Correlation of Wireless Link Quality: A Distributed Approach for Computing the Reception Correlation

Stefano Paris* and Antonio Capone†
*Dep. of Inf. Technology and Math. Methods †Dip. di Elettronica e Informazione
University of Bergamo Politecnico di Milano
stefano.paris@unibg.it capone@elet.polimi.it

***Abstract*—The adaptive estimation of packet reception correlation on different wireless links might provide a performance indicator for opportunistic and network coding protocols, leading to the design of novel routing schemes. This paper presents an on-line algorithm to compute the packet reception correlation in a fully distributed fashion. We exploit our algorithm to compute adaptively the κ factor, a new metric proposed recently to capture this correlation on different links. The results obtained on real-life network scenarios show that it accurately estimates the reception correlation of wireless links.**

***Index Terms*—Wireless Measurement Study, Wireless Protocol Design, Experimental Testbed.**

I. Introduction

The broadcast nature of the wireless channel has fostered the design of new data dissemination protocols based on opportunistic and network coding techniques. However, the packet loss correlation on different links can greatly impair their achievable performance, reducing their validity with respect to classical routing protocols based on the shortest path algorithm. Therefore, the knowledge of the inter-link reception correlation permits to select the best alternative between network coding mechanisms and classical approaches.

Recently, the κ factor has been proposed in [1] to capture the correlation of the packet losses affecting two unidirectional wireless links $(t; x)$ and $(t; y)$ that share a common transmitter t. Factor κ is modeled as a function of the correlation coefficient $\rho_{t,x,y}$ of two random variables, x and y, representing the reception or the losses at the two receivers of packets transmitted by the source t (they are therefore Bernoulli distributions). Specifically, $\kappa_{t,x,y}$ is defined according to (1), where the correlation coefficients $\rho_{t,x,y}$, $\rho^{max}_{t,x,y}$ and $\rho^{min}_{t,x,y}$ are defined as expressed in (2). In this latter equation, P_x and P_y represent the packet reception ratios of links $(t; x)$ and $(t; y)$, respectively, while $P^{(t)}_{x,y}(1, 1)$ is the probability that both the receivers of links x and y receive the same packet.

$$\kappa_{t,x,y} = \begin{cases} \frac{\rho_{t,x,y}}{\rho^{max}_{t,x,y}} & \text{if } \rho_{t,x,y} > 0 \\ \frac{-\rho_{t,x,y}}{\rho^{min}_{t,x,y}} & \text{if } \rho_{t,x,y} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

In this paper, we propose a fully distributed and adaptive approach to compute the inter-link reception correlation in wireless networks. Although, the proposed algorithm has been designed to compute the κ factor, its core component (i.e. the estimation of the joint reception probability) can be employed to dynamically estimate any metric proposed recently in the literature to evaluate the inter-link reception correlation, like [2], [3] where the cross-conditional loss probability is used to measure the loss correlation. Furthermore, we observe that the Expected Number of Anypath Transmissions (EATX) metric proposed in [4] can greatly benefit from our estimation technique, since each node can compute the joint delivery probability of its outgoing wireless links without assuming that they are affected by independent losses.

$$\begin{aligned} \rho_{t,x,y} &= \begin{cases} \frac{P^{(t)}_{x,y}(1,1) - P_x \cdot P_y}{\sqrt{P_x \cdot (1-P_x) \cdot P_y \cdot (1-P_y)}} & \sigma_x \cdot \sigma_y \neq 0 \\ 0 & \text{otherwise} \end{cases} \\ \rho^{max}_{t,x,y} &= \frac{min(P_x, P_y) - P_x \cdot P_y}{\sigma_x \cdot \sigma_y} \\ \rho^{min}_{t,x,y} &= \begin{cases} \frac{-P_x \cdot P_y}{\sigma_x \cdot \sigma_y} & P_x + P_y \leq 1 \\ \frac{P_x + P_y - 1 - P_x \cdot P_y}{\sigma_x \cdot \sigma_y} & \sigma_x \cdot \sigma_y \neq 0 \end{cases} \\ \sigma_x &= \sqrt{P_x \cdot (1 - P_x)} \end{aligned} \tag{2}$$

The proposed algorithm has been integrated in the OLSR[1] (Optimized Link State Routing) protocol [5] and its performance has been evaluated using both a virtual scenario and the real-life ORBIT wireless testbed. Experimental results show that the proposed distributed algorithm estimates correctly the packet reception correlation, and thus the κ factor.

The paper is structured as follows: Section II illustrates the proposed solution to estimate the value of κ in a distributed fashion, while Section III illustrates the results obtained testing our solution though simulations and on the ORBIT testbed. Finally, conclusions are presented in Section IV.

II. A Distributed Algorithm for Estimating κ

The proposed distributed algorithm computes the correlation of packet losses affecting wireless links, and thus the κ factor, extending the probe mechanism used by several routing protocols to estimate ETX (Expected Transmission Counter) [6].

More specifically, every node n_i computes the delivery probability P_{x_i} (known also as *link quality*) of the adjacent wireless link $x_i = (t, n_i)$ as the ratio of the received probes $RP(t)$ periodically transmitted by t and those expected over a sliding time window of w seconds, $P_{x_i} = \frac{RP(t)}{w/\tau}$.

The delivery probability on the reverse link $-x_i = (n_i, t)$, P_{-x_i}, which is used by n_i to compute the ETX of the link x_i, is piggybacked by node t within probe messages.

[1] Code available on-line at http://cs.unibg.it/paris/kappa/index.html

In order to inform neighbor nodes about the probe messages received from the same transmitter, their format is extended to contain the sequence number of the last received or expected probe and a bit-mask, whose k^{th} bit represents the reception ($b_k = 1$) or the loss ($b_k = 0$) of the k^{th} probe starting from that identified by the aforementioned sequence number. Thus, a probe message broadcasted by node t contains for each neighbor n_i the following entry $(n_i, P_{-x_i}, s_{-x_i}, bm_{-x_i}, P_{x_i}, s_{x_i}, bm_{x_i})$, where n_i represents the neighbor identifier (i.e., its address), P_{-x_i} and P_{x_i} represent the delivery probabilities of links $-x_i = (n_i, t)$ and $x_i = (t, n_i)$, respectively. The values s_{-x_i} and bm_{-x_i} represent the sequence number of the last received or expected probe and the bit-mask of the received or lost probes periodically transmitted by the neighbor node n_i. Finally, s_{x_i} and bm_{x_i} are the sequence number and the bit-mask of probes received by n_i on the unidirectional link $x_i = (t, n_i)$. Note that these two latter values are obtained from a probe message previously transmitted by n_i on the reverse link $-x_i = (n_i, t)$.

In the example network scenario illustrated in Figure 1, node t embeds in every probe message three tuples representing the wireless links established with its neighbor nodes $\{i, j, k\}$.

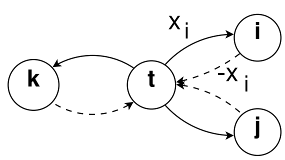


Fig. 1: Network topology advertised by node t. Solid arrows represent the links that are used to compute κ_{t,x_i,x_j}, κ_{t,x_i,x_k} and κ_{t,x_j,x_k}.

Upon receiving a probe message from t, node n_i computes the delivery probability P_{x_i} and updates the local information representing the probes received from t, i.e., s_{x_i} and bm_{-x_i}. Moreover, for all advertised neighbors such that $n_j \neq n_i$, n_i updates the local information representing the statistics of the unidirectional links $x_j = (t, n_j)$ and $-x_j = (n_j, t)$.

At the end of each time window w, each node computes the joint reception probability $P^{(t)}_{x_i,x_j}(1,1)$ that the receiving nodes of links $x_i = (t, n_i)$ and $x_j = (t, n_j)$ received a packet from the same transmitting node t, according to (3). In such Equation, the function $\zeta(bm)$ counts the number of '1' in the bit-mask bm, whereas p^{tx}_{ij} is the number of transmitted probes considered for the computation of $P^{(t)}_{x_i,x_j}(1,1)$. The value of p^{tx}_{ij} is equal to w/τ only if at the end of the time window w the relation $s_i = s_j$ holds.

$$P^{(t)}_{x_i,x_j}(1,1) = \begin{cases} \frac{\zeta(bm_{x_i} \wedge bm_{x_j})}{p^{tx}_{ij}} & s_i = s_j \\ \frac{\zeta(bm'_{x_i} \wedge bm_{x_j})}{p^{tx}_{ij}} & s_i > s_j \\ \frac{\zeta(bm_{x_i} \wedge bm'_{x_j})}{p^{tx}_{ij}} & s_i < s_j \end{cases} \quad (3)$$

Assuming that the last bit of the two bit-masks represents the reception or the loss of the same probe (i.e., when $s_i = s_j$), the number of '1' resulting from the logical AND of the two bit-masks represents the probes that are received by both n_i and n_j. Therefore, the ratio of this latter value and the number of expected probes in a time window w is the joint reception probability of links x_i and x_j.

When the last bit of the two bit-masks does not refer to the same probe (i.e., $s_i \neq s_j$), the bit-masks are realigned reducing the number of bits considered in the logical AND. Let us assume, without loss of generality, that $s_i > s_j$. Then, the bit-mask used in the logical AND, bm'_i, is obtained from the original bit-mask bm_i, right-shifting its bits by the difference between the two sequence numbers, $s_j - s_i$. Moreover, the total number of probes considered in the estimation has to be decreased of the same quantity, $p^{tx}_{ij} = w/\tau - (s_j - s_i)$.

The previous algorithm used to estimate the joint reception probability $P^{(t)}_{x_i,x_j}(1,1)$ can be further refined to update iteratively the estimate of the packet reception correlation, considering all probes transmitted during an interval of duration higher than the time window w. The new mechanism used to estimate the joint reception probability of two unicast wireless links at the end of an estimation interval e is listed in Algorithm 1 on page 2, where $|bm_h|$ is the size of the bit-mask bm_h (e.g., 32 if we use an unsigned integer to represent the received or lost probes). The result provided by the algorithm can be used along with P_{x_i} and P_{x_j} to compute the cross-correlation index ρ_{t,x_i,x_j} and the κ factor κ_{t,x_i,x_j} according to (1) and (2).

Algorithm 1: Joint Reception Probability Estimation

input : $bm_i(e), s_i(e), bm_j(e), s_j(e), s_i(e-1), s_j(e-1), p^{tx}_{ij}, p^{rx}_{ij}$
output: $P^{(t)}_{x_i,x_j}(1,1)$
$s_0(e) = min\{s_i(e-1), s_j(e-1)\}$;
foreach $h \in \{i, j\}$ **do**
  $bm'_h = CircularRightShift(bm_h(e), (s_h(e) - s_0(e)))$;
  $bm'_h = RightShift(bm'_h, |bm_h| - (s_h(e) - s_0(e)))$;
  $bm'_h = RightShift(bm'_h, min\{s_i(e), s_j(e)\})$;
end
$p^{tx}_{ij} = p^{tx}_{ij} + (min\{s_i(e), s_j(e)\} - s_0(e))$;
$p^{rx}_{ij} = p^{rx}_{ij} + \zeta(bm'_i \wedge bm'_j)$;
$P^{(t)}_{x_i,x_j}(1,1) = \frac{p^{rx}_{ij}}{p^{tx}_{ij}}$;

We observe that the proposed estimator of the joint reception probability defined in (3) and computed by Algorithm 1 on page 2, is not biased, since it is obtained by applying the joint reception probability definition.

Theorem II.1. *Algorithm 1 executed for each tuple (t, x_i, x_j) has time and space complexity $O(n^3)$, where n is the number of network nodes.*

Proof: In a completely connected network, each transmitting node can establish at most $n - 1$ unicast links. The pairs of unicast links for each network node are $\frac{(n-1)\cdot(n-2)}{2}$. Therefore, the overall number of link pairs in the network that are considered by the proposed algorithm (i.e., with the same transmitting node) is at most $\frac{n\cdot(n-1)\cdot(n-2)}{2}$. ■

We underline that the proposed on-line algorithm reaches the lowest computational complexity necessary to compute the κ factor ($O(n^3)$) in a distributed fashion. Therefore, the proposed solution is optimal in this regard.

To illustrate how the joint reception probability is evaluated at the end of an estimation interval e, let us refer to the example scenario shown in Figure 2. The circular right shift and the first right shift eliminate the '1' representing the reception of probes already considered in previous estimation intervals. The second right shift synchronizes the bit-masks of links x_i and x_j, deleting the last two bits of the link x_i that correspond to the 92^{nd} and 93^{rd} probes.

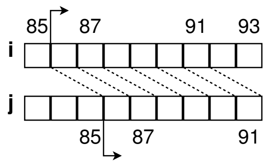


Fig. 2: Joint reception probability estimation. The sequence numbers of the last received probes are $s_i(e) = 93$ and $s_i(e) = 91$ for links i and j, respectively. The value $s_0(e) = 85$ represents the minimum sequence number of the previous interval.

The scheme described above needs to be further refined to allow nodes farther away than 2 hop to compute the κ factor. To this end, we require that each node periodically broadcasts to the entire network its adjacent links with the necessary information to compute the κ factor. Specifically, the transmitting node t broadcasts for each neighbor n_i the following tuple $(n_i, P_{-x_i}, s_{-x_i}, bm_{-x_i}, P_{x_i}, s_{x_i}, bm_{x_i})$.

Note that the proposed algorithm can be easily integrated in link state routing protocols like OLSR, since the information necessary to estimate the κ factor can be embedded into HELLO and TC (Topology Control) messages that are periodically transmitted by network nodes to update the topology.

III. Experimental Study

In order to evaluate the effectiveness of the proposed solution, we implemented the proposed estimation algorithm in *olsrd* [7] as a loadable plug-in, and evaluated its performance using both a virtualized scenario and the wireless mesh network testbed developed under User Mode Linux and the ORBIT project [8], respectively.

The virtualized scenario was composed of three machines connected to the same virtual collision domain. The code enabling the transmission among virtual machines was modified to simulate pairs of links on which packet receptions of broadcast transmissions are positively correlated (i.e., packets received on the best link are necessarily received also on the other link of the pair, $\kappa = 1$), negatively correlated (i.e., the packet reception on a link corresponds to a packet loss on the other link of the pair, $\kappa = -1$) and uncorrelated ($\kappa = 0$).

Table I shows the κ factor of the three link pairs computed by both a centralized and our distributed approaches. As illustrated in the table, the proposed distributed approach accurately estimates the κ value of each link pair. The low estimation error incurred by our algorithm is caused by the message buffering performed by *olsrd* and the lower transmission rate of TC messages that introduce latency in the distributed estimation process.

The wireless mesh network testbed was composed of 12 nodes grouped in three subsets configured to use orthogonal wireless channels. The maximum transmission rate at the data-link layer was fixed to 6 Mb/s.

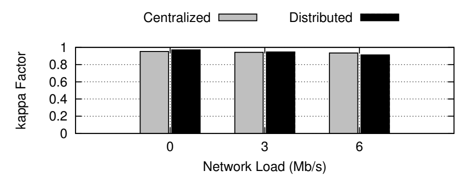
TABLE I: κ factor measured in the virtualized scenario.

Real	Centralized	Distributed
-1	-1	-0.96
0	0	0
1	1	1

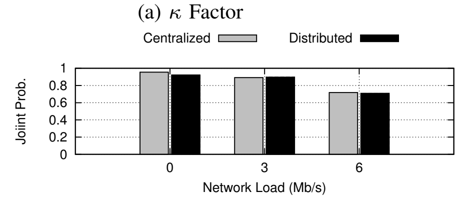
To evaluate the effect of the traffic load on the algorithm accuracy, we establish two CBR (Constant Bit Rate) connections between two pair of nodes, whose radio interfaces were set on two of the three channels used by network nodes.

Figure 3(a) and 3(b) illustrate the mean values of the κ factor and the joint reception probability as a function of the network load imposed by the CBR connections.

In addition to confirming the high accuracy of the proposed distributed algorithm, these results show that our solution is not affected by the network traffic imposed by data connections; thus representing a feasible and effective solution for estimating adaptively the κ factor in wireless multi-hop network scenarios.



(a) κ Factor



(b) Joint Probability

Fig. 3: Performance of the proposed algorithm on the ORBIT testbed.

IV. Conclusion

In this paper we propose a distributed approach for computing adaptively the cross-correlation of packet losses affecting unicast wireless links. Experimental results show that the proposed algorithm accurately captures such correlation by estimating correctly the κ factor of all links pairs.

References

[1] K. Srinivasan, M. Jain, J.I. Choi, T. Azim, E.S. Kim, P. Levis, and B. Krishnamachari. The κ factor: inferring protocol performance using inter-link reception correlation. *ACM MobiCom*, 2010.

[2] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based Models of Delivery and Interference in Static Wireless Networks. *ACM SIGCOMM*, 2006.

[3] A. Miu, G. Tan, H. Balakrishnan, and J. Apostolopoulos. Divert: Fine-grained Path Selection for Wireless LANs. *ACM MobiSys*, 2004.

[4] R. Laufer, H. Dubois-Ferriere, and L. Kleinrock. Multirate Anypath Routing in Wireless Mesh Networks. *IEEE INFOCOM*, 2009.

[5] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR) RFC 3626. *http://www.ietf.org/rfc/rfc3626.txt*, 2003.

[6] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. *ACM Comp. Comm. Review*, 2004.

[7] olsrd: Ad hoc wireless mesh routing. *Available at http://www.olsr.org*.

[8] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, and K. Ramachandran et al. Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols. *IEEE WCNC*, 2005.