AMD: Audit-based Misbehavior Detection in Wireless Ad Hoc Networks

Yu Zhang, Loukas Lazos, *Member, IEEE,* and William Jr. Kozma
Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, 85721
E-mail:{yuzhang, llazos, kozma}@ece.arizona.edu

APPENDIX A

In order to determine whether the set of packets dropped by a node are due to misbehavior, we must specify the values of $w_1, w_2, \ldots, w_K$ and γ_0, which are used for the computation of δ in (4). The value of γ_0 is determined by the physical layer parameters of the communication. Taking into account the channel model (AWGN, Raleigh, etc.), the modulation type (BPSK, QPSK, QAM, etc.), the channel coding technique and the application of any error correction, an expected value of the packet error rate (PER) of a benign node is derived. Assuming that packets of all types are uniformly dropped due to pool channel conditions for any distribution of weights $w_1, w_2, \ldots, w_K$, one can compute the threshold γ_0 as,

$$\gamma_0 = \frac{\sum_{i=1}^{K} w_i \text{PER}_i X_i}{\sum_{i=1}^{K} w_i X_i} = \text{PER}. \quad (1)$$

Formula 1 can be adjusted to take into account different PER values for different packet types. Such differences can occur due to the varying packet sizes, varying channel coding techniques and the use of different transmission rates. As an example, in 802.11 networks, control packets are always transmitted at the lowest data rate for reliability purposes while data packets can be transmitted at a higher rate, and therefore, experience a higher PER.

The values of $w_1, w_2, \ldots, w_K$ are application dependent. The source can assign large values to packet types that are deemed to be highly important for the application at hand. For instance, control packets are typically of higher importance. One method for determining $w_1, w_2, \ldots, w_K$ is by setting a condition under which misbehavior is always detected. For instance consider only two types of packets (e.g., control packets and data packets). Assume that we want to always detect a misbehaving node that drops a fraction $\frac{Y_1}{X_1} > f_0$ packets. Then w_1 and w_2 are determined as a feasible solution region derived from the following inequalities,

$$\delta < \gamma_0, \quad (2)$$
$$\sum_i w_i = 1, \quad (3)$$
$$\frac{Y_1}{X_1} > f_0. \quad (4)$$

In the case of two packet types,

$$\frac{w_1 Y_1 + w_2 Y_2}{w_1 X_1 + w_2 X_2} < \gamma_0, \quad (5)$$
$$w_1 + w_2 = 1, \quad (6)$$
$$\frac{Y_1}{X_1} > f_0. \quad (7)$$

Solving for w_1 and w_2 yields the desired weight distribution.

APPENDIX B

Proposition 1: When $|M| = 1$, the source converges to the misbehaving link in at most $\lceil \log_2(k) \rceil + 1$ audits, without reputation information and $(k-1)$ audits when information reputation is available.

Proof: Let $|M| = 1$ with n_ω being the misbehaving node. Initially, $\mathcal{V} = P_{SD}$ and hence $n_\omega \in \mathcal{V}$. Let the source select node n_i upstream of n_ω for audit. Since $|M| = 1$, n_i responds honestly that it forwarded packets to the next hop ($a_i = 1$), reducing $\mathcal{V}$ to $\{n_i, \ldots, n_k\}$, with $n_\omega \in \mathcal{V}$. Similarly, if a node n_j downstream of n_ω is audited, it will respond that no packets were received and forwarded ($a_j = 0$), reducing $\mathcal{V}$ to $\{n_1, \ldots, n_j\}$. If n_ω is audited, its response will indicate that misbehavior occurs either upstream of downstream, depending on whether $a_\omega = 1$ or $a_\omega = 0$. In either case, n_ω remains in $\mathcal{V}$, since the a node that is being audited cannot exclude itself from $\mathcal{V}$. The convergence of the binary search will end in either suspicious set $\mathcal{V} = \{n_{\omega-1}, n_\omega\}$ or $\mathcal{V} = \{n_\omega, n_{\omega+1}\}$, depending on whether $a_\omega = 0$ or $a_\omega = 1$, i.e., n_ω indicated that misbehavior occurs upstream or downstream. This is true since If n_ω lied in its audit reply indicating that misbehavior occurs downstream of n_ω, the set of suspicious nodes will be reduced to $\mathcal{V} = \{n_\omega, \ldots, n_k\}$. Every node in $\mathcal{V}$ besides n_ω is honest ($|M|$=1) and hence, when audited, will indicate that misbehavior occurs upstream. Therefore, the search will converge on $(n_\omega, n_{\omega+1})$. Likewise, if n_ω claims it did not receive and forward any packets to the next hop, implying that misbehavior occurs upstream of n_ω, the search will converge on link $(n_{\omega-1}, n_\omega)$. In any case, the identified link is a misbehaving one since per the definition its two incident nodes provide conflicting audit replies.

When all nodes have the same reputation value, the search strategy is converted to a binary search process which is known to converge in at most $\lceil \log_2(k) \rceil$ steps. One more audit is needed to simultaneously audit the nodes which are incident on the misbehaving link, yielding a total number of audits equal to

$\lceil \log_2(k) \rceil + 1$. When all nodes have different reputation values, the source audits the node preceding the one with the smallest reputation value. The worst case scenario is realized when nodes along the path are arranged in descending reputation order, i.e., $r_S^i > r_S^j$ when $i < j$, and $n_\omega = n_1$ (or when $r_S^i < r_S^j$ $i > j$ and $n_\omega = n_k$). In this case, the source will audit nodes $n_{k-1}, n_{k-2}, \ldots, n_2$ in this order before converging to misbehaving link (n_1, n_2). Upon convergence, the source will simultaneously audit (n_1, n_2), yielding a worst case number of audits equal to $(k-1)$. Note that although the worst-case number of audits needed by the source when reputation information is taken into account is larger than the worst-case of a binary search, the former method is faster on average due to the low reputation values assigned to misbehaving nodes. □

APPENDIX C

Proposition 2: When $|M| = 1$, the source converges to the misbehaving in at most $4\lceil \log_2(k) \rceil + 2$ audits.

Proof: Let $|M| = 1$ with n_ω being the misbehaving node. Initially, $\mathcal{V} = P_{SD}$ and hence $n_\omega \in \mathcal{V}$. Consider the source to be at stage $\mathcal{V}_j = \{n_i, \ldots, n_k\}$ with $n_\omega \in \mathcal{V}_j$ and select node n_h for audit, creating membership sets $A = \{n_i, \ldots, n_h\}$ and $B = \{n_{h+1}, \ldots, n_k\}$. If $n_\omega \neq n_i, n_h, n_{h+1}, n_k$, then all audit responses will be honest and the source will conclude either $n_\omega \in A$ or $n_\omega \in B$, thus proceeding to the next stage with $\mathcal{V}_{j+1} = A$ or $\mathcal{V}_{j+1} = B$ and $n_\omega \in \mathcal{V}_{j+1}$. As long as the source audits honest nodes, the set of suspicious nodes V_j will be cut by half, thus converging to $|V_j| = 2$.

Now assume one of the n_i, n_h, n_{h+1}, n_k is n_ω. When audited, n_ω will either respond honestly, or lie. If n_ω responds honestly, this is equivalent to both audited nodes being honest and the search will proceed to state $\mathcal{V}_{j+1}$ with $n_\omega \in \mathcal{V}_{j+1}$ and $|\mathcal{V}_{j+1}| = \frac{|\mathcal{V}_j|}{2}$. Thus the search continues to converge. If n_ω lies, the source will obtain negative answers from both membership questions. Hence, the source will be unable to reduce $\mathcal{V}_j$ further and will return to stage $\mathcal{V}_{j-1}$ with $n_\omega \in \mathcal{V}_{j-1}$. The source will then pick a different n_h, and repeat the set splitting and auditing, thus preventing the same lie from repeating. Since there is only one misbehaving node, and each node is audited at most once, the auditing strategy will converge to $|\mathcal{V}_j| = 2$, with n_ω in $\mathcal{V}_j$.

For computing the number of steps needed for the convergence to the misbehaving link, assume the worst case and let each stage always require two membership questions, i.e., "Is $n_\omega \in A$?" and "Is $n_\omega \in B$?" In the absence of lies, the total number of membership questions needed is $2\lceil \log_2(k) \rceil$. This is true, since at each stage we split the suspicious set to half similar to a binary search. To realize a membership question we need to simultaneously audit two nodes, requiring a total of $4 \log_2(k)$ audits in the worst case (assume no lies). If n_ω is audited and lies, the search backtracks to the previous stage, resulting in the waste of two audits. Given that $|M| = 1$ and the fact that the source always selects a different node to audit after a backtrack, the misbehaving node will be audited at most once. Thus, in the worst case, the source requires at most $4\lceil \log_2(k) \rceil + 2$ audits. Note that to realize membership questions audits occur simultaneously in pairs. Hence, the worst-case delay for converging to the misbehaving node is $2\lceil \log_2(k) \rceil + 1$ □

APPENDIX D

Corollary 1: The source can never converge to a link containing two behaving nodes.

Proof: According to the MEM algorithm, the source must receive conflicting reports from two simultaneously audited nodes to proceed from stage $j-1$ to stage j. Hence, for terminating to a set $\mathcal{V}_j = \{n_i, n_{i+1}\}$ the source must receive conflicting audit replies from n_i, n_{i+1}, when simultaneously audited. However, as shown in Corollary 1, this cannot occur if both n_i, n_{i+1} are behaving nodes. □