2006 2nd Conference on Next Generation Internet Design and Engineering

NGI 2006

3–5 April 2006 • Valencia, Spain

APOHN: Subnetwork Layering to Improve TCP Performance over Heterogeneous Paths

Dzmitry Kliazovich and Fabrizio Granelli
University of Trento
Via Sommarive 14, I-38050 Trento, Italy
E-mail: [klezovic,granelli]@dit.unitn.it

Giovanni Pau and Mario Gerla
UCLA
Computer Science Department, CA 90095, USA
E-mail: [gpau,gerla]@cs.ucla.edu

Abstract— **This paper proposes a novel architecture, named APOHN, designed for data communications over heterogeneous networks. APOHN enables easy implementation of various existing and upcoming performance optimization solutions with the main objective to keep the standardized TCP/IP reference model untouched. APOHN architecture extends ISO/OSI protocol stack model with an additional layer designed for subnetwork communications and optimized with specific physical subnetwork characteristics in mind. TCP/IP flow speedup, subnetwork flow multiplexing and optimized subnetwork communications result in great performance improvements in heterogeneous networks. Moreover, the support of IPsec enables secure communications. APOHN performance is evaluated through simulations using a combined satellite and WLAN network scenario - commonly used in disaster recovery as well as in a variety of military applications. APOHN constitutes an architectural solution competitive with leading architectures such as Performance Enhancement Proxies (PEP) as well as Delay-Tolerant Networking (DTN).**[1,2]

Keywords- Heterogeneous networks, Performance Enhancement Proxy (PEP), Delay-Tolerant Network (DTN), IP-security (IPsec)

I. INTRODUCTION

The standardization of TCP/IP protocol suite used nowadays in Internet dates back to early 80s. TCP/IP design was developed on the basis of the characteristics of networks and technological solutions available at the moment: most networks constituting the Internet were formed of computer equipment or other terminals interconnected using a cable or fiber. Basically, such networks were characterized by a strong hierarchical structure, large processing power of network terminals, static routing, stable connectivity, small propagation delays, and low error-rates.

Nowadays, the environment where TCP/IP operates greatly differs from the one it was designed for. Wireless communications clearly represent a fast-growing sector in the framework of data networks [1]. Mainly, wireless technologies provide mobile access to networks and services – omitting the requirement for a cable (and fixed) infrastructure, thus enabling fast and cost-effective network organization, deployment and maintenance. Moreover, the mobility provided by wireless networks represents a revolutionary feature for applications where mobility itself becomes the central requirement. Examples of such purely wireless scenarios range from simple internet access on anytime-, anywhere-basis to systems for disaster recovery and military applications.

Wireless domain is characterized by limited bandwidth, high latency and propagation delay, signal fading and shadowing, high error rates, terminal mobility and the necessity of handling transparent handoffs and handovers.

Several studies underlined that the TCP/IP protocol suite performs poorly in the wireless network domain [2, 3]. For that reason, the research community is actively trying to find a way to enable TCP/IP to reach a reasonable performance level in non-friendly networks for more than a decade [4].

Moreover, wireless networks are not the only environment where TCP/IP protocols perform poorly. In general, its performance drastically degrades in networks with high bandwidth-delay product [5], which include satellite [6] and space communication domains [7].

Basically, existing improvements follow one of two general principles:

Transparent Adaptation. The effort to adapt existing TCP/IP protocols to the non-friendly heterogeneous environment is mostly concentrated on the introduction of techniques designed to hide or mitigate undesirable characteristics of a section of the network. For example, in order to compensate high error rate, most of wireless networks implement ARQ or FEC at the link layer.

The principle of adaptation has obvious limitations, since not all the undesirable characteristics can be compensated transparently with respect to the TCP/IP stack.

TCP Modification. An alternative to adaptation approach is the modification of TCP/IP protocol suite - conquering the roots of the problem - in order to achieve the desired behavior. The examples of such modifications are TCP-DOOR [23] enabling out-of-order detection and corresponding response or TCP Westwood [24] which modifies congestion control based on the end-to-end capacity estimate.

The drawbacks of such approach are in the difficulty to consider the variety of heterogeneous networks while maintaining end-to-end semantics of data communications as well as to provide means for a wide deployment of such

[1] This work was done while Dzmitry Kliazovich was visiting researcher at the Computer Science Department of the University of California at Los Angeles.
[2] The work was partially supported by the Italian Ministry for Education, University and Research (MIUR) in the framework of the "WOMEN" project.

solutions. Indeed, each modification requires a huge effort from the standardization community as well as from industry in order to deploy it and deliver the resulting advantage to the end-user.

In this paper, we design a novel protocol stack architecture targeted at TCP performance improvement over heterogeneous paths with the main objective to keep the standardized TCP/IP reference model untouched.

In order to develop the core of the proposed architecture, we analyze Performance Enhancement Proxy (PEP) and Delay-Tolerant Network (DTN) architectures (Section II) as leading optimization solutions available in the literature, underlining their limitations and potential enhancements.

Optimization techniques proposed in the paper are based on the physical characteristics of every particular environment the communications are performed on and implemented at the newly introduced subnetwork layer (Section III). TCP/IP flow speedup as well as flow multiplexing at the subnetwork layer ensures high performance, while the support of IPsec enables secure communication on the end-to-end as well as on the subnetwork basis.

The performance of APOHN architecture analyzed in Section IV shows a high level of improvement compared with standard TCP/IP protocol suite as well as with split-connection PEP approach.

II. Background and Related Works

The literature on TCP/IP optimization is huge. For that reason, we only review solutions which have conceptual similarities or related to the basics for the proposed architecture.

A promising and most recent group of solutions designed to improve performance of the Internet protocols is Performance Enhancement Proxies (PEP) [8]. PEPs are designed to operate over individual links or subnetworks where, due to their characteristics, the performance of TCP/IP protocols is drastically decreased.

There are different types of PEPs [8]. However, the most commonly used PEP implements a connection splitting approach at the transport layer: an end-to-end connection is split into two or more separate connections, some of which employ a protocol optimized for a specific link or subnetwork.

Typical scenario for PEP implementation is a satellite link along the data path or, more rarely, a last mile wireless access network. Satellite channels are characterized by stable dedicated available bandwidth, low error rate and very large propagation delays. Large propagation delays are the main cause of performance degradation of Transmission Control Protocol (TCP), which performance is inversely proportional to the round trip propagation delay [16]. The PEP agent is installed at the edge router bridging terrestrial and satellite networks. As a result, the end-to-end communication flow is split into two sections: a low-delay terrestrial link using standard TCP and a high-delay satellite link using a protocol specifically designed for satellite networks.

The examples of commercially available PEPs are SkyX product line from Mentat or XipLink gateways.

The connection splitting approach has the relevant drawback of breaking end-to-end semantics of a data connection: the reliability of transmission is achieved on a hop-by-hop basis, and the sender node is not able to infer data delivery information from its actual destination but only from its nearer PEP.

A second important drawback to mention is PEP inability to handle IP-security - due to the requirement to access transport layer headers on the node running the PEP agent.

A similar optimization problem is targeted by the networking architecture called *Delay-Tolerant Networking* (DTN) [9]. The main idea behind DTN is to build an overlay network where communications are performed in posting manner by exchanging large portions of data called *bundles*. Bundles are sent between DTN nodes which protocol stack is extended with *bundling* layer (located on the top of the transport layer). The purpose of the bundling layer is to support DTN naming and addressing, and to provide routing, reliable and secure communications.

In contrast to PEP, DTN modifies the protocol stacks, not only at bridge routers, but also at sender and receiver nodes, through the introduction of an additional layer. Moreover, it increases computational processing requirement for DTN router implementing routing functionality on top of the protocol stack.

The novel network architecture proposed in this paper is designed to operate in the environments addressed by both PEP and DTN. It intends to mitigate link-related degradations by combining architectural and performance advantages of PEP and DTN and reducing the number of drawbacks.

III. Architecture for Performance Optimization in Heterogenious Networking

A. Architecture Description

The proposed Architecture for Performance Optimization of data communications in Heterogeneous Networking (APOHN) is presented in Fig. 1.

APOHN is designed for communications performed over several subnetworks, each characterized by different physical characteristics (propagation delay, available bandwidth, etc.). In this scenario, a finer optimization can be achieved within a single subnetwork rather than over the end-to-end path, where physical characteristics can vary a lot.

The requirement to preserve TCP/IP suite implemented in operating system motivates the introduction of an additional protocol layer, called *Subnetwork layer*, designed to support communications within a single subnetwork.

Subnetwork protocol (SBP) operates on top of the link layer and supports such functions like flow control and subnetwork routing.

Physical, link and subnetwork layers can be fully implemented in the network interface hardware card or partially in the software driver associated with the card. As a result, the resources needed for SBP operation are consumed only in case the network terminal operates in the particular type of network and optimization is desired.

Network layer (IP) packets constitute input data for the Subnetwork layer. The control of data flow between operating system and interface card is performed by a specially-designed protocol booster [10].

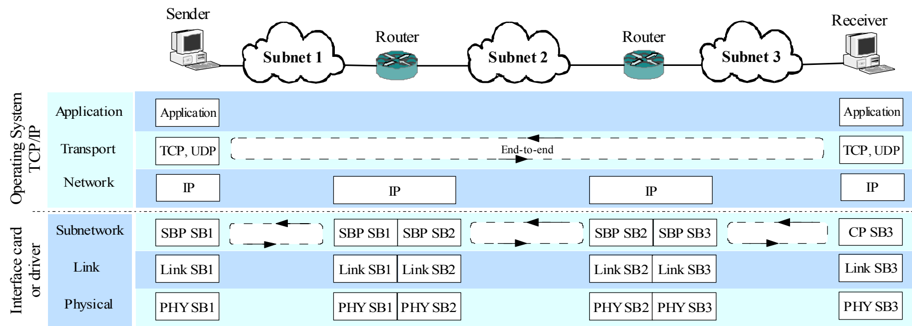

Fig. 1. Architecture for Performance Optimization in Heterogeneous Networking (APOHN).

B. *Protocol Booster*

The main task of the proposed protocol booster is to provide an interface between layers of the protocol stack (implemented inside the operating system) and network-dependant layers. Being implemented at the sender node, the protocol booster is able to access transport and network layer headers of all packets generated by the system.

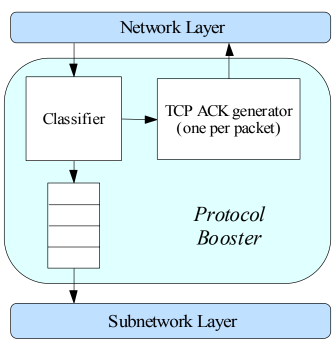

Fig. 2. Basic structure of the protocol booster.

Fig. 2 presents the basic structure of the protocol booster agent, which basically receives network layer packets, stores them in buffer, and then forwards them to the subnetwork layer for the transmission.

In case a reliable flow control protocol such as TCP operates at the transport layer, the booster agent is required to support a functionality which allows to deliver to the receiver the desired amount of data (packets).

TCP protocol implements flow control mechanism, which limits the amount of outstanding unacknowledged data using the congestion and receiver's advertised window parameters. For that reason, the protocol booster presented in Fig. 2, equipped with a TCP ACK generation module, generates an acknowledgement back to the transport layer for every received packet.

In order to control the amount of incoming data from TCP, the receiver advertised window (*rwnd*) field [17] of generated TCP ACKs is set to one packet in case the protocol booster's buffer is not full. Otherwise, the booster agent sets *rwnd* equal to zero, thus freezing TCP, which can be resumed by sending a single duplicate ACK for the last transmitted data packet with a positive value for the advertised window[3].

In summary, the protocol booster agent completely disables flow control mechanism performed by TCP without introducing any direct modifications at the transport layer. As a result, for the subnetwork layer, TCP becomes a controlled source of packets.

C. *End-to-end Internet Communications*

End-to-end operation is the driving principle in the design of Internet protocols from the early beginning. Data delivery, flow control, and reliability of TCP protocol used by overwhelming majority of applications [11, 12] are based on the end-to-end paradigm, resulting in simplicity and flexibility of communication protocols design.

Connection splitting optimization approach, commonly used in PEPs, violates end-to-end TCP semantics by generating acknowledgements from the PEP agent located on the intermediate node on the path between sender and receiver. As a result, the TCP sender can not infer successful data delivery to its destination.

The proposed APOHN architecture shifts the point controlling end-to-end reliability from the transport layer to the booster agent located below the transport layer in the protocol stack of the sender node.

The booster agent is designed to keep a TCP packet in its buffer until the corresponding TCP ACK generated by the transport layer of the destination node is successfully received.

As a result, APOHN architecture violates end-to-end TCP semantics in case a transport layer is considered as an end point. However, in contrast to connection splitting PEP, it supports end-to-end semantics in case a node appears as an end entity.

D. *Subnetwork Layer Multiplexing and Flow Control*

The overwhelming majority of data communications in Internet is multi-flow based, i.e. an internet terminal initiates several separate transport layer connections, one for every transmitted object (html text, picture, binary file, etc.). Multiple applications running on network terminals communicate among each other by using different connections. However, within a subnetwork, most of the traffic produced by the terminal is routed to one or several nodes (like an edge router).

---

[3] Single duplicate ACK does not trigger a retransmission or window reduction since at least three duplicate ACKs are needed for that.

In order to benefit from the scenario described above, the subnetwork layer is equipped with traffic flow multiplexing feature: a single data flow at the subnetwork layer can carry packets generated by different flows at the transport layer.

The network terminal initiates and maintains a connection opened to every station in the subnetwork it communicates with. After that, all packets going to the same destination are transmitted using a single subnetwork layer connection (see Fig. 3).

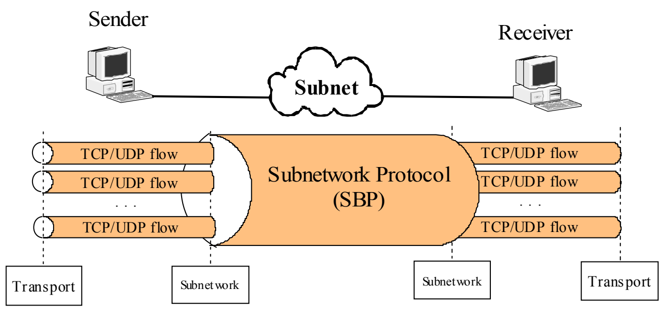

Fig. 3. Flow multiplexing on the subnetwork layer.

The main benefit achieved from flow multiplexing is in speeding up TCP flows as well as per-flow capacity allocation.

Depending on the physical nature of the channel, the flow control of SBP can be rate-based or window-based. Rate-based flow control is typically used in case of operation over a channel with dedicated capacity (like satellite channel), while window-based congestion control is useful where multiple network terminals contend for a shared medium. Nevertheless, in case the connection is maintained open for a large period of time, the subnetwork layer has its capacity information (current rate, window size) available at every moment of time.

Based on the knowledge of outgoing capacity, the subnetwork layer then multiplexes TCP/UDP flows providing each flow with a fair portion of the available bandwidth. Further differentiation is achieved by considering the class of Quality of Service (QoS) assigned to a particular data flow.

Flow multiplexing results in speed up of TCP flows. The booster agent (see Section III B) can obtain the desired amount of data from the transport layer flow by disabling TCP slow start and congestion avoidance phases. For that reason, TCP flow can always grab full portion of bandwidth provided by the subnetwork layer.

An example presented in Fig. 4 illustrates the difference between sending rates achieved by Additive Increase Multiplicative Decrease (AIMD) strategy of standalone TCP flow and TCP flow multiplexed at the subnetwork layer (the proposed approach).

A standalone TCP flow continuously probes the capacity available for the connection, first exponentially (during slow start phase) and then linearly (during congestion avoidance phase). Upon congestion loss detection, it reduces its window to a half of its size – resulting in underutilization of the available bandwidth (12 packets in the example).

TCP flow multiplexed at the subnetwork layer always produces the correct amount of data to fulfill available resources for the flow bandwidth.

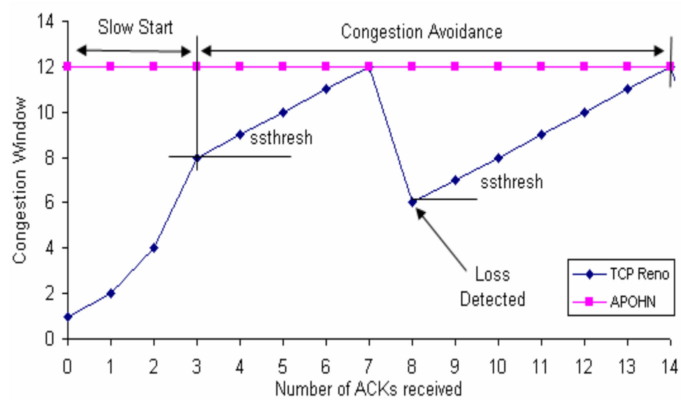

Fig. 4. Sending rates of TCP Reno and subnetwork-multiplexed flow.

E. Secure Communications and IPSec Support

IPsec is a standard mechanism providing end-to-end security in the Internet [13]. It implements end-to-end encryption of the outgoing IP datagrams. IPsec protocol considers the payload of IP datagram (TCP header + data payload) for encryption. As a result, PEP-like approaches can not provide performance improvement for IPsec packets due to the requirement for the PEP agent located on the intermediate gateway to have an access to the TCP header.

In order to deal with the problem of IPsec datagrams, Zhang at el. proposed multilayer IP security (ML-IPsec) [14] as a modification of the IPsec standard. The main idea is to divide the communication path into a number of zones and to provide encryption on a per-zone basis rather than end-to-end. In more details, ML-IPsec encrypts data payload between the sender and receiver nodes (end-to-end), while TCP header is encrypted on a per-hop basis: (i) between sender and PEP agent, and (ii) between PEP agent and receiver. The authors of [15] proposed several ML-IPsec improvements, including allowing protocol headers alterations in size and change accordingly for different encryption zones.

APOHN architecture requires an access to transport and network headers only at the sender and the receiver nodes; in case of end-to-end encryption using IPsec, the keys required to decrypt an IP datagram are available. Moreover, different security schemes which operate at subnetwork layer can be implemented on top of IPsec.

Fig. 5 overviews a security framework employed by APOHN network architecture. TCP header and data are encrypted using the IPsec protocol at sender node and decrypted at the receiver in agreement with the RFC 2401 [13].

Additional level of security, which can be optionally inserted at the subnetwork layer, encrypts an SBP datagram (IP header, IPsec header and IP data) on the links between sender and edge router. As the result, subnetwork security is directed to hide routing information available in IP header – disclosing it only to the edge router as a trusted point but not to other stations in the subnetwork.

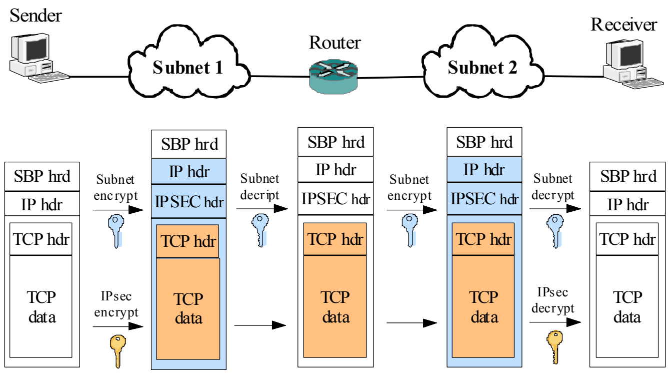

Fig. 5. IPsec and subnetwork security in APOHN architecture.

IV. Performance Evaluation on a Case Study

A. Simulation Scenario

The evaluation of APOHN architecture is performed using the ns-2 network simulator [18]. The simulated network consists of satellite and wireless subnetworks. The sender located in the satellite part of the network communicates with the receiver which is placed into the wireless LAN network operating in infrastructure mode. A single IP router equipped with satellite and wireless interfaces bridges communications between the subnetworks.

Fig. 6 illustrates an implementation of APOHN architecture in the simulation scenario. Transport and network layers are fully consistent with the standard TCP/IP reference model. TCP NewReno with selective acknowledgements (SACK) is chosen for simulation experiments as the most widely used transport layer protocol nowadays.

Satellite Transport Protocol (STP) [19] and LLE-TCP [20, 21] provide the delivery of incoming IP packets at the subnetwork layer. STP protocol implementing rate-based congestion control is optimized for the satellite links with dedicated capacity. LLE-TCP protocol, designed for protocol stacks with multiple positive acknowledgement schemes at different layers, provides cross-layer ACK suppression over the WLAN link, shifting the ACK feedback point from the mobile receiver to the bridge router. The generation of TCP ACKs is driven by acknowledgements from the receiver node arriving at the link layer. As a result, LLE-TCP ensures the successful data delivery up to the destination node, while the main performance improvement is due to the avoidance of TCP ACKs transmission over the WLAN shared link.

The details of ns-2 implementations are presented in Fig. 7. The Booster agent attached to the sender node obtains data packets from the TCP NewReno source generating one TCP ACK per data packet back to the TCP agent (as described in Section III B). The obtained TCP packets are transported over a satellite link using STP subnetwork layer protocol implemented using two pairs of STP agent and its corresponding sink.

The satellite link is modeled using asymmetrical bidirectional GEO-stationary link model with 20 Mbps, 300 ms downlink channel and 6 Mbps, 300 ms uplink channel.

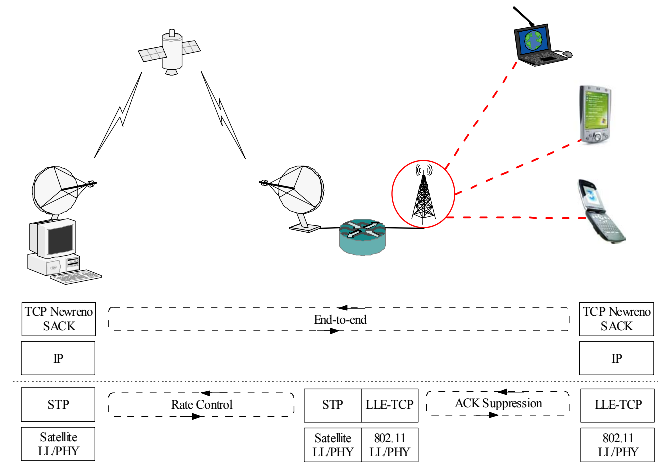

Fig. 6. Simulated scenario: satellite-to-wireless network for disaster reciver or military applications.

LLE-TCP agent and sink introduce acknowledgement suppression over the shared WLAN medium, which is configured to follow the most widely-used IEEE 802.11b specifications. The parameters used in simulation scenario are provided by Table I.

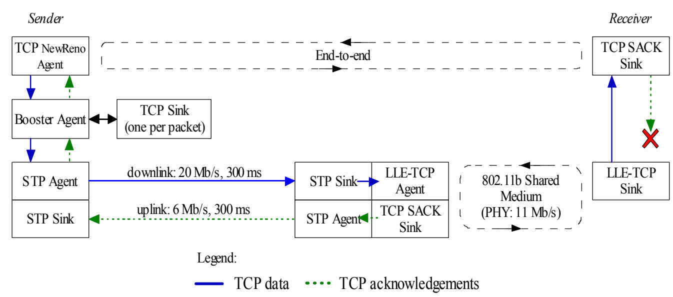

Fig. 7. Ns-2 implementation of the scenario used in simulations.

TABLE I. WLAN Simulation Parameters

Parameter Name	Value
Slot	20 µs
SIFS	10 µs
DIFS	50 µs
PLCP preamble + header	192 µs
Data Rate	11 Mbps
Basic Data Rate	1 Mbps
RTS/CTS exchange	OFF

Being a general solution, the APOHN architecture is not limited to the scenario described above. The choice of this scenario is mainly motivated by the fact that standard TCP/IP protocol suite does not perform well neither in satellite [6], nor in wireless LAN networks [3], and the use of APOHN architecture leads to relevant improvements in terms of throughput, delay and utilization performance of communications (as underlined in the next section).

The simulation scenario models a typical network architecture which can be used for disaster recovery or in automated battlefield military applications with no terrestrial infrastructure available.

APOHN performance evaluation results are presented in the following sub-section. We decided to compare the proposed approach against TCP NewReno with SACK implementation as well as with a satellite PEP solution represented by Satellite TCP Performance Enhancing Proxy (SaTPEP) [22] implementing the connection-splitting approach.

B. Experimental Results

The sender node communicates with the receiver using TCP NewReno flows, which are started at the beginning of each simulation. The sender node is in saturated state - always having data to send for the entire 2000 seconds of simulation duration. The size of TCP data packet is fixed on the Ethernet MTU and equal to 1460 bytes.

The goodput of TCP connection, which is defined as the number of bytes successfully received at the transport layer by the receiver node in a unit of time, is chosen as the main parameter for performance evaluation.

Fig. 8 presents the goodput achieved by evaluated approaches in the scenario where communications are performed using a single TCP flow.

The performance of standard TCP/IP protocol stack is poor: TCP NewReno experiences multiple timeouts which periodically reduce its throughput down to zero. It underutilizes the available for the connection capacity for most of the simulation time, since its additive window increase window policy takes a considerable amount of time to reviver a transmission rate after each congestion drop and the corresponding window reduction.

Congestion window evolution of TCP NewReno flow with respect to the connection capacity is presented in Fig. 9. Most of packet losses due to congestion are not detected based on the receiver feedback. As a result, TCP sender triggers multiple timeouts. However, even in case of successful loss detection based on the reception of a selective acknowledgement, the flow underutilizes capacity available for the connection reducing its window the half of its size.

SaTPEP connection-split approach uses two different protocols for application data delivery. Application data produced at the sender node is first transmitted to the bridge router using a rate-based protocol implementing a negative acknowledgement scheme. Then, after being stored at the application layer in a memory buffer or a hard-drive, the data are transmitted to its destination using TCP NewReno protocol over the WLAN network.

IEEE 802.11b link is clearly the bottleneck for the end-to-end data delivery, thus determining the maximum achievable goodput level. As a result, the performance of SaTPEP is close to the performance of TCP NewReno over the IEEE 802.11b network, with an average goodput fixed at 5.2 Mb/s (see Fig. 8).

Similarly to SaTPEP, the proposed APOHN architecture implements a connection-splitting approach. However, being placed to the newly introduced subnetwork layer, it avoids the requirement for application data storage at the bridge router as well as the delivery of sender-generated TCP packet to the designation preserves end-to-end semantics on per-node level.

The performance of APOHN architecture in the simulated scenario is mostly determined by the LLE-TCP flow operating over the WLAN bottleneck link. Cross-layer ACK suppression performed by LLE-TCP leads to the higher level of goodput with an average of 6.1 Mb/s.

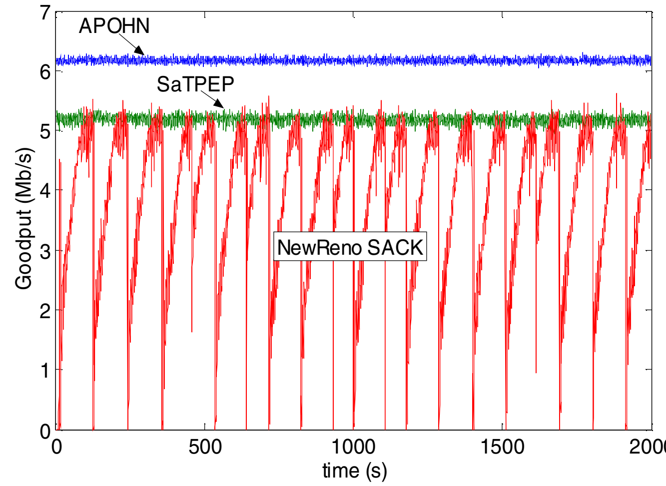

Fig. 8. Goodput simulation results obtained in single-flow scenario.

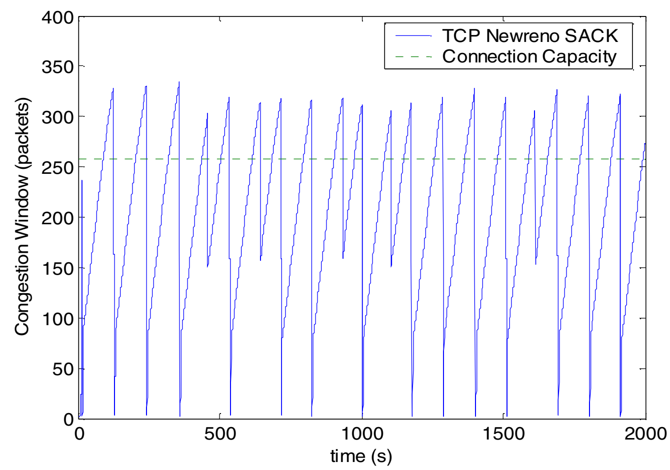

Fig. 9. TCP Newreno congestion window evolution.

TCP NewReno performance degradation is mainly caused by considerably large delay inserted by the satellite link which separates the sender node and the bottleneck link where congestion drops occur. The delayed reaction of TCP sender to a packet loss causes multiple losses which can not be resolved, even with selective acknowledgement policy enabled at the receiver.

Fig. 10 illustrates buffer usage at the bridge router by TCP/IP as well as by the APOHN architecture. Periodically, TCP NewReno completely fills the buffer and then overflows the buffer, which is limited to 50 packets (75 Kbytes). The backpressure congestion control implemented by the APOHN architecture limits the rate of satellite connection. As a result, the bottleneck buffer is used mainly in the range of 5–7 packets, corresponding to 5–15% of the entire space provided by the buffer.

The dynamics of buffer usage are not presented for PEP, due to the fact that PEP approach stores the transmitted data at the application level and the bottleneck's buffer is utilized only by second part of split-connection TCP.

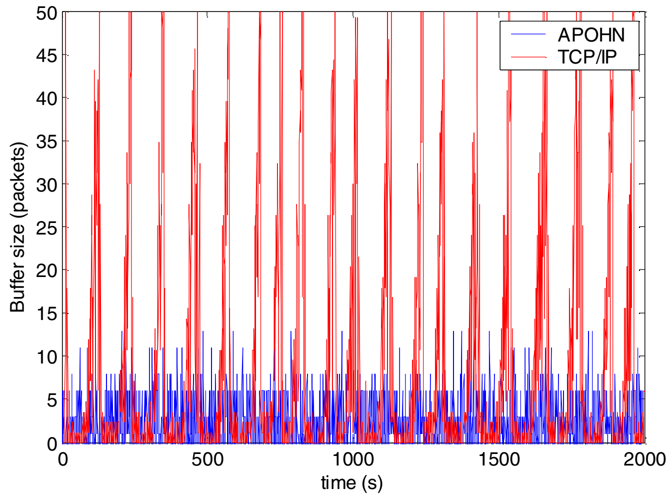

Fig. 10. Buffer usage at the brigde router.

Another performance metric which can be used along with the goodput comparison is the amount of data transported by a given architecture within the simulation time. Fig. 11 presents a comparison between evaluated approaches in terms of the number of TCP data packets successfully transported to the receiver.

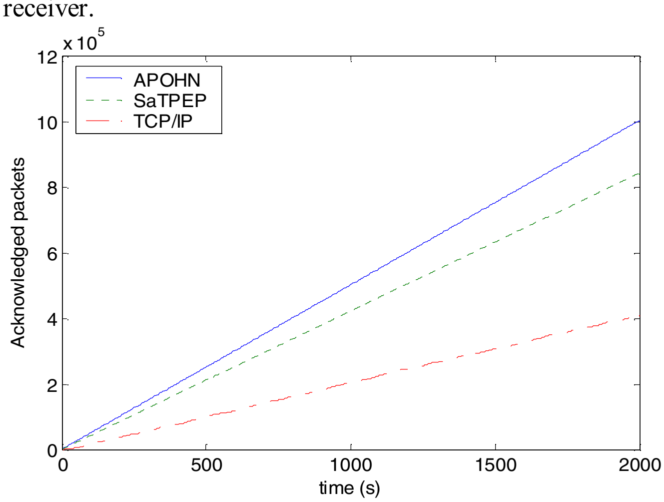

Fig. 11. Comparison by the amount of data transported within the simulation time.

The presented results for single-flow scenario underline performance advantages obtained from the use of APOHN architecture. However, communications performed by network terminals are commonly performed by using multiple flows.

Fig. 12 presents a comparison in terms of achieved cumulative goodput of multiple TCP flows averaged on the simulation duration.

The goodput of TCP NewReno flows produced by TCP/IP protocol stack is dramatically decreased as an increasing number of flows compete for the small bottleneck buffer. The performance of SaTPEP approach is relatively stable with minor decrease of about 10% for the large number of flows (right part of the graph). APOHN architecture, and more specifically LLE-TCP protocol implemented at the WLAN link, is less sensitive to the number of competing flows - which keeps goodput variation within 5%.

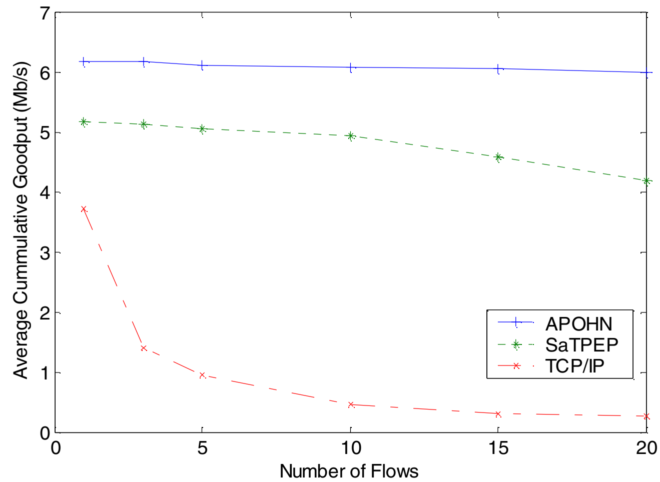

Fig. 12. Throuput performance comparison in multi-flow communication scenario.

V. CONCLUSIONS

The poor performance of TCP/IP protocol suite over heterogeneous network has prompted the proposal of several new architectures. The most successful architectures in such performance optimization domain are Performance Enhancement Proxies (PEP) and Delay-Tolerant Networking (DTN).

In this paper, we propose a novel APOHN architecture for TCP performance improvement over heterogeneous network paths.

APOHN architecture extends ISO/OSI protocol stack model with an additional subnetwork layer, which aims at the introduction of specific protocols operating in a particular subnetwork, being optimized to a particular physical characteristics of this subnetwork.

Flow multiplexing supported at the subnetwork layer combined with the speedup of individual flows produced by Protocol Booster module results in relevant performance improvement.

Another key feature of the proposed architecture is the support of IPsec security mechanism as well as the possibility for an additional enhancement targeted at secure communications within a subnetwork range.

Evaluation of the proposed architecture is performed through simulations in a combined satellite and WLAN network environment, typically used in disaster recovery and several military applications. The results underline performance advantages enabled by APOHN architecture which are compared with standard TCP/IP reference model as well as with a connection-splitting PEP solution.

Summarizing, APOHN is a powerful architecture providing the basis for optimized communication for a variety of data, voice and multimedia applications over non-friendly heterogeneous networks, providing potential insights for the design of next generation networks.

ACKNOWLEDGEMENT

The authors are grateful to Rosario Firrincieli, University of Bologna, Italy for his help, support and valuable comments in the field of satellite communications and PEPs.

References

[1] T. S. Rappaport, A. Annamalai, R.M. Buehrer, and W.H. Tranter, "Wireless communications: past events and a future perspective," IEEE Communications Magazine, vol. 40, no. 5, May 2002, pp. 148 – 161.

[2] C. Barakat, E. Altman, and W. Dabbous, "On TCP performance in a heterogeneous network: a survey", IEEE Communications Magazine, vol. 38, no. 1, January 2000, pp. 40 – 46.

[3] G. Xylomenos, G.C. Polyzos, P. Mahonen, and M. Saaranen, "TCP performance issues over wireless links", IEEE Communications Magazine, vol. 39, no. 4, April 2001, pp. 52 – 58.

[4] F. Granelli, D. Kliazovich and Nelson L. S. da Fonseca, "Performance Limitations of IEEE 802.11 Networks and Potential Enhancements", Yang Xiao and Yi Pan (ed), "Wireless LANs and Bluetooth", Nova Science Publishers, Hardbound, 2005.

[5] T. V. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss", IEEE/ACM Transactions on Networking, vol. 5, no. 3, June 1997, pp. 336 – 350.

[6] C. Partridge and T. J. Shepard, "TCP/IP performance over satellite links", IEEE Network, vol. 11, no. 5, Sept.-Oct. 1997, pp. 44 – 49.

[7] I. F. Akyildiz, O. B. Akan, C. Chen, J. Fang, and W. Su, "InterPlaNetary internet: state-of-the-art and research challenges", Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 43, no. 2, October 2003, pp. 75 – 112.

[8] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations," IEFT, RFC 3135, June 2001.

[9] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary Internet," IEEE Communications Magazine, vol. 41, no. 6, June 2003, pp. 128 – 136.

[10] D. C. Feldmeier, A. J. McAuley, J. M. Smith, D. S. Bakin, W. S. Marcus, T. M. Raleigh, "Protocol boosters," IEEE Journal on Selected Areas in Communications, vol. 16, no. 3, April 1998, pp. 437 – 444.

[11] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," IEEE Network, vol. 11, no. 6, November-December 1997, pp. 10 – 23.

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

[13] S. Kent and R. Atkinson, "Security acrchitecture for the Internet protocol," IETF, RFC 2401, November 1998.

[14] Y. Zhang, "A multilayer IP security protocol for TCP performance enhancement in wireless networks," IEEE Journal on Selected Areas in Communications, vol. 22, no. 4, May 2004, pp. 767 – 776.

[15] J. Sing and B. Soh, "A Critical Analysis of Multilayer IP Security Protocol," Third International Conference on Information Technology and Applications (ICITA), vol. 2, July 2005, pp. 683 – 688.

[16] E. Altman, K. Avrachenkov, and C. Barakat, "A stochastic model of TCP/IP with stationary random losses," Transactions on Networking, IEEE/ACM, vol. 13, no. 2, April 2005, pp. 356 – 369.

[17] D. Clark, "Window and Acknowledgement Strategy in TCP," IETF, RFC 813, July 1982.

[18] NS-2 simulator tool home page. http://www.isi.edu/nsnam/ns/, 2000.

[19] T. R. Henderson and R. H. Katz, "Transport protocols for Internet-compatible satellite networks," IEEE Journal on Selected Areas in Communications, vol. 17, no. 2, February 1999, pp. 326 – 344.

[20] D. Kliazovich and F. Granelli, "A Cross-layer scheme for the TCP Performance Improvement in Wireless LANs," IEEE Global Communications Conference, GLOBECOM'04, Dallas, U.S.A, December 2004.

[21] F. Granelli and D. Kliazovich, "Cross-layering for Performance Improvement in Multi-Hop Wireless Networks", (invited paper) International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN), Las Vegas, Nevada, USA, December 2005.

[22] D. Velenis, D. Kalogeras, and B. Maglaris, "SaTPEP: a TCP Performance Enhancing Proxy for Satellite Links," in PROC 2nd International IFIP-TC6 Networking Conference, Pisa Italy, May2002.

[23] F. Wang and Y. Zhang, "Improving TCP performance over mobile ad hoc networks with out-of-order detection and response," ACM MOBIHOC, Lausanne, Switzerland, Jun. 2002, pp. 217–225.

[24] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," in Mobile Computing and Networking, 2001, pp. 287–297.