

Cross-layer Fairness-driven Concurrent Multipath Video Delivery over Heterogenous Wireless Networks

Changqiao Xu, *Member, IEEE*, Zhuofeng Li, Jinglin Li, Hongke Zhang and Gabriel-Miro Muntean, *Member, IEEE*

Abstract—The growing availability of various wireless access technologies promotes increasing demand for mobile video applications. Stream Control Transmission Protocol (SCTP)-based Concurrent Multipath Transfer (CMT) improves the wireless video delivery performance with its parallel transmission and bandwidth aggregation features. However, the existing CMT solutions deployed at the transport layer only are not accurate enough due to lower layer uncertainties such as variations of the wireless channel. In addition, CMT-based video transmission may use excessive bandwidth in comparison with the popular TCP-based flows, which results in unfair sharing of network resources. This paper proposes a novel Cross-Layer Fairness-Driven SCTP-based Concurrent Multipath Transfer solution (CMT-CL/FD) to improve video delivery performance while remaining fair to the competing TCP flows. CMT-CL/FD utilizes a cross-layer approach to monitor and analyze path quality, which includes wireless channel measurements at data-link layer and rate/bandwidth estimations at transport layer. Furthermore, an innovative window-based mechanism is applied for flow control to balance delivery fairness and efficiency. Finally, CMT-CL/FD intelligently distributes video data over different paths depending on their estimated quality to mitigate packet reordering and loss, under the constraint of TCP-friendly flow control. Simulation results show how CMT-CL/FD outperforms existing solutions in terms of both video delivery performance and TCP-friendliness.

Index Terms—Mobile video services, cross-layer design, fairness, SCTP, heterogeneous wireless networks.

I. INTRODUCTION

The increasing availability of various wireless access technologies, such as WiFi, WiMax, LTE, etc., and

Copyright (c) 2014 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

Manuscript received May 30, 2014. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61372112 and Grant 61232017, in part by the Beijing Natural Science Foundation under Grant 4142037, in part by the National Key Basic Research Program of China (973 Program) under Grant 2013CB329102, in part by the Fundamental Research Funds for the Central Universities, and in part by Science Foundation Ireland under the International Strategic Cooperation Award Grant SFI/13/ISCA/2845.

C. Xu, Z. Li and J. Li are with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. (e-mail: {cqxu, lizo, jlli}@bupt.edu.cn).

H. Zhang is with the National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing, China. (e-mail: hkzhang@bjtu.edu.cn).

G.-M. Muntean is with the Performance Engineering Laboratory, School of Electronic Engineering, Network Innovations Center, RINCE, Dublin City University, Dublin, Ireland. (e-mail: gabriel.muntean@dcu.ie).

growing computing power and storage capacity of mobile devices encourage the demand for real-time content-rich video applications [1], [2]. Supporting these video applications with bandwidth-intense and delay-sensitive requirements while maximizing the wireless network resource utilization is a significant challenging task [3], [4]. The Stream Control Transmission Protocol (SCTP) [5]-based Concurrent Multipath Transfer (CMT) [6] has been recognized as a promising transport layer technology to meet the balance between stringent quality-of-service (QoS) requirements for video distribution and efficient resource utilization. CMT is able to utilize multiple interfaces to transport data in parallel manner [7]. Moreover, the sender side can schedule the traffic and balance the congestion across multiple paths in order to increase the QoS of video applications. Therefore, CMT has an excellent potential for providing benefits in terms of bandwidth aggregation, fault tolerance and load balancing for video distribution over wireless networks [8]-[10].

Although the advantages of employing CMT have been demonstrated to be very useful for video content delivery, there is still significant ongoing work addressing many remaining challenges. The most important concern of CMT-based video transfer is related to handling video data reordering and loss. Due to the highly dissimilar and dynamic characteristics of different paths, the classic CMT's round-robin scheduler is bound to buffer blocking. The buffered video data cannot be handed over to application layer before its playout time. Furthermore, since wireless channels are very unreliable, packet loss happens frequently and the resulted reception gaps also affect the processing of data reordering. In addition, packet loss is often caused by wireless errors, and is improperly handled as congestion in the SCTP original design. Although many SCTP CMT solutions focus on solving these issues [11], [12], most of them explore data exchanges, monitor sender-receiver interactions or retrieve connection parameters at transport layer only. Because the time and frequency variations of wireless link parameters do not relate directly to the transport layer, this type of schemes may not be accurate enough without considering first-hand information from the lower layers.

Another concern when applying CMT to video delivery is fairness towards other TCP-like traffic. By adopting independent TCP-like congestion control on each path, SCTP CMT sessions will tend to have sending rates multiple times larger than that of a single-path flow. When the network is congested, CMT-based video flows aggressively occupy excessive band-

width share in comparison with other competing traffic flows. It is necessary to include some TCP friendly flow control in conjunction with CMT in order to reduce its aggressiveness and ensure friendliness in sharing network resources.

This paper proposes a novel Cross-Layer Fairness-Driven SCTP-based CMT solution (CMT-CL/FD) for parallel video transfer over heterogeneous wireless networks. CMT-CL/FD employs a cross-layer optimized scheduling mechanism for handling data reordering and loss. Additionally, CMT-CL/FD introduces a flow control mechanism which ensures fairness towards other TCP-like flows. This paper's major contributions are as follows:

- Upgrades the calculation of signal-to-noise ratio at data-link layer to render the wireless measurements more accurate and convenient.
- Combines both rate and bandwidth estimations at transport layer to provide dual knowledge of path average and instantaneous sending behaviors.
- Designs an innovative window-based flow control mechanism to maintain the balance between transport efficiency and TCP-friendly fairness.
- Develops a novel cross-layer path evaluation model to obtain path quality integrating information from data-link and transport layers, and applies it into the parallel data distribution algorithm with a retransmission policy.

The proposed CMT-CL/FD was thoroughly tested and results showed how it improves user quality of experience levels during concurrent multipath video transfer, while still remaining fair to the competing TCP flows.

II. RELATED WORK

In the recent years, the growing interest in multipath video streaming has resulted in many peer-reviewed publications. Li *et al.* focus on robust multipath rate-control and bandwidth reservation to improve video quality for scalable video multicast streaming [13]. Kserawi *et al.* designed an application of field-based anycast routing (FAR) protocol to increase video delivery ratio and decrease end-to-end delay [14]. Zhu *et al.* proposed multi-path provisioning algorithms for cloud-assisted SVC streaming in heterogeneous networks [15]. These works mainly involve the network layer for optimization of routing to support efficient multipath video streaming.

An increasing number of researchers have concentrated their efforts on the promising transport layer multi-homed protocol SCTP for video transmission [8]-[10]. SCTP is a new transport layer protocol that supports multi-homing and multi-streaming features [5]. SCTP uses the primary path for data transmission and secondary paths as alternative paths, so it does not support parallel transmissions over multiple paths in its original description. Iyenger *et al.* were first to propose some effective algorithms aiming at efficient CMT operations [6]. CMT uses SCTP's multihoming feature to concurrently distribute data across multiple independent end-to-end paths. This enables CMT support the high-bandwidth delay-intolerant video content delivery over heterogeneous wireless networks.

Lately, there has been extensive interest in the research on SCTP CMT-based multimedia delivery. Huang *et al.* designed

a timed reliable CMT for multimedia content delivery by combining techniques of CMT, SCTP's partially reliable extension and prioritized stream transmission [9]. Baek *et al.* proposed an SCTP-based video partially reliable multicast protocol appropriate for multi-homed wireless network environment [10]. However, the authors ignored that the CMT performance was largely degraded by data reordering due to path dissimilarity.

Many efforts have been devoted to addressing the packet reordering problem. Wallace *et al.* developed a renewal theory and Markov chain-based framework to model the expected throughput of a CMT session [11]. Galante *et al.* extended the round-robin scheduling based on two types of bandwidth estimations (i.e. Packet Pair and TCP Westwood+) [12], and chooses the paths with lowest transmission time. However, all above solutions depend solely upon transport layer QoS-related parameters and they do not make any difference between packet loss due to wireless error or congestion and therefore are not suitable to be applied in wireless scenarios.

Other research works have proposed cross-layer designs to tackle problems specific to wireless network transmissions. Fitzpatrick *et al.* combined signal strength with quality-related metrics (E-model) in order to make handover decisions between SCTP paths and guarantee high VoIP quality [16]. Taenaka *et al.* proposed a frame-retry-based handover method with multi-path transmission for VoIP communication [17]. However, both of these solutions focus on the handover and do not address the parallel data transfer. We previously proposed a novel MAC-SCTP based cross-layer cognitive CMT for video distribution in varying wireless transmission [18]. However the simple path selection and data distribution algorithm proposed cannot make full use of the multipath resources.

Apart from the CMT-based video delivery-related performance issues, there are also fairness-related research aspects. As TCP is the dominant transport-layer protocol (more than 90% over Internet backbone [19]), any new protocol should be friendly towards the TCP traffic in order to be compatible with today's Internet. The *TCP-friendly fairness* has been introduced and is widely agreed to be a very important overall performance indicator. A data flow is said to be TCP-friendly if it has similar behavior with a TCP flow and shares reasonable bandwidth with the other flows in congestion conditions. IETF's TCP-Friendly Rate Control (TFRC) [20] is asserted to be "reasonably fair", but this cannot be stated for CMT without multihoming support.

More recently, Scharf *et al.* presented a TCP-friendly multi-connection TCP (MCTCP) solution [21], which makes use of the combined bandwidth of several paths. Wischik *et al.* deployed a resource pooling principle to Multipath TCP (MPTCP) [22] in order to achieve fairness towards competing traffic. Network Coding was used to improve the performance of multipath TCP in NC-MPTCP [23], MPLOT [24], HMTCP [25] and FMTCP [26]. FMTCP, NC-MPTCP and MPLOT also preserve fairness towards TCP flows, but their proposers have ignored the dynamic nature of the wireless networks, which may lead to improper flow control due to incorrect loss handling. Additionally they do not take into account specific QoS requirements (i.e. delay and bandwidth), very important in the case of video applications.

Our previous work CMT-QA [8] provides generic path-quality-aware mechanisms for SCTP CMT in order to enable smooth high quality service provisioning for real-time video delivery. However, as CMT-QA concentrates on improving the performance and lacks any consideration for fairness towards other traffic flows, it may occupy excessive bandwidth resources in loaded wireless network environments. Additionally, it estimates path quality and determines the reason for packet loss at transport layer only, which may have limited accuracy due to low-layer uncertainty in wireless channels. Addressing CMT-QA limitations, this paper proposes CMT-CL/FD with the following goals: (i) optimal cross-layer data scheduling, and (ii) fairness to TCP flows.

III. CMT-CL/FD ARCHITECTURE OVERVIEW

Fig. 1 illustrates the architectural design of CMT-CL/FD including the sender, receiver and multiple paths via the wireless networks. The sender acquires information from both data-link and transport layers in order to assess the path quality. Based on this path quality, the sender distributes and transfers data packets in parallel over the most promising paths in an efficient and friendly manner. The receiver accepts these packets over the multiple paths, assembles them in the correct order and then sends feedback with information on data reception status. CMT-CL/FD is composed of eight featured modules, namely *Effective Signal-to-Noise Ratio (ESNR) Calculation*, *Reference Round Trip Time (RTT) Measurement*, *Transmission Rate (Tx-Rate) Estimation*, *Bandwidth (BW) Estimation*, *Friendliness Window (FWND) Flow Control*, *Cross-layer Evaluation Model*, *Parallel Data Scheduler* and *Loss-Cause Dependent Retransmission (Rtx) Policy*.

The **ESNR Calculation** module is placed at data-link layer to provide support for low-layer detections. Since direct measurement of Signal-to-Noise Ratio (SNR) is not accurate in today's devices (e.g. network interface cards) [27], [28], this module calculates the Effective Signal-to-Noise Ratio (ESNR) as an equivalent SNR of the same signal in the Additive White Gaussian Noise (AWGN) channel. This is a simplified substitution of SNR complex computation in real wireless channels. This equivalence is able to integrate assorted effects on wireless communication for high-layer modeling. One of the major tasks of this module is the regular recording of Frame Error Rate (FER). Then the module derives ESNR according to the in-use wireless configurations (e.g. access technology, antenna assembly, modulation type) and reports it periodically to the transport layer.

At transport layer, the **Reference RTT Measurement** module calculates path delays by using the SCTP Heartbeat mechanism, rather than via the normal Smoothed Round Trip Time (SRTT) measurements. This path delay information is useful in the adjustment of loss handling mechanism. The **Tx-Rate Estimation** and **BW Estimation** modules provide dual knowledge about how fast the data can be sent across multiple paths. **Tx-Rate Estimation** gives a relative long-term average sending rate of the path, expressed as the proportion of the overall path quality used. **BW Estimation** indicates the real-time and smooth sending efficiency of each path, reflecting

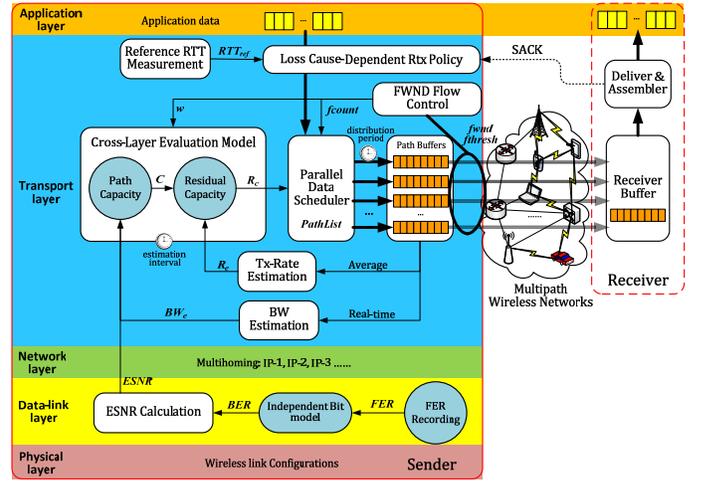


Fig. 1. CMT-CL/FD architectural design

the change of path's available bandwidth. Both estimations are performed by monitoring and interacting with the path buffers.

The newly designed **FWND Flow Control** module addresses the fairness problem, putting the emphasis on the trade-off between fairness and efficiency. The former refers to the friendly behavior towards single-path TCP flows, which would impose self-discipline on CMT's sending rate. The latter allows for a moderate growth of the sender window to enable good utilization of network bandwidth. More importantly, CMT-CL/FD deals with this trade-off by properly scheduling data transmission and balancing the load according to path congestion level. This conforms to the goals set for fairness in [22] and also enables high network utilization.

The **Cross-layer Evaluation Model** is employed to quantify each path's quality in terms of *Path Capacity*, computed based on the ESNR and bandwidth estimation results for each path. It is used to determine the potential of how much data can be sent via each path. Next, by subtracting the current transmission rate estimation and under the FWND constraints, *Residual Capacity* values are obtained to control the data distribution priority as well as the data amount allocated to each path.

Finally, based on the evaluation results, the **Parallel Data Scheduler** module distributes appropriate data amounts to path buffers iteratively. An independent buffer is dynamically allocated for each individual path at the sender side. Consequently, each path manages its own sender buffer and holds the ready-to-send data in that buffer. The data in path buffers will be transmitted to the receiver via the multipath network under FWND control. During transmission, the sender locally recognizes frame errors at data-link layer (e.g. by using MAC protocols), while the receiver reports packet loss at the transport layer to the sender by using selective acknowledgments (SACK). Frame errors are used for ESNR calculation in the evaluation loop, and the packet loss will be handled by **Loss-Cause Dependent Rtx Policy**. In contrast to standard SCTP which just halves the congestion window for any loss, this module analyzes the loss cause by looking at the *Residual Capacity* and decides how to adjust the congestion window.

Table I summarizes all the parameters used in the CMT-CL/FD. In the following sections, the functionality and algo-

TABLE I
PARAMETERS USED IN CMT-CL/FD

Symbol	Description
u	path estimation interval
BER	bit error rate
FER	frame error rate
$ESNR$	path Effective SNR
RTT_{ref}	path Reference RTT
R_e	path transmission rate estimation
BW_e	path bandwidth estimation
a, b	smoothing factors in moving average
Q, R	noise covariance in Kalman filter
f_{wnd}	friendliness window
f_{thresh}	friendliness threshold
f_{count}	path friendliness penalty count
α	fairness reward coefficient
β	fairness penalty coefficient
P_d	distribution period
T_{handle}	path handling ability
D_{max}	maximum distributed data amount in one P_d
C	Path Capacity
w	path fairness weight coefficient
R_c	path Residual Capacity
$PathList$	candidate path list sorted in descending order of R_c

gorithms behind these CMT-CL/FD modules are described and tested in details.

IV. CROSS-LAYER BASIC OPERATIONS

In the cross-layer design of CMT-CL/FD, the first question is how long the ESNR should be calculated for, carry the estimations and obtain the path quality. A similar approach to CMT-QA is employed [8] and a dynamical *estimation interval*, denoted by u is setup for each path. The estimation interval is loss-event triggered and statistically updated by means of a Confidence Interval. This estimation interval is also of interest in Path Activity. If any path has $u \leq RTO$ (Retransmission Time-Out), meaning that the loss occurs frequently, then the path is marked as “inactive”; otherwise, this path remains “active”. Both states have identical semantics as in standard SCTP [5]. Note that any inactive path is not allowed to participate in the following procedures.

A. ESNR Calculation at Data-Link Layer

The wireless channel is very unreliable and unpredictable with various types of noise, fading, distortion and interference. To counteract the negative impact of these aspects, various types of modulation, coding, diversity and equalization are currently applied. As a result, wireless communications are affected simultaneously by both negative and positive effects. In general, the metric SNR¹ is considered to be capable of integrating all these effects and giving a holistic evaluation of wireless communication quality. However, SNR measurement is highly inaccurate in real-life. Today’s commercial or experimental devices report signal strength² and noise floor level during the reception of the Physical Layer Convergence Protocol (PLCP) preamble and header that is operating at a

¹Sometimes, SNR is also referred to as Signal-to-Interference/Noise Ratio (SINR).

²The signal strength is often measured in form of the Received Signal Strength Indicator (RSSI).

basic/low rate. The data frame reception, which in contrast operates at some higher rate, is not taken into account in this direct-report SNR. In addition, this kind of SNR is not able to capture co-channel interference, frequency-selective fading and signal multipath effects [27].

This is reason behind using the concept of “Effective SNR” (*ESNR*) [29] instead of SNR, addressing the shortcomings of the direct-report SNR. The **ESNR Calculation** module is responsible for this ESNR management. ESNR relies on the relationship between Bit Error Rate (BER) and SNR in a simple AWGN channel. For any specific configuration of wireless communications, this relationship is certainly in one-to-one mapping format, denoted by $F()$, which can be expressed as a function, table or plot.

$$BER = F(SNR) \quad \text{for the AWGN channel} \quad (1)$$

The BER-SNR mapping is common in many types of wireless communications, reflecting how good the wireless communications are. This mapping differs for various wireless technologies [30]. For example, the Table 2 in [28] gave out some BER-SNR relationships in IEEE 802.11a/g for different modulations with a single antenna. If BER is known from the frame reception measurements, we can inverse eq. (1) and generalize the SNR for any channel type. This inversion defines ESNR, as in eq. (2):

$$ESNR = F^{-1}(BER) \quad \text{for any kind of channels} \quad (2)$$

where $F^{-1}()$ is the inverse function of $F()$. In this sense, we push all the negative effects into the influence of AWGN and converge all the positive effects in the relationship between BER and SNR.

Further, since BER direct measurement at physical layer has large overhead³, it is preferred to employ FER at data-link layer instead. The **ESNR Calculation** module collects error frame numbers and records FER, as in eq. (3):

$$FER = \frac{error_frame}{total_frame} \quad (3)$$

where *error_frame* is the number of data frames requiring retransmissions at data-link layer (e.g. due to link contentions) and *total_frame* is the total number of data frames sent in one estimation interval. For example in IEEE 802.11, the retransmitted data frames collection relies on MAC information. Then, a frame is successfully accepted if and only if every bit is correctly decoded. Hence, BER can be derived as in the independent-bit model [18], described in eq. (4):

$$FER = 1 - (1 - BER)^{8L} \\ \text{or } BER = 1 - \sqrt[8L]{1 - FER} \quad (4)$$

where L is the number of bytes in one data frame.

In summary, the **ESNR Calculation** module directly measures FER by counting the number of retransmitted data frames during the estimation interval. Then it calculates BER by using the independent-bit model, and further ESNR by making use of the configured BER-SNR relationship. Finally, this ESNR is reported to the transport layer.

³The BER measurement requires appending massive checking data redundancy to the frames, buffering large amounts of data and then conducting duplicative bit-by-bit comparisons.

Algorithm 1 Path Bandwidth Estimation

```

/* When it is ready to send DATA packets on any active path p */
k = k + 1;
p.length_sample = the total length of the sending DATA;
p.interval_sample = NOW - p.last_sending_time;
Send out the DATA packets;
p.last_sending_time = NOW;

/* Moving average to obtain one bandwidth sample */
p.length[k] = a * p.length[k - 1] + (1 - a) * p.length_sample;
p.interval[k] = b * p.interval[k - 1] + (1 - b) * p.interval_sample;
p.BW_sample[k] = p.length[k] / p.interval[k];

/* Kalman filter to smooth the samples */
/* Time Update step */
p.BW_prior = p.BW_e;
p.ev_prior = p.ev_post + Q;
/* Measurement Update step */
Kgain = p.ev_prior / (p.ev_prior + R);
corrector = Kgain * (p.BW_sample[k] - p.BW_prior);
p.BW_e = p.BW_prior + corrector;
p.ev_post = (1 - Kgain) * p.ev_prior;

```

where **NOW** is the current time fetched from the system clock. k is the current index, $k - 1$ is the previous-time index (e.g. $BW_sample[k]$ is the bandwidth sample of the current estimation, $BW_sample[k - 1]$ is the sample of at previous-time). a and b are smoothing factors in the moving average. Q and R are the two parameters (noise covariance) in Kalman filter and $Kgain$ denotes the Kalman gain.

B. Transport Layer Measurements and Estimations

1) Reference RTT Measurement. As SCTP allows replying SACK messages on different paths than those used by their corresponding DATA packets, and as often one SACK reports the reception of packets sent over multiple paths, the classic SRTT measurement results in incorrect delay estimations. Instead, SCTP's Heartbeat mechanism is used for path probing, which requires HEARTBEAT_ACK messages be replied over the same path as used by the HEARTBEAT packets [5]. Hence, the **Reference RTT Measurement** module employs this mechanism and changes the constant heartbeat interval (default 30s) to the dynamic estimation interval (u). At the beginning of each estimation interval, the sender probes the path by sending a pair of HEARTBEAT packets. When both HEARTBEAT_ACK packets are successfully received, two RTT values (denoted RTT_1 and RTT_2) are obtained for the two probes. This module takes the smaller one as the path *Reference RTT* (RTT_{ref}), as in eq. (5):

$$RTT_{ref} = \min \{RTT_1, RTT_2\} \quad (5)$$

This value will be used in the **Loss-Cause Dependent Rtx Policy** module.

2) Transmission Rate Estimation. For transport-layer information perception, the **Tx-Rate Estimation** module captures how fast the data is being sent over each path. Its purpose is to determine how much of *Path Capacity* is used and accordingly to enable the control of data distribution. During the SCTP data exchange, this module estimates regularly the transmission rate by computing the average sending rate in one estimation interval (u), as in eq. (6):

$$R_e = \frac{sendsize}{T_l - T_e} \quad (6)$$

where $sendsize$ is the successful data amount sent over the path during one estimation interval, T_e is the time when the

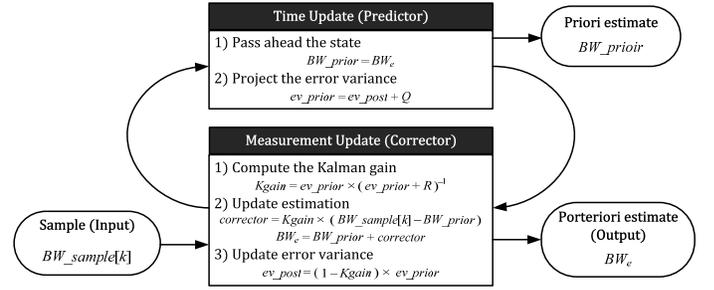


Fig. 2. Circulating steps of Kalman Filter

first DATA chunk enters the path buffer and T_l is the time when the last DATA chunk leaves the buffer.

3) Bandwidth Estimation. The rate estimation R_e is an time-average value, thereby somewhat stable and insensitive to path dynamic variations. CMT-CL/FD also requires tracking path bandwidth variations in real time and the **BW Estimation** module is in charge of this. The estimation result will be used to calculate the *Path Capacity*. Despite existing many methods for bandwidth estimation, as explained above, the fact that CMT allows sending SACK and DATA packets over different paths makes all the methods based on acknowledgment receipts not appropriate, including Packet Pair and Westwood+ [12]. Similarly, all the methods using normal RTT or RTO measurements, like TCP Vegas [31], are also not accurate in this case.

Instead, CMT-CL/FD focuses on the sending process and uses the principle of TIBET [32]. The estimated bandwidth is computed as the ratio between the average packet length and average inter-packet sending time. In order to reduce oscillations due to random sending behavior, the bandwidth samples are smoothed to give the result. Algorithm 1 presents in details the procedure of our bandwidth estimation. It includes a two-level filtering in the estimation process.

The first filtering involves moving average computation of both packet length and inter-packet sending time, fed with one-order samples. When there is a data sending opportunity, the sum length of DATA packets is accumulated and the time interval since the last sending is recorded. These values act as input into an iteration of the moving average computation in order to get one bandwidth sample.

The second filtering employs the Kalman filter [33], a well-known discrete-time recursive filter used to smooth the instantaneous values and obtain a more accurate estimated bandwidth. Fig. 2 shows the cycle composed of two steps illustrating how the Kalman filter is applied to the bandwidth samples. The *Time Update* step is responsible for handing ahead the state (bandwidth estimation) and error variance (estimation noise). This is called *Predictor* since it can priori predict the current state of bandwidth based on previous results. The *Measurement Update* step takes the new samples into account in order to obtain a more accurate posteriori estimation. It is called *Corrector* since it corrects the prediction of the current state relying on the new input samples. The output of the filter (BW_e of path p) is the BW estimation result after one-round Predictor-Corrector treatment.

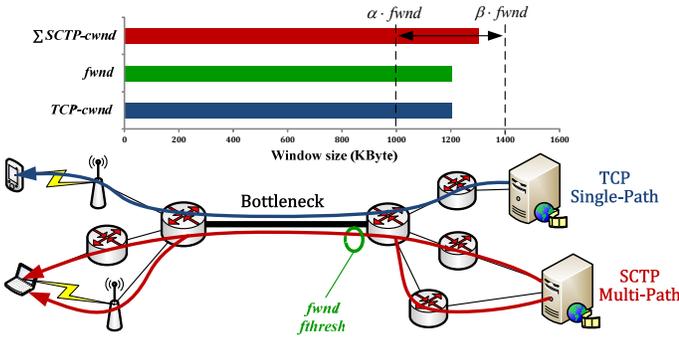


Fig. 3. Balance efficiency and fairness based on $fwnd$

V. FAIRNESS-DRIVEN FLOW CONTROL

CMT-CL/FD is fairness-driven to keep TCP-friendly. For example in Fig. 3, this implies the CMT flow should acquire similar bandwidth with that used by the TCP flow. Different from direct modifications of congestion window mechanism in [22], the **FWND Flow Control** module imitates the behavior of TCP congestion control and compares with the total sending rate as the reference for flow control.

We define $fwnd$ and its threshold $fthresh$ for the whole SCTP association. They have similar behaviors to TCP's (Reno) $cwnd$ and $ssthresh$ in relation to the congestion control. The Additive Increase and Multiplicative Decrease (AIMD) principle is applied, such that $fwnd$ increases stepwise when DATA packets are acknowledged, and decreases by half when any retransmission is needed. And $fthresh$ decides whether $fwnd$ executes the slow start or congestion avoidance algorithm. If there is data loss due to congestion, $fthresh$ is halved followed by decreasing $fwnd$. The identification of whether loss is due to congestion or wireless error will be provided by the **Loss-Cause Dependent Rtx Policy** module discussed later (see subsection VI-C). Additionally, we define the friendliness penalty count $fcnt$ for each path. Whenever data retransmission is required, $fcnt$ increases.

Algorithm 2 details the behaviors of $fwnd$, $fthresh$ and $fcnt$. In order to perform the trade off between fairness and efficiency, the joint effect of $fwnd$ and $fthresh$ should allow path $cwnd$ to grow sufficiently, but prevent excessive bandwidth occupation. As Fig. 3 shows, this can be achieved by maintaining the total sending rate (represented by the sum of all path $cwnd$ -s) in a rational interval centered at $fwnd$, as in eq. (7):

$$\sum_{i=1}^n cwnd_i \in [\alpha * fwnd, \beta * fwnd], \alpha < 1, \beta > 1 \quad (7)$$

where α and β are the reward and penalty coefficients, respectively.

When there is a change in the $cwnd$ of any path, CMT-CL/FD executes Algorithm 3 which composed the *Reward* and *Penalize* methods to update the balance between the total sending rate and $fwnd$ interval. If $\sum_{i=1}^n cwnd_i < \alpha * fwnd$, all path $cwnd$ -s are rewarded with additional increases to grow the efficiency of data transfer. More importantly, if $\sum_{i=1}^n cwnd_i > \beta * fwnd$, the path q with the largest $fcnt_q$ is penalized by reducing its aggressiveness. This path will

Algorithm 2 Friendliness Window Maintenance

```

/* When the SCTP association is established */
fwnd = 0; fthresh = ssthresh(init);
for (∀ active Path p)
  p.fcount = 0;
  fwnd = fwnd + cwnd(init);
end for
/* During data transfer */
if (there is an increase Δcwnd on p.cwnd of any active Path p)
  if (fwnd < fthresh)
    fwnd executes slow start algorithm;
  else
    fwnd executes congestion avoidance algorithm;
    Δcwnd = Δcwnd/n; /* suppress window in advance */
  end if
  p.cwnd = p.cwnd + Δcwnd;
end if
if (fast retransmission happens on any Path p)
  /* Loss reason analysis by Rtx policy */
  if (this retransmission is due to congestion)
    fthresh = max {fwnd/2, n * cwnd(init)};
    fwnd = fthresh;
  end if
  p.fcount = p.fcount + 1;
end if
if (time-out retransmission happens on any Path p)
  fthresh = max {fwnd/2, n * cwnd(init)};
  fwnd = n * cwnd(init);
  p.fcount = p.fcount + 2;
end if

```

where n is the number of active paths, MTU is the Maximum Transmission Unit and is assumed to be identical across all paths. $ssthresh(init)$ and $cwnd(init)$ return the initial values of SCTP path $ssthresh$ and $cwnd$, respectively.

halve its $cwnd_q$ as if fast retransmission happened, be forbidden from data distribution in the current distribution period (defined in subsection VI-B) and have its $fcnt_q$ decremented. After this penalization, if $\sum_{i=1}^n cwnd_i > \beta * fwnd$ still holds, another path is selected for subsequent penalty. By combining *Reward* and *Penalize* methods, CMT-CL/FD maintains the range defined in eq. (7) and performs well the trade-off between fairness and efficiency.

VI. DATA DISTRIBUTION AND LOSS HANDLING

A. Path Quality Determination

All the above-mentioned modules have acquired comprehensive information about active paths in cross-layer manner. The **Cross-layer Evaluation Model** modules process this information to obtain *Path Capacity* and determine the path quality (distribution priority) based on *Residual Capacity*.

We start from the famous Shannon Capacity theory. This theory has told a concise formula for the point-to-point AWGN channel capacity calculation presented in eq. (8):

$$C_{shannon} = BW_f * \log_2(1 + SNR) \quad (8)$$

where BW_f is the bandwidth at physical layer and SNR is the real signal-to-noise ratio. Then we make an abstraction from the end-to-end path to the point-to-point channel, as they are both transporting data in a transparent tunnel from one side to the other. The abstraction is reasonable: how much data can be transmitted over one channel is influenced by the width of this tunnel (BW_f) and the success rate of the transmission through the tunnel (indicated by SNR); the amount of data which can be transported through one path is affected by similar factors (bandwidth and loss rate). Therefore, the path should have similar capacity definition with that of the channel.

Consequently, in our cross-layer design, the bandwidth factor is obtained by BW estimation (BW_e) as described

Algorithm 3 Reward and Penalize Methods

```

/* When a change happens in any one cwnd */
cwnd_sum = the sum of cwnd in all active paths;
Re_sum = the sum of Re in all active paths;
/* Reward Method */
if (cwnd_sum < alpha * fwnd)
  for (forall active Path p)
    p.cwnd = p.cwnd + (fwnd - cwnd_sum) * p.Re/Re_sum;
  end for
end if
/* Penalize Method */
while (cwnd_sum > beta * fwnd) do
  fcount_max = 0; q = null; /* initial traverse variables */
  for (forall active Path p && p.cwnd > 4 * MTU)
    if (fcount_max < p.fcount)
      /* find the path with the largest fcount */
      q = p; fcount_max = p.fcount;
    else if (fcount_max == p.fcount && p.Re < q.Re)
      /* if fcount is equal, find the path with smaller Re */
      q = p;
    end if
  end for
  if (q == null) /* no path is selected */
    break;
  end if
  q.ssthresh = max {q.cwnd/2, 4 * MTU};
  q.cwnd = q.ssthresh;
  Stop data distribution on q in current distribution period;
  q.fcount = max {q.fcount - 1, 0};
  Update cwnd_sum to the sum of cwnd in all active paths;
end while

```

in subsection IV-B, and the loss rate factor is reflected by the Effective SNR ($ESNR$) as defined in subsection IV-A. Inspired by eq. (8), we define the *Path Capacity* C in an analogous form in eq. (9):

$$C = \frac{BW_e}{w} * \log_2(1 + ESNR) \quad (9)$$

$$w = \max \left\{ \frac{cwnd}{fwnd} + \Delta fcount, \beta \right\}$$

where $\Delta fcount$ is the increment of $fcount$ since the last distribution period (defined in subsection VI-B). The fairness weight coefficient w is applied for the consideration of friendly data dispatch: one factor depends on $cwnd$, $fwnd$ and $fcount$, and reflects the runtime transport fairness; the other factor is the penalty coefficient β , introduced in eq. (7), which reflects the aggressiveness of our flow control mechanism.

Further, since the path is using the capacity as the transmission rate R_e , then the *Residual Capacity* R_c for that path can be computed as in eq. (10):

$$R_c = C - R_e \quad (10)$$

Note that R_c can be positive or negative. Next, all candidate paths that stay active and pass the *Penalize* method in flow control (section V), are sorted in descending order of their R_c to form a new ordered path list *PathList*, such that the paths with larger *Residual Capacity* have higher priority for data distribution.

B. Parallel Data Distribution

The **Parallel Data Scheduler** module performs data distribution over multiple paths regularly over another dynamic time interval denoted *distribution period*. This distribution period (P_d) can be different from the estimation interval (u), enabling the cross-layer evaluation and data distribution to

proceed simultaneously but at different paces. At this stage, P_d is related to the handling ability of each path [8]. And the handling ability of path i is defined as the handling time of one path's $cwnd$, as in eq. (11):

$$T_{handle_i} = \frac{cwnd_i}{BW_{e_i}} \quad (11)$$

In order to take full usage of all the active paths, P_d is selected as the maximum of the handling times recorded for the n active paths, as in eq. (12).

$$P_d = \max \{T_{handle_1}, T_{handle_2}, \dots, T_{handle_n}\} \quad (12)$$

After that, the data from the application layer will be concurrently dispatched to all candidate paths listed in *PathList* during P_d . The maximum data amount D_{max} that can be distributed is limited by eq. (13).

$$D_{max} = \min \left\{ \sum_{i=1}^n (cwnd_i - outstanding_i), a_rwnd - \sum_{i=1}^n outstanding_i \right\} \quad (13)$$

where $outstanding_i$ is the data amount that is sent, but not acknowledged on path i and a_rwnd is the advertised receiver window declared in the newest SACK.

As already mentioned, each path manages the size of its buffer and independently transmits the data in its buffer. Therefore, the only task is to partition D_{max} amount of data and schedule it to path buffers to be sent out. Algorithm 4 details the progress of data distribution in CMT-CL/FD. It conforms to the priority order of *PathList* and distributes appropriate data amounts to paths based on R_c . If R_c is positive, meaning that the path has some potential to improve its throughput, then the scheduler keeps the current rising tendency of $cwnd$ and appends $R_c * P_d$ to $cwnd$. Otherwise, the scheduler is conservative to fill in the current $cwnd$ on top of the outstanding data. The distributed data amount scheduled to each path is then taken away from D_{max} and dispatched gradually to the selected path buffer whenever the path $cwnd$ allows.

When all D_{max} data is allocated to paths, the task of data distribution is completed in this P_d . Any other data dispatch is stopped although some low-priority paths may still have no data partitioned. Meanwhile, the paths continue sending the data in their buffers. When the current P_d ends, the scheduler re-computes the next distribution period according to eq. (12). A new iteration of Algorithm 4 will start after updating the *Path Capacity* C and *Residual Capacity* R_c according to eq. (9) and (10).

C. Loss Handling and Retransmission

If packet loss happens during data exchanges, the **Loss-Cause Dependent Rtx Policy** module analyzes the cause of the loss with the help of R_c , distinguishing wireless error from congestion loss. Then, it takes corresponding actions for loss recovery.

When a packet loss is detected by fast retransmission on one path, the sender identifies the reason: if $R_c > 0$, the path is under-used and the loss is considered to be due to wireless

Algorithm 4 Parallel Data Distribution

```

/* When PathList is determined in this distribution period  $P_d$  */
 $D_{total} = 0$  and initial  $D_{max}$  according to eq. (13);
for ( $\forall$  Path  $p$  ordered by  $PathList$ )
  if ( $p.R_c > 0$ ) /* potential for throughput increase */
     $p.D = \lfloor \frac{P_d}{p.T_{handle}} \rfloor * p.cwnd + p.R_c * P_d$ ;
  else /* full or overloaded */
     $p.D = \lfloor \frac{P_d}{p.T_{handle}} \rfloor * (p.cwnd - p.outstanding)$ ;
  end if
   $D_{total} = D_{total} + p.D$ ;
  if ( $D_{total} \leq D_{max}$ )
    Distribute  $p.D$  data amount to  $p$  whenever  $p.cwnd$  allows;
  else /* all  $D_{max}$  data was allocated */
    break;
  end if
end for

```

error. Consequently $cwnd$ is not adjusted, but $ssthresh$ is according to eq. (14).

$$ssthresh = \max \{R_e * RTT_{ref}, cwnd/2, 4 * MTU\} \quad (14)$$

where R_e is result of the Tx-Rate estimation, and RTT_{ref} is the *Reference RTT* described in subsection IV-B. Periodic Tx-Rate estimation is used here since the wireless error is random and instantaneous, so that the adjustment of $ssthresh$ should be related to the average sending rate. Otherwise, the path is over-used and the loss is due to congestion. Then both $ssthresh$ and $cwnd$ are updated as follows:

$$\begin{aligned} ssthresh &= \max \{BW_e * RTT_{ref}, cwnd/2, 4 * MTU\} \\ cwnd &= \min \{cwnd, ssthresh\} \end{aligned} \quad (15)$$

where BW_e is the result of BW estimation. Real-time BW estimation is used here since congestion means the load of the connection has changed significantly and the adjustment should depend on the most up-to-date estimation value.

For packet loss detected by time-out retransmission, we conservatively follow the handling of standard SCTP. This is as RTO has a relative large time value, so that its expiration indicates severe congestion or link failure.

$$\begin{aligned} ssthresh &= \max \{cwnd/2, 4 * MTU\} \\ cwnd &= MTU \end{aligned} \quad (16)$$

After the above adjustments, the path with the largest $cwnd$ is selected to retransmit the lost packets as soon as possible (i.e. before all the other packets in the path buffer). This choice is to enable the best retransmission success probability [6].

VII. PERFORMANCE EVALUATION

This section evaluates the throughput performance of CMT-CL/FD for conventional FTP-like data transmissions and real-time video delivery, respectively. In our tests, we have tested two versions of our newly proposed solution against three important previously proposed schemes. The two versions of our solution are: CMT-CL/FD, in which both Cross-layer Evaluation Model and FWND Flow Control modules are used and CMT-CL, in which the Cross-layer Evaluation Model module is deployed only.

The testing-based comparison will be made against one of our previous works CMT-QA [8] and two MPTCP-based solutions NC-MPTCP [23] and FMTCP [26] in order to reveal the advantages provided by CMT-CL and CMT-CL/FD.

A. Simulation Setup

The simulation is carried out in the Network Simulator version 2.35 (NS-2). Based on prior experiments, we set the best combination of parameters for CMT-CL/FD configuration: in BW estimation $a = b = 0.99$, $Q = 0.01$ and $R = 1$; in flow control $\alpha = 0.5$ and $\beta = 1.5$. A heterogeneous wireless network topology has been designed for this testing (Fig. 4). The SCTP sender and receiver access the network through three wireless access links and then inter-connect with each other through the wired core network. Therefore, there are three different paths (denoted A, B and C) between them, and we make the wireless access technology in each path identical for both sides. The wireless channel model considered is AWGN. Regarding WiMAX, the carrier frequency, channel bandwidth and frame duration are 2.5 GHz, 10 MHz, and 5 millisecond, respectively. The same parameters for WCDMA are 2 GHz, 5 MHz, 10 millisecond, respectively. The configuration of the three paths is shown in Table II.

To simulate frame loss at data-link layer, we attach each wireless link with the *Uniform loss model* to represent distributed loss due to random contention, wireless interference or link handoff; plus the *Two-State Markov loss model* to represent infrequent continuous loss due to signal fading, transient failure or stream burst. To simulate congestion loss at transport layer, we give each link a queue limit, such that any saturated link would drop new incoming packets. In real deployment, the data rate is mainly constrained by the narrow bandwidth of the wireless access links, so the wired links' bandwidth is set to 100 Mbps in order not to limit the transmission in the core network (but to introduce delay).

Moreover, we inject cross traffic to simulate the Internet background traffic. The cross traffic is generated by a Variable Bit Rate (VBR) sender with Pareto distribution, which then flows into three paths to reach a VBR receiver. Based on the results in [19], the packet sizes in cross traffic are chosen as follows: 49% are 44 bytes, 1.2% are 576 bytes, 2.1% are 628 bytes, 1.7% are 1300 bytes and 46% are 1500 bytes. 90% of these packets are carried by TCP and the rest 10% are over UDP. The aggregate cross traffic on each path varies randomly between 0–50% of the access link bandwidth.

For the BER-SNR relationship in the **ESNR Calculation** module, we have separately configured the three types of wireless links in Matlab (R2013a) and then carried out experiments, in order to obtain BER-SNR tables. This methodology was extensively applied for wireless channel modeling in simulations, such as for instance in [34] and therefore, the tables are reliable and reflect the characteristics of the channels. Hence we use these tables to look up the corresponding ESNR for FER mapping. The receiver buffer is set to 256 KByte, the sender buffer is set large enough, and other parameters use the default values. All simulation results presented are average values of 100 trials with confidence level of 95%, which makes the effect of cross traffic on different schemes be general and not influenced by any stochastic factors.

For real-time video delivery comparison, the High-Definition (HD) video is considered in the testing and video bit rate is set 1Mbps, according to [35]. We use the Peak Signal-

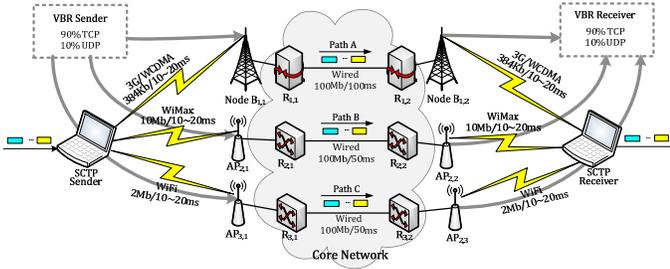


Fig. 4. Heterogeneous wireless network topology used in the bulk-data transmission

to-Noise Ratio (PSNR) metric to measure video quality, and estimate it according to eq. (17) [36].

$$PSNR = 20 \cdot \log_{10} \left(\frac{MAX_Bitrate}{\sqrt{(EXP_Thr - CRT_Thr)^2}} \right) \quad (17)$$

where $MAX_Bitrate$ is the average bit rate of the video stream as resulted from the encoding process, EXP_Thr is the average throughput expected from the delivery of the video stream over the network and CRT_Thr denotes the actual throughput measured during delivery. Both $MAX_Bitrate$ and EXP_Thr are set to 1Mbps in the simulations.

B. Simulation Results

We fix the loss rate of Path A ($p_A = 0.02$) and vary those of the other two paths. Fig. 5 compares average throughput when delivering the FTP-like data with CMT-QA, CMT-CL, CMT-CL/FD, NC-MPTCP and FMTCP, respectively. Within each interval in the figure (alternatively colored with white and gray), the loss rate of Path B (p_B) is fixed while the loss rate of Path C (p_C) varies from 0 to 0.1. Between intervals, p_B takes different values from 0.01 to 0.04.

As the figure shows, the throughput decreases with the increase in the link loss probability for all the five mechanisms. However, CMT-CL shows significant improvement compared with other mechanisms, since its scheduling mechanism is more superior based on cross-layer cooperation. This reveals the **Cross-layer Evaluation Model** is capable of acquiring first-hand information from lower layers and then obtains more accurate path quality at transport layer. Furthermore, our **BW Estimation** module is also able to flexibly capture the available bandwidth and assess path quality. The delivery performance associated with both NC-MPTCP and FMTCP is worse than that of CMT-QA at low loss rates. When the loss rate is low and with a large receiver buffer, the advantages of network coding could not be fully taken advantage of while the redundant packets for encoding still waste some of the available bandwidth.

Furthermore, CMT-CL/FD tends to have less throughput variations compared with CMT-QA, CMT-CL, NC-MPTCP and FMTCP. The reason is that the **FWND Flow Control** module has suppressed the radical CMT behavior and provided it with a moderate rate increase, in order to alleviate network overload and congestion. Further, by using the cross-layer evaluation, CMT-CL/FD is still efficiency-oriented. At high loss rates ($p_B \geq 0.04$), the throughput of CMT-CL/FD becomes close to that of CMT-QA, NC-MPTCP and FMTCP.

TABLE II
PATH PARAMETER SETTINGS IN THE SIMULATION

Parameters	Path A	Path B	Path C
Wireless technology	WCDMA	IEEE 802.16	IEEE 802.11
Access bandwidth	384Kbps	10Mbps	2Mbps
Access link delay	10-20ms	10-20ms	10-20ms
Access link queue limit	80	50	50
Uniform loss rate	0.02	0.01-0.04	0-0.1
Markov loss rate	0.01	0.01	0.01
Core network delay	100ms	50ms	50ms

The reason behind this is the CMT-CL/FD's **Parallel Data Scheduler** module can assign data to different path based on the **Cross-layer Evaluation** and **BW Estimation** modules cooperation to decrease the reordered packets.

Fig. 6 presents the comparison results of average video quality, expressed in terms of PSNR (dB). In order to better illustrate the comparison, the results are presented only when the loss rate of Path B (p_B) is fixed while the loss rate of Path C (p_C) is from 0 to 0.03. As the figure shows, CMT-CL outperforms other four methods in all the different lossy situations studied; the difference is very much in favour of CMT-CL when increasing the packet loss probability. This is because apart from its more sophisticated scheduling ability, CMT-CL also benefits from the **Loss-Cause Dependent Rtx Policy** which uses cross-layer cooperation to give more accurate suggestions for loss handling. For example, in the case when p_B is 4% and p_C is 3%, the average PSNR of CMT-QA, CMT-CL/FD, NC-MPTCP and FMTCP are 22.73 dB, 22.68 dB, 22.70 dB and 22.71 dB, respectively, but the average PSNR of CMT-CL is as high as 24.75 dB.

Moreover, since CMT-CL/FD is trying to be friendly towards the background traffic, by cutting the congestion window down and reducing the transmission rate, it is natural to have lower performance in terms of PSNR than CMT-QA, CMT-CL, NC-MPTCP and FMTCP at low loss rates. For example taking $p_B = 0.01$ and $p_C = 0$, the average PSNR when CMT-QA, CMT-CL, NC-MPTCP and FMTCP are employed in turn are 36.00 dB, 36.42 dB, 35.34 dB and 35.50 dB, respectively, while the average PSNR for CMT-CL/FD is slightly lower: 35.18 dB. However, the PSNR of CMT-CL/FD tends to become close to that of CMT-QA, NC-MPTCP and FMTCP, especially as p_B and p_C are larger. This is because CMT-CL/FD applies a cross-layer evaluation for its data scheduling, which helps. In addition, the **Loss-Cause Dependent Rtx Policy** module provides support to handle wireless errors and have better adaptation in wireless networks.

When video buffer underflow occurs, video playback stops until enough video data is buffered [37]. Therefore the video buffer underflow metric can act as a temporal video quality metric to describe the characteristics of the concurrent video transmission over multiple paths. Fig. 7 compares testing results in terms of the number of video buffer underflows for CMT-QA, CMT-CL, CMT-CL/FD, NC-MPTCP and FMTCP, respectively. In order to illustrate the comparison, the results are presented when $p_A = 0.02$, while p_B varies from 0.01 to 0.04, and p_C takes different values from 0 to 0.03. As the figure shows, the number of video buffer underflows increases with the increase in the link loss probability for all five mechanisms. However, due to its highly efficient **Cross-layer Evaluation Model** and **BW Estimation** module, CMT-CL

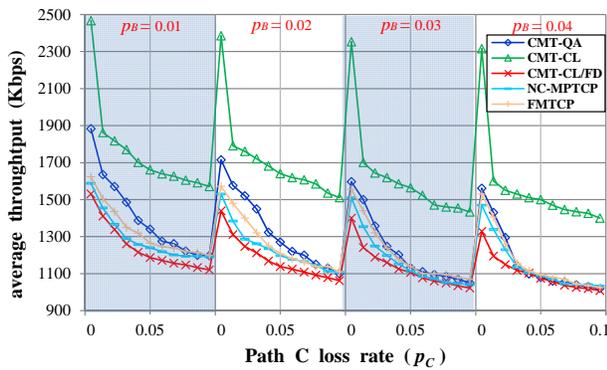


Fig. 5. FTP-like data transmission comparison

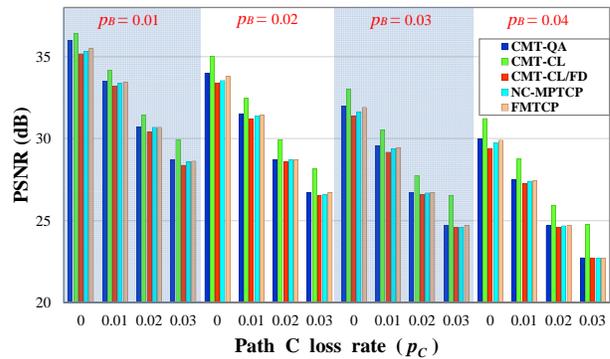


Fig. 6. Comparison of average PSNR (dB)

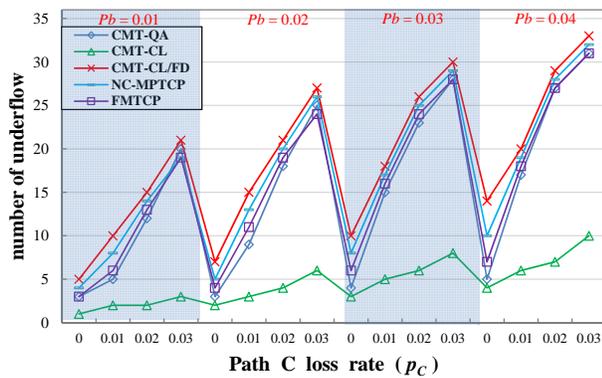


Fig. 7. Comparison of video buffer underflow

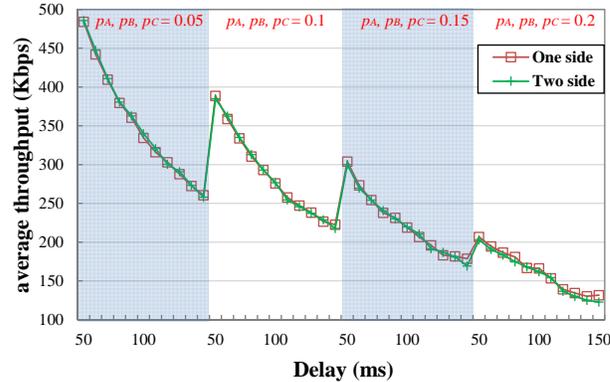


Fig. 8. Investigating impact of receiver side's loss rate and delay

generates less buffer underflow events and requires decreased reordering than other four schemes. Furthermore, CMT-CL/FD presents more video buffer underflows than CMT-QA, NC-MPTCP and FMTCP. This is because CMT-CL/FD tries to remain fair to competing TCP flows under the control of its **FWND Flow Control** module. However, benefiting from the **Cross-layer Evaluation Model** and **Loss-Cause Dependent Rtx Policy**, the difference between the number of underflows when compared with NC-MPTCP and FMTCP becomes smaller with increasing loss rate.

In order to evaluate the impact of receiver side's wireless channel state on our scheme, we designed one side and two side cases by using the topology illustrated in Fig. 4. The one side case denotes that the loss occurs in the wireless access links of the sender only. The two side case considers that the loss occurs in the wireless access links of both sender and receiver, equally. However, the total loss rate of two side case is equivalent to that of the one side case. For example, when p_B is 0.1, in the one side case, the sender and receiver side's loss rates are set to 0.1 and 0, respectively. In the two side case, both sender and receiver sides' loss rates are set to $0.051317 \approx 1 - \sqrt{1 - 0.1}$ according to the probability of independent lose rate events between sender and receiver. In the testing, the loss rates of p_A , p_B and p_C all varies 0.05, 0.1, 0.15 and 0.2. The delay of wireless access links of the receiver changes from 50 ms to 150 ms. Fig. 8 shows that the two cases achieve nearly the same throughput when the loss rate is less than roughly 0.15. When the loss rate increases to 0.2 and the delay increased to 140 ms, the one side case obtains a slightly higher throughput than that in the

two side case. The results show that our scheme can deal with the receiver side's different wireless channel states when the quality of receiver side's wireless channel is not too bad. It also shows that it is useful to consider in future work applying a cross-layer mechanism at the receiver to further improve the CMT-CL/FD performance and provide additional support in difficult wireless environments with very high loss rates and large delays.

VIII. TCP FAIRNESS TESTING

We consider two topologies for the fairness test, each having two paths (denoted A and B) directing the multipath flow alongside TCP (Reno) flows. The first topology (Fig. 9) involves a *two-path aggregate bottleneck*, meaning that the two SCTP paths share the same link with two TCP flows to form a bottleneck. The second topology (Fig. 10) considers a *one-path aggregate bottleneck*, where only one path (Path A) encounters a TCP flow, while the other path (Path B) is left independent.

To better reflect the TCP-friendly behavior of the two schemes, on the one hand, we render reasonably good and symmetrical wireless access channels. Both TCP and SCTP endpoints access the network via WiFi (IEEE 802.11) links configured with 2 Mbps bandwidth, 10 ms delay and 0.01 Uniform loss model and the receiver buffer set to the default 64 KB. On the other hand, the wired bottleneck link bandwidth is set to values between 0.5 Mbps and 2 Mbps in these tests, such that it is always an insufficient resource and causes congestion when shared by the flows. All the other settings are identical to those described in the previous tests.

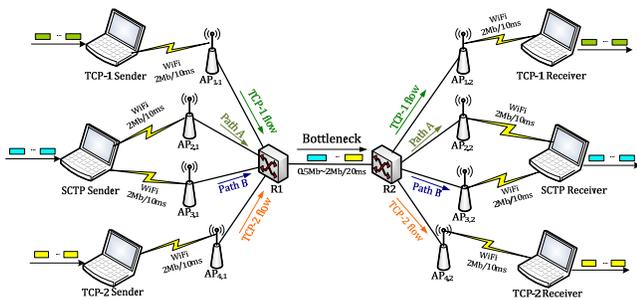


Fig. 9. Two-path aggregate bottleneck

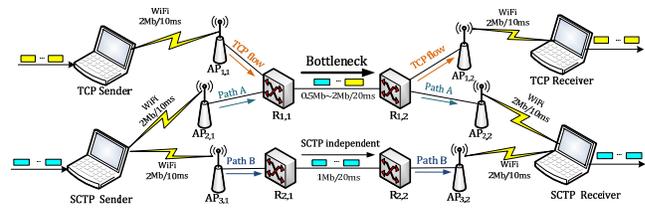
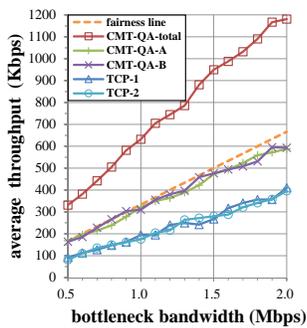
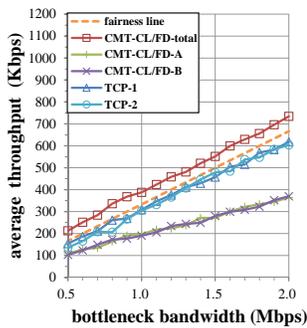


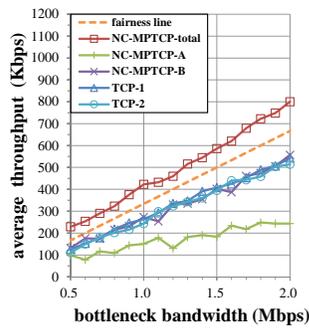
Fig. 10. One-path aggregate bottleneck



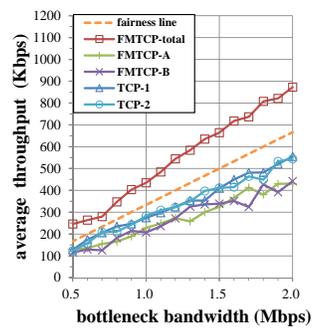
(a) CMT-QA



(b) CMT-CL/FD



(c) NC-MPTCP



(d) FMTCP

Fig. 11. Average throughput for different bandwidth values of the two-path aggregate bottleneck

A. Two-Path Aggregate Bottleneck Scenario

As shown in Fig. 9, there are two TCP flows (denoted TCP-1 and TCP-2) in addition to the multipath flow. The level of fairness for the multipath flow is assessed by observing the bandwidth share of the bottleneck. Obviously, the ideal case is that each flow occupies 1/3 bottleneck bandwidth indicated by the fairness line in the plots. Furthermore, it is necessary to check the fairness against a TCP flow starting earlier and another later than the multipath flow. Hence, we make TCP-1 start at 0s, Sctp flow at 50s and TCP-2 at 100s, then record the stable bandwidth share at 300s.

As shown in Fig. 11(a), CMT-QA is very aggressive without TCP-friendly flow control. For example in the 1Mbps bottleneck bandwidth case, CMT-QA uses about 0.64Mbps, leaving only 0.36Mbps to the TCP (in fact 0.35Mbps due to some loss). This is more than two times to the TCP share, because each Sctp individual path outperforms the TCP single path due to fast and abundant arrivals of acknowledgements, which encourages Sctp paths' *cwnd* to grow rapidly. Another factor is CMT-QA's component strategies, such as Selective ACK, path-quality-aware scheduling and better retransmission policy in comparison with TCP Reno. (Just like TCP SACK and TCP Westwood outperform TCP Reno.)

As shown in Fig. 11(b), it is obvious that CMT-CL/FD is generally successful in applying TCP-friendly flow control. The Sctp-total flow is very close to the fairness line, and the throughput of two paths are almost equal in terms of dividing the bandwidth share due to the topology symmetry. Although it may occupy slightly more bandwidth than TCP flows, the extra share is accounted for some good features of CMT-CL/FD (e.g. Selective ACK). Thus, it can be concluded that CMT-CL/FD has suppressed the radical CMT transmission behavior, placing

an upper limit on the throughput and making it moderate in order to prevent network overload and congestion loss, which may be frequent in aggressive CMT-QA. As shown in Fig. 11(c) and Fig. 11(d), although both NC-MPTCP and FMTCP have TCP-friendly flow control, network coding they used will filter some packet loss events, which are very important signals for the TCP-friendly mechanism. This leads to both NC-MPTCP and FMTCP being more aggressive than CMT-CL/FD.

To demonstrate the importance of the **Parallel Data Scheduler** module in this test, we change the parameters of Path B wireless link to 1Mbps/100ms with Uniform loss rate 0.05 (in poor condition), in order to break the symmetry of the two paths. When the bottleneck link is set to 1Mbps/20ms, Fig. 12 shows the throughput variations of four schemes. CMT-QA distributes more data sending to the poor path (Path B). As a result, CMT-QA thus suffers performance degradations due to data reordering and loss. CMT-QA should be not capable of best-path selection as well as accurate data scheduling in this resource-insufficient situation. In contrast, CMT-CL/FD does not only preserve the efficiency by scheduling more data to the better path (Path A), but also remains very good in terms of fairness by using about 1/3 of the shared bottleneck bandwidth (368Kbps). Therefore, as the **FWND Flow Control** module maintains well the TCP-friendly behavior, the **Parallel Data Scheduler** module is the key contributor to the high efficiency in the existence of path diversity within wireless networks. Although, NC-MPTCP and FMTCP can still achieve good fairness, they need more time to converge to the desired fairness level. This is an issue in the case of fluctuating traffic.

Fig. 13 illustrates the comparison results of average video quality, expressed in terms of PSNR (dB) when CMT-QA,

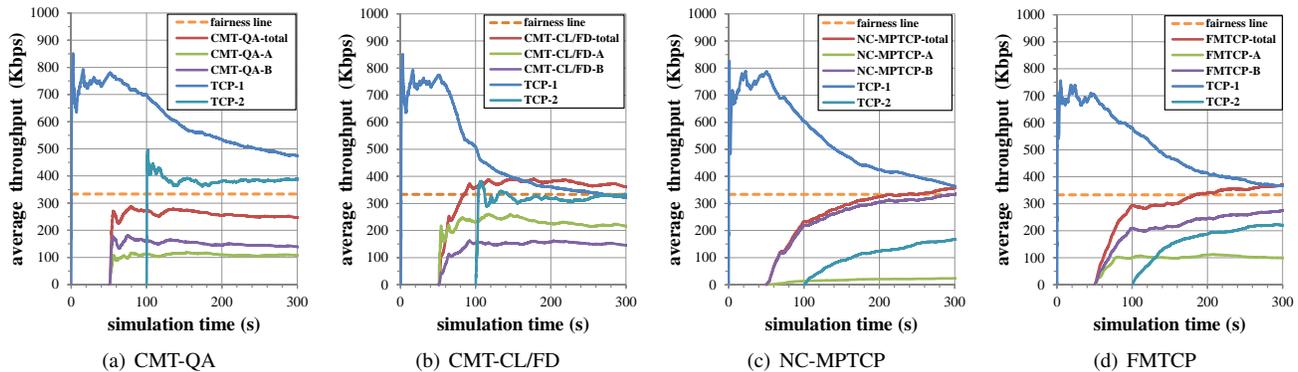


Fig. 12. Throughput variations for breaking the symmetry in the two-path aggregate bottleneck (1Mbps/20ms)

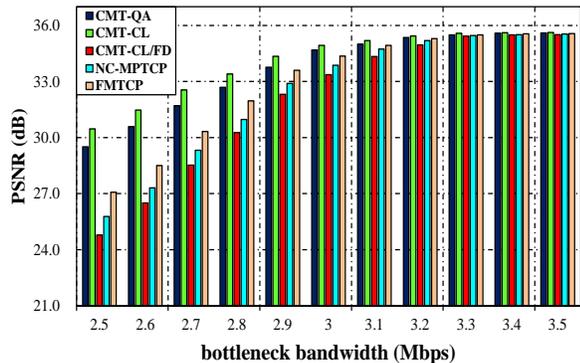


Fig. 13. Video delivery comparison in the two-path aggregate bottleneck scenario

CMT-CL, CMT-CL/FD, NC-MPTCP and FMTCP are used in turn. In this case, the receiver buffer is set to 256 KB. The x -axis shows the bottleneck bandwidth varying from 2.5 Mbps to 3.5 Mbps, and the y -axis is associated with the PSNR value. As the figure shows, the conservative behavior of the CMT-CL/FD’s FWND Flow Control influences directly the PSNR (dB) which is less than that of other mechanisms. This is actually the cost for deploying a TCP-friendly behavior. We argue that this is acceptable to the users and worthy for TCP fairness since CMT-CL/FD can obtain the same Mean Opinion Score (MOS) as that of CMT-QA does, according to the relationship between MOS and PSNR mentioned in Table 1 and [38]. When the bottleneck bandwidth increases more than 3.2 Mbps, the average PSNR (dB) differences between all the mechanisms become smaller. The reason is the available bandwidth is close to the video rate of 1 Mbps.

B. One-Path Aggregate Bottleneck Scenario

Next, we consider the one-path aggregate bottleneck scenario (Fig. 10). The independent path (Path B) is passing through a limited 1Mbps/20ms wired route, while the bottleneck link is shared by Path A and one TCP flow. We apply three TCP fairness-related goals recommended by IETF in [22] to discriminate whether the multipath flow is TCP-friendly. Let F_{total} and F_{TCP} denote the throughput of multipath (total) and TCP flows, F_A and F_B denote the throughput on Path A and Path B of the multipath flow, \hat{F}_A and \hat{F}_B denote the expected throughput supposed that a single-path TCP flow

would get on Path A and Path B, respectively. In this scenario, the IETF fairness goals can be quantified as follows:

- Improve throughput: “A multipath flow should perform at least as well as a single path flow would on the best of the paths available to it.”

$$F_{total} \geq \max \{ \hat{F}_A, \hat{F}_B \} \quad (18)$$

- Do no harm: “A multipath flow should not take up more capacity from any of the resources shared by its different paths than if it were a single flow using only one of these paths.”

$$F_A \leq \hat{F}_A, F_B \leq \hat{F}_B \quad (19)$$

- Balance congestion: “A multipath flow should move as much traffic as possible off its most congested paths, subject to meeting the first two goals.”

$$F_{TCP} > \hat{F}_A > F_A \quad (20)$$

Fig. 14 plots the average throughput against the bottleneck bandwidth (0.5–2Mbps) for CMT-QA, CMT-CL, CMT-CL/FD, NC-MPTCP and FMTCP. \hat{F}_A and \hat{F}_B are represented by the dotted lines of TCP-A and TCP-B. The solid blue line of TCP is standing for F_{TCP} in the test. One can find that CMT-CL/FD meets the three goals formalized in eq. (18)–(20), indicating that it achieves TCP-friendly fairness. CMT-CL/FD makes appropriate concession to the TCP flow on Path A. It occupies on average 35% of the bottleneck link bandwidth. More importantly, CMT-CL/FD is able to detect the path congestion level by the dual knowledge from **BW Estimation** and **Tx-Rate Estimation** modules, then intelligently shifts the load from Path A to Path B to reduce congestion by making use of the collaboration between the **Cross-layer Evaluation Model** and **Parallel Data Scheduler** modules. CMT-QA does not focus on TCP-friendly data transmission, but instead it greedily grows the sending rate by relying on multipath advantages and pushes away the TCP flows with which it shares the delivery path. NC-MPTCP also achieves good fairness, but the efficiency of PATH A is lower than that of CMT-CL/FD when the bottleneck bandwidth is greater than 1.0 Mbps. FMTCP is aggressive to TCP flow due to the same reason mentioned in the Two-Path case already described. By analyzing the performance of the four schemes in the two bottleneck topologies, CMT-CL/FD demonstrates an outstanding behavior as it best balances fairness and efficiency.

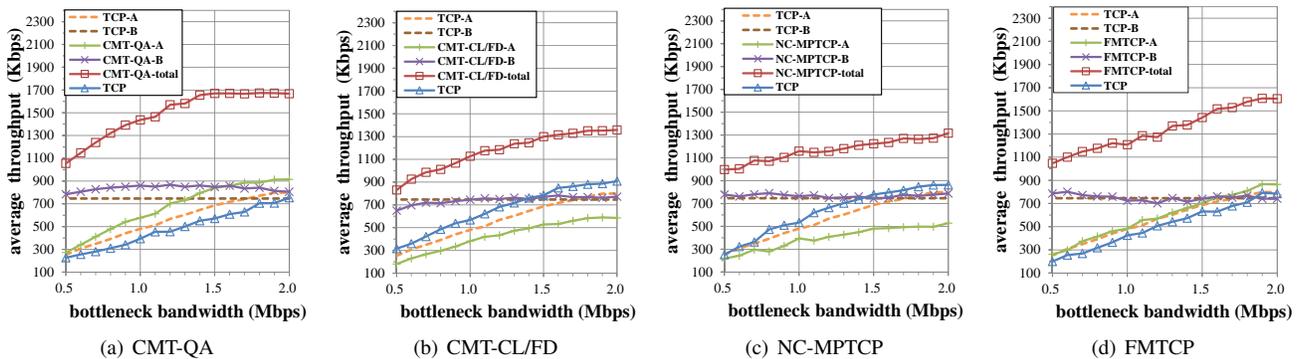


Fig. 14. Average throughput for different bandwidth values of the one-path aggregate bottleneck

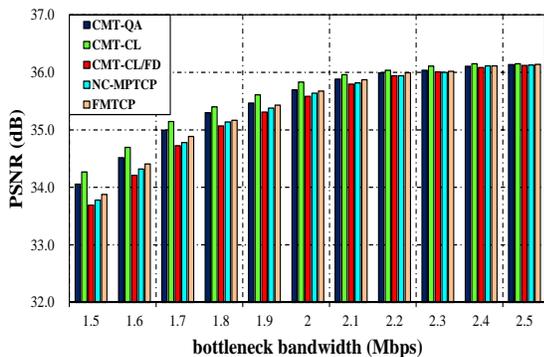


Fig. 15. Video delivery comparison in the one-path aggregate bottleneck scenario

Fig. 15 plots the average PSNR (dB) against the bottleneck bandwidth (1.5–2.5Mbps) for the CMT-QA, CMT-CL, CMT-CL/FD, NC-MPTCP and FMTCP mechanisms. CMT-CL achieves highest PSNR in all cases. The differences between CMT-CL/FD and other solutions decrease with the increase in bottleneck bandwidth. For example the PSNR difference between CMT-CL/FD and CMT-QA, and FMTCP is 0.37 dB and 0.19 dB, respectively, for 1.5 Mbps bottleneck bandwidth. However, with the bottleneck bandwidth increasing to 2 Mbps, the PSNR differences between CMT-CL/FD and CMT-QA, and FMTCP decrease to 0.11 dB and 0.09 dB, respectively. Moreover, it can also be observed that CMT-CL/FD achieves the same MOS as that of other mechanisms in all cases in this test scenario. It can be concluded that CMT-CL/FD remains fair to the competing TCP flows, while still satisfying users' high quality of experience expectations for video streaming services. The PSNR differences between all the mechanisms become smaller when the bottleneck bandwidth is more than 2.1 Mbps, which is because the available bandwidth is sufficient to support the video rate of 1 Mbps.

IX. CONCLUSION AND FUTURE WORK

This paper proposes a novel SCTP-based Cross-Layer and Fairness-Driven CMT scheme (CMT-CL/FD) for parallel video transfer over heterogeneous wireless networks. Its cross-layer evaluation model relies on both ESNR calculation at data-link layer and rate/bandwidth estimation at transport layer. An innovative window-based mechanism is also proposed for achieving TCP-friendly fairness. A loss-cause

dependent retransmission policy is used for differentiated loss recovery due to congestion and wireless errors, respectively. Simulation results show how the proposed cross-layer evaluation mechanism is able to improve the throughput significantly, with more accurate path quality and superior data scheduling ability, compared to state-of-the-art solutions, including CMT-QA, NC-MPTCP and FMTCP. Additionally, the more accurate data scheduling solution employed balances well the TCP-friendly fairness and video delivery efficiency. Our future work will consider employing a cross-layer mechanism at the receiver in order to further improve the CMT-CL/FD performance, especially in difficult wireless environments with very high loss rates and large delays.

REFERENCES

- [1] C. Xu, F. Zhao, J. Guan, H. Zhang and G.-M. Muntean, "QoE-driven User-centric VoD Services in Urban Multi-homed P2P-based Vehicular Network," *IEEE Trans. on Vehicular Technology*, vol.62, no.5, pp.2273-2289, 2013.
- [2] L. Zhou, Z. Yang, Y. Wen, J.P.C. Rodrigues, "Distributed Wireless Video Scheduling with Delayed Control Information," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.24, no.5, pp.889-901, May 2014.
- [3] C. Xu, S. Jia, L. Zhong, H. Zhang and G.-M. Muntean, "Ant-Inspired Mini-Community-Based Solution for Video-On-Demand Services in Wireless Mobile Networks," *IEEE Transactions on Broadcasting*, vol.60, no.2, pp.322-335, Jun. 2014.
- [4] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang and G. Muntean, "Performance-Aware Mobile Community-based VoD Streaming over Vehicular Ad Hoc Networks," *IEEE Trans. on Vehicular Technology*, vol.PP, no.99, pp.1-1, June 2014.
- [5] R. Stewart, "Stream Control Transmission Protocol," *RFC 4960*, IETF, Sep. 2007.
- [6] J. R. Iyengar, P. Amer and R. Stewart, "Concurrent Multipath Transfer using SCTP Multihoming over Independent End-to-End Paths," *IEEE/ACM Trans. on Networking*, vol.14, no.5, pp.951-964, Oct. 2006.
- [7] P. Amer, M. Becke, T. Dreiholz, N. Ekiz, J. Iyengar, P. Natarajan, R. Stewart and M. Tuexen, "Load Sharing for the Stream Control Transmission Protocol (SCTP)," *draft-tuexen-tsvwg-sctp-multipath-08.txt*, IETF, work in progress, Mar. 2014.
- [8] C. Xu, T. Liu, J. Guan, H. Zhang and G. Muntean, "CMT-QA: Quality-aware Adaptive Concurrent Multipath Data Transfer in Heterogeneous Wireless Networks," *IEEE Trans. on Mobile Computing*, vol.12, no.11, pp.2193-2205, Sept. 2013.
- [9] C.-M. Huang, M.-S. Lin, "Multimedia streaming using partially reliable concurrent multipath transfer for multihomed networks," *IET Communications*, vol.5, no.5, pp.587 - 597, Apr. 2011.
- [10] J. Baek, P. Fisher, M. Jo, H. Chen, "A Lightweight SCTP for Partially Reliable Overlay Video Multicast Service for Mobile Terminals," *IEEE Transactions on Multimedia*, vol. 12, no. 7, pp.754-766, Nov. 2010.
- [11] T. D. Wallace and A. Shami, "Concurrent Multipath Transfer using SCTP: Modelling and Congestion Window Management," *IEEE Transactions on Mobile Computing*, vol.PP, no.99, Feb. 2014.

- [12] F. Perotto, C. Casetti and G. Galante, "SCTP-based Transport Protocols for Concurrent Multipath Transfer," in *Proc. IEEE WCNC*, Mar. 2007.
- [13] C. Li, H. Xiong, J. Zou and C. W. Chen, "Distributed Robust Optimization for Scalable Video Multirate Multicast Over Wireless Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.22, no.6, pp.943-957, Feb. 2012.
- [14] M. Kserawi, S. Jung, D. Lee, J. Sung and J. K. Rhee, "Multipath Video Real-Time Streaming by Field-Based Anycast Routing," *IEEE Transactions on Multimedia*, vol.16, no.2, pp.533-540, Feb. 2014.
- [15] Z. Zhu, S. Li and X. Chen, "Design QoS-Aware Multi-Path Provisioning Strategies for Efficient Cloud-Assisted SVC Video Streaming to Heterogeneous Clients," *IEEE Transactions on Multimedia*, vol.15, no.4, pp.758-768, June 2013.
- [16] J. Fitzpatrick, S. Murphy, *et al.*, "Using cross-layer metrics to improve the performance of end-to-end handover mechanisms," *Computer Communications*, vol.32, no.15, pp.1600-1612, Sep. 2009.
- [17] Y. Taenaka, S. Kashiara, K. Tsukamoto, S. Yamaguchi and Y. Oie, "An implementation design of a cross-layer handover method with multi-path transmission for VoIP communication," *Ad Hoc Networks*, vol.13, Part B, pp.462-475, Feb. 2014.
- [18] Y. Cao, C. Xu, J. Guan, J. Zhao and H. Zhang, "Cross-layer Cognitive CMT for Efficient Multimedia Distribution over Multi-homed Wireless Networks," in *Proc. IEEE WCNC*, April 2013.
- [19] W. John, "Characterization and Classification of Internet Backbone Traffic," PhD Thesis, Dept. of Computer Science and Engineering, Chalmers Univ. of Tech., Gothenburg, Sweden, Feb. 2010.
- [20] S. Floyd, M. Handley, J. Padhye and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," *RFC 5348*, IETF, Sep. 2008.
- [21] M. Scharf and T. Banniza, "MCTCP: A Multipath Transport Shim Layer," in *Proc. IEEE Globecom*, Dec. 2011.
- [22] C. Raiciu, M. Handly and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols," *RFC 6356*, IETF, Oct. 2011.
- [23] M. Li, A. Lukyanenko and Y. Cui, "Network Coding Based Multipath TCP," in *Proc. IEEE INFOCOM Workshop*, Mar. 2012.
- [24] V. Sharma, S. Kalyanaraman, *et al.*, "MPLoT: A transport protocol exploiting multipath diversity using erasure codes," in *Proc. IEEE INFOCOM*, April 2008.
- [25] Y. Hwang, B. Obele and H. Lim, "Multipath transport protocol for heterogeneous multi-homing networks," in *Proc. ACM CoNEXT Student Workshop*, Nov. 2010.
- [26] Y. Cui, L. Wang, *et al.*, "FMTCP: A Fountain Code-Based Multipath Transmission Control Protocol," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, Jan. 2014.
- [27] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. Ni, "CSI-Based Indoor Localization," *IEEE Trans. on Parallel and Distributed Systems*, vol.24, no.7, pp.1300-1309, July 2013.
- [28] D. Halperin, W. Hu, A. Sheth and D. Wetherall, "Predictable 802.11 Packet Delivery from Wireless Channel Measurements," *ACM SIGCOMM CCR*, vol.40, no.4, pp.159-170, Oct. 2010.
- [29] S. Nanda and K. Rege, "Frame error rates for convolutional codes on fading channels and the concept of effective Eb/N0," *IEEE Trans. Vehicular Technology*, vol.47, no.4, pp.1245-1250, Nov. 1998.
- [30] J. Proakis and M. Salehi, "Digital communications (fifth edition)," NJ: McGraw-Hill, Chapter 4, pp.160-289, Nov. 2007.
- [31] L. Brakmo and L. Peterson, "TCP Vegas: end to end congestion avoidance on a global Internet," *IEEE Journal on Selected Areas in Communications*, vol.13, no.8, pp.1465-1480, Oct. 1995.
- [32] A. Capone, L. Fratta and F. Martignon, "Bandwidth Estimation Schemes for TCP over Wireless Networks," *IEEE Trans. on Mobile Computing*, vol.3, no.2, pp.129-143, Apr. 2004.
- [33] G. A. Einicke, "Discrete-Time Minimum-Variance Prediction and Filtering," *Smoothing, Filtering and Prediction: Estimating the Past, Present and Future*. Rijeka, Croatia: Intech, 2012, Chapter 4, pp.75-100.
- [34] H. Yang, and S. Choi, "Implementation of a Zero-Forcing Precoding Algorithm Combined with Adaptive Beamforming Based on WiMAX System," *International Journal of Antennas and Propagation*, Article ID 976301, vol. 2013, 2013.
- [35] "A Practical Guide for Assessing the Relative Value of Standard and High Definition H.264 Video conferencing," University of Houston, Texas, USA. Available [Online]: http://www.uh.edu/uhevn/docs/Practical_HD_Guide.pdf
- [36] S.-B. Lee, G.-M. Muntean and A. F. Smeaton, "Performance-Aware Replication of Distributed Pre-Recorded IPTV Content," *IEEE Transactions on Broadcasting*, vol.55, no.2, pp.516-526, June 2009.
- [37] T. Porter, X. Peng, "An Objective Approach to Measuring Video Playback Quality in Lossy Networks using TCP," *IEEE Communications Letters*, vol.15, no.1, pp.76-78, Jan. 2011.

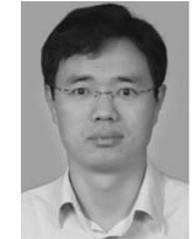
- [38] E. Cerqueira, L. Veloso, M. Curado, *et al.*, "Quality Level Control for Multi-User Sessions in Future Generation Networks," in *Proc. of IEEE GLOBECOM 2008*.



Changqiao Xu received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences (ISCAS) in January 2009. He was an Assistant Research Fellow in ISCAS from 2002 to 2007, where he was a Research and Development Project Manager in the area of communication networks. During 2007 - 2009, he worked as a Researcher with the Software Research Institute at Athlone Institute of Technology, Athlone, Ireland. He joined Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in December 2009, and was an Assistant Professor from 2009 to 2011. Currently, he is an Associate Professor with the Institute of Network Technology, and Vice-Director of the Next Generation Internet Technology Research Center at BUPT. He has published over 100 technical papers in prestigious international journals and conferences. His research interests include wireless networking, multimedia communications, and next generation Internet technology. He serves as a Co-Chair and Technical Program Committee (TPC) member for a number of international conferences and workshops.



Zhuofeng Li received his B.S. degree in telecommunication engineering from both Queen Mary University of London (QMUL), United Kingdom and Beijing University of Posts and Telecommunications (BUPT), China in 2011. He is currently working toward the M.S. degree in the Institute of Network Technology at BUPT. His research interests include wireless communication and multimedia transmission over wired/wireless network.



Jinglin Li received his Ph.D. degree from Beijing University of Posts and Telecommunications (BUPT), China in 2004. He is currently an associate professor at BUPT. He has published a number of technical papers, has authored two books and holds more than 10 patents. His research interests are in the areas of communication networks, mobile internet technology, internet of things and vehicular networks.



Hongke Zhang received his Ph.D. degree in electrical and communication systems from the University of Electronic Science and Technology of China in 1992. From 1992 to 1994, he was a postdoctoral research associate at Beijing Jiaotong University (BJTU), and in July 1994, he became a professor there. He has published more than 150 research papers in the areas of communications, computer networks, and information theory. He is the chief scientist of a National Basic Research Program ("973" program).



Gabriel-Miro Muntean received the Ph.D. degree from Dublin City University, Dublin, Ireland, for research in the area of quality-oriented adaptive multimedia streaming in 2003. He is a Senior Lecturer with the School of Electronic Engineering at Dublin City University (DCU), Dublin, Ireland. He is a Co-Director of the DCU Performance Engineering Laboratory, and Consultant Professor with Beijing University of Posts and Telecommunications, China. His research interests include quality-oriented and performance-related issues of adaptive multimedia delivery, performance of wired and wireless communications, energy-aware networking, and personalized e-learning. He has published over 200 papers in prestigious international journals and conferences, has authored three books and 16 book chapters and has edited six other books. He is an Associate Editor of the *IEEE Transactions on Broadcasting*, Associate Editor of the *IEEE Communications Surveys and Tutorials*, and reviewer for other important international journals, conferences, and funding agencies.