ChinaRxiv 中X

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-201806.00098

A Network-Coding-Aware Opportunistic Routing Scheme for Software-Defined Space-Based Backbone Networks (Postprint)

Authors: Ding Ran, Liang Jun, Xiao Nan, Weichao Sun

Date: 2018-06-19T00:00:00+00:00

Abstract

To guarantee the efficient execution of airborne communication missions, and to address the shortcomings in terms of delay, overhead, and reliability during routing information interaction in software-defined airborne backbone networks, this paper proposes an improved CAOR-SD scheme that integrates the advantages of both software-defined networking and network coding. Building upon the fundamental principles of coding-aware opportunistic routing, the scheme constructs a forwarding cost function that comprehensively considers node distance and coding gain to allocate priorities among candidate forwarding nodes, thereby improving network transmission efficiency. Furthermore, the feedback mechanism for control packets and the flow table structure of nodes are specifically designed to effectively reduce control overhead. Simulation results demonstrate that this routing scheme can adapt well to airborne communication scenarios, reducing transmission delay and routing overhead while improving network throughput and successful data transmission rate.

Full Text

Preamble

Coding-Aware Opportunistic Routing Scheme for Software-Defined Airborne Backbone Networks

Ding Ran, Liang Jun, Xiao Nan, Sun Weichao

(Information & Navigation College, Air Force Engineering University, Xi' an 710077, China)

Abstract: To ensure efficient implementation of airborne communication missions, this paper proposes an improved CAOR-SD scheme that combines the advantages of software-defined networking and network coding to address deficiencies in delay, overhead, and reliability during routing information exchange in software-defined airborne backbone networks. While following the basic principles of coding-aware opportunistic routing, this scheme constructs a forwarding cost function that integrates node distance and coding gain to allocate priorities among candidate forwarding nodes, thereby improving network transmission efficiency. Additionally, the feedback mechanism for control packets and the flow table structure are redesigned to effectively reduce control overhead. Simulation results demonstrate that this routing scheme adapts well to airborne communication scenarios, reducing transmission delay and routing overhead while improving network throughput and data delivery success rate.

Keywords: software-defined network; airborne backbone networks; network coding; opportunistic routing; priority assignment

0 Introduction

Airborne backbone networks are high-speed backbone carrier networks with relatively stable topologies. Since the U.S. Department of Defense proposed the Global Information Grid (GIG) in the late 1990s, such networks have become critical information infrastructure for implementing information warfare and an important aerial network facility for Department of Defense information exchange, providing air connectivity for GIG. For China, airborne backbone networks will also become a key enabler supporting future long-range and overseas combat communications. In recent years, due to poor interoperability among heterogeneous networks, low flexibility, and long deployment cycles in traditional backbone networks, the contradiction between forwarding capabilities and application requirements has become increasingly apparent, making network control and management more complex. The development of Software-Defined Networking (SDN) provides new ideas for airborne backbone network design, with increasing numbers of experts proposing to decouple the tight coupling between control and forwarding planes and introduce the flexible, open, and programmable SDN concept into airborne backbone networks.

Software-defined airborne backbone networks refer to more flexible and efficient networks based on SDN architecture. The separation of control and forwarding enables logically centralized resource allocation, improved link utilization, shortened network deployment cycles, and enhanced mission-oriented elastic adaptability. To ensure efficient communication transmission in airborne backbone networks, designing efficient routing strategies is a key issue that directly affects network performance. In recent years, routing schemes combining network coding and opportunistic routing have shown great potential for improving network performance. In 2006, Katti et al. first proposed the COPE network coding transmission scheme at the protocol layer, which greatly improved network communication throughput performance. In 2007, the MORE protocol (MAC-independent opportunistic routing and encoding protocol) was proposed, reducing duplicate transmission probability and significantly improving network throughput. In 2012, CAOR (coding-aware opportunistic routing) for wireless Mesh networks was introduced, which actively creates coding opportunities based on COPE and integrates network coding decisions into candidate node selection and priority allocation to achieve greater routing gains.

Considering the advantages of CAOR in improving wireless network performance, this paper adopts the coding-aware opportunistic routing concept to implement routing functions at the control plane of software-defined airborne backbone networks. However, since software-defined airborne backbone networks employ logically centralized routing computation performed only by control nodes, control node overhead is particularly significant. Compared with traditional airborne backbone networks, there are higher requirements for timeliness, transmission reliability, network throughput, and routing overhead. Therefore, this paper proposes an improved CAOR-SD (coding-aware opportunistic routing for software-defined network) protocol scheme with better performance in software-defined airborne networks.

1 System Model and Basic Principles

Airborne backbone networks consist of large aerial platforms with predictable dynamic trajectories (such as command and communication aircraft, early warning aircraft, electronic reconnaissance aircraft, high-altitude UAVs, and heavy-lift helicopters) connected through stable, high-bandwidth links to provide high-speed backbone carrier services to other nodes or networks. This paper introduces the SDN concept to construct a logically centralized yet physically distributed airborne backbone network, enabling aerial platforms to fully leverage their advantages of rapid large-scale deployment and flexible, efficient networking. As shown in [Figure 1: see original paper], airborne backbone nodes serve as control nodes. Each control node must obtain the topology of its local subnet, manage switches within the subnet, and exchange topology information with adjacent control nodes to acquire a global topology structure, enabling each control node to compute forwarding paths based on a global view. Ordinary nodes within access subnets send their topology information to control nodes, which compute forwarding paths and issue routing instructions through flow table distribution.

The basic idea of CAOR can be illustrated with [Figure 2: see original paper]. Assume packets P1 and P2 are sent from source nodes S1 and S8 to destination nodes S3 and S6, respectively. Traditional routing strategies would yield forwarding paths S1-S2-S3 and S8-S7-S6. However, neighbor nodes S4, S5, and S6 can all overhear packet P1, and similarly, S3, S4, and S5 can overhear packet P2. In CAOR, when the source node sends data packets, it selects nodes as candidate forwarding nodes based on certain metrics. Nodes in this set perform logical encoding on overheard packets and forward them according to priority determined by some mechanism. Assuming S5 has the highest forwarding priority, after it sends P1 P2 to node S6, S6 can decode P2 by XORing with overheard

P1, and other nodes operate similarly. Thus, CAOR fully utilizes the broadcast nature of wireless channels to increase coding combination opportunities with other flows. Especially in lossy networks where each batch of packets may not arrive completely at destination nodes, this mechanism reduces transmission times through coded forwarding, increases coding opportunities, and improves network throughput.

Introducing coding-aware opportunistic routing into software-defined airborne backbone networks shows great potential. To further improve network performance and reduce routing overhead, this paper proposes an improved CAOR-SD scheme. The control node selects the node with maximum forwarding efficiency from the candidate forwarding node set based on computed forwarding cost function results for prioritized transmission. Meanwhile, control packets employ a hop-by-hop feedback mechanism to reduce packet forwarding times and improve link utilization. The entire routing decision process is computed by control nodes, which exchange global topology information and implement elastic control over data packet transmission paths in access subnets through OpenFlow protocol-based flow table distribution, thereby realizing a logically centralized, physically distributed large-scale airborne network where the centralized control approach effectively compensates for routing adjustment lag.

2 CAOR-SD Implementation Mechanism

2.1 Workflow Description

The main modules of control nodes and ordinary nodes during CAOR-SD implementation are shown in [Figure 3: see original paper]. The workflow proceeds as follows:

a) Network topology discovery and management is the prerequisite for control node routing computation. Control nodes broadcast probe packets, which neighbor nodes store and continue forwarding. Ordinary nodes periodically forward collected topology information to control nodes through neighbor nodes while obtaining paths to their corresponding control nodes. Through information sharing among control nodes and packets from ordinary nodes, control nodes maintain real-time awareness of global topology status to compute routes and distribute flow tables to ordinary nodes.

b) When ordinary nodes send data, they first query flow table entries. If a matching entry exists, they forward according to the corresponding rules. If no match exists (new flow), they request forwarding rules from their control node.

c) Upon receiving a request packet, the control node extracts source and destination IP addresses, performs routing computation based on candidate forwarding node selection and priority sorting methods, and distributes forwarding rules to the corresponding node through response packets.

d) After receiving a certain number of encoded packets, the destination node decodes them into source data packets. The source node switches to the next batch upon receiving an ACK packet. The source node then sends a termination packet to inform the control node, which stops computing and updating forwarding rules. Ordinary nodes delete flow table entries through timeout mechanisms. This completes the successful forwarding of one batch of data packets.

2.2 Candidate Forwarding Node Selection and Priority Allocation

In CAOR-SD, nodes broadcast data packets to candidate forwarding node sets. Candidate nodes must satisfy three conditions: mutual overhearing capability, being one-hop neighbors of the sending node, and being closer to the destination than the sending node. This paper adopts ETX (expected transmission count) as the routing metric for "distance" judgment.

Let N(S,D) denote the set of candidate forwarding nodes for node r on the path from source node S to destination node D, and n(S,D)i represent the ith node in this set. During the route discovery phase, node ETX values can be obtained iteratively from destination to source. The sum of the ETX value of link r-n(S,D)i and the ETX value of node n(S,D)i is recorded as Wi. The expectation of W1, W2, ···Wn constitutes node r' s ETX value. Data packet headers carry next-hop candidate forwarding node set information. When a node receives a data packet, it first checks whether it belongs to the candidate set based on header information. If it does, the node caches the packet and seeks coding opportunities for further forwarding; otherwise, it discards the packet.

Coding gain is a crucial parameter characterizing the effectiveness of network coding in routing. It can be represented by the forwarding distance gained when a packet is encoded with other packets. Let Z denote the ETX forwarding distance of a data packet, defined as the difference between the ETX value of the forwarding node and the expected ETX value of its next-hop candidate nodes. The coding gain NCGr for node r encoding source packet P with m other data packets is defined as:

$$NCGr = \sum_{k=1}^{m} Z_{S_k} - \sum_{k=1}^{m} Z_k$$

where ZS is the ETX forwarding distance of the source data packet, and Zk represents the ETX forwarding distance of the m other packets carried by the data packet.

This paper defines a forwarding cost function COST(r) to evaluate forwarding node effectiveness:

$$COST(r) = ETX_r - \alpha \cdot NCG_r$$

where ETXr represents the expected transmission count of node r, NCG_r is the network coding gain of node r, and is a network coding attribute factor (= 1 if the packet has coding opportunities, = 0 otherwise). Through computing this cost function, priorities are allocated to candidate forwarding nodes with coding opportunities. Analysis shows that this forwarding cost function simultaneously incorporates both node distance and coding gain into priority ordering—smaller cost values indicate higher forwarding performance and thus higher priority. Let n(S,D)_K denote the node with priority K in the candidate forwarding node set (smaller K indicates higher priority). The prioritized candidate forwarding node set can be expressed as N(S,D) = [n(S,D)_1, n(S,D)_2, ···, n(S,D)_K].

The priority allocation algorithm for candidate forwarding nodes is as follows:

Input: Node r
Output: Prioritized next-hop candidate forwarding node set result for node r

Begin
1. result = NULL
2. for each neighbor ni of node r, i ← 1 to M // node r has M neighbors
3. if ni satisfies candidate conditions
4. then add ni to result // select K nodes from r' s neighbors as next-hop candidates
5. for i ← 1 to K-1
6. for j ← i+1 to K
7. do if COST(ni) > COST(nj) // compare forwarding costs of candidate nodes
8. then exchange the order of ni and nj in result // smaller cost indicates higher priority
9. return result = [n1, ···, nK]
End

2.3 Hop-by-Hop Feedback Mechanism for Control Packets

Ordinary data packets are broadcast among candidate forwarding nodes until the destination receives them and confirms with an ACK packet sent back to the source. In this software-defined airborne network, control packets are forwarded by continuously selecting the highest-priority node in the candidate forwarding node set. End-to-end ACK feedback consumes significant time, and given the time-varying nature and packet loss characteristics of airborne backbone networks, using conventional end-to-end ACK feedback would cause substantial retransmissions due to packet loss. This would not only affect data transmission real-time performance but also increase overall network overhead and reduce throughput.

Therefore, this paper adopts a hop-by-hop feedback mechanism where each node, upon receiving a control packet, immediately sends an ACK confirmation to the previous hop according to forwarding rules in the flow table. The sending node ensures next-hop reception through timeout and retransmission mechanisms. This hop-by-hop feedback mechanism for control packets guarantees data transmission reliability and reduces delay while preventing massive retransmissions that would increase control overhead, thereby improving network throughput.

3 Flow Table Structure Design

In traditional CAOR protocols, forwarding node selection and priority ordering are encapsulated in data packets, reducing effective throughput and affecting node forwarding efficiency. The introduction of flow tables in software-defined airborne backbone networks provides new opportunities. Control nodes distribute candidate forwarding node selection, priority ordering, and ACK feedback paths for flows to ordinary nodes through flow table distribution, which ordinary nodes store in their flow tables. When forwarding data, nodes in access subnets first match flow table entries and execute corresponding forwarding rules.

Drawing from the OpenFlow v1.3 flow table structure, this paper improves the node flow table structure as shown in [Figure 4: see original paper]. The structure contains fewer bytes than traditional packet encapsulation and maintains more concise control information, thereby reducing control overhead. The fields are:

a) **Match Field:** Matches data packets based on flow ID.

b) **Flow Priority:** Determines the matching order of flow table entries.

c) **Forwarding Priority:** Indicates priority level if the node is in the candidate forwarding node set for the corresponding flow; otherwise, it is zero.

d) **ACK Packet Next-Hop Address:** Contains the next-hop IP address for ACK packets if the node is on the ACK feedback path; otherwise, it is 0.

e) **Counter:** Updates the count of matched data packets.

f) **Instruction:** Modifies the action set or processing for data packets. If forwarding priority is zero, the packet is discarded.

g) **Timeout Timer:** Specifies the maximum lifetime or idle time for a flow. Flow table entries are deleted if not matched for a certain period.

4 Simulation and Analysis

Airborne backbone networks exhibit high stability, with relative positions of backbone nodes remaining essentially unchanged over certain periods. Therefore, this paper abstracts airborne node movement as a uniform particle motion model. In a MATLAB simulation environment, we compare three routing protocols—AODV (Ad-hoc On-demand Distance Vector), CAOR (Coding-Aware

Opportunistic Routing), and the improved CAOR-SD—in a scenario with 6 backbone control nodes and 30 ordinary access nodes. Considering factors such as large aerial platform flight speeds, the simulation sets a random distribution area of 25 km × 25 km, node mobility speed of 300 m/s, communication radius of 50 km, and simulation time of 30 minutes. Different data sending rates simulate various network load conditions. Since software-defined airborne backbone networks have high requirements for timeliness, reliability, and low overhead, we select transmission delay, network throughput, routing overhead, and successful transmission rate as performance evaluation metrics.

4.1 Transmission Delay and Network Throughput

Transmission delay refers to the time duration from when a data packet is sent by the source node until it reaches the destination node. Network throughput refers to the aggregated data rate of all flows per unit time. [Figure 5: see original paper] and [Figure 6: see original paper] show the variation of transmission delay and network throughput for the three routing schemes under different network loads as data sending rates change.

As seen in [Figure 5: see original paper] and [Figure 6: see original paper], using candidate forwarding nodes instead of fixed forwarding nodes in AODV effectively alleviates network congestion. Both the improved CAOR-SD scheme and CAOR scheme consistently outperform AODV (which does not employ network coding) in delay and throughput performance. When the data sending rate is below 40 kbps, the difference is not significant. As network load increases and collisions grow, the advantage of CAOR-SD, which determines forwarding priority based on coding opportunity size, gradually becomes evident among the three curves. This is because the improved scheme can proactively sense coding opportunities and simultaneously incorporate both node distance and coding efficiency into forwarding node selection and priority ordering, thereby reducing data forwarding delay and improving network throughput. However, when the data sending rate exceeds 180 kbps, severe network congestion offsets coding gains, and the improvement in delay and throughput performance becomes less pronounced.

4.2 Routing Overhead and Successful Transmission Rate

Routing overhead is defined as the ratio of control packets to total communication data volume within the statistical period. Successful transmission rate is defined as the ratio of data packets successfully received by nodes to those sent by source nodes. [Figure 7: see original paper] and [Figure 8: see original paper] present simulation results for routing overhead and successful transmission rate under different network loads.

[Figure 7: see original paper] shows that when load is very light, AODV has an advantage over CAOR and CAOR-SD because AODV performs route discovery only when data is sent, resulting in smaller routing overhead. As network load increases and the data sending rate exceeds 60 kbps, the number of control packets required for AODV to maintain routing information increases dramatically, causing overhead to rise rapidly. In contrast, CAOR-SD maintains smaller and more stable routing overhead because it requires only control nodes to compute and maintain global routing information, significantly reducing the number of control packets needed compared to maintaining routing information across the entire network. The introduction of flow tables reduces packet encapsulation overhead, substantially decreasing overall network control overhead. Additionally, the hop-by-hop feedback mechanism for control packets reduces retransmissions and further lowers routing overhead.

[Figure 8: see original paper] demonstrates that CAOR-SD generally achieves better successful transmission rate performance than the other two schemes, with its advantage becoming increasingly apparent as network load grows. Particularly, AODV' s transmission rate performance drops by up to 40%, while CAOR and CAOR-SD maintain relatively stable transmission rates. This shows that network coding greatly improves data transmission reliability. Furthermore, the hop-by-hop feedback mechanism for control packets in CAOR-SD further ensures data transmission reliability.

5 Conclusion

Software-defined airborne backbone networks are an emerging network architecture developed to meet the burgeoning demands of airborne communications. Compared with traditional airborne networks, they feature shorter deployment cycles, improved mission-oriented flexible communication capabilities, and can resolve issues such as inconsistent network cognition and routing adjustment lag in traditional networks, significantly enhancing overall network performance. To meet higher requirements for timeliness, reliability, and low overhead in such networks, this paper proposes a coding-aware opportunistic routing scheme called CAOR-SD, tailored for this novel logically centralized, physically distributed airborne backbone network. The scheme defines a forwarding cost function that integrates node distance and coding gain for candidate forwarding node priority allocation, while redesigning the hop-by-hop feedback mechanism for control packets and the node flow table structure. Simulation results demonstrate that CAOR-SD reduces transmission delay, improves network throughput, increases data delivery success rate, and reduces routing overhead compared with traditional wireless routing, achieving better overall network performance.

References

[1] Liang Jun, Hu Meng, Guan Hua, et al. Research on space backbone network architecture and key technologies [J]. Journal of Air Force Engineering University: Natural Science Edition, 2016, 17(4): 52-58.

[2] Xia Wenfeng, Wen Yonggang, Foh C H, et al. A survey on software-defined networking [J]. Communications Surveys & Tutorials IEEE, 2015, 17(1): 27-51.

[3] Bernardos C J, De L O A, Serrano P, et al. An architecture for software defined wireless networking [J]. Wireless Communications IEEE, 2014, 21(3): 52-61.

[4] Open Networking Foundation. SDN architecture overview [R]. 2013.

[5] Huang Tao, Liu Jiang, Wei Liang, et al. SDN core principles and application practice [M]. 2nd ed. Beijing: People' s Posts and Telecommunications Press, 2016.

[6] Zhang Weifeng. The depth resolution for SDN: profit, strategy, technology, practice [M]. Beijing: Publishing House of Electronics Industry, 2014.

[7] Cheng Bownan, Block F J, Hamilton B R, et al. Design considerations for next-generation airborne tactical networks [J]. IEEE Communications Magazine, 2014, 52(5): 138-145.

[8] Sampigethaya K. Software-defined networking in aviation: opportunities and challenges [C]// Proc of IEEE Integrated Communication, Navigation, and Surveillance Conference. 2015: 1-21.

[9] Elmasry G, Mcclatchy D, Heinrich R, et al. A software defined networking framework for future airborne connectivity [C]// Proc of IEEE Integrated Communications, Navigation and Surveillance Conference. 2017: 2C2-1-2C2-9.

[10] Wang Jun, Chen Zhihui, Tian Yongchun, et al. Application of software-defined network technology in tactical communication network [J]. Communication Technology, 2014, 47(12): 1392-1399.

[11] Ramirez R. Link management in the air force airborne network [C]// Proc of IEEE Military Communications Conference. 2005.

[12] Tiwari A, Ganguli A, Sampath A. Mobility aware routing for the airborne network backbone [C]// Proc of IEEE Military Communications Conference. 2008: 1-7.

[13] Pahlevani P, Lucani D E, Pedersen M V, et al. PlayNCool: opportunistic network coding for local optimization of routing in wireless mesh networks [C]// Proc of IEEE GLOBECOM Workshops. 2013: 812-817.

[14] Katti S, Rahul H, Hu Wenjun, et al. XORs in the air: practical wireless network coding [C]// Proc of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications ACM. 2006: 243-254.

[15] Chachulski S, Jennings M, Katti S, et al. Trading structure for randomness in wireless opportunistic routing [C]// Proc of ACM SIGCOMM 2007. New York: ACM Press, 2007: 169-180.

[16] Chung Kuncheng, Chou Yichun, Liao Wanjiun. CAOR: coding-aware opportunistic routing in wireless Ad hoc networks [C]// Proc of IEEE International Conference on Communications. 2012: 136-140.

[17] Katti S, Rahul H, Hu Wenjun, et al. XORs in the air: practical wireless network coding [J]. IEEE/ACM Trans on Networking, 2008, 16(3): 497-510.

[18] Guo Bin, Li Hongkun, Zhou Chi, et al. Analysis of general network coding conditions and design of a free-ride-oriented routing metric [J]. IEEE Trans on Vehicular Technology, 2011, 60(4): 1714-1727.

[19] Yan Yan, Zhang Baoxian, Zheng Jun, et al. CORE: a coding-aware opportunistic routing mechanism for wireless mesh networks [J]. Wireless Communications IEEE, 2010, 17(3): 96-103.

[20] Fan Yongbing, Chen Nan, Huang Zhilan, et al. Guide for SDN [M]. Beijing: People' s Posts and Telecommunications Press, 2015.

[21] Glabbeek R, Portmann M, Tan W L, et al. Modelling and verifying: the AODV routing protocol [J]. Distributed Computing, 2016, 29(4): 279-315.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.