ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 4 April 2015, Page No. 11148-11152

Hop Voting scheme with Voting tag Verification Scheme for wireless Networks

Dr. M. Senthamil Selvi[1],Adhars Ram B[2], Ashok K[3], Babu V[4]

[1]Prof. /Head-Information Technology,
Sri Ramakrishna Engineering College, Coimbatore-22
Email: hod-it@srec.ac.in
[2,3,4]Final B.Tech- Information Technology,
Sri Ramakrishna Engineering College, Coimbatore-22
Email: aadarshram4@gmail.com, k.ashok129@gmail.com, babu030694@gmail.com

Abstract: ***Network coding is a process where an intermediate node encodes incoming packets before forwarding. In existing system, they decouple the routing and scheduling components of the algorithm by designing a probabilistic routing table that is used to route packets to per-destination queues. The back-pressure algorithm, while being throughput-optimal, is not useful in practice for adaptive routing since the delay performance can be really bad. However, using this algorithm will lead to deadlock, power consumption and less security. We proposed a system that implements the novel HOP VOTE scheme along with tag based authentication scheme for effective pollution attack detection recovery and blocking. This allows a node to verify if it's received packet belong to specific rule criteria, even if the encrypted key is expanding over a time. Based on the id and tag creation the data will be routed. Using tag encoding scheme, the communication overhead can be routed. It is fast and reliable. Further, it improves security, throughput and efficiency of the system.***

Keywords: Tag encoding scheme, Optimal link state routing protocol, OSPF.

1. Introduction

In wireless networks, computers are connected and communicate with each other not by a visible Medium, but by transmissions of electromagnetic energy in the air. The most widely used transmission support is radio waves. The design abstracts the wireless channel as a point-to-point link, and grafts wired network protocols onto the wireless environments. For example, routing uses shortest path protocols, routers forward packets but do not modify the data, and reliability relies on retransmissions. The design has worked well for wired networks, but less so for the unreliable and unpredictable wireless medium. The wireless medium is fundamentally different. While wired networks have reliable and predictable links, wireless links have high bit error rate, and their characteristics could vary over short time-scales. Further, wired links are unicast links, but the majority of wireless links (with Omni-directional antennas) are broadcast links. Transmissions in a wired network do not interfere with each other, whereas interference is a common case for the wireless medium. Wired nodes are usually static, while wireless was built to support mobility and portability. The wired network design conflicts with the characteristics of the wireless medium. As a result, current wireless networks suffer low throughput, dead spots, and inadequate mobility support. The characteristics of wireless networks might all seem disadvantageous at first sight, but a newer perspective reveals that some of them can be used to our advantage, albeit with a fresh design. The broadcast nature of wireless provides an opportunity to deal with unreliability; when a node broadcasts a packet, it is likely that at least one nearby node receives it, which can then function as the next-hop and forward the packet. This is in stark contrast to the present wireless design, where there is a single designated next-hop, and when it does not receive the packet, the previous hop has to retransmit it

2. Network Model

We consider a multihop wireline or wireless network represented by a directed graph, where the set of nodes is and is the set of directed links. A directed link that can transmit packets from node to node is denoted by (n,j) € L[1]. We assume that time is slotted and define the link capacity to be the maximum number of packets that link can transmit in one time-slot. Let be the set of flows that share the network. Each flow is associated with a source node and a destination node, but no route is specified between these nodes. This

means that the route can be quite different for packets of the same flow. Let and be source and destination nodes, respectively, of flow. Let be the rate (packets) at which packets are generated by flow. If the demand on the network, i.e., the set of flow rates, can be satisfied by the available capacity, there must exist a routing algorithm and a scheduling algorithm such that the link rates lie in the capacity region.

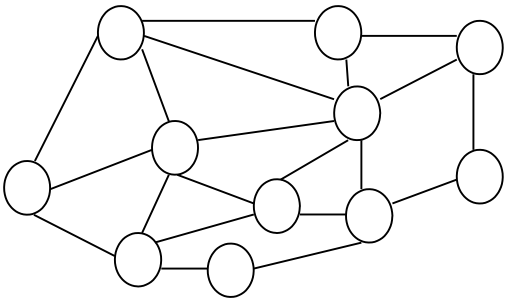

Figure 1: Wireless network topology with 11 nodes

3. Implementation Details

3.1. Network structure analysis

Client-server computing or networking is a distributed application architecture that partitions tasks or workloads between service provider's servers and service requesters, called clients. Often clients and servers operate over a computer network on separate hardware. A server machine is a high-performance host that is running one or more server programs which share its resources with clients [7]. A client also shares any of its resources; Clients therefore initiate communication sessions with servers which await (listen to) incoming requests. The followings are the parameters to construct a network.

1. Node Name
2. Host Number
3. IP Address

The node name is nothing but the system name, which can be given by the user. The next value is host number which can be get from our network configuration details. The next one is the IP address of the system. These can be identified by a simple command on DOS environment. The command 'netstat' helps to get all details about the network configuration. Topology is constructed by getting the names of the nodes and the connections among the nodes as input from the user. While getting each of the nodes, their associated port and ip address is also obtained. For successive nodes, the node to which it should be connected is also accepted from the user. While adding nodes, comparison will be done so that there would be no node duplication. Then we identify the source and the destinations.

3.2. Routing Protocol

Open Shortest Path First (OSPF) and Intermediate System-Intermediate System (IS-IS), split traffic evenly over shortest paths based on link weights [3]. Designing a link-state routing protocol has three components. First is weight computation: The network- management system computes a set of link weights through a periodic and centralized Optimization [8]. The second is traffic splitting: Each router uses the link weights to decide traffic- splitting ratios among its outgoing links for every Destination [7]. The third is packet forwarding: Each router independently decides which outgoing link to forward a packet based only on its destination prefix in order to realize the desired traffic splitting.

3.3. Message Transmission

Suppose that a network consists of a source S, some intermediate nodes, and a set R of sinks, and random linear network coding is exploited in the network without the knowledge of its global topology[2]. The source S is to multicast a file of n data blocks $B_1, B_2, \ldots B_n$ to a set R of sinks, where each $B_i (1 \le i \le n)$ is a vector of m dimensions over field F_p. To mark the coefficients of encoded data blocks, before sending B1,B2,….,Bn, a unit vector Ei of length n over field Fp is appended to Bi($1 \le i \le n$), where the ith coordinate of Ei is 1. Set Pi=(Ei,Bi)€ Fpm+n as a data packet that consists of a data block Bi and the encoded coefficient vector Ei. With random linear network coding, an encoded data packet is a linear combination of data packets P1,P2,…..,Pn[5]. Let W be an encoded data packet, then W will be an m+n dimensional vector with the first n coordinates being encoding coefficients.

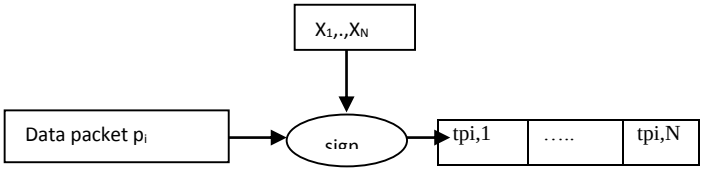

Figure 2: Tag generation at source S

3.3.1. KEPTE

Three Basic Algorithms

Before presenting KEPTE, we give an overview of three basic algorithms, namely Sign, Combine, and Verify[4]. The algorithm Sign computes N tags for each of the n data packets P1; P1,….,Pn, where N is a security parameter and we will analyze the security performance that N tags can provide in the next section. Given multiple data packets, each with N tags, Combine produces N tags for a linear combination of those multiple data packets. Verify checks the correctness of a data packet with its N tags[2]. In KEPTE, the source, intermediate nodes or sinks will perform one or two of those three algorithms: data packets, each with N tags, Combine produces N tags for a linear combination of those multiple data packets. Verify checks the correctness of a data packet with

its N tags[2]. In KEPTE, the source, intermediate nodes or sinks will perform one or two of those three algorithms:

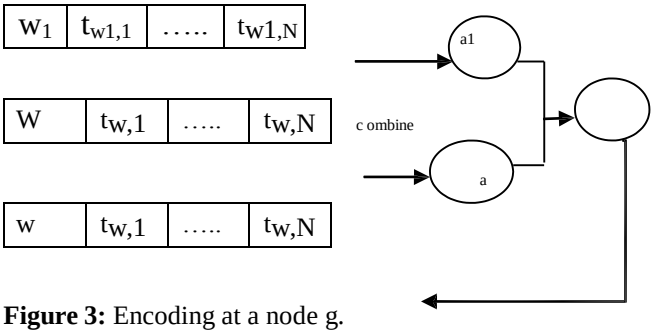

Figure 3: Encoding at a node g.

- Sign($X_1,\ldots\ldots,X_N$, P_i):
  Input: N secret vectors $X_1,\ldots..,X_N \in F_p^{m+n}$, the i^{th} data packet $P_i \in Fp^{m+n}$.

  Output: N tags $t_{Pi,1,\ldots..,}t_{Pi,N} \in Fp$ for Pi, where $t_{Pi,j} = P_i \cdot X^T_j \in F_p(1\leq j\leq N)$

- Combine($(W_i, t_{Wi,1,}\ldots.., t_{Wi,N})^h_{i=1},(\alpha_i)^h_{i=1}$):
  Input: h vectors $W_i \in F_p^{m+n}$ $(1\leq i\leq h)$ and each with N tags $t_{Wi,1,}\ldots\ldots\ldots,t_{Wi,N} \in F_p(1\leq i\leq h)$, h constants $\alpha_1,\ldots..,\alpha_h \in F_p$.

  Output:$(W,t_{W,1,}..,t_{W,N})=\sum^h_{i=1}\alpha_i(w_i,t_{wi},1,\ldots\ldots t_{wi,N}) \in F_p^{m+n+N}$.

- Verify(Z_g,Vg,$(W,t_{W,1,}\ldots\ldots..,t_{W,N})$):
  Input: two secret vectors $Z_g \in F_p^N$ and $V_g \in F_p^{m+n}$, a vector $W \in F_p^{m+n}$ with its N tags $t_{W,1,}\ldots\ldots..,t_{W,N}$.

  Output: If $Z_g \cdot (t_{W,1,}\ldots\ldots,t_{W,N})^T = W.V_g^T$, output 1; otherwise, output 0.

3.3.2. Preventing pollution

In this section, we describe the process of generating tags for $P_1,\ldots\ldots,P_n \in F_p^{m+n}$ at the source S, the encoding of multiple data packets at intermediate nodes, and the correctness verification of data packets at each node g except S in a network. In the following of this paper, we will simply refer to each node g except the source s in a network with each node g :

1. Setup. KDC distributes N secret vectors $X_1,\ldots\ldots,X_N \in F_p^{m+n}$ p to the source S, and distributes two secret vectors $Z_g \in F_p^N$ and $V_g \in F_p^{m+n}$ to each node g, where Zg, Vg, and $X_{,\ldots..,}X_N$ satisfy the following:

$$V_g = Z_g \cdot \begin{pmatrix} X_1 \\ \ldots. \\ X_N \end{pmatrix}$$

2. Tag Generation. For each $P_i \in F_p^{m+}n(1\leq i\leq n)$, the

source uses the algorithm Sign $(X_{1,\ldots..,}X_N,P_i)$ t generate N tags $t_{Pi,1,}\ldots\ldots..,t_{Pi,N}$. Fig. 2 depicts this

process.

3. Encoding. Assume that an intermediate node g

Receives h correct data packets $Wi \in F_p^{m+n}(1\leq i\leq h)$, each with N tags $t_{Wi,1,}\ldots\ldots..,tW_{i,N} \in F_p(1\leq i\leq h)$.For an output link, g randomly selects h constants $\alpha_1,\ldots\ldots,\alpha_h \in F_p$ and performs the algorithm Combine $((W_i ,t_{Wi,1,}\ldots\ldots..,t_{Wi,N})^h_{i=1}, (\alpha_i)^h_{i=1})$to generate

a new encoded data packet W with N tags $t_{W,1,}\ldots\ldots..,t_{W,N}$ as the output of this link. Fig. 3 depicts this process.

4. Verification. Upon receiving a data packet W with its N tags $t_{W,1},\ldots\ldots,t_{W,N}$, a node g checks the correctness of W with algorithm Verify and its secret vectors Z_g,Vg. If its output is 1, g judges W being correct, otherwise, g judges W being fake or polluted, and discards W. Fig. 3 depicts this process. The correctness of the verification algorithm will be analyzed in correctness of KEPTE.

3.3.3. The Correctness of KEPTE

Given an encoding data packet $W= (w_1,w_{2,\ldots\ldots,}w_{m+n})$ with its N tags $t_{W,1,}\ldots\ldots..,t_{W,N}$. Combine, the following is satisfied:

$$(W,t_{W,1,}\ldots\ldots..,t_{W,N})=\sum^n_{i=1} w_i\,(p_i,t_{p,1},\ldots\ldots,t_{pi,N}) \quad (1)$$

Then, according to the algorithm Sign, we have

$$t_{w,j} = (w1,\ldots.wn) \cdot (tp1,j,\ldots..,tpn,j)^T$$
$$= (w_1,\ldots..,w_n).(P_{T1},\ldots.P^T_n)^T.X^T_j \quad (2)$$
$$= W.X^T_j$$

Upon receiving an encoding data packet $W =(w_1,w_{2,\ldots.}w_{m+n})$ with its N tags $t_{W,1,}\ldots..,t_{W,N}$, a node g will use the algorithm Verify and two secret vectors Z_g; V_g to check the correctness of W.

$$W.V_g^T= W.((x_1)^T,\ldots..,(X_N)^T \cdot Z_g^T$$
$$= (t_{w,1},\ldots..,t_{w,N}).Z_g^T. \quad (3)$$

From (3), the node g can use the algorithm Verify to check

the correctness of a packet with N tags $t_{W,1,}\ldots.. t_{w,N}$ and two secret vectors Z_g,V_g. So with KEPTE, a node is able to check the correctness of a received encoded data packet.

4. Conclusion

The system is better at the interaction between the raw analog data and digitized form of information. The design will greatly enhance the quality of the computing and data security in a drastic way. There is the tradeoff between accuracy and availability in case of intolerance.

5. References

[1] Eleftheria Athanasopoulou, LocX.Bui, Tianxiong Ji, R. Srikant, *Fellow*, Alexander Stolyar "Back-Pressure-Based Packet-by-Packet Adaptive Routing in Communication Networks" IEEE/Acm Transactions On Networking, Vol. 21, No. 1, February 2013.

[2] Xiaohu Wu, Yinlong Xu, Chau Yuen, and Liping Xiang "A Tag Encoding Scheme against Pollution Attack to Linear Network Coding" IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 1, January 2014.

[3] Dahai Xu, Mung Chiang, Jennifer Rexford, "Link-State Routing With Hop-by-Hop Forwarding Can Achieve Optimal Traffic Engineering" IEEE/Acm Transactions On Networking, Vol. 19, No. 6, December 2011.

[4] Qiming Li, John C.S. Lui, Dah-Ming Chiu "On the Security and Efficiency of Content Distribution via Network Coding" IEEE Transactions On Dependable And Secure Computing, Vol. 9, No. 2, March/April 2012.

[5] Frederique Oggier and Hanane Fathi "An Authentication Code Against Pollution Attacks in Network Coding" IEEE/Acm Transactions On Networking, Vol. 19, No. 6, December 2011.

[6] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, "Optimization based rate control for multicast with network coding," in *Proc. IEEEINFOCOM*, Anchorage, AK, May 2007, pp. 1163–1171.

[7] A. Eryilmaz, D. S. Lun, and B. T. Swapna, "Control of multi-hop communication networks for inter-session network coding," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1092–1110, Feb. 2011.

[8] Q. Li, J.C.S. Lui, And D.-M. Chiu, "On The Security And Efficiency Of Content Distribution Via Network Coding," IEEE Trans. Dependable And Secure Computing, Vol. 9, No. 2, Pp. 211-221, Jan. 2011.

6. Author profile

Dr. M. Senthamil Selvi M.E., Ph.d
Prof. /Head-Information Technology
Sri Ramakrishna Engineering College,
Coimbatore-641022.

Adhars Ram.B pursuing final year B.Tech Information Technology in Sri Ramakrishna Engineering College from the period of 2011-2015,respectively.

Ashok.K pursuing final year B.Tech Information Technology in Sri Ramakrishna Engineering College from the period of 2011-2015,respectively.

Babu.V pursuing final year B.Tech Information Technology in Sri Ramakrishna Engineering College from the period of 2011-2015,respectively.