A low complexity mechanism for congestion notification in rural IPSec-enabled heterogeneous backhaul networks

Carlos Figuera | Eduardo Morgado | Esteban Municio | Javier Simó-Reigadas

1Department of Signal Theory and Communications, Rey Juan Carlos University, Madrid, Spain
2IDLab—Department of Mathematics and Computer Science, University of Antwerp—imec, Antwerp, Belgium

Correspondence
Javier Simó-Reigadas, Department of Signal Theory and Communications, Rey Juan Carlos University, Madrid, Spain. Email: javier.simo@urjc.es

Present Address
Javier Simó-Reigadas, URJC - Campus Fuenlabrada, Camino Del Molino, s/n - 28943 Fuenlabrada, Madrid

Funding information
FP7 Coordination of Non-Community Research Programmes, Grant/Award Number: TUCAN3G IST-601102 STP

Summary
Management in wireless backhaul networks is a challenging task, especially in rural and isolated environments. In these scenarios, the backhaul network usually consists of a set of heterogeneous wireless links that provide limited and variable bandwidth to the access networks, often 3G/4G small cells. Because of the highly constrained nature of this type of backhaul network, intelligent and joint management in both the backhaul network and the access network is crucial in order to avoid performance degradation caused by traffic congestion. In order to avoid the saturation in the backhaul network, access networks should consider the backhaul state when taking decisions in the admission control and scheduling procedures. However, no standardized mechanisms currently exist for sharing management information between both networks. In this work, we propose to use the Explicit Congestion Notification (ECN) bit in the outer IP headers present in the Iuh 3GPP IPSec-enabled interfaces in order to notify the backhaul congestion state to the access network. We analyze for the referred scenario, compatibility and security details, validating our approach by running numerically simulations and implementing the notification mechanism. Our low complexity approach offers 2% accuracy and backhaul update latency lower than 10 ms during 80% of the time, which makes the solution appropriate for admission control and scheduling intervals in small cells.

KEYWORDS
ECN, IPSec, network congestion, rural networks, wireless backhaul

1 INTRODUCTION

3G and 4G networks are widely common in urban areas worldwide. However, most rural areas, especially in developing countries, lack of decent connectivity infrastructures. Usually operators only provide coverage as far as revenues compensate their infrastructure investment, which leaves many sparsely populated areas without internet connectivity or even 2G telephony. For these cases, new approaches have been proposed in order to offer Information and Communications Technology (ICT) services to these regions, such as community infrastructures, where users are the owners of the network. However, these approaches are mainly focused on household connectivity and mobile 2G.

Abbreviations: AN, access network; BH, backhaul; ECN, Explicit Congestion Notification.
For providing 3G and 4G connectivity, one of the most cost-effective approaches is to use shared low-cost private infrastructures where the operator offers services adjusted to the real needs of the target population. These infrastructures may consist of an heterogeneous wireless backhaul (BH) network that interconnects a number of small cells deployed as access networks (ANs). In many real setups, the BH network is composed of a set of long-distance wireless links (e.g., WiFi, WiMAX, and VSAT). An example of such network is illustrated in Figure 1. ANs are usually small cells built with inexpensive lightweight low-power Home NodeB’s (HNB). This approach allows to provide guaranteed 3G and 4G services in specific isolated areas at lower cost than traditional wide range cellular networks operating with nonconstrained BH networks.

However, small cells require stable and low latency BH networks in order to provide guaranteed QoS, and therefore an intelligent BH management becomes crucial. When small cells are used in urban environments, where xDLS or fiber technologies are widely available, the BH management among the small cells is not particularly problematic. However, when the BH is a highly constrained, heterogeneous set of wireless links, managing the BH becomes a challenge, since the provided capacity may be limited and highly variable, and the latency and throughput perceived by the AN may be seriously affected by the level of saturation in the BH. For example, if a WiFi link is reaching its saturation point, an increase of 1% in the load can increase the latency up to 20 times without even increasing the offered throughput. Usually, this issue is solved in the edges of the network with traffic engineering and QoS rules. However, additional mechanisms are needed to notify the HNB’s and modify their admission control policy (i.e., blocking current and future exceeding connections), improving eventually the overall users’ experience.

The AN dependency from the BH makes crucial to jointly control the operation of both networks, meaning that AN admission control and scheduling should take into account the actual instantaneous QoS provided by the BH. Since small cells are not designed to perform these tasks and are not even able to communicate with the BH nodes, a new approach for low complexity lightweight real-time monitoring of the BH is required. This approach must provide to the AN-node with specific information about its own BH, no matter what happens with the rest of the transport network. However, the 3GPP ANs are designed to be BH technology-agnostic, and there is no standardized interface to communicate both
network segments. The BH network supports the Iur or Iuh 3GPP interface by encapsulating the user data and signaling as TCP or RTP/UDP traffic from the AN and are always secured using IPSec tunnels.

In this work, we proposed a simple approach to use the Explicit Congestion Notification (ECN) bit of the outer IP Header in the IPSec tunnels in order to provide to each AN with instantaneous feedback about the congestion state of the BH network segment that supports its Iuh interface. This way, ANs can adjust their admission control and scheduling policies in order to reduce congestion in the BH or increase the service load in case resources are available in the BH.

The proposed mechanism has very low complexity (lightweight), is distributed (it only affects the interface between the AN-node and the edge node in the BH), and is highly scalable. Also, it is compatible with the standard usage of ECN bits, since the original ECN encoding is carried in the inner IP headers. Our results show that the proposed method provides a simple but effective way for notifying congestion from the BH to the AN in the targeted scenarios.

The reminder of this paper is organized as follows. Section 2 provides an overview of the different existing monitoring solutions for BH networks. Section 3 introduces the ECN bit and its different usages, and Section 4 describes our proposed ECN-based solution, addressing compatibility and security details. Finally, Section 5 validates the proposed solution for different scenarios and configurations.

2 | RELATED WORK

Since the outbreak of Software Define Networks (SDN), most of the state of the art in network management mechanisms follows this approach, and by using a centralized controller, the BH congestion problem would become trivial. There is a number of works that study how small cells can be optimized while being aware of interference, energy efficiency, dynamicity, and latency. These studies locate the intelligence in the centralized controller that runs an optimization algorithm or an heuristic to solve the problem. However, applying SDN principles in rural BH networks is not straightforward. On one hand, moving away from distributed routing protocols, which are scalable and robust, to a comparable SDN controller with the same scalability and reliability can be challenging. Second, most of the current wireless BH networks in rural areas are not SDN-enabled, are equipped with old management tools, and have human operators that are not familiar with the technology. In order to solve this, a number of SDN architectures exist that support evolutionary upgrades to SDN. These architectures allow to keep the legacy management tools and to gradually update the network with SDN capabilities. Another alternative is to use Hybrid SDN networks that allow to combine non-SDN with SDN equipment and techniques. Despite this, we think that the proposed ECN-based mechanism is still simpler and more cost-effective for the given task.

In addition to that, additional techniques to measure congestion inside the BH are needed. Traditional flooding approaches to measure the available throughput are not suitable for BH networks, since they significantly degrade the network operation while the tests are being performed. This implies that they do not accurately provide the actual non-saturate throughput in a link, ie, the throughput offered with bounded. There are also a number of passive methods that provide an accurate measurement of the actual traffic in an interface at almost real time. Although these approaches are simple and lightweight, they require extra information about the actual maximum capacity of a given link in order to estimate its actual congestion and how far from the saturation point its current load is. The actual maximum capacity in a link is usually pre-established when designing the network by a given modulation and codification scheme; however, it is not an accurate enough figure since other variables such as interference and momentary fading are not considered. More elaborated approaches involve a minimum of signaling overhead while being efficient in identifying the level of congestion of the link regardless of its nominal capacity. In the present work, we take an equivalent approach for measuring the link congestion level as described in Section 4.6.

3 | EXPLICIT CONGESTION NOTIFICATION

The use of ECN with TCP is described in Ramakrishnan et al. and its usage with RTP/UDP in Westerlund et al. TCP's congestion control algorithms are based on the idea that the network is a black box. This means that end nodes measure the network state without any interaction with the intermediate nodes. With basic TCP mechanisms, if the network is congested and some packets are dropped in any intermediate router, TCP detects the packet drops and reacts by diminishing the sender bit rate. However, if active queue management is used, eg, Random Early Detection (RED), alternatives to packet dropping exist. The main one is to use the Congestion Experienced (CE) codepoint in the IP header.
FIGURE 2  Location of Explicit Congestion Notification (ECN) in the IPv4 header (Type of Service) and in the IPv6 header (Traffic Class)

TABLE 1  ECN bit usage

| ECN bit usage          | Description                        |
|------------------------|------------------------------------|
| 00                     | Non-ECN Capable Transport Network  |
| 01                     | ECN Capable Transport Network ECT(0)|
| 10                     | ECN Capable Transport Network ECT(1)|
| 11                     | Congestion Encountered (CE)        |

Note. ECT(0) and ECT(1) are treated as equivalent. Abbreviations: ECN, Explicit Congestion Notification.

to explicitly notify congestion. This is possible since active queue strategies aim to detect congestion before it happens (ie, before the queue is full). When a network is ECN capable, it can modify the CE code point and therefore, it is possible to notify congestion to the transport layer in the sender node without dropping packets in the path.

ECN is based on two bits that are placed in the IP header as it is illustrated in Figure 2. Specifically, IPv4 Type of Service octet and IPv6 Traffic Class octet is divided in two parts. The six first bits are used as Differentiated Service Code Point (DSCP), and the last two bits (7 and 8) are used for the ECN field. Table 1 shows the different meanings to the four ECN values.

If a connection is ECN-enabled, the sender will set the ECT codepoint (ECT(0) or ECT(1)) in all the packets. In general, routers will indistinctly use the ECT(0) and ECT(1) codepoints. Then, if an intermediate router detects congestion, it sets the ECN bits to CE. If during the negotiation of the transport connection, sender and receiver choose to use ECN, the sender will set the ECT codepoint in all the packets, setting up a ECN-enabled connection. If an intermediate router detects congestion and the ECT codepoint is set, instead of dropping a packet, it simply sets the ECN bits to CE. When an intermediate router receives a CE packet, it forwards the packet as usual without changing the CE codepoint. The role of the receiver and the sender varies depending on the transport layer. The use of ECN has been defined for TCP, SCTP, DCCP, and RTP with UDP. In the Iuh interface, the main transport protocols are TCP and RTP over UDP, this one optionally controlled by Real Time Control Protocol (RTCP).

3.1 | ECN with TCP

When a TCP receiver detects the CE, it sets the ECN-echo flag in the next TCP ACK. When the sender receives a packet with this indication, it reacts in the same way as when detecting a packet drop. Then, the sender sets a flag in the TCP header to acknowledge the reception of the ECN-echo.

3.2 | ECN with RTP/UDP

Congestion control for real time applications over RTP/UDP is particularly important, since there is no retransmission when a packet is lost. Before the transport connection is established, RTCP initializes the ECN mechanism by ensuring that all nodes are ECN-capable. When the receiver detects a packet with the CE codepoint, it notifies this to the sender using RTCP feedback mechanisms.

When an IPSec tunnel is used between the sender and the receiver (or in a part of the path), the TCP or UDP sender may not take the ECN bits in the external IP header into account, since they are not secured. In this case, the ECN bits in the external header may be simply discarded. This is called limited-functionality. Alternatively, the other option is full-functionality that takes these bits into account together with the ECN of the internal IP header (which is secured). In that case, an OR operation is done between the congestion notifications (there exists congestion if it is marked in the inner or the outer headers). Table 2 represents all the possible values for the ECN field in the inner and outer headers at the tunnel ingress and egress. ECT(0) or ECT(1) equal to a ECN-capable network with no congestion detected, and CE equals to congestion detected in a ECN-capable network. This means that when the outer header is removed at the tunnel egress, the CE codepoint is set in the ECN field of the inner header if it is set in the outer header. Otherwise, the inner header is not changed and will contain the value it had before the tunnel ingress point.
TABLE 2  ECN field in the inner and outer headers at the IPSec tunnel ingress and egress: Standard ECN usage

| Tunnel Ingress | Tunnel Egress |
|----------------|--------------|
| Inner h. | Outer h. | Not affected | Outer h. | Inner h. |
| Not-ECT | Not-ECT | ECT | Cong. det. | CE | CE |
| ECT | ECT | Cong. det. | CE | CE |
| ECT | ECT | No cong. det. | ECT | ECT |
| CE | ECT | Cong. det. | CE | CE |
| CE | ECT | No cong. det. | ECT | CE |

Abbreviations: CE, Congestion Experienced; ECN, Explicit Congestion Notification.

### FIGURE 3
Reference backhaul (BH) with two parts (managed and not-managed). Blue dashed line represents the interface between the access network (AN) and the BH.

### 4 | DESCRIPTION OF THE PROPOSED SOLUTION

#### 4.1 | Reference scenario

We consider the reference scenario depicted in Figure 3. The BH is divided into two parts: one between the edge node and the gateway router (GW), called **managed BH**, and another between the GW and the Core Network Gateway (CN-GW), called **not-managed BH**. This split aims to consider a general scenario by including a part of the network, which is controlled by the operator (e.g., a multi-hop wireless BH) and another that is not under its control (e.g., a leased transport network).

If the connection between the AN and the CN-GW uses ECN, the nodes in the not-managed BH may notify congestion using the CE codepoint. Our solution shall consider this situation to preserve that congestion notification, if desired.

The main objective of our approach is to use the ECN bits to notify not only the existence of congestion in the managed BH but also its severity, while preserving congestion notifications that may come from the not-managed BH and supporting both TCP and RTP/UDP traffic.

#### 4.2 | Overview of the solution

The edge node is assumed to know the congestion level of the managed BH at any time instant. In our scenario, the mechanism to achieve this is described in Section 4.6. The nodes in the managed BH are configured to be not-ECN capable, so they do not modify the ECN bits by default. We propose to change the meaning of the outer ECN bits only in the interface between the edge node and the AN-node (blue dashed vertical line in Figure 3). This interface is a wired connection between each edge node and each AN-node (e.g., HNB). The new usage of the bits is:

1. First bit ($b_0$) is used to notify congestion in the not-managed BH. This bit is set to one if the packet is marked with the external-header CE codepoint.
2. Second bit ($b_1$) is used to notify the congestion level in the managed BH.

For using $b_1$, the following method is applied. Let $\eta$ denote the percentage of congestion. Then, for coding the value of a congestion of $\eta$, bit $b_1$ is set to one for only 1 of each 100/$\eta$ packets. This is basically to modify the $b_1$ bit to notify congestion by setting it to 1 or 0 during a certain period. The proportion of ones and zeroes during that period is used to calculate the congestion level. For example, if we set four levels of congestion (0%, 50%, 75%, and 100%), the ECN of at least four packets must be read in the AN-node in order to be able to estimate the congestion level. Then, for a congestion of 25%, the edge node of the managed BH will set the bit $b_1$ to 1 in 1 of every four packets. For a congestion of 50%, two of four packets should have their bit $b_1$ set to 1, and so on. The granularity in the congestion notification will determine the minimum number of packets $N_\eta$ that have to be received before obtaining $\eta$, and hence the update period in which the HNB will be notified of the congestion level.
TABLE 3  Explicit Congestion Notification (ECN) fields at the IPSec tunnel ingress and egress with the proposed method

| Tunnel Ingress | Tunnel Egress |
|---------------|--------------|
| Inner header  | Outer header  |
| Not-ECT       | Not-ECT      |
|               | Not-managed network |
| ECT           | ECT          |
| CE            | ECT          |
| CE            | ECT          |

| Tunnel Egress | Reconstructed outer header | Inner header |
|---------------|---------------------------|--------------|
| Outer header  | (b₀, b₁)                  | (0, x)       |
| Not-ECT       | ECT                       | CE           |
| ECT           | (0, x)                    | ECT          |
| CE            | (1, x)                    | CE           |
| CE            | (0, x)                    | ECT          |
| ECT           | (1, x)                    | CE           |

4.3 Compatibility with standard ECN usage

If the AN-node establishes a connection with full-functionality ECN with CN-GW, after reading $b_0$ and $b_1$, the outer ECN field must be set back to the original value. Table 3 shows all the possible combinations of ECN values in the IPSec tunnel ingress and egress. Three states are possible for the outer ECN field (ECT, Not-ECT and CE) but only two values have been stored ($0$ or $1$ in $b_0$). However, if the outer ECN field in the egress part of an IPSec tunnel is ECT or Not-ECT, the receiver must ignore it, so in fact, only two values are relevant in the outer ECN field at the receiver: CE and any other value. Hence, after reading the congestion information for both the managed BH and the not-managed BH, the outer ECN is rewritten and set to CE (ie, congestion is notified to the receiver transport layer) if $b_0$ is 1, or to ECT otherwise. Comparing the two first and the last columns in Tables 1 and 2, we observe that standard usage of the ECN bits is not compromised.

4.4 Implementation details

The congestion level ($\eta$) is coded as the fraction of packets in a window of length $N_w$ (packets) with $b_1 = 1$. Two types of window are considered: nonoverlapping window (NW) and sliding window (SW).

4.4.1 Nonoverlapping window

With an NW of length $N_w$ packets, a new estimation of the congestion level is obtained each $N_w$ packets. The length of $N_w$ depends on the precision required for the congestion level at the HNB, which is obviously quantized. Let $e_{max}$ be the maximum error allowed for the congestion level estimation. Then,

$$N_w \geq \left\lceil \frac{100}{2e_{max}} \right\rceil.$$  \hspace{1cm} (1)

For example, applying Equation 1 in an window with congestion level $\eta \in \{0, 25, 50, 75, 100\}\%$, the minimum value for $N_w$ would be 4, and $e_{max} = 12.5\%$. On the other hand, the congestion is estimated in the HNB reading $b_1$ during $N_w$ packets as the mean value of the stored vector:

$$\eta(t_0 + T_w) = \frac{100}{N_w} \cdot \sum_{k=n-N_w+1}^{n} b_1[k],$$  \hspace{1cm} (2)

where $b_1[k]$ is an indication function that equals 1 if $b_1 = 1$ in packet k and $T_w$ is the time duration of the window.

4.4.2 Sliding window

The SW approach can be used in order to reduce the latency in the notification congestion. This approach can be useful for those cases in which the scheduling period is short (eg, in HSPA is 2 ms) and when sudden changes in the BH are usually present. In order to compute and update the congestion level $\hat{\eta}[n]$ each time a packet arrives, a general windowing function can be applied:

$$\hat{\eta}[n] = 100 \sum_{k=n-N_w+1}^{n} b_1[k] \cdot \omega[k - n].$$  \hspace{1cm} (3)

where $b_1[n]$ is the value of bit $b_1$ for the $n$th packet and $\omega[n]$ is a positive function that performs a weighted average of the last $N_w$ values of $b_1$ (eg, rectangular or exponential window). Finally, the transmitter shall set the value of $b_1[n]$ according...
to the following rule:

\[ b_1[n] = \text{argmin}_{i \in \{0, 1\}} |\hat{\eta}[n]|_i - \eta[n] \]

where \( \eta[n] \) is the true congestion at time instant \( n \) and \( \hat{\eta}[n]|_i \) is the congestion level estimated by the AN-node if a packet with \( b_1[n] = i \) is received. Note that, since the AN-node and the edge node are linked by a short dedicated wired connection, we can assume that all packets are successfully received without loss.

### 4.5 Security issues

The proposed mechanism is designed to cope with a general scenario where the transport network may have a segment, which is not controlled by the operator. The method notifies not only about the congestion level in the managed BH but also about the existence of congestion in the not-managed BH. This part of the network may not be secured, so ECN bits of the external IP header may be changed by a malicious attacker if ECN with full functionality is used. In that case, a false congestion notification would be generated in the AN. This is a general limitation of the use of fully functional ECN, with or without our approach.\(^40\) Hence, if security in the unmanaged-BH cannot be assured, bit \( b_0 \) should be ignored (limited functionality), and then no information about congestion in this part of the network would be provided. The value of \( b_1 \) is based on internal congestion measurements, and it is used only in the link between the AN-node and the edge node, which is typically secure.

### 4.6 Measuring congestion

As stated in Section 2, a way to estimate the actual congestion of the link regardless of its nominal capacity is required. On the basis of the work of Rattaro and Belzarena,\(^33\) we have implemented a statistical descriptor of a common wireless link state.

This descriptor is the mean value of the variable \( K_n \), computed as

\[ K_n = K_0 + \sum_{j=1}^{n} \Delta_j \]

where \( \Delta_j \) is the jitter at time instant \( j \) (difference of delays between packet \( j \) and packet \( j-1 \)), and \( K_0 \) an initialization parameter. From Rattaro and Belzarena,\(^33\) the correlation coefficient between \( E(K_n) \) and predicted throughput is 0.86. Since statistical learning tools are used, we take a set of samples with known capacity (C), in which each sample is represented as \( x = (E(K_n), C) \). With this set of samples, a support vector machine (SVM) have been trained, which learns the relationship between input variables and congestion level. In the test phase, a periodic short train of packets is transmitted to estimate the BH state. Using the model created by the SVM and the jitter information read in the receiver, an estimation of the actual network throughput is obtained.

In our setup, we run different experiments to compute and validate the correlation between jitter and capacity in a testbed with two IEEE 802.11n links as described in Figure 4.

From node A, traffic is injected towards the node C in order to congest the network, using D-ITG.\(^41\) Once the flooding traffic is in the network, additional measurement bursts of 200 packets are used each time, and 1500 bursts are periodically transmitted for each experiment, progressively increasing the congestion level in the network. Nodes 1 and 3 are synchronized with NTP\(^42\) with a maximum accuracy of 2 \( \mu s \). Measurements are made in both directions (alternating node A and C as burst transmitters). The receiver measures the unidirectional delay, round trip time, packet loss, and received throughput. For all the case cases, we compute the mean jitter, the standard deviation of the jitter, and packet loss, for different congestion levels. In order to obtain more robust results, we also experiment with different

![FIGURE 4](image-url)
modulation and codification schemes (MCS), with and without VoIP traffic (which is transmitted with absolute priority) and unidirectional and bidirectional flooding traffic for three different flows A-B, B-C, and A-B-C.

Each experiment consists in 16 different configurations with 1500 samples each. Then, correlation coefficients between mean jitter and input rate and standard deviation of jitter and input rate are computed. In all cases, the correlation coefficient is higher than 0.65 and goes up to 0.75 for mean jitter and 0.85 for standard deviation of the jitter. As an example, Figures 5, 6, and 7 represent how the mean jitter, its standard deviation, and the packet losses vary regarding the congestion level, respectively. In this example, 1500 samples were taken in each load point for a wireless link with MCS1 (QPSK-1/2), injecting unidirectional traffic without VoIP and only congesting the A-B segment. As it can be seen, the packet loss and the mean jitter suddenly increase for a congestion level of approximately 80%. This is because the link always reaches the saturation point before reaching the maximum actual capacity. Precisely, wireless BH links should always work under the saturation point, since although all available throughput is not used, the delay is kept bounded.

Although our figures are lower than the ones obtained in Rattaro and Belzarena, they allow us to be confident enough to use this measurement congestion mechanism in order to estimate the congestion in our ECN-enabled BH tests.

5 | VALIDATION OF THE PROPOSED SOLUTION

5.1 | Basic functionality

We have tested the basic functionality of the proposed procedure in a real network similar to the one described in Figure 4 and have implemented the two procedures, one for modifying the ECN in the edge node (ECN writer, encl injector) and other for read and decode the ECN in the HNB (ECN reader, encl reader).†

For this test, the injector generates its own Secured IP packets and codes the congestion level $\eta$% by setting $b_{1} = 1$ in $N_{1}$ packets during a window of $N_{w}$ packets, with $N_{1} = \left\lfloor N_{w} \frac{\eta}{100} \right\rfloor$, where $\lfloor \cdot \rfloor$ is the rounding operator. The ECN reader reads all IP packets that come from a specific interface and estimates the congestion level at the end of each window as $(1/N_{w}) \sum_{n=1}^{N_{w}} b_{1}[n]$, which corresponds to using a rectangular NW (ie, the congestion estimate is updated each $N_{w}$ packets). Figure 8 shows the real and estimated congestion using these scripts. Blue line represents the actual load of an UDP flow, and colored lines represent the estimated congestion level for different values of $N_{w}$ (5, 10, 20, and 40). It can be observed

---

*This effect is especially significant in content-based wireless technologies such as IEEE 802.11
†These procedures are available in https://github.com/ECN-CL/ECN-CL including usage examples.
FIGURE 8 Real and estimated congestion level for the basic functionality experiment

FIGURE 9 A, shows real and estimated congestion level with two types of windows (nonoverlapping window [NW] and sliding window [SW]) for different values of congestion. B, shows the detail of one the transients at t=20s. Congestion values and estimated error are given as % values.

that the estimated congestion follows the real value, although since the inter-arrival time between packets is random, the delay when updating congestion during transitions varies with time.

5.2 | Numerical simulations

In order to provide some more insight of the performance of the proposed mechanism, a simulator is implemented in Matlab. The simulator has three main elements:

1. Traffic generator. Packet inter-arrival time ($\delta[n]$) is simulated with a Pareto distribution. Being $C$ (in bps) the BH capacity, $L$ the packet length (in bits), and $\alpha$ the shape parameter of the Pareto distribution, the mean packet inter-arrival time is $\bar{\delta} = \frac{\alpha}{\alpha^2 - 1} \cdot \frac{1}{\bar{r}}$, with $\bar{r} = \frac{n \cdot C}{100 \cdot L}$ the desired mean packet rate.

2. ECN writer. Only the bit $b_1$ is considered. Equation (3) is applied at the transmitter. An NW and an SW are used and compared. In the last case, congestion is updated each time a packet arrives.

3. ECN reader. The HNB reads the bit $b_1$ in each packet and generates an estimate of congestion using Equation (2).

Two experiments are run. First, a set of packets is generated during 100 seconds with congestion level incrementing in steps of 10%, capacity $C = 10$ Mbps and $L = 1400$ bytes. A window length $N_w = 20$ is used. Figure 9 shows the result of the estimation process.

Since $N_w$ is high, the congestion levels are accurately notified and mean square error (MSE) values are zero for most time instants. However, as it is shown in detail in Figure 9B, during the transients, the delay in the response produces error peaks. As it is illustrated, when NW is used, the delay in estimating the congestion becomes higher since the receiver must wait until the end of the window. This problem is partially solved by using SW.

In the second experiment, 10 periods of 200 ms are generated, each one with a constant random congestion level. Window lengths of 5, 10, and 20 packets are tested with both NW and SW; 1000 repetitions of the experiment are run. The cumulative distribution function (CDF) of the relative mean square error ($e$) and the updating delay ($\Delta$) are represented...
Δ is computed as the time $\hat{\Delta}$ needed to reach $\Delta$ when there is a variation in $\Delta$. As expected, $\Delta$ is lower for shorter windows. Since error happens mainly at transitions, shorter windows also provide more accurate results.

Equivalent results are shown for different $C$ and $N_w$ in Figure 11. As expected, Figure 11A shows that the delay and mean squared error increases for higher values of $N_w$. This is caused because with the lowest value $N_w = 10$, the congestion levels used in the experiment (10% and 90%) can be exactly represented. This means that the minimum window length that provides enough accuracy should always be used. Also, as can be seen in Figure 11B, mean squared error and mean delay decreases when congestion increases. Obviously, this is because with higher rates, the number of received packet (with congestion information in the ECN bit) per second in the HNB is higher, and hence, the response to congestion level is faster updated. Since tests have been performed for a capacity of 10 Mbps, it is expected that for high capacity BH networks, delay and error will decrease significantly.

6 | CONCLUSION

In this work, we have presented a low complexity method to notify the BH congestion to the AN, conceived for rural cellular networks that use a multi-hop IPSec wireless network as BH. The proposal is based on a nonstandard local usage of the ECN bits between the edge node and the HNB, which does not compromise the standard usage of ECN bits in the end-to-end transport connection. The method allows to inform the HNB about the congestion level in the BH, and even about the existence of congestion in a not-managed part of the BH if this part of the network is secure. The proposed solution is focused on BH networks such as multi-hop wireless networks that serve 3G/4G small cells, where the resources are limited and variable, and where SDN techniques are not possible. In these scenarios, the AN has to perform access
control and scheduling procedures, which are dependent of the BH congestion level. Our solution aims to solve this issue in an easy and efficient manner.

In order to validate the approach, two functions, ECN read/write, have been implemented and tested in a testbed network. Also, extensive simulations have been performed to test the solution for a wide range of configurations. Results show on one hand that the solution is simple, lightweight, and compatible with previous ECN IPSec standards, making it suitable to be implemented in a wide variety of devices. Second, the results for different configurations show that using small SWs provide significant performance, with delays lower than 10 ms and accuracies of 2% during 80% of the time. These figures make the solution suitable for admission control and scheduling intervals in current 3G/4G mobile technologies. The integration of this ECN-based approach into future hybrid SDN network designs remains as an interesting future work direction.

ACKNOWLEDGMENTS
Part of the research in this work was founded by the European Community FP7 project TUCAN3G IST-601102 STP.

ORCID
Esteban Municio https://orcid.org/0000-0002-9865-2538
Javier Simó-Reigadas https://orcid.org/0000-0002-4550-0506

REFERENCES
1. Baig R, Roca R, Navarro L, Freitag F. Guifi. Net: a network infrastructure commons. In: Proceedings of the Seventh International Conference on Information and Communication Technologies and Development. Singapore: ACM; 2015:27.
2. Adeyeye M, Gardner-Stephen P. The Village Telco project: A reliable and practical wireless mesh telephony infrastructure. EURASIP J Wirel Commun Netw. 2011;2011(1):78.
3. Rhizomatica. http://rhizomatica.org/ [Online; accessed 15-Aug-2017]; 2015.
4. Martínez-Fernández A, Vidal J, Simo-Reigadas J, et al. The TUCAN3G project: wireless technologies for isolated rural communities in developing countries based on 3G small cell deployments. IEEE Commun Mag. 2016;54(7):36-43.
5. Heimerl K, Hasan S, Ali K, Brewer E, Parikh T. Local, sustainable, small-scale cellular networks. In: Proceedings of the Sixth International Conference on Information and Communication Technologies and Development: Full Papers-Volume 1. ACM; 2013; Cape Town, South Africa:2-12.
6. Hwang I, Song B, Soliman SS. A holistic view on hyper-dense heterogeneous and small cell networks. IEEE Commun Mag. 2013;51(6):20-27.
7. Razavi R, Clausen H. Urban small cell deployments: impact on the network energy consumption. In: Wireless Communications and Networking Conference Workshops (WCNCW). IEEE; 2012; Paris, France:47-52.
8. Jungnickel V, Manolakis K, Zirwas W, et al. The role of small cells, coordinated multipoint, and massive mimo in 5G. IEEE Commun Mag. 2014;52(5):44-51.
9. Baillargeon S, Johansson I. Conex lite for mobile networks. In: Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop, CSWS '14. ACM; 2014; New York, NY, USA:33-38. http://doi.acm.org/10.1145/2630088.2630091
10. Simó-Reigadas J, Figuera C, Morgado E, Municio E, Martinez-Fernández A. Assessing IEEE 802.11 and IEEE 802.16 as backhauling technologies for 3G small cells in rural areas of developing countries. Mobile Information Systems. 2019;2019:Article ID 4383945, 15 pages.
11. Simo-Reigadas J, Municio E, Morgado E, et al. Sharing low-cost wireless infrastructures with telecommunications operators to bring 3G services to rural communities. Comput Netw. 2015;93:245-259.
12. Rey-Moreno C, Bebea-Gonzalez I, Foche-Perez I, Quispe-Tacas R, Liñán-Benitez L, Simo-Reigadas J. A telemedicine wifi network optimized for long distances in the amazonian jungle of Peru. In: Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition. ACM; 2011; Manaos, Brazil:9.
13. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UTRAN Iu interface data transport and transport signaling. 3GPP TS 25414; 2012.
14. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Iuh interface data transport (Release 13). 3GPP TS 25444; 2015.
15. Kent S, Seo K. Security architecture for the Internet protocol; 2005.
16. Casado M, Koponen T, Shenker S, Tootoonchian A. Fabric: a retrospective on evolving SDN. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks. Helsinki, Finland: ACM; 2012:85-90.
17. Jain S, Kumar A, Mandal S, et al. B4: Experience with a globally-deployed software defined WAN. ACM SIGCOMM Comput Commun Rev. 2013;43(3):3-14.
18. Yassine A, Rahimi H, Shirmohammadi S. Software defined network traffic measurement: current trends and challenges. IEEE Instrum Meas Mag. 2015;18(2):42-50.
19. Lara A, Kolasa N, Ramamurthy B. Network innovation using OpenFlow: a survey. IEEE Commun Surv Tutorials. 2014;16(1):493-512.
20. Samarakoon S, Bennis M, Saad W, Latva-aho M. Backhaul-aware interference management in the uplink of wireless small cell networks. IEEE Trans Wirel Commun. 2013;12(11):5813-5825.
21. Ge X, Tu S, Han T, Li Q, Mao G. Energy efficiency of small cell backhaul networks based on Gauss–Markov mobile models. IET Netw. 2015;4(2):158-167.
22. Bojic D, Sasaki E, Cvijetic N, et al. Advanced wireless and optical technologies for small-cell mobile backhaul with dynamic software-defined management. IEEE Commun Mag. 2013;51(9):86-93.
23. Cui Z, Adve R. Joint user association and resource allocation in small cell networks with backhaul constraints. In: 2014 48th Annual Conference on Information Sciences and Systems (CISS). IEEE; 2014; Princeton, NJ, USA:1-6.
24. Marsch P, Fettweis G. Rate region of the multi-cell multiple access channel under backhaul and latency constraints. In: 2008 IEEE Wireless Communications and Networking Conference. IEEE; 2008:830-834.
25. Costa-Requena J. SDN integration in LTE mobile backhaul networks. In: The International Conference on Information Networking 2014 (ICOIN2014). IEEE; 2014; Phuket, Thailand:264-269.
26. Shantharama P, Thyagaturu AS, Karakoc N, Ferrari L, Reisslein M, Scaglione A. Layback: SDN management of multi-access edge computing (MEC) for network access services and radio resource sharing. IEEE Access. 2018;6:57545-57561.
27. Amin R, Reisslein M, Shah N. Hybrid SDN networks: a survey of existing approaches. IEEE Commun Surv Tutorials. 2018;20(4):3259-3306.
28. Huang X, Cheng S, Cao K, Cong P, Wei T, Hu S. A survey of deployment solutions and optimization strategies for hybrid SDN networks. IEEE Commun Surv Tutorials. 2018;21(2):1483-1507.
29. Tajiki MM, Akbari B, Mokari N, Chiaraaviglio L. SDN-based resource allocation in MPLS networks: a hybrid approach. Concurrency Comput: Pract Exper. 2019;31(8):e4728.
30. Wijesinha AL, Song Y-t, Krishnan M, Mathur V, Ahn J, Shyamasundar V. Throughput measurement for UDP traffic in an IEEE 802.11g WLAN. In: Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network. IEEE; 2005; Towson, MD, USA:220-225.
31. Hubert B, Graf T, Maxwell G, et al. Linux advanced routing & traffic control; 2002.
32. Lucente P. pmacct: steps forward interface counters. Tech Rep; 2008.
33. Rattaro C, Belzarena P. Throughput prediction in wireless networks using statistical learning. In: LAWDN - Latin-American Workshop on Dynamic Networks. Buenos Aires, Argentina; 2010-4.
34. Ramakrishnan K, Floyd S, Black D. The addition of explicit congestion notification (ECN) to IP; 2001.
35. Westerlund M, Johansson I, Perkins C, O’Hanlon P, Carberg K. Explicit congestion notification (ECN) for RTP over UDP; 2012.
36. Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. IEEE/ACM Trans Netw (ToN). 1993;1(4):397-413.
37. Stewart R. Stream control transmission protocol. IETF RFC 4960; 2007.
38. Kohler E, Handley M, Floyd S. Datagram congestion control protocol (DCCP). IETF RFC 4340; 2006.
39. Wing D. Symmetric RTP/RTP control protocol (RTCP); 2007.
40. Ramakrishnan K, Floyd S. A proposal to add explicit congestion notification (ECN) to IP; 1998.
41. Avallone S, Guadagno S, Emma D, Pescape A, Ventre G. D-ITG distributed internet traffic generator. In: First International Conference on the Quantitative Evaluation of Systems (QEST 2004). IEEE; 2004; Enschede, The Netherlands:316-317.
42. Mills DL. Internet time synchronization: the network time protocol. IEEE Trans Commun. 1991;39(10):1482-1493.

How to cite this article: Figuera C, Morgado E, Municio E, Simó-Reigadas J. A low complexity mechanism for congestion notification in rural IPSec-enabled heterogeneous backhaul networks. Int J Commun Syst. 2019;32:e4082. https://doi.org/10.1002/dac.4082