Mitigating Routing Misbehavior in the Internet of Drones Environment

Cong Pu and Pingping Zhu

Dept. of CSEE, Marshall University

Huntington, WV 25755, USA

cong.pu@ieee.org

zhup@marshall.edu

MARSHALL UNIVERSITY

IEEE

2022 IEEE 95th Vehicular Technology Conference - Spring

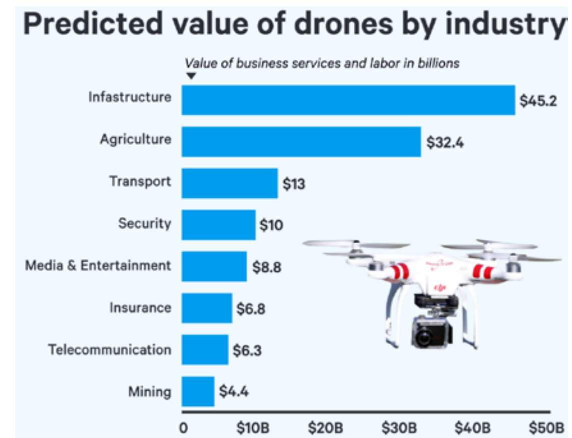
Outline

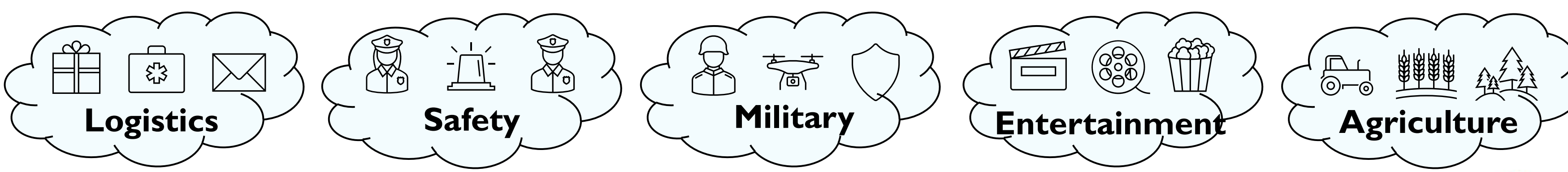
- Introduction & Motivation
- Related Work
- Proposed Routing Misbehavior Detection/Mitigation
  - System Model
  - Distributed Countermeasure
- Performance Evaluation and Analysis
- Concluding Remarks

Introduction

- Initially used as military strike weapons, drones discover a variety of civilian applications
  - goods delivery
  - aerial surveillance
  - combating COVID-19

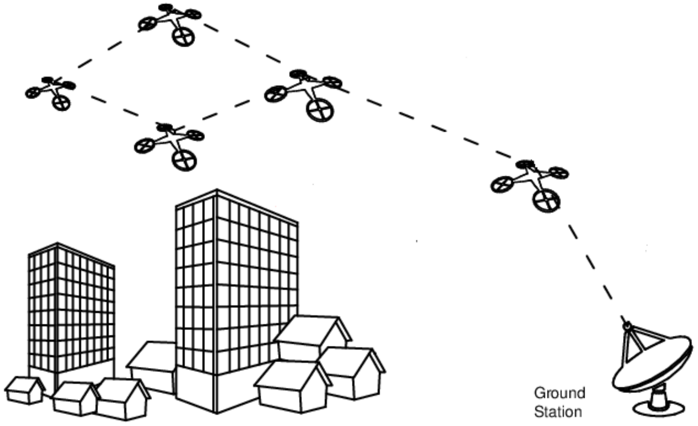
- "Drone Market Report 2020"
  - the drone industry is expected to grow to $43 billion by 2025
- The demand for drones by various unites is high; deployed for a wide range of apps.





Introduction

- To fully exploit drones, Internet of Drones (IoD) is proposed
  - mobile drones
  - stationary ground stations
    - acts as access point
  - drone-to-drone (D2D) comm.
  - drone-to-ground station (D2I) comm.
  - exploiting intermittent connect.



- The IoD is lack of persistent connectivity
  - between drone and drone, and between drone and ground station

store-carry-and-forward strategy

the most promising candidate for delivering data in the IoD

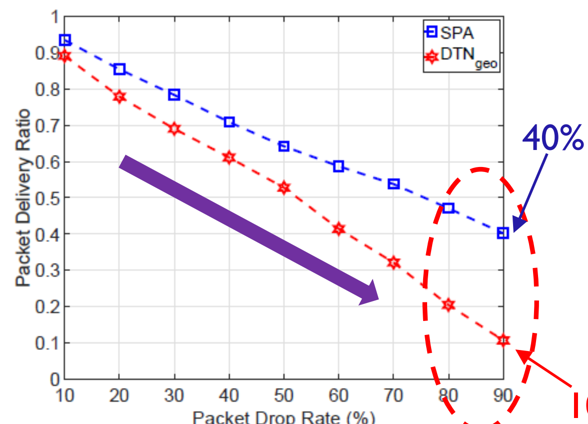
a drone stores the received packets in the storage, carries them while flying around, and forwards them to the next-hop drone or destination (i.e., ground station)

Motivation

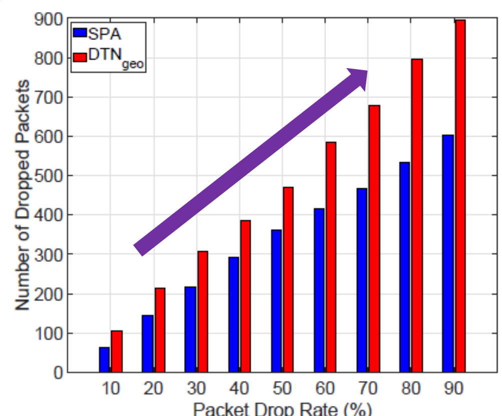
- *Routing protocol*: efficient info. sharing and team performance
- As a result of high mobility and resource constraints, the IoD is vulnerable to *routing attacks*
  - an adversary strategically misbehave by dropping the packets
    - saving its energy power or launching attacks
- *Routing attacks/misbehaviors degrade network performance*
  - packet delivery ratio (PDR) reduction; dropped packets increase
  - preliminary experiments (*SPA* [8] and DTN_{geo} [10])

SPA [8]: a stochastic packet forwarding algorithm

DTN_{geo} [10]: a shortest path forwarding algorithm



(a)



(b)

Motivation (cont.)

- *Routing attacks* are an old research topic in diverse environments
  - traditional computer network
  - mobile ad hoc network
  - wireless ad hoc network
  - vehicular ad hoc network
  - etc.

- existing countermeasures
  - monitoring-based
  - acknowledgment-based
  - bait-based
  - cryptography-based

- no/low mobility is considered
- exiting schemes do not apply in IoD

- In addition, there is no available work concentrating on routing attacks and their countermeasures in the IoD
  - our work fill this research gap in the community

Our Contribution

- This paper
  - proposes a distributed countermeasure (*Counter*^{Romir}) to detect / mitigate routing misbehavior in the IoD environment
    1. a drone keeps the previous signed communication invoice and shares it with the next-hop drone so that the next-hop drone can detect whether the drone has dropped any packets
    2. each drone saves and sends a small number of past communication invoices to the ZSP which can detect the misstating drone who misstates its communication invoices to avoid detection
  - extensive simulation experiments showing *Counter*^{Romir} is an efficient approach to mitigate routing misbehavior in the IoD

Most Countermeasures in the IoD

- monitor-based approach [15,17,18]
  - implicitly monitor the activity of next-hop node
    - determine whether it forwards the recently received packets
  - depends on stable connectivity between sender and receiver
    - difficult to achieve in the IoD environment
- acknowledgment-based approach [16,19,20]
  - explicit acknowledgement packet is required to confirm the receipt of packet from the receiver
  - relies on stable end-to-end routing path
    - not applicable in the IoD environment
- bait-based approach [7,21]
  - lure adversaries to launch attack with fictitious information
  - "fake" packets might get lost during the transmission
    - the high mobility of drones in the IoD environment

Most Countermeasures in the IoD

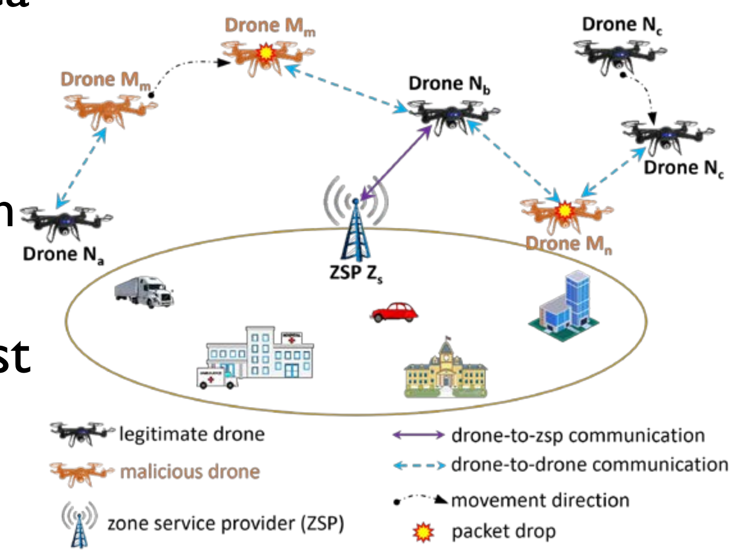
- trust management scheme [22]
  - a fuzzy trust scheme examining node's trustworthiness and converge trust, reward, and punishment values.
  - the trust evaluation process relies on neighbor monitoring
  - the cluster-head selection incurs extra communication overhead
- our approach *Counter*^{Romir} borrows the idea of *store-carry-and-forward mechanism* and *delay tolerant networking technique*
  - for each drone
    1. keeps the previous signed communication invoices
    2. shares them with the next-hop drone or nearby ZSP
    3. detect the routing misbehavior or misstating drones
  - a network-layer approach which can be implemented as an add-on to existing routing protocols (e.g., SPA [8], DTN_{geo} [10], etc.)
  - the *first* distributed approach against routing misbehavior in the IoD

Most Countermeasures in the IoD

- Four important issues should be addressed to detect routing attacks in the IoD
  i. intermittent connectivity in the IoD
     - store-carry-and-forward & delay tolerant networking techniques
  ii. routing attacks/misbehaviors
     - keeps signed communication invoice
  iii. misstating drone (fabricating communication invoice)
     - sharing invoices with ground station
  iv. integration with off-the-shelf routing protocols
     - designing countermeasure as a network layer add-on module
- This paper provides
  - in-depth analysis of routing attacks
  - distributed countermeasure against routing attack
  - bridge the research gap in the community

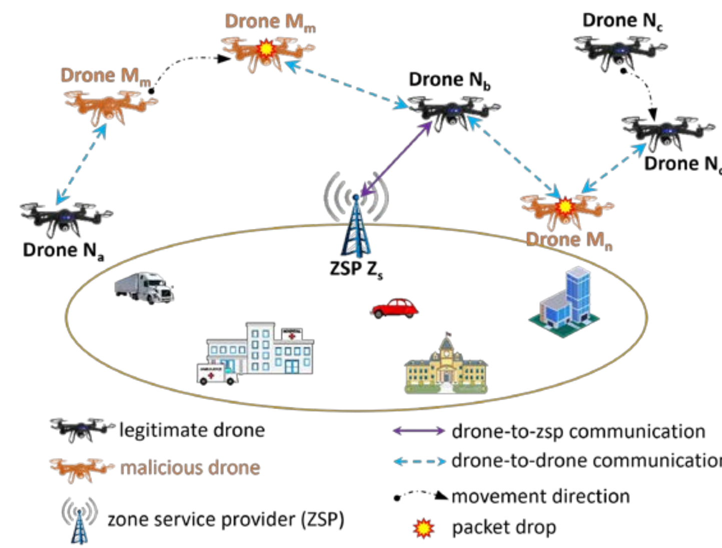
Counter^{Romir}: System Model

- A generic IoD scenario (combating COVID-19 pandemic)
  - a set of drones is deployed in the area
  - when a drone detects an event
    - generates data packets
    - sends them to nearby ground station
      - multi-hop relays
  - end-to-end path does not always exist
    - store-carry-and-forward strategy
      - stores received packets
      - carries them while flying
      - forwards them to next-hop (i.e., drone or ground station)
  - drone has limited storage space
    - a timer is used to purge stale packets
  - public-key cryptography [26,27] is being utilized



Counter^{Romir}: Adversary Model

- In wide-open airspace, drones can be captured ("anti-drone-gun")
  - compromising legitimate drones
  - making them behave maliciously
  - sending it back to the mission area



- The primary goal of adversary
  - degrade the network performance
    - strategically dropping the received packets
      - saving energy power or launching attacks
  - collusive routing attacks are not considered
    - a small number of malicious drones might collude together to drop the packets without being detected

Counter^{Romir}: Distributed Countermeasure

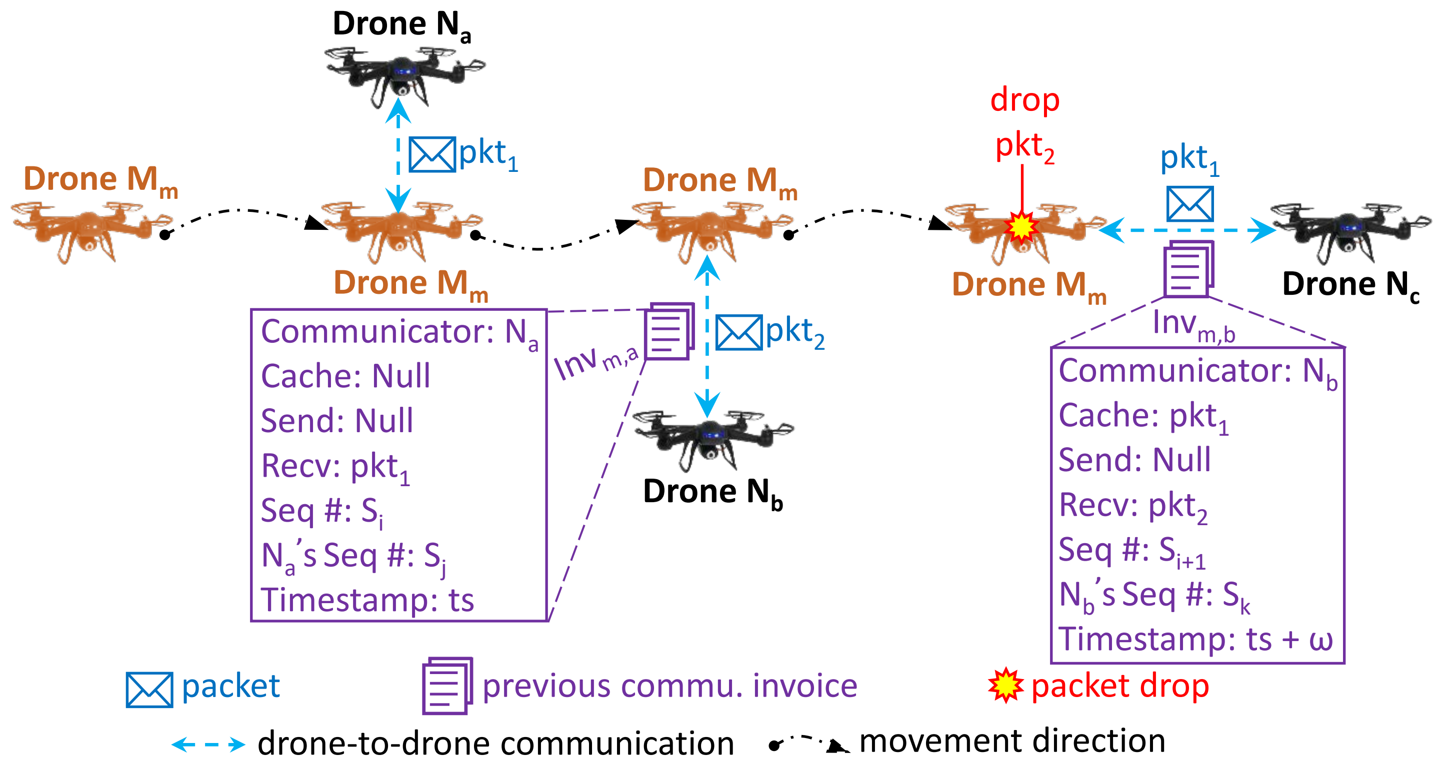
- When two drones contact,
  - exchange packets to be sent to next-hop drone
  - create communication invoice
    - communicators' ID
    - timestamp of communication
    - unique communication sequence number
    - what packets are in their caches before the communication
    - what packets they receive and send during the communication
    - their digital signatures
  - keep previous communication invoice
  - share next-hop drone with the following
    - previous communication invoice; the vector of packets in its cache

communication invoice:
a certified record that contains all communication related information of two drones.

determines whether the sending drone has dropped any packet

Yes, quit sending ← → No, continue sending

$Counter^{Romir}$: Distributed Countermeasure



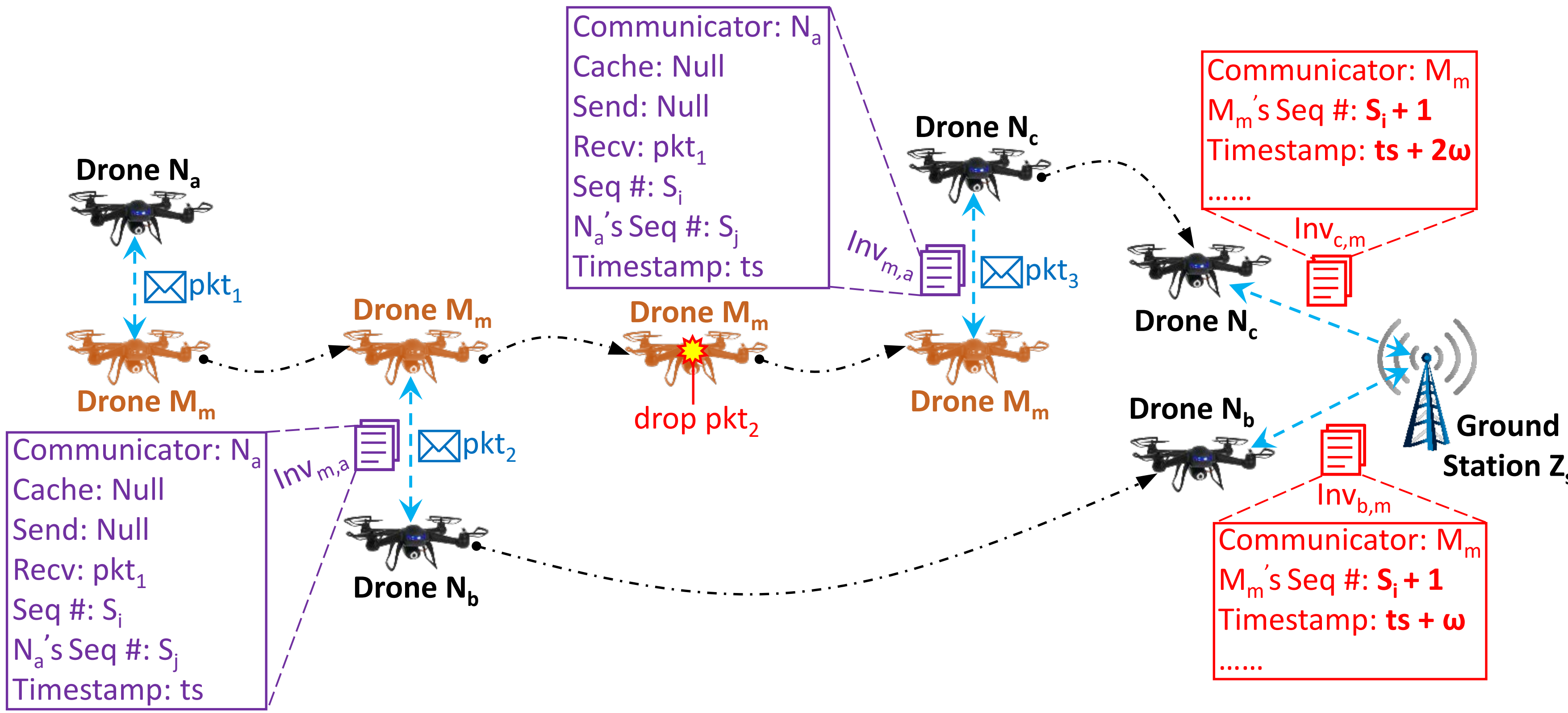
Counter^{Romir}: Distributed Countermeasure

- A malicious drone might share the incorrect communication invoice
  - cover up its packet dropping activity
  - avoid detection

  ⬇ inconsistent communication invoices

- ZSP detects the misstating activity of malicious drone
  - assign a unique comm. seq. number to each communication
  - the same seq. number will not be used twice
    - e.g., 1st seq. #: 1, 2nd seq. #: 2, 3rd seq. #: 3, ……
    - 2^{32} possible seq. # ⟶ large enough for packets
  - basic idea of detecting misstating activity:
    - each drone
      1. saves a small number of invoices of communications with other drones
      2. sends them to ground station for verification
    - ground station identifies inconsistent communication invoices from two different drones ⟶ detecting misstating activity

Counter^{Romir}: Distributed Countermeasure



Performance Evaluation

- Performance metrics
  - detection rate
  - miss/error detection rate
  - packet delivery ratio
  - the number of dropped packets
- Benchmark schemes
  - EYES [15]
    - monitor-based approach
  - SCAD [16]
    - acknowledgement-based approach
- Simulation environment
  - OMNeT++ [8]
    - event-driven network simulator

Algorithm 1: Routing Misbehavior Countermeasure

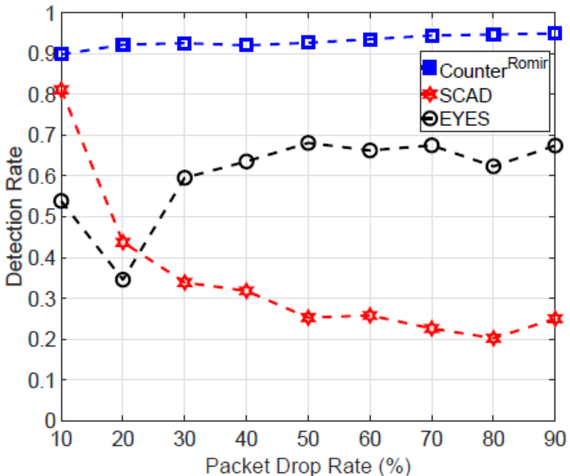
Input: $Inv_{m,a}$, Ca_m, $Inv_{b,m}$, $Inv_{c,m}$

/* drone detects packet dropping attack */
Function DroneDetect ($Inv_{m,a}$, Ca_m):
  /* $Inv_{m,a}[Ca_m]$ is the vector of cached packets at the beginning of previous communication; Ca_m is the vector of cached packets at the beginning of current communication. */
  /* *pkt* indicates the packet. */
  if *pkt* $\in (Inv_{m,a}[Ca_m] \cup Inv_{m,a}[Rec_m])$ **and** *pkt* $\notin Ca_m$ **and** *pkt* $\notin Inv_{m,a}[Sen_m]$ **then**
    detect packet dropping misbehavior;
  else
    exchange packets;
  end

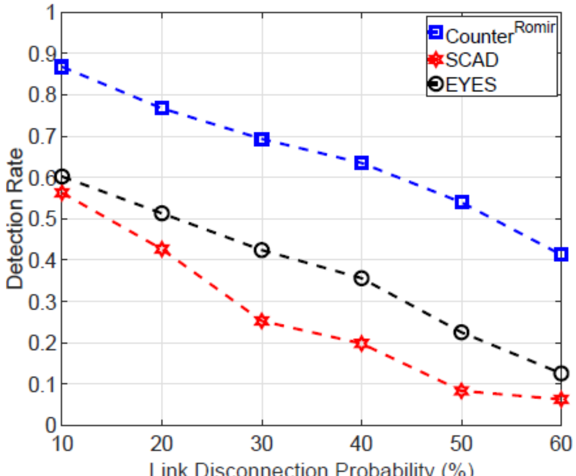
/* ZSP detects commu. invoice misstating */
Function ZSPDetect ($Inv_{b,m}$, $Inv_{c,m}$):
  if $Inv_{b,m}[TS] < Inv_{c,m}[TS]$ **then**
    if $Inv_{b,m}[Seq_m] \geq Inv_{c,m}[Seq_m]$ **then**
      detect communication invoice misstating;
      broadcast *Alarm* packet;
    end
  end
  if $Inv_{b,m}[TS] > Inv_{c,m}[TS]$ **then**
    if $Inv_{b,m}[Seq_m] \leq Inv_{c,m}[Seq_m]$ **then**
      detect communication invoice misstating;
      broadcast *Alarm* packet;
    end
  end

Performance Evaluation (cont.)

Detection Rate

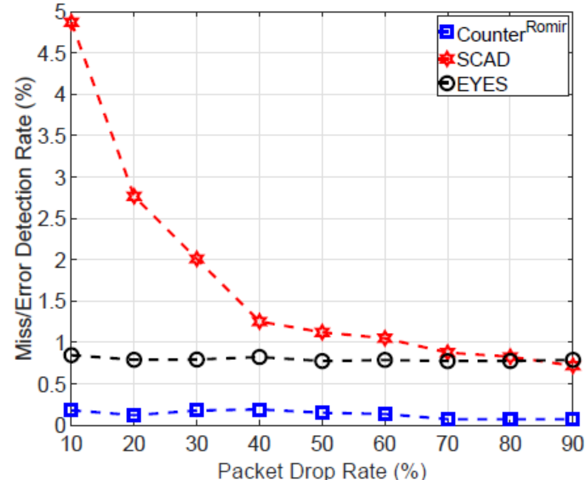


(a)

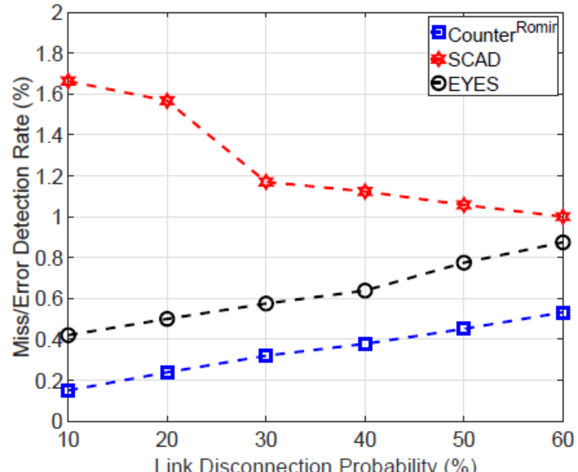


(b)

Miss Detection Ratio



(a)

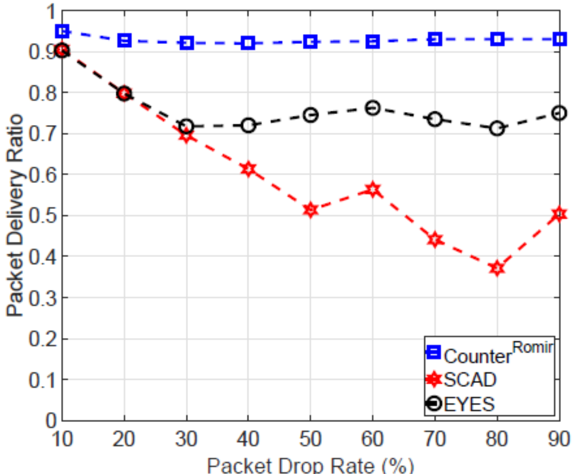


(b)

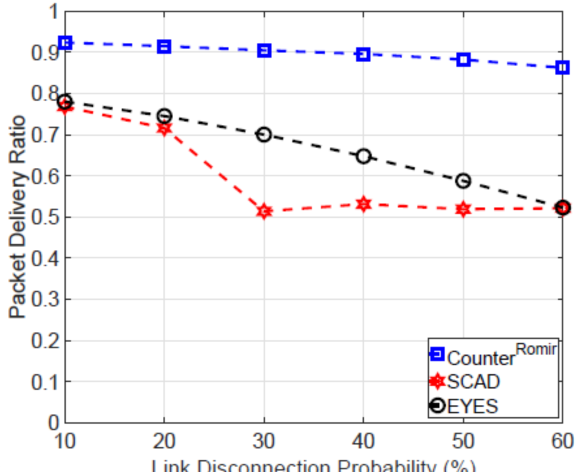
MARSHALL UNIVERSITY.

Performance Evaluation (cont.)

Packet Delivery Ratio

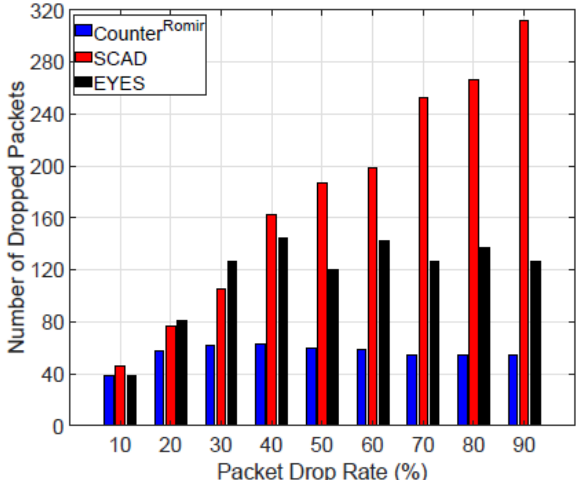


(a)

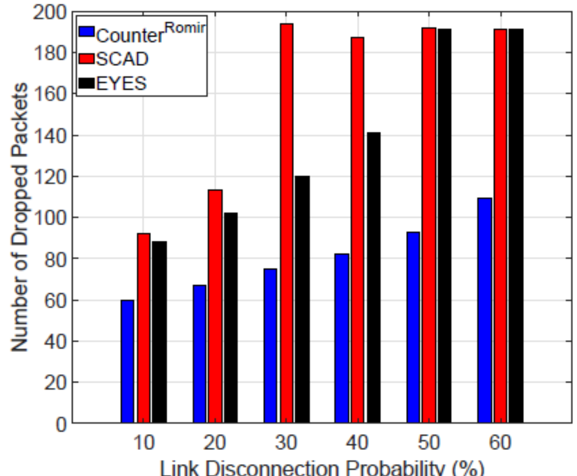


(b)

Number of Dropped Packets



(a)



(b)

MARSHALL UNIVERSITY.

Concluding Remarks

- Developed a distributed countermeasure (*Counter*^{Romir}) to detect / mitigate routing misbehavior in the IoD.
  - a drone keeps the previous signed communication invoice and shares it with the next-hop drone to detect any packet dropping activity
  - each drone saves and sends a small number of past communication invoices to the ground station which can detect the misstating drone
- *Counter*^{Romir} achieves
  - 90% detection rate
  - packet delivery ratio above 90%,
  - lower miss/error detection rate
- Under investigation…
  - a large number of communication invoices to be exchanged
  - data reduction strategy
  - a real-world testbed to explore the full potential of *Counter*^{Romir}

Any Questions?

Email: cong.pu@ieee.org