video codecs and can potentially be applied in a generalised form for the concurrent multipath transmission of other types of application flow.

4.3. Delay and Jitter. In addition to PSNR-based video quality measurement and packet loss statistics, we also consider end-to-end delays and interpacket delay variation (IPDV) or jitter calculated based on RFC 1889 [39]. Both metrics have important impacts on a user's QoE [40, 41].

In Figures 19 to 21 we illustrate typical delay and IPDV comparisons of ABC-NEMO and CMT-NEMO using the *Paris* sequence at CIF resolution and 30 fps. A fixed delay of 25 ms is introduced by the WAN emulation module on each network path. The available bandwidth of 1.5 Mbps has been allocated in a 50 : 50 split between the paths (equal paths) and equates to the 1500 Kbps testing point shown in Figure 14. The running IPDV is shown in Figure 19.

Although both schemes maintain a mean IDPV below 50 ms, the CMT-NEMO based framework performs significantly better at less than 10 ms as shown in the inset. The
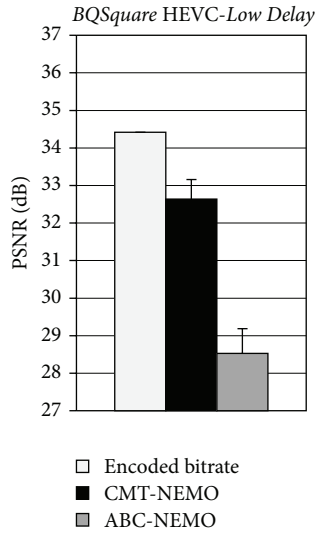
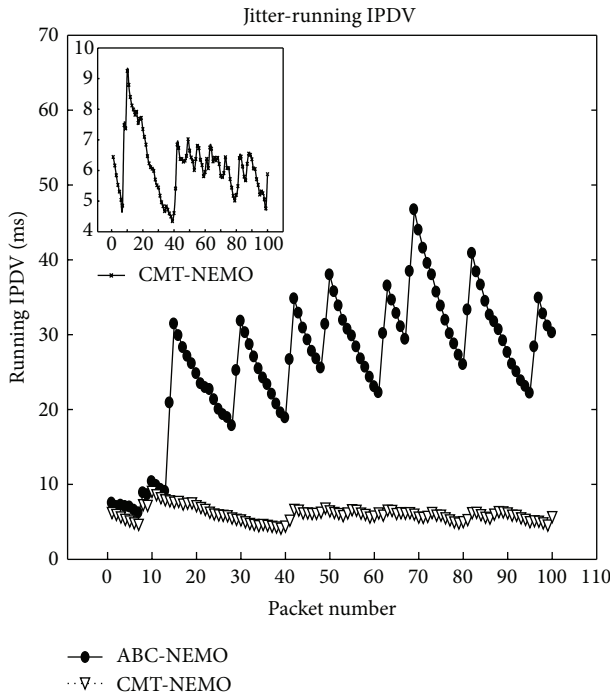FIGURE 18: Comparison of HEVC at 763 Kbps.

FIGURE 19: Comparison of jitter (running IPDV).

FIGURE 20: Comparison of jitter (IPDV).

FIGURE 21: Comparison of end-to-end delays.

instantaneous IDPV for each packet is plotted in Figure 20, where the "spikes" caused by path switching induced delays in ABC-NEMO can be clearly identified. Similar spikes can also be observed in running IPDV (Figure 19) and delay (Figure 21). The improvement offered by CMT-NEMO significantly reduces delay and IPDV, which can have beneficial impact on client buffer sizing and mitigating potential buffering problems that lead to degraded QoE. Table 3 shows that the maximum and average IPDVs are reduced by 152.1 ms and 23.7 ms, respectively, when using CMT-NEMO compared with ABC-NEMO.

In Figures 20 and 21, a spike in both IPDV and delay for CMT-NEMO occurs in the region of packet number 10.
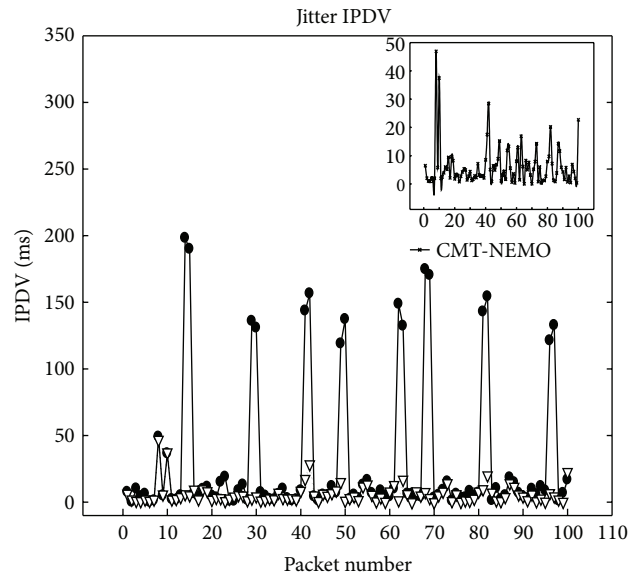
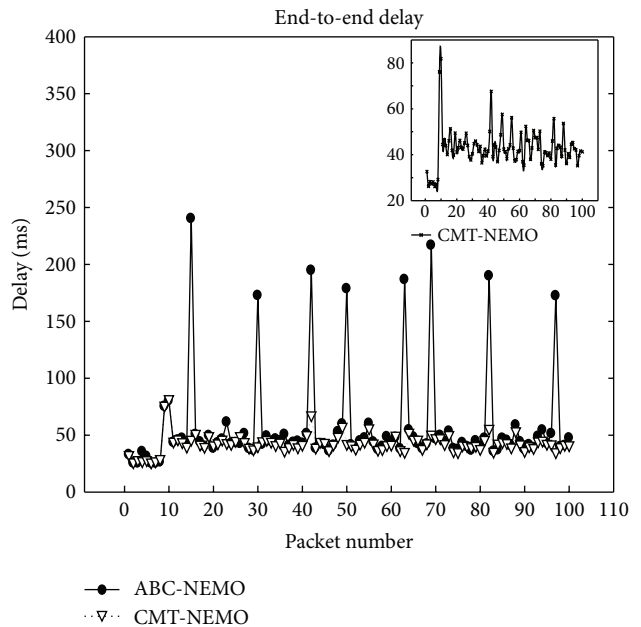This is due to the nature of an H.264/SVC stream where there can be a significant variation in NAL unit (and thus packet) size. The first few packets in the stream are parameter set NAL units, which are very small in comparison with the immediately following first video coding layer (VCL) NAL units, which are the instantaneous decoding refresh (IDR) units and, as such, are generally the largest in the stream.

This difference in propagation time between the smallest and the largest NAL units in the stream results in a spike in delay and IPDV. Table 3 shows that IPDV is further reduced in CMT-NEMO when the initial parameter set packets are

Table 3: IPDV comparison (milliseconds or ms).

	Max. IPDV	Avg. IPDV	STDEV
ABC-NEMO	198.00	29.77	53.55
CMT-NEMO	46.90	6.07	7.33
CMT-NEMO Excluding Parameter NALs	28.42	5.67	5.41

filtered. However, it should be noted that filtering these packets has the opposite effect on ABC-NEMO as discounting the very small IPDV between each of the parameter set packets increases the mean IPDV for the testing range.

4.4. Background Traffic Injection. In addition, when background traffic is emulated by reducing the available bandwidth on a path within the network, we observe that there is a short lag between the reduction taking place and the scheduler reacting to the change (not illustrated here for brevity). Typically, two or three packets experience an additional delay of between 20 ms and 50 ms and increased jitter (IPDV), before the system adapts to the new bandwidth when there is a uniform increase in delay (but not jitter) reflecting the reduced bandwidth.

A number of experiments were conducted in which real background video traffic was introduced between the streamer server and the client node. Performance of the streaming framework was measured under a range of background traffic rates. The background traffic was transmitted over a single path and also simultaneously over multiple paths. For each of the n available paths in the system the bandwidth set at the WAN emulator is p_i (Kbps), and for each of the m background flows in the system the bandwidth used by the flow is f_i (Kbps). The background traffic injection rate R is the sum of all background traffic flows β_t divided by the sum of all path bandwidths β_a

$$R = \frac{\beta_t}{\beta_a} = \frac{\sum (f_1, f_2, \ldots, f_m)}{\sum (p_1, p_2, \ldots, p_n)}. \tag{1}$$

The gross transmission efficiency G of a streaming system is a measure of the utilisation of the aggregated bandwidth of all available paths, while the effective transmission efficiency E is a measure of how much of the aggregated bandwidth is being used to deliver useable video payload to the client side decoder. For each packet i the size (in Kbytes) of the NAL unit(s) contained in the packet payload is s_i, and the network overheads required to encapsulate them for transmission by RTP over UDP in the NEMO environment is Δ_i. The full network resource (in Kbytes) required to deliver a packet is $\varphi_i = s_i + \Delta_i$. The number of packets successfully delivered to the client is r, and the number of useable packets containing NAL units which could be successfully decoded (arrived on time to be of use in the decoding process and had no unmet

dependencies or missing fragments) is μ. L_s is the duration of the streaming session in seconds. Consider

$$G = \frac{\left(\sum (\varphi_1, \varphi_2, \ldots, \varphi_r)/L_s\right)}{(\beta_a - \beta_t)},$$

$$E = \frac{\left(\sum (s_1, s_2, \ldots, s_\mu)/L_s\right)}{(\beta_a - \beta_t)}. \tag{2}$$

In (2) the maximum potential throughput which could be achieved by an application flow is shown for the case where R remains constant for the duration of a streaming session. In cases, such as those shown in Figure 22, where R was varied at every 30 to 50 seconds during a streaming session, the maximum throughput potential used in efficiency calculations was the sum of $(\beta_{ax} - \beta_{tx})L_{sx}$, where x is a time slot during which R remained constant (e.g., in Figure 22, R is constant at 50% from elapsed time = 40 seconds until elapsed time = 70 seconds).

Distortion efficiency D is measured as the ratio of achieved visual quality δ_a to maximum possible visual quality δ_m for each encoded sequence

$$D = \left(\frac{\delta_a}{\delta_m}\right). \tag{3}$$

From Figure 22, it can be seen that, in both ABC-NEMO and CMT-NEMO, the streaming mechanism responds to changes in the rate of background traffic injection. CMT-NEMO delivers a consistently higher PSNR than ABC_NEMO. It was observed that, for a short period after any change in R, both schemes delivered a reduced PSNR on one or two frames as the streamer adapts to changes in background traffic. This initial drop in PSNR after a change in R was consistently more pronounced in ABC-NEMO than in CMT-NEMO.

It can be seen from Figure 12 to Figure 16 that the relative difference in bandwidth and delay between paths in the multipath streaming system leads to a difference in the measured quality of the received video stream. The injection of background traffic, when applied to a single path, changes the bandwidth and delay ratios between the paths. This leads to an increased deviation from the mean PSNR for each test sequence. For example, in a system with two unequal paths, if traffic is added to the higher bandwidth path, this leads to an equalisation of the path characteristics. As CMT-NEMO performs better in equal path situations, the drop in PSNR due to added background traffic is offset in part by the increased effectiveness of the scheme, whereas in ABC-NEMO the move towards equal paths makes the scheme less efficient with the loss in PSNR being exacerbated by the reduction in scheme efficiency. The reverse also holds true when traffic is injected into one path in an equal path system. In Figure 23 it can be clearly seen that CMT-NEMO has a transmission efficiency that is approximately 20% higher than that of ABC-NEMO resulting in vastly increased distortion efficiency. In both schemes the relative difference between gross transmission efficiency and effective transmission efficiency increases slightly as R
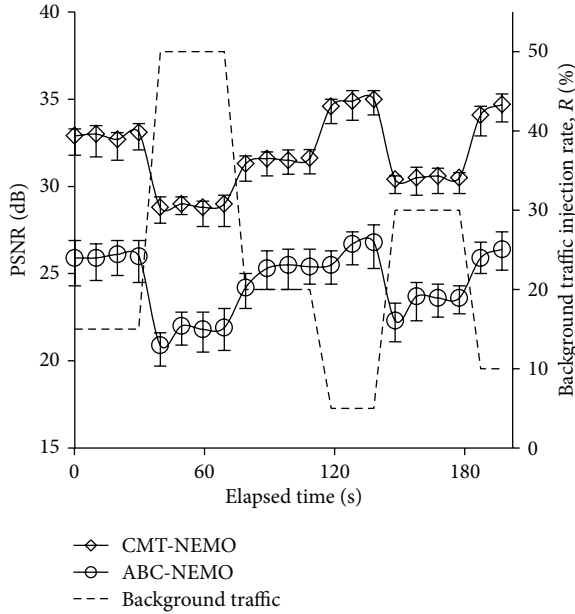
Figure 22: PSNR comparison of how each scheme responds to the injection of background traffic.

increases; however the distortion efficiency (D) decreases as R increases.

5. Conclusions

In this paper we have presented and empirically evaluated a comprehensive concurrent streaming framework for optimised, efficient delivery of H.264/SVC and HEVC video streams over multiple paths in mobile networks. The framework consists of and integrates multiple key components that provide a means of firstly adapting a video stream to the prevailing network conditions in a rate-distortion optimised manner and then using the aggregated bandwidth of multiple network paths concurrently to overcome bandwidth limitations on any single path. The framework comprises a concurrent multipath transfer scheme for NEMO-based mobile networks, which can be generalised for the concurrent multipath delivery of different types of application flows in NEMO environments. Other components include a video codec specific packet prioritisation scheme for optimised selective packet dropping in low aggregated bandwidth situations, a new mitigation scheme to minimise out-of-sequence delivery, an "on-the-fly" codec specific ancestor checking scheme suitable for real-time applications including scalable video streaming based on H.264/SVC, and the next generation video coding standard HEVC.

The proposed CMT-NEMO streaming system has been implemented on a realistic, hardware-based multipath mobile network testbed, with experimental results for H.264/SVC and HEVC showing a substantial improvement in the quality of the received stream compared with a previously proposed system ABC-NEMO. In the H.264/SVC case, an average PSNR improvement of 6.08 dB and 4.20 dB is achieved across all four video sequences in equal and differential path situations, respectively, with a maximum
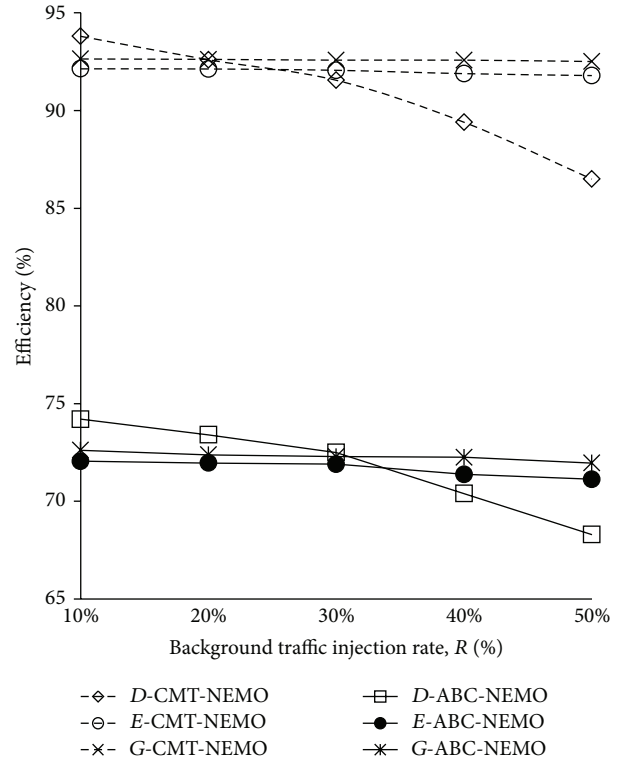

Figure 23: A comparison of gross transmission efficiency, effective transmission efficiency, and distortion efficiency for each scheme.

improvement of up to over 8 dB observed in some tests. The framework has been experimentally shown to improve the viability of multipath video streaming in mobile networks by delivering up to 24% more video packets over the same network while reducing the average jitter by 23.7 ms and the maximum jitter by 151 ms. A pilot implementation of the framework for HEVC further demonstrates that it can be readily adapted to the needs of emerging video encoding schemes and boasts clear-cut performance gains in contrast to the alternative ABC-NEMO scheme too.

While these results show a remarkable performance improvement, achieving the maximum user's quality of experience in the demanding multihomed mobile networking environments is an area that would benefit from further research. The gross transmission efficiency of our framework at approaching 95% is almost 20% higher than that of ABC-NEMO, but further work is still desired to provid more effective bandwidth aggregation and out-of-sequence delivery mitigation to enable optimal transmission efficiency. Similarly, while distortion efficiency is also over 90%, further work on improved packet weighting schemes for HEVC will raise this threshold. The reduced efficiency observed when using HEVC compared with using H.264/SVC indicates that further research on not only packet weighting and selective dropping but also error concealment schemes is required. Finally, all of the work performed in this evaluation uses objective measurement of video quality; future work would consider quality of experience using subjective and/or pseudosubjective evaluation techniques.

Conflict of Interests

Acknowledgments

References

[1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[3] T. Schierl, T. Stockhammer, and T. Wiegand, "Mobile video transmission using scalable video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1204–1217, 2007.

[4] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon, "Real-time system for adaptive video streaming based on SVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1227–1237, 2007.

[5] International Telecommunications Union (ITU), "High efficiency video coding," ITU-T Recommendation H. 265, http://www.itu.int/rec/T-REC-H.265-201304-I.

[6] J. Chen, J. Boyce, Y. Ye, and M. Hannuksela, "SHVC working draft 2," JCT-VC Document JCTVC-M1008, April 2013.

[7] K. Chebrolu and R. R. Rao, "Bandwidth aggregation for real-time applications in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 4, pp. 388–403, 2006.

[8] K. Chebrolu and R. R. Rao, "Selective frame discard for interactive video," in *Proceedings of the IEEE International Conference on Communications (ICC '04)*, pp. 4097–4102, June 2004.

[9] J. C. Fernandez, T. Taleb, M. Guizani, and N. Kato, "Bandwidth aggregation-aware dynamic QoS negotiation for real-time video streaming in next-generation wireless networks," *IEEE Transactions on Multimedia*, vol. 11, no. 6, pp. 1082–1093, 2009.

[10] D. Jurca and P. Frossard, "Video packet selection and scheduling for multipath streaming," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 629–641, 2007.

[11] M.-F. Tsai, N. Chilamkurti, J. H. Park, and C.-K. Shieh, "Multipath transmission control scheme combining bandwidth aggregation and packet scheduling for real-time streaming in multipath environment," *IET Communications*, vol. 4, no. 6, pp. 937–945, 2010.

[12] J. Nightingale, Q. Wang, and C. Grecos, "Optimised transmission of H.264 scalable video streams over multiple paths in mobile networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2161–2169, 2010.

[13] J. Nightingale, Q. Wang, and C. Grecos, "Removing path-switching cost in video delivery over multiple paths in mobile networks," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 38–46, 2012.

[14] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network mobility (NEMO) basic support protocol," RFC 3963, 2005.

[15] Q. Wang, T. Hof, F. Filali et al., "QoS-aware network-supported architecture to distribute application flows over multiple network interfaces for B3G users," *Wireless Personal Communications*, vol. 48, no. 1, pp. 113–140, 2009.

[16] R. Stewart, "Stream control transmission protocol," RFC 4960, 2007.

[17] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 951–964, 2006.

[18] J. R. Iyengar, P. D. Amer, and R. Stewart, "Performance implications of a bounded receive buffer in concurrent multipath transfer," *Computer Communications*, vol. 30, no. 4, pp. 818–829, 2007.

[19] J. Liao, J. Wang, T. Li, X. Zhu, and P. Zhang, "Sender-based multipath out-of-order scheduling for high-definition videophone in multi-homed devices," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1466–1472, 2010.

[20] C. Perkins, "IP mobility support for IPv4," RFC 3344, 2002.

[21] D. Johnson, C. Perkins, and J. Arko, "Mobility support for IPv6," RFC 3775, 2004.

[22] "HEVC Reference Software Test Model (HM)," https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/.

[23] J. Nightingale, Q. Wang, and C. Grecos, "OPSSA: a media-aware scheduling algorithm for scalable video streaming over simultaneous paths in NEMO-based Mobile Networks," in *Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC '11)*, May 2011.

[24] Y. Yuan, Z. Zhang, J. Li et al., "Extension of SCTP for concurrent multi-path transfer with parallel subflows," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, pp. 1–6, Sydney, Australia, April 2010.

[25] B. Wang, W. Feng, S.-D. Zhang, and H.-K. Zhang, "Concurrent multipath transfer protocol used in ad hoc networks," *IET Communications*, vol. 4, no. 7, pp. 884–893, 2010.

[26] C.-M. Huang, M.-S. Lin, and L.-H. Chang, "The design of mobile concurrent multipath transfer in multihomed wireless mobile networks," *Computer Journal*, vol. 53, no. 10, pp. 1704–1718, 2010.

[27] "Standard and metropolitan area networks: media independent handover services," IEEE Standard 802. 21, 2006.

[28] T. Dreibholz, E. P. Rathgeb, R. Seggelmann, and M. Tuxen, "Transmission scheduling optimizations for concurrent multipath transfer," in *Proceedings of the 8th International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports (PFLDNeT '10)*, pp. 2074–5168, Pa, USA, November 2010.

[29] P. Natarajan, N. Ekiz, P. D. Amer, and R. Stewart, "Concurrent multipath transfer during path failure," *Computer Communications*, vol. 32, no. 15, pp. 1577–1587, 2009.

[30] M.-F. Tsai, N. K. Chilamkurti, S. Zeadally, and A. Vinel, "Concurrent multipath transmission combining forward error correction and path interleaving for video streaming," *Computer Communications*, vol. 34, no. 9, pp. 1125–1136, 2011.

[31] J. Liao, J. Wang, T. Li, and X. Zhu, "Introducing multipath selection for concurrent multipath transfer in the future internet," *Computer Networks*, vol. 55, no. 4, pp. 1024–1035, 2011.

[32] I. Amonou, N. Cammas, S. Kervadec, and S. Pateux, "Optimized rate-distortion extraction with quality layers in the scalable extension of H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1186–1193, 2007.

[33] J. Reichel, H. Schwarz, and M. Wien, "Joint Scalable Video Model 11 (JSVM 11)," Tech. Rep. no. JVT-X202, Joint Video Team, July 2007.

[34] S. Wenger, Y.-K. Wang, T. Schierl, and A. Eleftheriadis, "RTP payload format for scalable video coding," RFC 6190, 2011.

[35] D. Kaspar, K. Evensen, A. F. Hansen, P. Engelstady, P. Halvorsen, and C. Griwodz, "An analysis of the heterogeneity and IP packet reordering over multiple wireless networks," in *Proceedings of the IEEE Symposium on Computers and Communications (ISCC '09)*, pp. 637–642, July 2009.

[36] D. T. Nguyen and J. Ostermann, "Congestion control for scalable video streaming using the scalability extension of H.264/AVC," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 2, pp. 246–253, 2007.

[37] Wireshark, http://www.wireshark.org/docs/.

[38] J. Nightingale, Q. Wang, and C. Grecos, "Performance evaluation of scalable video streaming in multihomed mobile networks," *Performance Evaluation Review*, vol. 39, no. 2, pp. 29–31, 2011.

[39] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobsen, "RTP: a transport protocol for real-time applications," RFC 1889, 1996.

[40] J. Asghar, F. Le Faucheur, and I. Hood, "Preserving video quality in IPTV networks," *IEEE Transactions on Broadcasting*, 2009.

[41] J. Zhang and N. Ansari, "On assuring end-to-end QoE in next generation networks: challenges and a possible solution," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 185–191, 2011.