Journal of Network and Computer Applications 35 (2012) 1674–1690

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Bandwidth aggregation in heterogeneous wireless networks: A survey of current approaches and issues

Allen L. Ramaboli *, Olabisi E. Falowo, Anthony H. Chan

Department of Electrical Engineering, University of Cape Town, Rondebosch 7701, South Africa

ARTICLE INFO

Article history:
Received 4 November 2011
Received in revised form
29 May 2012
Accepted 29 May 2012
Available online 13 June 2012

Keywords:
Bandwidth aggregation
Reordering
Reorder buffer
Heterogeneous wireless networks

ABSTRACT

Future wireless networks are envisaged to consist of a variety of integrated and jointly managed radio access technologies (RATs). This is motivated by the complementary features of the individual RATs. When in the overlapping coverage of the integrated RATs, a multimode terminal can use them simultaneously, thus aggregating bandwidth to enhance performance of high-bandwidth applications. However, there are challenges that must be addressed to achieve efficient bandwidth aggregation. Packet reordering is the most dominant challenge. Packet reordering can lead to excessive delays that can affect real-time applications; it can also affect throughput of TCP applications adversely. To circumvent the reordering problem and other challenges associated with simultaneous use of the terminal's multiple interfaces, bandwidth aggregation solutions are developed. This paper reviews existing bandwidth aggregation solutions in heterogeneous wireless networks. Challenges and several open research issues in the design of bandwidth aggregation approaches are also outlined. To the best of our knowledge, this is the first comprehensive review of existing bandwidth aggregation techniques in heterogeneous wireless networks. This paper, therefore, provides important lessons and information from current bandwidth aggregation solutions, which can be used to guide the development of more efficient bandwidth aggregation approaches.

Contents

* Corresponding author. Tel.: +27 79 635 2580.
E-mail addresses: allen@crg.ee.uct.ac.za (A.L. Ramaboli), bisi@crg.ee.uct.ac.za (O.E. Falowo), h.a.chan@ieee.org (A.H. Chan).

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2012.05.015

1. Introduction

In recent years, we have witnessed the emergence of a plethora of radio access technologies (RATs) with diverse transmission features and capabilities. Examples include Worldwide Interoperability for Microwave Access (WiMAX), Wireless Fidelity (WiFi) and Long Term Evolution (LTE). Nonetheless, radio resources in wireless networks are still scarce. Therefore, network operators find it challenging to meet the high-bandwidth demands that are posed by emerging multimedia applications, such as video streaming, online gaming and high definition television. One of the visions of future wireless networks for coping with the high-bandwidth demands is the integration of different radio access technologies to form a heterogeneous wireless network.

A heterogeneous wireless network allows network operators to combine the complementary advantages of individual RATs, thus paving a way for improved quality of service (QoS) provisioning and increased revenue (Lopez-Benitez and Gozalvez, 2011). In a heterogeneous wireless network, subscribers with multi-interface (multimode) terminals can toggle between different RATs and select the RAT that best suits the characteristics and requirements of their applications. If there is no single RAT that offers enough bandwidth to meet the application's requirements, two or more RATs can be selected, and their offered bandwidth can be aggregated to create a single logical link that has enough bandwidth to serve the application. Bandwidth aggregation, therefore, can provide an opportunity for network operators to scale up network capacity to meet high-bandwidth demands and enhance performance in terms of increased application throughput and reduced latency (Kaspar et al., 2009; Miu et al., 2003). Moreover, bandwidth aggregation can increase service reliability, improve fault tolerance by transmitting redundant data over multiple RATs (Miu, 2006), and enhance mobility by combining the coverage areas of individual RATs.

For a multimode terminal to exploit the aggregated bandwidth capacity in a heterogeneous wireless network, it needs to connect to the selected RATs simultaneously. Then, its application traffic can be transmitted using one RAT at a time and dynamically switching the traffic to the best RAT any number of times during the transmission period, or the application can be split into smaller streams, which can be transmitted in parallel via the selected set of RATs. While this can significantly boost operators' network capacity for enhanced service provisioning, it brings along challenges that need to be addressed. For instance, when packets belonging to the same application are transmitted simultaneously via multiple paths (RATs) with different latencies, they are likely to arrive at the receiver out of the intended order, and this can adversely affect performance of real-time and TCP applications (Chebrolu and Rao, 2002). Also, activating multiple network interfaces on a multimode terminal may significantly increase battery power consumption, thereby shortening the terminal's battery lifetime and risking premature transmission termination. To deal with the challenges, bandwidth aggregation solutions have been and continue to be developed.

This paper provides a comprehensive review of the current bandwidth aggregation solutions in heterogeneous wireless networks; particularly, we focus on protocols, traffic scheduling and distribution schemes designed to exploit the aggregated wireless bandwidth capacity. The aim of the paper is to familiarize the readers with the state-of-the-art research on bandwidth aggregation in wireless networks.

The remainder of the paper is organized as follows. Section 2 presents an overview of bandwidth aggregation including definition, benefits, challenges and key functional components of a bandwidth aggregation solution. Section 3 presents the proposed taxonomy of bandwidth aggregation solutions. A comprehensive review of existing bandwidth aggregation approaches in heterogeneous wireless networks is presented in Sections 4 and 5. Section 6 is a summary of the surveyed approaches and open research issues on bandwidth aggregation in heterogeneous wireless networks. Section 7 concludes the paper.

2. Overview of bandwidth aggregation in heterogeneous wireless Networks

Bandwidth aggregation can be defined as harvesting and aggregating of offered bandwidth from multiple RATs to create a high-speed logical link. The link can then be used by a multimode terminal for applications with high-bandwidth demands. Fig. 1 shows sample high-level architecture to support bandwidth aggregation in a heterogeneous wireless network. The figure depicts a heterogeneous wireless network with two RATs that are connected through the Internet Protocol (IP) backbone, forming loosely-coupled network architecture

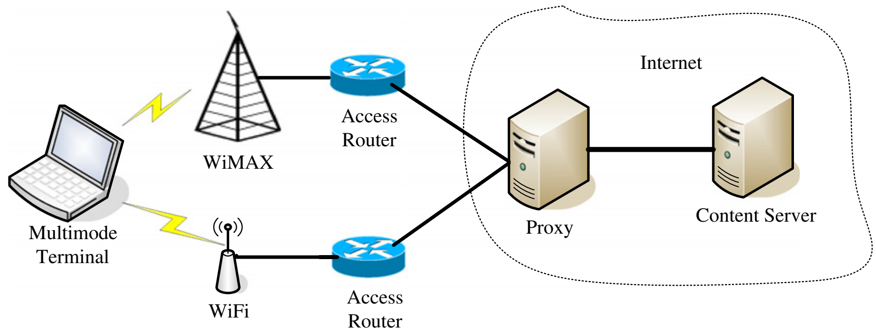

Fig. 1. Architecture to support bandwidth aggregation in heterogeneous wireless networks.

(Hossain, 2008), where each RAT retains its core network. Another way to integrate the RATs is to connect them through the same core network, thus creating a tightly-coupled heterogeneous wireless network (Hossain, 2008). A multimode terminal, when it is in the overlapping region of the RATs, can use them simultaneously. This is possible through multi-homing (Map et al., 2011), which gives the terminal ability to maintain multiple IP connections to multiple RATs.

A multimode terminal's ability to use multiple RATs at the same time is, however, not enough to efficiently utilize the aggregated bandwidth capacity. This is because the varying delay, bandwidth and loss characteristics of the different RATs can result in data transmission anomalies, such as packet reordering, which can affect application performance negatively. Packet reordering, especially, can introduce undesirable delays for real-time applications and unnecessary retransmissions for TCP applications (Chebrolu and Rao, 2002). The anomalies can be addressed by designing appropriate bandwidth aggregation solutions, aiming at distributing application traffic across multiple RATs, ensuring that data units from the same application arrive at the receiver in correct order. Besides in-order delivery, bandwidth aggregation solutions should target efficient load balancing and resource utilization. Bandwidth aggregation solutions are the focus of this paper.

2.1. *Benefits of bandwidth aggregation*

Aggregating effective offered bandwidth from individual RATs to create a high-bandwidth logical link has several benefits to both the operator and subscribers with multimode terminals. The benefits are discussed below.

2.1.1. *Increased throughput*

In a heterogeneous wireless network, when an application requires higher throughput than individual low-bandwidth RATs can provide, the bandwidth offered by the RATs can be aggregated to create a larger logical link with enough bandwidth to meet the desired throughput guarantees. For instance, consider two RATs, one offering 100 kbps and the other providing 80 kbps of bandwidth. The simultaneous use of the RATs results in 180 kpbs of total bandwidth. To exploit the aggregated bandwidth capacity, application traffic can be split and transmitted simultaneously over the RATs. This has been shown to yield throughput as high as the sum of throughput of the individual RATs (Bazzi et al., 2008). Also, end-to-end delay can be minimized as a direct effect of large aggregated bandwidth capacity (Miu et al., 2003), thereby creating an opportunity for real-time data transmission.

2.1.2. *Improved packet delivery and reliability*

Besides performance guarantees in terms of throughput and delay, bandwidth aggregation can be used to improve packet delivery and reliability (Miu, 2006; Al et al., 2009). That is, data packets can be duplicated and transmitted over multiple RATs to the same destination. When copies of the same packet arrive at the receiver, the error-free copy will be used, while the erroneous one will be discarded. If the packet copies both have errors, error correction and frame combining (Miu, 2006) schemes can be used to remove the errors, thus improving packet delivery. Reliability can be accomplished by maintaining redundant RAT connections so that data traffic can be switched to unused connections when link failures occur.

2.1.3. *Load balancing*

The ability to use more than one transmission path (RAT) simultaneously helps to ease load on one particular link by dispersing traffic over several links. The load can be distributed evenly or unevenly among the available paths; this is dictated by dynamic traffic and network conditions. While distributing traffic over multiple RATs, the traffic distribution policy should preserve correct packet order; otherwise, performance will degrade. Load balancing can be performed at per-flow level or per-packet level (Veitch et al., 2009). Per-flow load balancing confines packets of the same flow to the same path (Veitch et al., 2009; Cheng et al., 2008; Venkatasubramanian and Gopalan, 2009). This can lead to poor load balancing when traffic flows have different sizes. However, packet reordering can be minimized since packets of the flow travel the same path in a first-in first-out (FIFO) manner.

Per-packet load balancing, on the other hand, allows packets of the same flow to take different paths to the same destination, thus ensuring more efficiently balanced traffic load and preventing the formation of packet clusters in the network (Venkatasubramanian and Gopalan, 2009). However, per-packet load balancing over multiple paths that have different delay characteristics is more likely to cause high data reordering. As it was mentioned, packet reordering at the receiver is not desirable since it can lead to poor performance of TCP and real-time applications.

2.1.4. *Low-cost capacity increase*

When link capacity is to be increased, there are usually two options: either to purchase and deploy additional infrastructure or to aggregate two or more of the low-bandwidth links to create a large virtual link that can be used to serve demanding applications. The latter option is cheaper, and it offers a quicker solution to deal with capacity demands while waiting for procurement of new infrastructure. It also prolongs the lifetime of older link equipment, thus protecting investments in existing infrastructure.

2.2. *Major challenges of bandwidth aggregation*

As outlined above, using aggregated bandwidth capacity can bring benefits in the form of high throughput, reliability, and load

Sender	Receiver
1 2 3 4 5 6	1 **4 5 3** 2 6

Fig. 2. Packet reordering illustration.

balancing. However, there are some challenges that need to be addressed in order to efficiently reap the benefits. The main challenges are: Packet reordering and increased power consumption. The challenges are described below.

2.2.1. *Packet reordering*

Packet reordering occurs when the order of packets of the same flow at the receiver is different from the order of the same packets at the sender (Przybylski et al., 2005). That is, the sequence number of the arriving packet of a flow is lower than the sequence number of the consecutive packet that has already arrived at the receiver. The example in Fig. 2 illustrates packet reordering at the receiver. The reordered packets are shown in bold.

Packet reordering in heterogeneous wireless networks can be caused by simultaneous transmission of packets of the same flow across multiple links that have different end-to-end delays and transmission rates, resulting into consecutive packets of a flow arriving at the receiver out of the intended order. Packet reordering can adversely affect the performance of any real-time applications (Przybylski et al., 2005). The time taken to put the received packets in correct order increases the packets' end-to-end delays, thus causing some of the packets of real-time applications to miss their respective deadlines and get discarded. Transmission Control Protocol (TCP) can also be affected by packet reordering. TCP can allow packet reordering by a maximum of two positions, and it can be corrected by inherent re-sequencing mechanism (Przybylski et al., 2005; Bennett et al., 1999). However, packet reordering beyond two positions will be interpreted as a loss, and the transmission window will be reduced. Consequently, the aggregated capacity will be underutilized, and the application throughput may drop drastically. An efficient bandwidth aggregation solution is, therefore, one that includes mechanisms to minimize packet reordering to alleviate its effects.

2.2.2. *Increased battery power consumption*

Limited battery power has always been the weakness of handheld devices. During operation and idle periods, a handheld terminal consumes a significant amount of power, and its battery gets depleted. When the terminal is equipped with multiple radio interfaces, its battery power consumption can increase even more (Chowdhury et al., 2009; Mahkoum et al., 2009; Koudouridis et al., 2005). This can reduce the terminal's operational lifetime, risking premature transmission termination. Therefore, an efficient bandwidth aggregation solution should include mechanisms to minimize a multimode terminal's battery power consumption cost, thereby prolonging the battery lifetime to ensure uninterrupted multipath communication.

2.3. *Packet reordering metrics*

Since packet reordering is the dominating challenge to the design of efficient bandwidth aggregation solutions, it is important for designers to know how to quantify it so that they can reliably assess performance of their solutions and make appropriate optimizations to ensure improved performance. Several metrics have been proposed to quantify packet reordering. Here, we briefly present Reorder Density (RD) and Reorder Buffer-occupancy Density (RBD), which, according to Jayasumana et al. (2008), are deemed the most important.

2.3.1. *Reorder density*

Reorder Density measures the amount and extent of out-of-order packets in a sequence of packets arriving at the receiver, i.e., the distribution of displaced packets, which is normalized to the number of packets in the sequence. An early packet has a negative displacement and a late packet has a positive displacement. RD has been demonstrated to provide comprehensive information about the amount of packet reordering. A more detailed coverage of RD can be found in Jayasumana et al. (2008).

2.3.2. *Reorder buffer-occupancy density*

Reorder Buffer-occupancy Density (RBD) is another way of measuring reordering. RBD measures buffer occupancy frequencies normalized to the number of non-duplicate packets in the arriving packet sequence (Jayasumana et al., 2008). RBD is especially important for predicting the amount of resources (e.g., buffer space) required to correct packet reordering.

2.4. *Funtional components of a bandwidth aggregation architecture*

To address the challenges and reap the benefits of simultaneous use of multiple radio interfaces on a multimode terminal, efficient bandwidth aggregation architecture should be developed. Typical bandwidth aggregation architecture consists of, but not limited to the functional elements (Chebrolu and Rao, 2002; Wang et al., 2009): radio interface selection, scheduling algorithm, packet re-sequencing unit, and link monitor. Fig. 3 shows a sample block configuration of these elements.

Interface selection is responsible for picking an optimal set of radio interfaces to achieve the desired bandwidth aggregation. This can be done by constructing a cost function, taking into consideration the available radio interfaces and their characteristics (bandwidth, delay, loss, etc.). Then, a set that better optimizes the cost function can be selected for bandwidth aggregation.

Scheduling algorithm is the core component of any bandwidth aggregation architecture. Its function is to decide how packets can be striped and the order in which they are scheduled for transmission over multiple links, ensuring that they arrive at the receiver within their decoding deadlines. For instance, packets can be striped and distributed to the available links using round robin (RR) policy, which allocates packets in order from the first link to the last. A variety of scheduling policies for bandwidth aggregation have been proposed, and we discuss them in great details in Sections 4 and 5.

Packet Re-sequencing unit usually resides on the receiver side, and its primary purpose is to assemble arriving packets into the original packet stream, ensuring that the packets follow according to their sequence numbers. A re-sequencing unit maintains a buffer to hold displaced packets until they can be correctly ordered and delivered to higher layers. Determining the size of a re-sequencing buffer is an important design issue; a buffer that

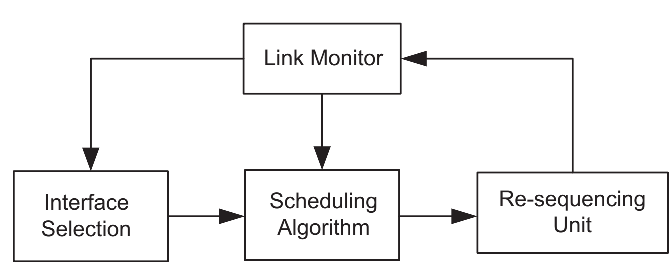

Fig. 3. Functional components of bandwidth aggregation architecture.

is too large can hold out-of-sequence packets for too long, thereby increasing their end-to-end delays. On the other hand, a buffer that is too small can quickly overflow, resulting in significant packet losses. Hari et al. (1999) show how reorder buffer size can be determined.

Link Monitor is crucial for assuring significant bandwidth aggregation performance gains (Sharma et al., 2004). The link monitor accurately measures the link dynamics and avails the measurements to other components of the bandwidth aggregation architecture, thus enabling the selection of a proper set of interfaces and the derivation of optimal packets schedules over the selected interfaces.

3. Classification of bandwidth aggregation solutions in heterogeneous wireless networks

Bandwidth aggregation solutions can be classified according to the following criteria.

3.1. Adaptation to dynamic conditions

Bandwidth aggregation techniques, whose operation is influenced by varying link and traffic characteristics, are classified as adaptive. Adaptive bandwidth aggregation approaches can better optimize throughput and link utilization by dynamically configuring interface selection and scheduling decisions in a way that matches traffic and network dynamics. Bandwidth aggregation solutions that operate based on constant link and traffic conditions are called non-adaptive. Non-adaptive bandwidth aggregation approaches may achieve lower performance gains, especially when link and traffic conditions change, as they do not have the ability to adjust their resource allocation and traffic schedules.

3.2. TCP/IP protocol stack layers

Bandwidth aggregation can be addressed at different layers of the network protocol stack, that is, application, transport, and network and link layer. These are described below and summarized in Table 1.

3.2.1. Application layer solutions

At the application layer, application data can be demultiplexed over multiple heterogeneous paths in order to improve performance and reliability. The obvious hurdle to the use of application layer bandwidth aggregation techniques is the fact that application developers need to embed functions to handle multiple interfaces in the application. This can increase the application's complexity. It may also impede compatibility with existing applications. Examples of applications that can exploit the aggregated bandwidth include XFTP (Allman et al., 1996) and GridFTP (Allcock, 2003).

3.2.2. Transport layer solutions

Most transport layer solutions focus on TCP, where application data can be fragmented into multiple TCP sessions, which can be transmitted via multiple heterogeneous channels. TCP bandwidth aggregation offers reliable multipath transmission. However, it compromises interoperability between existing TCP based network infrastructure. Since TCP is not optimized for timely data delivery, real-time applications may not benefit from TCP bandwidth aggregation. Transport layer bandwidth aggregation can also be achieved using stream control transmission protocol (SCTP) (Stewart et al., 2004), which supports multi-streaming and multi-homing to allow establishment of multiple transport layer connections, which can be used to send segments of the same application via multiple physical links. Bandwidth aggregation based on TCP and SCTP is not immune to reordering. Therefore, appropriate mechanisms to correct reordering are necessary.

3.2.3. Network layer solutions

Network layer based bandwidth aggregation offers packet level traffic distribution over multiple interfaces. Due to the flexibility of the Internet Protocol, aggregating bandwidth of heterogeneous links can be achieved across different domains and infrastructure. However, the inherent behavior of IP packets makes network layer bandwidth aggregation more prone to out-of-sequence packet arrival. This may be mitigated by additional packet processing, i.e., adding sequence numbers to enable re-sequencing at the receiver before packets are delivered to higher protocol layers. Network layer solutions are some of the most widely studied in the literature.

3.2.4. Data link layer solutions

Link layer bandwidth aggregation allows for multiple data link channels to be bundled into a single logical channel with larger capacity than the individual channels. Therefore, link layer protocol data units (PDUs), belonging to a single IP flow, can be split over the channels for higher throughput gains. Multilink PPP (Sklower et al., 1996) is an example of an architecture that enables link layer bandwidth aggregation. There is also the concept of a generic link layer (GLL) (Sachs, 2003), which emerged recently. The rationale behind the GLL is to offer universal link layer processing functions over multiple heterogeneous radio interfaces in order to exploit various forms of multi-radio diversity (Koudouridis et al., 2005) to improve performance. These approaches, however, require special hardware or software, and are therefore limited to local domains controlled by the same operator.

For the review presented in this paper, we use the classification criteria described above to characterize existing bandwidth aggregation approaches in heterogeneous wireless networks. We first categorize the approaches into non-adaptive and adaptive bandwidth aggregation solutions. Then, we use the network

Table 1
Bandwidth aggregation at different layers of the network protocol stack.

Layer	Description	Advantages	Disadvantages
Application	Application is aware of multiple interfaces, and can split the traffic into several application layer protocol data units, which can be transmitted simultaneously via the interfaces.	Finer granularity and efficient application specific optimizations due to full knowledge of application characteristics	Increased application complexity; compromised interoperability with existing applications; head-of-line blocking at the transport layer
Transport	Multiple transport layer connections are created to transmit application traffic.	Reliable multipath transmission in the case of TCP and SCTP	Compromised interoperability with existing TCP based infrastructure
Network	IP packets from the same transport layer session are transmitted across multiple network interfaces.	Transparent to higher layers; compatible with existing infrastructure	Poor TCP performance due to high packet reordering
Link	Multiple links are bundled into a single logical communication link.	Higher utilization of the aggregated capacity	Limited to tight-coupled networks belonging to the same operator

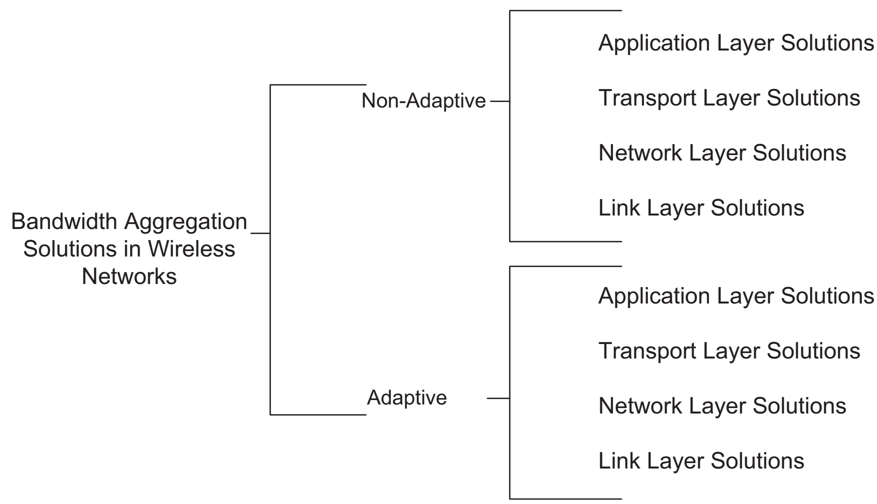

Fig. 4. Taxonomy of bandwidth aggregation solutions.

protocol stack criterion for further classification. Fig. 4 shows the proposed taxonomy of bandwidth aggregation solutions.

4. Non-adaptive bandwidth aggregation solutions

Non-adaptive bandwidth aggregation solutions enable data transmission over multiple heterogeneous links, assuming non-varying traffic and link dynamics. Non-adaptive bandwidth aggregation approaches can be designed and implemented at different layers of the network protocol stack. This section provides a review of current non-adaptive bandwidth aggregation solutions.

4.1. Application layer

Here, bandwidth aggregation solutions stripe application data into several application flows, which can be transmitted across multiple channels. Early work on application layer bandwidth aggregation was based on multiple logical channels belonging to the same interface. Such work includes: XFTP (Allman et al., 1996) and GridFTP (Allcock, 2003) to enable transfer of FTP data through multiple TCP connections; Parallel Sockets (PSockets) (Sivakumar et al., 2000) to create multiple parallel sockets for data transfer over multiple logical TCP connections. The performance of these proposals is limited by the available bandwidth offered by the interface. Some changes may be required to adopt the proposals to work over multiple wireless physical links.

One of the first application layer solutions to utilize multiple physical links is MuniSocket (Mohamed et al., 2002), which is a middleware solution that enables huge data transfers over multiple network interfaces. MuniSocket associates each path with sending and receiving thread pair. In addition, it maintains a counter to provide sequence numbers to identify different fragments of a message. The sending thread uses sequence numbers derived from the counter to prepare message fragments for transmission. The fragments can then be put in the appropriate transmission buffer. If the buffer is full, the fragments will be blocked until some space can be freed. Due to the heterogeneity of the paths, the fragments may arrive at the receiver out of their original order. To correct this, the receiving thread retrieves arriving segments and holds them in the receiver buffer until they can be ordered correctly. To ensure reliable multiple path transmission, MuniSocket uses acknowledgment triggered retransmission. For load balancing, only the threads that are connected to unloaded networks are allowed to accept fragments for transmission. Despite its elegance, MuniSocket cannot automatically adapt to varying network conditions; thus, its performance may not be optimal in highly variable wireless environments.

Application	
MPTCP	
Sub-flow(TCP)	Sub-flow(TCP)
IP	IP

Fig. 5. MPTCP Architecture (Nguyen et al., 2011).

4.2. Transport layer

The IETF has defined Multipath TCP (MPTCP) (Ford et al., 2011), which is a modified version of TCP. MPTCP is developed to utilize multiple physical paths simultaneously to improve throughput and resilience. The architecture of MPTCP is shown in Fig. 5. MPTCP divides application stream into several segments that can be transmitted over more than one interface at the same time. Segments through the same interface constitute a sub-flow. To establish a multipath connection between two entities, e.g., client and server, the MPTCP client sends a SYN segment through one of its interfaces to the server. One of the fields in the segment (MP_CAPABLE) indicates that the client supports MPTCP. If the server is MPTCP capable, it will respond with a SYN that has the MP_CAPABLE field. To add more interfaces, a SYN segment with MP_JOIN can be exchanged between appropriate interfaces. MPTCP maintains two types of sequence numbers: one at MPTCP level to enable in-order segment delivery, and the other at sub-flow level to manage segment retransmissions within individual flows.

In its current state, MPTCP does not include intelligent interface selection and traffic distribution mechanisms to alleviate the effects of heterogeneous path characteristics, such as packet reordering. Rather, MPTCP employs a buffer to hold out-of-order segments until they can be correctly ordered and sent to the application layer. The first implementation of MPTCP is reported in Barre et al. (2010). Performance evaluation is presented in Nguyen et al. (2011). It has been shown that MPTCP can utilize multiple paths to improve throughput and reliability. However, when the paths have different path characteristics, performance of MPTCP degrades.

4.3. Network layer

The round robin (RR) scheduling policy is a notable non-adaptive scheduling mechanism that can be used to schedule packets in multipath environments. The RR scheduler assigns packets of the same flow to multiple paths in equal portions and in a cyclic manner, assuming the paths have homogeneous transmission characteristics. It also assumes that packets have the same size. A sample operation of RR over multiple links is shown in Fig. 6. In the first scheduling round, packets 1 to 3 are allocated equally and in order to the links. Packets 4 to 6 are scheduled in the same manner in the second round.

The attractive feature of the RR scheduler is its simplicity—it has per packet computational complexity of O(1). However, in a more realistic scenario, where the packet size is variable and link rates are heterogeneous, the round robin scheduling mechanism performs poorly in terms of load balancing; for instance, when a low-bandwidth link gets, on average, larger packets to transmit than high-bandwidth links, the load cannot be efficiently balanced. Also, packets on the low-bandwidth link may encounter longer delays. Different packet delays that are a result of dissimilar link rates can lead to packets of the same flow reaching the receiver out-of-order, thus raising the need to implement a large buffer to correct the reordering, which may exacerbate end-to-end delays.

To support multipath packet transmission over links with different transmission rates, there are variants of RR, such as weighted round robin (WRR) (Zhang et al., 2009) and surplus round robin (SRR) (Hari et al., 1999), which can be used. A WRR scheduler distributes packets of the same flow over multiple links according to a normalized weight. The link with a larger weight carries more packets to the receiver in a scheduling round. A link weight can be defined as a function of link bandwidth. WRR can achieve better load balancing than the traditional RR. However, when packet lengths are variable, it may result in poor load balancing and high packet reordering. An improvement to WRR is SRR, which is a practical scheduler proposed to disperse variable-length packets over multiple links with heterogeneous capacities. The main purpose of SRR is to efficiently balance traffic load across multiple paths. This is achieved by transforming fair queuing algorithms into load sharing algorithms at the sender. However, SRR does not consider the order in which packets of the same flow are delivered to the receiver; therefore, it can introduce high packet reordering, which can adversely affect performance of TCP and real-time applications. In fact, SRR solely relies on a re-sequencing mechanism (buffer) implemented at the receiver to restore the original packet order; this may not be favorable for mobile terminals with limited resources.

Kaspar et al. (2009), based on practical measurements, quantify the impact of the simultaneous use of multiple heterogeneous interfaces on packet reordering. They show that packet reordering is higher in multipath transmission than in common Internet connections. Then, they propose a static packet scheduler to reduce packet reordering in order to efficiently reap the benefits of multipath transmission. The proposed scheduler assumes prior knowledge of estimated delays on the selected paths. The rationale of the proposed scheduler is to delay packets on the faster link to ensure that they experience similar delays to packets on the slower link. Compared to round robin, the scheduler achieves lower packet reordering delay. However, like other static scheduling algorithms, the proposed scheduling scheme may experience throughput degradation and increased packet reordering when link conditions vary over time.

Kim et al. (2008) present a bandwidth aggregation scheme that consists of two functional units: bandwidth estimation and packet partition scheduling. The bandwidth estimation determines the amount of traffic in bytes that can be transmitted on a path without causing congestion. The partition packet scheduler decides how packets should be allocated to appropriate paths to ensure that the load can be efficiently balanced. To achieve this, the scheduler maintains a partition counter (a weight assigned to a link based on available bandwidth) to determine if a link can accept any more bytes of the packet. If the current link has a lower partition counter and cannot accommodate the incoming packet, the next link with a larger counter will be used to serve the packet. Even though the proposed scheme can achieve more efficient load balancing than simple weighted multipath schedulers, such as WRR, its performance may be adversely affected by possible high packet reordering in rapidly varying channel dynamics.

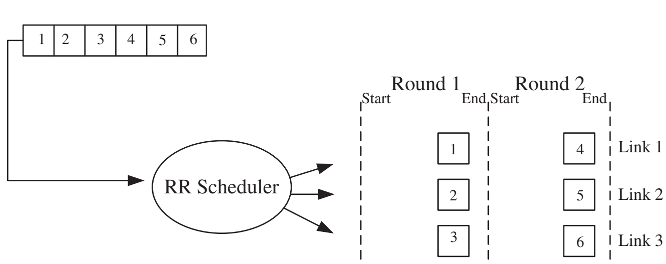

Fig. 6. The operation of RR at the sender.

4.4. Link layer

Koudouridis et al. (2005) introduce the concept of generic link layer (GLL) to support multi-radio cooperation at the radio level. Among other features, the GLL enables multi-radio transmission diversity (MRTD), which is the sequential or parallel use of the aggregated bandwidth capacity to transmit a traffic flow. The important GLL functions for realizing different forms of MRTD include: access selection to schedule users' data on appropriate RATs; performance and monitoring to provide performance information to the accession selection unit so that transmission schedules that match network and traffic dynamics can be drawn; flow and error control to regulate data flow and ensure error-free transmission on the links.

Even though the GLL concept is an innovative step towards enabling efficient bandwidth aggregation in heterogeneous wireless networks, challenges in the form of packet reordering and unbalanced load distribution are still a concern. This is especially true for naïve parallel MRTD that blindly disperses packets across multiple links (Yaver and Koudouridis, 2009). Yaver and Koudouridis (2009) show that parallel MRTD without adapting to varying channel and traffic conditions can only achieve average delay and drop probability, which are, respective averages of average delay and drop probability of individual links. Moreover, naïve parallel MRTD does not show any quantitative gains in throughput. Therefore, to enhance the performance of parallel MRTD, intelligent and dynamic traffic distribution mechanisms are needed.

5. Adaptive bandwidth aggregation solutions

Adaptive solutions address the pitfalls of non-adaptive bandwidth aggregation schemes and take into consideration varying traffic and link conditions to derive optimal resources allocation and scheduling decisions. Similar to non-adaptive, adaptive bandwidth aggregation schemes can be designed and implemented at different layers of the network protocol stack.

5.1. Application layer

Luo et al. (2003) introduce one of the first application layer based bandwidth aggregation solutions to transmit data units of the same application through tight-coupled wireless LAN (WLAN) and 3 G networks. The proposed solution implements a joint session scheduler (JOSCH) to split application data into important and optional streams. For instance, video traffic can be striped into base and enhancement layer information; and Hyper Text Transfer Protocol (HTTP) traffic can be split into main and inline objects. The important streams are delivered via the reliable UMTS network whereas optional information is transmitted on the WLAN. Transmitting traffic streams from the same application simultaneously across heterogeneous links may lead to synchronization loss, thus affecting the playback performance of the application. To deal with this, Luo et al. propose periodic interactive approach, where a small data packet, known as a training sequence, is used to estimate average delay on each path. The average delay information is then used to adjust traffic quantum sizes for the links in order to reduce delay difference. Flow-level traffic splitting adopted by Luo et al. is coarse-grained and may suffer inefficient load balancing when the flows have distinct sizes. Furthermore, the training sequence used to derive delay estimates is too small to provide reliable delay estimates.

Sharma et al. (2007) present three heuristic-based algorithms, Layer-Priority Striping (LPS), Frame-Priority Striping (FPS) and Independent-Path Striping (IPS), to split video traffic over multiple channels to enhance service quality for a collaborating community of wireless terminals. Similar to (Luo et al., 2003), LPS is used to transmit layer-encoded video streams over multiple paths, where a video is encoded into base layer l_0 containing information necessary to decode the video, and one or more enhancement layers ($l_i = i = 1,\ldots,n$) to enhance the perceived video quality. Layer index i corresponds to layer priority—the lower the index, the higher the priority. The LPS mechanism maps layer priority to path reliability; for instance, the base layer, with the highest priority, is allocated to the most reliable path, and the enhancement layers are assigned to any available paths according to their priorities. Video packets from the layers are then allocated to the respective channels based on WRR scheduling discipline.

FPS, on the other hand, can be used to transmit video traffic that can be divided into sub-streams according to frame types. For instance, MPEG (Orozco-Barbosa and Taeseop, 2000) video can be separated into I, P and B frames, where I frames have the highest priority, whereas B frames are assigned the lowest priority. Similar to LPS, The highest priority frames are transmitted through the most reliable channel. For multi-state video coding (MSVC) (Ekmekci and Sikora, 2003), IPS is proposed to exploit multiple paths. MSVC encodes a video stream into several substreams that can be decoded independently. However, information from one stream can be used to fix errors in another stream. Therefore, for optimal error correction, it is desirable that the substreams should have low correlation of loss and errors. The IPS achieves this by assigning the sub-streams to different channels. Like other flow-based load distribution schemes, the approaches studied by Sharma et al., especially LPS and FPS, may encounter load balancing problems when application flows have different sizes.

Xue et al. (2008) introduce a bandwidth aggregation scheme, which uses distributed re-enforcement learning to disperse traffic load on multiple paths. The purpose of the scheme is to provide QoS guarantees and efficient load balancing. The proposed scheme stripes scalable services, such as layer-encoded video, into sub-streams that can be transmitted through multiple links and combined into the original stream at the receiver. To accomplish this, each terminal is assigned an agent to select an appropriate traffic distribution strategy. The agents learn and share experience with each other using Q-values stored in the Q-repository on the network. The traffic distribution problem is formulated as a set of states, actions, and rewards. The states represent session arrivals and departures; actions are simply traffic allocation and splitting strategies triggered by session arrivals; rewards describe user experience (UE) due to certain actions. The task of the agents is to learn the current state of the network and then schedule sessions on multiple paths to optimize UE. The proposed re-enforcement learning based scheme achieves higher service quality than random traffic distribution algorithms. However, its flow-level load balancing is coarse-grained and may be inefficient when flows have different sizes.

Qureshi and Guttag Horde (2005) propose a middleware called Horde, which gives an application control over how to utilize the aggregated bandwidth capacity. Guttag Horde uses Network Channel Manager modules to monitor channel characteristics of the available networks. The modules also perform congestion control. Channel information from the channel managers can be used by the scheduler to develop optimal transmission schedules, ensuring that application data units are well balanced over the channels and can arrive at the receiver in correct order. Most importantly, the transmission schedules are influenced by the application's performance objectives. The objectives are presented using a specification language as demonstrated in Fig. 7.

The objective specification in Fig. 7 tells the scheduler that the application data unit to be scheduled is a video frame of type I from stream 17; the frame must have loss probabilities lower than other frame types. Based on the specification, the scheduler can derive a schedule that better optimizes the application's objective. Applications that do not specify any objectives are scheduled in a round robin fashion. Even though Horde is elegant and efficient, it imposes increased complexity on application development, and existing applications would need to be rewritten to leverage Horde's functionality.

Kaspar et al. (2010) study the challenges and benefits of using multiple access networks simultaneously to improve video-on-demand playout. The study is based on an adaptive application layer multilink solution that can enable progressive download of multimedia data over multiple network interfaces to the receiver. The authors identify the major challenges of progressively downloading multimedia files over multiple networks as long startup latency and large buffer requirements. Startup latency simply refers to the waiting time before the downloaded video can be played out. The wireless link variations make it difficult to predict optimal startup latency. However, the desired playout bitrate,

```
Objective    {
    context   {
        adu: foo { (stream_id == 17) &&
                   (frame_type == "I")   }
        adu: bar { (stream_id == 17) &&
                   (frame_type != "I")   }
    }
    goal   { prob (foo::lost?)
             < prob(bar::lost?) }
    utility  { foo { 100 } }
}
```

Fig. 7. Application's objective specification (Qureshi and Guttag Horde, 2005).

which is bounded by throughput over the aggregated capacity, can be used to estimate the optimal startup latency. By setting high playout bitrate (but below the aggregated throughput), startup latency can effectively be reduced, and this is important for small video clips, where the user expects instantaneous startup.

To keep the received data (usually out-of-order data) that cannot be played out yet, a buffer is used. The buffer size is reported to increase with segment size and the number of used network interfaces. A buffer that is too large is not desirable for handheld devices, which have limited resources, including finite battery power. There is also a correlation between segment size and startup latency. The important task is how to find an optimal segment size that can result in optimal startup latency and buffer size. While the required buffer size can be affected by the number of interfaces used, startup latency is affected more by the difference in delay between the interfaces. For instance, large delay difference can lead to long startup latency, which can negatively affect the perceived video quality. Therefore, finding the right number and combination of network interfaces is also an important design factor for achieving efficient bandwidth aggregation.

Based on observations from Kaspar et al. (2010), a pipelining approach (Kaspar et al., 2010) is proposed to meet the requirements of progressive video download and playback over multiple network interfaces. The proposed approach allows the client to make a request for the next byte range while downloading the current video segment, thus reducing the server's idle time and increasing the aggregated throughput for small segment sizes. Even though the approach can improve throughput when using a small segment size, there is a danger of choosing too small segment size, leading to the server processing a segment too fast and not allowing enough time for the client to pipeline the next request. To mitigate this problem, the amount of pipelined data must exceed the path's bandwidth-delay product. The proposed approach achieves this by pipelining multiple segments. However, the use of fixed segment size may lead to inefficient adaptation to link heterogeneity.

5.2. Transport layer

Casetti and Gaiotto (2004) present modifications to SCTP to support multipath transmission. They refer to the modified SCTP as Westwood SCTP (W-SCTP). The objective of W-SCTP is to efficiently balance load across multiple paths using a bandwidth-aware scheduler, which is implemented at the sender. W-SCTP maintains multiple send buffers, and each connection within the W-SCTP association is managed independently. The SCTP selective acknowledgments (SACKs) management is also modified to work with multiple send buffers. To transmit a data chunk, W-SCTP estimates the chunk's delivery time over all the available paths, and the path with the shortest delivery time is selected to serve the chunk. The process is repeated until the congestion windows of the available paths have been exhausted. At the receiver, a single association buffer is implemented to store data chunks received from multiple interfaces. Even though W-SCTP is a significant improvement to SCTP for support of bandwidth aggregation, it does not explicitly detail how to deal with segment reordering. Moreover, it does not capture all the important path characteristics, such as loss rates, when deciding the path over which a data chunk can be delivered. Thus, it may not attain optimal multipath transmission.

In Fiore and Casetti (2005), Casetti and Gaiotto extend their work in Casetti and Gaiotto (2004) to enable multipath transmission using SCTP with Partial Reliability. The new solution is called Westwood SCTP-PR. The main objective of Westwood SCTP-PR is to support real-time applications by exploiting SCTP's Partial Reliability extension. To avoid long delays, which may impede the transmission of real-time multimedia applications, the authors propose to disable SCTP retransmissions. Westwood SCTP-PR periodically maintains a list of available interfaces and frequently removes unusable paths from the list to improve robustness against unfavorable network conditions.

A load sharing SCTP (LS-SCTP) is presented in El A et al. (2004) to enable end-to-end data transport over multiple paths. The main functional elements of LS-SCTP are: flow and congestion control; path assignment and monitoring. LS-SCTP separates congestion and flow control by performing flow control on association basis, whereas congestion control is done on path basis; that is, the sender maintains a separate congestion control window for each link, thus providing the sender with a logical congestion window whose size is an aggregate of the individual congestion windows. To support load balancing across multiple paths, LS-SCTP defines a 'load sharing chunk', which is associated with a path by path identity (PID). PID identifies the path that is used to transmit the data chunk. Within the same path, data chunks are identified by path specific identifiers called path sequence numbers (PSNs). The path assignment module then allocates appropriate paths to the data chunks, assigning suitable PID and PSN for each chunk. Path assignment is done based on available bandwidth on the paths.

Instead of distributing data chunks over the paths in a round robin manner, which may limit multipath transmission throughput to the slowest path, LS-SCTP allocates chunks to the paths based on the current congestion window of a path; specifically, the ratio of congestion window to the round trip time is used. To mitigate the reordering that may occur as a result of different RTTs across the paths, LS-SCTP uses the association buffer at the receiver, whose size increases with the amount of reordering. However, a large buffer can increase end-to-end delays, thus affecting the efficiency of multipath communication. LS-SCTP adapts to varying path characteristics by frequently monitoring the paths to gather channel information and then adjust the allocation ratios accordingly. A similar bandwidth aggregation approach to LS-SCTP is concurrent multi-path SCTP (cmp-SCTP) in Liao et al. (2008). The glaring difference between the approaches is cmp-SCTP's use of multiple send buffers, where a send buffer is maintained for each path. The benefit of using separate send buffers is that when head-of-line (HOL) blocking occurs, it is pinned to a specific connection on a path and does not affect connections on other paths.

Hasegawa et al. (2005) introduce multipath TCP communication scheme called Arrival-Time matching Load-Balancing (ATLB) to deliver data segments to the receiver in correct order. ATLB assigns a score to each path based on estimated end-to-end delay; the lowest score corresponds to the lowest delay. Based on this, ATLB schedules a data segment on the path that has the lowest score. ATLB includes a reorder buffer at the receiver to deal with residual segment reordering that may occur. ATLB handles path failures by maintaining a timer that expires after a carefully set timeout period. If a segment, just after the timeout, is not received properly, the path is assumed to have failed. To recover from failure, ATLB probes the failed path periodically, calculating packet loss rate. If the loss rate is small enough, data transmission can resume on the path; otherwise, the path is removed the list of available paths. Despite its robustness against failure, ATLB may not accomplish optimal performance because the path scores are based only on end-to-end delays, omitting other important channel characteristics (ATLB uses loss rates only for recovering from failure, not for deciding the best path for a segment).

Hsieh et al. (2002) present a new transport layer protocol called parallel TCP (pTCP) to enable applications to exploit the mobile terminal's multiple interfaces for performance improvement. pTCP is implemented as a wrapper around a modified TCP called TCP-virtual (TCP-v). For every socket that an application opens, pTCP creates and maintains a TCP-v connection on each interface that is used to aggregate bandwidth. pTCP controls what data should be sent through the available interfaces, whereas TCP-v determines the volume of data that can be sent through the interfaces. Thus, reliability and congestion control can effectively be separated. pTCP stripes application data over active TCP-v connections based on available space in their congestion windows. That is, data can only be allocated to a TCP-v connection if there is enough space in the congestion window. The adopted congestion window based splitting assumes that the congestion window is a true reflection of the bandwidth-delay product, which may not be the case at times. Consequently, an over-estimation of the congestion window may drive the TCP-v link into congestion, resulting in undesirable delay and loss. When undesirable loss and delay occur on a path, pTCP reassigns the data on the affected TCP-v to another TCP-v with enough free space in the congestion window. However, if a path shuts down completely, data in the congestion window will stall. To deal with this, redundant striping of the data is performed. Even though pTCP may be efficient in handling congestion across the paths, it does not incorporate any mechanism to assure in-order data delivery to the receiver.

In Anand (2007), another new TCP-based multipath transport protocol – *concurrent multipath real-time TCP (cmpRTCP)* – is proposed to transmit real-time data streams via multiple paths. cmpRTCP uses congestion window manager to dynamically monitor congestion status of the available paths. The information from the congestion window manager is used by a real-time scheduler to efficiently distribute packets over the paths. The proposed real-time scheduler schedules a burst of packets across the paths in a round robin manner, filling each path to full capacity before moving onto the next one. This can, however, lead to unbalanced load and poor utilization of the paths; for instance, when the burst can fit in one path, the other paths will be idle. cmpRTCP reacts to missing packets by gathering the paths with missing packets, ordering them in the increasing number of missing packets, and filling them up in the same order. This can reduce the probability of loss in the next transmission rounds. cmpRTCP does not include any mechanism to ensure in-order delivery at the sender; instead, it relies on the receiver buffer to correct possible reordering. For every packet that is transmitted to the receiver, a real-time delay tolerance limit (RTDTL) is set, and the packets arriving beyond their limit are discarded from the buffer immediately. The packets that arrive within their RTDTL and in correct order are delivered to higher layers; otherwise, they are kept in the buffer until reordering can be corrected, or until their RTDTL expires. This ensures that the reorder buffer space is almost constant regardless of the amount of reordering.

Mirani et al. (2010) propose a scheduling algorithm, Forward Prediction Scheduling (FPS), to transmit multiple SCTP sessions over multiple links, ensuring that the segments traversing different paths arrive at the receiver in correct order. FPS consists of a scheduling module to dynamically estimate end-to-end latencies of the selected paths and then transmit the segments over the paths in a manner that can minimize reordering. The main idea behind FPS is to determine the amount of data that can be sent on the fast path before the arrival of data on the slow path. For instance, consider two paths i and j, with latencies RTT_i and RTT_j, respectively, where $\mathrm{RTT}_i < \mathrm{RTT}_j$. The latency of a path is a function of the path's round trip time (RTT). To ensure in-order delivery, the segments to be sent on j are advanced by the number of segments delivered on i before reception on j. The path latencies are updated according to Eq. (1).

$$\mathrm{RTT}_i = \alpha \mathrm{RTT}_i + (1-\alpha) rtt_i \tag{1}$$

where α is a constant between 0 and 1, which determines the rate of adaptation of the latency estimation, and rtt_i is a packet's actual value RTT.

The proposed FPS is compared in simulation to W-SCTP, and it achieves higher throughput. It also achieves lower reordering delay than multipath round robin SCTP (Mirani et al., 2011). However, the FPS's adaption to varying path conditions is limited to path latency (computed as a function of RTT), disregarding other important path characteristics, such as loss rates. As a result, FPS may not achieve optimal multipath transmission. A more adaptive version of FPS is presented in Mirani et al. (2011). Here, a cross-layer mechanism is proposed to enable FPS to use link-layer information to better adapt to varying wireless channels. Thus, robust multipath transmission can be accomplished.

Wang et al. (2009) present a generic transport layer model to enable bandwidth aggregation over multiple heterogeneous wireless networks. The proposed generic transport layer model consists of two functional elements: multiple transmission control protocol (MTCP) and common transport control (CTP). MTCP controls simultaneous connections over multiple radio access networks (RANs). That is, it schedules data on multiple RANs; it also establishes and manages multiple transport layer connections over the RANs. CTP is responsible for path specific connection management.

The performance of the proposed transport layer model depends mainly on a scheduler, which is implemented as part of the MTCP. The objective of the scheduler is to minimize delay difference between the selected RANs, thus reducing segment reordering at the receiver. This is achieved by dynamically allocating the traffic load to the RANs according to appropriate allocation ratios. The allocation ratios are calculated using estimated transmission rates and delays. To adapt to the varying network conditions, segment based feedback approach is used. This divides traffic into fixed length segments, which are transmitted via the RANs. On receiving a segment, the receiver sends back transmission delay information to the sender to adjust the allocation ratios. The scheduler transmits a segment by diving it into sub-segments, which can then be delivered simultaneously over multiple RANs. The sub-segments are numbered and timestamped for proper reception by the receiver. The transmission process is illustrated in Fig. 8. MTCP may not be robust against losses, as it does not consider transmission and congestion losses when computing traffic allocation ratios.

5.3. Network layer

The most notable network layer-based solution to enable efficient and adaptive packet transmission using aggregated bandwidth capacity is the earliest delivery path first (EDPF) scheduling proposed by Chebrolu and Rao, 2002, 2006). The objective of EDPF is to deliver packets over multiple paths within the shortest time possible while ensuring that the packets arrive at the receiver in correct order. EDPF's rationale is to dynamically estimate the delivery time of the next packet on each link. Then, EDPF transmits the packet through the path that delivers it the earliest. The delivery time is estimated according to Eq. (2).

$$d_i^l = \mathrm{MAX}(a_i + D_l, A_l) + \frac{L_i}{B_l} \tag{2}$$

where d_i^l, a_i, and L_i denote the estimated delivery time of packet i via path l, the arrival time of packet i at the network proxy, and the length of packet i, respectively. D_l, A_l, and B_l represent the delay from the proxy to the base station on path l, the time when

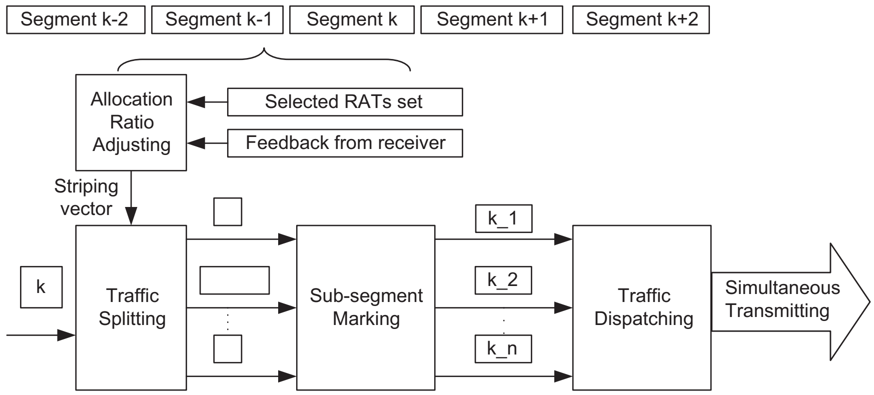

Fig. 8. Segment transmission process (Wang et al., 2009).

path l is available for transmission, and the bandwidth of path l, respectively. The first component of the equation determines the time at which transmission can resume on path l, while the second one returns the packet transmission time along the path. EDPF, therefore, schedules a packet on path p, satisfying

$$p = l : d_i^l < d_i^m, l \leq m \leq N, \tag{3}$$

where N is the number of interfaces and p is the path with the earliest delivery time. d_i^l is computed dynamically before the next scheduling round, ensuring that all the available paths can be used while preserving the order in which packets are delivered to the receiver. The work complexity of EDPF is O(n) since the packet's delivery time is computed for each of the available links to determine one with the shortest time.

EDPF outperforms simpler schedulers, such as static RR-based multipath schedulers in terms of packet reordering and end-to-end delay. However, it does not fully utilize bandwidth of the selected set of paths, as it uses only one interface at time to transmit packets, switching to the best interface any number of times during the transmission session. In addition, EDPF considers only path bandwidth and latency, leaving other important link characteristics, such as packet loss rate. Therefore, it may achieve sub-optimal performance gains in the presence of high losses.

Chebrolu and Rao (2004) present a mechanism to deliver interactive video traffic over multiple paths to enhance overall video quality. The proposed scheme includes a frame discard strategy (Min Cost Drop) to selectively discard frames, ensuring that high priority frames can be delivered to arrive within their playback times. Min Cost Drop (MC-Drop) exploits the high correlation of frame sizes across group of pictures (GOPs) to predict the sizes of future frames. Based on this, a decision on whether or not to drop a packet can be made. MC-Drop discards the current packet if, by doing so, the future high priority frame can meet its playback time. Moreover, MC-Drop drops all packets of the frames, together with their dependents, that cannot meet their deadlines. Packets of the frames that can meet their deadlines are scheduled over multiple paths according to EDPF. Even though the scheme uses application layer information to drop packets, its striping point is at the network layer. The presented selective frame discard approach achieves higher video quality in terms of peak signal-to-noise-ratio (PSNR) than most techniques that attempt to transmit all the application frames. However, the scheme experiences an increase in frame loss as the number of parallel paths with asymmetric characteristics increases. Therefore, it is important to determine the optimal number of paths to use to maintain acceptable video quality.

In Liu et al. (2007) present an improvement of EDPF, where the proposed scheduler considers transmission rates and losses to estimate packet delivery time. Packet delivery time can be determined as

$$d_i^l = D_l + d_l + \frac{L_i}{R_l(1-\alpha_l)} \tag{4}$$

where D_l is the one way wire line delay in the core network of path l, d_l is the one way wireless delay on the path, R_l is the data rate of the wireless interface l, α_l is the packet loss rate on path l, and L_i is the length of packet i. Similar to EDPF, Liu's scheduler selects a packet for transmission according to (3), where d_i^l is estimated using (4). The two schemes achieve similar throughput gain at low loss rates; but Liu et al.'s scheme is superior to EDPF during high loss rates. However, Liu's scheme only considers losses due to wireless transmission errors, ignoring losses caused by congestion. Therefore, its optimality may diminish considerably during high congestion periods.

Taleb et al. (2007) and Fernandez et al. (2009) consider time slot allocation of the aggregated bandwidth capacity and propose a scheduling mechanism called time-slotted earliest deadline path first (TS-EDPF) to exploit the aggregated capacity. In a time-slotted system, each terminal is assigned a time slot to access the wireless channel. To make accurate estimate of the delivery time for the next packet on the channel, the network proxy needs to know the start and end of the time slot that is assigned to the terminal. The delivery time can then be estimated according to Eq. (5) below.

$$d_i^l = g\left(f(\mathrm{MAX}(a_i + D_l, A_l), l) + \frac{L_i}{B_l}, l\right) \tag{5}$$

where $f(\ldots)$ is a function that returns the next valid time at which transmission can begin at the base station on path l. $g(\ldots)$ returns a valid time at which the transmission of packet i can complete. The functions are defined to ensure that packet transmission occurs within the assigned time slot. TS-EDPF achieves lower reordering delay and packet losses than EDPF in a time-slotted wireless network system. However, like EDPF, it is not optimized to handle cases where congestion and wireless path losses may be high.

Another improvement of EDPF – Packet-Pair EDPF for TCP applications (PET) – is studied in Chebrolu et al. (2005). PET estimates the delivery time of packets by sending packet pairs to the receiver, which computes inter-arrival time between the pairs and reports to the network proxy using signaling acknowledgments (SIG-ACKs). The proxy uses the inter-arrival information to estimate bandwidth and delay on the paths. Then, EDPF is used to schedule packets for transmission over the available links. The residual delay at the receiver is hidden from TCP using a buffer management policy (BMP), which holds out-of-order packets and reorder them correctly before delivering them to TCP. To detect and handle lost packets, the proposed BMP uses timer-based and comparison-based loss detection techniques. The *timer-based* loss detection associates a sequence number (N) with a timer. When the timer expires and N has not been received, it is assumed that the packet was lost. Then, buffered packets can be delivered to TCP so that duplicate ACKs can be sent to trigger fast-retransmit. *Comparison-based* approach, on the other hand, assumes that packets always arrive in correct order; i.e., when sequence numbers greater than N arrive at the receiver, it is assumed that N was lost. To avoid burstiness of TCP ACKs to the sender or packets from the reorder buffer to higher layers, a technique similar to ACK pacing (Partridge, 1998) is implemented. Even though PET, in combination with BMP, can effectively handle TCP traffic, it has similar drawbacks to EDPF because its operation assumes lossless channels, which is not practical especially in wireless networks.

Prabhavat et al. (2011) propose a load distribution model called effective delay-controlled load distribution (E-DCLD) for multipath packet transmission. The objective of the model is to minimize latency difference among the paths in order to reduce packet reordering at the receiver and to efficiently balance load across the paths. E-DCLD consists of three functional components: traffic splitter to derive allocation ratios for the paths, path selector to select an appropriate path for the packet, and load adapter to dynamically estimate end-to-end delay on each path and adjust the allocation ratios accordingly.

To efficiently compute allocation ratios, the traffic splitter uses end-to-end delay estimates of the paths. The ratios are then re-adjusted by the load adapter for every packet arrival. The main idea of E-DCLD is to decrease load on the path that has the longest delay and increase it on the path with the smallest delay. Load increase on the faster path is by the same amount of load reduced on the slower path. Even though E-DCLD can efficiently balance load across multiple links, it has similar drawbacks to EDPF. For instance, it ignores wireless path losses and congestion losses. Moreover, its work complexity increases with the number of paths since it requires every path to be monitored for traffic ratio adjustment when a new packet arrives.

Lin and Tsao (2005) present a dynamic bandwidth aggregation (DBA) scheduler, which resides in the network proxy. The objective of the scheduler is to enhance throughput by scheduling packets of the same stream over multiple links. The DBA scheduler architecture consists of two functional elements: traffic monitor and scheduler. The DBA traffic monitor observes traffic over the wireless paths and feeds the information to the DBA scheduler to select a suitable link for the next packet. The scheduler computes the packet's expected departure time on each path, and similar to EDPF, the packet is scheduled on the path that offers minimum departure time. To adapt to the varying wireless link conditions, the departure time is calculated for each new packet arrival. The work complexity incurred to schedule a packet is, therefore, linear, and increases with the number of paths considered.

Ahmad et al. (2009) propose multi-server delay-budget ordered (MDO) scheduling architecture to schedule backlogged packets of an application over multiple links to the receiver. To ensure fairness among multiple applications (sessions), MDO uses some form of SRR. The MDO architecture includes a link monitor to estimate effective one way trip time on the link between the sender and the receiver. The effective one way trip time estimation is based on past observations. MDO uses effective one way trip time values to rank links that are available for use to achieve the desired bandwidth aggregation. Then, a packet that has the smallest delay tolerance (delay budget) is allocated to a path that offers the shortest trip time. This is a desirable feature for real-time applications, which usually need to be delivered to the receiver the soonest. The MDO scheduler attempts to schedule packets of the same flow on a single link, thus increasing the probability of in-order arrival. However, this may lead to inefficient load balancing when flow sizes are different. To adapt to varying link conditions, MDO performs link ranking in every scheduling round; but, like other latency-based multipath schedulers, it fails to capture important characteristics, such as channel losses.

Tsai et al. (2008) developed Concurrent Multipath Transmission Scheme (CMTS) to distribute packets of a single application over multiple paths to enhance throughput while preserving the correct packet order. The main idea behind CMTS is to artificially equalize average delays of the available links by scheduling more packets on the faster link. To accomplish this, the average transmission interval time TI_i on the faster link is estimated. Let t_d denote end-to-end delay difference between the slow path and the fast path. The number of packets that can be scheduled on the fast link can be determined as

$$N_{\text{fast}} = \frac{t_d}{TI_i} + 1 \tag{6}$$

The total the number of packets allocated to the paths in a scheduling round is given as

$$\sum_{i=1}^{N-1} N_{\text{fast}} + 1 \tag{7}$$

where N is the total number of links. To ensure in-order delivery, the last packet from (7) should be allocated to the slowest link. However, the scheme has no mechanism to deal with the residual packet reordering at the receiver. Also, characterizing the wireless channel by latency alone and ignoring other important characteristics, such as wireless link losses, may not yield optimal gains.

Evensen et al. (2009) introduce a multilink proxy to stripe IP packets of the same flow over multiple heterogeneous links, ensuring that the packets arrive in correct order while balancing the traffic load efficiently across the links. The proposed multilink proxy implements the functional elements: packet scheduler, delay equalizer, and path monitor. The packet scheduler determines traffic allocation ratios according to weighted round robin, where the weights are calculated based on link throughputs. The delay equalizer maintains a buffer to queue packets traversing the fast link so that they can experience similar delays to packets on the slow link. The total amount of time a packet on the fast link can be held in the buffer is bounded by the difference in delay between the fast link and the slow link. The packet scheduler and the delay equalizer rely on the path monitor for periodic updates on throughput and delay so that they can adapt allocation ratios and transmission schedules to network changes.

Even though the presented multilink proxy can achieve efficient load balancing and low reordering delay, it has some drawbacks. For instance, link throughput estimation is only based on available bandwidth, missing important channel characteristics, such as loss. This can lead to sub-optimal performance, especially for applications that are sensitive to loss. Furthermore, preventing packet reordering by throttling the transfer rate of a fast link is not efficient, as it reduces link utilization, thus leading to considerable throughput degradation and almost obviating the use of multiple links.

Evensen et al. (2011) improve their work in Evensen et al. (2009) to work with middleware network elements, such as network address translators (NATs). The proposed solution uses an IP tunnel between a multi-interface client and the proxy, thus creating a multi-path overlay network. IP packets are then transmitted to the client through the tunnels. To enable the solution to work with a NAT, NAT hole punching (Zhang et al., 2009) can employed. To efficiently aggregate bandwidth over the multi-path overlay network, an adaptive packet scheduler is introduced to intelligently distribute packets across the network, ensuring that traffic is well balanced. The scheduler includes a congestion control mechanism based on CCID2

```
max_capacity = MIN_VALUE
scheduled_link = None
links = [set of links with open congestion window]

if links == Empty then
    drop packet
    return None
end if

for all links do
    if capacity_link > max_capacity then
        max_capacity = capacity_link
        scheduled_link = link
    end if
end for
return scheduled_link
```

Fig. 9. Congestion window-based packet scheduler (Evensen et al., 2011).

(Azad et al., 2009), where the proxy periodically checks the links' congestion windows and assigns an incoming packet to a link with a larger congestion window. If no link has a congestion window that is big enough to accommodate the packet, it is dropped. The scheduler algorithm is depicted in Fig. 9. The proposed bandwidth aggregation solution solely depends on the reorder buffer to correct packet reordering. Thus, a large buffer may be required to achieve efficient re-sequencing. However, as it was mentioned earlier, the implementation of a large buffer can yield long end-to-end delays.

Mao et al. (2009) present an analytical framework for optimally splitting traffic over multiple paths. In the framework, traffic splitting is formulated as a constrained optimization problem. The objective is to minimize end-to-end delay, which includes path and re-sequencing delay. The incoming traffic stream is regulated by a (ρ, σ) leaky bucket, where ρ is long-term average rate, and σ is the maximum burst size of the stream. A closed-form solution to the optimization problem is derived. The solution provides optimal traffic split ratios that minimize end-to-end delay and conform to the leaky bucket parameters. However, the framework does not provide any mechanism at the sender to dispatch traffic units over the selected paths in correct order. Instead, it relies on the re-sequencing buffer at the receiver to correct packet reordering. Thus, it may require a large buffer to efficiently control packet reordering.

Zhong et al. (2011) propose an adaptive load balancing algorithm (ALBAM) whose objective is to achieve efficient utilization of the aggregated bandwidth capacity while reducing packet reordering. To select the best path for a packet, ALBAM lists all possible paths for incoming packets; each path is associated with a quality value defined by

$$S_i = Q_i + V_i^* \tag{8}$$

where Q_i is queuing delay experienced by a packet on path i, and V_i^* is the weight parameter on the path. V_i^* is used to configure traffic allocation ratios to conform to the desired application performance. When packets arrive, they are not scheduled immediately—rather, they are temporarily buffered to create a time gap between consecutive packets; this helps to control the transmission order. When scheduling resumes, a packet from the transmission buffer is allocated to a path that minimizes the quality value as defined by Eq. (8). To adapt to varying channel conditions, ALBAM monitors the paths and updates the quality value (S_i) periodically. ALBAM reduces packet reordering and enhances throughput, even for TCP applications. However, its use of transmission buffers to control the transmission order may introduce delays that may not be tolerated by real-time applications.

Manousakis et al. (2007) present INTELiCON, an intelligent connectivity framework for simultaneous use of multiple network interfaces to deliver high QoS in unreliable and resource limited networks. INTELiCON consists of four functional modules: Packet Processing, Decision, Measurements, and Control. The Packet Processing module implements transmission strategies, particularly round robin, to manipulate packets and transmit them dynamically across multiple network interfaces. Due to the use of round robin packet distribution, INTELiCON may not adapt well to variable wireless channel characteristics, thus leading to high packet reordering and unpredictable throughput and delay performance. To solve packet reordering, making higher layers oblivious to the use of multiple interfaces, INTELiCON maintains a reorder buffer to hold packets before delivering them to the application.

The Decision module uses an optimization algorithm to decide on the optimal connectivity strategy that the Packet Processing module should use to achieve efficient multipath communication. Varying traffic and network characteristics form an important set of input parameters to the optimization algorithm, thus enabling dynamic adjustment of the connectivity strategies to ensure adaptive multilink transmission. The Measurements module collects and distributes information about traffic and network conditions to all other modules. The information can be communicated using in-band and out-band methods, which are implemented by the Control module.

5.4. Link layer

Yaver and Koudouridis (2009), Koudouridis et al. (2005) study switched MRTD based on the GLL concept. Switched MRTD allows for transmission of user's data via one RAT at a time, switching between RATs with favorable characteristics any number of times during the transmission period. Switched MRTD is, therefore, a more adaptive form of MRTD. In Koudouridis et al. (2005), switching between the available RATs is triggered by throughput measurements. In every transmission interval, user's traffic is moved to a RAT with the highest supportable throughput. In Yaver and Koudouridis (2009), traffic is switched to a RAT with the shortest RTT. In both cases, loss rates are not considered. It is shown that switched MRTD achieves better performance in delay and good put than naïve parallel MRTD. However, at medium traffic loads when the RATs show similar performance distributions, the performance of switched MRTD degrades considerably due to frequent switching and feedback information overheads. Consequently, further extensions are required to optimize switching and update frequency in order to enhance the performance of switched MRTD.

A similar approach to the GLL concept is studied by Kim et al. (2008). They introduce a cognitive convergence layer (CCL) to harmonize the different link layers' functions, thus creating a single virtual link layer interface between higher layers and the underlying link layer interfaces. To efficiently exploit the aggregated bandwidth, the CCL implements a traffic distribution policy at the sender and a reorder buffer at the receiver. The traffic distribution policy disperses traffic across the links in proportion to their available capacities. The available capacity is calculated as a function of link transmission time (LTT), which estimates the amount of channel occupancy time. The proposed multilink architecture relies only on the receiver to solve data reordering; thus, a large reorder buffer may need to be implemented.

The CCL-based multipath traffic distribution system is further studied in Kim (2010) using link delay as a metric to decide how data can be striped over tight-coupled WiMAX and WiFi networks. The traffic distributor calculates and adjusts traffic allocation ratios using link layer delay information that is periodically communicated to the sender by the receiver through a feedback mechanism. The solution can balance traffic load more efficiently than those that blindly disperse data units over the links. However, similar to Kim et al. (2008), it solely relies on the reorder buffer to correct possible data reordering at the receiver and may require a large buffer to correct the reordering. Furthermore, assessing link quality based solely on delay information may not be optimal, as there are other important link characteristics, such as wireless path loss, which can affect performance of the link.

In Kim et al. (2011), air-time cost based traffic splitting over tight-coupled WiMAX and WiFi networks is presented. Air-time cost is defined as a measure of channel occupancy time for a single packet transmission. Cumulative air-time cost (CAC) is, therefore, total channel occupancy time for packet transmission on a link, and it is determined by recording the total number of packets transmitted through link i during a time interval (T_i). CAC is normalized to available channel time and used to compute traffic allocation ratios. A flowchart in Fig. 10 summarizes the procedure to determine traffic allocation ratios. The proposed air-time cost-based model is reported to achieve more rapid adaptation to link variations than RTT-based model. However, more attributes, such as loss and delay, should be incorporated to further improve performance.

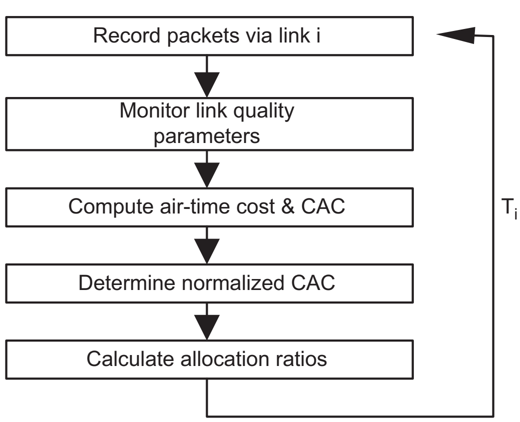

Fig. 10. Air-time-based traffic splitting (Kim et al., 2011).

6. Summary of bandwidth aggregation solutions and open research issues

6.1. Summary of bandwidth aggregation solutions

The reviewed bandwidth aggregation solutions are classified based on their ability to adapt to changing traffic and network conditions. The solutions that configure their traffic allocation ratios and distribution policies to match varying traffic and network characteristics are classified as adaptive, while those that are based on static configurations are called non-adaptive. The solutions are further classified according to the layer of the network protocol stack at which they perform traffic striping, that is, they are classified into application, transport, network and link layer solutions. Table 2 is a summary of non-adaptive bandwidth aggregation solutions. The non-adaptive bandwidth aggregation solutions can achieve load balancing to some extent. However, when traffic and network conditions vary rapidly, the load balancing ability may diminish significantly.

Also, non-adaptive approaches mostly rely on the reorder buffer implemented at the receiver to correct reordering; so, to efficiently solve data reordering, a large reorder buffer may be required. A reorder buffer that is too large can increase the packets' end-to-end delays, thus affecting performance of delay sensitive applications. Moreover, a buffer that is too large may not be practical for handheld devices with limited resources. Despite lack of adaptation to traffic and network conditions, non-adaptive solutions have low computational complexity and communication overhead, so they can be easy to implement.

A summary of adaptive bandwidth aggregation solutions is depicted in Table 3. Adaptive bandwidth aggregation solutions rectify most of the deficiencies of non-adaptive solutions. However, this may come with increased complexity. The complexity may result from the communication that is required to periodically gather information about traffic and network conditions. It can also be due to finding the best path from a list of the available paths for each packet. Most of the solutions in Table 3 use a single metric to adapt to traffic and network dynamics. Therefore, such solutions may not be efficient in a practical scenario where the network may be characterized by multiple variables. Also, very few adaptive solutions deal with reordering at both the sender and the receiver; most approaches implement reordering solution at either the sender or the receiver. Furthermore, almost all the reviewed solutions do not incorporate cross-layer design to allow communication between the layers of the network protocol stack. The load balancing problem is addressed by all the solutions. How efficient the solutions can handle load balancing depends on a number of factors. For instance, packet-level solutions can achieve higher load balancing efficiency than flow-level approaches. Feedback information reporting time scale can also affect load balancing efficiency. That is, when the information is reported to the sender in a short time scale, capacity variations that can warrant reconfiguration of allocation ratios can be captured soon enough to prevent load imbalance. Long time scale reporting, on the other hand, may miss important capacity variations and incur a drop in load balancing efficiency.

6.2. Open research questions

As presented in the previous sections, several bandwidth aggregation solutions have been developed to enable simultaneous use of multiple wireless paths to enhance performance in

Table 2
Summary of non-adaptive bandwidth aggregation solutions.

Approach	Striping point	Load balancing	Reordering solution	Adaptation metric	Cross-layer information	Computational complexity
MuniSocket (Mohamed et al., 2002)	Application layer	Yes	Receiver-based	None	No	-
MPTCP (Ford et al., 2011)	Transport layer	Yes	Receiver-based	None	No	-
WRR (Zhang et al., 2009)	Network layer	Yes	None	None	No	O(1)
SRR (Hari et al., 1999)	Network layer	Yes	Receiver-based	None	No	O(1)
Kaspar et al. (2009)	Network layer	Yes	Receiver & Sender-based	None	No	O(n)
Kim et al. (2008)	Network layer	Yes	None	None	No	O(1)
GLL (Koudouridis et al., 2005)	Link layer	Yes	No	None	No	-

Table 3
Summary of adaptive bandwidth aggregation solutions.

Approach	Striping point	Load balancing	Reordering solution	Adaptation metric	Cross-layer information	Computational complexity
Luo et al. (2003)	Application layer	Yes	No	Delay	No	O(1)
LPS (Sharma et al., 2007)	Application layer	Yes	No	Loss	No	O(1)
FPS (Sharma et al., 2007)	Application layer	Yes	No	Loss	No	O(1)
Xue et al. (2008)	Application layer	Yes	No	Bandwidth	No	O(1)
Qureshi and Guttag Horde (2005)	Application layer	Yes	No	Delay; loss	No	O(n)
Kaspar et al. (2010)	Application layer	Yes	No	Bandwidth	No	–
W-SCTP (Casetti and Gaiotto, 2004)	Transport layer	Yes	No	Delay	No	O(n)
LS-SCTP (El A et al., 2004)	Transport layer	Yes	Receiver-based	Bandwidth	No	O(1)
cmpSCTP (Liao et al., 2008)	Transport layer	Yes	Receiver-based	Bandwidth	No	O(1)
ATLB (Hasegawa et al., 2005)	Transport layer	Yes	Sender & receiver-based	Delay	No	O(n)
pTCP (Hsieh et al., 2002)	Transport layer	Yes	Receiver	Bandwidth	No	O(1)
cmpRTCP (Anand 2007)	Transport layer	Yes	Receiver-based	Bandwidth	No	O(1)
FPS (Mirani et al., 2010)	Transport layer	Yes	No	Bandwidth	No	–
FP-cross-layer (Mirani et al., 2011)	Transport layer	Yes	No	Bandwidth	Yes	–
MTCP (Wang et al., 2009)	Transport layer	Yes	Sender-based	Delay	No	–
EDPF (Chebrolu and Rao, 2002; Chebrolu and Rao, 2006)	Network layer	Yes	Sender-based	Delay	No	O(n)
(Liu et al., 2007)	Network layer	Yes	Sender-based	Delay	No	O(n)
TS-EDPF (Taleb et al., 2007; Fernandez et al., 2009)	Network layer	Yes	Sender-based	Delay	No	O(n)
PT (Chebrolu et al., 2005)	Network layer	Yes	Sender & receiver-based	Delay	No	O(n)
E-DCLC (Prabhavat et al., 2011)	Network layer	Yes	Sender-based	Delay	No	O(n)
DBA (Lin and Tsao, 2005)	Network layer	Yes	Sender-based	Delay	No	O(n)
MDO (Ahmad et al., 2009)	Network layer	Yes	Sender-based	Delay	No	O(n)
CMTS (Tsai et al., 2008)	Network layer	Yes	Sender-based	Delay	No	–
Evensen et al. (2009)	Network layer	Yes	Sender-based	Bandwidth	No	O(1)
Mao et al. (2009)	Network layer	Yes	Receiver-based	Delay	No	O(n)
ALBAM (Zhong et al., 2011)	Network layer	Yes	Sender-based	Delay	No	O(n)
INTELiCON (Manousakis et al., 2007)	Network layer	Yes	Receiver-based	Delay; loss	No	O(1)
Yaver and Koudouridis (2009)	Link layer	Yes	Receiver-based	Delay	No	O(n)
Koudouridis et al. (2005)	Link layer	Yes	Receiver-based	Bandwidth	No	O(n)
Kim et al. (2008)	Link layer	Yes	Receiver-based	Delay	No	O(1)
Kim (2010)	Link layer	Yes	Receiver-based	Delay	No	O(1)
Kim et al. (2011)	Link layer	Yes	Receiver-based	Air-time balance	No	O(1)

terms of throughput, load balancing, reduced latency, etc. However, there are still some issues that need further research. The issues can be identified as follows:

6.2.1. Multidimensional adaptation metrics

Most of the existing bandwidth aggregation solutions consider a single metric to adapt to varying traffic and network conditions. However, the wireless channel is characterized by multiple attributes (e.g., delay, loss, bandwidth, etc). Consequently, bandwidth aggregation solutions that are based on a single attribute may not achieve optimal performance gain. For instance, it is possible for a solution that adapts based path latency to schedule data on a path that is experiencing high loss rates, thus resulting in performance loss due to losses on the path. Therefore, solutions that incorporate multiple adaptation metrics need to be investigated.

6.2.2. Switching cost and optimization

Some of the reviewed bandwidth aggregation solutions are based on switched MRTD, where traffic is switched between the available paths any number of times based on variations in path characteristics, such as path latency. However, frequent switching can lead to performance degradation (Yaver and Koudouridis, 2009). Therefore, bandwidth aggregation techniques that can quantify the switching cost and optimally switch traffic between the available paths are required. For instance, performance may be enhanced by limiting path switching to channel variations large enough to assure considerable performance gain instead of switching even for small changes that may not yield any significant gains.

6.2.3. Comprehensive analysis of packet reordering

Reordering can have adverse impact on performance of real-time and TCP applications. For real-time applications, reordered protocol data units may be considered lost; for TCP applications, unnecessary retransmissions may occur. Thus, network resources utilization efficiency may degrade significantly. It is, therefore, important to accurately and comprehensively quantify reordering so that its effects can be efficiently captured and dealt with. However, very few bandwidth aggregation solutions pay attention to comprehensive analysis of reordering; as a result, they may miss important optimizations that could come out of the analysis. Also, the few that analyze the effects of reordering use different metrics without any consensus, thus making it difficult to compare their performance. More work is, therefore, required to compressively capture reordering using standardized metrics such as RD and RBD.

6.2.4. Intelligent parallel MRTD

Most bandwidth aggregation solutions are based on switched MRTD. Switched MRTD is non-work conserving and may allow aggregated capacity resources to go unused, thus resulting in low utilization. Work-conserving property can be achieved using parallel MRTD. Whenever there is data to transmit (especially a backlog), parallel MRTD ensures that all paths are used, thus improving utilization. However, most of the current bandwidth aggregation solutions employing parallel MRTD blindly disperse data across the paths, risking high reordering, which can obviate the use of multipath communication. Therefore, parallel MRTD solutions that can intelligently stripe traffic over the paths while trying to preserve the correct data sequence are required.

6.2.5. Cross-layer design

Most bandwidth aggregation schemes use only information gathered at the layer at which they perform traffic striping. Even though this may simplify the solution, important layer specific characteristics cannot be captured, thus leading to sub-optimal performance. To solve this, bandwidth aggregation solutions that can enable communication between the different layers of the network protocol stack need to be investigated.

6.2.6. Intelligent interface selection

Activating a network interface on a terminal to access a service is associated with a cost, which may be a function of power consumption, money paid for the service, user and network preferences, etc. The cost may increase when multiple interfaces are used. Solutions that incorporate intelligent interface selection to achieve efficient bandwidth aggregation at minimal cost are desirable. However, most of the existing bandwidth aggregation solutions do not pay attention to this.

7. Conclusion

Bandwidth aggregation solutions have been proposed to enable simultaneous use of a multimode terminal's multiple network interfaces to improve performance in terms of throughput, load balancing and reduced latency. Low-cost capacity increase can also be realized by simply aggregating low-bandwidth links to create a large logical link that can serve applications that require high bandwidth. This article has presented a comprehensive survey of the bandwidth aggregation solutions developed to exploit bandwidth capacity from multiple wireless links simultaneously. The solutions have been categorized based on their adaptability to varying traffic and network conditions (e.g., adaptive or non-adaptive) and a layer of the network protocol stack at which they perform traffic splitting (e.g., application, transport, network and link layer). Major performance characteristics of the bandwidth aggregation solutions have been analyzed. Several open research issues have also been discussed.

References

Ahmad SZ, Qadir MA, Akbar MS QoS optimization of in-elastic flows stripped over multiple asymmetric channels in mobile networks. In ICUMT, 2009.

Al AAbd El, Saadawi T, Lee M. Improving throughput and reliability in mobile wireless networks via transport layer bandwidth aggregation. Computer Network (Special Issue on Military Communications Systems and Technologies) 2009;46(5): 635–49.

Allcock W Gridftp: protocol extensions to ftp for the grid. In Global Grid ForumGFD-R-P.020, 2003.

Allman M, Kruse H, Ostermann S An application-level solution to TCP's satellite inefficiencies'In Workshop proceedings on satellite-based information services, 1996.

Anand J, D. Sarkar, cmpRTCP: concurrent multi-path real-time TCP. In Proceedings of IEEE GLOBECOM 2007.

Azad MA, Mahmood R, Mehmood T. A comparative analysis of DCCP variants (CCID2, CCID3), TCP AND UDP for MPEG4 video applications. In ICICT 2009.

Barre S, Bonaventure O, Raiciu C, Handley M, Experimenting with multipath TCP. In Proceedings of ACM SIGCOMM 2010.

Bazzi A, Pasolini G, Andrisano O. Multiradio resource management: parallel transmission for higher throughput? EURASIP Journal on Advances in Signal Processing 2008;2008.

Bennett JCR, Partridge C, Shectman N. Packet reordering is not pathological network behavior. IEEE/ACM Transactions on Networking 1999;7(6):789–98.

Casetti C, Gaiotto W Westwood sctp: load balancing over multipaths using bandwidth-aware source scheduling. In Proceedings of IEEE VTC, 2004.

Chebrolu K, Rao R Communication using multiple wireless interfaces. In Proceedings of IEEE WCNC 2002, pp. 327- 331.

Chebrolu K, Rao R, Selective frame discard for interactive video. In Proceedings of IEEE ICC, 2004.

Chebrolu K, Rao R. Bandwidth aggregation for real-time applications in heterogeneous networks. IEEE Transactions on Mobile Computing 2006;5:4.

Chebrolu K, Raman B, Rao R. A network layer approach to enable TCP over multiple interfaces. Journal on Wireless Networks (WINET) 2005.

Cheng H, Jin Y, Gao Y, Di Yu Y, Hu W, Ansari N Per-flow re-sequencing in load-balanced switches by using dynamic mailbox sharing. In Proceedings of IEEE ICC, 2008, 5680–5684.

Chowdhury MZ, Jang Yeong Min, Sub C Choi Ji S, Jeon H, Jee J, Park C. Interface selection for power management in UMTS/WLAN overlaying network. In Proceedings of ICACT, 2009, pp. 795-799.

Ekmekci S, Sikora T. Multistate vs. single-state video coding over error-prone channels. Record of the thirty-seventh Asilomar: signals. Systems and Computers 2003;2:1544–7.

El A A, Saadawi T, Lee M. LS-SCTP: a bandwidth aggregation technique for stream control transmission protocol. Computer Communications 2004;27:1012–24.

Evensen K, Kaspar D, Engelstad P, Hansen AF, Griwodz C, Halvorsen P, Network-Layer A Proxy for bandwidth aggregation and reduction of IP packet reordering. In Proceedings of IEEE LCN, 2009.

Evensen K, Kaspar D, Engelstad P, Hansen AF, Griwodz C, Halvorsen P Using multiple links to increase the performance of bandwidth-intensive UDP-based applications.In Proceedings of IEEE ISCC, 2011.

Fernandez JC, Taleb T, Guizani M, Kato N. Bandwidth aggregation-aware dynamic QoS negotiation for real-time video streaming in next-generation wireless networks. IEEE Transactions on Multimedia 2009;116:1082–93.

Fiore M, Casetti C An adaptive transport protocol for balanced multihoming of real-time traffic. In Proceedings of IEEE GLOBECOM, 2005.

Ford A, Raiciu C, Handley M, Barre S, Iyengar J Architectural guidelines for multipath TCP development. IETF RFC 6182, March 2011.

Hari A, Varghese G, Parulkar G. An architecture for packet-striping protocols. ACM Transations on Computer Systems 1999;174.

Hasegawa Y, Yamaguchi I, Hama T, Shimonishi H, Murase T Improved data distribution for multipath TCP communication. In Proceedings of IEEE GLOBECOM, 2005.

Hossain E. Heterogeneous wireless access networks, (edited volume). Springer; 2008 ISBN: 978-0-387-09776-3.

Hsieh H-Y, Sivakumar R, Transport layer, A. approach for achieving aggregate bandwidths on multihomed mobile hosts. In Proceedings of ACM MOBICOM 2002.

Jayasumana A, Piratla N, Banka T, Bare A, Whitner R. Improved packet reordering metrics. Request for Comment 2008;52(36).

Kaspar D, Evensen K, Hansen AF, Engelstad P, Halvorsen P, An C Analysis of the heterogeneity and IP packet reordering over multiple wireless networks In IEEE Symposium on Computer and Communications 2009, pp. 637–642.

Kaspar D, Evensen K, Engelstad P, Hansen AF, Halvorsen P, Griwodz C Enhancing video-on-demand playout over multiple heterogeneous access networks. In Proceedings of IEEE CCNC, 2010.

Kaspar D, Evensen K, Engelstad P, Hansen AF Using HTTP Pipelining to improve progressive download over multiple heterogeneous interfaces. In Proceedings of IEEE ICC, 2010, 1–5.

Kim J. Feedback-based traffic splitting for wireless terminals with multi-radio devices. IEEE Transactions on Consumer Electronics 2010;56:2.

Kim J-O, Ueda T, Obana S. MAC-level measurement based traffic distribution over IEEE 802.11 multi-radio networks. IEEE Transactions on Consumer Electronics 2008;54(3):1185–91.

Kim J-O, Davis P, Ueda T, Obana S. Splitting downlink multimedia traffic over WiMAX and WiFi heterogeneous links based on airtime-balance. Wireless Communications and Mobile Computing 2011.

Kim PS, Yoon Y, Kim H. A packet partition scheduling mechanism for bandwidth aggregation over multiple paths. Journal of Convergence Information Technology 2008;3(4):325–41.

Koudouridis GP, Karimi HR, Dimou K Switched multi-radio transmission diversity in future access networks. In Proceedings of the IEEE Vehicular Conference, 2005.

Koudouridis GP, Agüero R, Alexandri E, Choque J, Dimou K, Karimi HR, Lederer H, Sachs J, Sigle R Generic link layer functionality for multiradio access networks. In Proceedings of mobile summit, 2005.

LIU G, ZHOU X, ZHU G. A scheduling algorithm for maximum throughput based on the link condition in heterogeneous network. Journal Communication and Computer 2007.

Liao J, Wang J, Zhu X. cmpSCTP: an extension of SCTP to support concurrent multi-path transfer. In Proceedings of IEEE ICC, 2008.

Lin YH, Tsao S Dynamic bandwidth aggregation for a mobile device with multiple interfaces. In Proceedings of ISCOM 2005.

Lopez-Benitez M, Gozalvez J. Common radio resource management for multimedia heterogeneous wireless networks. IEEE Transactions on Mobile Computing 2011;10(9):1201–13.

Luo J, Mukerjee R, Dillinger M, Mohyeldin E, Schulz E. Investigation of radio resource scheduling in WLANs coupled with 3G cellular network. IEEE Communications Magazine 2003;416:108–15.

Mahkoum H, Sarikaya B, Hafid A A framework for power management of handheld devices with multiple radios. In Proceedings of IEEE WCNC 2009, pp. 1–6.

Manousakis K, Gopalakrishnan P, Famolari D, Van Den Berg E. iCON, I.N.T.E.: Intelligent connectivity framework for the simultaneous use of multiple interfaces. In Proceedings of ICC, 2007.

Mao S, Panwar SS, Hou YT. On minimizing end-to-end delay with optimal traffic partitioning. IEEE Transactions on Vehicular Technology 2009;55(2):681–90.

Map G, Aiash M, Guardia HC, Crowcroft J. Exploring multi-homing issues in heterogeneous environments. In Proceedings of IEEE WAINA 2011.

Mirani FH, Boukhatem N, Minh, AT. Data-scheduling, A. mechanism for multi-homed mobile terminals with disparate link latencies. In Proceedings of IEEE VTC, 2010.

Mirani FH, Kherraz M, Boukhatem N Forward prediction scheduling: implementation and performance evaluation. In Proceedings of ICT, 2011.

Mirani FH, Xiaofei, Z, Boukhatem, N, Thi-Mai-Trang, N, Cross-layer, FPS: SCTP, A-based cross-layer data scheduling approach. In Proceeding of IEEE CCNC 2011.

Miu KA Improving packet delivery efficiency using multi-radio diversity in wireless LANs, PhD Thesis, MIT, 2006.

Miu KA, Apostolopoulos J, Tan W, Trott MLow-latency wireless video over 802.11 networks using path diversity. In Proceeding of IEEE ICM, 2003.

Mohamed N, Al-Jaroodi J, Jiang, H, Swanson D, A user-level socket layer over multiple physical network interfaces. In Proceedings of the international conference on parallel and distributed computing and systems, 2002, pp. 810–815.

Nguyen, SC, Zhang X, Trang Nguyen Thi Mai, Pujolle G, Evaluation of throughput optimization and load sharing of multipath TCP in heterogeneous networks. In Proceedings of wireless and optical communications networks, 2011, pp. 1-5.

Orozco-Barbosa L, Taeseop H. On the use of frame-based slice size for the robust transmission of MPEG video over ATM networks. IEEE Transactions on Broadcasting 2000;46(2):134–43.

Partridge C ACK spacing for high delay-bandwidth paths with insufficient buffering. IETF Internet Draft, Sept. 1998, draft-rfced-info-partridge-01.txt.

Prabhavat S, Nishiyama H, Ansari N, Kato N. Effective delay-controlled load distribution over multipath networks. IEEE Transactions on Parallel and Distributed Systems 2011;22(10):1730–41.

Przybylski M, Belter B, Binczewski A. Shall we worry about packet reordering. Computational Methods in Science and Technology 2005;11(2):141–6.

Qureshi, A, Guttag Horde J, Separating network striping policy from mechanism. In Proceedings of ACM MobiSys, 2005.

Sachs J, A generic link layer for future generation wireless networking. In Proceedings of IEEE ICC 2003, pp. 834– 838.

Sharma, P, Lee S-J, Brassil, J, Shin KG, Distributed channel monitoring for wireless bandwidth aggregation. In Proceeding of the IFIP-TC6 Networking, 2004, pp. 345–356.

Sharma P, Lee S, Brassil J, Shin KG. Aggregating bandwidth for multihomed mobile collaborative communities. IEEE Transactions on Mobile Computing 2007;6(3):280–96.

Sivakumar, H, Bailey, S, Grossman, S, PSockets: the case for application-level network striping for data intensive applications using high speed wide area networks. In Proceedings of ACM/IEEE SC, 2000.

Sklower, K Lloyd G, McGregor D, Coadetti T, The PPP Multilink Protocol. RFC 1990, 1996.

Stewart R, Ramalho M, Xie Q, Tuexen M, Conrad P, Stream control transmission protocol partial reliability extension. Request For Comment 3758, May 2004.

Taleb T, Fernandez, JC, Hashimoto K, Nemoto Y, Kato N, A bandwidth aggregation-aware qoS negotiation mechanism for next-generation wireless networks. In Proceedings of IEEE GLOBECOM, 2007.

Tsai M, Naveen K, Zeadally CS, Shieh C, A concurrent multi-path transmission control scheme to reduce packet reordering latency at the receiver. In Advanced Technologies for Communications, 2008.

Veitch, D, Augustin B, Teixeira R, Friedman T, Failure control in multipath route tracing. In: Proceedings of IEEE INFOCOM 2009, pp. 1395–1403.

Venkatasubramanian S, Gopalan NP. QOS based robust multipath routing protocol for mobile ad hoc networks. IACSIT International Journal of Engineering and Technology 2009;1:5.

Wang XF, Fan D, Xue Y, Le VA Segment-based, adaptive joint session scheduling mechanism in heterogeneous wireless networks. In Proceedings of IEEE VTC, 2009, pp. 1–5.

Xue Y, Lin Y, Feng Z, Cai H, Chi C, Autonomic joint session scheduling strategies for heterogeneous wireless networks. In Proceedings of IEEE WCNC, 2008, pp. 2045–2050.

Yaver, A, Koudouridis, GP. Performance evaluation of multi-radio transmission diversity: QoS Support for Delay Sensitive Services. In Proceedings of IEEE VTC, 2009, pp. 1–5.

Zhang Y, Wang C, Gao Y, Weighted size-aware packet Distribution for multipath live streaming. In Proceedings of IEEE ICC 2009, pp. 1-5.

Zhang Z, Wen X, Zheng W, Traversal ANAT Mechanism for peer-to-peer networks. In Intelligent Ubiquitous Computing and Education, 2009.

Zhong F, Yeo CK, Lee BS, Adaptive load balancing algorithm for multi-homing mobile nodes in local domain. In Proceedings of IEEE CCNC 2011.