

Mesh Slicing: Improving Robustness of A Mesh Network via Multiple Slices

Yilong Deng
the University of Tokyo
yilongdeng@gmail.com

Akihiro Nakao
the University of Tokyo
nakao@iii.u-tokyo.ac.jp

ABSTRACT

Wireless mesh networking has recently become a viable solution for networking for the environment where it is hard to prepare for networking infrastructure. However, it is well known that packet loss increases exponentially as the size of the network grows, and thus, it is often not robust enough in practice. In this paper, we present *mesh slicing*, a technique for making a wireless mesh network (WMN) more robust against high packet loss by duplicating traffic over multiple virtual wireless mesh networks that run different routing algorithms. We introduce the design of *Mesh slicing* and evaluate using ns-3 simulator to show our proposed technique can successfully make a given WMN more robust against high packet loss situation.

1. INTRODUCTION

Wireless mesh networking has been originally developed for military applications, but has recently become a viable solution to networking where it is hard to build networking infrastructure. Its characteristics of self-forming, self-healing, self-organization is also fit for the extreme circumstances (for example, natural disasters). But the packet loss rate increases exponentially as the number of hops to travel increases.

Theoretically, suppose the loss rate at each hop be p , if a route includes n hops, then the loss rate will be $P_{total} = 1 - (1 - p)^n$. To be more concrete, suppose $p = 4\%$, which is typical for 802.11n network, then the loss rate for paths with 10 hops will be 33.5%. The most naive way to survive in the environment with high packet loss is to retransmit packets repeatedly. If we transmit the same packet for r times, then the loss rate goes down to $P'_{total} = (1 - (1 - p)^n)^r$. In the same setup

as above, if we send the same data twice, then the loss rate goes down to 11.2%, and with three times, loss rate becomes 3.8%. In this way, packet loss decreases significantly. The only down-side is that this scheme must consume more wireless channel resources.

However, although this naive method appears viable theoretically, it does not always work well in reality. For example, a path with high packet loss tends to lose more packets in practice than are predicted theoretically, and thus, the retransmission over the same path is inefficient. Also, once the path is disconnected, there is nothing we can do.

This paper presents the design and evaluation of *mesh slicing*, a practical approach to decrease packet loss. We first virtualize a physical wireless interface into several independent virtual interfaces and then create multiple slices (virtual L2 networks) connecting these virtual interfaces. Each slice runs a different routing algorithm (or the same routing algorithm with slightly different parameters). We then duplicate a packet onto multiple slices toward its destination, but before duplicated packets reach the destination, we merge them so that only the earliest arriving packet may reach there. Each slice may be allocated to a single wireless channel or multiple ones. As the slices run different routing algorithms, routes on different slices also differ from one another, thus, even if a wireless link goes down, chances are good that there exists a slice that has a route to the destination.

2. DESIGN

Figure 1(a) illustrates the overall design of *mesh slicing*. We first virtualize a physical wireless interface into multiple virtual ones. We call these virtual NICs *slicing devices*. A *slicing device*, identified by a *slice id*, is a logical unit for packet transmission. A *weaving device* is a controller-like device that controls the whole process of *mesh slicing*. And it will be explained by example later in this section.

We have to add a packet header for each packet through the *weaving device*, which includes two fields (Figure 1(b)), *slice id* (used for physical interface multiplexing)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM CoNEXT Student Workshop, December 6, 2011, Tokyo, Japan.
Copyright 2011 ACM 978-1-4503-1042-0/11/0012 ...\$10.00.

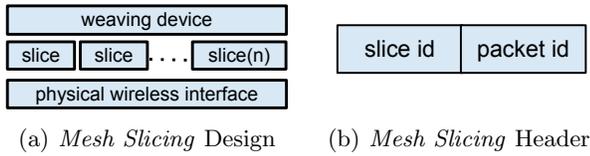


Figure 1: *Mesh Slicing Design*

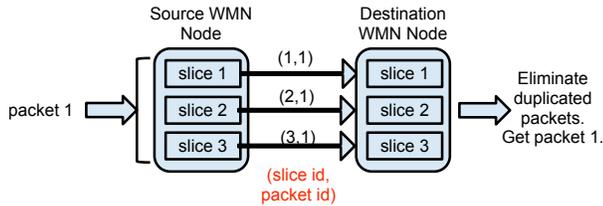


Figure 2: *Packet Flow in Mesh Slicing*

and *packet id* (used for eliminating duplicate packets).

Figure 2 shows a packet flow from a source to a destination. In sending a packet, the *weaving device* first adds a such header to the packet, then assigns *packet id* (sequentially) to it, and duplicates it onto each of the *slicing devices*. Each *slicing device* assigns its own *slice id* to the packet and sends it out. When the packet arrives at the destination, the physical wireless device first demultiplexes it to the corresponding *slicing device* by examining *slice id*, and finally to the *weaving device*. For each flow of packets, the *weaving device* maintains a data structure (for example, sliding-window) to record all received *packet ids*. By checking this data structure, *weaving device* eliminates duplicated packets. As it is for L2 routing, the sliding window can be relatively small and the header can be as short as three or four bytes long. A *slicing device*, in fact, is a routing algorithm container, which can host a specific kind of WMN routing algorithm. In this figure, *slicing device* 1,2 and 3 can each run a different routing algorithm.

3. EVALUATION

In order to evaluate the performance of mesh slicing, we have conducted two simulations with ns-3[1]. We use the 802.11s wireless mesh protocol stack and 802.11a [2] as an underlying physical layer. We construct a typical grid-topology where the mesh nodes construct a grid. As shown in 3(a), with this topology, each pair of nodes may have many possible routes, which is common in the real-world environment. In our evaluation, we set step size (the distance between (i,j) node and (i,j+1) node) to be 70 meters. We evaluate two scenarios in sending packets from the node at (1,1) to the one at (m,n) recording the loss rate as the following formula.

$$loss_rate = packet_received / packet_sent$$

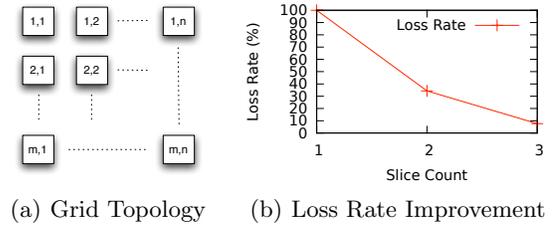


Figure 3: *Mesh Slicing Evaluation*

In the first scenario, we evaluate three cases: (a) sending with the original mesh device, (b) sending with the original mesh device but repeating n times, and (c) sending with mesh-slicing explained in Section 2. We have run the evaluation with $10 * 10$ grid topology 10 times for the duration of 10 minutes, getting the results for the three cases, a) 39%; b) 13.7% (2 times), 10.5% (4 times); c) 6.3% (2 slices), 2.16% (4 slices). From the evaluation result, we find that *mesh slicing* can get lower packet loss with the same channel resources by comparing the results of b) and c), and significantly better than the original scheme, which just sends each packet for once, by comparing a) and c). By sending via different routes, *mesh slicing* tolerates errors in the background and thus ends up with lower packet loss.

In the second scenario, we have also done an evaluation with $15 * 15$ grid topology and have gotten lower packet loss as shown in Figure 3(b). As for the original 802.11s, packets can not even pass through the mesh network, as the loss rate is high. By enabling *mesh slicing*, the loss rate decreases significantly as the number of slices increases. From this result, we conclude that *mesh slicing* can span across a large area that cannot be covered by the existing wireless mesh routing approaches.

4. CONCLUSION AND FUTURE WORK

In this paper, we present *mesh slicing*, a technique for making a WMN more robust against high packet loss. Our evaluation shows that *mesh slicing* can make a given WMN more robust against the situation with high packet loss. In future work, we will implement it in the real-world wireless mesh network and conduct more thorough evaluations.

5. REFERENCES

- [1] ns-3 manual.
- [2] M. Lacage and T. R. Henderson. Yet another network simulator. In *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, WNS2 '06, New York, NY, USA, 2006. ACM.