Defending Against Pollution Attacks in Network Coding for Wireless Mesh Network

Reza Curtmola, Jing Dong, Andrew Newell, Cristina Nita-Rotaru

Department of Computer Science and CERIAS

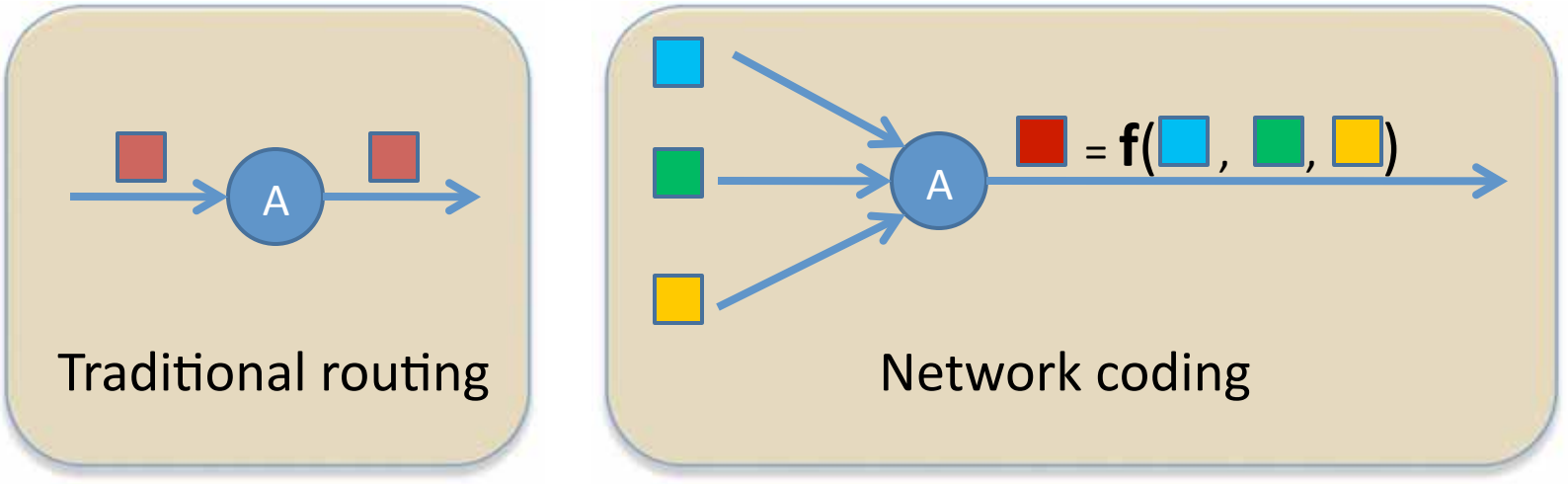
Purdue University

PURDUE UNIVERSITY

CERIAS

Network Coding: A New Paradigm

- **Key principle:** packet mixing at intermediate nodes



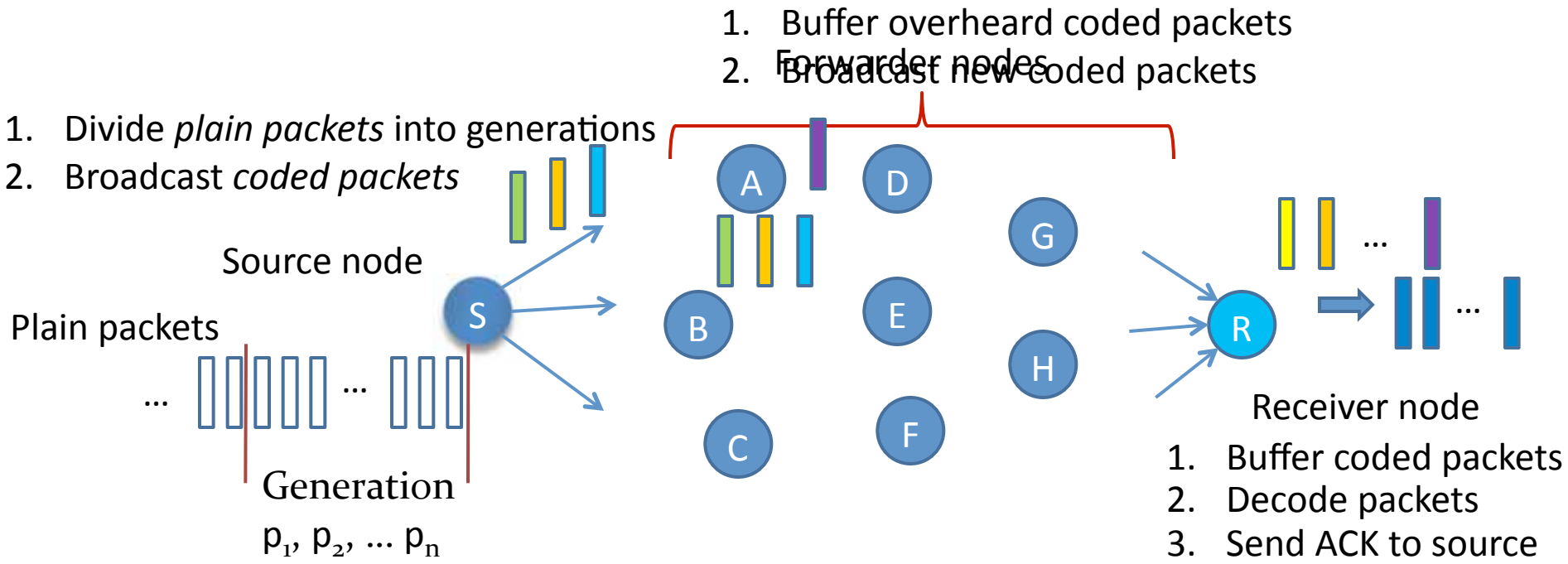
- **Benefits**: Higher throughput, reliability, robustness, energy efficiency
- **Applications**: wireless unicast and multicast, p2p storage and content distribution, delay-tolerant networks, vehicular networks

Wireless Network Coding Systems

- Intra-Flow Network Coding
  - Mix packets within individual flows
  - Examples: [Park; 2006], MORE [Chachulski; 2007], [Zhang; 2008a], [Zhang; 2008b], MIXIT [Katti; 2008], [Lin; 2008]

- Inter-Flow Network Coding
  - Mix packets across multiple flows
  - Examples: COPE [Katti; 2006], DCAR [Le; 08], [Das; 2008], [Omiwade; 2008a], [Omiwade; 2008b]

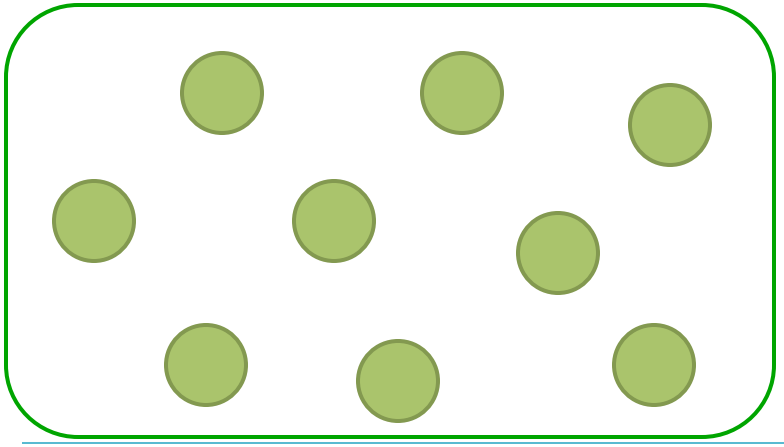
Intra-flow Network Coding



Need for Security in Wireless Networks

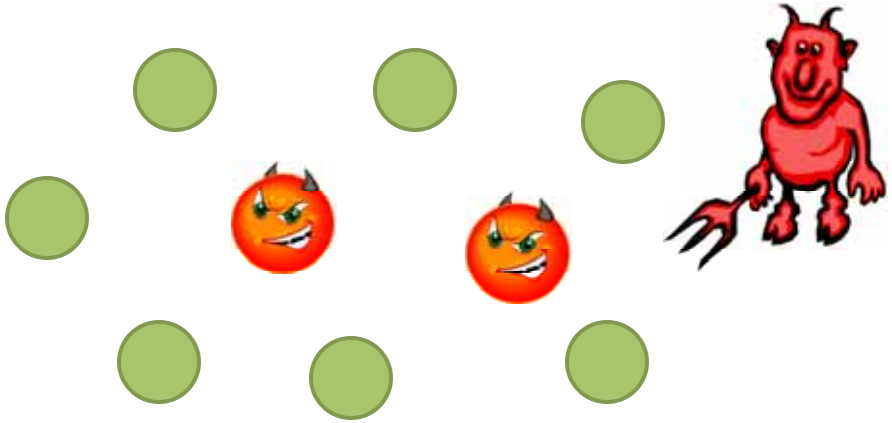
Ideal

- Benign environment
- All nodes are
  - Fully cooperative
  - Unselfish
  - Non-misbehaving



Reality

- Malicious outsiders
  - Packet jamming, injection, spoofing, replay, man-in-the-middle, …
- Malicious insiders
  - Captured and compromised
  - Byzantine behavior



Pollution Attacks

Definition

- Pollution attacks are attacks where *attackers* inject ***polluted coded packets*** into the network.
- A coded packet (c, e) is a polluted *coded packet* if

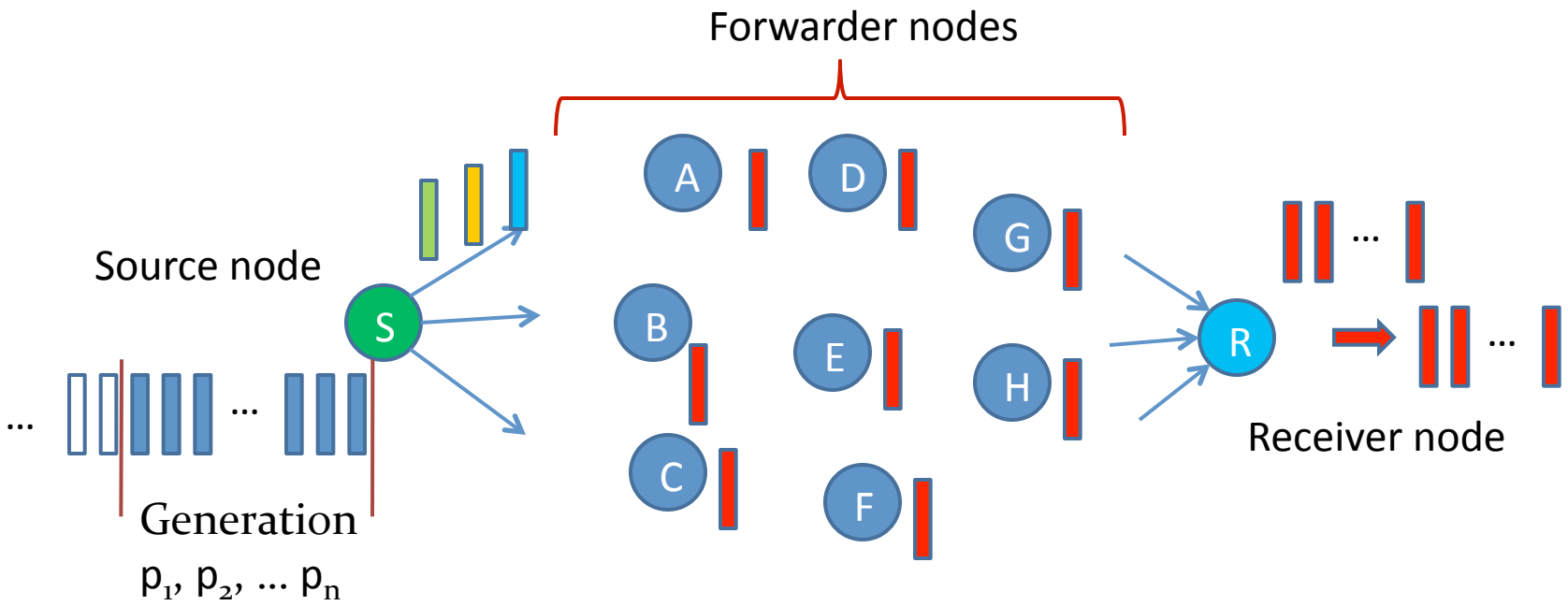
$$\mathbf{c} = (c_1, c_2, \ldots, c_n),\ c_i \in F_q$$

but

$$\mathbf{e} \neq c_1\mathbf{p_1} + c_2\mathbf{p_2} + \ldots + c_n\mathbf{p_n}$$

- Generic attack to any network coding system

Impact of Pollution Attacks

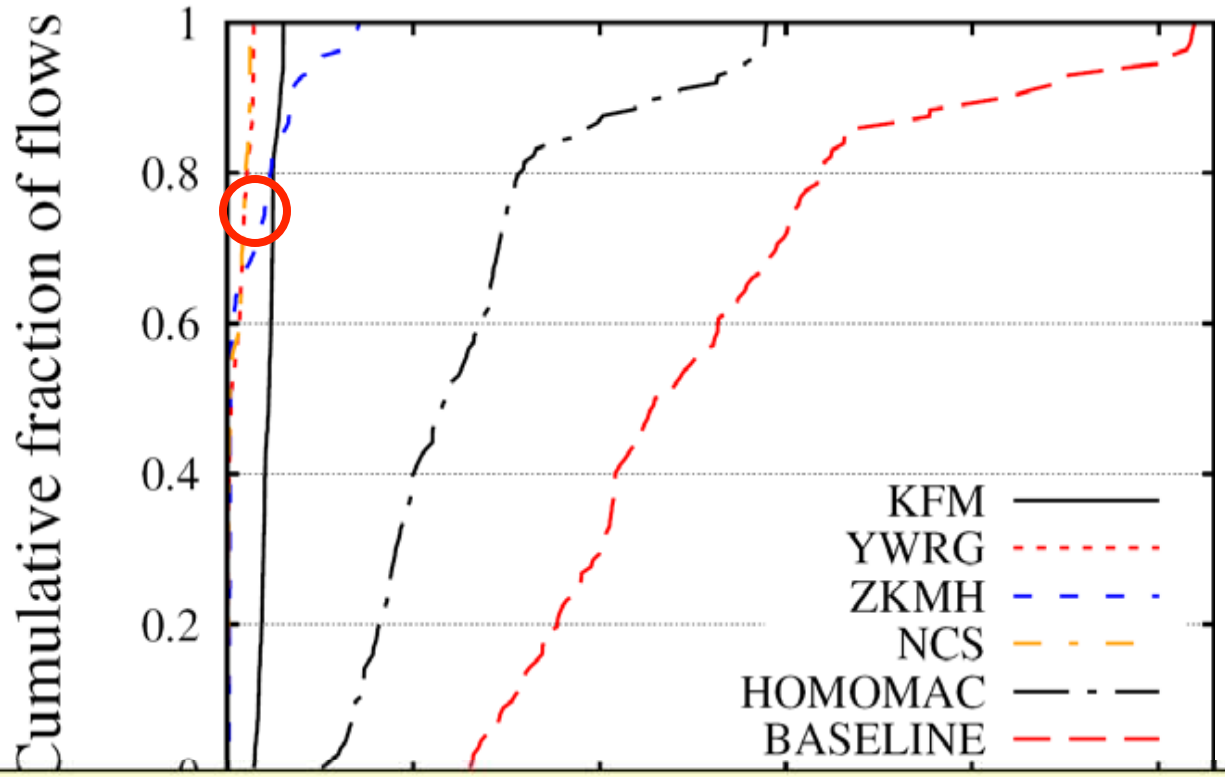


Epidemic attack propagation

Looks Like an Old Problem …

- At first sight
  - Looks like an authentication problem
    - Digital signatures, HMACs
- At a closer look
  - Forwarders need to verify that linear combinations of linear combinations of … linear combination of packets were sent by the source
    - Brute force approach where the source computes and disseminates signatures for all possible combinations is prohibitive in cost
- Solution requires a signature or hash scheme that is *homomorphic*

Previous Solutions (MORE) – NO ATTACK



The high overhead of crypto-based schemes render them impractical for wireless networks

Our Approach

Non-cryptographic checksum created by the source

Based on lightweight random linear transformations
Carries the timestamp of when it was created
Disseminated by the source in an authenticated manner
Not pre-image or collision resistant!

Security Relies on Time Asymmetry Checksum Verification

A node verifies a packet against a checksum that is created ***after*** the packet is received

Checksum Computation and Verification

- A generation of packets $G = [\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_n}]$

Checksum computation

- Compute H_s a random $b \times m$ matrix from a seed s
- Compute the checksum

$$\mathrm{CHK}_s(G) = H_s G$$

- b is a system parameter that trades off security and overhead

Checksum verification

Given $\mathrm{CHK}_s(G)$, *s and t*, check if a coded packet ($\mathbf{c}$, $\mathbf{e}$) is valid

- Check

$$\mathrm{CHK}_s(G)\, \boldsymbol{c} = H_s \boldsymbol{e}$$

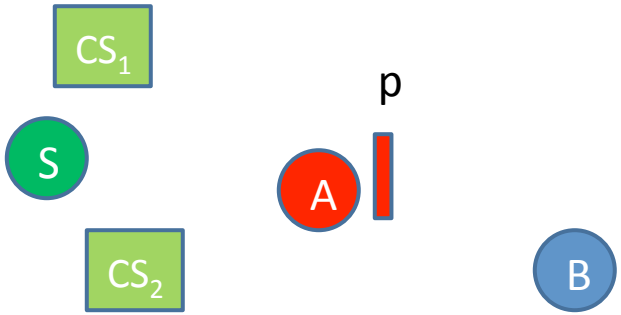
- Why?

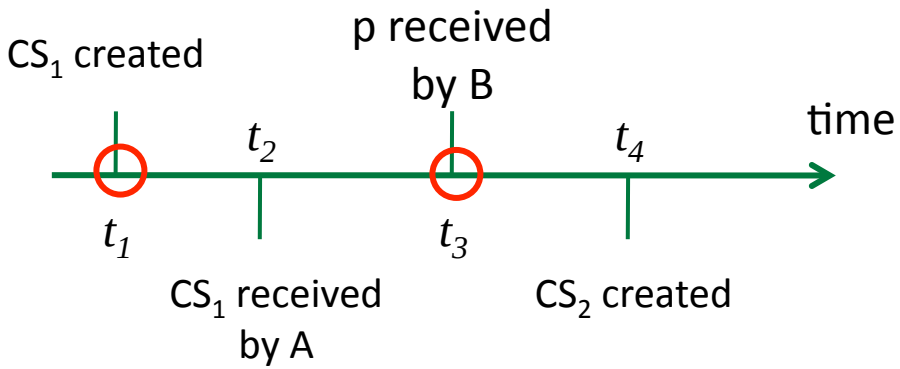
$$\mathrm{CHK}_s(G)\mathbf{c} = (H_s G)\mathbf{c} = H_s(G\boldsymbol{c}) = H_s\mathbf{e}$$

- **No false positive, may have false negative**

Our Approach: Example

Attacker can not inject a checksum or modify timestamp because checksum is signed by source





Packet p will be verified against CS_2 and not CS_1. The attacker does not gain anything by observing CS_1.

DART and EDART

- DART
  - Forwarder nodes buffer packets for checksum verification
  - Only verified packets are combined to form new packets for forwarding
  - Polluted packets are dropped at first hop, eliminating epidemic propagation
- EDART
  - Improves performance with optimistic forwarding

EDART

- DART delays packets for verification, increasing latency

Ideally:

- Delay polluted packets for verifying
- Forward correct packets without delay

However:

- Nodes do not know which packets are correct and which are polluted

EDART Overview

- Packets are **always** verified BUT
- Nodes *"closer"* to the attacker **delay** packets for verification
- Nodes *"farther"* away from the attacker **forward** packets without delay and will verify them when possible

Polluted packets are restricted to a region around the attacker

Correct packets are forwarded without delay

In case of no attack, all packets are forwarded without delay – **almost no impact on performance**

How to Decide when to Delay?

- h : Add a hop count that captures the number of hops a packet has traveled since the last verification
  - All verified packets will have h_{uv} set to 0
  - **Packets that traveled less than δ hops will be forwarded without delay, otherwise a node delays them**
  - When coding a new packet, set $h_{uv} = h_{max} + 1$ for the new packet to be the maximum h_{uv} in the packets used to create the new packet
  - If pollution was detected, the node will switch to delaying all packets for a time proportional with how big h

EDART Security Analysis

- Maximum pollution scope
  - Bounded by δ+1
- Average pollution scope
  - Bounded by δ/α
- Maximum pollution success frequency
  - Bounded by δ/α

Security

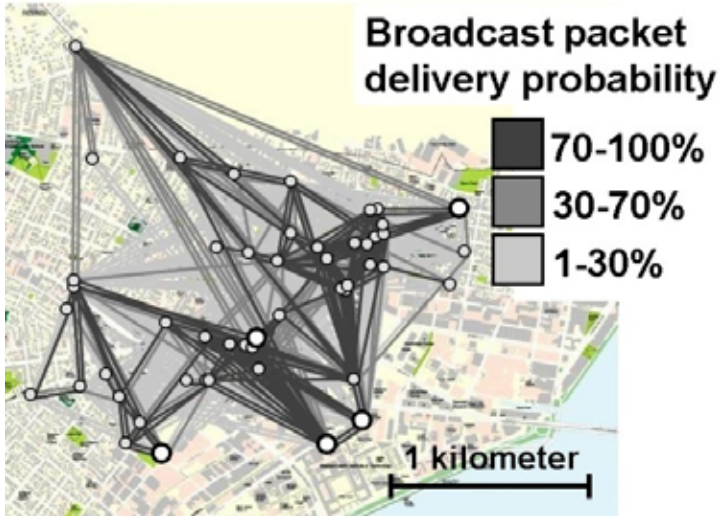
- Unnecessary delay
  - Nodes at i hops away from the attacker ($2 \cdot i \cdot \delta\text{-}h\text{-}1$): $\alpha(1 - (h+i)/\delta)$
  - Nodes more than δ-h-1 hops away: 0

Performance

The selection of δ and α trades off security and performance

Experimental Evaluations

- Network coding system: MORE
- Simulator: Glomosim
- Trace driven physical layer
  - MIT Roofnet trace

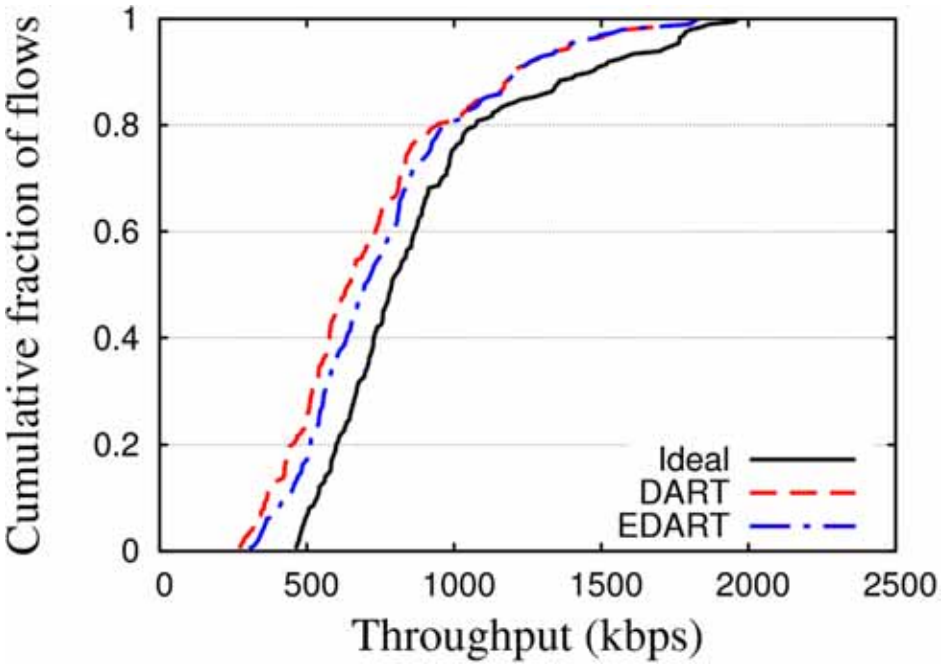


- MORE setup
  - GF(2^8), generation size 32, packet size 1500 bytes
- Defense setup
  - RSA-1024 digital signature
  - Checksum size parameter b = 2
  - EDART setup $\delta = 8$, $\alpha = 20$

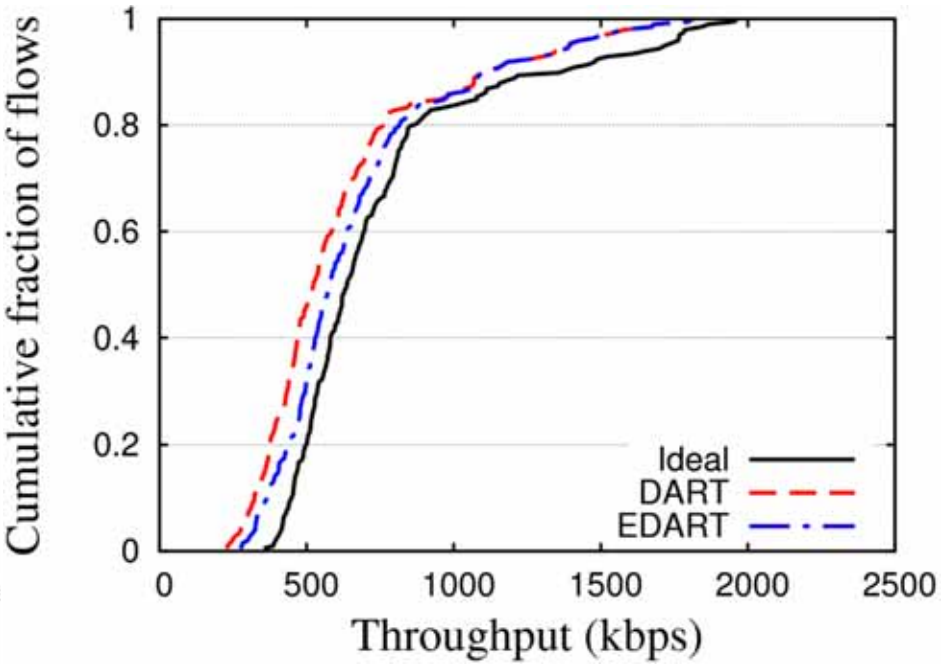
Effectiveness of DART and EDART

Ideal Defense: defense scheme that drops polluted packets with zero overhead

Defense under 5 attackers



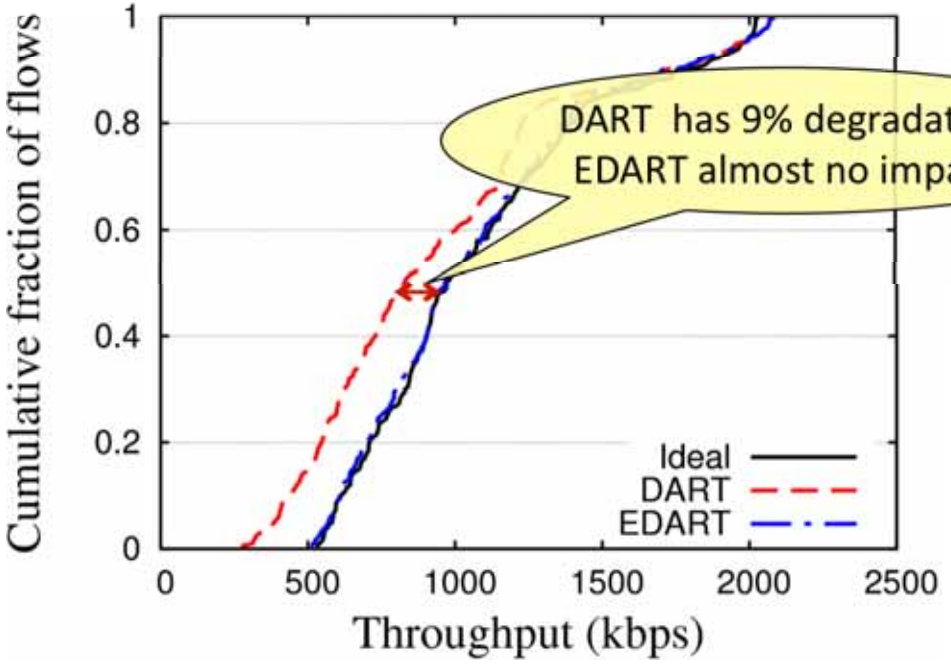
Defense under 10 attackers



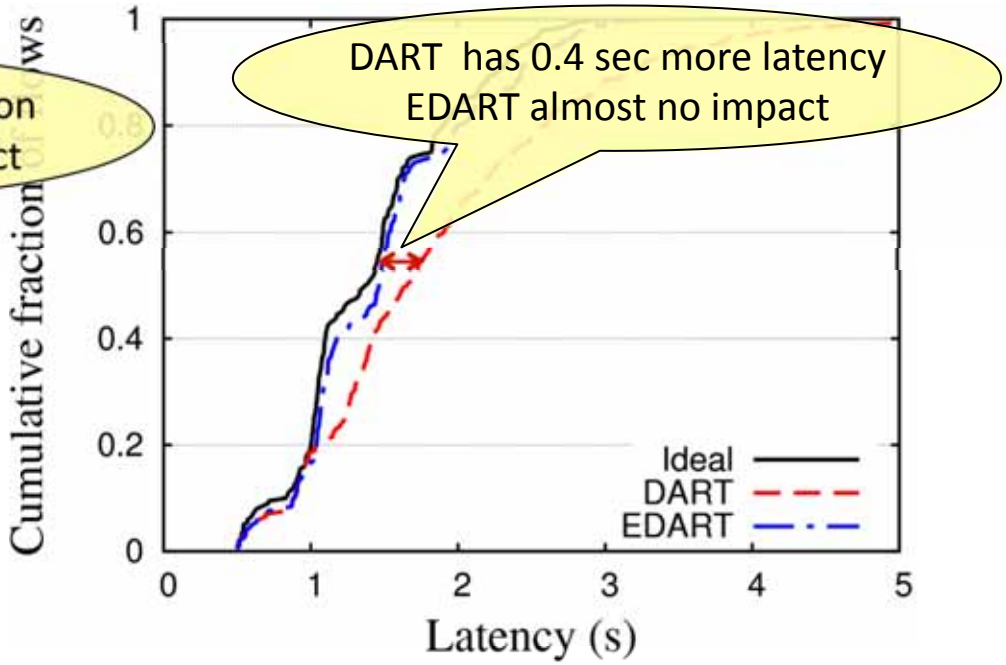
Both DART and EDART are effective against pollution attacks

Performance in Benign Networks

Throughput CDF



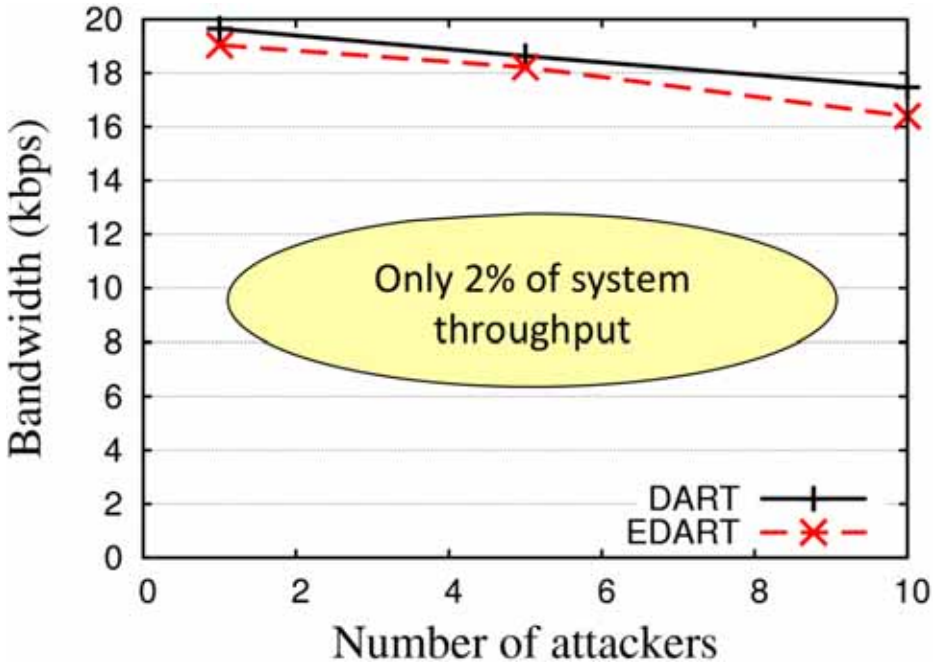
Latency CDF



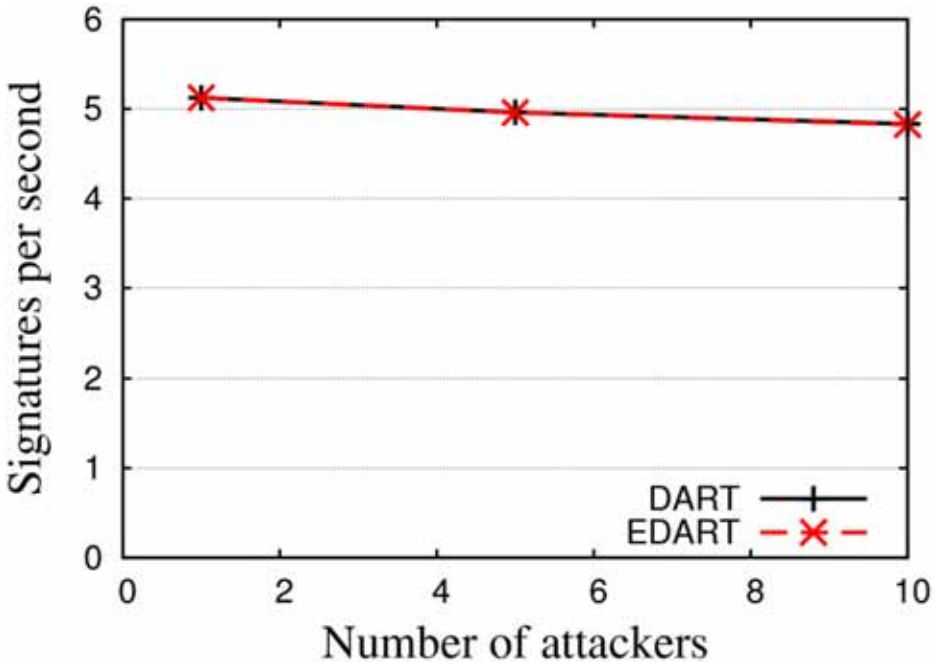
Both DART and EDART have good performance
EDART has almost zero performance impact

Overhead of DART and EDART

Bandwidth overhead



Computation overhead



Both DART and EDART incurs small bandwidth and computation overhead

Summary

- Network coding is a new paradigm for network protocol design for WMNs
- Network coding is vulnerable to a severe attack, known as the packet pollution attack
- We propose efficient and effective defenses against pollution attacks

Practical Defenses Against Pollution Attacks in Wireless Network Coding. J. Dong, R. Curtmola, and C. Nita-Rotaru. To appear in ACM Transactions on Systems and Information Security