Multiple-Path Layer-2 based Routing and Load Balancing Approach for Wireless Infrastructure Mesh Networks

[Extended Abstract]

Fabio Feuli
Univ. Rome "Tor Vergata"
fabio.feuli@uniroma2.it

Alessandro Ordine
Univ. Rome "Tor Vergata"
alessandro.ordine@uniroma2.it

Giuseppe Bianchi
Univ. Rome "Tor Vergata"
giuseppe.bianchi@uniroma2.it

ABSTRACT

This work presents a general layer-2 approach for routing and load balancing in Wireless Infrastructure Mesh Network The key idea is dynamic select routes among a set of slowly changing alternative network paths. Our approach decouples the routing and load balancing problem into two distinct sub-problems: path creation and path selection. Paths are created through the reuse of classical 802.1Q multiple spanning tree mechanisms. This guarantees that, for each formed tree, a path is deployed from each mesh node to the Mesh Gateway. Moreover, each tree (path) is assigned a Virtual LAN identifier. Path selection is driven by a local algorithm running at each mesh node, fed by measurements (taken along each path connecting the mesh node to the gateway) which allow to dynamically determine which are the best paths. In order to route a packet it is sufficient to mark the packet with the VLAN tag corresponding to the chosen path. The described approach provides a very general and flexible framework: performance/stability trade-offs can be tuned through the choice of i) the mechanism used to measure the path quality; ii) the algorithm employed to select the path, and iii) the system parameter used (link costs and link weights) for the multiple spanning tree formation.

1. MOTIVATION

The emergence and growth of many companies (such as Tropos Network, BelAir, FireTide, MeshDynamics, etc.) specialized in the provisioning of wireless infrastructure solutions, as well as the recent launch of standardization activities (such as 802.11s), demonstrate that Wireless Infrastructure Mesh Networks (WIMN) may represent a viable and cost-effective alternative to traditional wired infrastructure access networks. Unlike traditional routing algorithms [1] [2] [3] [4], designed for general ad hoc networks, our proposal is specifically devised to benefit from the unique characteristics of WIMN, i.e.: i) static mesh nodes, and ii) traffic mostly addressed from/to backhaul gateways. Furthermore, another motivation underlying our proposal is to provide a layer-2 approach which attempts to reuse as much as possible well established and commercially available 802.1D/Q bridging and access control techniques. This provides a number of assets, including the ease of integration of wired infrastructure segments into a wireless Mesh network infrastructure deployment, and the ability to see the whole network as a single layer-2 802.11 Extended Service Set, thus inheriting the management approaches therein devised (e.g. 802.11f IAPP or the currently under standardization IETF CAPWAP protocol). Moreover, a layer-2 approach is in line with the requirements recently set in the 802.11s Task Group.

CoNEXT 2006 Lisboa, Portugal

2. APPROACH

Layer-2 Ethernet Switched networks rely on spanning tree as forwarding/routing mechanism. However, its adoption in WIMN is questionable. Even if the Mesh topology is static, we expect that link qualities may nevertheless vary in time, especially if they furthermore depend on the relevant traffic load. Indeed, extensive literature paper shows that the routing effectiveness highly benefits from the adoption of dynamic metrics capable to account for both channel quality and load (see e.g. [5] [6]). However, by relying on time-varying link costs, frequent rearrangements of the spanning tree would be required. This is a costly process that causes abrupt interruption of connectivity. Even if the Rapid Spanning Tree protocol is employed, unacceptable impairments are deemed to occur.
Our proposal relies on spanning tree, but in a quite different manner, and consists in i) creating quasi static paths through a spanning tree protocol, and ii) dynamically route and balance the load through appropriate selection of the deployed paths. The next section illustrate into additional details these two sub-problems.

2.1 Multiple Path creation

Rather than using a single time-varying tree, we deploy partially overlapping spanning trees, meant to remain quasi static (i.e. changes occur only upon significant events such as link failures). This is accomplished by slightly adapting, to the WIMN context, the Multiple Spanning Tree Protocol (MSTP) specified in the 802.1Q standard, and, most important, by using the related Virtual LAN (VLAN) identifiers therein assigned and managed, in a significantly different manner. In addition, our solution provides the key advantage that more than 1 backhaul gateway can be managed.
Specifically, let N be the number of gateways. We configure the network as the superposition of $M \geq N$ VLANs: for convenience of presentation, assume (not restrictively) that $M = K \cdot N$, where K is the number of alternative paths to be deployed per each gateway. As well known from 802.1Q, the MSTP protocol deploys one single spanning tree per each VLAN. We configure the MSTP so that each gateway is the root for K trees. Furthermore, through appropriate

per-bridge (i.e. Mesh router) configuration[1] it is possible to differentiate the trees converging to a same gateway in order to avoid that they collapse into a single one tree. The above described operation deploys M static trees covering the whole network. Trees are managed by the MSTP operation in a fully standard manner, thus reconfigured only upon link failures. We underline that this provides each Mesh node with M alternative paths towards the backhaul gateways. Frames may be forwarded along any of this path by simply tagging them with the VLAN identifier corresponding to the selected path (the standard 802.1Q bridging operation guarantees that a VLAN tagged packet is routed accordingly).

2.2 Dynamic Path Selection

First, notice that a Mesh router exerts a very different treatment to frames, depending on the fact that they are locally originated (i.e. incoming from the clients associated to the node), or forwarded (i.e. received by an adjacent Mesh Router). In the latter case, frames have been already VLAN tagged, and their forwarding decision is hence fully determined by the 802.1Q forwarding operation (in turns of course depending on the relevant VLAN tagging). Conversely, locally originated frames are untagged (the end mesh clients are not involved in the MSTP operation). Dynamic path selection is applied only to the local frames: the decision of which path to take is delegated to a locally running path selection mechanism which receives as input the quality of the available paths, and takes the routing decision by tagging the locally received frame with the VLAN identifier of the path.

The approach proposed in this work does not depend on which specific path selection mechanism is used: in terms of network operation, this can be considered as a black box whose output is the choice of the frame VLAN tag. However, it is obvious that the performance of the proposed approach dramatically depend on how this algorithm is designed, and specifically: i) according to which criteria path quality is evaluated, and ii) whether - and how - hysteresis are considered in the path selection procedure[2].

At this stage of this work, we have not yet faced the issue of optimizing the network operation, and we have simply relied (as a proof of concept) on gross heuristics. Specifically, our preliminary simulation study have been obtained with the following technical approaches.

Path quality measurements - Our approach requires to associate a cost to each path. Since several paths are deployed (assuming 100 nodes and 10 trees, as much as 1000 paths should be assigned an explicit costs!), the approach devised to quantify such cost should be scalable and least invasive as possible. This forces us to exclude per-path active approaches where costs are computed through a per-path probing. A more natural approach is therefore to compute the cost associated to a single link, and combine such costs for determining the path cost. It is convenient to combine link costs through an additive process. This allows to incrementally compute the path cost along the tree in a manner identical to what the standard spanning tree protocol BPDU does. To this purpose, we have introduced specially labelled BPDUs which simply use, instead of the fixed link cost used for the spanning tree formation (in our case a same constant value for all the active network links), a dynamically updated link cost. These BPDUs are periodically generated and forwarded as standard STP-BPDUs, as they only differ in their cost field contents and not in their format. Computation and maintenance of such a per-link dynamic cost is delegated to each mesh node. Any proposed link cost metric for which additivity is reasonable (such as the widely adopted Expected Transmission Time [7]) can be used. How to optimize such a metric for the application to our considered framework is a current research issue.

Path selection decision - Our current selection algorithm is based on gross preliminary heuristics, which nevertheless have allowed us to understand some general requirements of the path selection mechanism. First, we noticed that it is important to de-synchronize the mesh node operation, as this would raise severe traffic oscillations. To this purpose, mesh nodes do not base their per-frame tag decision on the bases of the instantaneous path cost (i.e. that carried in the most recently received BPDU). Rather, each node randomly sets a timer, samples the path cost load, and maintain the tag decision until the next timer expires. In addition, the decision whether to change path includes hysteresis thresholds, so that a path is changed only if the difference in cost with respect to the actual chosen path is greater than a given amount. Also in this case, we have found that using the same threshold in all the mesh nodes can be critical. Therefore, hysteresis threshold are also randomly modified when the previously mentioned timer expires. Whether these two mechanisms should be used in conjunction, or one of the two is sufficient (as we believe, if properly tuned settings are provided) is object of current research issue.

[1]Specifically, through appropriate setting of the MSTI priority vectors associated to each Bridge: details are necessarily omitted for reasons of space

[2]It is intuitive that a load balancing mechanism devised to send traffic over the less instantaneously loaded path would suffer, in the best cases, of flapping phenomena, and, in the worst case, of dramatic instability.

3. REFERENCES

[1] C.E. Perkins and P. Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. *Proc. Communications architectures, protocols and applications*, pages 234–244, 1994.

[2] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol. *IEEE INMIC Pakistan*, 2001.

[3] C.E. Perkins et al. *Ad hoc networking.* Addison-Wesley Boston, 2001. Chapter: Ad hoc On-Demand Distance Vector Routing.

[4] D.B. Johnson, D.A. Maltz, and J. Broch. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. *Ad Hoc Networking*, pages 139–172, 2001.

[5] R. Draves, J. Padhye, and B. Zill. Comparison of Routing Metrics for Static Multi-Hop Wireless Networks. *SIGCOMM*, 2004.

[6] B. Awerbuch, D. Holmer, and H. Rubens. High throughput route selection in multi-rate ad hoc wireless networks. *First Working Conference on Wireless On-demand Network Systems (WONS)*, 2004.

[7] R. Draves, J. Padhye, and B. Zill. Routing in Multi-radio, Multi-hop Wireless Mesh Networks. *ACM MobiCom*, 2004.