CMT-QA: Quality-aware Adaptive Concurrent Multipath Transfer in Heterogeneous Wireless Networks

Changqiao Xu, *Member, IEEE,* Tianjiao Liu, Jianfeng Guan, Hongke Zhang and Gabriel-Miro Muntean, *Member, IEEE*

Abstract—Mobile devices equipped with multiple network interfaces can increase their throughput by making use of parallel transmissions over multiple paths and bandwidth aggregation, enabled by the Stream Control Transport Protocol (SCTP). However, the different bandwidth and delay of the multiple paths will determine data to be received out of order and in the absence of related mechanisms to correct this, serious application-level performance degradations will occur. This paper proposes a novel Quality-aware Adaptive Concurrent Multipath Transfer solution (CMT-QA) which utilizes SCTP for FTP-like data transmission and real-time video delivery in wireless heterogeneous networks. CMT-QA monitors and analyses regularly each path's data handling capability and makes data delivery adaptation decisions in order to select the qualified paths for concurrent data transfer. CMT-QA includes a series of mechanisms to distribute data chunks over multiple paths intelligently and control the data traffic rate of each path independently. CMT-QA's goal is to mitigate the out-of-order data reception by reducing the reordering delay and unnecessary fast retransmissions. CMT-QA can effectively differentiate between different types of packet loss to avoid unreasonable congestion window adjustments for retransmissions. Simulations show how CMT-QA outperforms existing solutions in terms of performance and quality of service.

Index Terms—Quality-aware, concurrent multipath transfer, SCTP, heterogeneous wireless network, video delivery.

1 Introduction

In recent years, wireless communication technologies have experienced an extremely rapid development. Supported by the latest technological advances, mobile devices have also become smarter and many are already equipped with multiple network interfaces [1]. Large number of increasingly complex services and applications in various areas of interest, including business and entertainment, are widely offered to users of these mobile devices over the wireless networks, making use of their ubiquitous access support [2], [3], [4]. However, the heterogeneity of the wireless network environment requires additional solutions in order to enable smooth high quality service provisioning. The Stream Control Transmission Protocol (SCTP) [5], [6], [7], with its multihoming feature [8] and SCTP's dynamic reconfiguration extension (mSCTP) [9] are very promising protocols to support efficient data transmission, including seamless handover in heterogeneous wireless networks.

Concurrent Multipath Transfer (CMT) uses SCTP's multihoming feature to concurrently distribute data across multiple independent end-to-end paths in a multihomed SCTP association [10], [11]. Mobile devices equipped with multiple network interfaces can achieve bandwidth aggregation by using CMT to improve data throughput, bandwidth resource utilization and system robustness [12]. Figure 1 illustrates CMT usage in a heterogeneous wireless environment. It shows how a smart phone can concurrently use both 3G and WiFi access links to communicate with the server. It also indicates how a vehicle can communicate with the server by connecting to nearby Road Side Units (RSU) covered by gateways in a vehicular network scenario. The vehicle can avail from seamless handover between RSUs using IEEE 802.11r and can use 3G and IEEE 802.11p for communication concurrently. This approach improves the communication reliability and protects against connection failures, common in vehicular scenarios [13]. CMT is regarded as the ideal solution for content-rich real-time multimedia streaming applications with stringent bandwidth, delay, and loss requirements in heterogeneous wireless networks [12], [14], [15].

However, there is still significant ongoing work addressing many challenges of the SCTP CMT. The classic CMT strategy mainly uses a round-robin method to split SCTP packets over all available paths in an equal-share way without considering the path quality differences in

- *C. Xu is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. He is also with the Institute of Sensing Technology and Business, Beijing University of Posts and Telecommunications, Wuxi, Jiangsu, China. E-mail: cqxu@bupt.edu.cn.*
- *T. Liu and J. Guan are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. E-mail: {liutj, jfguan}@bupt.edu.cn.*
- *H. Zhang is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. He is also with the National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing, China. E-mail: hkzhang@bupt.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: munteang@eeng.dcu.ie*

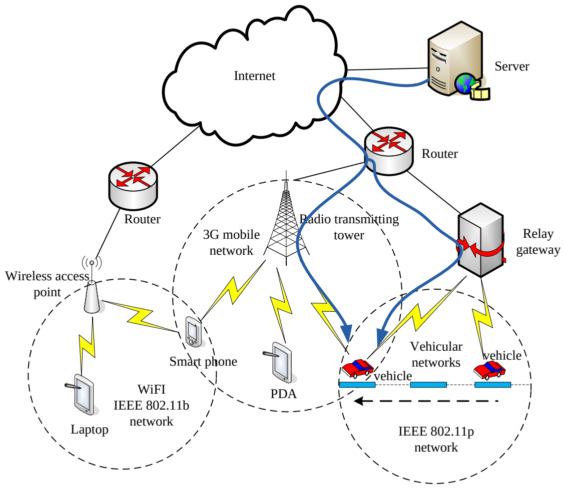


Fig. 1. CMT in a heterogeneous wireless network environment.

terms of bandwidth, delay and other QoS-related networking parameters. The "blind" round-robin approach for scheduling data chunks over heterogeneous wireless networks will undoubtedly cause serious problems in data delivery because asymmetric paths with different quality characteristics are more common and sensitive to variations in wireless networks than in wired networks. The receiver side has to maintain a great number of out-of-order data chunks for reordering. Consequently, CMT often suffers from significant receiver buffer blocking problems, which degrades transmission efficiency. Further, the increased out-of-order data and Selective Acknowledgement (SACK) segments will result in higher number of unnecessary fast retransmissions, additional reductions of the congestion window and higher overhead due to SACKs. At the same time, in multihomed wireless mobile networks, the mobile devices, such as PDAs, smart phones and embedded systems, have in general very limited memory capacity and little free space for the receiver buffer. Constrained receiver buffers cause even more serious concerns if different paths have disparate path characteristics in the heterogeneous wireless network environment.

This paper proposes a novel Quality-aware Adaptive Concurrent Multipath Transfer solution (CMT-QA) for data delivery in heterogeneous wireless networks. CMT-QA is aware of multiple paths' communication status and evaluates their quality in real time. Based on the evaluation, CMT-QA distributes SCTP packets over various paths in optimal manner according to their different handling capabilities. Furthermore, CMT-QA introduces an intelligent retransmission policy which avoids possible unreasonable performance degradations caused by data retransmissions using the current approaches. The simulation results show how CMT-QA effectively achieves better performance in comparison with basic SCTP's Concurrent Multipath Transfer strategies in scenarios with various network characteristics.

2 Related Work

Recently CMT has attracted extensive academic research interests. Dreibholz *et al.* [7] investigated the ongoing SCTP standardization progress in the IETF and gave an overview of activities and challenges in the areas of concurrent multipath transport and security. Wallace *et al.* [8] presented a comprehensive review of the SCTP and discussed contributions in three related research areas: concurrent multipath transfer, handover management, and cross-layer activities. CMT is highlighted as one of the hot research topics in the context of the multihoming-based SCTP.

Huang *et al.* [13] proposed a fast retransmission solution enabled by the use of relay gateways for CMT (RG-CMT) in vehicular networks to deal with packet loss. When the packets are lost due to handover, RG-CMT is able to fast retransmit them from the relay gateway to the vehicle, which saves transmission time and bandwidth. A wireless CMT SCTP (WCMT-SCTP) was proposed by our team in a previous work [16]. Both simulation and analysis results show how WCMT-SCTP improves the system throughput significantly in ad-hoc networks.

CMT-based multimedia streaming has attracted increasing attention from various researchers. Huang *et al.* [14] proposed a partially reliable-concurrent multipath transfer (PR-CMT) protocol for multimedia streaming. PR-CMT prevents having large gaps between two playable frames in order to have good video quality. We proposed a novel realistic evaluation tool-set [12] [15] to analyze and optimize the performance of multimedia distribution when making use of a CMT-based multihoming SCTP approach.

Iyengar *et al.* [10] proposed CMT and identified CMT's three negative side effects: (1) unnecessary fast retransmissions; (2) overly conservative congestion window ($cwnd$) growth; (3) increased acknowledgement traffic. CMT with a Potentially Failed state (CMT-PF) was proposed by Natarajan *et al.* [11]. A path that experiences a single timeout is marked as a "potentially failed" (PF), indicating doubts in its communication reliability. A PF path is not used for data transmission or retransmission until it is back to a fully active state. CMT-PF reduces the detection latency of link failures and improves CMT's throughput. However, CMT-PF uses the same round-robin schedule of CMT to send packets equally over all the paths, despite their very likely different capacities.

Fracchia *et al.* [6] introduced WiSE, a strategy for best path selection among the available alternative paths. Unfortunately WiSE did not take into account any of the benefits brought by CMT. Yang *et al.* [17] proposed a range-based path selection method (RPS) for CMT. It was found that as the number of paths increases, the path selection solution space increases exponentially, while receiver buffer efficiency decreases. The authors model the CMT throughput and design RPS to select paths according to receiver buffer size. Liao *et al.* [18] also proposed a multipath selection strategy to exploit the paths diversity by taking potential path correlation into account, which avoids underlying shared bottlenecks. A rate allocation model for best path transfer was presented by our team in [19] and we showed that it achieves the global optimum. However, none of the above works consider the dynamic path selection according to the likely variation of the current network conditions.

Yilmaz *et al.* [20] introduced non-renegable selective acknowledgements (NR-SACK) in order to avoid retaining the non-outstanding gap ACK chunks in the sender buffer. NR-SACK gives possibility to free buffer space earlier and reuse it for new data chunks. Dreibholz *et al.* [21] presented a blocking fraction factor and proposed a preventive retransmission policy based on the factor for effective transmission. Adhari *et al.* [22] proposed an optimized strategy to enhance the send and receive buffer handling by avoiding one path to dominate the buffer occupation. These solutions achieve performance improvements. However, the researchers do not provide a proper data distribution mechanism to ensure data packets arrival at receiver in order as much as possible.

Cui *et al.* [23] introduced a fast selective ACK scheme for SCTP to enhance transmission throughput in multihoming scenarios. In the networks with asymmetric delays for forward and reverse paths, a multihomed receiver sends SACK chunks to the sender over the fastest reverse path, which facilitates to inflate the congestion window and to retransmit the lost data packets as quickly as possible. Yet the solution just considers the transmission of control chunks and fails to enhance the overall transmission efficiency.

3 CMT-QA System Design Overview

During multihomed communications in a heterogeneous wireless networks, delay, bandwidth and loss rate of alternative paths can be significantly different. If a round-robin data delivery approach is used, slower paths are easily overloaded, while faster paths remain underutilized. In order to avoid unbalanced transmissions, reduce received data reordering and alleviate the receiver buffer blocking problem caused by the use of dissimilar paths using CMT, CMT-QA makes important contributions in the following three stages:

- Accurately senses each path's current transmission status and estimates in real time each path's data handling capacity.
- Includes a newly designed data distribution algorithm to deliver optimally the application layer data over multiple paths and ensure the received data arrives in order.
- Introduces a proper retransmission mechanism to handle different kinds of packet loss and alleviate the packet reordering problem.

Fig. 2 illustrates the design of the CMT-QA architecture, which includes a Sender, a Receiver and *n* communications Paths via the heterogeneous wireless network environment. The Receiver receives data and recreates

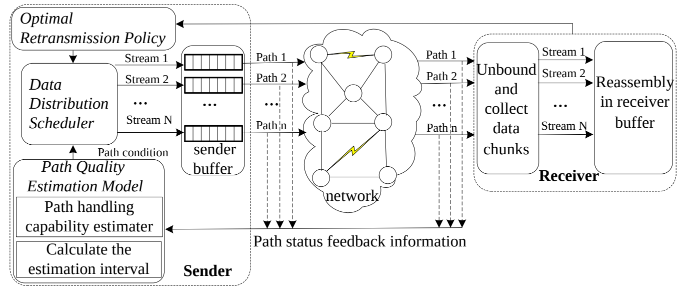


Fig. 2. CMT-QA architecture.

the original data chunks, if multiple data and control chunks are bundled together by the Sender into a single SCTP packet for transmission. In the case in which a user message is fragmented into multiple chunks, the Receiver reassembles the fragmented message in the receiver buffer before its delivery to the user. The feedback information of path status in the network is collected by the Sender and used to estimate the path quality. At the Sender there are three major CMT-QA blocks which are the *Path Quality Estimation Model* (PQEM), *Data Distribution Scheduler* (DDS) and *Optimal Retransmission Policy* (ORP). CMT-QA aims to intelligently adjust data distribution for each path and support in order data packet arrival at destination.

PQEM chooses a reasonable estimation interval to calculate the data handling rate of entering and leaving sender buffer for each path, which describes any path's communication quality. Any unfavorable conditions including packet loss rate, link delay, buffer size of routers, channel capacity and number of other data flows etc. will determine performance degradations of the paths handling capability. PQEM uses a comprehensive evaluation method to reflect the impact of above factors on the communication quality. PQEM's data handling rate of sender buffer describes better the end-to-end delivery conditions as its shorter estimation time enables its timely reflection of the current communication path status. Additionally, the samples for the time interval of distributed data's entering and leaving the sender buffer can be obtained easily to predict the path quality change trends.

Based on the path quality estimation results by PQEM, DDS chooses a subset of suitable paths for load sharing and dynamically assigns them appropriate data flows. In meanwhile, by forecasting the time of data arriving at destination in terms of each path quality, DDS can draft the period of packet distribution and also can adjust the distributed data amount for each path. In this way, the application data chunks are intelligently dispatched over multiple paths in real-time. Compared with the round robin scheme, we believe that the most effective approach to mitigate the reordering is to use a heuristic mechanism to decide the fraction of data scheduled to be transmitted on each path. The data distribution rate should be adjusted regularly according to each path's

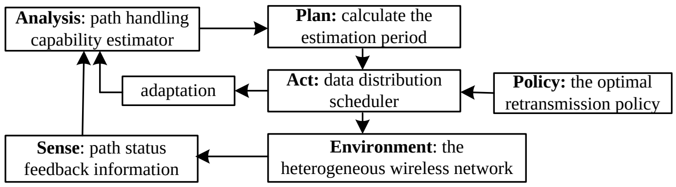


Fig. 3. The cognitive loop of the CMT-QA.

available buffer handling capability in order to ensure the data arrives at destination mostly in order, while also more efficiently utilizing each path's transmission capability.

ORP upgrades the basic CMT retransmission policies to improve packet retransmission efficiency. In the wireless network, most packet losses are caused by the dynamical wireless channel errors or path failures and not by the congestion. The standard CMT retransmission policy has no mechanism to distinguish random losses from congestion, and therefore treats all losses as congestion based. ORP differentiates random packet loss from congestion loss and path failure loss. ORP chooses the active path with the minimum value of the transfer delay to transmit these lost packets immediately, avoiding the rate-halving approach taken by the standard SCTP whenever random packet loss is detected.

The CMT-QA can be considered as a self-aware cognitive loop process [24] as illustrated in Fig. 3. In the estimation period, the feedback information is collected from the environment (***sense the surroundings***). Then the path condition is estimated (***analyze*** the correlative information automatically). After that, the next estimation period is calculated (***plan*** for the future). Based on the last estimation, the packets are distributed on the qualified path set and the retransmission is accelerated by making use of the upgraded retransmission policy (***make a decision and act accordingly***). It learns from feedback about past decisions (***study and adapt***) which helps achieve better accuracy and provide necessary experience for later decision, through sustained renewal of knowledge and feedback to prediction. This cognitive mechanism enables CMT-QA adapt to the dynamic wireless network environment and achieve very high concurrent multipath transfer efficiency.

4 Path Quality Estimation Model

RTT is generally used as the most important parameter for path quality estimation. Its computation considers the time of data transmission, data handling time at receiver and time of SACK transmission. In CMT, SACK can be sent on different paths and different delays on different paths lead to incorrect RTT estimations. Furthermore, by calculating the RTT of every packet sent on each path in an individual sampling approach can not reflect accurately the RTT variation process and estimate well the trend of path quality variation. CMT-QA will not utilize directly RTT information to distribute the data waiting in the sender buffer to each path. Instead PQEM divides the total time of sending data into dissimilar periods in terms of the sending situation of distributed data. PQEM also collects the amount of data sent and calculates the time interval between sending the data and receiving its corresponding SACKs. The above process is employed to calculate the rate of the distributed data entering and leaving the sender buffer. In this way, the transport layer can estimate very well each end-to-end path's transport capacity.

The current CMT maintains a single shared sender buffer, which makes obtaining each individual path's communication information impossible. Meanwhile, transmission blocking constrained by the sender buffer may happen. When data chunks are sent to the receiver side and until the sender receives the acknowledgements, these data chunks are stored in the sender buffer and marked with the outstanding status. When paths with significant transport capacity difference exist, the shared sender buffer can often be full with data chunks marked outstanding on the slow paths. In this situation no new data chunks on the fast path can be transmitted even if the current congestion and flow control mechanisms allow. In order to correctly estimate each path's quality and improve the transmission efficiency, in PQEM, the shared sender buffer is divided into individual sender sub-buffers for each path and each path connection manages its own sender sub-buffer independently. PQEM uses a dynamic buffer allocation mechanism to allocate different buffer space sizes to each path, according to its current transport capacity.

Based on the above architectural design of separate sending buffers for multiple paths, formula (1) is proposed to calculate the path quality.

$$Q_i = \frac{T_{l_i} - T_{e_i}}{buffersize_i} \tag{1}$$

where T_{e_i} is the time of the first $chunk_e$ entering path i's sender buffer from a group of distributed data chunks. It is obtained by recording the $chunk_e$'s entering time at the sender. T_{l_i} is the time of the last $chunk_l$ leaving path i's sender buffer from the group of distributed data chunks. T_{l_i} is obtained by recording the receiving time of corresponding SACK for $chunk_l$ at the sender. $buffersize_i$ is the size of path i's sender buffer which is occupied and later released during the transmission, namely the number of units of data entering and leaving the path i's sender buffer in a special period of time. Q_i denotes path i's sender buffer data handling rate. The lower the value of Q_i is, the higher the quality of current path i is. The value of $buffersize_i$ reflects the communication status of current path i in the total process of sending data.

There is still a question about how long should we calculate and update the estimation value of Q_i for a path i. In wireless conditions, if the evaluation interval is too short (e.g. shorter than or as short as the RTT), it may not correctly reflect the path condition when data

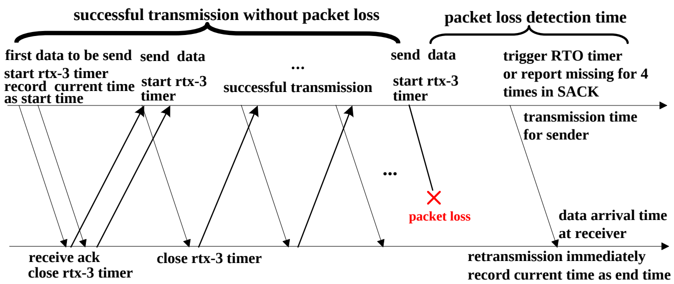


Fig. 4. Collect one interval sample.

chunks are randomly lost due to the varying wireless network conditions. However, if the evaluation interval is set too long, it would not reflect dynamically the path condition in time. Consequently the interval is adjusted based on the historic information and a proper length interval is selected to accurately update the value of Q. Confidence intervals [25] are widely used to quantify statistical uncertainty. Based on a sequence of previous statistical samples, PQEM uses the confidence interval to determine the next interval to calculate the value of Q. We select the time interval without packet loss as a sample. At the beginning, PQEM takes three heartbeat intervals as the initial sample. If packet loss occurs, the time period from the sending time of the first packet to the sending time of the last packet before packet loss happens is collected as one sample. Fig. 4 describes graphically this sampling process.

Algorithm 1 reveals how a sample for path i is collected. Each path has associated an individual Retransmission Timer T3-rtx to ensure data delivery in the absence of any feedback from its receiver. During the data distribution interval, when the first data chunk is to be sent, the sending time is recorded as the starting time of the time interval for a successful transmission. Whenever a data chunk is sent to any path (including a retransmission), if the T3-rtx timer associated with that path has not already been started, the sender starts the timer, so that it expires after the Retransmission Timeout (RTO) of that path. If the timer for that path is already running, the sender restarts the timer whenever an outstanding data chunk earlier sent over that path is being retransmitted. Whenever a SACK is received that acknowledges the data chunk with an outstanding Transmission Sequence Number (TSN) for that path, the T3-rtx timer is restarted for that path with its current RTO (if there is still outstanding data on that path). If all outstanding data sent to a path has been acknowledged, T3-rtx timer should be turned off for that path. If the data chunk is acknowledged through the Cumulative TSN ACK (cumACK), it can be dequeued from the sender's retransmission queue buffer. If packet loss occurs, either detected by the retransmission timeout or reported as missing by consecutive SACKs, the sender should retransmit the loss chunk immediately, in order to mitigate the reordering. In the packet loss case, the

Algorithm 1 Collecting a sample

∀ destination address d_i, initialize d_i.FirstData=TRUE; END=0;
while (!END)
    if (d_i.FirstData == TRUE)
        recording current time as the start time;
        d_i.FirstData = FALSE;
    end if
    transmit a data chunk;
    record current time as the chunk timestamp;
    if (d_i.RtxTimerIsRunning == FALSE)
        start T3-rtx timer;
        d_i.RtxTimerIsRunning = TRUE;
    end if
    if (the outstanding data's acknowledge arrived)
        restart T3-rtx timer;
    end if
    if (T3-rtx timer expired after RTO time || reported as missing for 4 times in the SACK)
        recording the last chunk timestamp as end time;
        END = 1;
        retransmit immediately; /*packet loss occurs*/
    end if
end while
calculate a distribution interval sample by end time minus start time;

timestamp (indicating the sending time) of the last data chunk is recorded as the end time of the successful data transmission interval. Having collected the latest data, there is a need to recalculate the path handling capability and update Q. Q is updated in two situations: in the packet loss case, as already described and by considering the confidence interval.

After collecting samples as described in Algorithm 1, by combining the historic interval samples, we calculate the confidence interval per path. Assuming the value of the time interval samples in one path are x_1, x_2, x_3, ..., x_n, we calculate the mean value of the time interval samples by using the formula from equation (2):

$$\overline{X_N} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{2}$$

where x_i is the successful transmit interval without packet loss for every time sample, N is the number of samples and $\overline{X_N}$ is the average time interval.

Equation (2) presents the general formula to calculate the mean value. In order to avoid storing all the collected samples at the sender, we use an iterative method to calculate the time interval mean, shown in equation (3).

$$\overline{X_{N+1}} = \frac{\overline{X_N} \times N + x_{N+1}}{N+1} \tag{3}$$

We use the previous time interval mean $\overline{X_N}$ and the new time interval x_{N+1} to calculate the current time interval mean $\overline{X_{N+1}}$. This means that the sampling intervals of the following sample is updated according to the newly recorded time interval, so the synchronous tendency will be approximated step by step. Thus we can determine the interval to estimate Q which will represent the path quality.

Similarly, the general formula presented in equation (4) is used to calculate the standard deviation.

$$S_N = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{X_N})^2}{N-1}} \tag{4}$$

where x_i is the successful transmit interval without packet loss for every sample. N is the sample size. $\overline{X_N}$ is the average time interval. S_N is the standard deviation of all the samples.

Equation (4) presents the general formula to calculate the standard deviation. We also use an iterative method to calculate the standard deviation to avoid storing all the collected samples at the sender and reduce the computational complexity. This method uses Equation (5).

$$S_{N+1} = \sqrt{\frac{{S_N}^2 \times (N-1)}{N} + \frac{(x_{N+1} - \overline{X_N})^2}{N+1}} \tag{5}$$

As equation (5) shows, we can calculate the new standard deviation S_{N+1} using only four variables: the previous time standard deviation S_N, the previous average time interval mean $\overline{X_N}$, the current time sample x_{N+1} and the previous sample size N.

When we learn about the coefficient of variation (standard deviation/mean) of a successful transmission, we can adapt the estimation interval. After obtaining the mean value and the standard deviation from equation (3) and equation (5), we use the Central Limit Theorem to calculate the confidence interval by using the formula from equation (6).

$$P\{\overline{X} - Z_{1-\frac{\alpha}{2}} \times \frac{S}{\sqrt{N}} < u < \overline{X} + Z_{1-\frac{\alpha}{2}} \times \frac{S}{\sqrt{N}}\} = 1 - \alpha \tag{6}$$

Where N is the number of samples and $1-\alpha$ is the confidence level. S is the standard deviation of all the samples. $\overline{X}$ is the mean value of all the samples.

Assuming the probability of transmission with no packet loss set to 95%, we have $\alpha = 0.05$. As shown in table 1, we can get $Z_{1-\frac{\alpha}{2}}$ =1.96 by using the look-up table method. Consequently, we obtain the confidence interval u which is derived as a reference for further evaluation to update the path quality and predict its trends.

The value of confidence interval is also used as a reference to assist selecting paths. If it is less than RTO, the packet loss may occur in a short time, which means the path is not in a good condition and we cannot detect it until after a relatively long time. If we used it to transmit in parallel the data chunks, we need to wait for the retransmission of the lost chunk in this path and thus the transmission efficiency decreases. So, the decision was to mark the path whose successful transmission interval is less than RTO with an inactive status. For transmissions we select the sub-set of paths with high quality, so that data packets can be received in order with much less retransmissions than the traditional way.

TABLE 1
Confidence Levels and Corresponding α and Z Value

Confidence level	$\frac{\alpha}{2}$	$Z_{1-\frac{\alpha}{2}}$
80%	0.1	1.282
90%	0.05	1.645
95%	0.025	1.96
98%	0.01	2.326
99%	0.005	2.576

5 Data Distribution Scheduler

After estimating each path's data delivery capability, data is distributed to these paths accordingly. When sending new data, the sender is constrained by three factors: the congestion window (congestion control), the advertised receiver window (a_rwnd in relation to the flow control) and the sender buffer size. As we already know, the basic SCTP CMT restricts the maximum amount of data to be transmitted via the size of the receiver window ($rwnd$). The $rwnd$ is shared by multiple paths across an SCTP association. The receiver buffer is used to store all the data chunks received out-of-order and they are delivered to the application when all the missing data chunks are received only. Since an SCTP association allows multihomed source and destination endpoints, a source maintains several parameters per destination such as the $cwnd$ and the amount of outstanding data $outstanding$. A multihomed sender can transmit chunks across all available paths as long as $cwnd$ allows it. SCTP's $cwnd$ limits the data a sender can send to a particular destination transport address before receiving an acknowledgement. In order to avoid buffer blocking, first it is essential to control the maximum data amount that can be sent in total to all the active paths at the sender. Equation (7) is used to describe the maximum data amount for the sender:

$$D_{max} = min(\sum_{i=1}^{n}(cwnd_i - outstanding_i), rwnd) \tag{7}$$

where n denotes the number of active paths. $cwnd_i$ is the congestion window for path i. $outstanding_i$ is the number of outstanding bytes (the data that has been sent, but not yet acknowledged) in path i. $cwnd_i - outstanding_i$ means the data amount that allowed transmitting for path i. This prohibits new segments from being transmitted when old ones are still outstanding. $\sum_{i=1}^{n}(cwnd_i - outstanding_i)$ is the total data amount that can be sent for all the paths. $rwnd$ is shared between all the paths and indicates the maximum data amount that can be received and therefore allowed to be sent. If $cwnd$ is larger than the $rwnd$, the sender is limited by $rwnd$.

Assuming that the $cwnd$ value is chosen to indicate the maximum amount of data to be sent, more data is sent to the receiver than it can handle and receiver buffer blocking will happen. In this case, the receiver buffer is full and the sender can only transmit one data chunk to the receiver, if allowed by $cwnd$ to probe for a change in $rwnd$ and a SACK is sent to the sender. In conclusion, the sender is constrained by D_{max} - the minimum value between the $cwnd$ and $rwnd$ and data transmission can be monitored by looking at D_{max}.

The value of $rwnd$ in equation (7) can be obtained from a_rwnd through calculation according to RFC4960 [5] as follows. $rwnd$ is set as equal to the newly received a_rwnd minus the number of bytes still outstanding after processing the cumACKs and the Gap Ack Blocks. When the sender receives the SACK from any path, it can obtain the value of a_rwnd. This value represents the current available buffer space size of the receiver at the time of transmitting the SACK. As data chunks are received and buffered, the decrement of a_rwnd is set to the number of bytes received and buffered. In fact, this reduces the size of $rwnd$ at the data sender and restricts the amount of data it can transmit. The above process is formalised in equation (8):

$$rwnd = a\_rwnd - \sum_{i=1}^{n} outstanding_i \tag{8}$$

Equation (8) includes the total amount of outstanding data from all the paths calculated from each path's outstanding data, namely $\sum_{i=1}^{n} outstanding_i$. $rwnd$ indicates the maximum data amount that the receiver can handle.

Knowing the total amount of data that can be sent, data chunks need to be dispatched to the various paths. Based on the PQEM and the maximum sending data amount, the data distribution strategy over the multiple paths is detailed next. The period each path needs to deliver the data stored in its sender buffer in one round trip time will be calculated using the formula from equation (9):

$$T_{handle_i} = Q_i \times cwnd_i \tag{9}$$

where Q_i indicates the path i's quality, as estimated by PQEM. $cwnd_i$ is the congestion window of path i. T_{handle_i} is the time for path i to deliver the data in its sender buffer per round.

To avoid significant differences between various paths, we select a subset of the paths which have close capabilities to deliver the data in terms of T_{handle_i}. In order to make use of as many of the active paths in the path subset as possible, we select the maximum T_{handle_i} as the data distribution period, P_d, as in equation (10):

$$P_d = max(T_{handle_1}, T_{handle_2}, ..., T_{handle_n}) \tag{10}$$

Data is dispatched concurrently to all active paths in the subset in the distribution period P_d. The distribution frequency is determined by $\left\lceil \frac{P_d}{T_{handle_i}} \right\rceil$.

Before a data chunk is to be transmitted, the time it takes from when it enters the sender buffer to its arrival at the receiver per path should be estimated. The path with the shortest time of arrival is selected as the transmission path.

The following formula is used to describe the relationship between the data amount that a path i can distribute and its current congestion window. Assuming k is the round in which the data chunk can be sent, the maximum data amount transmitted after k rounds of distribution is computed using equation (11).

$$\sum_{j=0}^{k-1}(cwnd_i + j \times MTU) \geq D_i, 0 < D_i < D_{max} \tag{11}$$

Assuming that the STCP connection has MTU bytes/packet, after receiving an acknowledgement, $cwnd$ should be increased by one MTU per RTT according to [5]. $\sum_{j=0}^{k-1}(cwnd_i + j \times MTU)$ is the total amount of data that can be delivered after k rounds of data distribution. D_i is the total data amount distributed on path i. Equation (12) can be derived from equation (11):

$$\begin{cases} k = \left\lceil \frac{\sqrt{num} - (2cwnd_i - MTU)}{2 \times MTU} \right\rceil \\ num = (2cwnd_i - MTU)^2 + 8 \times MTU \times D_i \end{cases} \tag{12}$$

Equation (12) calculates how many rounds (k) are required to deliver the data dispatched over the path i, where D_i is the data amount already distributed over path i. Equation (13) uses equation (12) to predict data delivery time per path; the path with the minimum predicted delivery time is selected and the data chunk is dispatched over it.

$$T_i = k \times Q_i \times cwnd_i \tag{13}$$

Equation (13) indicates that the path i needs to take T_i time to deliver the dispatched data chunks. The value of k is calculated as in equation (12), Q_i indicates the path i's quality. We select the path with the shortest time among all the T_i-s and dispatch the data chunks over that path. The data amount distributed in path i for this round is $cwnd_i + (k-1) * MTU$. The chunks are then queued into the chosen path sender buffer. The total amount of the data distributed on path i, namely D_i is updated and according to equation (12) data is dispatched in the next round.

Based on the research described above, the proposed Data Distribution Scheduler (DDS) is summarized through the following aspects. Equation (7) offers the total amount of data that the sender can deliver during one data distribution period. According to equation (1), the data handling capabilities per path is determined. Then the data amount distributed per path during the next data distribution period is estimated. The data handling capabilities is decided not only by the $cwnd$

Algorithm 2 Data distribution scheduler

```
P_d = 0; /*initialize the dispatch period*/
for(∀ destination address d_i)
    obtain d_i's quality Q_i and current cwnd_i;
    calculate its period to handle the data in the send
    buffer one round trip by equation (9);
    if(T_handle_i > P_d)
        P_d = T_handle_i;
    end if
end for
while(the dispatch timer is not expired during P_d)
    initialize the dispatch destination datadest;
    initialize the minimum time min_t = ∞;
    for(∀ destination address d_i)
        calculate the times k for the next dispatched data to
        sent in d_i according to equation (12);
        predict its time to arrive at the receiver by equ. (13);
        if(T_i < min_t)
            min_t = T_i; datadest = d_i;
        end if
    end for
    dispatch the data to the datadest with the minimum time;
    datadest.D_i = datadest.D_i + cwnd_i + (k − 1) ∗ MTU;
end while
```

value of the current path, but also limited by the D_{max} as in equation (7). The time required for handling data distribution over each path is estimated as in equation (13). DDS utilizes this time to predict the arrival time of data distributed per path and then it can intelligently know how much and when to distribute data over the multiple paths. In this way, DDS makes sure that the data distributed per path arrives at receiver in order. Algorithm 2 reveals the details of the process of data distribution scheduler.

6 Optimal Retransmission Policy

The frequency and time-varying characteristics of the wireless channel will cause unpredictable packet loss, so the retransmission is inevitable in order to guarantee the service quality. As is well known, SCTP has the responsibility to keep the received data in order, so it waits for the lost packets' arrival before pushing the whole data segment to the upper layer.

The SCTP standard defines two retransmission algorithms: fast retransmission and timeout retransmission. When packet loss occurs in one path, recognized either by the SACKs on gap report or after a RTO time (via T3-rtx timer expiration) without acknowledgement, a retransmission is required.

An SCTP endpoint uses a T3-rtx timer to ensure data delivery in the absence of any feedback from its receiver. For the destination address for which the timer expires, $cwnd$ is set to one maximum segment size and the end host enters the slow start mode. A retransmission timeout will double the RTO, whereas a successful retransmission will not refresh the RTO which can only be updated by the heartbeat chunks. Consequently, the RTO is usually a large value which causes the data loss detection time to become very long and degrades the delivery performance. By introducing a fast retransmission function, loss can be recovered rapidly and the delivery quality for the users can be maintained at high levels. Fast retransmission helps avoid the long waiting for the retransmission timer to expire and reduces the mean delay. Fast retransmission is considered if SACK indicated that a segment has been missing four times and therefore packet loss has occurred. SCTP retransmits the loss packet immediately and modifies the congestion window ($cwnd$) and the slow start threshold ($ssthresh$). Set $ssthresh$ equal to $max(\frac{cwnd}{2}, 4 \times MTU)$ and $cwnd = ssthresh$.

When packet loss occurs in the condition of concurrent multipath transfer, this loss phenomenon reduces the transmission efficiency of current path through sharply decreasing the $cwnd$. Meanwhile, the existing mechanism does not make a distinction between random packet loss in wireless networks and the congestion loss. The long period to detect the timeout packet in path failures also decreases the transmission efficiency. In conclusion, there is a definite need to design new strategies to handle packet loss more efficiently.

In the heterogeneous wireless networks, packet loss can be classified into three categories: 1) packet loss due to congestion as there is limited bandwidth or buffer size; 2) error loss caused by noise or interference in the wireless networks; 3) path failure loss or handover loss. In the wireless network, most packet losses are due to dynamical wireless channel fluctuations or due to path failure and not due to congestion. A path failure loss is usually detected by timeout events, whereas an error loss is detected by the gap report in the SACKs.

This paper proposes the Optimal Retransmission Policy (ORP) which detects the cause of data loss and reacts in an optimum manner. When a packet loss occurs and $\frac{rtt_i}{cwnd_i} \geq Q_i$, loss is considered as random packet loss due to wireless conditions and the sending rate is not limited by adjusting the $cwnd$ until loss happens consecutively. If the packet is lost randomly due to dynamical interferences or noise, the path condition is still in good situation, and there is no need to halve the $cwnd$ value to limit the sending rate. However, if loss occurs more than once consecutively, this indicates a congestion and $cwnd$ value should be reduced in order to decrease the sending rate. Once timeout occurs on a path, it should be marked with an inactive status. To reduce the detection latency of link status change, a periodic heartbeat packet is sent to check whether links are alive or not. Meanwhile in both cases, the sender tries to reach the active path with the minimum value of the transfer delay to transmit these lost packets as soon as possible. After receiving the SACK for the retransmission data, the DDS strategy will continue to distribute data over the rest of the paths. The detailed procedure for the optimal retransmission strategy is described in Algorithm 3.

Algorithm 3 Optimal retransmission policy

if(packet loss)
  if($\frac{rtt_i}{cwnd_i} \geq Q_i$)
    do not adjust $cwnd$ to limit the sending;
    /*wireless error not congestion */
  end if;
  if(T3-rtx timer expired after RTO time)
    $ssthresh = max(\frac{cwnd}{2}, 4 \times MTU)$;
    $cwnd = MTU$;
  end if;
  if(received 4 duplicated SACK)
    $ssthresh = max(\frac{cwnd}{2}, 4 \times MTU)$;
    $cwnd = ssthresh$;
  end if;
  retransmit the lost packet as soon as possible;
end if;

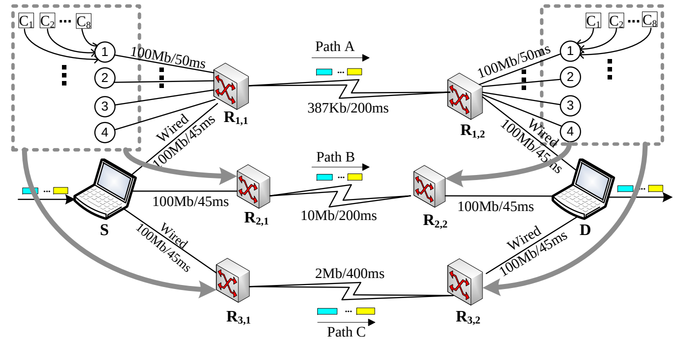


Fig. 5. Heterogeneous wireless network topology used in the simulations.

7 Performance Evaluation

This section evaluates CMT-QA's performance during conventional reliable FTP-like data transmission and real-time video delivery, respectively. CMT-QA is compared with two SCTP-based CMT mechanisms: the original CMT [10] and CMT-PF [11], respectively.

7.1 FTP Data Transmission

1) Simulation Setup

The evaluation has been carried out on the Network Simulator (NS-2.35) [26]. It includes the latest SCTP Module developed by the University of Delaware. The experiments considered the heterogeneous wireless network environment illustrated in Fig. 5. The default receiver buffer size is 64 KB. The Link queue limit and type are set 50 packets and Droptail, respectively. RTX-CWND is used as the retransmission policy [12], [15]. The other parameters use the SCTP default values [5]. The simulation time is $100s$ with infinite FTP flows.

As the figure shows, each router $R_{i,j}$ is attached to five edge nodes. Edge nodes S and D are the data sender and receiver, respectively. The other four edge nodes (denoted ①, ②, ③ and ④ for router $R_{1,1}$ for instance in Fig. 5) are single-homed and introduce bursty cross traffic to simulate congestion at the routers. Each of them has eight traffic generators C_1, C_2, …, C_8 producing cross traffic with a Pareto distribution. The cross traffic packet sizes are chosen to resemble the distribution found on the Internet: 50% are 44 bytes long, 25% have 576 bytes, and 25% are 1500 bytes long [27]. Between S and D, there are three alternative paths with different bottlenecks. Path A's bottleneck has 387 Kbps bandwidth and 200 ms transmission delay, which is representative for a 3G link. Path B's bottleneck has 10 Mbps bandwidth and 200 ms transmission delay, which corresponds to a WiMax (IEEE 802.16) link. Path C's bottleneck has 2 Mbps bandwidth and 400 ms transmission delay which is encountered in WiFi networks (IEEE 802.11). The simulation result is a data transfer between S to D, over a network with self-similar cross traffic (burst congestion) and packets loss (Bernoulli loss model) which resembles the nature of the traffic on data networks. The aggregate cross traffic loads on the three paths are similar and vary randomly between $0\% - 20\%$ of the bottleneck links bandwidth to simulate a highly dynamic wireless network environment. All testing results presented are calculated by averaging the results of 100 runs, which makes the effect of the cross traffic and loss rate on different strategies be representative and not influenced by any stochastic factors.

2) Simulation Results

(1) Packet arrival order-related indicators

Packet sending and receiving times: Fig. 6 illustrates sending and arrival times of several data packets when three schemes are used, respectively. In order to better illustrate the comparison, the results between t=10 s and t=11 s are presented only (part of congestion avoidance stage). The TSNs of these data packets growth has two main slopes for all the schemes: the upper one represents data flows over path A and path C, whereas the lower one indicates the data chunks transmitted over path B. In both CMT and CMT-PF schemes, the sender uses the round robin method to transmit data chunks over all the paths equally, without considering the path quality differences. In contrast, the flow of path B is utilized more efficiently by the CMT-QA solution as its TSNs increase steeply, while the TSN of path A and path C increase slower. This confirms that CMT-QA distributes the data chunks over the available paths in proportion to their respective data handling rate.

The packets are received out-of-order due to the dissimilar path characteristics and their reordering is likely to cause performance degradations. For example when using CMT, the data chunk with TSN 862 is lost. CMT detects the packet loss and then retransmits it at t=10.582 s. By analyzing the simulation traces, we notice that the lost data chunk is dropped at t=9.988 s in path C and re-enters the sending queue at t=10.051 s. Later on, at t=10.582 s, the data chunk is sent over path B and received at t=10.628 s. Similarly, we can see the loss in CMT-PF. The data chunk with TSN 853

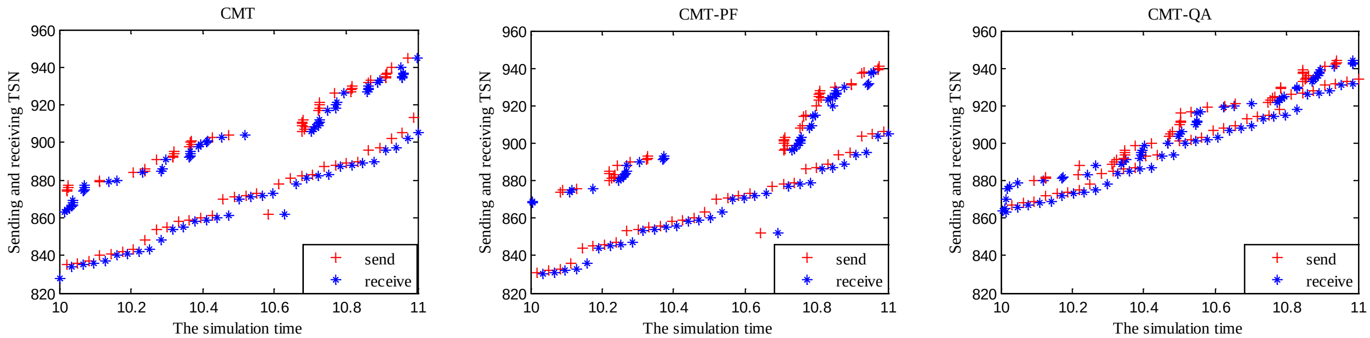


Fig. 6. Comparison of sending and receiving time of packets.

is retransmitted at t=10.644 s and received at t=10.691 s, respectively. In both CMT and CMT-PF, the path with the lost chunk fails abruptly for about 0.2 or 0.3 seconds and resumes later. That may be caused by the outstanding data chunks constraining the sender from transmitting any new data, which indicates the path is in low quality (i.e., high loss rate or undergoes congestion). Fortunately the path recovers and the retransmitted data is eventually received. The CMT-PF doubts path A's communication reliability and does not use that path for data transmissions for a while, until the heartbeat acknowledgement determines its set to an active state again. In meanwhile the receiver waits for the arrival of the retransmitted data. The subsequent data chunks which have already arrived are held in the transport layer receive buffer and unable to be delivered to the application until all the retransmitted data arrives. This phenomenon blocks the receiver buffer and intensifies the reordering problems. With the sender enhanced with the path quality-aware data distribution strategy, CMT-QA can predict the arrival time and decides the sending path based on it. In this way, CMT-QA can avoid the need for most reordering and it can be clearly observed how data chunks are received smoothly.

Out-of-order packets: Fig. 7 shows comparison of out-of-order chunks among CMT, CMT-PF and CMT-QA. The out-of-order TSN metric used in this experiment is measured by the offset between the TSNs of two consecutively received data chunks (the difference between the TSN of the current data chunk and that of the latest received data chunk). The out-of-order TSN metric portrays the characteristics of concurrent data transmission over multiple paths. Fig. 7 presents the out-of-order TSN metric variation between simulation time t=10 s and t=20 s, representative for the whole simulation results. As the figure shows, CMT and CMT-PF generate more out-of-order chunks and require increased reordering than CMT-QA. CMT-QA estimates the latest information available in terms of path quality and distributes the data according to the predicted arrival time. In this way, CMT-QA reduces the out-of-order data arrival and consequently performs better than the other two schemes. When comparing the three transfer methods, it is noted that peak out-of-order data reception at the receiver is approximately 45 using both CMT and CMT-PF, while it is only 20 when using CMT-QA. With a 64 KB receiver buffer and 1500 bytes chunk size, the receiver can store mostly 43 data chunks ($64 \times 1024/1500$). In the condition of the out-of-order TSN offset reach 45, the receiver buffer blocking is likely to happen and the transmission performance is seriously deteriorated.

(2) Average retransmission

Fig. 8 illustrates the average number of retransmissions when three methods are employed respectively with the increase of path loss rate (PLR) in all the paths. During the experiments, the PLR was varied from 0% to 10% for the three paths. The results show that the average retransmissions across all the paths increase with the increase in packet loss probability, directly affecting the throughput for all the mechanisms. Higher packet loss probability determines both more data chunk retransmissions, and more out-of-order data delivery. If the receiver buffer is full with out-of-order packets, waiting for the lost data retransmissions to fill the gaps, the transmission efficiency will decrease. After 4 duplications or rtx-timeout, the sender will retransmit the lost data. As the figure shows, the average number of retransmissions of CMT increases sharply with the packet loss probability increase. The CMT-PF performs better than CMT as it detects path failures and stops transmitting data over the path with bad delivery status. In contrast, CMT-QA is aware of characteristics difference between paths and adapts to each path's delivery conditions, intelligently distributing the data across the paths. In this case most of the data arrives at the receiver in the right order, reducing the number of retransmissions and hence CMT-QA performs the best among the three solutions compared. For example under a PLR of 10%, there are 134 retransmissions for CMT, 101 for CMT-PF and 73 only when CMT-QA is employed.

(3) Average throughput

Fig. 9 illustrates the comparison results of the total average throughput as the PLR increase. This experiment was to verify the ability of the three schemes to manage packet loss, which has significant impact on the end-to-end throughput. As the figure shows, network

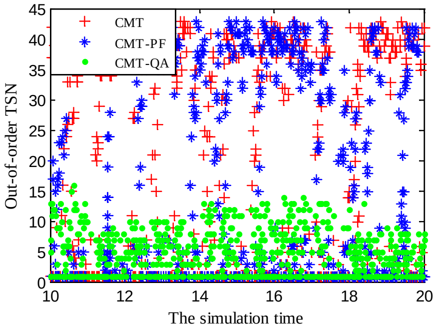


Fig. 7. Comparison of out-of-order TSN.

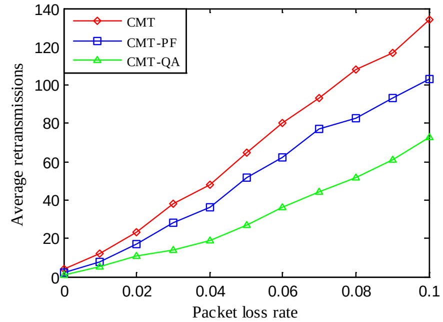


Fig. 8. Comparison of average retransmission with loss rate.

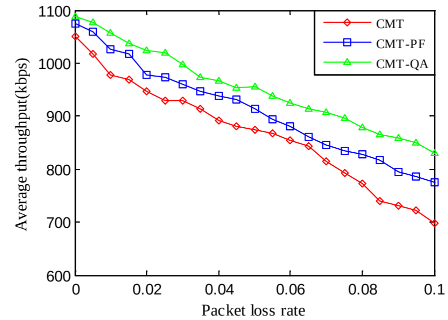


Fig. 9. Comparison of average throughput with loss rate.

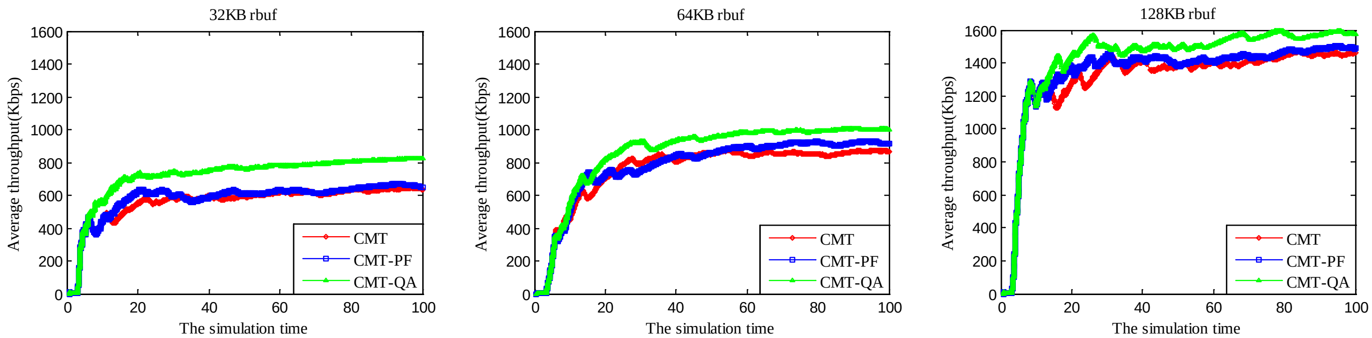


Fig. 10. Comparison of average throughput when using different receiver buffer sizes.

throughput decreases with the increase in the link loss probability for all mechanisms. However, the average throughput values of CMT and CMT-PF decrease more significantly than that of CMT-QA. For example, for a PLR of 5% CMT-QA's throughput is 8% higher than that of CMT and 2.5% higher than that of CMT-PF, whereas for a loss of 10% CMT-QA's throughput is 19% higher than that of CMT and 7% higher than that of CMT-PF. This result is as any increase in the PLR causes $cwnd$ to be reduced and the transmission delay to increase. CMT's throughput decreases sharply and performs the worst when the PLR increases. Because the congestion window is halved when packet loss occurs. As the CMT-PF solution can identify packet loss due to short term path failures, it performs better than CMT. CMT-QA can detect and differentiate random packet loss and path failure from congestion loss, sense the path condition in time and schedule the data delivery based on each path's transmission capability. Although the paths used for load sharing have different packet loss characteristics, CMT-QA achieves higher association throughput than both CMT and CMT-PF.

Fig. 10 compares average throughput when delivering content with receiver buffer sizes of 32 KB, 64 KB and 128 KB, respectively. The PLR of the three paths varied randomly from a uniform distribution between 0% and 10%. Three groups of simulation were run in order to study the effect of the receiver buffer size on the throughput. It can be seen that the throughput of all schemes increases with the increase in receiver buffer size. At first, the throughput increases rapidly because SCTP probes the available network capacity. The slow-start algorithm doubles repeatedly the $cwnd$ size. Next, throughput experiences variations for all the mechanisms due to the packet loss, then it recovers after retransmissions and $cwnd$ adjustments. Compared with CMT and CMT-PF, CMT-QA tolerates better packet loss and utilizes more efficiently the available aggregate bandwidth from different links. For instance after 100s of simulation time with a 32 KB receiver buffer, CMT-QA's throughput is 29% higher than that of CMT and 26.5% higher than that of CMT-PF. With a 64 KB receiver buffer size, the corresponding comparison of average throughput performance is 15% and 9% in favor of our proposed solution, respectively. Similarly, CMT-QA performs 7.6% and 5.5% better than CMT and CMT-PF, respectively when a 128 KB receiver buffer was employed.

We further evaluate the average throughput with different receiver buffer sizes with PLRs varying from 0% and 10%. CMT-QA outperforms CMT and CMT-PF in all cases; the difference is very much in favour of CMT-QA in limited receiver buffer situations. In the heterogeneous wireless networks with dynamic path conditions, the more varied handling capability of different paths is, the larger receiver buffer is required to maintain the transmission efficiency at high levels. Receiver buffer blocking depends on the frequency of the loss events

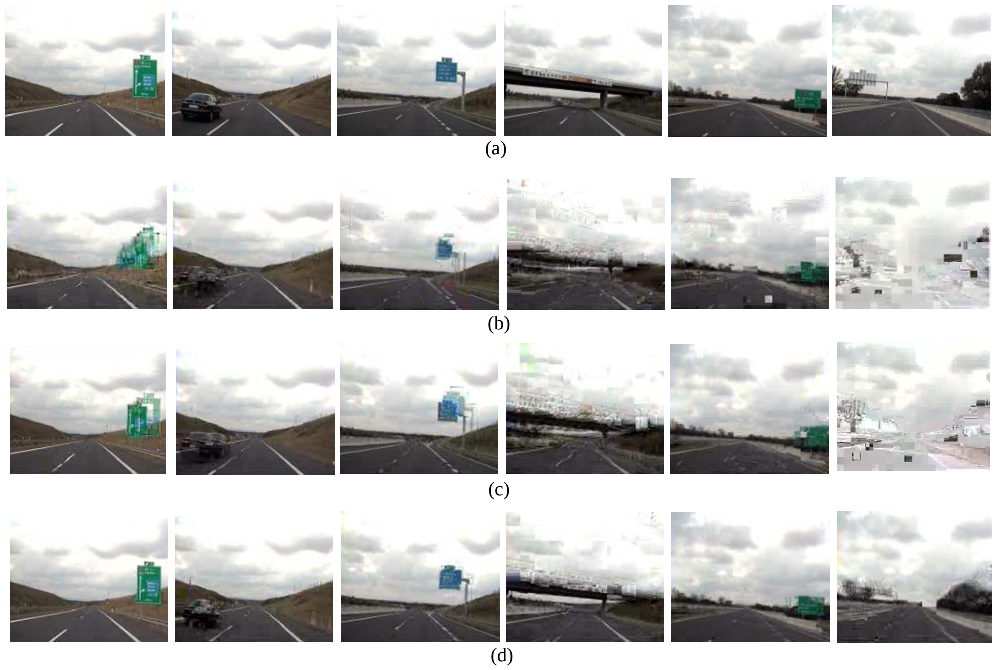


Fig. 11. Frames taken from received and reconstructed videos. Sent video (a) vs received video using CMT (b), CMT-PF (c) and CMT-QA (d), respectively.

and the duration of loss recovery. Using ORP, CMT-QA can both better detect and handle the packet loss in a shorter period of time. Transmitting through the fast path also speeds up the retransmission of the lost packets. Additionally in DDS of CMT-QA, the sending path is chosen according to the predicted arrival time. All the mechanisms employed by our solution mitigate the reordering of received packets and enable CMT-QA not to need large receiver buffer to store the out-of-order data chunks.

7.2 Real-time Video Delivery

This section investigates how CMT-QA's performance compares with that of CMT and CMT-PF for real-time video transmissions. This set of experiments makes use of our previously developed tool-set Evalvid-CMT for video quality evaluation [12], [15]. Evalvid-CMT enables performing comprehensive video delivery quality evaluation when employing SCTP network simulations. It supports accurate objective video quality and user perceived quality assessments. The SCTP version set is SCTP Partial Reliability extension (PR-SCTP) [12], [15]. The numbers of retransmission for each packet are set to no more than two times. The simulation topology and SCTP parameters values used are the same with that used in section 7.1.

The original test video sequence used is known as Highway QCIF (176×144) which consists of 2000 frames with average quality. After pre-processing stage [12], [15], a MPEG-4 video which includes 223 I frames, 445 P frames and 1332 B frames is produced. Those frames are fragmented into 2250 packets which include 463 packets storing I frames, 453 packets including P frames and 1334 packets carrying B frames. A corresponding MPEG-4 video trace file including these packet-based information is fed to the NS2. These 2250 packets will be transferred over the SCTP simulation model network.

TABLE 2
Comparison of average PSNR (dB), VQM, SSIM and number of frames lost

PLR	Methods	PSNR	VQM	SSIM	I	P	B
2%	CMT	35.65	0.095	0.996	2	1	3
2%	CMT-PF	35.70	0.066	0.997	1	1	2
2%	CMT-QA	35.72	0.005	0.999	0	0	0
4%	CMT	32.95	0.577	0.977	43	69	113
4%	CMT-PF	33.37	0.482	0.984	27	55	94
4%	CMT-QA	34.57	0.345	0.989	12	26	65
6%	CMT	30.82	1.432	0.895	53	124	194
6%	CMT-PF	31.62	1.396	0.899	50	105	161
6%	CMT-QA	33.10	0.975	0.914	33	78	121
8%	CMT	28.90	2.412	0.872	67	160	248
8%	CMT-PF	30.16	2.096	0.892	57	147	229
8%	CMT-QA	32.07	1.328	0.910	49	98	184
10%	CMT	26.17	3.120	0.842	108	190	300
10%	CMT-PF	28.13	2.759	0.867	88	178	262
10%	CMT-QA	30.15	1.560	0.891	57	149	228

Table 2 presents the comparison results of average video quality, expressed in terms of PSNR (dB), VQM, SSIM and the number of different dropped frames (I-frame/P-frame/B-frame) when CMT, CMT-PF, and CMT-QA are used, when PLRs are 2%, 4%, 6%, 8% and 10%, respectively. The dropped frames are either lost frames during network transfer or discarded frames at the receiver due to high delay/jitter which would have made their arrival too late for their playout time. As the table illustrates, CMT-QA outperforms CMT and CMT-PF in all the different PLRs situations studied, especially if the PLR is greater than 2%. For example, in the case of a path loss rate of 4%, the average PSNR of CMT and CMT-PF are 32.95 dB and 33.37 dB, respectively, but the average PSNR of CMT-QA is as high as 34.57 dB. The number of total dropped frames of CMT and CMT-PF are 225 (43I+69P+113B) and 176, respectively. However there are only 103 dropped frames for CMT-QA, 54.2% lower than the value experienced by CMT and 41.4% lower than the number of lost frames recorded for CMT-PF. The table also illustrates how CMT-QA achieves increasingly better results than CMT and CMT-PF with the increase in the PLRs. For example the average PSNR (dB) difference between CMT-QA and CMT is 1.62 with 4% PLR. However, with PLR increasing to 6%, 8% and 10%, the average PSNR (dB) differences between CMT-QA and CMT increase to 2.28, 3.17 and 3.98, respectively in favor of CMT-QA. The average difference between CMT-QA and CMT-PF in terms of PSNR (dB) is 1.20 with a 4% PLR, but it increases to 1.48, 1.91 and 2.02 when PLR increases to 6%, 8% and 10% respectively.

By using Evalvid-CMT, the received 2000 frame video can be reconstructed. We further compare the three delivery solutions in terms of other two video quality metrics: VQM and SSIM. We compared the reconstructed video clips with the sent video by using the MSU Perceptual Video Quality tool [28]. It can be seen how VQM values are the lowest and how SSIM results are the closest to 1 when using CMT-QA in comparison with the other solutions, regardless of the increase in loss probability. These results fully confirm that CMT-QA outperforms CMT and CMT-PF when assessed with a wide range of video quality metrics. Fig. 11 presents a sequence of frames taken from the sent video (a), received video using CMT (b), received video using CMT-PF (c) and received video employing CMT-QA (d), respectively when the PLR is 6%. This frame sequence illustrates the benefit of using CMT-QA in terms of perceived quality in comparison when CMT and CMT-PF are employed.

8 Conclusions and Future Works

This paper proposes a novel Quality-aware Adaptive Concurrent Multipath Transfer solution (CMT-QA) for SCTP-based data delivery over heterogeneous wireless networks. CMT-QA relies on three new mechanisms: the Path Quality Estimation Model, Data Distribution Scheduler and Optimal Retransmission Algorithm. Using these mechanisms, CMT-QA monitors and analyzes the dynamic network environment in real time and estimates each transmission path's quality. Based on the output of the path quality evaluation, CMT-QA intelligently adjusts data distribution across the multiple paths. Data distribution is also considering time of data arrival at the destination forecast, to increase the in-order data packets arrival. The optimal retransmission policy introduced by CMT-QA differentiates between different kinds of packet loss and accelerates the retransmission if required in order to improve data delivery efficiency. The simulation results demonstrate how the proposed CMT-QA obtains better performance results for both reliable data transmission and real-time video delivery than classic SCTP CMT and CMT-PF mechanisms. Future work will consider the fairness and TCP-Friendly issues of concurrent multipath transfer [29], [30]. We aim to make CMT-QA achieve high data delivery efficiency while still remain fair to concurrent TCP-like non-CMT flows on bottleneck links in wireless networks.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61001122, 61003283, in part by Beijing Natural Science Foundation of China under Grant No. 4102064, in part by the Natural Science Foundation of Jiangsu Province under Grant No. BK2011171, in part by the National High-Tech Research and Development Program of China (863) under Grant No. 2011AA010701, and in part by the Fundamental Research Funds for the Central Universities under Grant No. 2012RC0603, 2011RC0507.

References

[1] T. Alpcan, J. P. Singh and T. Basar, ”Robust Rate Control for Heterogeneous Network Access in Multihomed Environments,” *IEEE Trans. on Mobile Comput.*, vol. 8, no. 1, pp.41-51, Jan. 2009.

[2] C. H. Hsu and M. Hefeeda, ”Flexible Broadcasting of Scalable Video Streams to Heterogeneous Mobile Devices,” *IEEE Trans. on Mobile Computing*, vol. 10, no. 3, pp.406-418, March 2011.

[3] C. Xu, G.-M. Muntean, E. Fallon and A. Hanley, ”Distributed storage-assisted data-driven overlay network for P2P VoD services,” *IEEE Trans. on Broadcasting*, vol. 55, no. 1, pp.1-10, March 2009.

[4] L. Zhou, H. C. Chao and A. V. Vasilakos, ”Joint Forensics-Scheduling Strategy for Delay-Sensitive Multimedia Applications over Heterogeneous Networks,” *IEEE J. Select. Areas Commun.*, vol. 29, no. 7, pp.1358-1367, August 2011.

[5] R. Stewart, ”Stream Control Transmission Protocol(SCTP),” *RFC 4960*, 2007.

[6] R. Fracchia, C. Casetti, C. F. Chiasserini and M. Meo, ”WiSE: Best-Path Selection in Wireless Multihoming Environments,” *IEEE Trans. on Mobile Comput.*, vol. 6, no. 10, pp.1130-1141, Oct. 2007.

[7] T. Dreibholz, E. P. Rathgeb, I. Rungeler, R. Seggelmann, M. Tuxen and R. Stewartm, ”Stream Control Transmission Protocol: Past, Current and Future Standardization Activities,” *IEEE Communications Magazine*, vol. 49, no. 4, pp.82-88, Apr. 2011.

[8] T. D. Wallace and A. ShamiT, ”A Review of Multihoming Issues Using the Stream Control Transmission Protocol,” *IEEE Communications Surveys & Tutorials*, vol. pp, no. 99, pp.1-14, 2011.

[9] R. Stewart, Q. Xie, M. Tuexen, S. Maruyama and M. Kozuka, ”Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration,” *RFC 5061*, Sep. 2007.

[10] 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.

[11] P. Natarajan, N. Ekiz, P. D. Amer and R. Stewart, ”Concurrent Multipath Transfer during Path Failure,” *Computer Network*, vol. 32, no. 15, pp.1577-1587, Sep. 2009.

[12] C. Xu, E. Fallon, Y. Qiao, L. Zhong and G.-M. Muntean, ”Performance Evaluation of Multimedia Content Distribution Over Multihomed Wireless Networks,” *IEEE Trans. on Broadcasting*, vol. 57, no. 2, pp.204-215, June 2011.

[13] C. M. Huang and M. S. Lin, ”Fast Retransmission for Concurrent Multipath Transfer (CMT) over Vehicular Networks,” *IEEE Communication Letters*, vol. 15, no. 4, pp.386-388, April 2011.

[14] C. M. Huang and M. S. Lin, ”Multimedia Streaming using Partially Reliable Concurrent Multipath Transfer for Multihomed Network,” *IET Commun.*, vol. 5, no. 5, pp.587-597, March 2011.

[15] C. Xu, E. Fallon, Y. Qiao, G. -M. Muntean, X. Li and A. Hanley, ”Performance Evaluation of Distributing Real-time Video over Concurrent Multipath,” *Proc. IEEE WCNC* , 2009.

[16] B. Wang, W. Feng, S. Zhang and H. Zhang, ”Concurrent Multipath Transfer Protocol Used in Ad Hoc Networks,” *IET Communications* , vol. 4, no. 7, pp.884-893, April 2010.

[17] W. Yang, H. W. Li, F. H. Li, Q. Wu and J. P. Wu, ”RPS: Range-based Path Selection Method for Concurrent Multipath Transfer,” *Proc. ACM IWCMC*, Caen, French, June 2010.

[18] 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, March 2011.

[19] S. Li, Y. Qin and H. Zhang, ”Distributed Rate Allocation for Flows in Best Path Transfer Using SCTP Multihoming,” *Telecommunication Systems*, vol. 44, no. 1, pp.81-94, Jan. 2010.

[20] E. Yilmaz, N. Ekiz, P. Natarajan, P. D. Amer, J. T. Leighton, F. Baker and R. R. Steward, ”Throughput Analysis of Non-Renegable Selective Acknowledgement (NR-SACKs) for SCTP,” *Computer Communications*, vol. 33, no. 16, pp.1982-1991, Oct. 2010.

[21] T. Dreibholz, M. Becke, E. P. Rathgeb and M. Tuxen, ”On the Use of Concurrent Multipath Transfer over Asymmetric Paths,” *Proc. IEEE GLOBECOM* , 2010.

[22] H. Adhari, T. Dreibholz, M. Becke, E. P. Rathgeb and M. Tuxen, ”Evaluation of Concurrent Multipath Transfer over Dissimilar Paths,” *Proc. IEEE WAINA* , 2011.

[23] L. Cui, S. J. Koh and W. J. Lee, ”Fast Selective ACK Scheme for Throughput Enhancement of Multi-homed SCTP Hosts,” *IEEE Communications Letters*, vol. 14, no. 6, pp.587-589, June 2010.

[24] C. Fortuna and M. Mohorcica, ”Trends in the development of communication network: Cognitive networks,” *Computer Network*, vol. 53, no.25, pp.1354-1376, June 2009.

[25] Y. L. Hong, W. Q. Meeker and L. A. Escobar, ”The Relationship Between Confidence Intervals for Failure Probabilities and Life Time Quantiles,” *IEEE Trans. on Reliability*, vol. 57, no. 2, pp.260-266, June 2008.

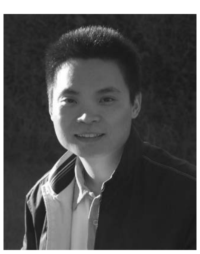
[26] UC Berkeley, LBL, USC/ISI and Xerox Parc, Ns-2 documentation and software, version 2.35, Available from: www.isi.edu/nsnam/ns, Nov. 2011.

[27] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely and S. C. Diot, ”Packet-level traffic measurements from the Sprint IP backbone,” *IEEE Network*, vol. 17, no. 6, pp.6-16, Dec. 2003.

[28] MSU Graphics&Media Lab, Video Group, MSU codecs, Available from: www.compression.ru/video.

[29] T. Dreibholz, H. Adhari, M. Becke and E. P. Rathgeb, ”Simulation and Experimental Evaluation of Multipath Congestion Control Strategies,” *Proc. IEEE AINA*, 2012.

[30] P. Amer, M. Becke, T. Dreibholz, N. Ekiz, J. Iyengar, P. Natarajan, R. Stewart and M. Tuexen, ”Load Sharing for the Stream Control Transmission Protocol (SCTP),” *Internet Draft,* IETF, <draft-tuexen-tsvwg-sctp-multipath-04>, March 2012, work in progress.



Changqiao Xu received his Ph.D. degree from Institute of Software, Chinese Academy of Sciences (ISCAS) in Jan 2009. He was an assistant research fellow in ISCAS from 2002 to 2007, where he held role as a project manager in the research and development area of communication networks. During 2007-2009, he worked as a researcher in Software Research Institute at Athlone Institute of Technology, Ireland. He joined Beijing University of Posts and Telecommunications (BUPT) in Dec. 2009, and was an assistant professor from 2009 to 2011. He is currently an associate professor with the Institute of Network Technology, and associate director of the Next Generation Internet Technology Research Center at BUPT. His research interests include wireless networking, multimedia communications and next generation Internet technology. He is member of IEEE.

Tianjiao Liu received her B.S. degree in computer science from Beijing University of Posts and Telecommunications (BUPT), China in 2010. She is currently working toward the M.S. degree in the Institute of Network Technology at BUPT. Her research interests include cognitive network, wireless communication and multimedia transmission over wired/wireless network.

Jianfen Guan received his Ph.D. degrees in communications and information system from the Beijing Jiaotong University, Beijing, China, in Jan. 2010. He is a lecturer in the Institute of Network Technology at Beijing University of Posts and Telecommunications (BUPT), Beijing, China. His main research interests focus around mobile IP, mobile multicast and next generation Internet technology.

Hongke Zhang received his Ph.D. degrees 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 author of eight books written in Chinese and the holder of more than 40 patents. He was the chief scientist of a National Basic Research Program (“973” program). He is now the head of Institute of Network Technology at Beijing University of Posts and Telecommunications (BUPT), Director of the Next Generation Internet Technology Research Center at BUPT, and the Director of National Engineering Laboratory for Next Generation Internet Interconnection Devices at BJTU.

Gabriel-Miro Muntean received the Ph.D. degree from Dublin City University, Ireland for research in the area of quality-oriented adaptive multimedia streaming in 2003. He is a Lecturer with the School of Electronic Engineering at Dublin City University, Ireland. Dr. Muntean is Co-director of the DCU Performance Engineering Laboratory research group, Director of the Network Innovations Centre, part of the Rince Institute Ireland and Consultant Professor with Beijing University of Posts and Telecommunications (BUPT), 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 personalised e-learning. Dr. Muntean has published over 120 papers in prestigious international journals and conferences, has authored a book and ten book chapters and has edited four other books. Dr. Muntean is 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. He is a member of IEEE, and IEEE Broadcast Technology Society.