Dynamic Energy-based Encoding and Filtering in Sensor Networks (DEEF)

Hailong Hou, *Cherita Corbett, Yingshu Li, Raheem Beyah

Georgia State University, Department of Computer Science

* Sandia National Labs

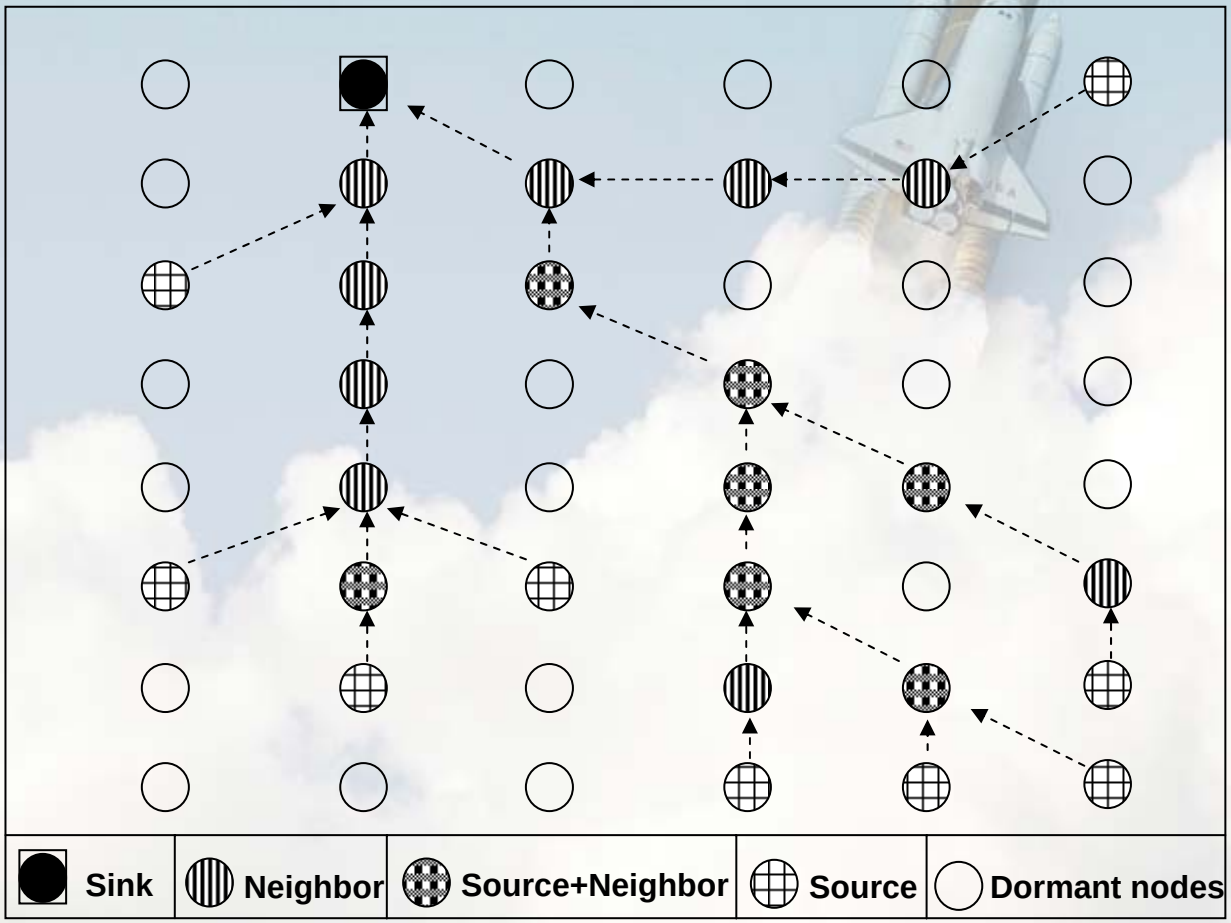
OUTLINE

- Motivation
- Overview
- Related Work
- Encoding Process
- Energy-Based Keying
- DEEF-T and DEEF-NT
- Performance Evaluation
- Conclusions
- Future Work

Motivation

WSNs
- Unattended
  - –Self organizing
- Limited capabilities
  - –Energy
  - –Computational
  - –Communication

For protocols
- Simplicity
- Effectiveness



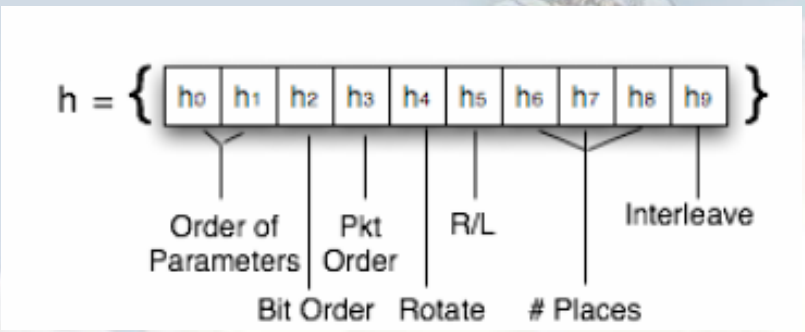
Related Work

- Tinysec introduced in [4] - link layer security architecture
- uTelsa proposed in [6] - comprehensive message authentication scheme
- Intrusion-tolerant routing (INSENS) was proposed in [12] - localize damage by establishing multiple disjoint paths
- Statistical En-route Filtering (SEF) proposed in [11] - probabilistically verify data authenticity and drops false data

*The aforementioned protocols require a significant amount of data packet overhead and/or an increase in message exchanges. We propose a lightweight technique for detecting outsider attacks, that consists of an encoding scheme, and dynamic energy-based keying, obviating the need to re-key

Encoding Process

- Instead of sending the hash code and the data, the resultant hash code is used locally at nodes
- The computed hash code is used to encode {ID,type,data}
- The hash code can be mapped to a set of actions on the data stream combination
- Each node sends its ID, type, data bits to its next hop
- The receiver node will perform the same hash operation



$$\text{Packet} = [\text{ID}, \{\text{ID}, \text{type}, \text{data}\}_k],$$

Tx-ed in clear

Hash function

As a result, we have a simple and effective encoding scheme that provides authenticity and integrity without redundant transmissions

Energy-based Keying

- Approach motivated by battery-based IDS [26-29] by Jacoby et.al.
- General Idea - Sensor nodes have limited states (Det, Tran, Enc, Dec, Idle, etc.)
- Compound states can be observed by downstream nodes that observe traffic
- Accordingly, watching nodes can infer the energy of the nodes they are observing
- This value changes every ***n*** transmission and is used as a shared key (seed)

Energy-based Keying (continued)

- How is the key computed?

$K_1 = F(E_c, IV)$ (1) (initial transmission)

$K_j = F(K_{j-1}, E_c)$ (2) (subsequent transmissions)

- *How is energy costs calculated (details on page 4 of paper)?*

$$F(E) = SUM(E_d, E_r, E_t, E_{enc}, E_{dec}, E_a)$$

Energy-based Keying (continued)

- We assume energy has been mapped to fairly coarse bins
- Authentication information stored in nodes
- Energy status is updated as packets from watched nodes pass the watching nodes
- Perceived and Bridge Energy is maintained (page 4)

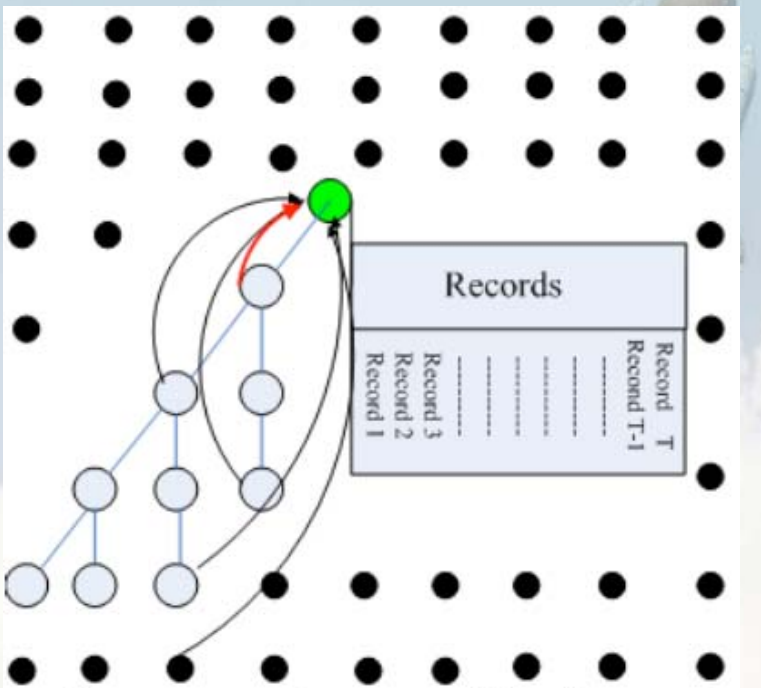


Figure 4. Record array stored in arbitrary node.

$$\textbf{perceived energy } E_p = E_{c(j-1)}$$

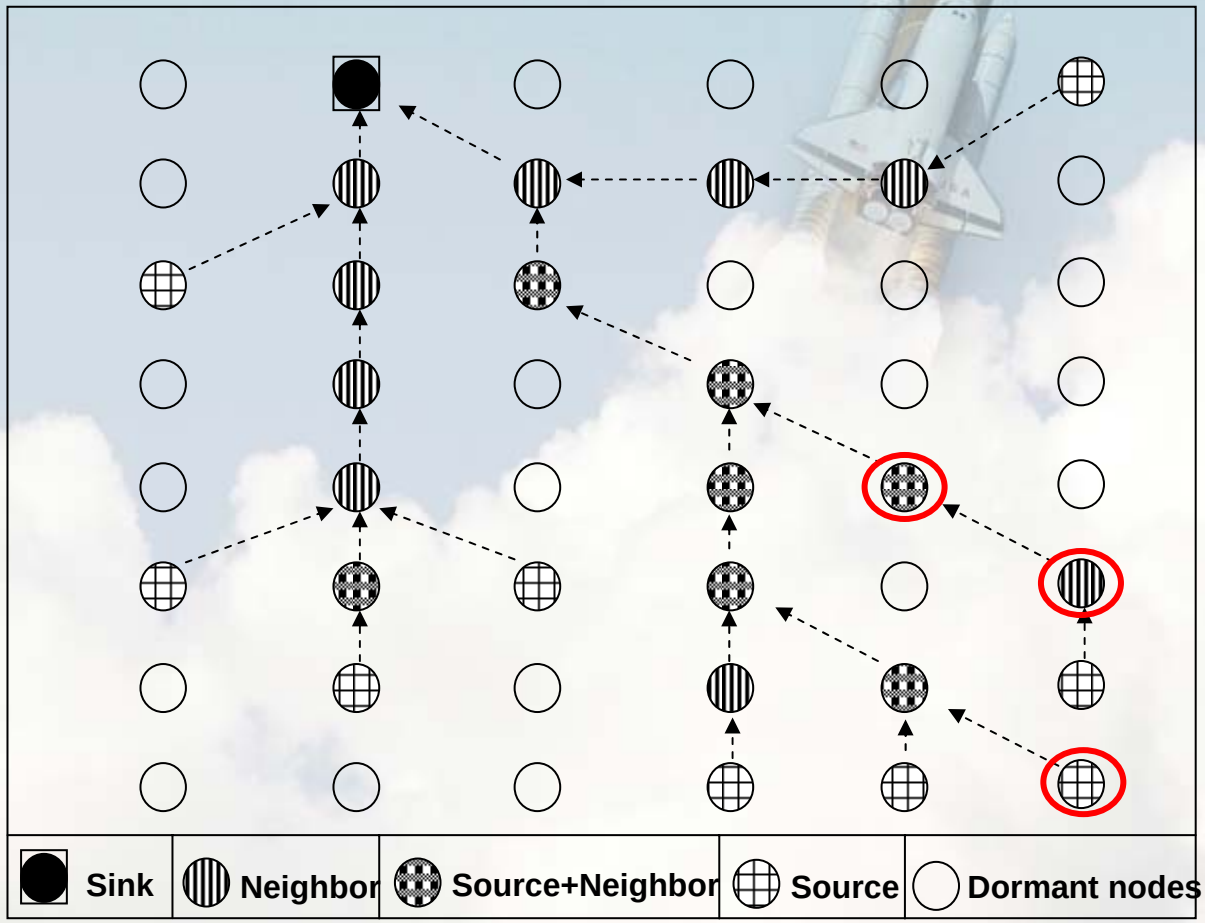
DEEF-T and DEEF-NT

- DEEF Traditional (DEEF-T)
  - Every node watches its peers
- DEEF Non-Traditional (DEEF-NT)
  - Every node probabilistically watches a set of predetermined nodes

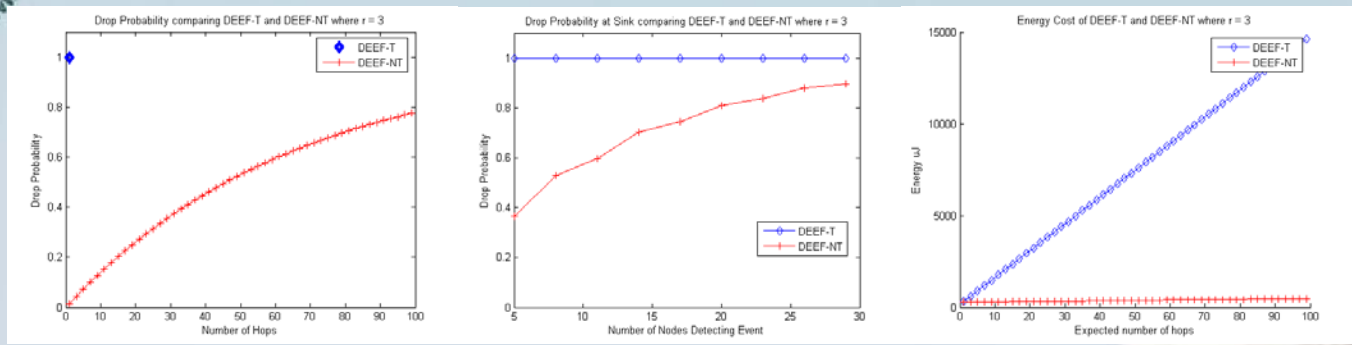
DEEF in Action→Topology-Revisit

- Node Responsibilities
  - Watcher
  - Forwarder
  - Originator
- Associated Energy Costs
  - Packet Tx (Etx)
  - Packet Rx (Erx)
  - Packet Sensing (Es)

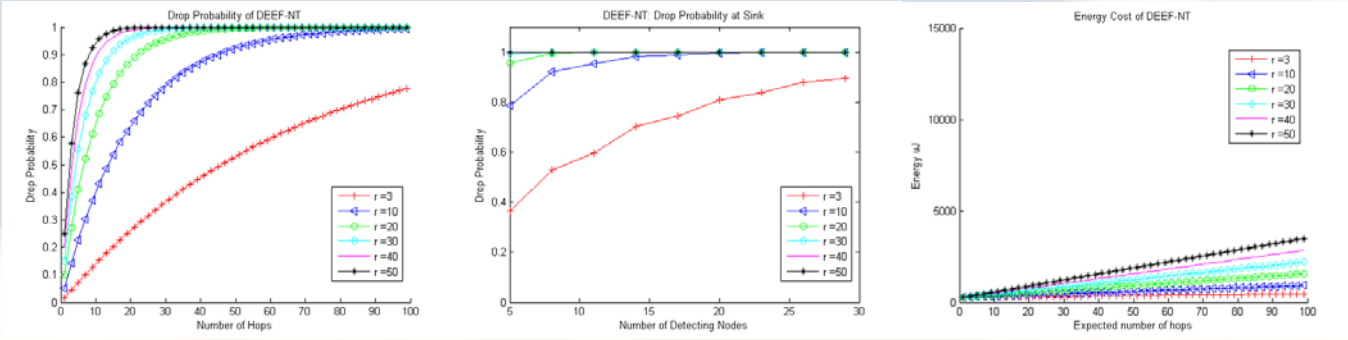
  (aka event detection (Ed))
  - Packet Encoding (Eenc)
  - Packet Decoding (Edec)
  - Keep-alive (Ea)
- Every node tracks its
  - Current Energy (Ec)
  - Perceived Energy (Ep)



Performance Evaluation



(a) (c) (e)



(b) (d) (f)

Figure 5. (a) Effectiveness of DEEF-T vs. DEEF-NT where $r = 3$; (b) Effectiveness of DEEF-NT with varying r; (c) Sink drop probability considering event detection set size DEEF-T vs. DEEF-NT where $r = 3$; (d) Sink drop probability considering event detection set size for DEEF-NT with varying r; (e) Average cost to send valid packet in DEEF-T and DEEF-NT where $r = 3$;
(f) Average cost to send valid packet in DEEF-NT with varying r.

Performance Evaluation - Observations (continued)

- While DEEF-T offers 99% drop probability of illegitimate packets at the first hop (Figure 5a), it more than doubles the amount of energy consumed for legitimate traffic in DEEF-NT at the fifth hop (Figure 5e)
- At the 15th hop, DEEF-T consumes 3 to 6 times more energy than DEFF-NT, and the separation of energy costs significantly increases as the network scales

Performance Evaluation - Observations (continued)

- When storage is a concern and DEEF-NT is watching the same number of nodes as DEEF-T (i.e., $r = 3$), DEEF-NT can achieve at best 77% drop probability at the 100th hop. At which point the illegitimate packet would have consumed 52% more energy than DEEF-T
- On the other hand, DEEF-NT can achieve at least a 90% drop probability within 9 hops if each node is watching at least 25% of the network ($r = 50$). Requiring more storage, however giving the benefit of a high drop probability within a few hops and lowering energy costs when dealing with legitimate packets

Performance Evaluation - Observations (continued)

- From this analysis, one could argue that DEEF-T is better suited where storage is the most critical resource and where the network tends to be less chatty, whereas DEEF-NT is most beneficial for chatty networks and/or when storage is not the crippling resource

Conclusion

The benefits of this scheme are the following:

- Since there is no hash code to transmit, the packet size does not grow unnecessarily, avoiding bandwidth overhead → increasing the overall network lifetime
- It is a simple technique → it is also viable for other resource constrained devices
- Albeit theoretically possible to insert malicious packets, as the packet size grows due to data bits, it is less likely
- The key changes dynamically → without re-keying.

•Downside of DEEF-NT is that it can be fooled if a compromised legitimate node inserts a malicious packet, which looks like a legitimate packet

Future Work

- Detect internal threats by employing intermediate nodes to utilize a weighted voting scheme
- Simulate more complicated scenarios where other than static routing is used
- Determine optimal bin size
- Address synchronization issues:
  - Borderline bin scenarios
  - Unreliable MAC, etc.
- Validate work via experimentation