# EVCP-EW: Enhanced Variable-structure Congestion-control Protocol for Encrypted Wireless Networks

**Pratima Bhujbal**[*], **Uma Nagaraj**[**]

[*] Department of Computer Engineering, Pune University
[**] MIT, Academy of Engineering, Pune

***Abstract*****-** Now a days congestion in computer network is becoming an important issue. Variable-structure Congestion-control Protocol (VCP) and Double Packet Congestion Control Protocol (DPCP) operating in wireless networks are potentially faced with some challenges of performance degradation. These sources are: 1) The oscillatory behaviour of VCP in the presence of link bandwidth estimation errors is not good. 2) VCP exhibits poor fairness characteristics in high delay networks. 3) DPCP faces deployment obstacles in encrypted wireless networks due to the fact that it relies on partial information in the TCP header and the TCP header information is lost when crossing encryption boundaries. In this paper we propose an alternative congestion control protocol to which we refer as Enhanced Variable-structure Congestion-control Protocol for Encrypted Wireless (EVCP-EW) networks. It does so by passively utilizing the IP Identification field of a packet header instead of the TCP header in conjunction with a heuristic algorithm to differentiate between different sources of loss.

***Index Terms*****-** Markov chain, fading links, congestion control, wireless networks, ECN, Variable-structure Congestion-control Protocol(VCP), eXplicit Congestion-control Protocol(XCP), Double Packet Congestion Control Protocol (DPCP).

## I. Introduction

It has been demonstrated in [4] that conventional TCP and end-to-end TCP-based Active Queue Management (AQM) schemes perform poorly in high Bandwidth-Delay Product (BDP) networks.

As surveyed by [8], there is necessity for protecting not only a congestion-control protocol's data, but more importantly its metadata against bit errors. Such protection was provided relying on the use of FEC codes and/or improving the quality of a link by utilizing multiple antennas. Their results further revealed that VCP represents a high-performing yet practical congestion-control protocol for wireless networks, especially for encrypted networks restricting the number of available header bits for use by a congestion-control protocol. There is necessity to improve the oscillatory behavior of VCP in the presence of link bandwidth estimation errors and enhancing the performance of VCP in encrypted wireless networks by designing and implementing a multipacket protocol version of the protocol.

The Variable-structure congestion Control Protocol (VCP) and eXplicit Congestion-control Protocol (XCP), both protocols encapsulates congestion related information into packet headers and exhibit high utilization and great fairness characteristics while maintaining low persistent queue length and reducing congestion caused loss in wired networks. While XCP requires the use of a large number of IP packet header bits to relay congestion information thereby introducing significant deployment obstacles, VCP only uses the two existing ECN bits in the IP header to encapsulate three congestion levels. Given that VCP demands the use of no extra bits in the IP header, it represents a more practical alternative of deployment than XCP. However, VCP can only deliver limited feedback to end hosts since two bits can at most represent four levels of congestion. In order to avoid sudden bursts, VCP has to control the growth of transmission rates by setting artificial bounds. The latter, yields slow convergence speeds and high transition times. Moreover, due to the use of fixed parameters for fairness control, VCP exhibits poor fairness characteristics in high delay networks.

In contrast, as demonstrated in previous work DPCP proposes a distributed framework that allows for using no more than 2 ECN-bits to deliver a 4-bit representation of the LF. That said, DPCP needs to access partial information in the TCP header in order to be able to efficiently distribute and reassemble the LF. However, in encrypted networks protected by IPSec, TCP header information is lost when crossing encryption boundaries. Thus, DPCP cannot operate in such encrypted networks. Furthermore, wireless networks are characterized by fading related error-caused loss in addition to queuing related congestion-caused loss. Experiments have shown that the performance of any congestion control protocols relies on appropriate reaction to loss according to its source. Like VCP, DPCP reacts to loss without differentiating between the sources of loss and thus performs inefficiently over wireless networks.

In this paper, we propose a new congestion control protocol that improves the design of VCP and DPCP. In contrast to DPCP, our new protocol to which we refer as Enhanced Variable-structure Congestion-control Protocol for Encrypted Wireless (EVCP-EW) networks proposes two new schemes: First, a novel distributed scheme that allows for operation within encrypted networks, and second, a new heuristic loss differentiating scheme that can distinguish between error caused loss and congestion caused loss. Notably, these new schemes are added to EVCP-EW while preserving all of the benefits of DPCP. In EVCP-EW, a congestion level is carried by a chain of two packets and each packet provides two bits out of four bits of information associated with a congestion level. Utilizing a distributed scheme that deviates from that of DPCP, routers compute and distribute a congestion signal into two packets. A congestion level can be specified by concatenating a group of two ECN bits together from a pair of packets at an end node. Incorporated with a novel heuristic algorithm, EVCP-EW can appropriately react to congestion caused loss while avoiding unnecessary reductions of the sending window sizes in response to error-caused loss.

The rest of the paper is organized as follows. In Section 2, we describe the methodology of VCP and DPCP along with their limitations that motivate the design of EVCP-EW. In Section 3, we provide a programmers design and detailed description of EVCP-EW. In Section 4, we discuss the results and related work. Finally, we present several coclusions in section 5.

## II. Related work

In the past few years, variety of techniques have been developed for increasing the efficiency of congestion-control protocols in high- BDP networks. All of the proposed works often fail to achieve both efficiency and fairness because they retain an integrated controller design. To improve the performance of TCP in wireless and satellite networks several approaches have been proposed. These approaches are grouped into three categories. These are split connection protocols such as I-TCP [5] and M-TCP [6], link-layer protocols such as AIRMAIL [7] and end-to-end protocols such as WTCP [9], TCP Westwood [10]. The split connection protocols and link-layer protocols are attempt to hide the errors of the wireless link from the TCP sender. Various approaches have been proposed for designing protocols for high-speed and long distance networks ranging from minor modifications to conventional TCP, to a complete protocol redesign. H-TCP[1] belongs to the former category and represents an evolution of conventional TCP rather than a radical departure from it. I-TCP [5] confines mobility related problems to the wireless link. Alleviate problems by adapting the TCP/IP software on the wireless link that requires no modifications from the hosts on the fixed network. I-TCP [5] particularly suited for applications which are throughput intensive. The snoop module deals with bit-error losses while the routing protocol eliminates the losses during handoff. The snoop modifications consist of caching packets and performing local retransmissions across the wireless link by monitoring the acknowledgments to TCP packets generated by the receiver.

Previous work evaluated the performance of VCP in wireless networks and highlighted several limitations of VCP. Besides DPCP from which EVCP-EW is derived, the closest bodies of work in congestion control to EVCP-EW include MLCP [11] and UNO [12]. The MLCP [11] analyzed the control algorithm of VCP and proposed a multi-level load-factor based protocol to increase the feedback information of VCP. However, MLCP requires the use of extra bits in the IP header. The UNO framework [12] utilizes the IPID field to passively encode the LF. The passive nature comes from a fact that the UNO framework does not modify the value of the IPID field. In EVCP-EW, the idea of passively using the LSB bit of the IPID field is inspired by the UNO framework. Nonetheless, while the work of UNO may seem to share a similar idea with EVCP-EW, it differs from EVCP-EW in several aspects. First, although UNO passively utilizes existing bits in the IPID field of the IP header, it introduces deployment issues. For example, UNO will not work in certain encrypted networks where only 6 ToS and 2 ECN packet header bits can pass through encryption boundaries. In contrast, EVCP-EW only requires the use of two ECN bits in each packet. Second, UNO senders need to collect at least 8 specific packets translating to an average of 8 ln8 = 24 consecutively transmitted packets in order to derive the maximum congestion level before regulating *cwnd*, while EVCP-EW senders perform regulations on a per-ACK basis. Over lossy wireless links, consecutive loss of packets associated with the maximum LF yield an oscillatory behavior in the case of UNO.

## III. Programmer's Design

VCP operates in three congestion regions and attempts at decoupling efficiency and fairness aspects of congestion control. Window management mechanism of VCP is quite different than that of TCP. Instead of using the slow start and congestion avoidance algorithm of TCP, VCP regulates the value of congestion window (*cwnd*) with different congestion control policies defined according to the level of congestion in the network. VCP represents the network congestion status by a load factor which is further mapped into one of three congestion levels labeled as low-load, high-load, and overload. The design of VCP allows for encoding the value of the LF into two ECN bits in the IP packet header. The LF is computed and mapped into one of the three congestion levels mentioned above at a VCP router. Once a data packet arrives, the VCP router extracts the congestion level associated with its most congested upstream link from the ECN bits of the packet itself. It then updates the ECN bits of the packet only if its downstream link is more congested than what is already indicated by the ECN bits of the packet. Eventually, the data packet will carry the congestion level of the most congested link of its session. At the receiver, the congestion level is retrieved and sent back to the sender via an ACK packet. Consequently, VCP applies three congestion control policies: Multiplicative Increase (MI) in the low-load region, Additive Increase (AI) in the high-load region, and Multiplicative Decrease (MD) in the overload region. While the MI operation is utilized to eliminate TCP's slow start behavior, the AI and MD operations attempt at preserving the fairness characteristic of TCP. Since VCP can only provide limited feedback to the sender, its efficiency and fairness characteristics are negatively impacted in moderate bandwidth high delay network operation scenarios. Unlike VCP, DPCP uses four bits to represent the LF. Although DPCP increases the amount of feedback, it utilizes the two ECN bits of a pair of packets in order to encode the LF in a distributed way. For a given LF, the packet that carries the Most Significant Bits (MSBs) of the LF is referred to as *MSP*. Similarly, the packet that carries the Least Significant Bits (LSBs) of the LF is referred to as *LSP*. Each packet has a sequence (*seq*) number and an acknowledge (*ack*) number in its TCP header. During transmission, these two numbers never change. Thus the relative order of these two numbers can be used as a binary indication to tell if a packet is *MSP* or *LSP*. More specifically, if the *seq* number has a greater value than the *ack* number, then the packet is the*MSP*. Otherwise the packet is the *LSP*. Furthermore, DPCP maintains an *MSP* flag at the end nodes. The end nodes flip over the *MSP* flag of every packet to indicate if the next packet should be *MSP*/*LSP*.

Based on the value of *MSP*, end nodes may swap the value of *seq* and that of *ack* in order to use the packet as *MSP* or *LSP* and thus yield an interleaved packet flow with the pattern

"MSP:LSP:MSP:LSP:...". Once a packet arrives at a router, the router identifies a packet as *MSP* or *LSP* by checking the relative order of the *seq* and *ack* of the packet. Then, the router assigns either MSB or LSB bits of the associated LF to the packet depending on whether it is *MSP* or *LSP*. This way, DPCP can provide a more accurate feedback to the sender.

### *3.1. Mathematical Model*

EVCP-EW: Enhanced Variable-structure Congestion-control Protocol for Encrypted Wireless Networks.

As surveyed by earlier work, the design of EVCP-EW is motivated by two observations. These sources are:1) most feedback based congestion control protocols are facing deployment challenges in encrypted network because these protocols either require the use of multiple bits in the IP header or even access to headers of the protocols above the IP layer. 2) most congestion control protocols are designed for wired networks and treat both types of loss as congestion caused loss. While error-caused losses are typically absent in wired networks, they are common in wireless networks. Thus, the target operating environments of EVCP-EW are IPSec-based encrypted wireless networks. The latter means that only eight bits of the IP header, two ECN bits and six Type of Service (ToS) bits, can bypass the encryption boundaries and are available for end to end signaling. As the ToS bits are reserved for signaling differentiated services as oppose to congestion control, EVCP-EW will only use the two ECN bits of the IP packet header for carrying congestion control signaling feedback.

### *A. Overview*

Relying on two new schemes, EVCP-EW extends VCP and DPCP to work efficiently in encrypted wireless networks. Just like DPCP, EVCP-EW uses a double packet four bit representation of the LF, but it introduces a packet ordering management that is quite distinct from that of DPCP. Unlike DPCP, EVCP-EW does not rely on the TCP header to manage packet ordering. Instead, it only utilizes the information available in the IP header and only manipulates two existing ECN bits to carry congestion information. The detail of new packet ordering management scheme of EVCP-EW will be presented in the next subsection. Second, EVCP-EW utilizes a heuristic scheme for differentiating error-caused loss from congestion-caused loss. This heuristic scheme runs at the transmitting side and maintains the history information of congestion status over the bottleneck link of a path. Upon detection of loss, the heuristic scheme makes an identification of the source of loss based on the saved history information. Other components of EVCP-EW such as the definitions of congestion levels, handling exceptions as well as encoding and decoding scheme remain the same as those of DPCP. In the following two subsections, we present the novel aspects of EVCP-EW, namely, how the protocol manages packet ordering and how it differentiates between two types of loss.

### *B. Packet Ordering Management*

As EVCP-EW distributes the LF into two packets, a binary signal is enough to determine packet ordering. However, no free bit is available in the IP header for such signaling. That said, the IPID field of the IP header originating from a host is either monotonically increasing or chosen uniformly at random. In either case, the LSB of IPID flips over quickly enough to be used for signaling *MSP*/*LSP*. Specifically, EVCP-EW only uses the LSB of the IPID field. Further, the use of IPID field bits is passive, i.e., the bit values are inspected but not changed by EVCP-EW. In contrast to DPCP, EVCP-EW uses the LSB of IPID field in order to differentiate *MSP* from *LSP* at the receiving end, instead of swapping TCP *seq* and *ack* numbers. Namely, a packet with an LSB value of zero is used as the *MSP* and a packet with an LSB value of one is used as the *LSP*. As mentioned above, the value of the IPID is set by the IP protocol either incrementally or according to a uniform random distribution. In the former case, the LSB bit flips over for any pair of consecutive packets which is perfect for differentiating *MSP* from *LSP*. In the latter case and despite the fact that the LSB bit might not flip over in every pair of consecutive packets, EVCP-EW uses the first packet with an LSB value of zero for carrying *MSP* and the first packet with an LSB value of one for carrying LSP. As evidenced in our experiments, it is safe to assume that bit flips, with a probability of 0.5, occur quick enough with respect to necessary congestion reaction speed specially over large BDP networks. In what follows, we explain how EVCP-EW operates in IPSec encrypted networks. Assuming that at the encrypted boundaries, only two ECN bits can pass the boundary.

### *C. Operation with IPSec*

IPSec operates in two modes: transport mode and tunnel mode. In the transport mode, the original IP header is kept after getting authenticated by IPSec. Thus, EVCP-EW can still access IPID and ECN bits as usual in IPSec transport mode. In contrast, the entire packet is encrypted and authenticated in IPSec tunnel mode. As a result, the original IP header becomes invisible in the encrypted packet. Since the LSB bit of the IPID in the original IP header may not necessarily be the same as the one in the new IP header, EVCP-EW utilizes the IPID only on the Cypher Text (CT) side but not on the Plain Text (PT) side for packet ordering. In what follows, we present the details of the operation of EVCP-EW in IPSec tunnel mode. As EVCP-EW will be installed and configured at the IPSec router, it is safe to assume that EVCP-EW will have access to both CT and PT headers of a packet.

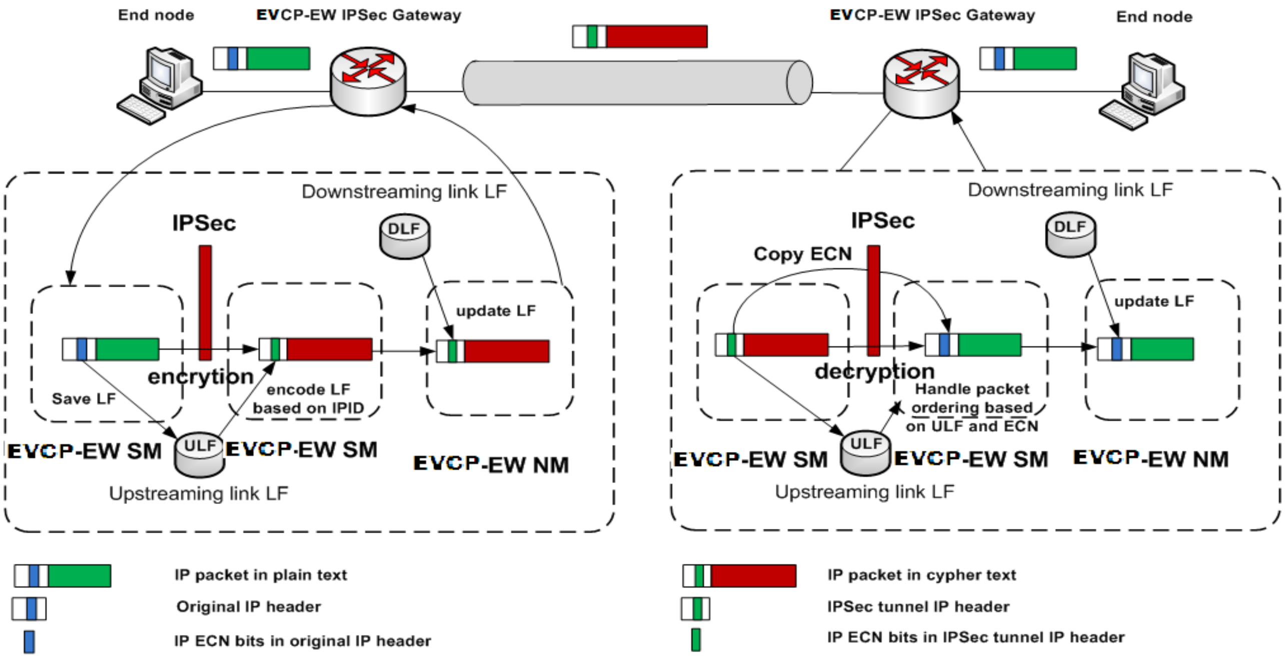


**Fig. 1. An example scenario of using EVCP-EW over an IPSec tunnel.**

Furthermore, because the operations of EVCP-EW in the PT side are similar to that presented in [15], we only focus on the operation of the IPSec router over encryption boundaries and the IPSec tunnel. Specifically, EVCP-EW provides two router modules: i) Security Module (SM) running only on IPSec routers that cooperates with IPSec gateways, and ii) Normal Module (NM) running on both IPSec gateways and other routers. Fig. 1 illustrates a scenario of using EVCP-EW over an IPSec tunnel. Assuming an FTP or a comparable connection has been established, the flow of events at the IPSec gateways is as follows:

1) A EVCP-EW packet arrives at the ingress of an IPSec gateway. Before the packet goes to the IPSec module for encryption, EVCP-EW SM will first catch the packet, save the packet ordering information, i.e., MSP/LSP and the value of the LF as indicated in the ECN bits. Then EVCP-EW SM delivers the packet to the IPSec module. After the new IP header is generated and ready to be transmitted through the tunnel, EVCP-EW SM catches the outgoing packet again and encodes ECN bits with MSB/LSB bits of the saved LF depending on the LSB bit of the IPID in the new IP header. Note that, after the original IP header is encrypted, EVCP-EW has no idea of if the new packet is a TCP packet or a packet using another protocol, e.g., UDP. Thus, EVCP-EW encodes ECN bits regardless of the original protocol type, which introduces overhead for non-TCP packets. In fact, this is the tradeoff between efficiency and protocol complexity. That said, we note that the resulting overhead is not significant because i) it is only introduced when transmitting over IPSec tunnels; and ii) it is only associated with the operations of encoding an LF.

2) At the output interface of the ingress IPSec gateway, EVCP-EW NM takes over. EVCP-EW NM compares the LF in the packet with the LF of its downstream link interface and updates the LF in the packet if necessary following the algorithm introduced in earlier work.

3) At the intermediate router on the CT side, EVCP-EW NM operates as DPCP router module except that EVCP-EW uses the LSB bit of IPID to identify *MSP*/*LSP*.

4) At the egress of the IPSec gateway and before the encrypted packet goes to the IPSec module for decryption, EVCP-EW SM will catch the packet and save the LF value as indicated by the ECN bits of the packet. Note that after the packet is decrypted, the IPSec module will copy the ECN bits from the new IP header to the original IP header on the PT side. However, the packet ordering information cannot be simply transferred to the PT side. While EVCP-EW SM can access both CT and PT side, EVCP-EW SM dedicates to change the contents of the packet as minimally as possible. Simply put, EVCP-EW SM does not directly pass any bits from the CT side to the PT side. Note that, the LSB bit of the IPID in the original IP header is not necessarily the same as the one in the new IP header. Thus, instead of changing the value of the LSB bit of the IPID field in the original IP header for the purpose of matching the one in the IP header used by the IPSec tunnel, EVCP-EW uses the relative order of the TCP *seq* and *ack* numbers as the indication of MSP/LSP after the original IP header is retrieved. In this way, EVCP-EW will not change any bits in the IP header of the decrypted packet. Furthermore, EVCP-EW SM has to keep a copy of the LF of the upstream link of the egress IPSec gateway for each IPSec tunnel. EVCP-EW SM inspects the ECN bits in the packet and compares it with the *MSP/LSP* of the saved copy of the LF of its upstream link. Based on the results of the comparison, EVCP-EW SM manipulates the *seq* and *ack* numbers in order to mark the packet as *MSP* or *LSP*. Then the packet is delivered to EVCP-EW NM. EVCP-EW NM updates the ECN bits according to the LF of its downstream link following the operating mechanism of DPCP.

*Loss Differentiation Heuristic Algorithm*

When operating in any network, a sender can build knowledge about congestion in network as it receiving feedback from its intended receiver.

The congestion status of a network can be continuously tracked by the sender because the feedback is updated with the receipt of every ACK. In this case it is important to realize that a congestion-caused loss event has a much longer duration than an error-caused loss event.

Depending on the above fact, the heuristic algorithm for EVCP-EW assumes that by keeping track of the status of the network a sender can identify the cause of a loss. The heuristic algorithm maintains a revolving congestion theory Bit Map of size N at the sending side in order to track the status of the network. When the ACK is received, the bit at position BM(1) is dropped. After that the bit at position BM(i) with i€{1,…,N} is shifted to the left so it takes the position of bit BM(i-1) , and the bit at position BM(N) is set to 1 if the new ACK indicates congestion, or otherwise to 0. A binary flag called congestion flag (CF) is set to 1 when the rightmost T consecutive bits with T≤N are set to 1 in the bit map. Otherwise, the flag is set to 0. After detection of a loss, if CF flag is set, then that loss is safely determined as a congestion-caused loss triggering an MD operation to *cwnd*. Otherwise, that loss is considered to be an error-caused loss, then the sender simply maintains the current. As opposed to the consecutive T could represent the total number of bits set to 1 in every N bits. In the case of EVCP-EW, the link load factor (LF) is encapsulated in ACK packets, and the *OVER_LOAD* represents a load factor beyond 100%. As, the *OVER_LOAD* represents a LF beyond 100% it is used as the indicator of congestion. According to our experiments, setting N to 32 and T to 16 represents optimal choices. With these choices of values, maintaining a revolving bit map history only requires 4 bytes of storage on a per-flow basis. We set the value of N to 32 for the convenience of implementation. In this case the value of *cwnd* for larger flows could be easily scaled to fit the 32 bits of N. Fig. 2 illustrates the operation of the heuristic algorithm of EVCP-EW.

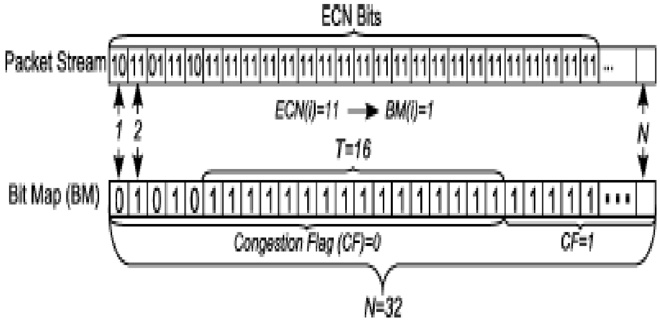
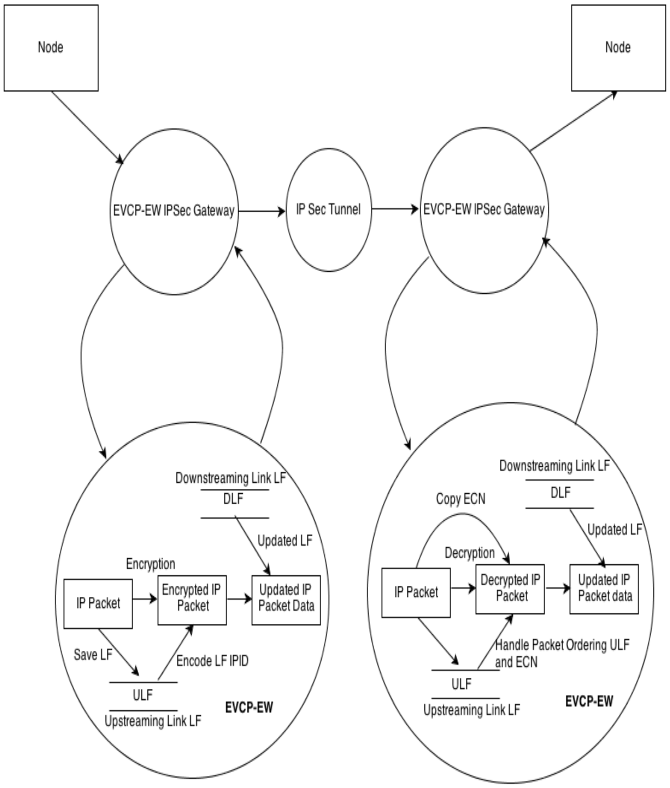


**Fig. 2. An illustration of the loss differentiation heuristic algorithm.**

*3.2. Data Flow architecture*

Figure 3 shows the flow of our proposed system.As shown in figure EVCP-EW packet arrives at the ingress of an IPSec gateway. Then packet goes to IPSec module for encryption.

**Figure 3: Data flow of proposed system**



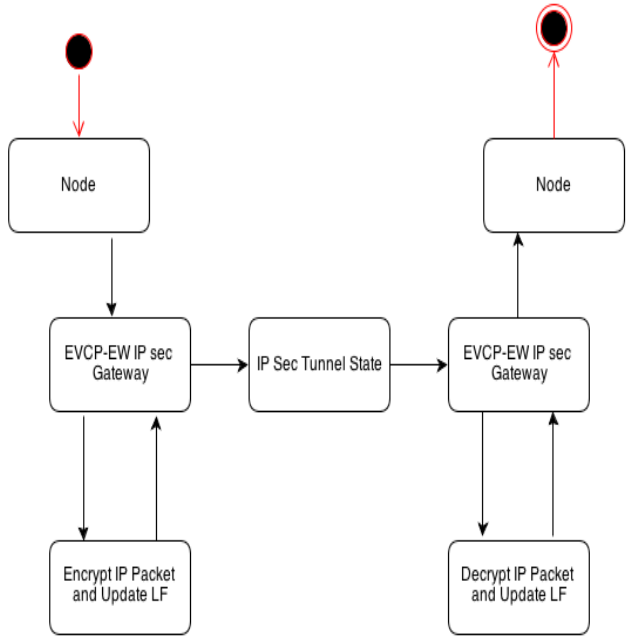
At the output interface of the ingress IPSec gateway, EVCP-EW NM takes over. EVCP-EW NM compares the LF in the packet with the LF of its downstream link interface and updates the LF in the packet if necessary.

At the egress of the IPSec gateway and before the encrypted packet goes to the IPSec module for decryption, EVCP-EW SM will catch the packet and save the LF value as indicated by the ECN bits of the packet. Note that after the packet is decrypted, the IPSec module will copy the ECN bits from the new IP header to the original IP header on the PT side.

*3.3. Turing Machine*
*State Diagram:*

We use the state diagram to specify the sequencing behavior of objects in a class. A state represents a discrete, continuous segment of time wherein the object's behavior will be stable. The object will stay in a state until it is stimulated to change by an event. Figure 4 shows the five main steps of our proposed system.

**Figure 4: State diagram of proposed system**



## IV. Results and Discussion

Our recent work of [12] proposes a distributed approach that can overcome the limitations of VCP by increasing the amount of feedback to the sender. By distributing a 4-bit representation of the LF into two consecutive packets, DPCP only needs to use two ECN bits in one packet preserving the deployment potential of VCP. However, DPCP requires access to TCP header in order to perform encoding and decoding of the LF. The latter introduces difficulties for working in encrypted networks. It is also important to note that all VCP alternatives are faced with similar deployment issues in encrypted networks. In contrast, EVCP-EW proposed in this paper is capable of working in encrypted networks by using an alternative packet ordering management scheme. As EVCP-EW also distributes the LF into two consecutive packets the same way as DPCP does, this work can be viewed as an extension of DPCP for wireless networks. Most importantly, EVCP-EW provides a loss identification algorithm to enable proper reaction to loss depending on its cause, while other VCP alternatives have no such capability.

## V. Conclusion

In this paper, we proposed EVCP-EW as an extension of VCP & DPCP. We demonstrated how EVCP-EW overcomes the limitations of DPCP and VCP by using an alternative packet ordering management scheme. Rather than accessing the TCP header, EVCP-EW passively inspected the LSB bit of the IPID field in the IP packet header to identify whether a packet is the *MSP* or *LSP* in a packet pair sequence. Furthermore, EVCP-EW utilized a heuristic loss identification scheme to differentiate error-caused loss from congestion-caused loss such that it can appropriately react to loss. As the main differentiating factors, we will show that i) unlike DPCP, EVCP-EW can operate over IPSec encrypted networks, and ii) relying on its heuristic loss identification algorithm, EVCP-EW will significantly outperform DPCP in wireless environments. As a future work, it would be interesting to study what improvements are possible in EVCP-EW by using more bits for the congestion related information.

## References

[1] D. Leith and R. Shorten, "H-TCP: TCP for high-speed and long-distance networks," in Proc. PFLDNet, Feb. 2004.

[2] C. Jin, D. Wei, and S. Low, "FAST TCP: Motivation, architecture, algorithms, performance," in Proc. IEEE INFOCOM, 2004, vol. 4, pp.2490–2501.

[3] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman, "One more bit is enough," in Proc. ACM SIGCOMM, Aug. 2005, pp. 37–48.

[4] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," IEEE/ACM Trans. Netw., vol. 5, no. 6, pp. 756–769, Dec.1997

[5] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," in Proc. 15th ICDCS, Vancouver, BC, Canada, May 1995, pp. 136–143.

[6] K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," Comput. Commun. Rev., vol. 27, no. 5, pp. 19–42, Oct. 1997.

[7] Xiaolong Li, Homayoun Yousefi'zadeh,"Analysis, Simulation, and Implementation of VCP: A Wireless

[8] Profiling," IEEE/ACM Trans. Netw., vol. 18, no. 5, Oct.2010

[9] E. Ayanoglu, S. P. T. F. LaPorta, K. K. Sabnani, and R. D. Gitlin, "AIRMAIL:A link-layer protocol for wireless networks," Wireless Netw., vol. 1, no. 1, pp. 47–60, Feb. 1995.

[10] C. Parsa and J. Garcia-Luna-Aceves, "Improving TCP performance over wireless networks at the link layer," Mobile Netw. Appl., vol. 5, no. 1, pp. 57–71, Mar. 2000.

[11] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bhargavan, "WTCP:A reliable transport protocol for wireless wide-area networks," in Proc.ACM MobiCom, Seattle, WA, Aug. 1999, pp. 231–241.

[12] C. Casetti, M. Gerla, S. Mascolo, M. Sanadidi, and R. Wang, "TCP Westwood: End-to-end congestion control for wired/wireless networks," Wireless Netw., vol. 8, no. 5, pp. 467–479, Sep. 2002.

[13] [12] I. A. Qazi and T. Znati, "On the design of load factor based congestion control protocols for next-generation networks," in Proc. of the IEEE INFOCOM 2008, Apr. 2008.

[14] [13] N. Vasic, S. Kuntimaddi, and D. Kostic, "One Bit Is Enough: a Framework for Deploying Explicit Feedback Congestion Control Protocols," in Proc. of the First International Conference on COMmunication Systems and NETworkS (COMSNETS), Jan. 2009.

[15] [14] M. Goutelle, Y. Gu, and E. He, "A Survey of Transport Protocols other than Standard TCP," 2004, https://forge.gridforum.org/forum/forum.php?forum id=410.

## Authors

**First Author** – Pratima Bhujbal received her Bachelor's degree in Computer Engineering. Now; she is pursuing her M.E degree in Computer Engineering from MIT Academy of Engineering, Pune University, Pune, India. Now she is Lecturer in JSPM's Bhivrabai Sawant Polytechnic, Wagholi, Pune Her research areas are Computer Networking., Email- pratima5bhujbal@gmail.com

**Second Author** – Prof. Uma Nagaraj, B.E., M.E. Computer was educated at Belgum University. Now, she is pursuing her Phd. She has worked in various capacities in academic institutions at the level of Professor, Head of Computer Engineering Department. Now, she is Prof and Hod in MAE, Alandi, Pune. Her areas of interest includes Neural Network, Image Processing, Ad-Hoc Networks, VANET., Email- umanagaraj67@gmail.com