Wormhole Attacks in Sensor Networks

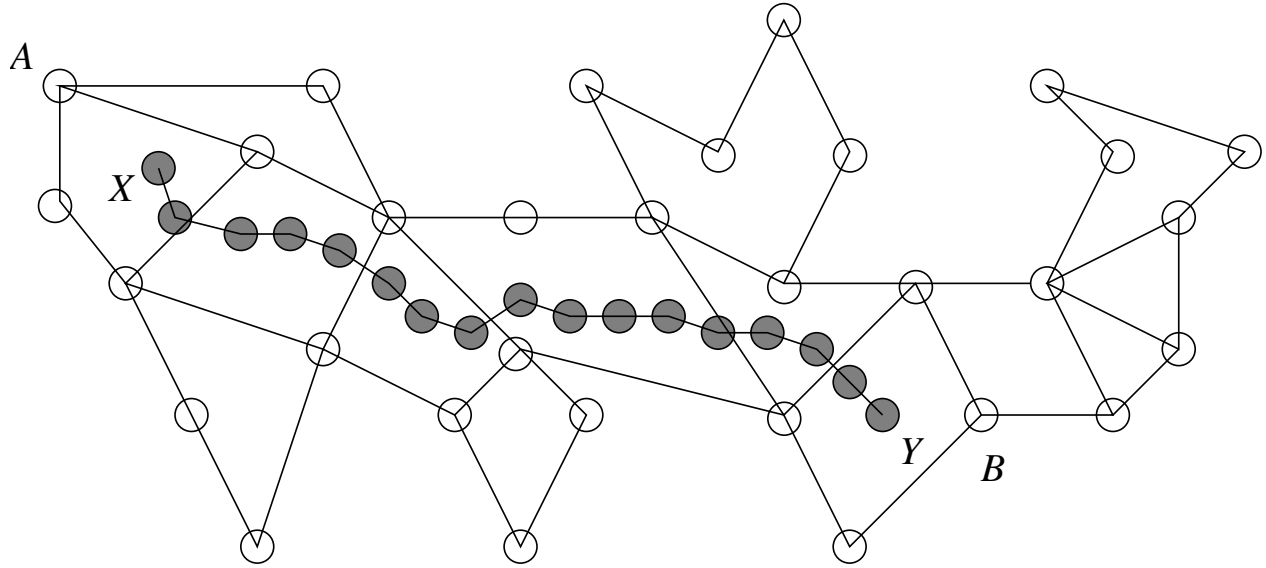
Introduction

- Wormhole Attacks
- Detecting Wormholes
- Preventing Wormholes with Directional Antennae
- Protocols

Wormhole Attacks

Wormhole Attacks

- A and B are not neighbors.
- The attacker can make A and B believe they are neighbors.



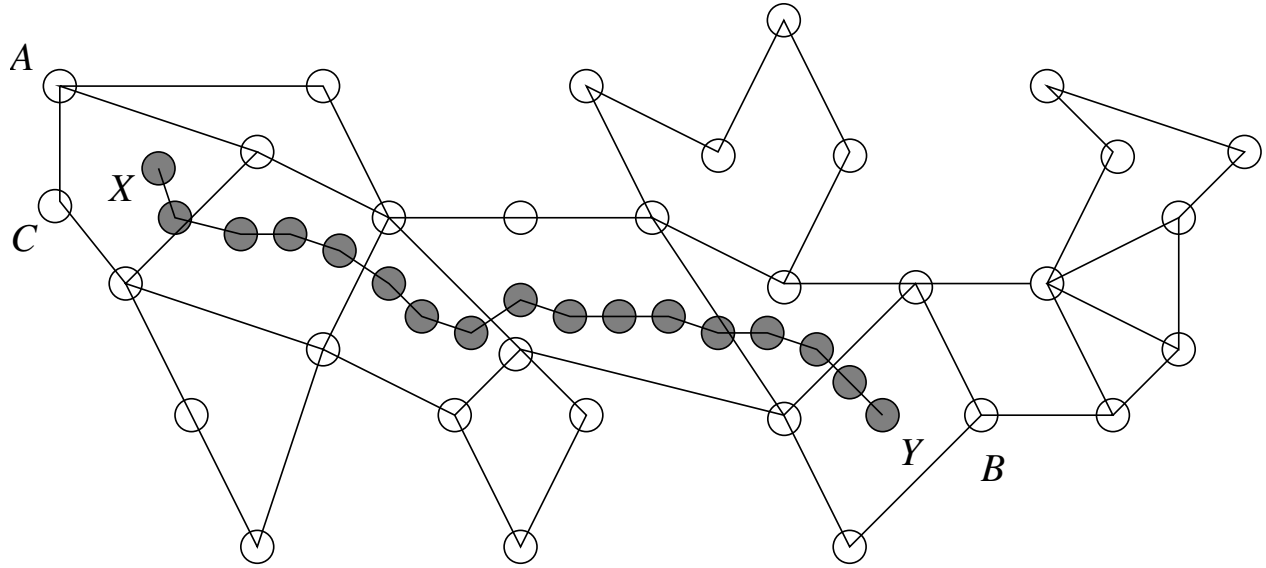
- The attacker replays packets received by X at node Y, and vice versa.

Wormhole Attacks

- In a wormhole attack, an attacker forwards packets through a high quality out-of-band link and replays those packets at another location in the network.
- The attacker replays packets received by X at node Y, and vice versa.
- If it would normally take several hops for a packet to traverse from a location near X to a location near Y, packets transmitted near X traveling through the wormhole will arrive at Y before packets traveling through multiple hops in the network.
- The attacker can make A and B believe they are neighbors by forwarding routing messages, and then selectively drop data messages to disrupt communications between A and B.

Impact on Routing Protocols: Beyond the Neighborhood

- For most routing protocols, the attack has impact on nodes beyond the wormhole endpoints' neighborhoods.



Impact on Routing Protocols: One-Hop Tunneling

- Node A will advertise a one-hop path to B so that C will direct packets towards B through A.
- For example, in on-demand routing protocols (DSR and AODV) or secure on-demand routing protocols (SEAD, Ariadne, SRP), the wormhole attack can be mounted by tunneling ROUTE REQUEST messages directly to nodes near the destination node.
- Since the ROUTE REQUEST message is tunneled through high quality channel, it arrives earlier than other requests.

Impact on Routing Protocols: Sinkholes and More

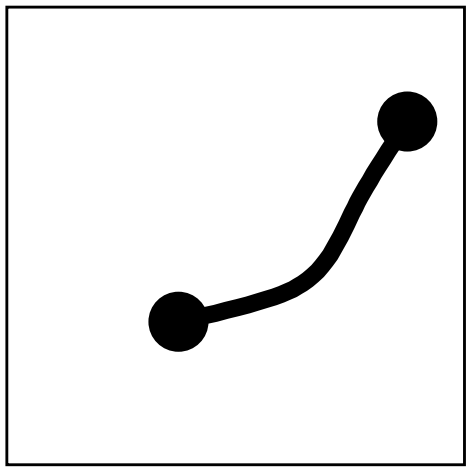
- Wormhole attacks prevent other routes from being discovered.
- The wormhole will have full control of the route.
- The attacker can discard all messages to create a denial-of-service attack, or more subtly, selectively discard certain messages to alter the function of the network.
- An attacker with a suitable wormhole can easily create a sinkhole that attracts (but does not forward) packets to many destinations.
- An intelligent attacker may be able to selectively forward messages to enable other attacks.

Impact on Sensor Networks: Disrupting Strategically

- An intelligent attacker may be able to place wormhole endpoints at particular locations.
- Strategically placed wormhole endpoints can disrupt nearly all communications to or from a certain node and all other nodes in the network.
- In sensor network applications, where most communications are directed from sensor nodes to a common base station, wormhole attacks can be particularly devastating.

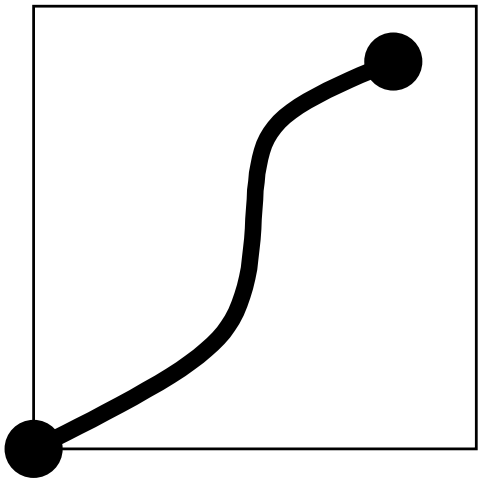
Impact on Sensor Networks: Location Matters

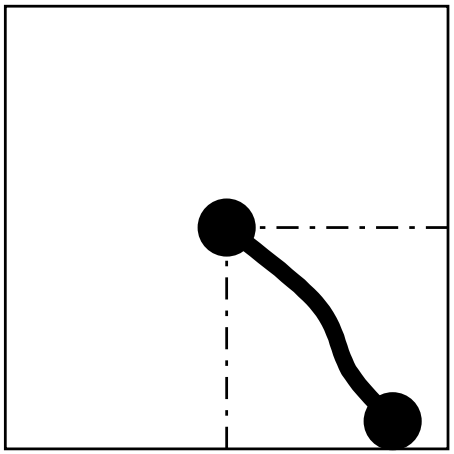
- In sensor networks traffic is directed from sensors to a base station.
- A wormhole can disrupt traffic depending on its location



Impact on Sensor Networks: Location Matters

- If the base station is at the corner of the network, a wormhole with one endpoint near the base station and the other endpoint one hop away (from base station) will be able to attract nearly all traffic from sensor nodes to the base station.
- If the base station is at the center of the network, a single wormhole will be able to attract traffic from a quadrant of the network.

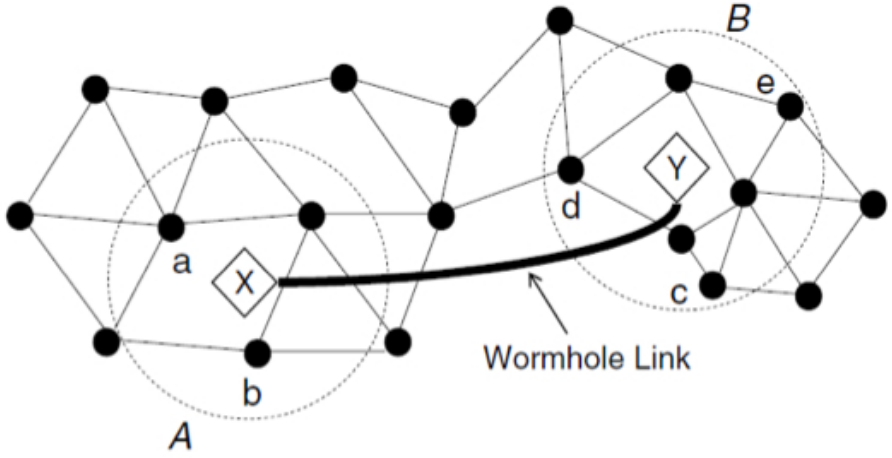




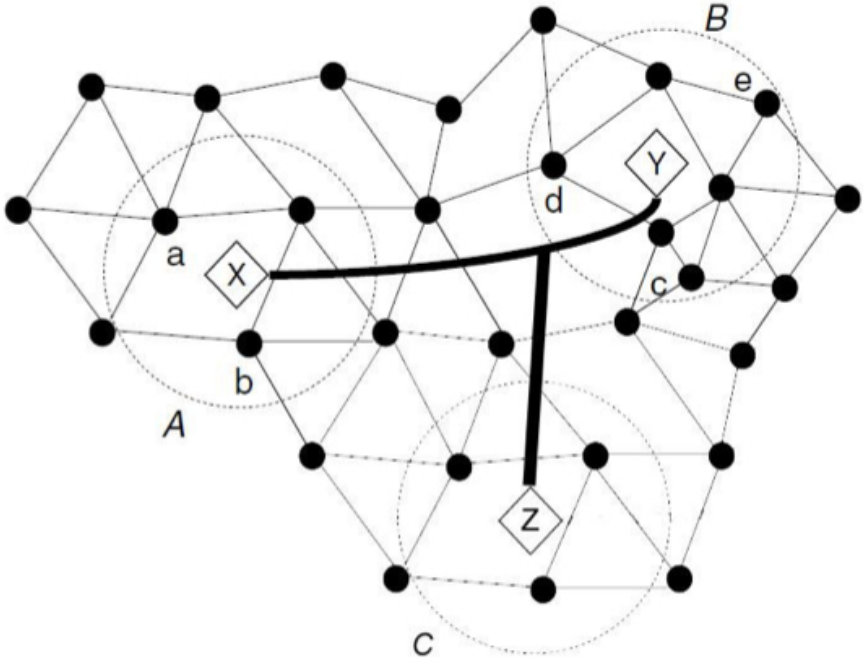
Detecting Wormholes

Wormhole: Example

- Let the network be represented by a graph, G.

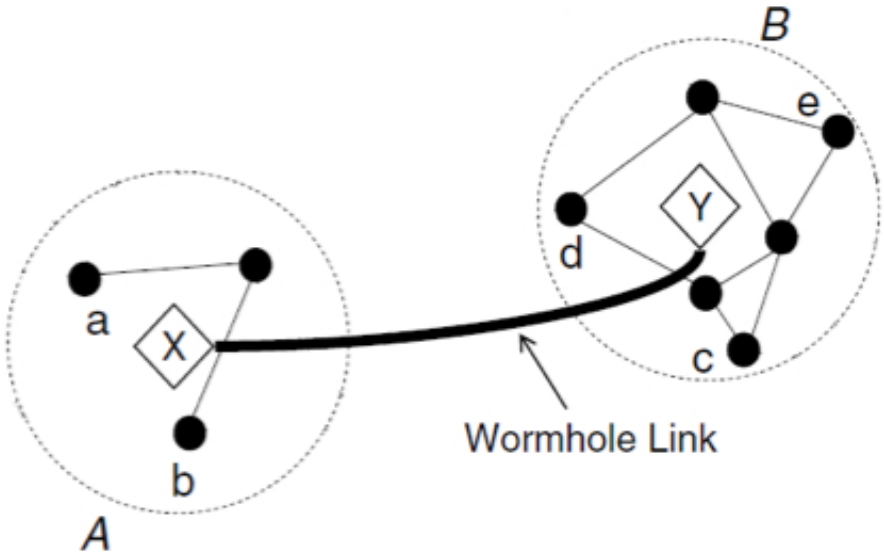


Multiple Wormholes: Example



Wormhole Subgraph

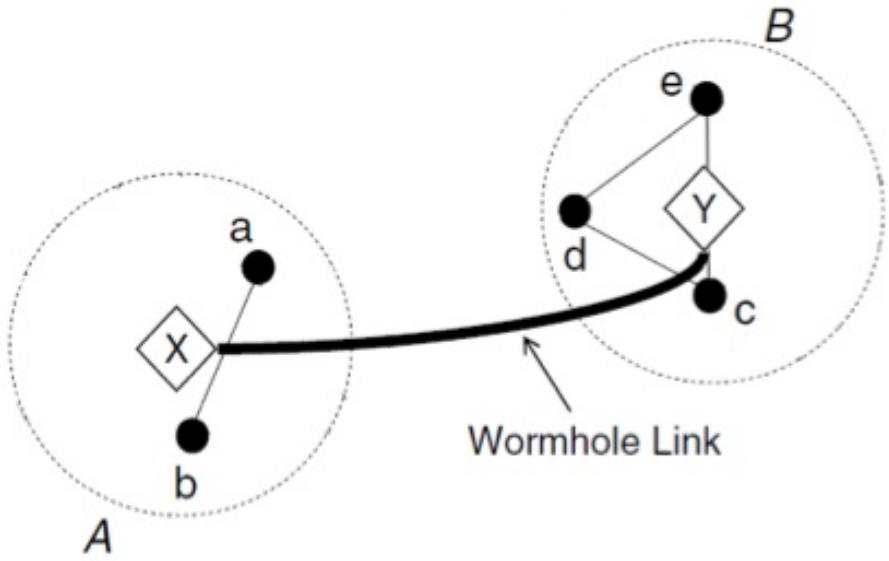
- Now consider the subgraph of G containing only nodes connected via a wormhole.



- Label 5 nodes in this subgraph $\{a, b, c, d, e\}$.

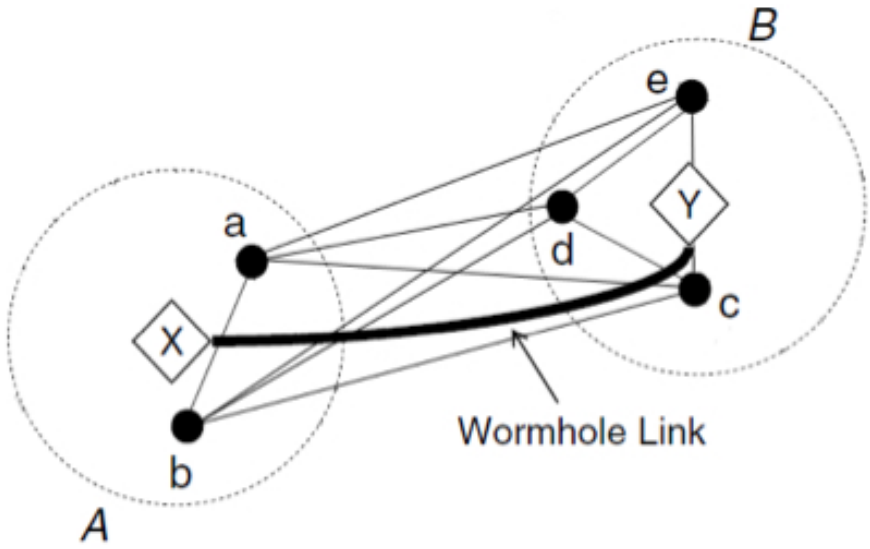
Edge Contraction

- Now contract the edges of the subgraph so that only the labelled nodes remain



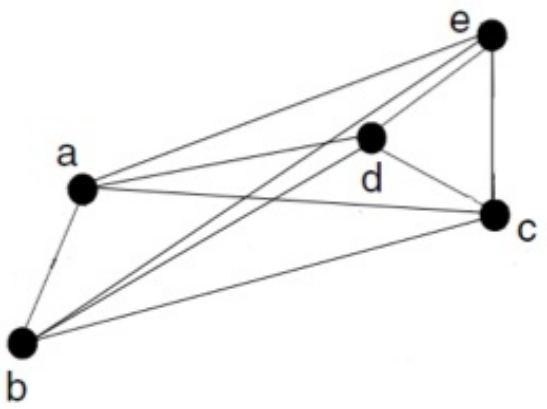
K_5

- If we now add the edges due to the wormhole connection we get K_5.



Contradicting Planarity

- But by Kuratowski's Theorem, this means that G is not planar.



- Therefore the existence of the wormhole has made the connectivity graph non-planar (and the routing algorithm, which requires planarity, will no longer work).

Altering the Topology: Impossible Graphs

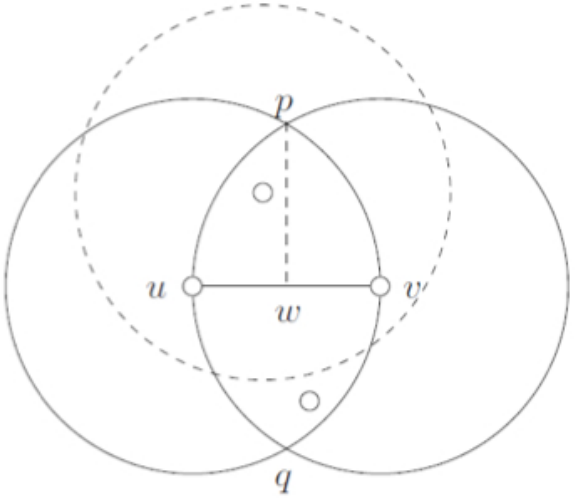
- Core problem in discovering wormholes: identifying neighbours who would not be if the wormhole did not exist.
- Two main approaches to accomplish this task:
  1. those that attempt to make the determination based solely on connection information, and
  2. those using in part location awareness of the nodes (even if only within a neighbourhood) and determine if arrangement of nodes is possible.

Test for Impossible Graphs

- Perform neighbour discovery (ND) and run a planarization algorithm.
- If we discover there are wormhole links and removed these links, the graph could become disconnected.
- This is the reason wormhole discovery is performed during ND, and we will refer to such an algorithm as secure neighbour discovery (SND).

Hop-1 Test

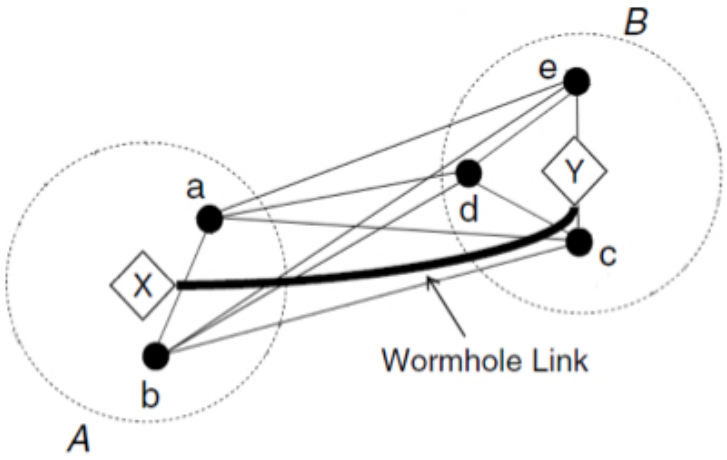
- Consider the process of ND between two neighbours u and v



- If u and v are neighbours, and not connected by a wormhole, they can have at most 2 independent neighbours in common

An Impossible Graph

- In the presence of a wormhole, two nodes (a and b if they were connected) can have three independent neighbours (c, d and e).



Hop-k Tests, $k \geq 2$

- Just because two nodes do not have 3 or more independent neighbours in common does not mean that they are not being affected by a wormhole.
- Therefore if no wormhole is detected within a 1-hop neighbourhood, we must examine the 2-hop neighbourhood and determine how many 2-hop independent neighbours these nodes have in common.
- Such tests depend on the density of the wireless network and may not always be feasible. [a]

[a] Detecting wormhole attacks in wireless networks using connectivity information R. Maheshwari, J. Gao, Samir Das, INFOCOM 2007

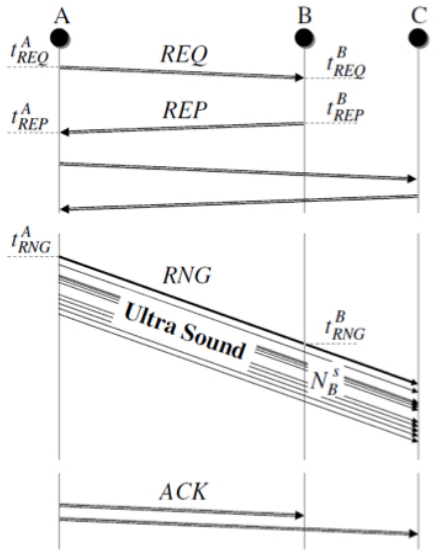
Using Time Difference of Arrival

- The SND proposed in [a] requires that nodes are equipped with microsecond precise clock (which is likely to be required in the node anyways) and an ultrasonic (UF) transceiver, which is not an especially expensive or power-consuming device.
- The algorithm proposed is localized to a 1-hop neighbourhood.
- It is assumed that all node share keys so that each node can identify and authenticate itself.

[a] A Practical Secure Neighbor Verification Protocol for Wireless Sensor Networks Reza Shokri, Marcin Poturalski, Gael Ravot, Panos Papadimitratos, and Jean-Pierre Hubaux Proceedings of the second ACM conference on Wireless network security, 2009

Neighbor Discovery (1/2)

- Each node (A) will broadcast a probe message (REQ).
- Neighbours of A will respond to this message and identify and authenticate themselves.



- This stage is used to eliminate attacking nodes (but will not prevent a wormhole attack).

Neighbor Discovery (2/2)

- After a suitable amount of time, A will broadcast a UF message and then begin sending messages to each verified neighbour indicating the time of events as recorded by A $(t^A_{REQ}, t^A_{REP}, t^A_{RNG})$ and encrypt this message specifically for the recipient.
- Each message will contain different t^A_{REP} values depending on when the response arrived from the specific neighbour. With this information, the neighbours of A can estimate their distance from A by the time difference of arrival. Since all the neighbours of A will be doing the same, A will eventually have an estimate for its distance to all its neighbours.
- Once it has all the estimates, it broadcasts its 1-hop neighbourhood–including its distance estimates–to all its neighbours.

Three Tests: A Validates Link (A, B) for each Neighbor B

1. **Symmetry:** A confirms that $d(A, B) = d(B, A)$.
2. **Maximum Range Test:** Assuming nodes know their range R. If a neighbour lies beyond this range, the link must be across a wormhole, so $d(A, B) \leq R$
3. **Quadrilateral Test:** For any link (A, B) find two nodes D and C, such that A, B, C, D form a 4-clique. If there is no wormhole, then is should be possible to arrange all four nodes so that they form a quadrilateral.

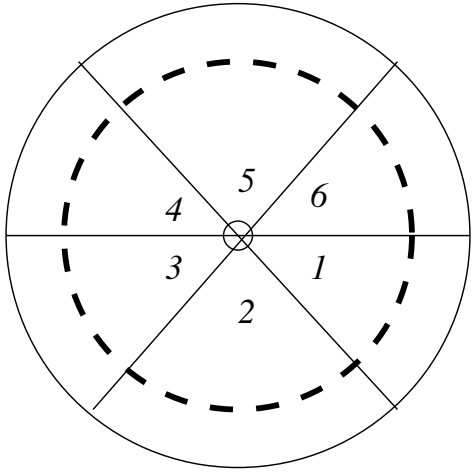
Preventing Wormholes: Directional Antenna Model

Directional Sensors and Zones

- The range of an antenna is divided into n zones.
- Each zone has a conical radiation pattern, spanning an angle of $2\pi/n$ radians.
- The zones are fixed with non-overlapping beam directions, so that the n zones may collectively cover the entire plane.
- When a node is idle, it listens to the carrier in omni mode.
- When it receives a message, it determines the zone on which the received signal power is maximal. It then uses that zone to communicate with the sender.

Directional Sensors and Zones

The zones are numbered 1 to 6 oriented clockwise starting with zone 1 facing east.



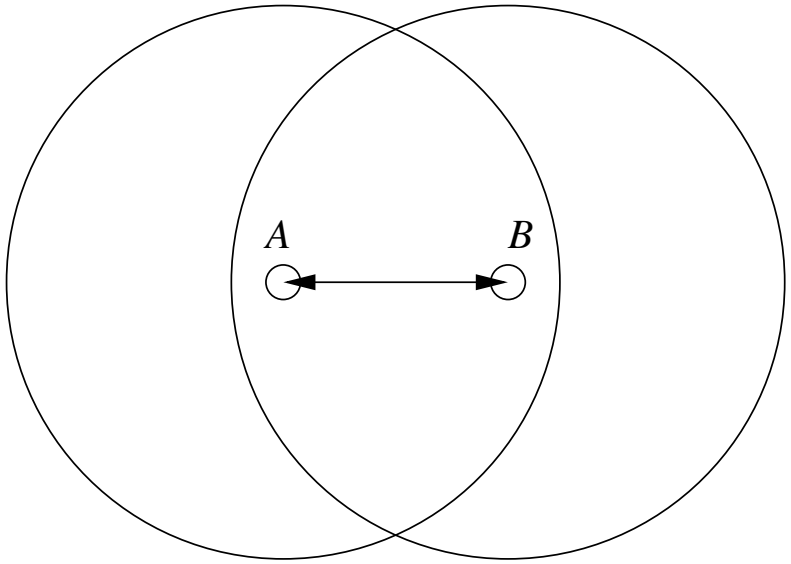
This orientation is established with respect to the earth's meridian regardless of a node's physical orientation. This is achieved in modern antennas with the aid of a magnetic needle that remains collinear to the earth's magnetic field. It ensures that a particular zone always faces the same direction.

Sending/Receiving

- **Receiving:**
  a node can receive messages from any direction.
- **Sending:**
  a node can work in omni or directional mode.
  - In omni mode signals are received with a gain G^o, while in directional mode with a gain of G^d.
  - Since a node in directional mode can transmit over a longer distance, $G^d > G^o$.

Assumptions on Security

- All communication channels are bidirectional: if A can hear B, then B can hear A.



- A mechanism is available to establish secure links between all pairs of nodes and that all critical messages are encrypted.
- Sensor network must be "relatively" dense.

Notations

- $A, B, C, \ldots$: Legitimate nodes
- X, Y: Wormhole endpoints
- R: Nonce
- $E_{KAB}(M)$: M encrypted with key shared by nodes A and B
- $zone$: The directional element, which ranges from 1 to 6.
- $\overline{zone}$: The opposite directional element. For example, if $zone = 1$ then $\overline{zone} = 4$.
- $zone(A, B)$: Zone in which node A hears node B
- $neighbors(A, zone)$: Nodes within one (directional distance) hop in direction zone of node A.

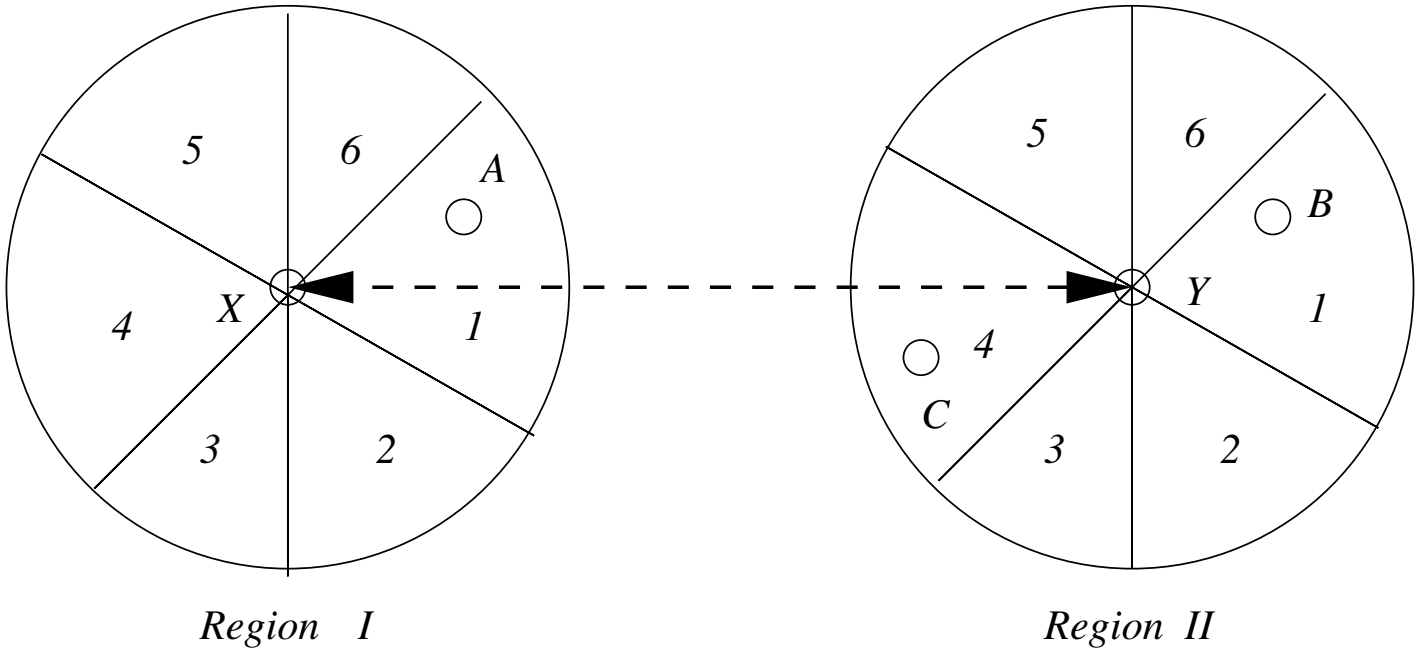
Protocols

Protocol 1: Directional Neighbor Discovery

- 1. $A \rightarrow Region$: HELLO and ID_A.
- 2. $N \rightarrow A : ID_N | E_{KNA}(ID_A | R | zone(N, A))$.
  All nodes that hear the HELLO message send their node ID and an encrypted message to the announcer. The encrypted message contains the announcer's ID, a random challenge nonce, and the zone in which the message was received.
- 3. $A \rightarrow N$: R.
  A decrypts message and verifies that it contains its node ID. It verifies $zone(A, N) = \overline{zone}(N, A)$. If correct, it adds the sending neighbor to its neighbor set for $zone(A, N)$. If message was not received in the appropriate zone, it is ignored. Otherwise, the announcer transmits the decrypted challenge nonce to the sending neighbor. Upon receiving the correct nonce, the neighbor inserts the announcer into its neighbor set.

Wormhole Vulnerability of Protocol 1

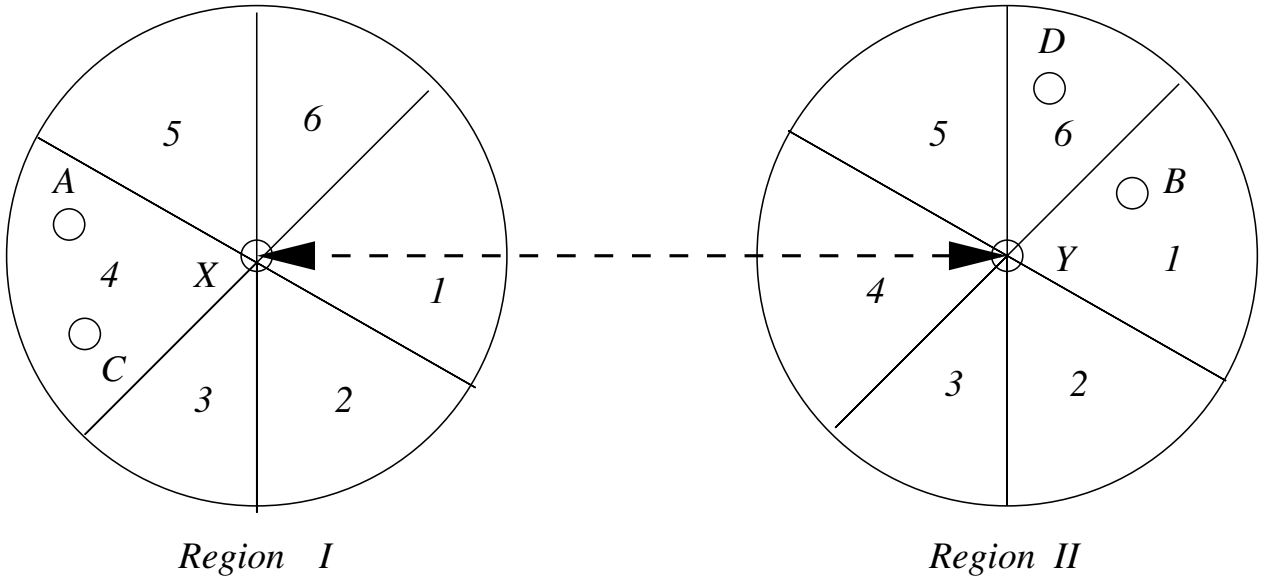
An attacker with a wormhole can establish a false distant neighbor.



The adversary establishes a wormhole between X and Y, and can trick A and C into accepting each other as neighbors by forwarding messages since they are in opposite zones relative to the respective wormhole endpoints.

Further Problems with Protocol 1

B will hear A and C from the west through the wormhole ($zone(B, A) = zone(B, C) = 4$), and C will hear A directly from the east ($zone(A, C) = \overline{zone}(C, A) = 1$) and C will hear B from the west through the wormhole ($zone(C, B) = \overline{zone}(B, C) = 4$).



Mitigating Wormhole Attacks

If nodes cooperate with their neighbors they can prevent wormholes since the attacker will only be able to convince nodes in particular regions that they are neighbors.

Assume the adversary has one transceiver at each end of the wormhole.

An adversary can only trick nodes that are in opposite directions from the wormhole endpoints into accepting each other as neighbors.

Hence, nodes in other locations can establish the announcer's legitimacy.

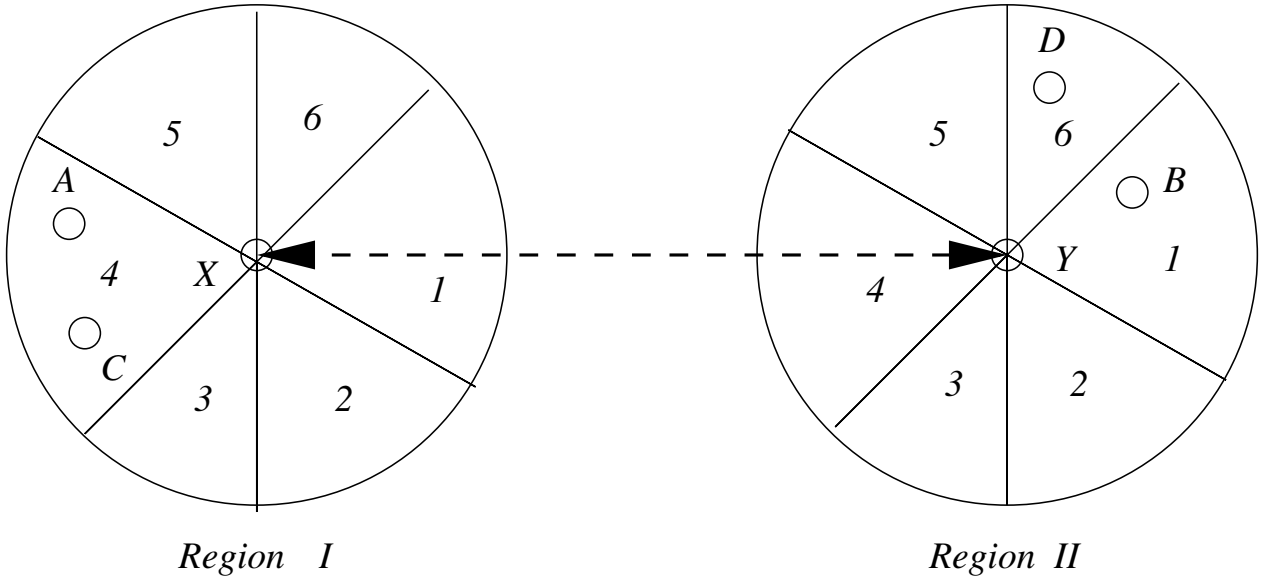
Such nodes are called verifiers.

Introducing Verifiers

How do we prevent verifiers from acting through the wormhole?

Node C cannot act as a verifier for the link AB since the wormhole attacker could make a node appear on the other end of the wormhole.

Node D could act as a verifier, since it satisfies the verifier properties.



Verifiers

A valid verifier V for the link $A \leftrightarrow B$ must satisfy the following properties:

1. $zone(B, A) \neq zone(B, V)$.
   Node B hears V in a different zone from node A, hence it knows A and V are in different locations, and both cannot be coming through a single wormhole endpoint.

2. $zone(B, A) \neq zone(V, A)$.
   Node B and V hear node A from different directions. A wormhole can deceive nodes in only one direction. So if both B and V are directionally consistent with A in different directions ($zone(B, A) = \overline{zone}(A, B)$ and zone $(V, A) = \overline{zone}(A, V)$), then they know A is not being retransmitted through a wormhole.

Protocol 2: Verified Neighbor Discovery

First three steps 1-3 are exactly as in Protocol 1.

- 4. $N \rightarrow Region$: $INQUIRY|ID_N|ID_A|zone(N,A)$
  All neighbor nodes that hear the HELLO message broadcast an inquiry in directions except for the received direction and opposite direction.

  So, if N received the announcement in zone 1, it will send inquiries to find verifiers to zones 2, 3, 5 and 6.

  The message includes $zone(N,A)$, so prospective verifiers can determine if they satisfy the verification properties by having heard A in a different zone.

Protocol 2: Verified Neighbor Discovery

- 5. $V \rightarrow N$: $ID_V | E_{KNV}(ID_A | zone(V, N))$
  Nodes that receive the inquiry and satisfy the verification properties respond with an encrypted message.

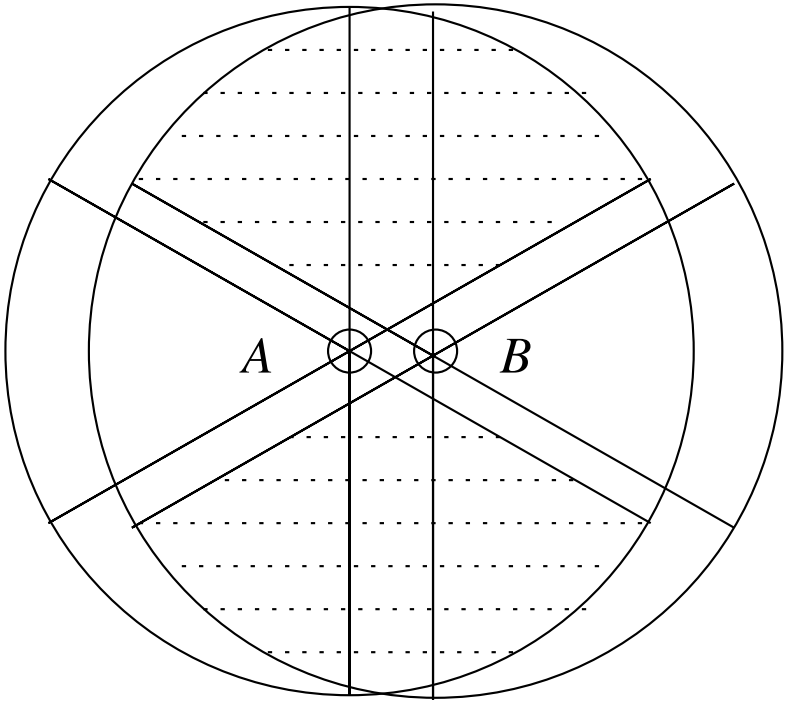
  This message confirms that the verifier heard the announcement in a different zone from N and has completed steps 1-3 for the protocol to authenticate A and its relative position.

  To continue the protocol, N must receive at least one verifier response. If it does, it accepts A as a neighbor, and sends a message to A:

- 6. $N \rightarrow A$: $ID_N | E_{KAN}(ID_A | ACCEPT)$
  After receiving the acceptance messages, the announcer adds N to its neighbor set.

Verifier Region

The shaded area is the verifier region of nodes A and B in verified neighbor discovery protocol.



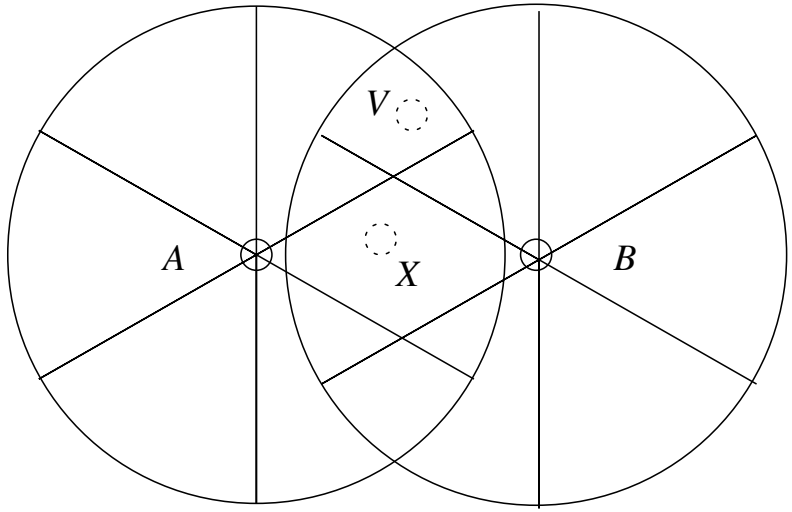
If there is a node in the shaded region, it can act as a verifier for A and B.

And Now the Density!

- Now you can see why you need the sensor network to be dense.
- It is required that with high probability there is a verifier node in the shaded region so as to enable A and B to have a successful protocol verification.
- The shaded region determines a given a area. The probability must be sufficiently high that sensors lie within this region so as to act as verifiers!
- The verifier region may still exist when two nodes are slightly out of radio range, and a smart adversary can use this to make them to be neighbors.

Worawannotai Attack: Wormhole Vulnerability of Protocol 2

Node B is located just beyond the transmission range of node A.

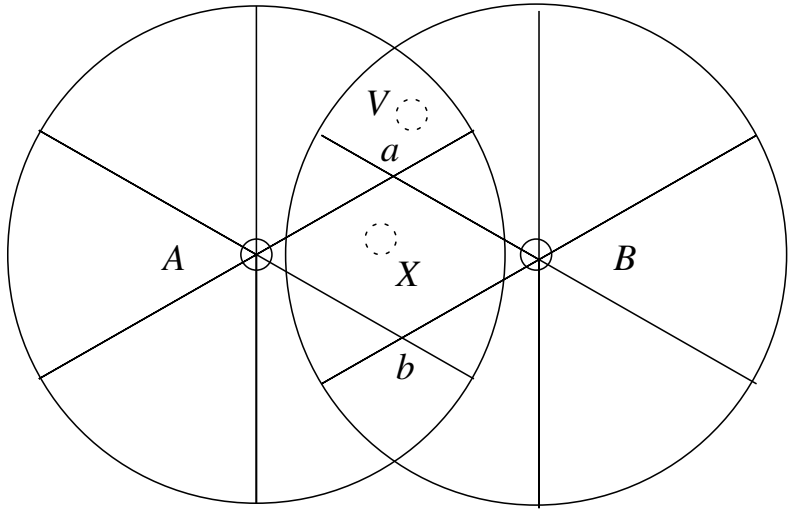


If there is a valid verifier in those areas, the attacker can just put one node in between A and B (node X) and use it to listen to and retransmit messages between A and B.

Nodes A and B will mistakenly confirm they are neighbors using verifier V, but the attacker will have control over all messages between A and B.

Preventing the Worawannotai Attack

There are two areas (a, b) that could have valid verifier for this protocol. If there is a valid verifier in those areas, the attacker can just put one node in between A and B (node X) and use it to listen to and retransmit messages between A and B.



A and B mistakenly confirm they are neighbors using verifier V, but the attacker will have control over all messages between A and B.

Protocol 3: Strict Verification Rules

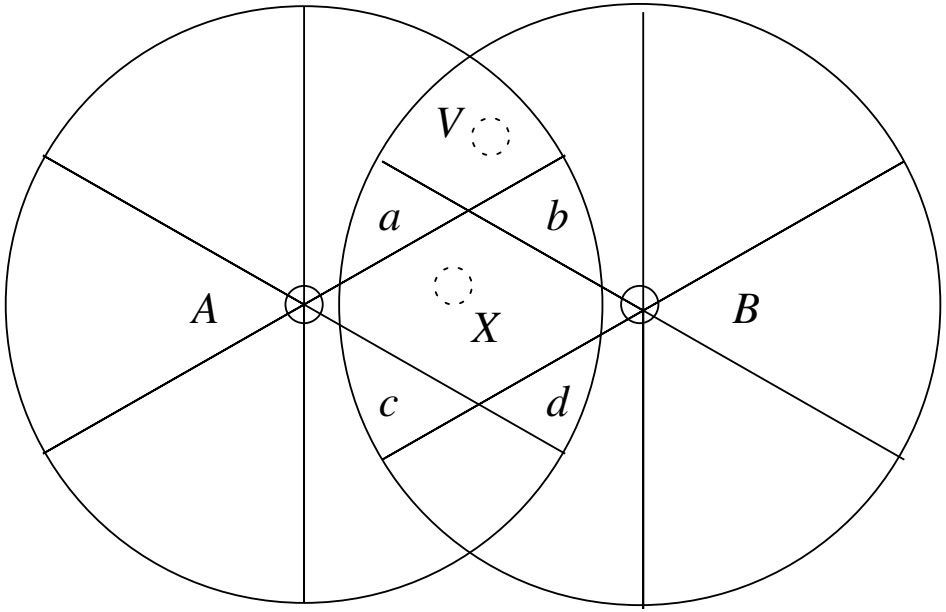
In the strict protocol, a valid verifier V for the link $A \leftrightarrow B$ must satisfy three properties:

1. $zone(B, A) \neq zone(B, V)$.
2. $zone(B, A) \neq zone(V, A)$.
3. $zone(B, V)$ cannot be both adjacent to $zone(B, A)$ and adjacent to $zone(V, A)$.

The first two conditions are the same as previous protocol, and they guarantee that the adversary cannot replay the confirmation message from verifiers. The third condition ensures that the verifier region is empty when two nodes are out of radio range, so the adversary cannot use this to conduct Worawannotai attack.

Protocol 3: Strict Neighbor Discovery

The verifier region determined by the previous three rules is depicted by the four regions a, b, c, d.



These areas are the verifier region's of node A and B in strict neighbor discovery protocol