Emulation of RDRN on an ATM-Testbed and a Comparative Evaluation of IP vs ATM

Syed Fazal Ahmad

Organization

- Introduction to RDRN
- Motivation
- Requirements
- Emulation Environment
- Scenarios
- Conclusion
- Future Work

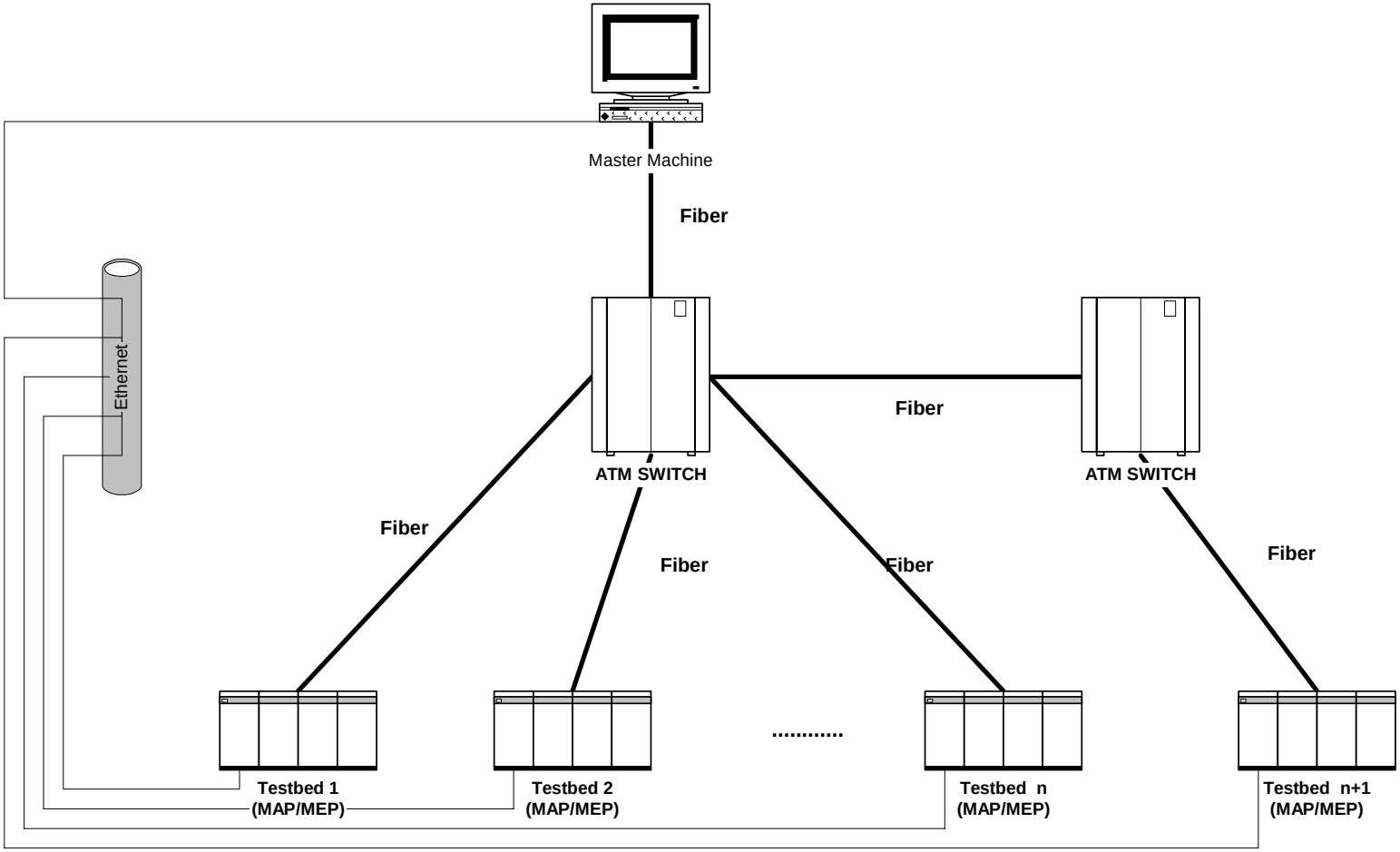
Introduction to RDRN

- Rapidly Deployable Radio Network is
  - Multi-hop Wireless ATM Network
  - Highly Dynamic Networking Environment
- RDRN consists of
  - a low bandwidth, low frequency, high reliability, omni-directional orderwire link, for node discovery and topology configuration
  - a high bandwidth radio link for high speed data transfer.
- RDRN consists of two types of Nodes
  - Mobile Access Point (MAP)
  - Mobile End Point (MEP)

Motivation

- To perform large-scale tests for the RDRN
- To measure the scalability of the Network Controller
- Three options
  - Use a network simulator & implement the system in it
  - Field Tests
  - Emulation Environment - existing software can be used with minimal changes
- Chose to provide an emulation environment
  - Isolate the actual radios (radio controller)
  - provide an alternate mode of connectivity
- To do an initial comparative evaluation of IP vs ATM

The Physical Connectivity of the Testbeds

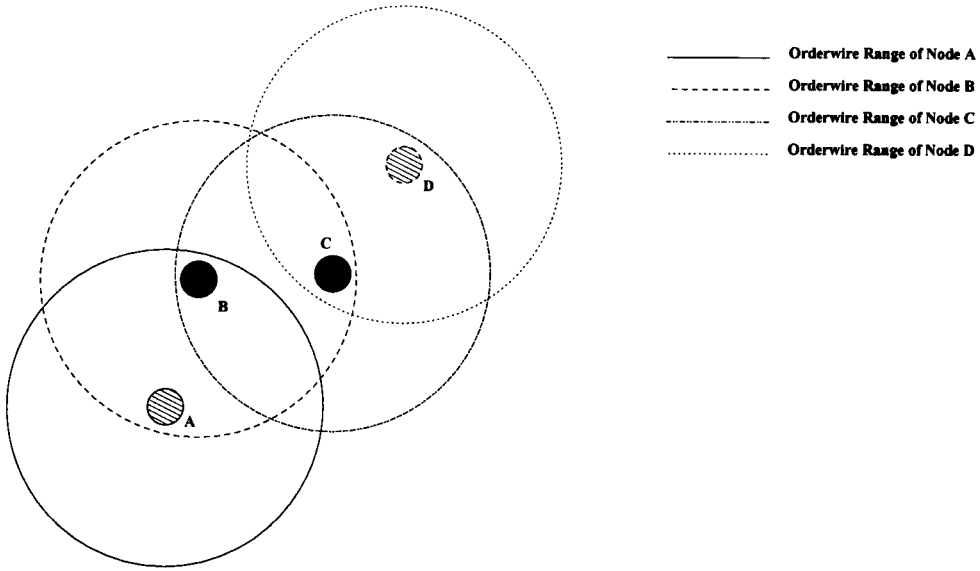


Software Modules

- Orderwire Module
  - Set up the topology
  - Create the High-Speed Point-to-Point Connectivity
- WATM Module
  - Mix of user-level and kernel drivers embedded in the Linux-ATM
  - Has a defined protocol stack
  - Linux-ATM provides native-mode ATM as well as TCP/IP over ATM
- Routing Module
  - Wireless Multi-path Routing Protocol (WMRP)

Identification of Requirements
Steps in a Field Scenario

- Step 1: Exchange of Information Over the Orderwire



!As soon as the nodes come up they retrieve their location from the GPS receiver
!Broadcast their position over the orderwire

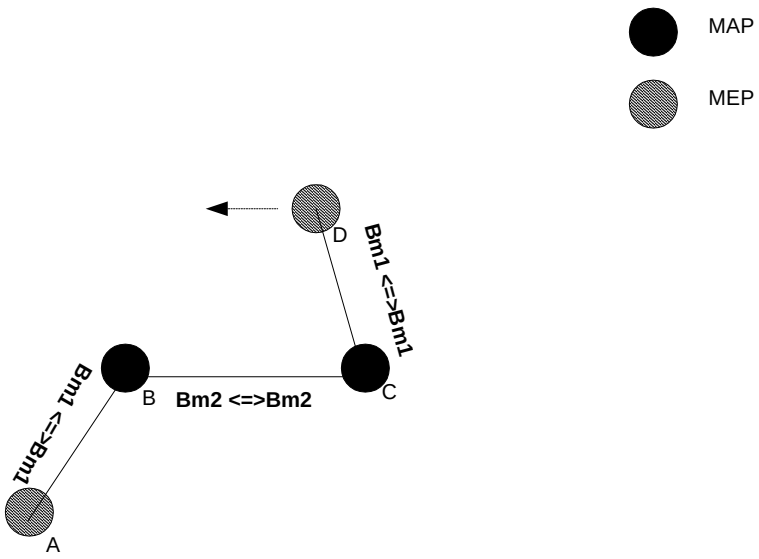
Requirements:
!Emulate the GPS receiver
!Ability to broadcast the orderwire packets to the other nodes within the orderwire-range

Requirements/Solution

- Node Motion and Location
  - Orderwire Module opens a UDP socket to the Emulation Manager (EM)
  - The EM sends the individual GPS locations to each of the nodes every 1.8 seconds
- Broadcast of the Orderwire Packets
  - Orderwire Module opens a UDP socket to the Emulation Manager (EM)
  - Orderwire sends the packets to the EM on the above socket
  - The EM re-transmits the same datagram to zero or more nodes which are within the orderwire-range or if the topography permits

Identification of Requirements
Steps in a Field Scenario

- Step 2: Establishment of Network Topology & High-Speed Connectivity



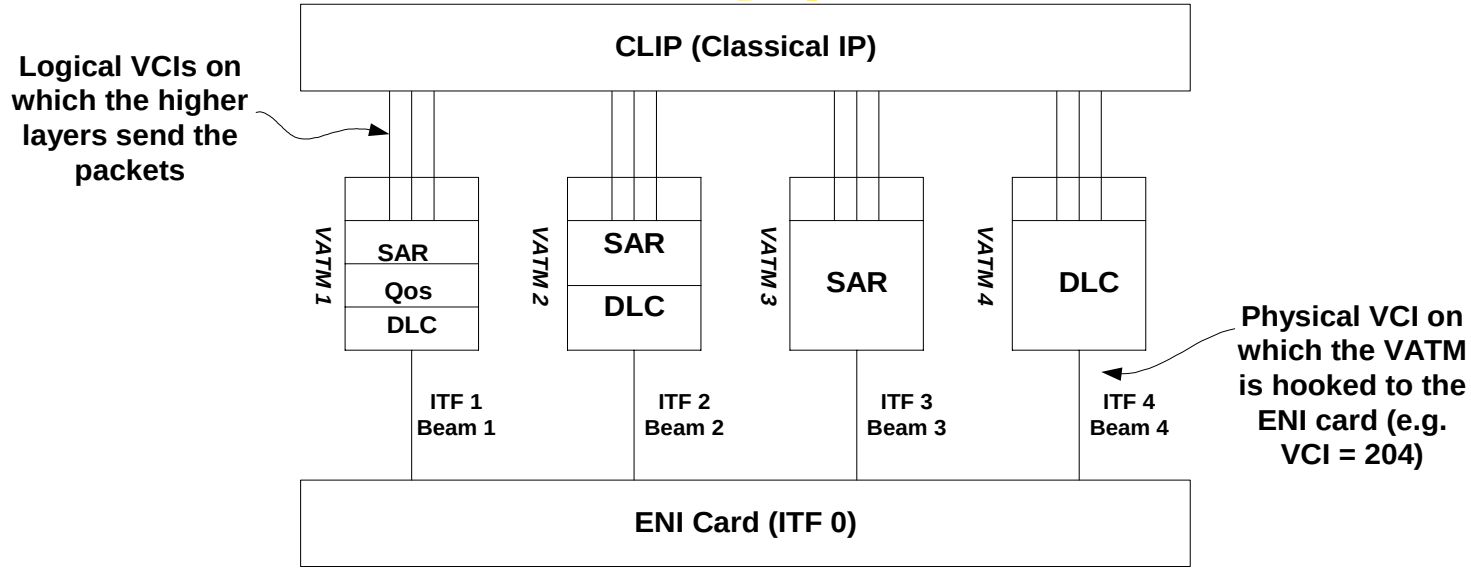
!After hearing from the other nodes, the topology algorithm is executed
!Topology algorithm works differently on the MAPs and the MEPs

Requirements:
!Mechanism to emulate the beams on the ATM-testbed
!Ability to multiplex at the source the traffic for different destinations on the same beam; and the ability to de-multiplex at the destination or the intermediate nodes
!Mechanism to establish and tear-down the beams between the neighbors because they getting out-of-range or the topography

Requirements/Solution

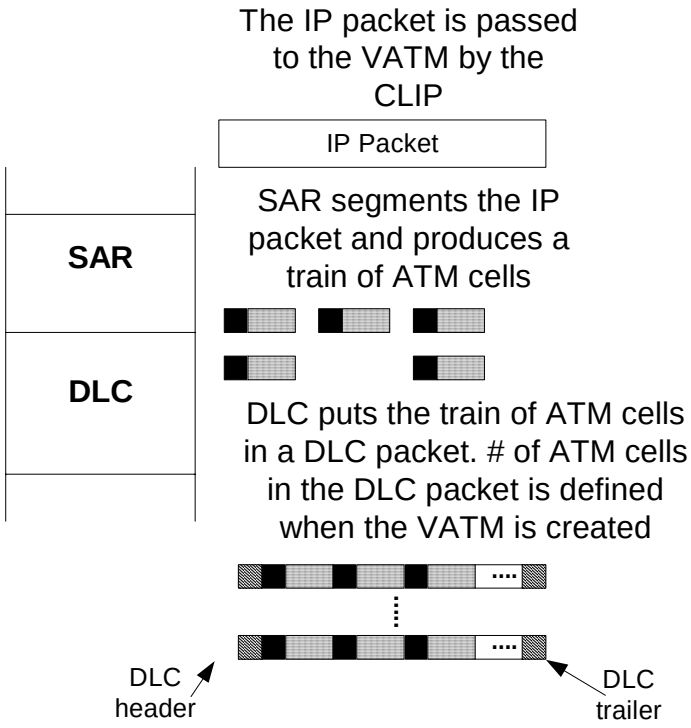
- Ability to establish/tear-down high-speed links
  - Nodes are connected to a FORE-ATM switch
  - To establish connectivity between neighbors the PVCs need to be established on the FORE-ATM switch
  - Orderwire Module on the nodes send a request to create/delete the PVCs to the EM
  - EM sends a corresponding SNMP request to the FORE-ATM switch
- Emulation of the beams & the ability to multiplex/ de-multiplex
  - Possible solution could have been to use 4 ATM cards; where each card would represent one beam. Neither feasible nor elegant
  - Implement something called Virtual ATM (VATM)

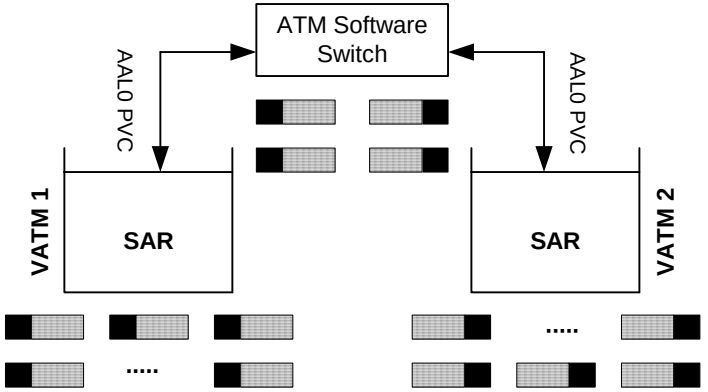
Virtual ATM (VATM)



- VATM is a driver that provides multiple logical ATM interfaces
- Hooked to the ATM card on a physical VCI (AAL5); the traffic to various destinations are sent over the logical VCIs
- Each VATM represent a beam with a configurable protocol stack
- Possible to build a VATM on a VATM

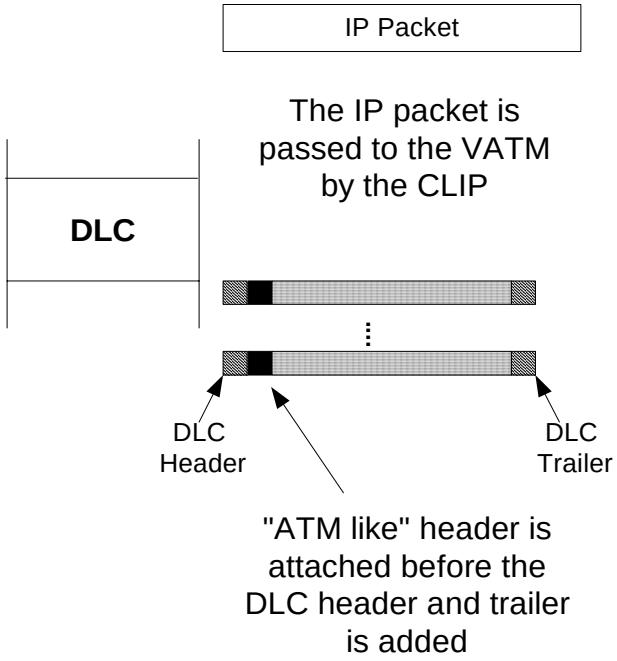
Protocol Stack on VATM SAR+DLC





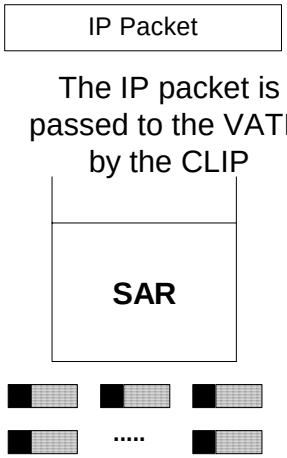
- SAR segments the packet into a train of ATM cells
- DLC packets the cells into a DLC packet and sends the packet to the ATM driver
- A VATM with the SAR layer can be hooked to the MicroSwitch. No re-assembly in this case.

Protocol Stack on VATM DLC



- Packets from the higher layer are first passed to the AAL_DLC_GLUE_LAYER
- The “glue_layer” attaches a 5 byte ATM-like header and passes the packet to the DLC layer
- The DLC puts its own header & trailer and passes the packet down to the ATM driver
- IP over ATM specification says that the MTU cannot be larger than 9180 bytes; hence the CLIP can pass a packet of the above size to the “glue_layer”. Hence, when the DLC layer would attach its own header and trailer, it would cause an overflow on the ENI card.
- In the above case the segmentation of the packet passed by CLIP needs to be done. This is the reason why the ATM-like header is added by the “glue_layer”

Protocol Stack on VATM SAR



- Packet passed from the higher layer is segmented into a train of ATM cells by the SAR
- These train of ATM cells are passed to the ATM driver which packets them in an AAL5 frame
- This particular protocol stack is not valid on the RDRN radios

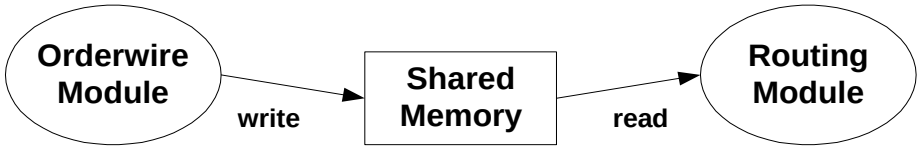
Protocol Stack on VATM SAR+QoS+DLC

- Packets passed from above are passed to the SAR which does the segmentation into ATM cells
- The train of ATM cells is passed to the QoS layer
- The QoS layer maintains different queues for traffics of different priority; and depending on its scheduling algorithm it sends the ATM cells to the DLC layer
- The DLC layer packets the ATM cells and adds its own header and trailer and passes the DLC packet to the ATM driver
- The ATM driver sends the DLC packets as AAL5 frames

Identification of Requirements
Steps in a Field Scenario

- Step 3: Creation/Exchange of Routing Information
  - ! Implement the Routing Protocol, Wireless Multi-path Routing Protocol (WMRP)

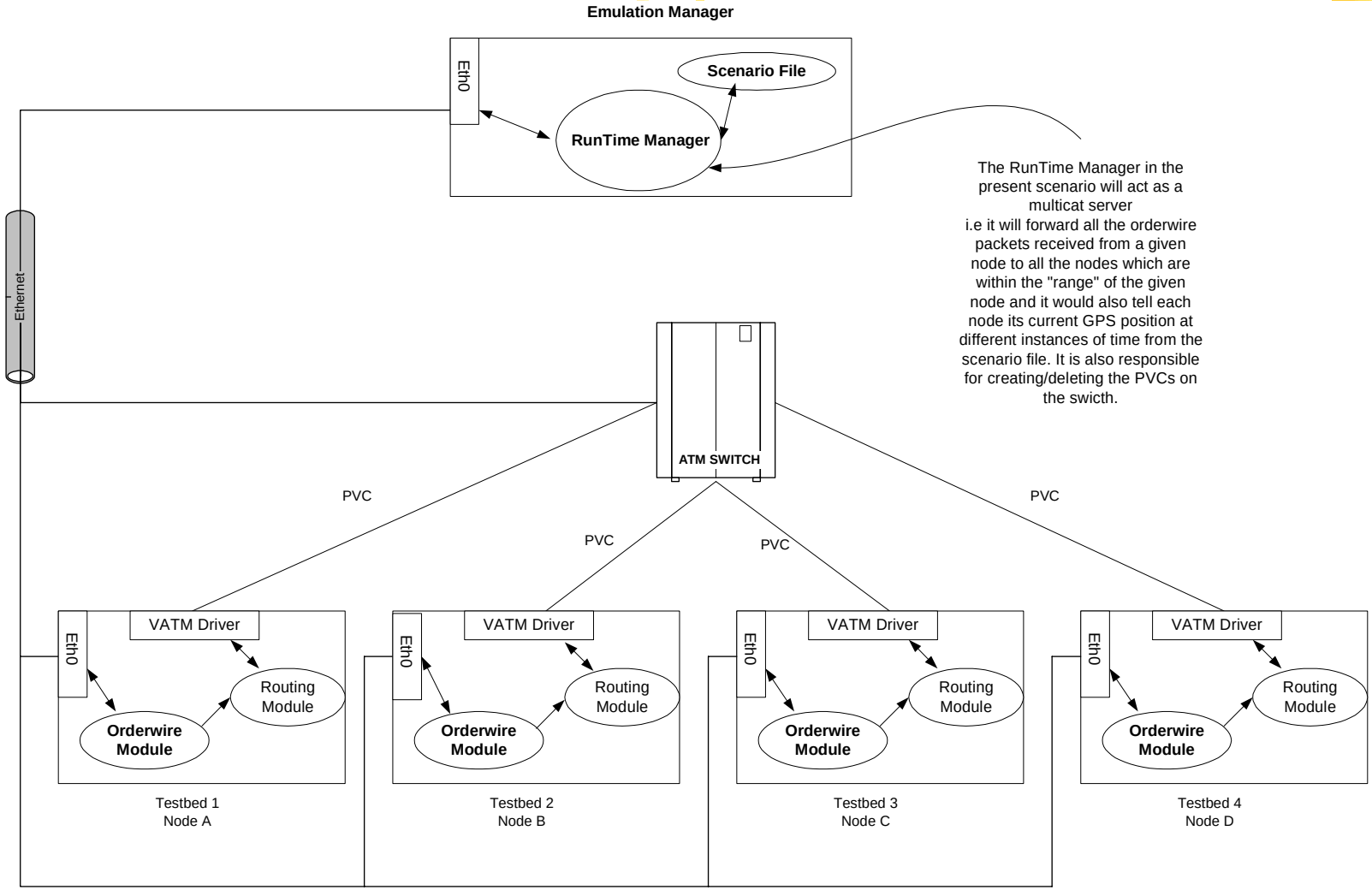
- Implementation of The WMRP
  - Orderwire Informs the Routing Module about the nodes to which it has established high-speed connectivity and on what beam



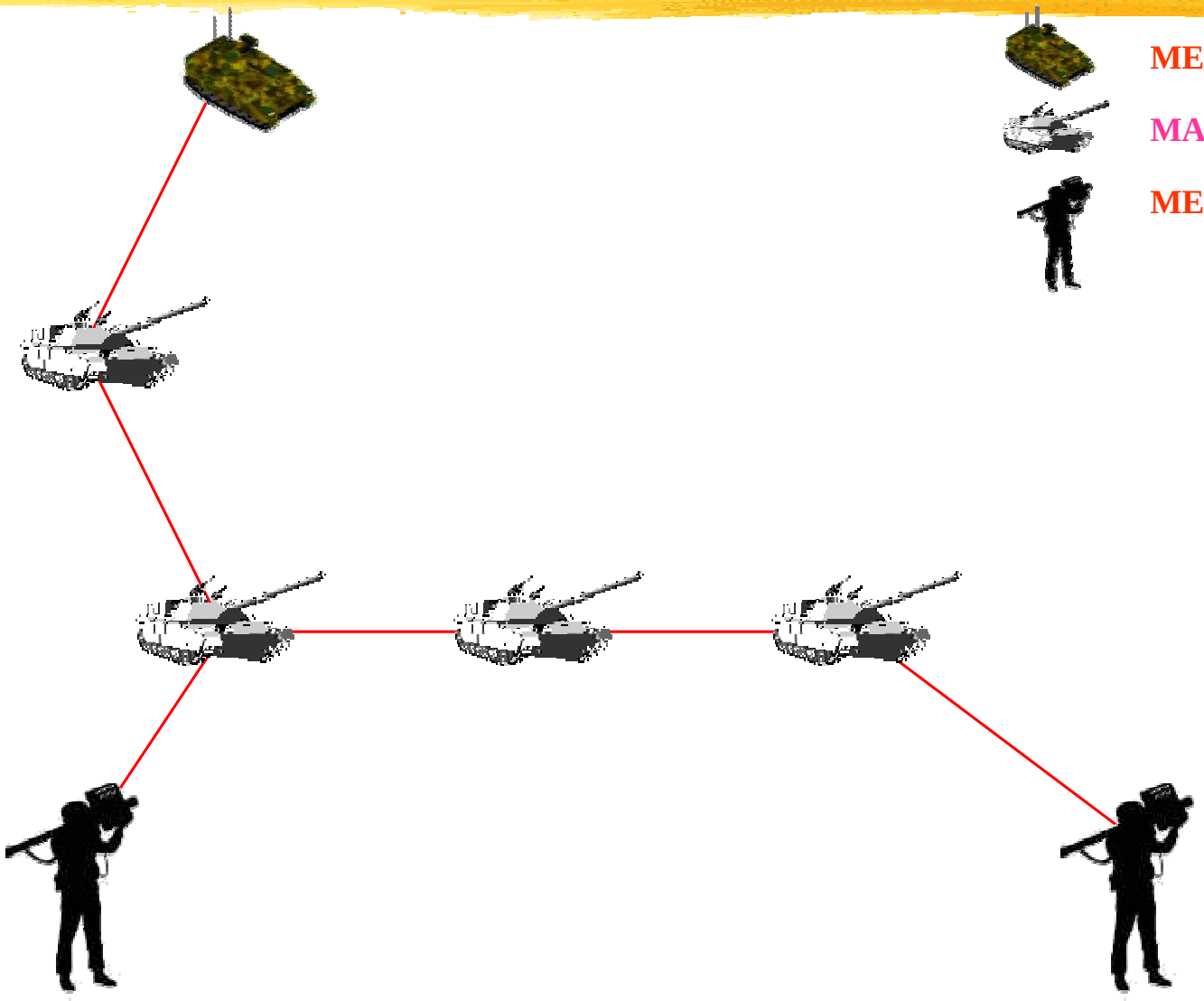
Implementation of Routing Protocol, contd.

- Routing Protocol exchange the Hello Packets and the Routing Updates over a TCP socket on the high-speed link
- Implemented as multi-threaded (Pthreads) application residing in the user-space
- Use Netlink sockets to change the Kernel Routing Table

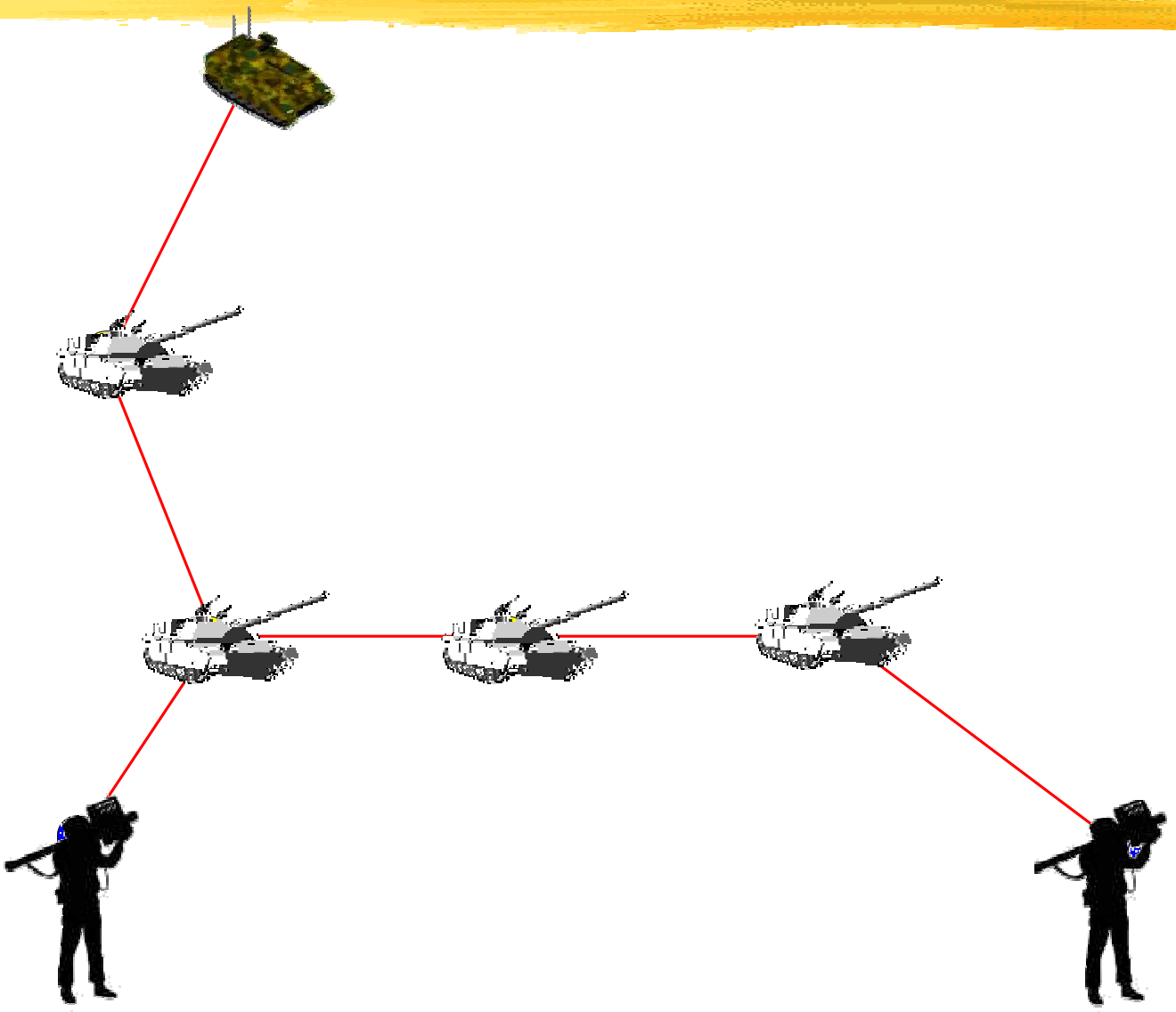
Software Modules



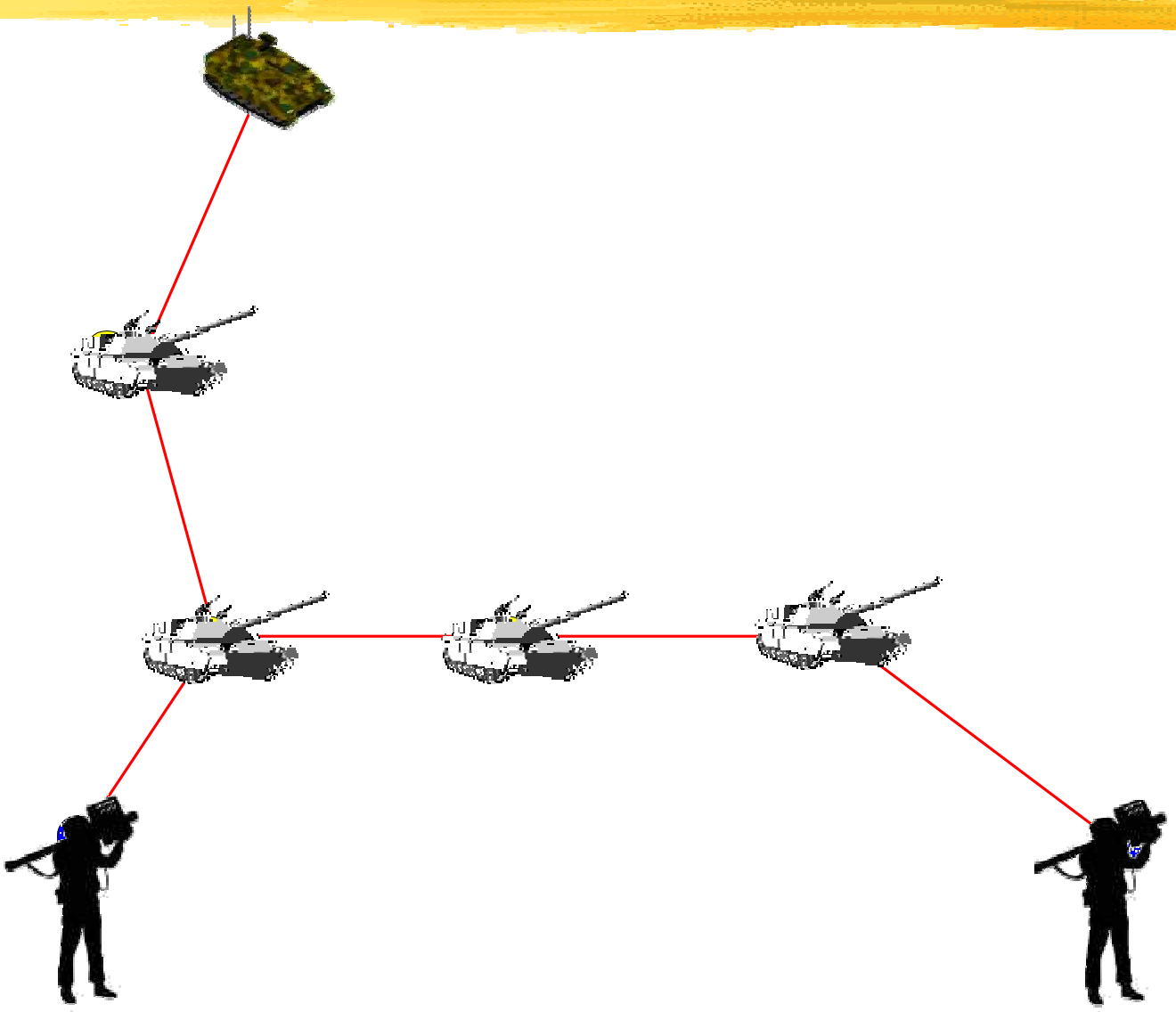
Scenario 1



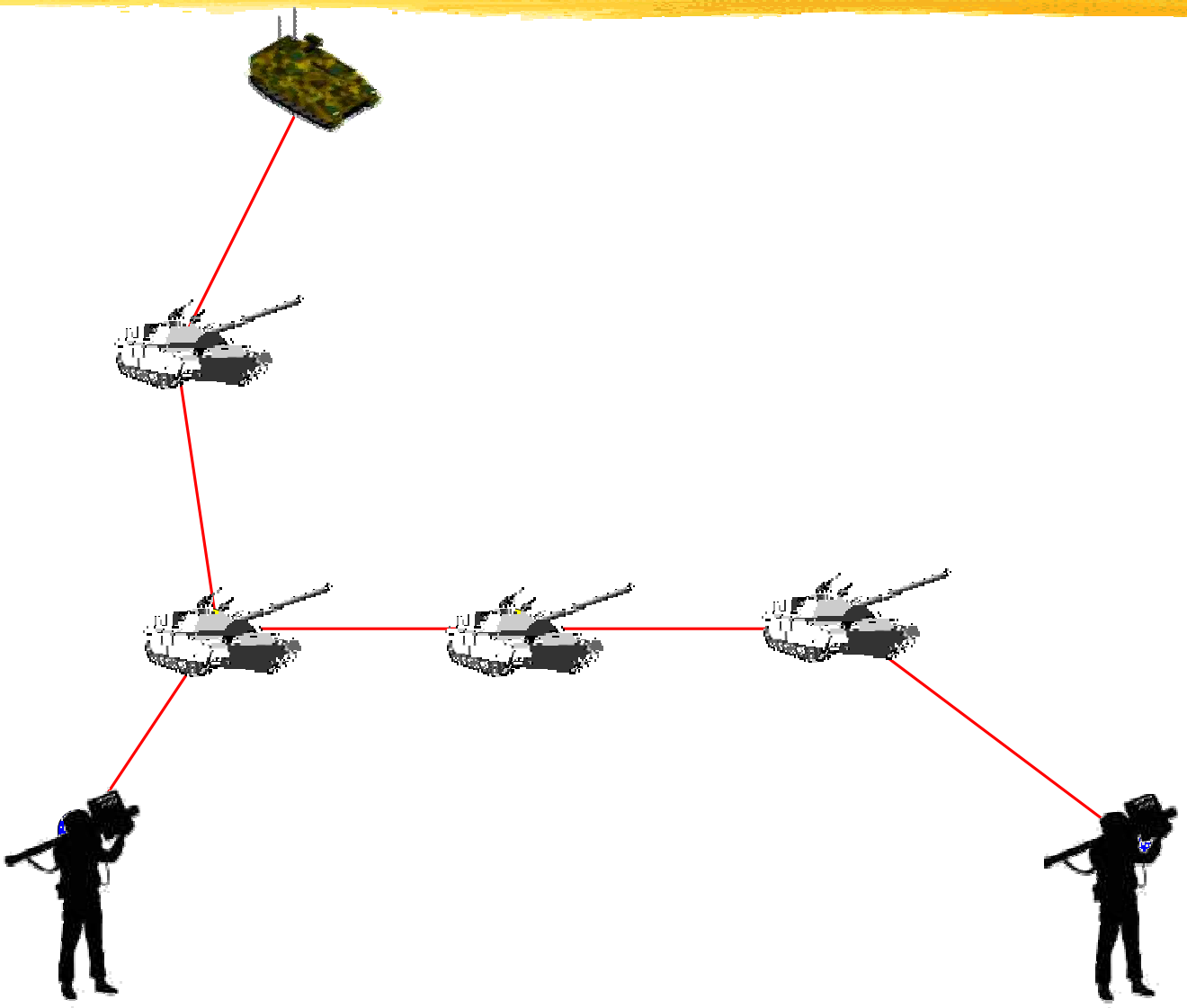
Scenario 1



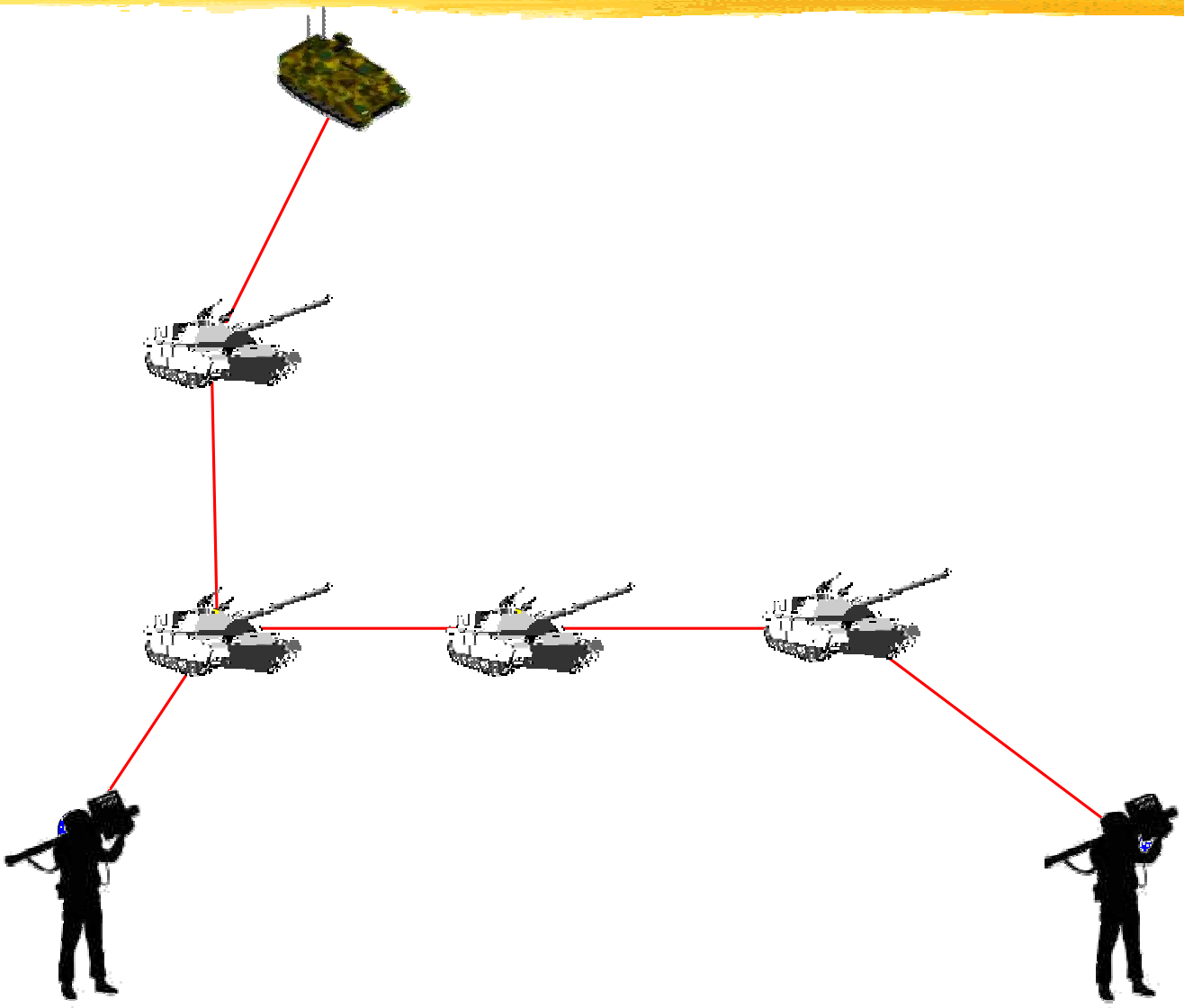
Scenario 1



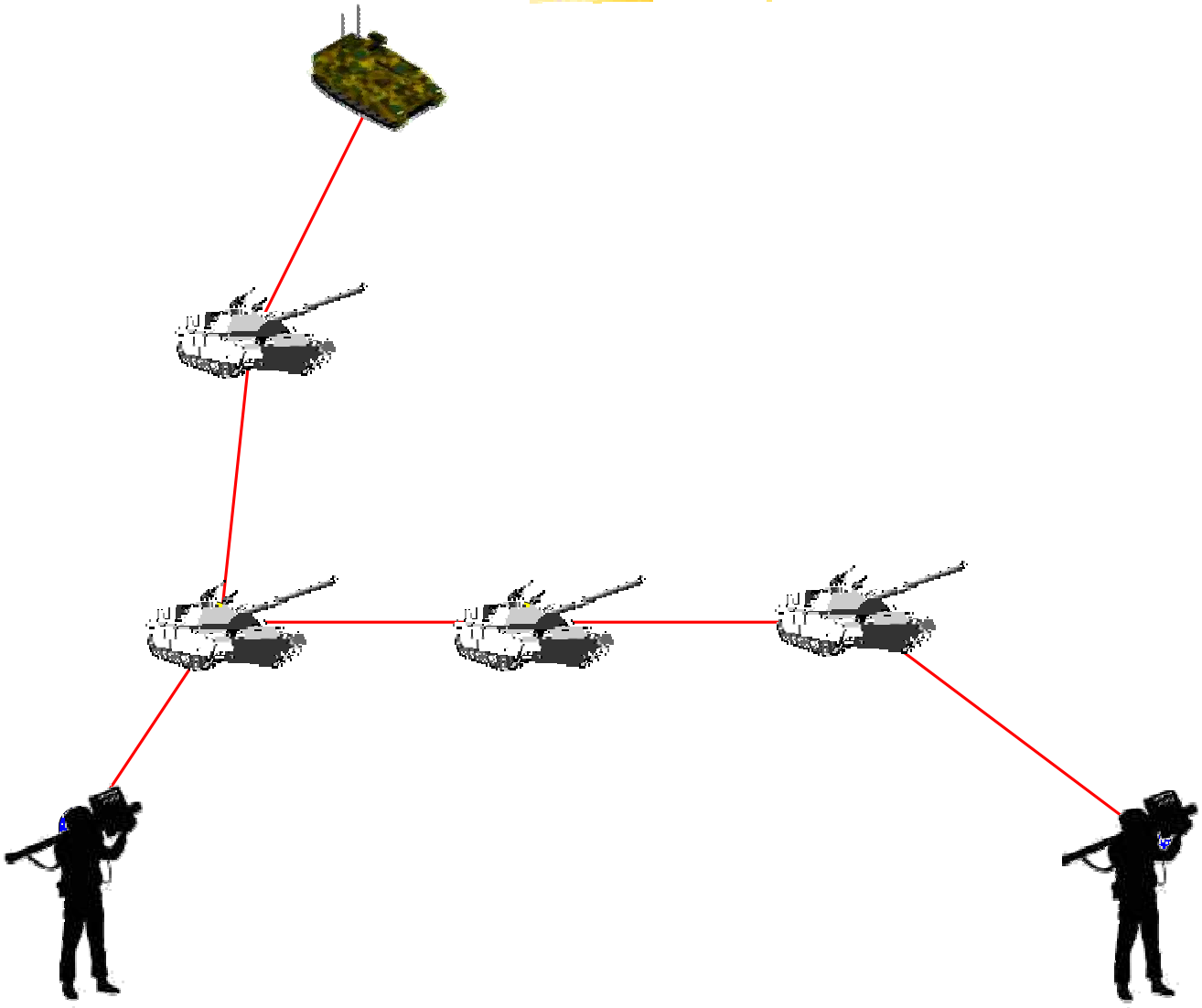
Scenario 1



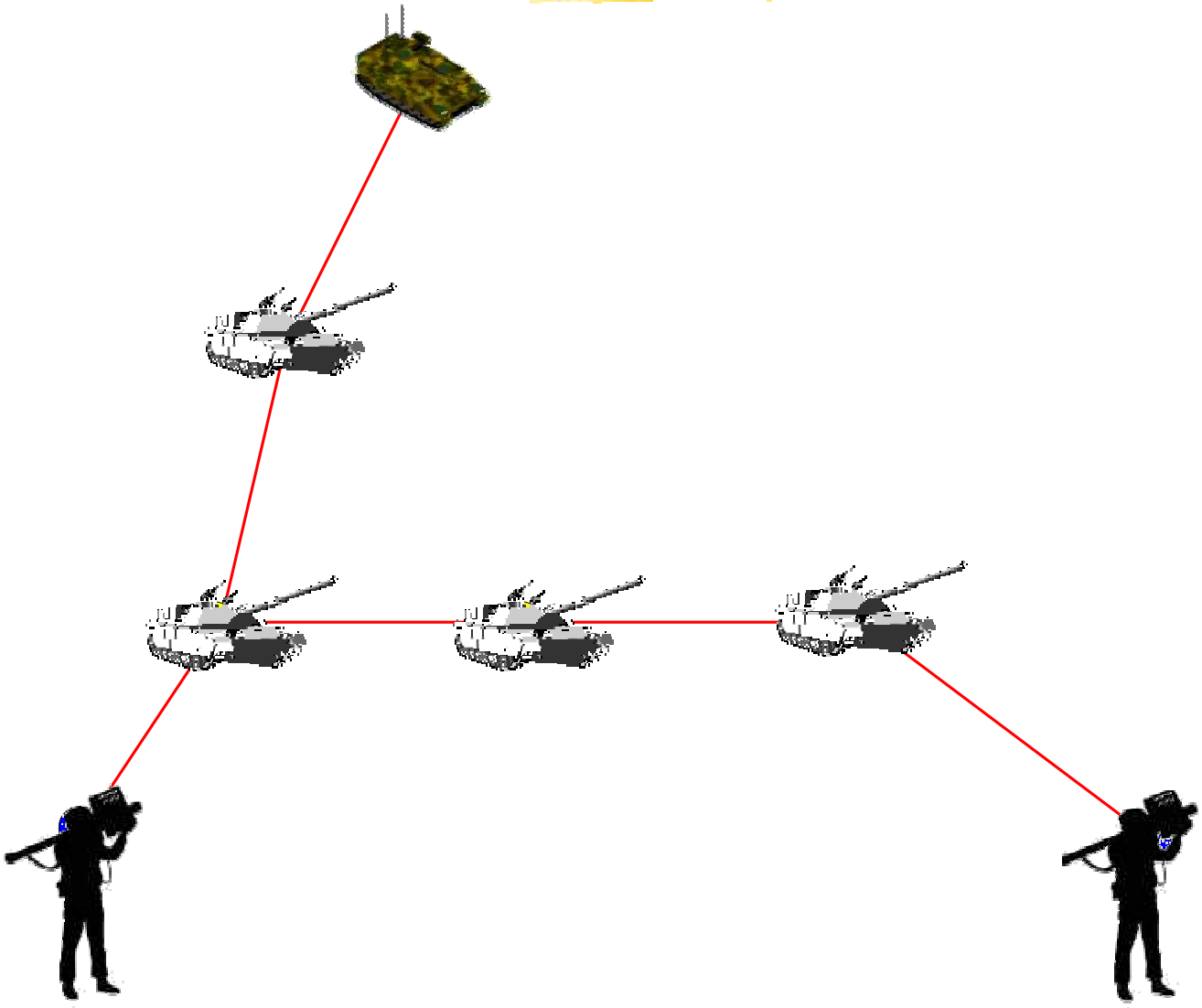
Scenario 1



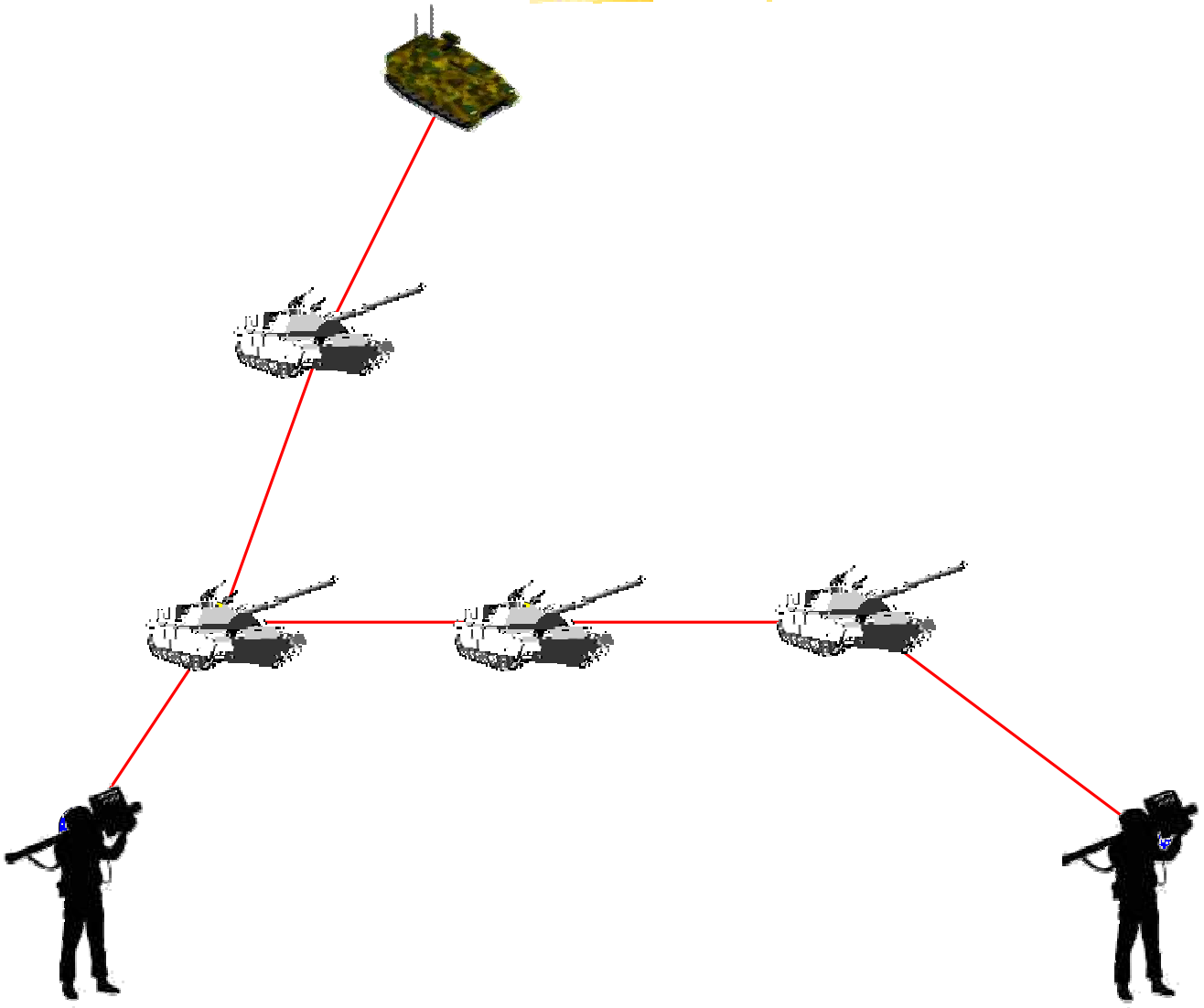
Scenario 1



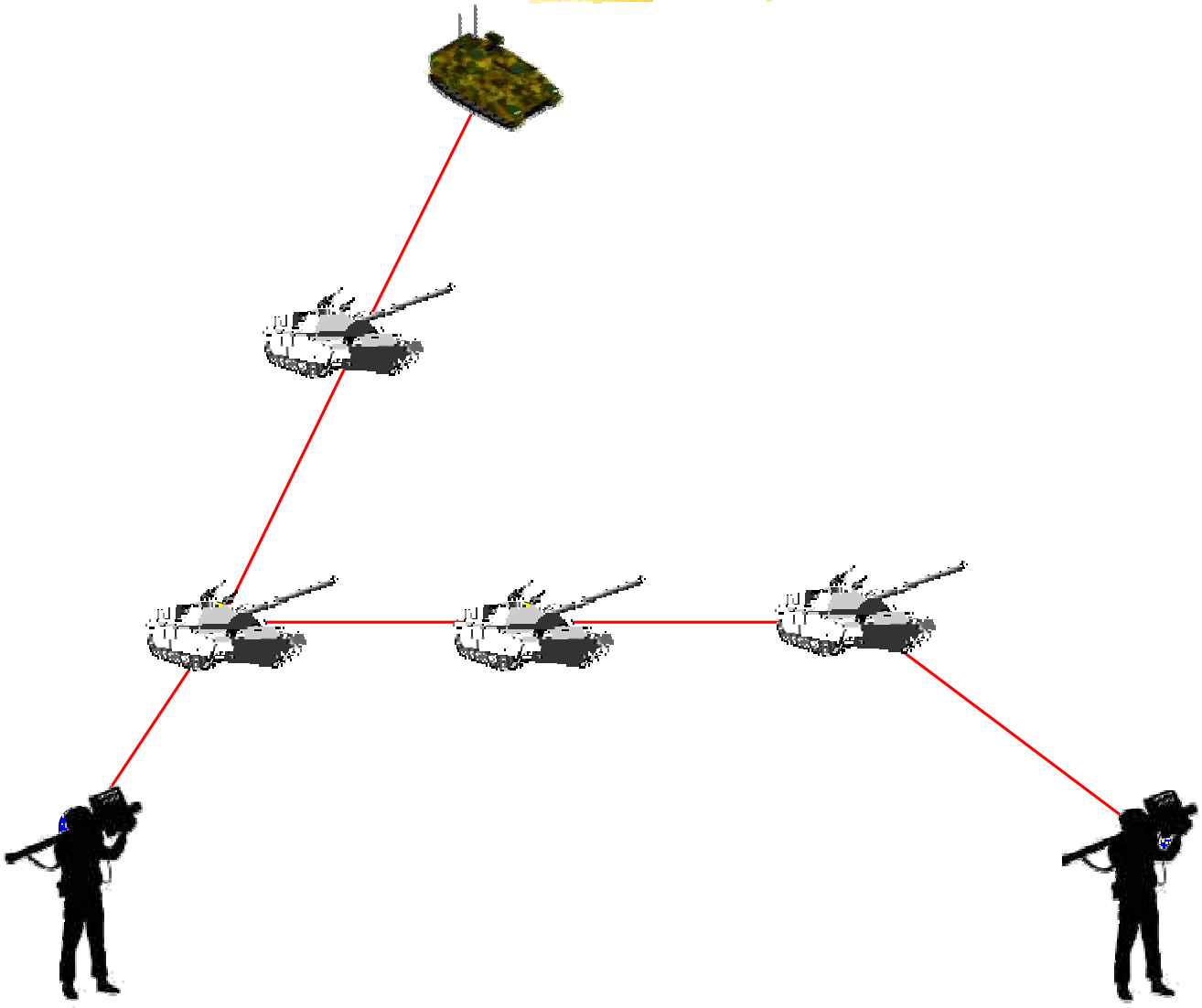
Scenario 1



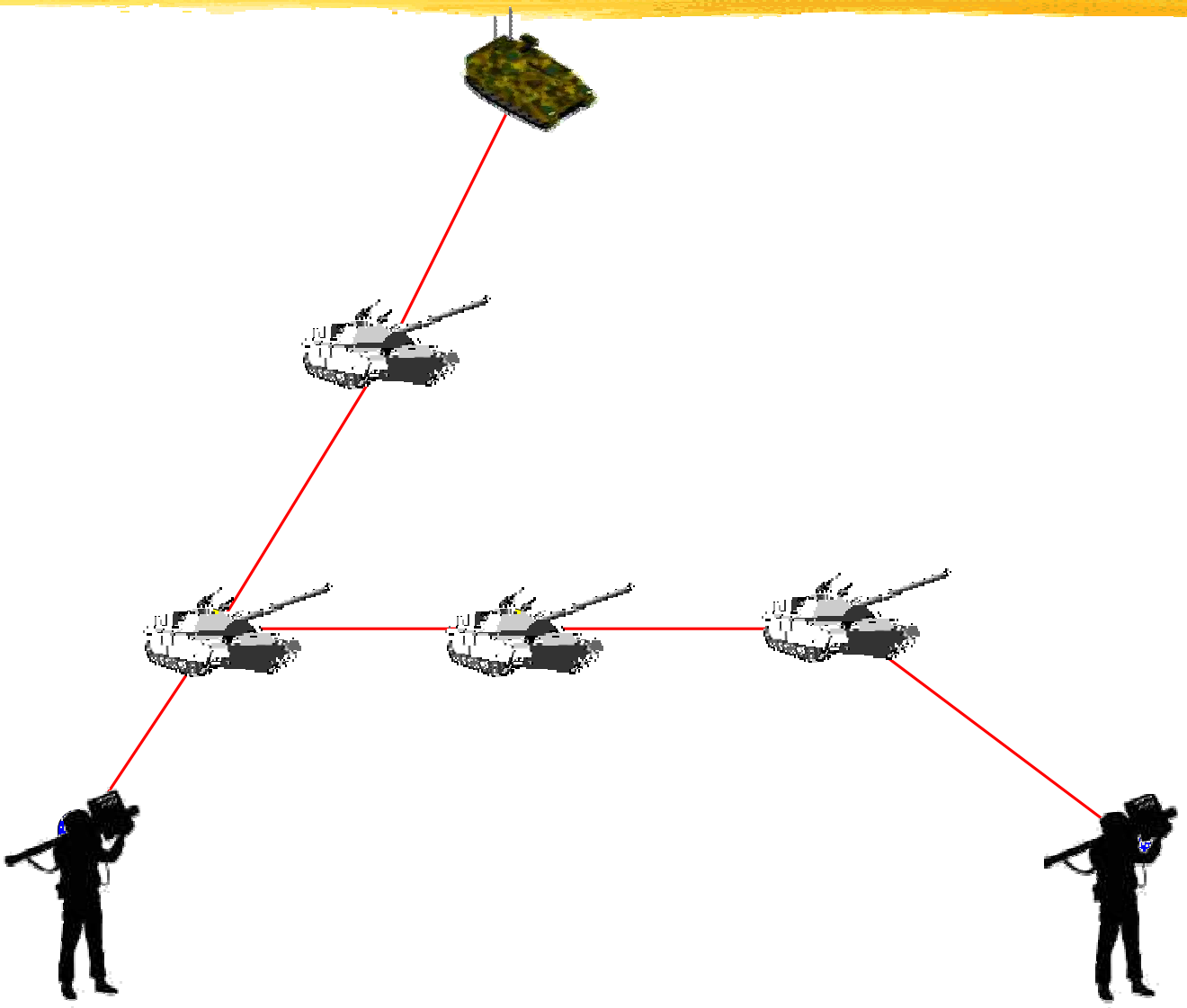
Scenario 1



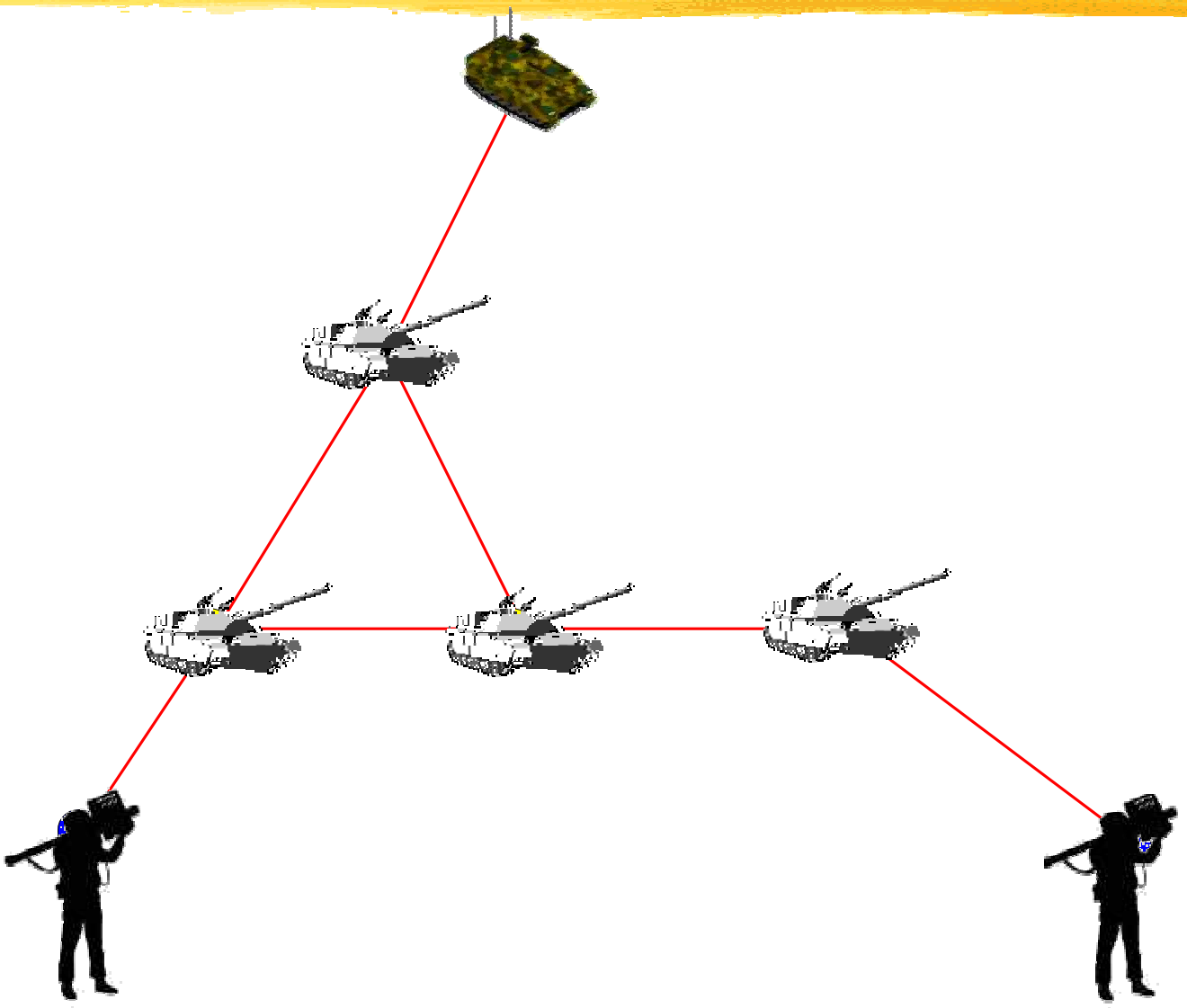
Scenario 1



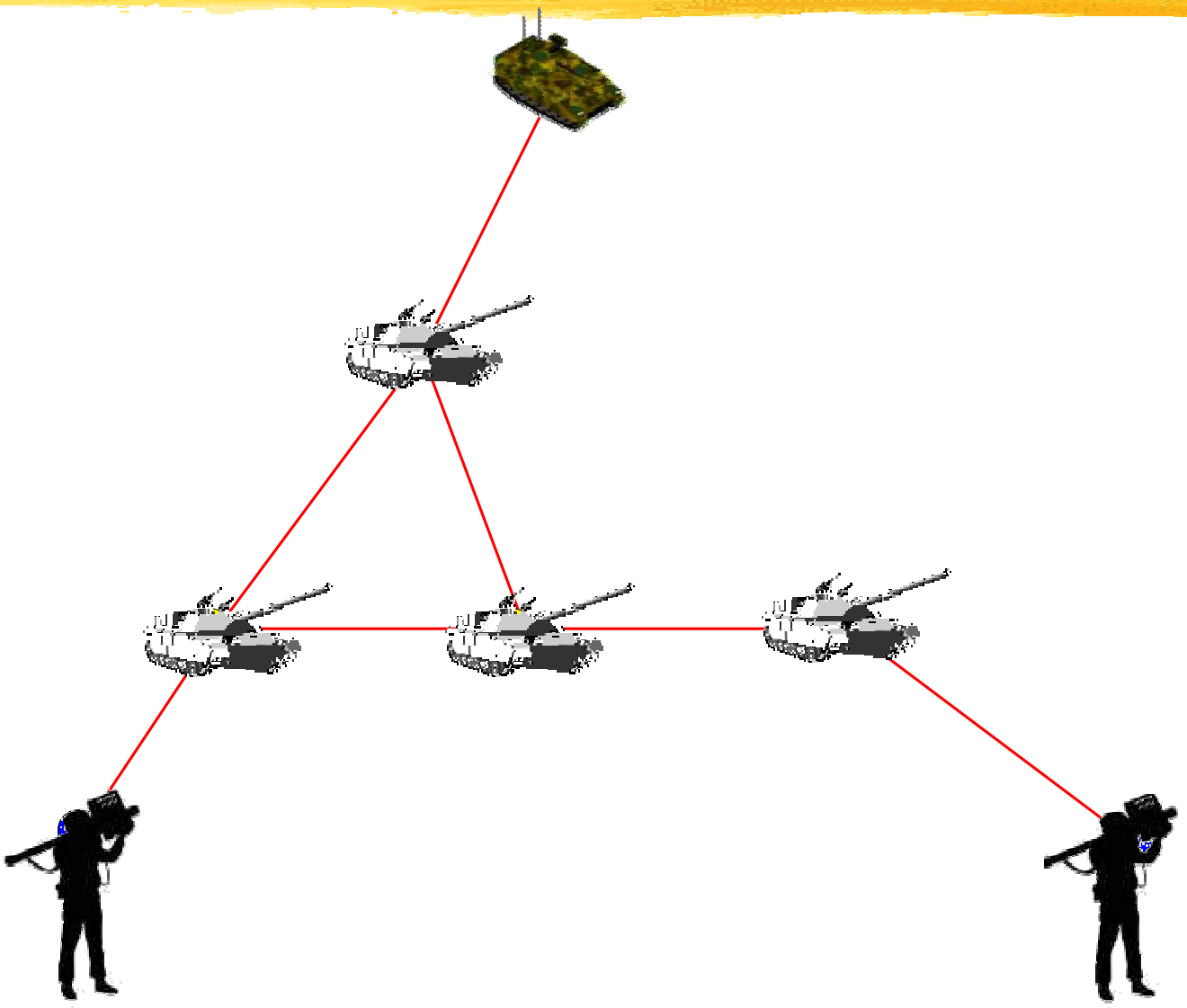
Scenario 1



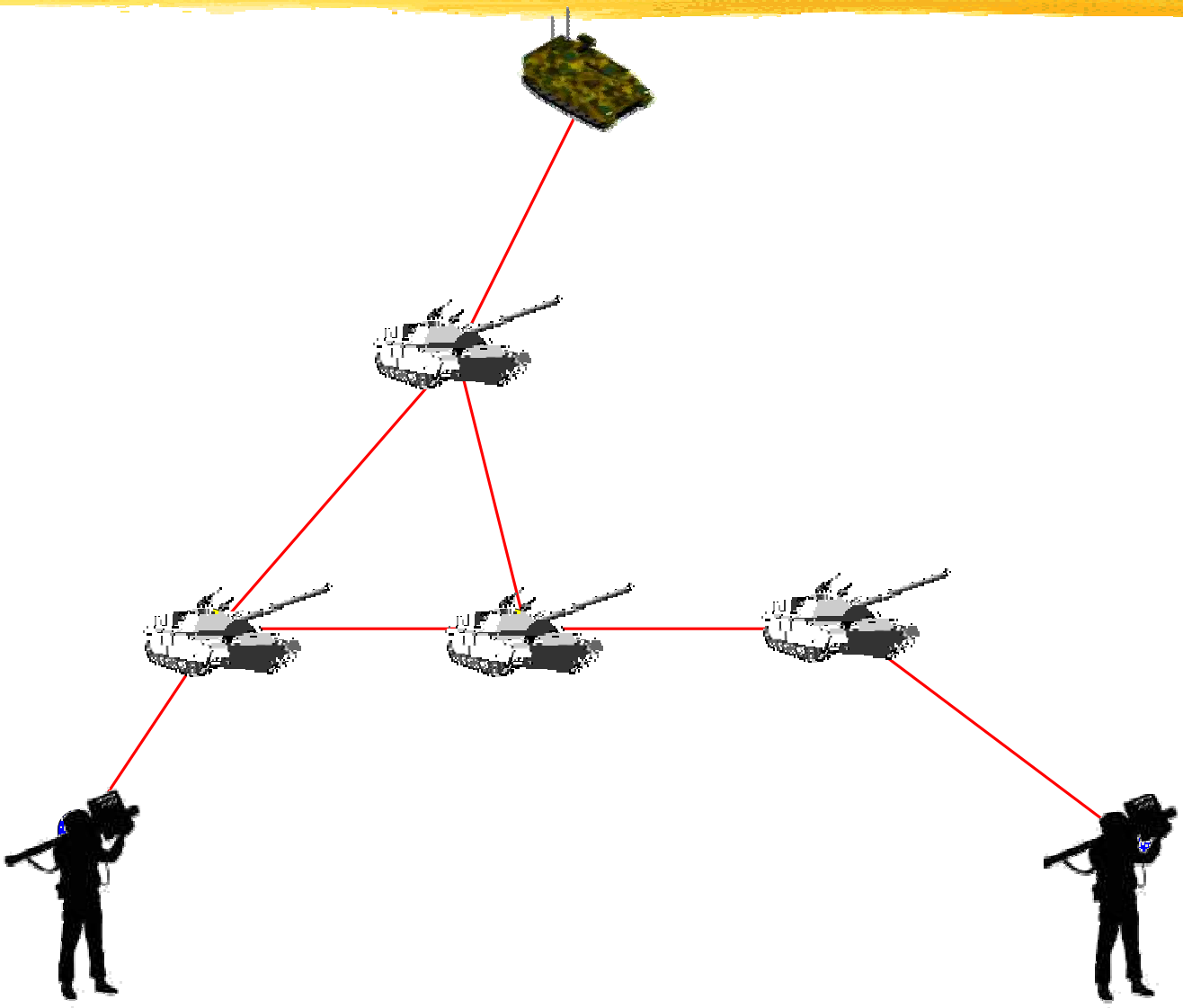
Scenario 1



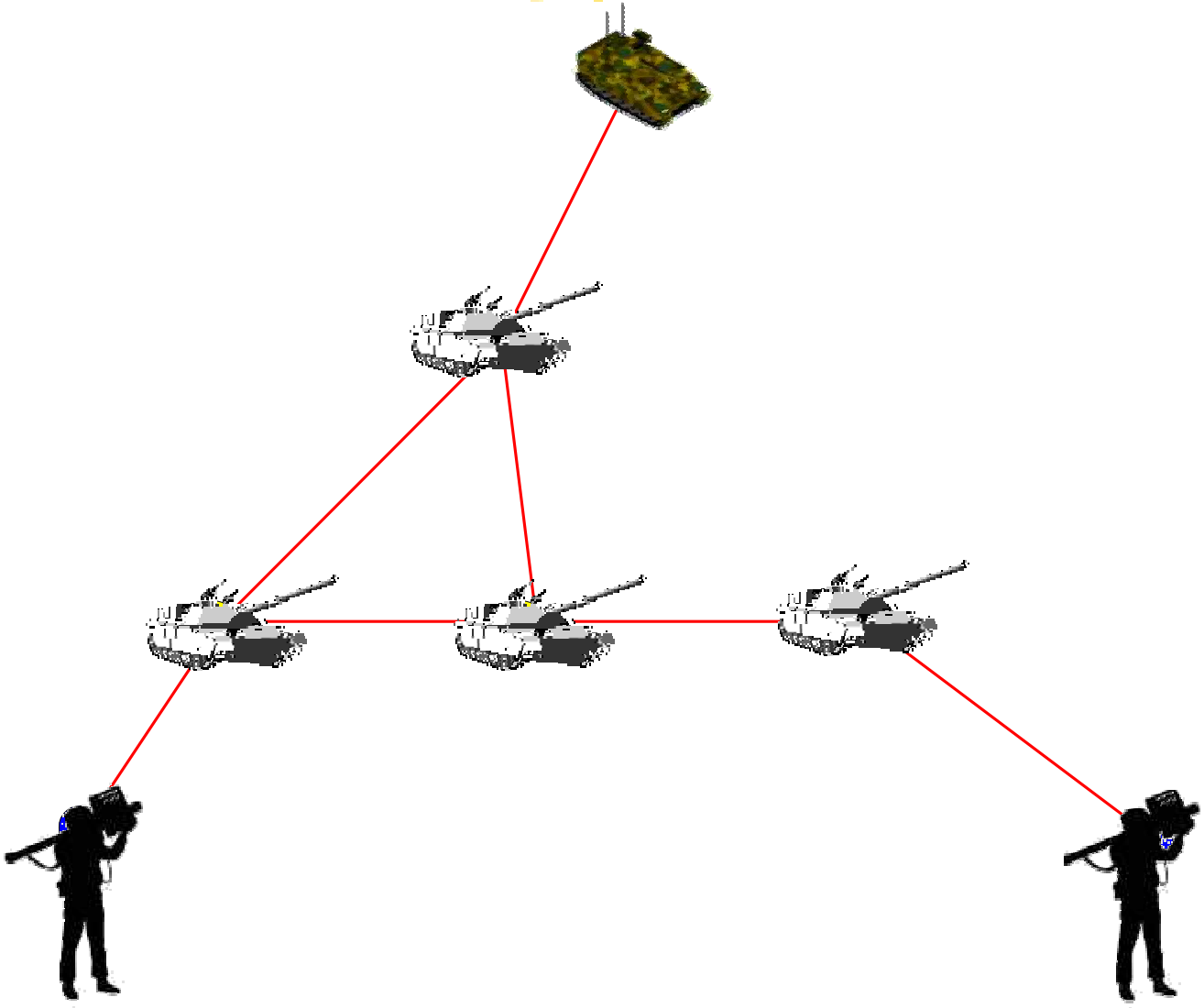
Scenario 1



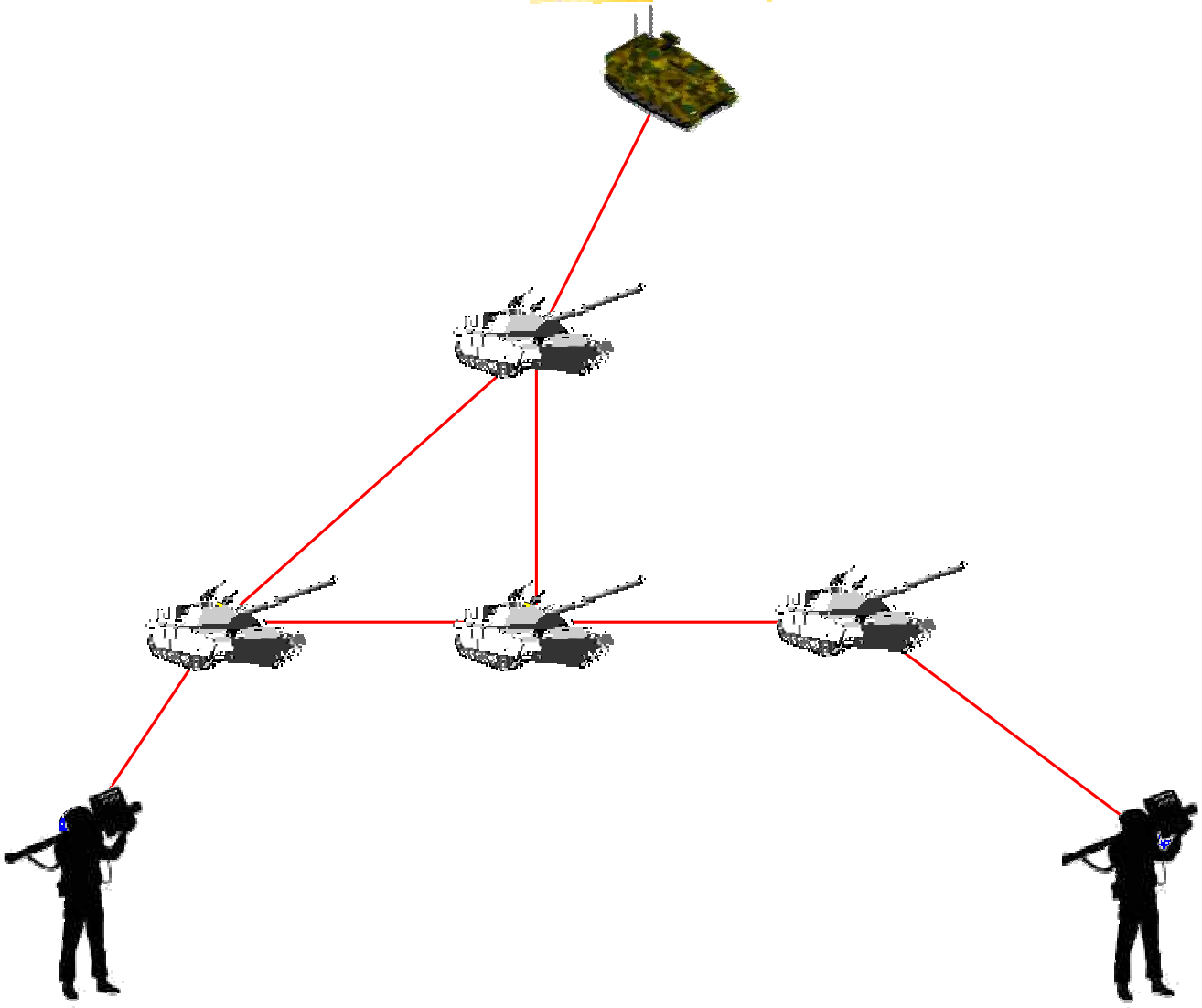
Scenario 1



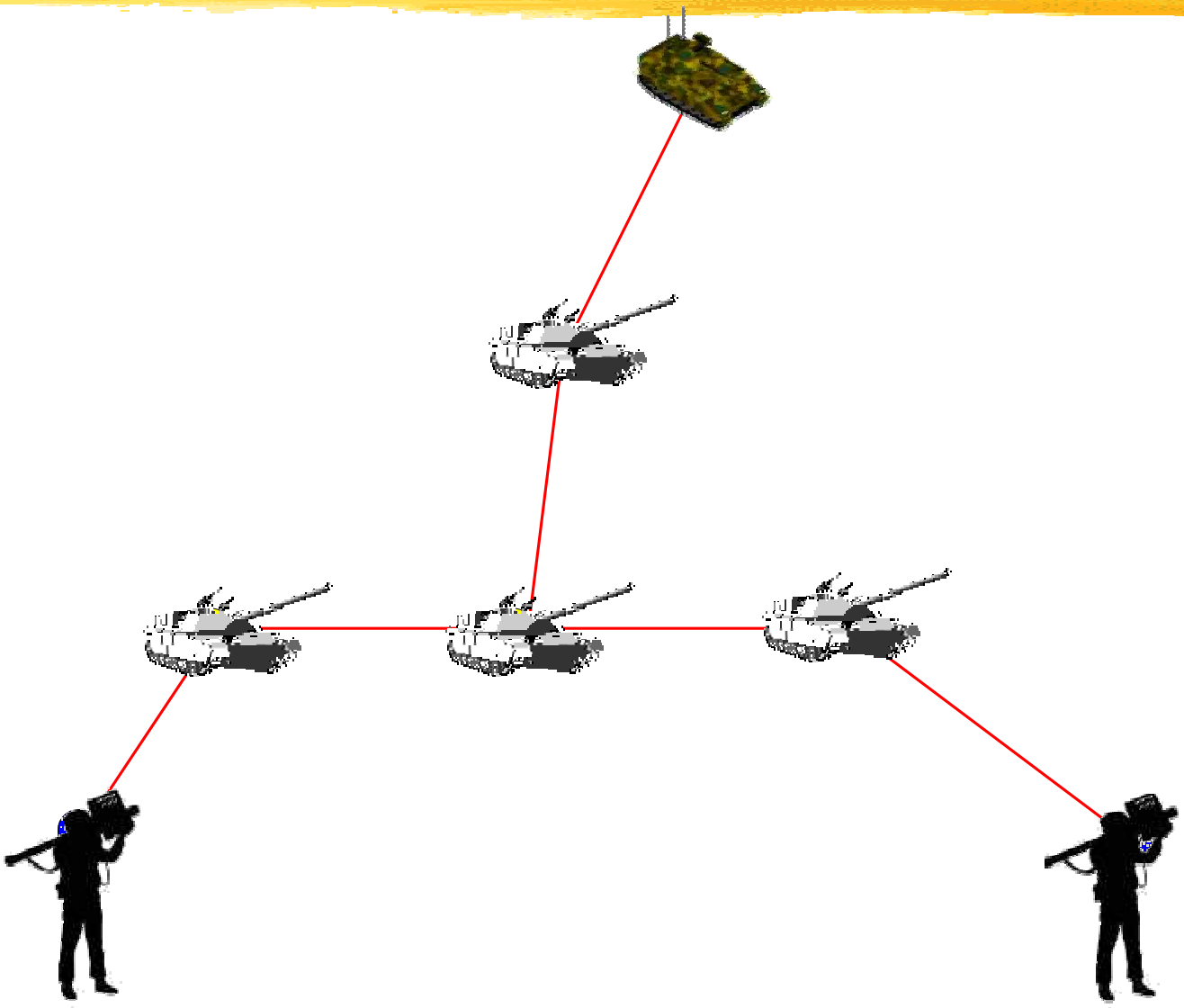
Scenario 1



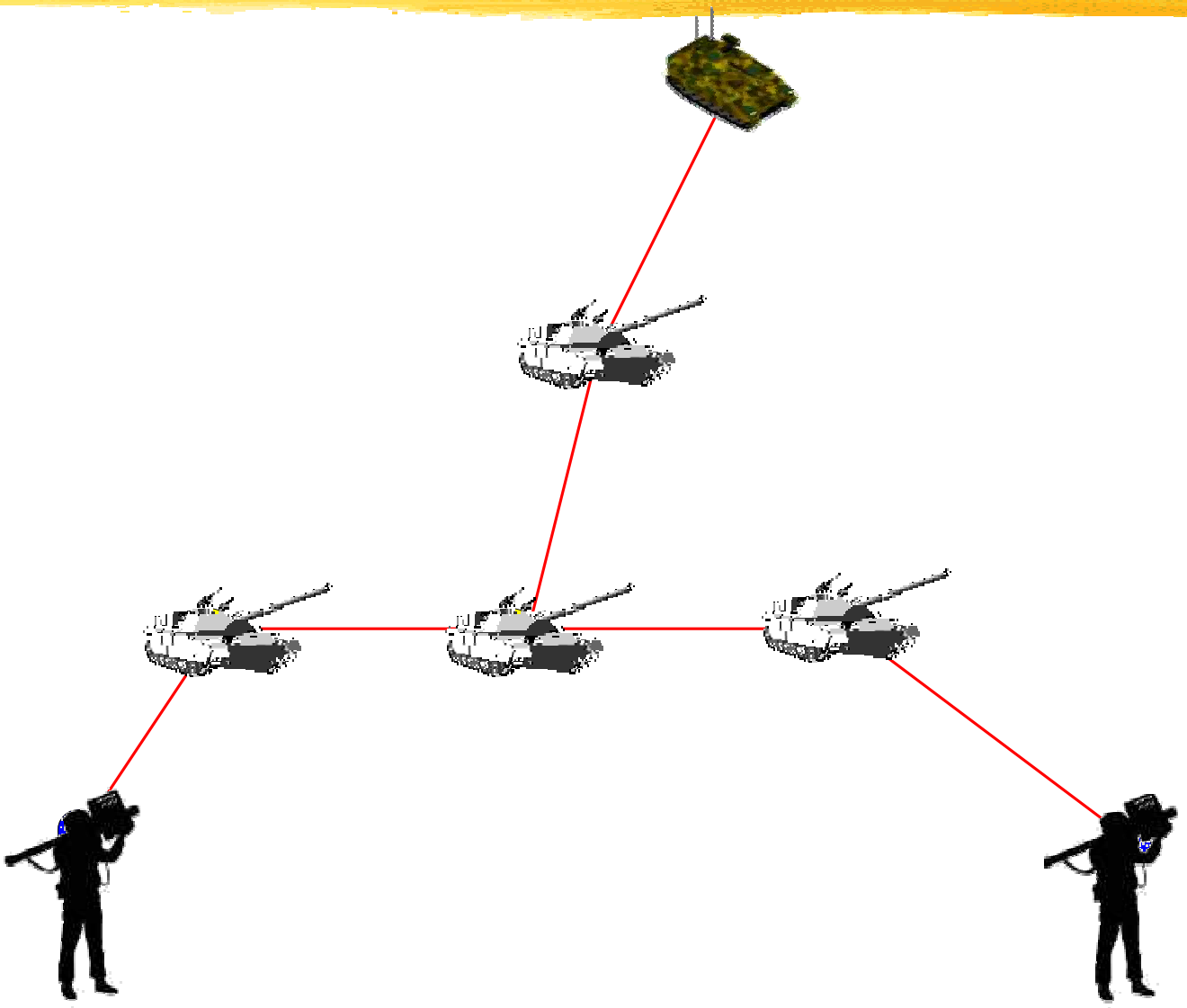
Scenario 1



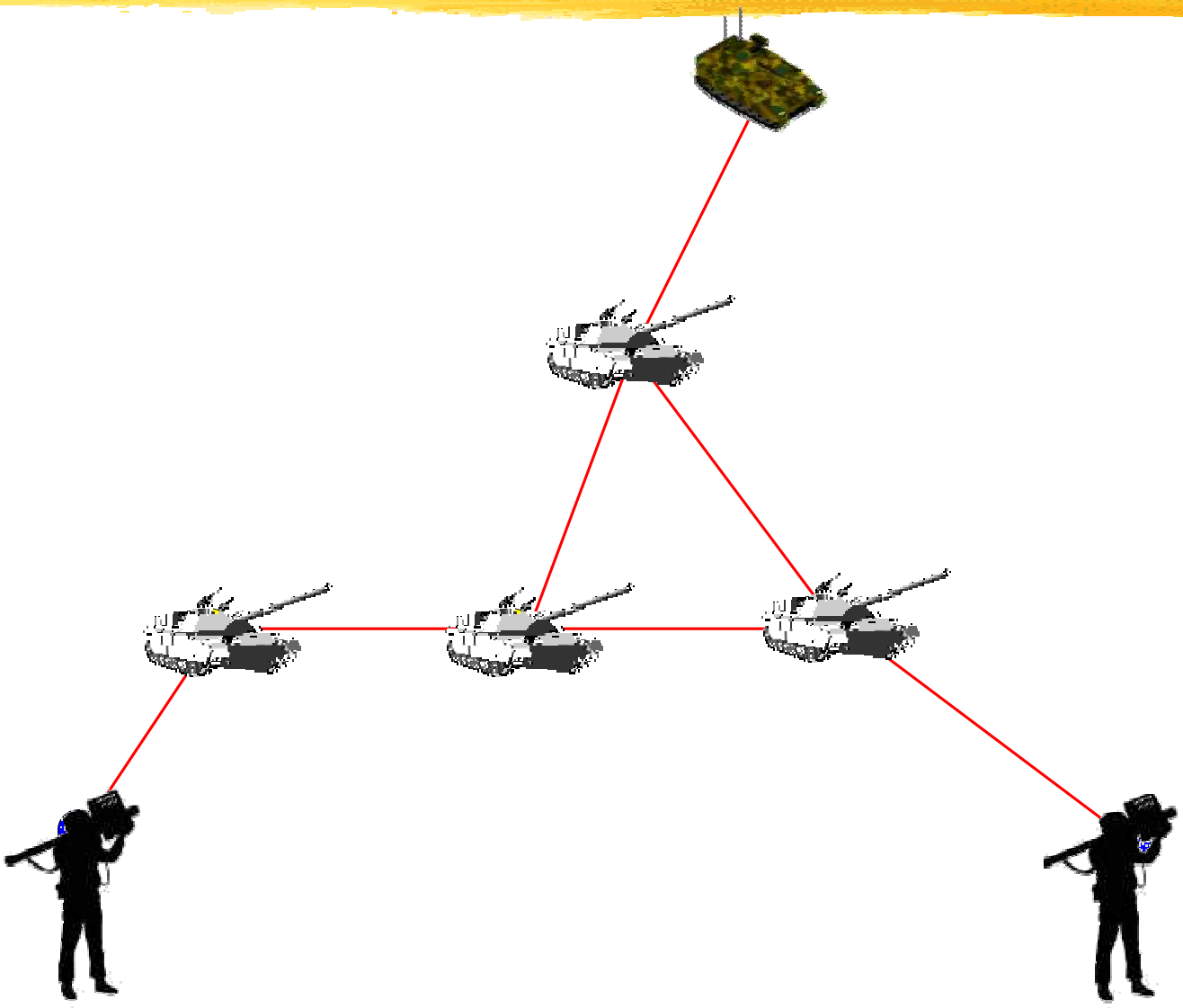
Scenario 1



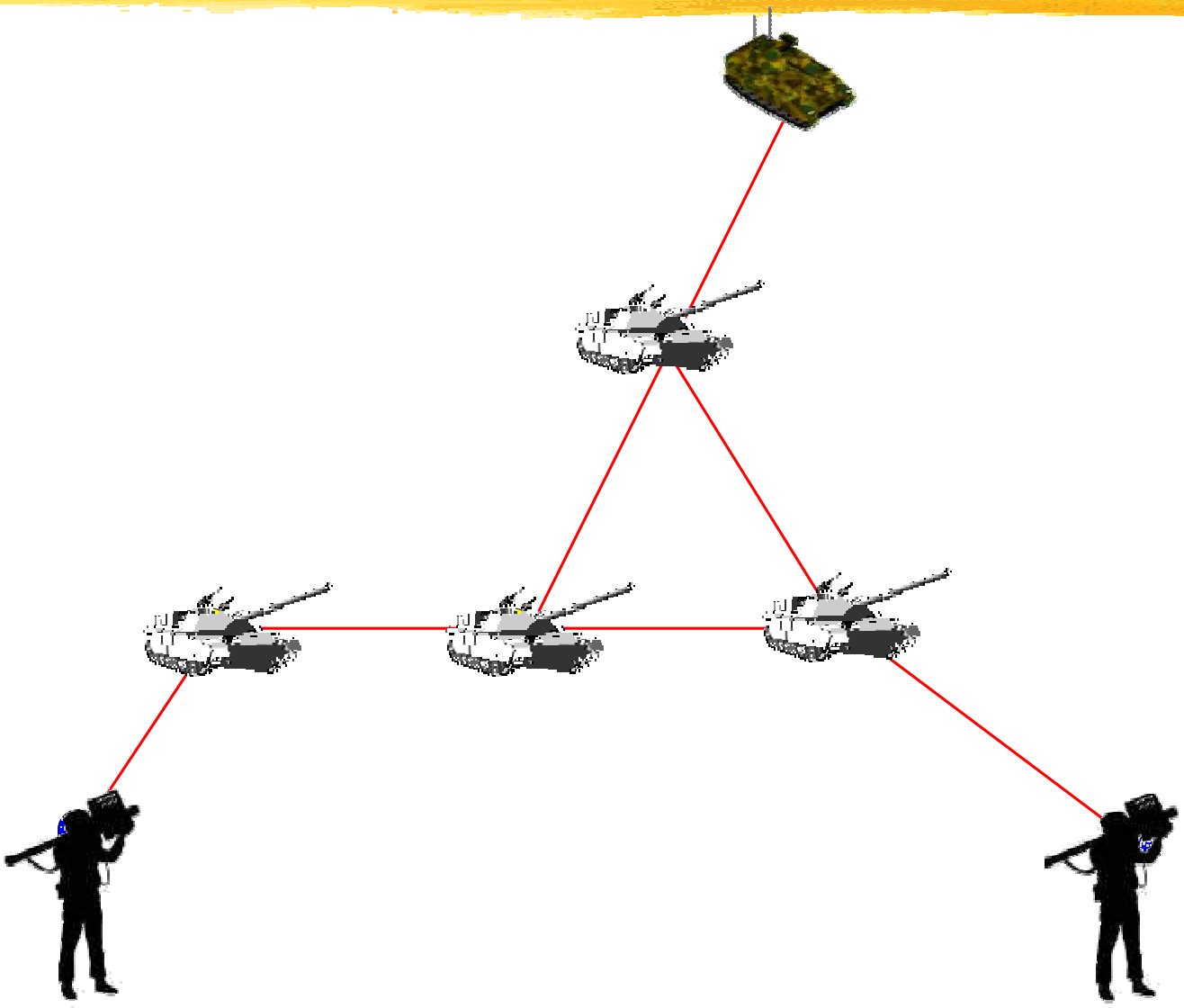
Scenario 1



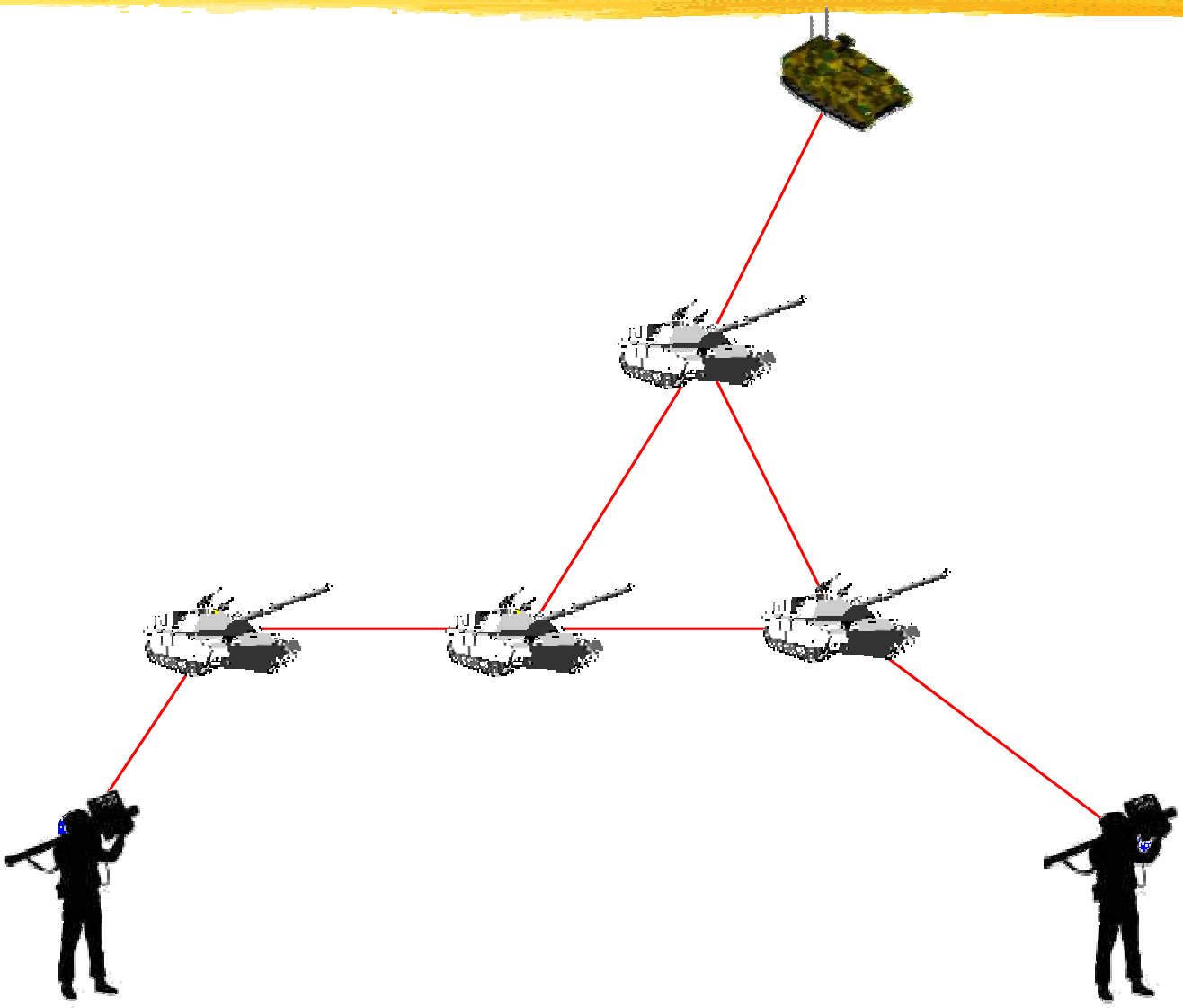
Scenario 1



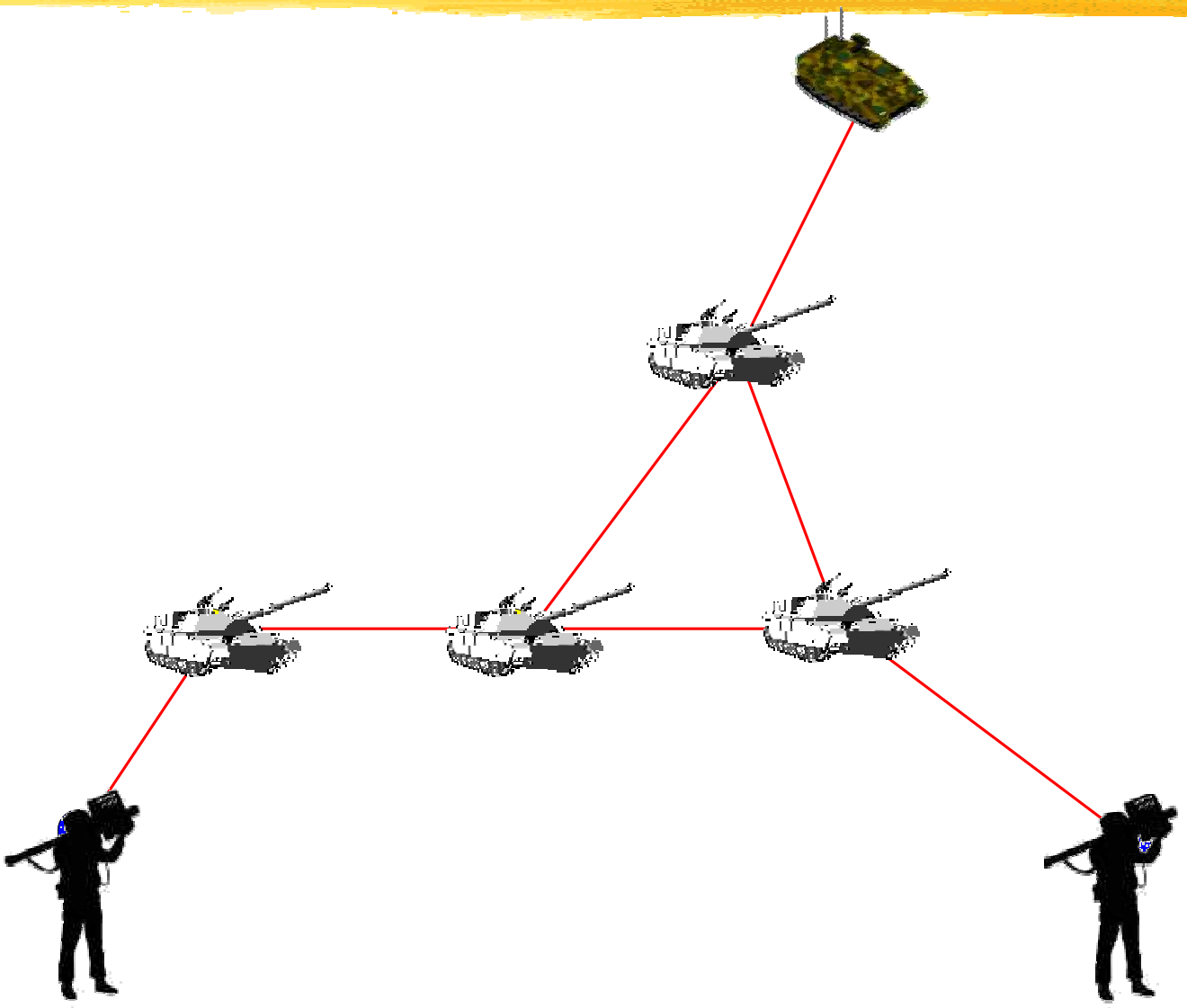
Scenario 1



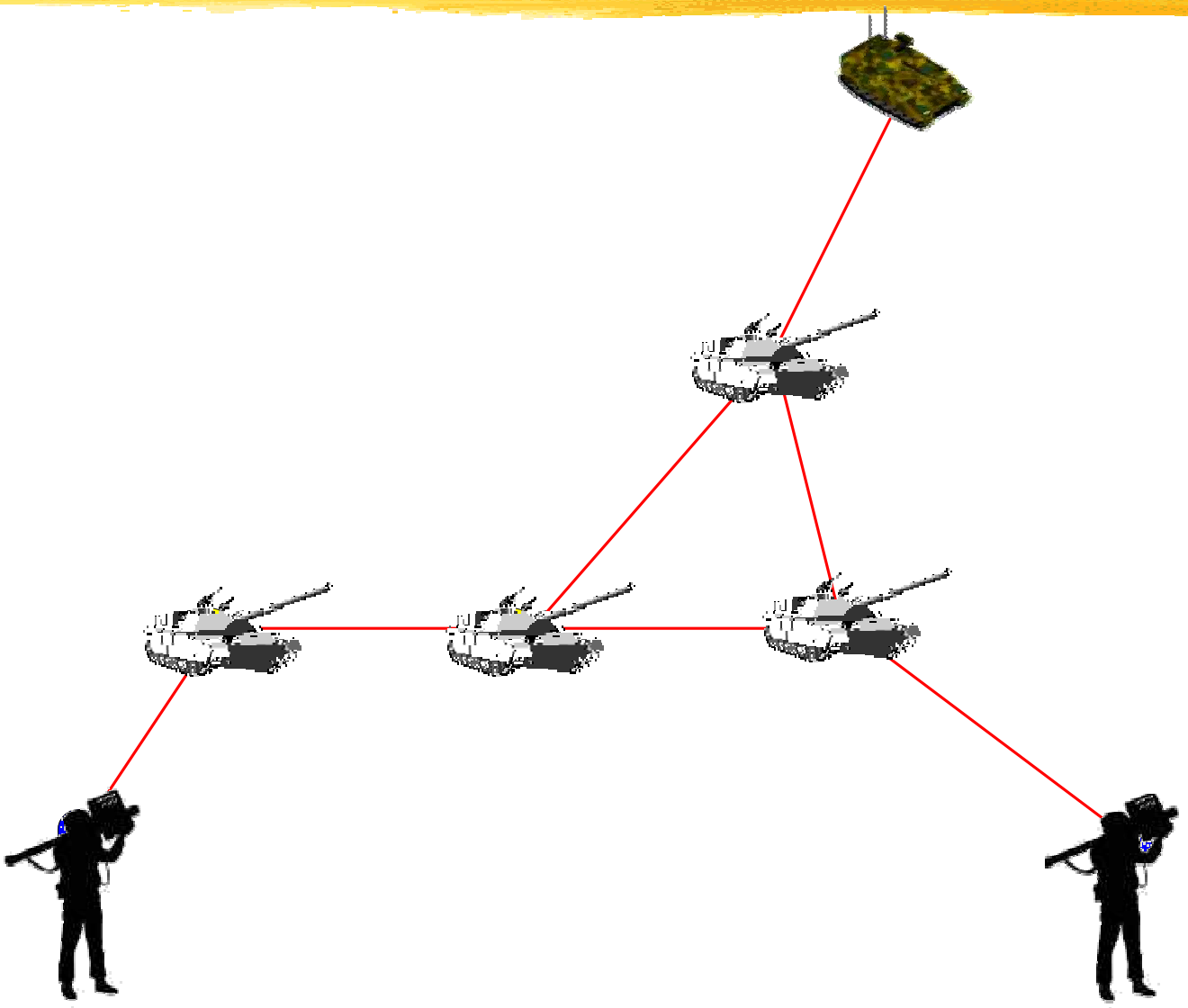
Scenario 1



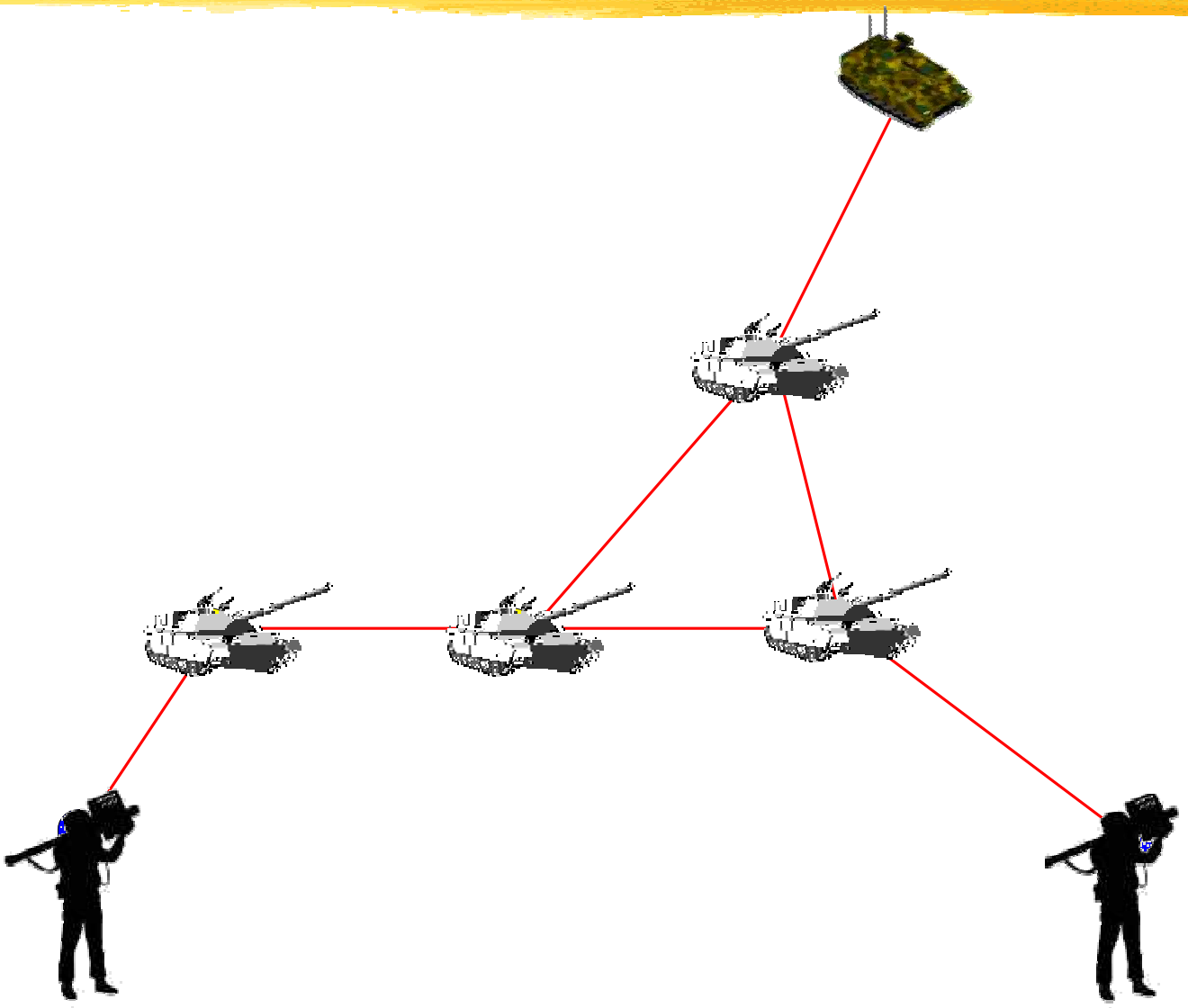
Scenario 1



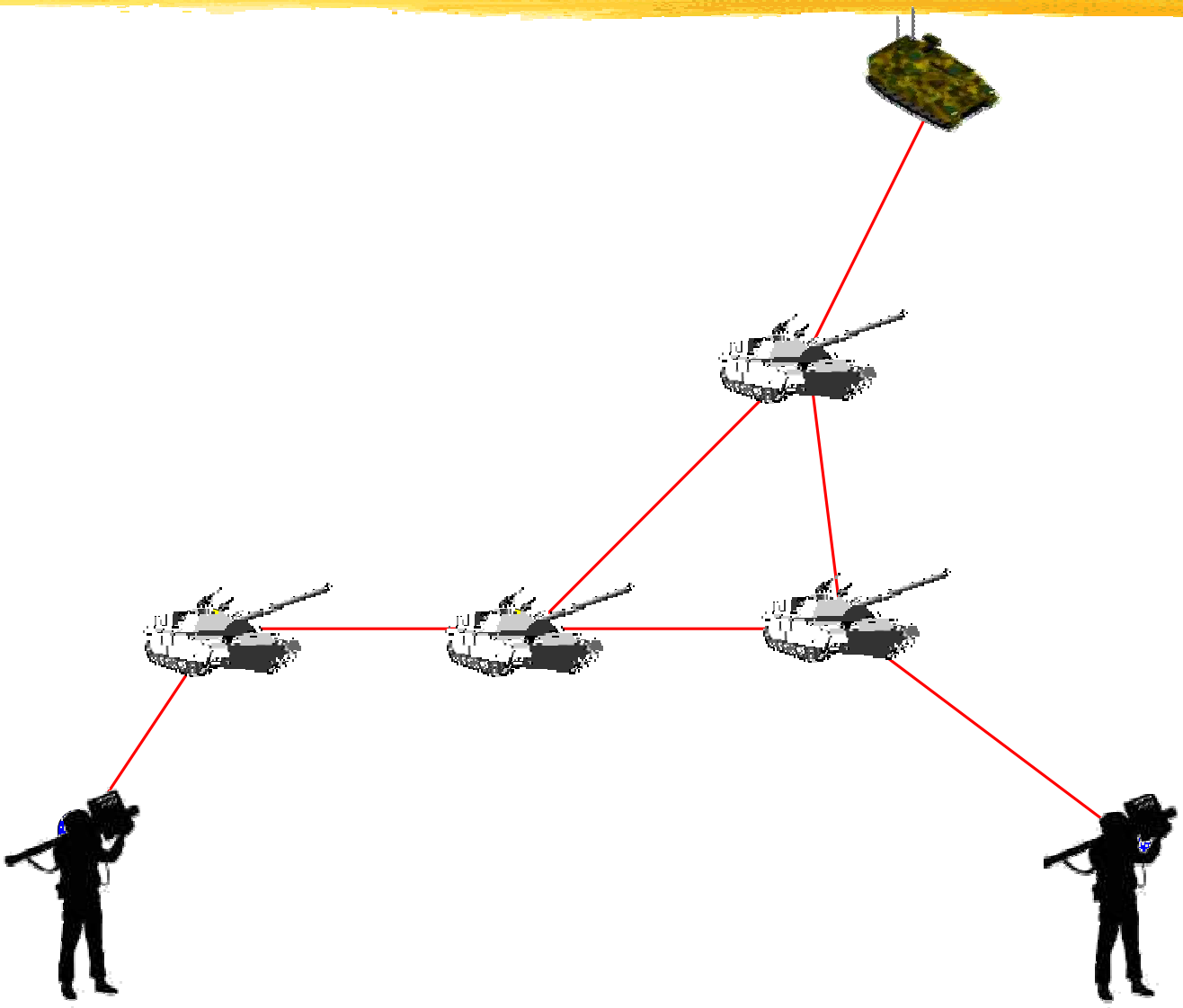
Scenario 1



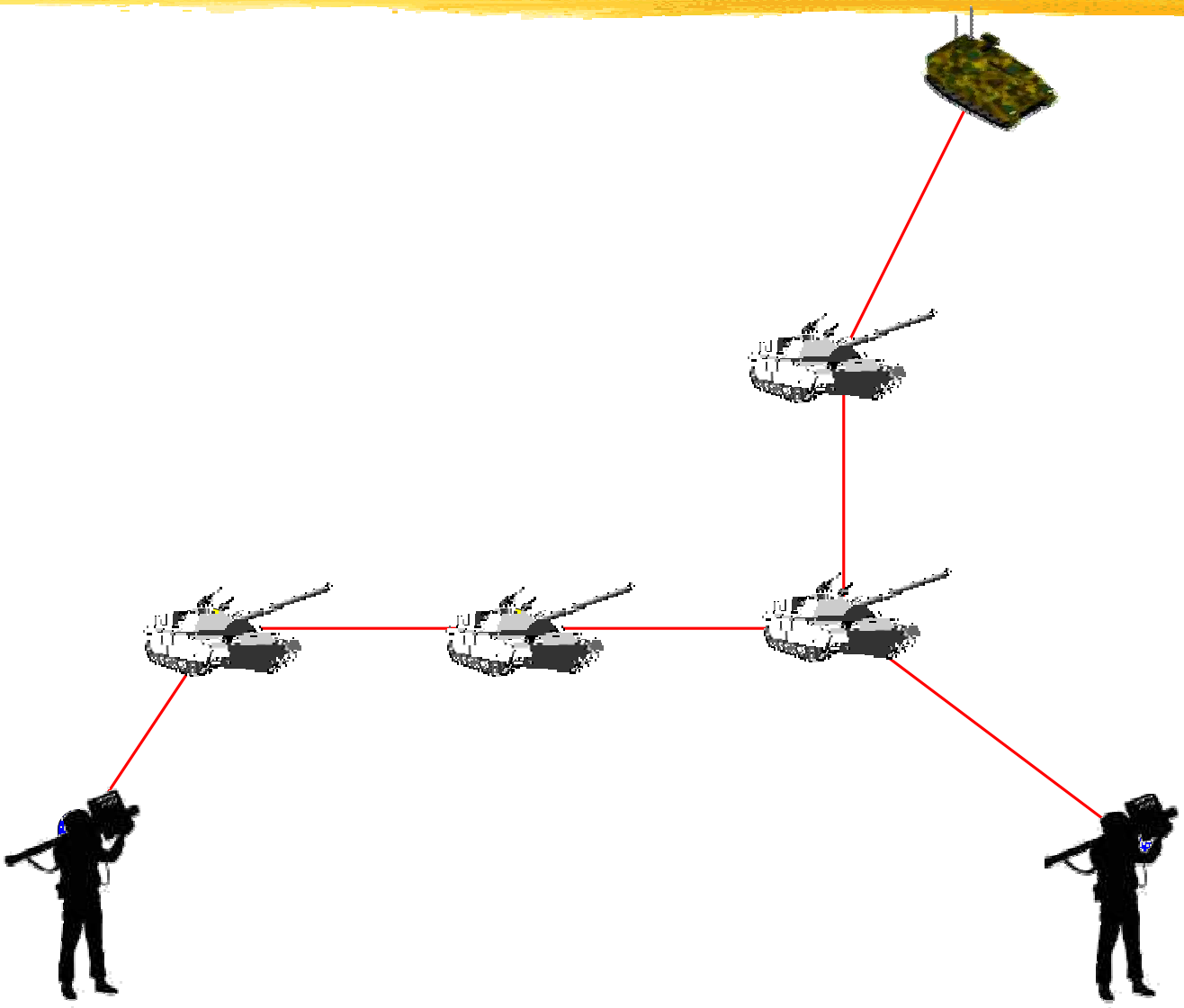
Scenario 1



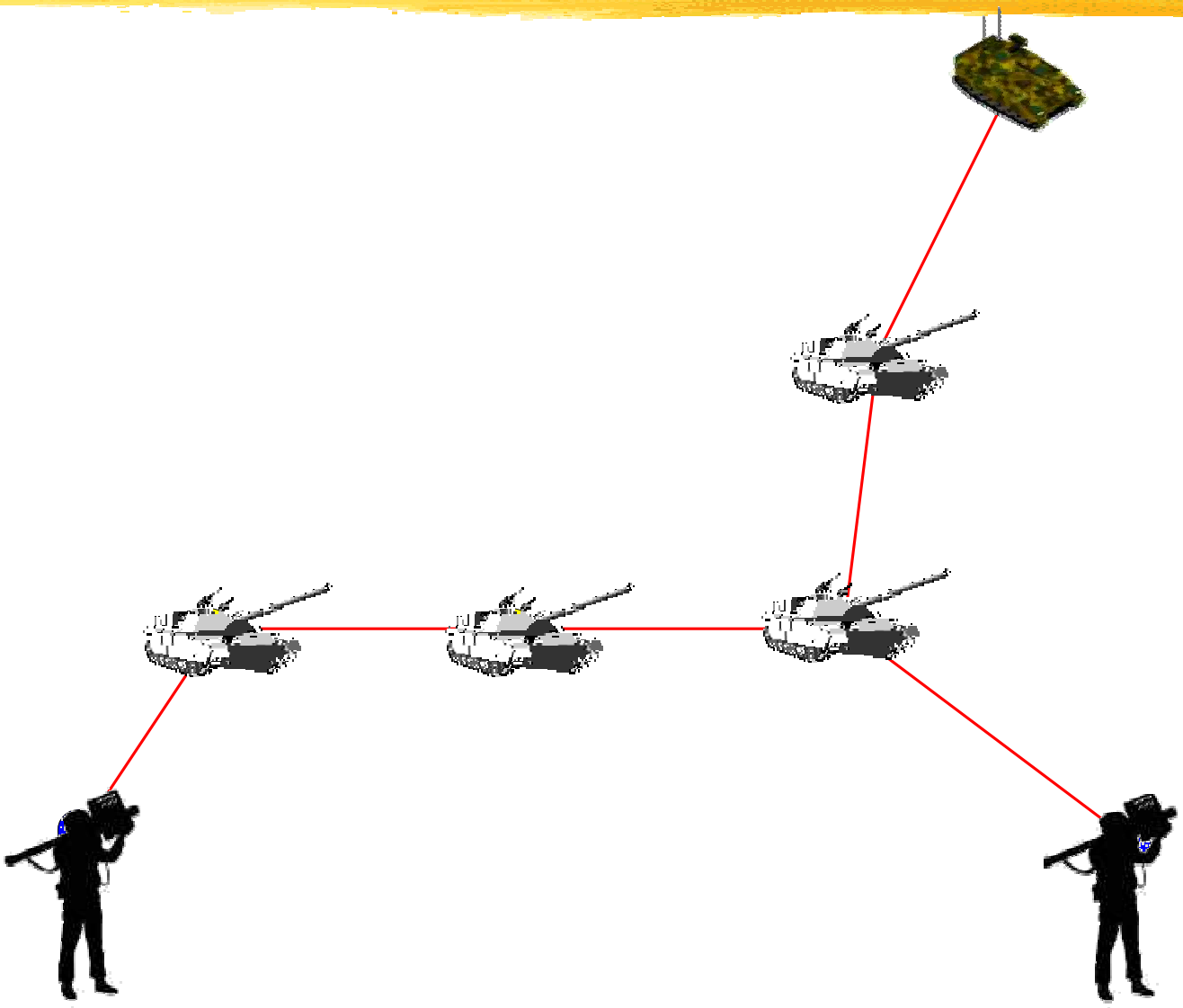
Scenario 1



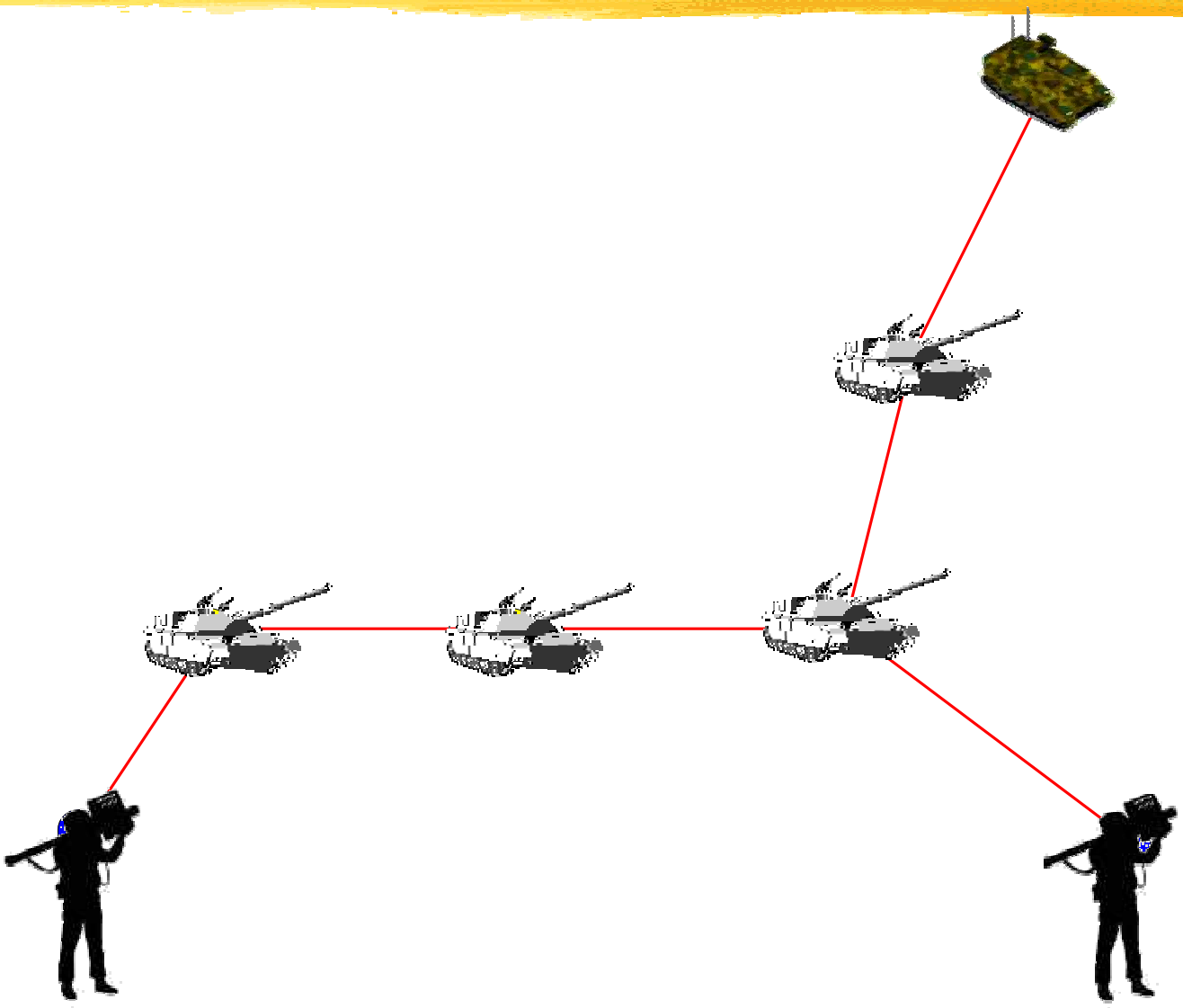
Scenario 1



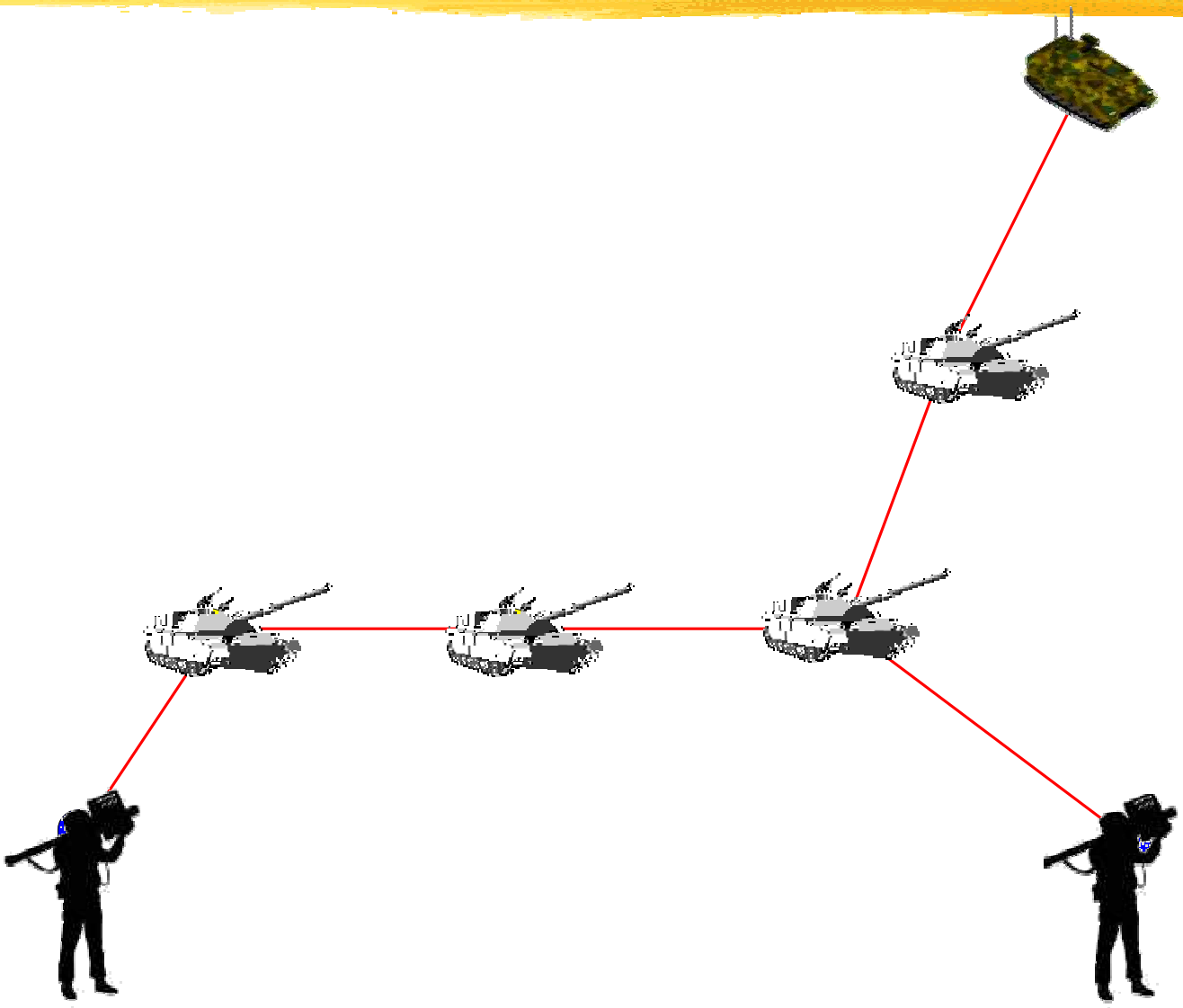
Scenario 1



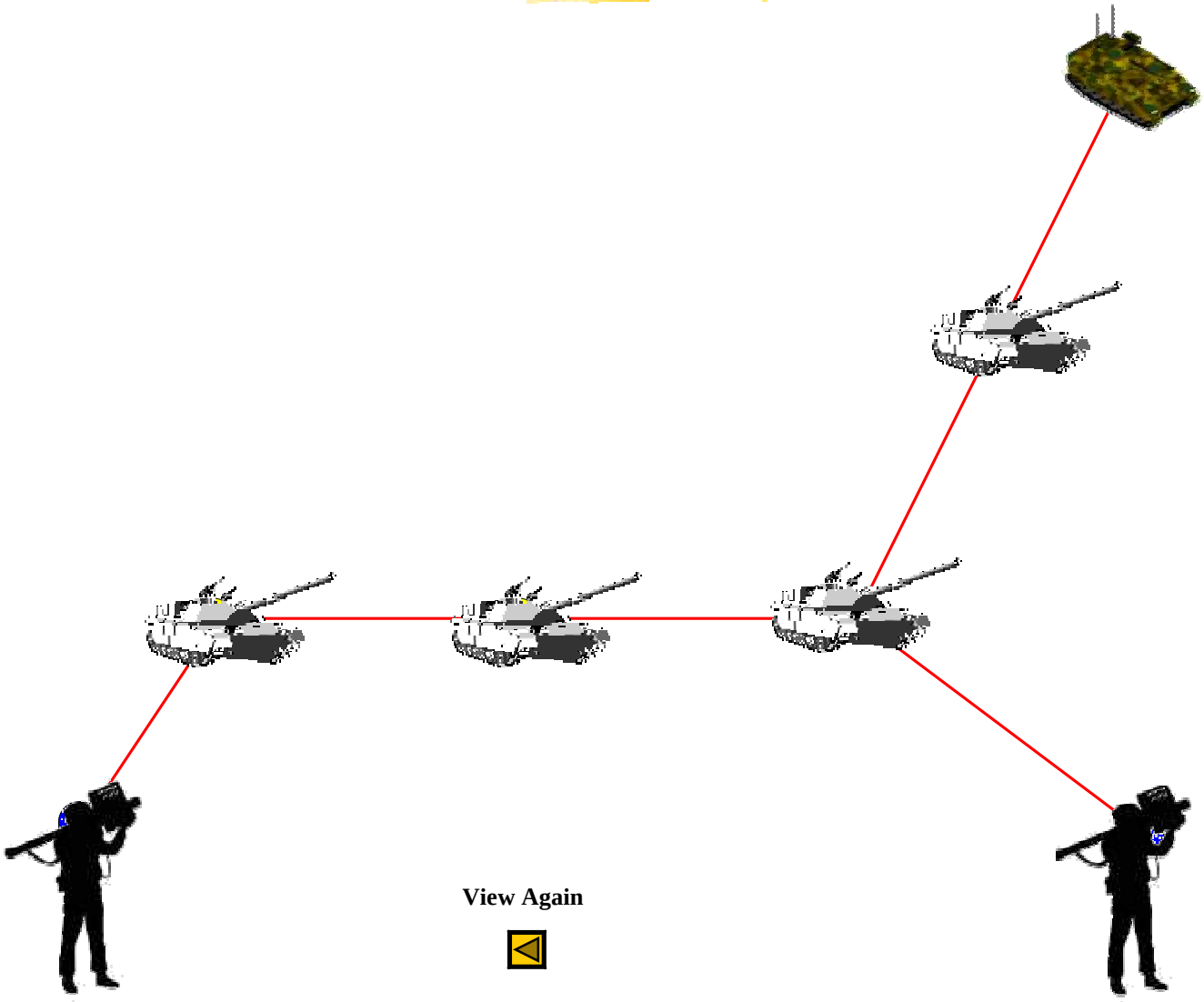
Scenario 1



Scenario 1

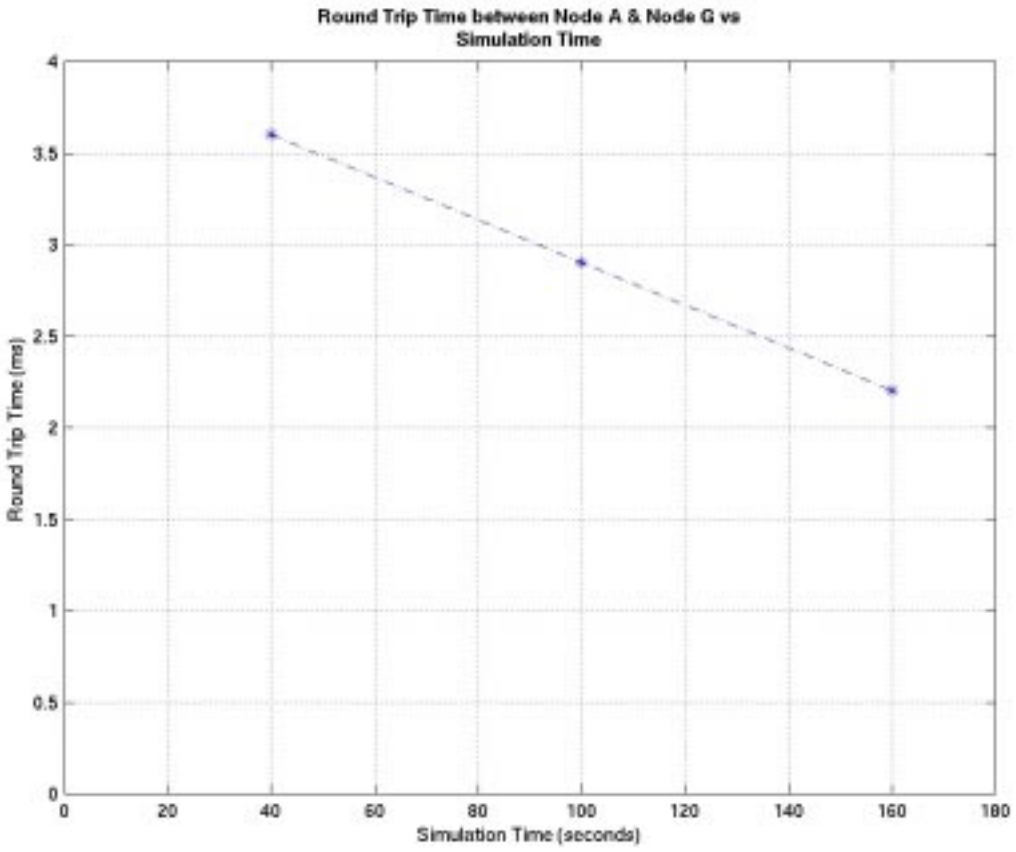


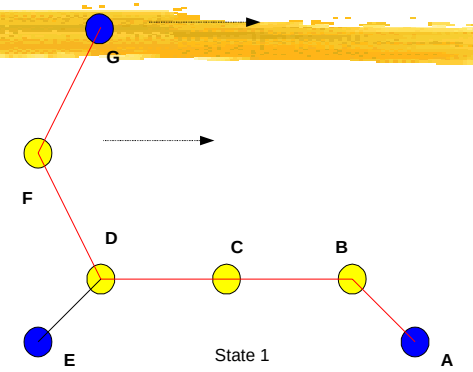
Scenario 1

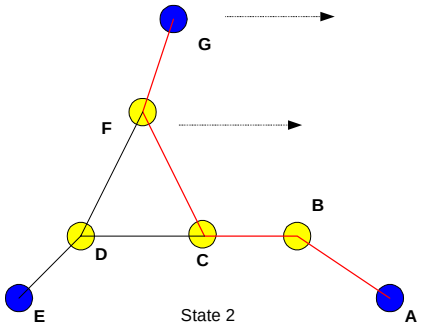


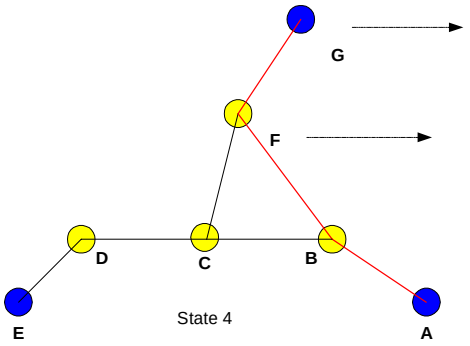
View Again

Results from Scenario 1

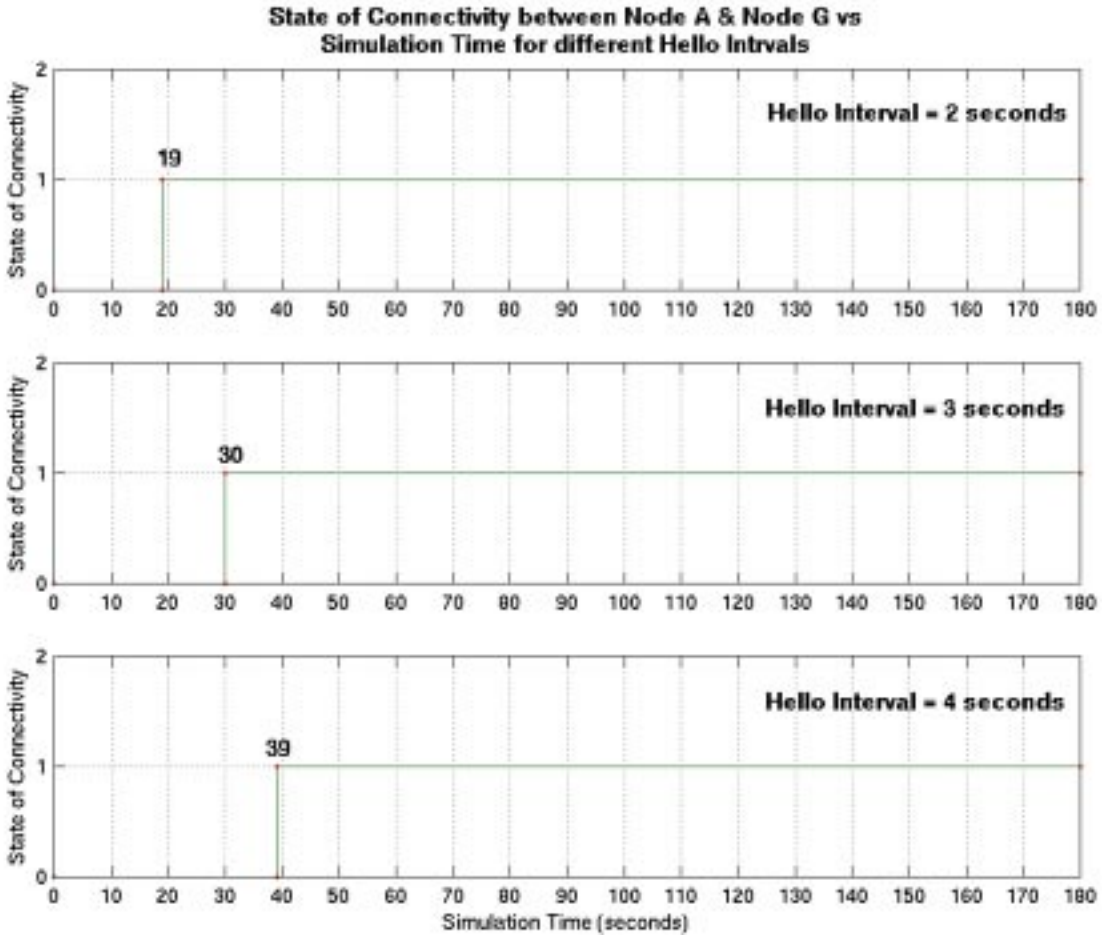


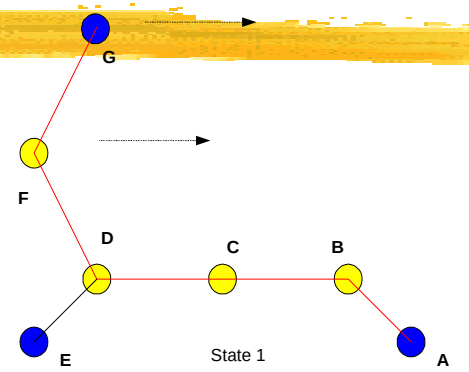


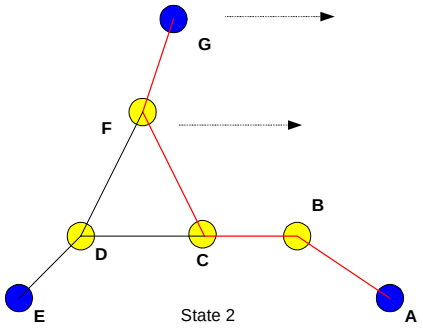


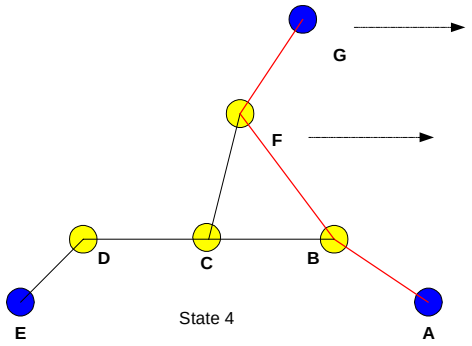


Results from Scenario 1

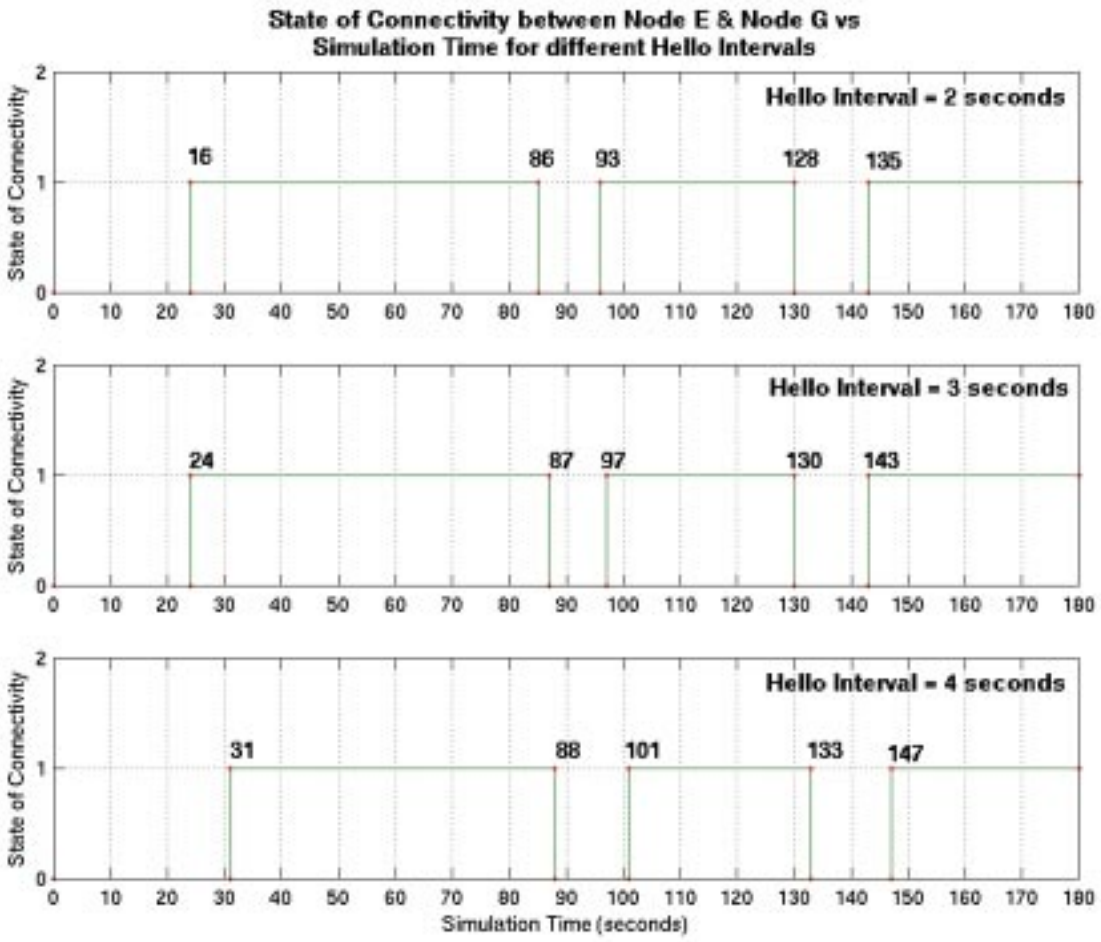


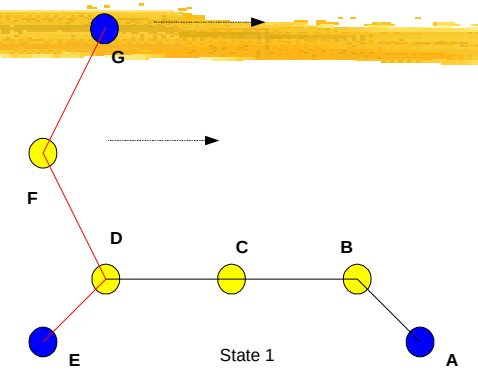


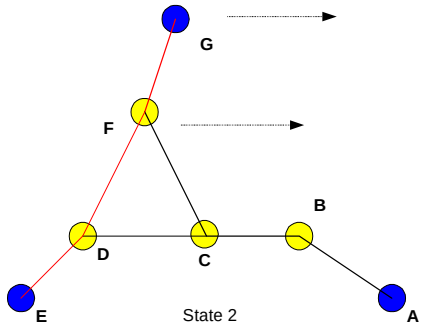


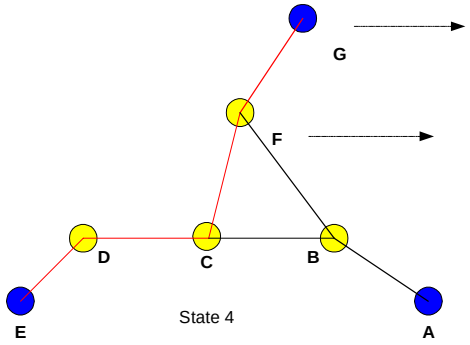


Results from Scenario 1

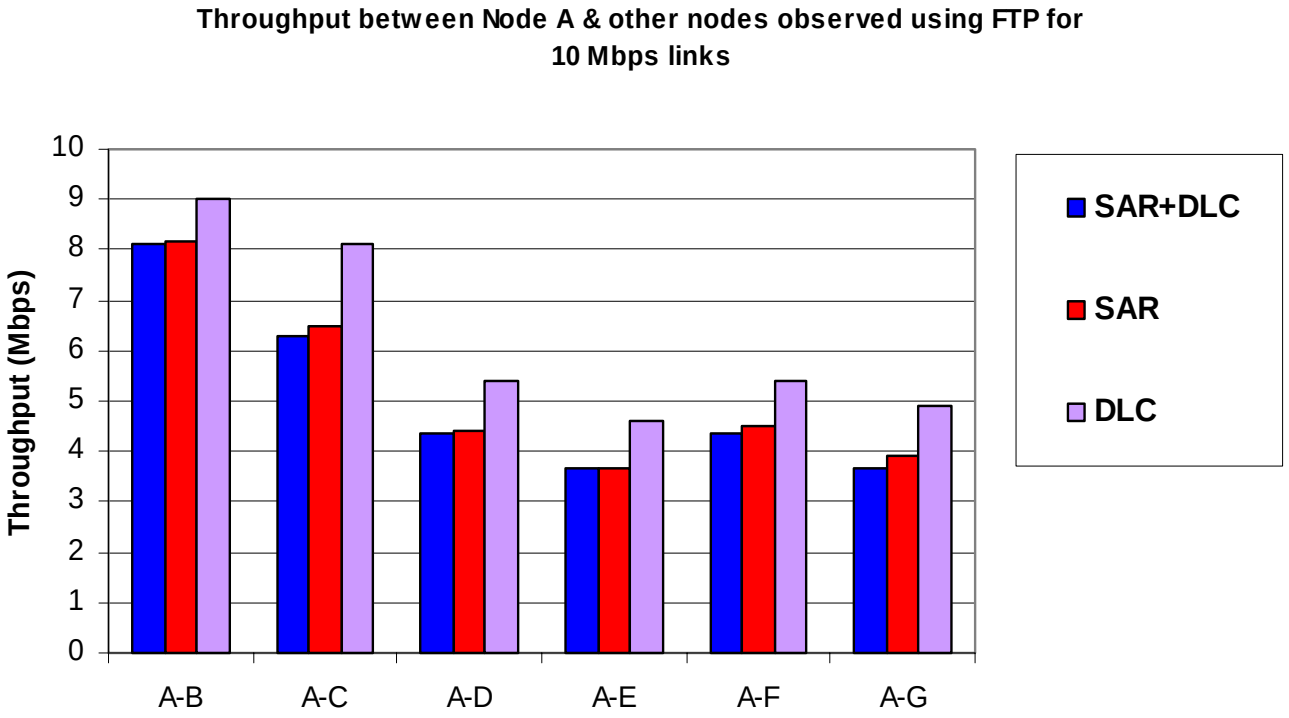


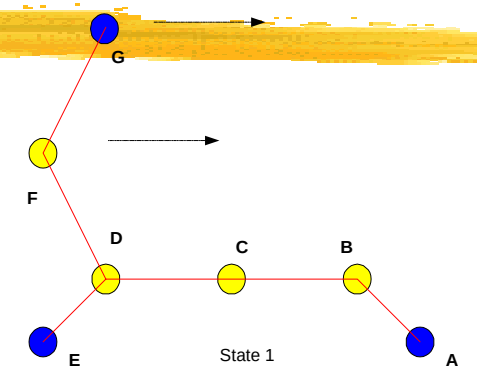


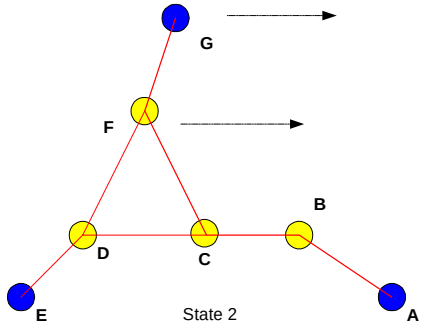


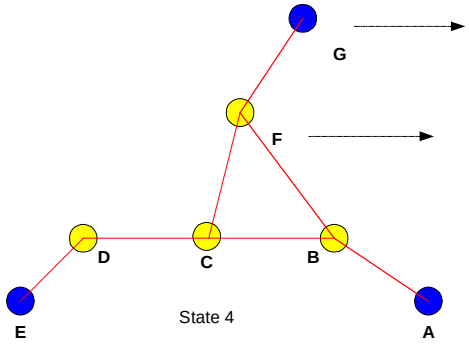


Results from Scenario 1







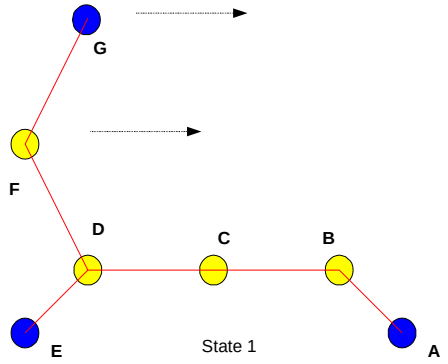


Results from Scenario 1

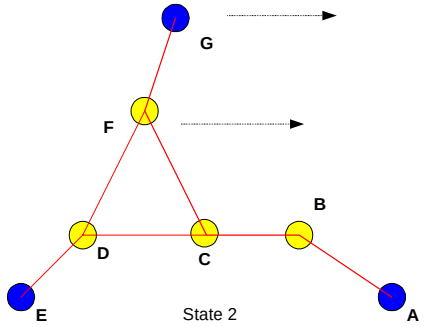
Throughput Between Node A and Node G for SAR+QoS+DLC

# of Packets	Size of the Packets (Bytes)	Tx Rate (Mbps)	Rx Rate (Mbps)
2048	512	3.6618	3.6563
2048	1054	6.7584	6.5782
2048	1536	9.8877	9.6583

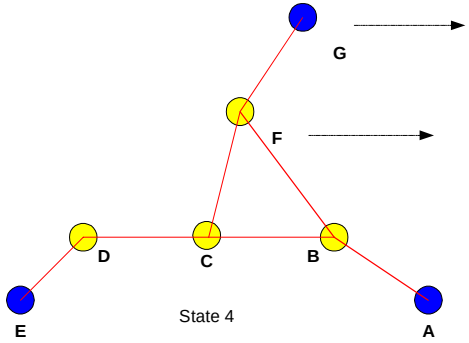
Source-Destination Pair	SAR+DLC (Mbps)	DLC (Mbps)
A-G	3.6864	4.9152



State 1

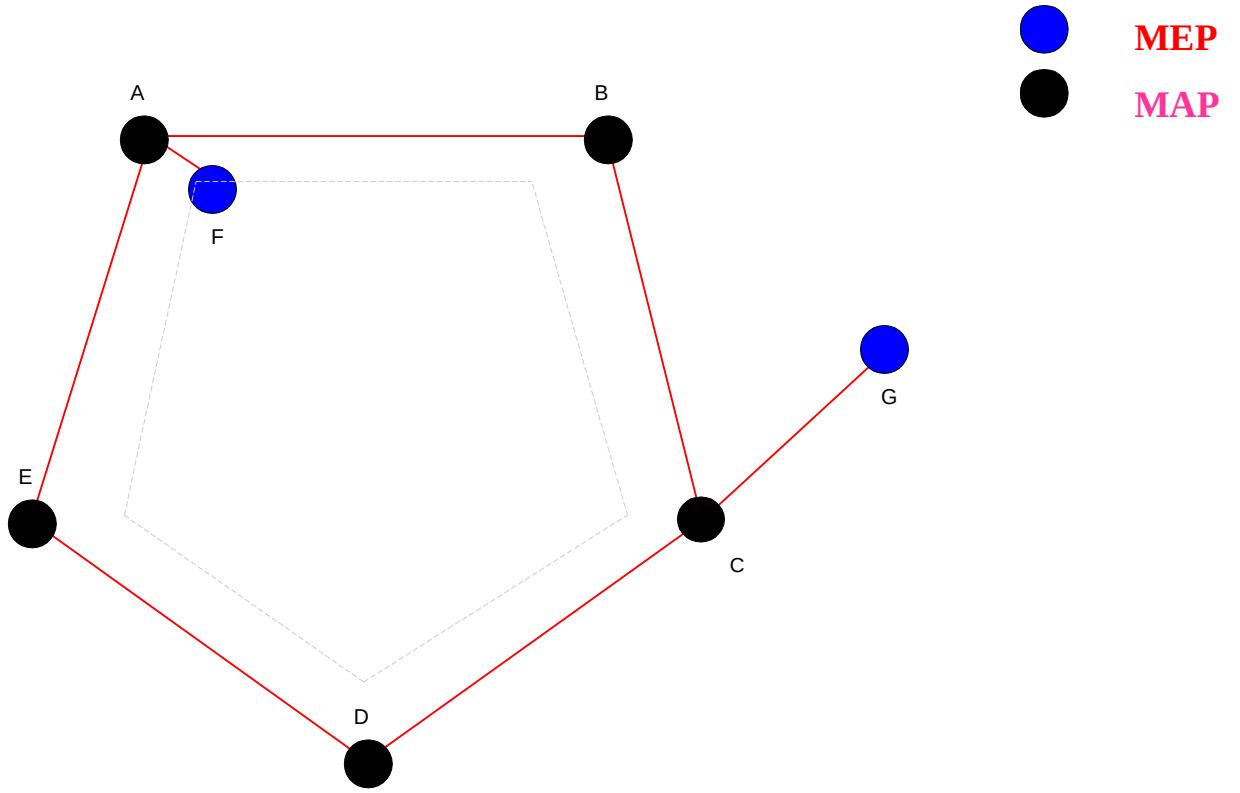


State 2

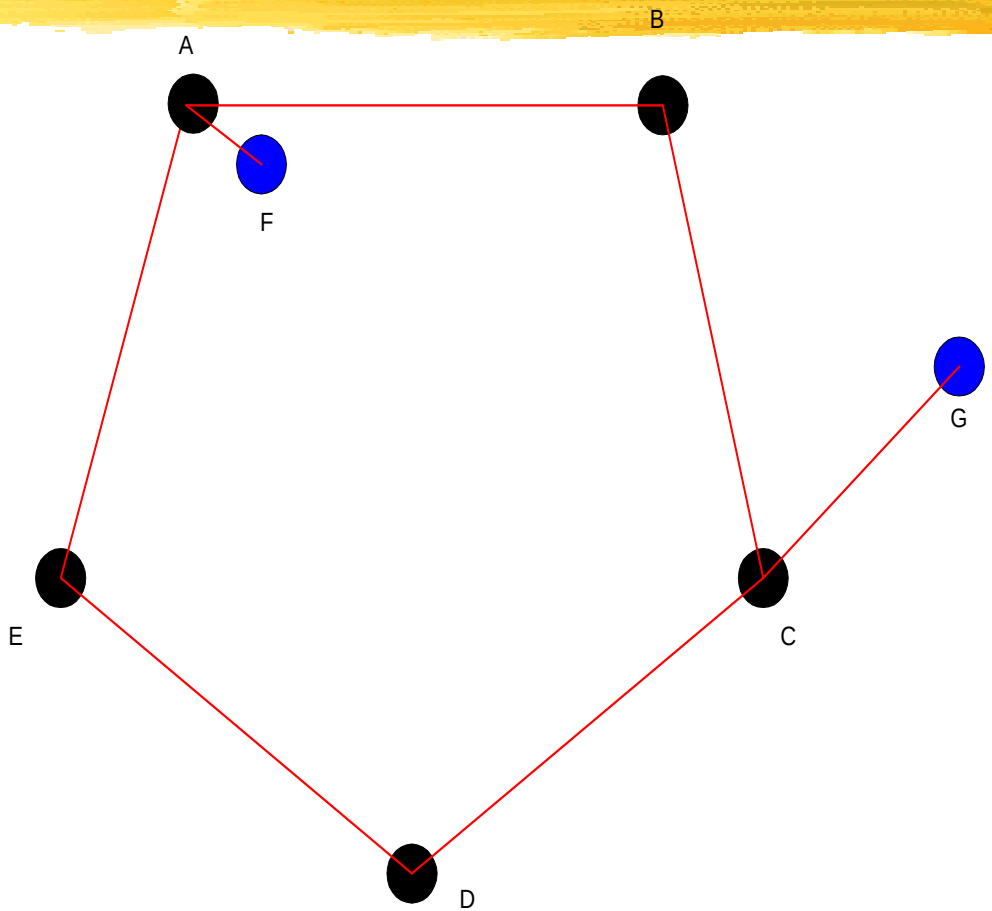


State 4

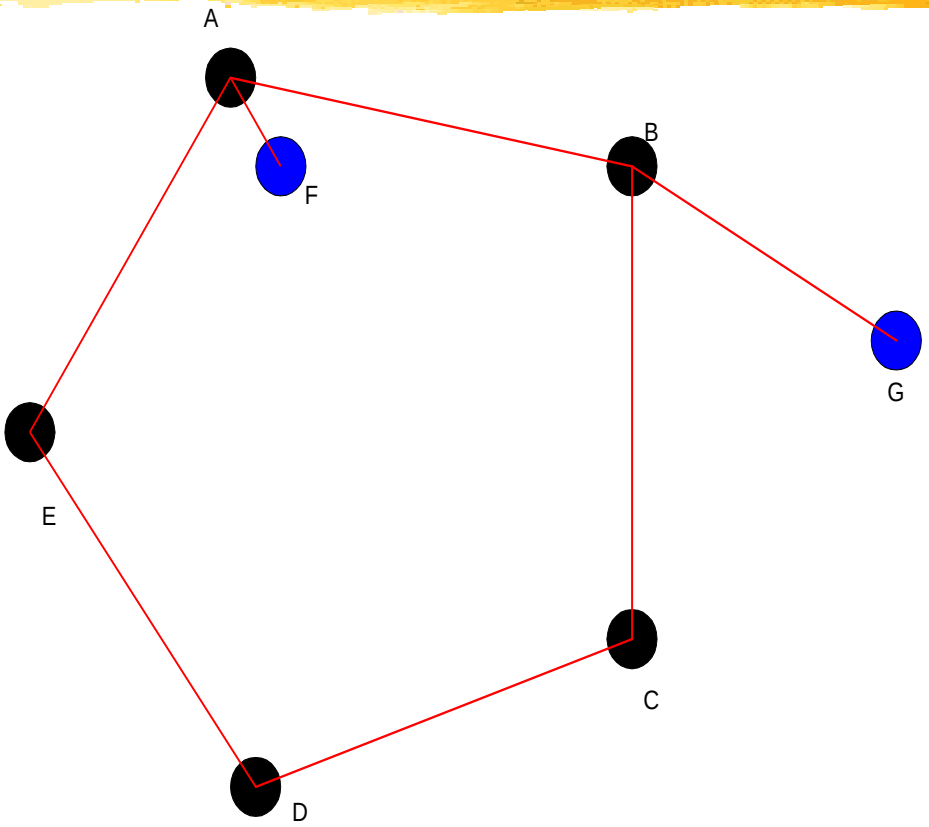
Scenario 2



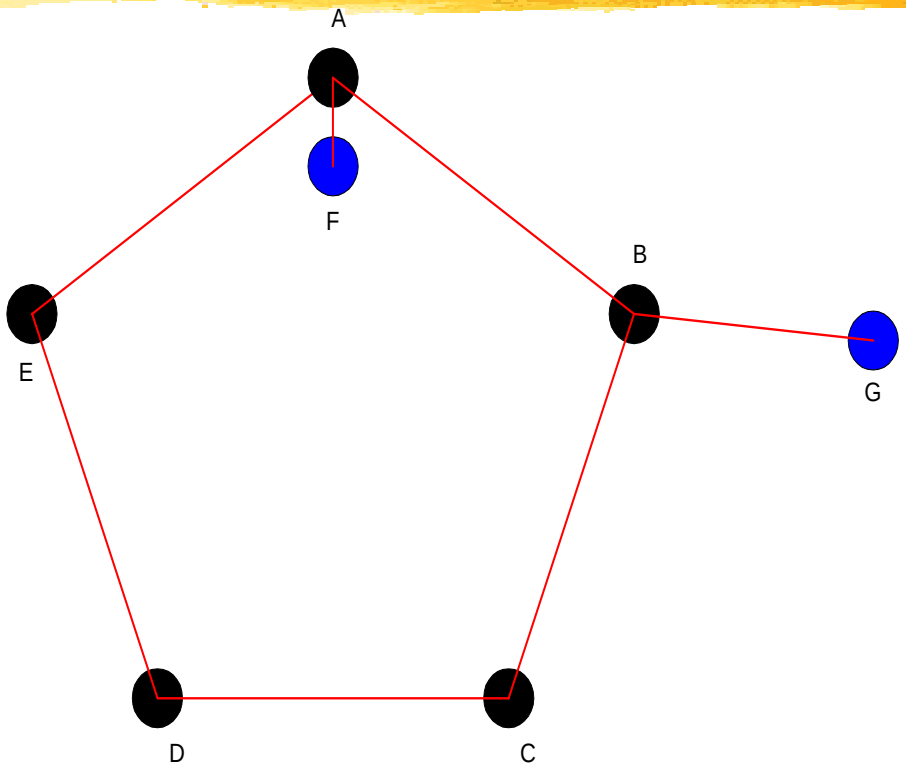
Scenario 2



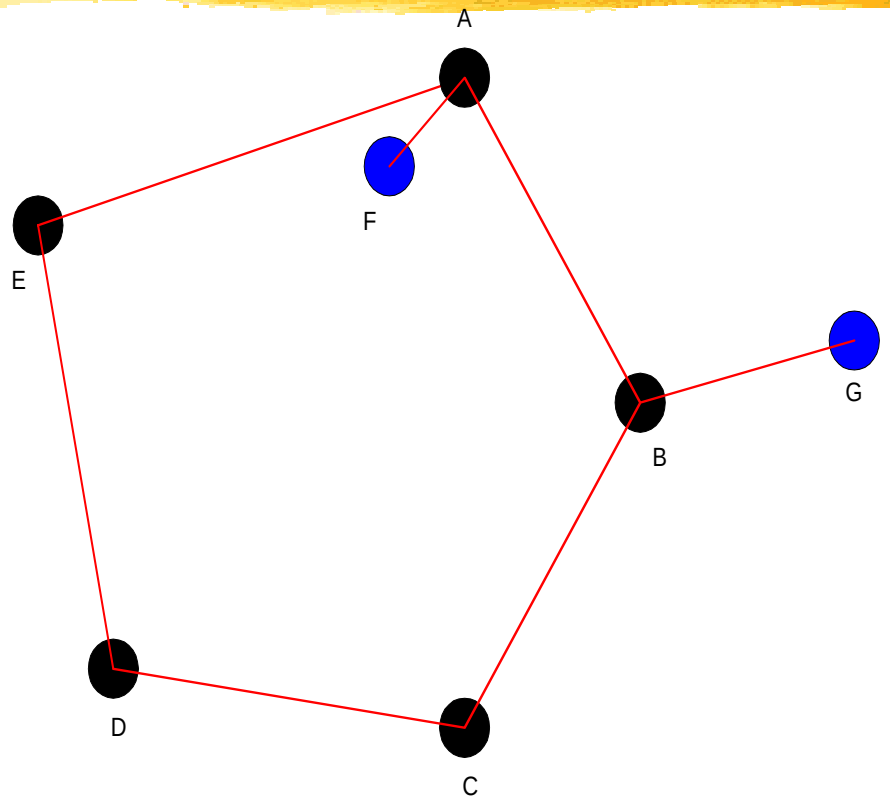
Scenario 2



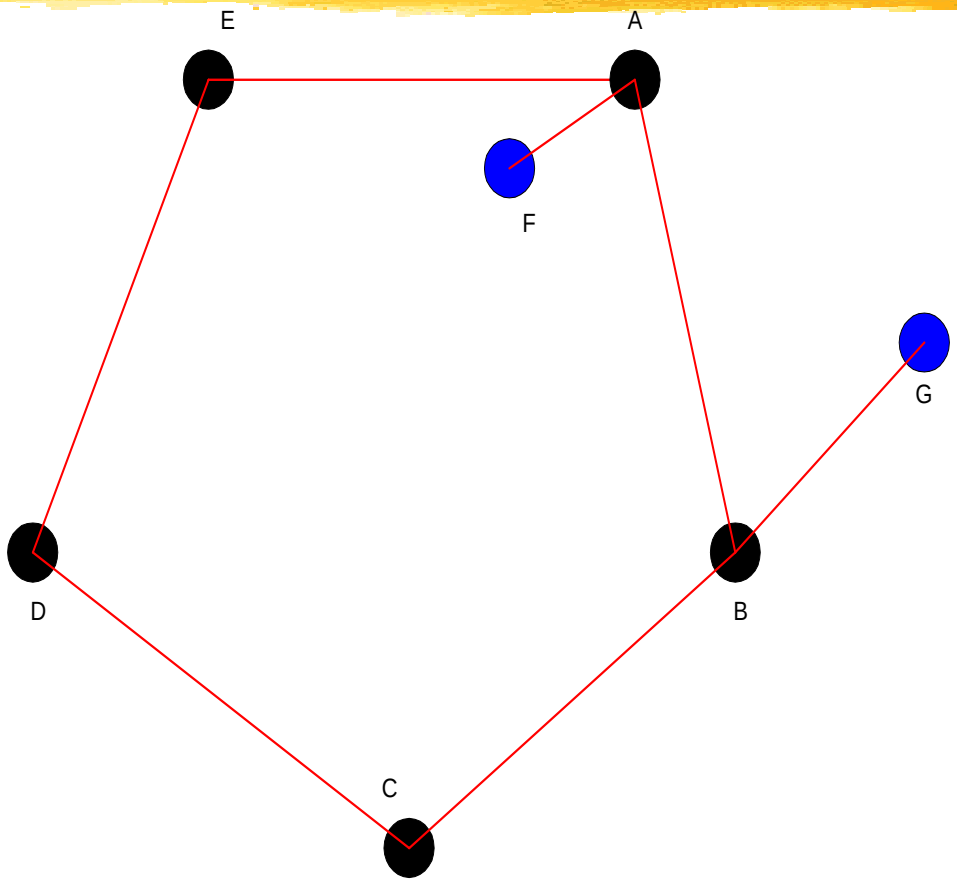
Scenario 2



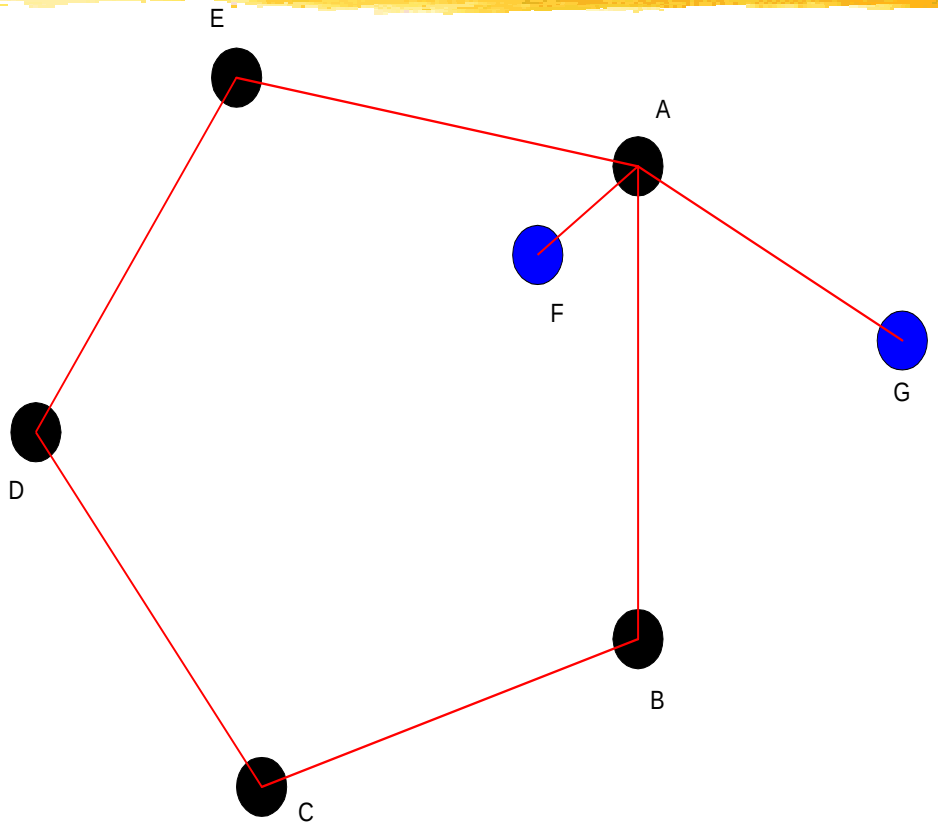
Scenario 2



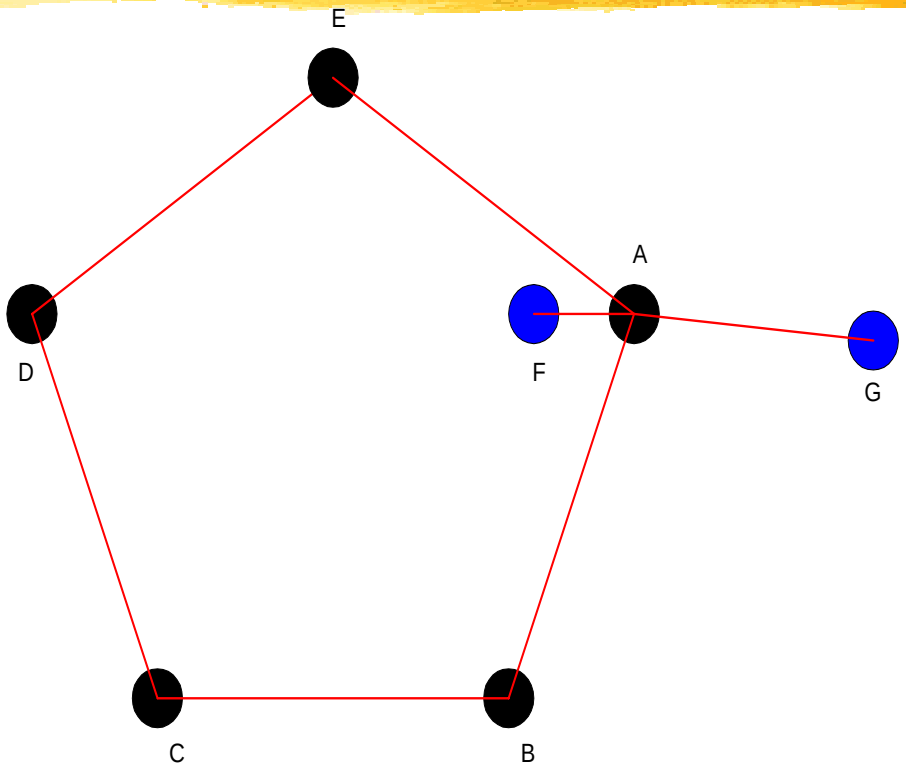
Scenario 2



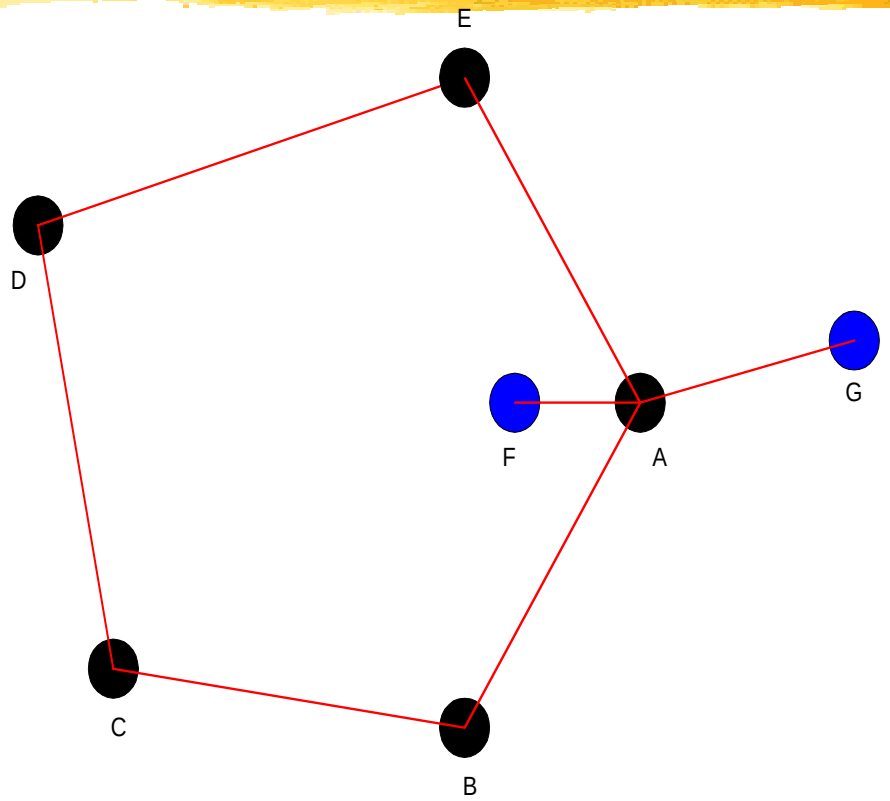
Scenario 2



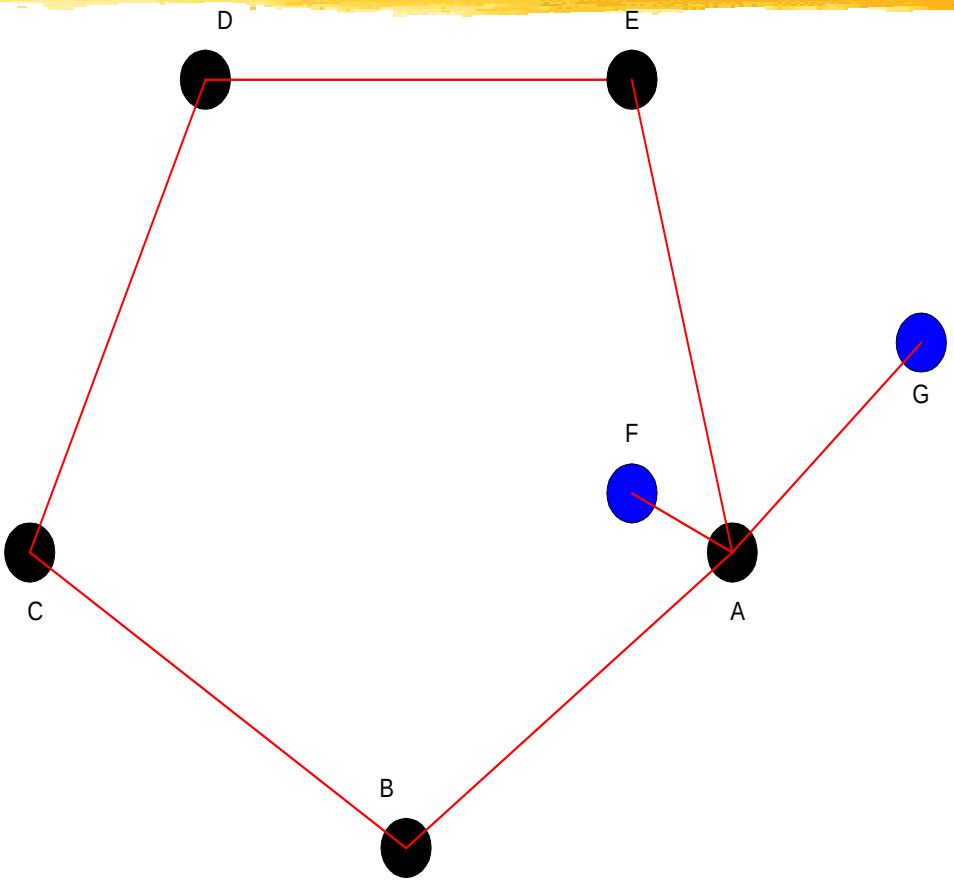
Scenario 2



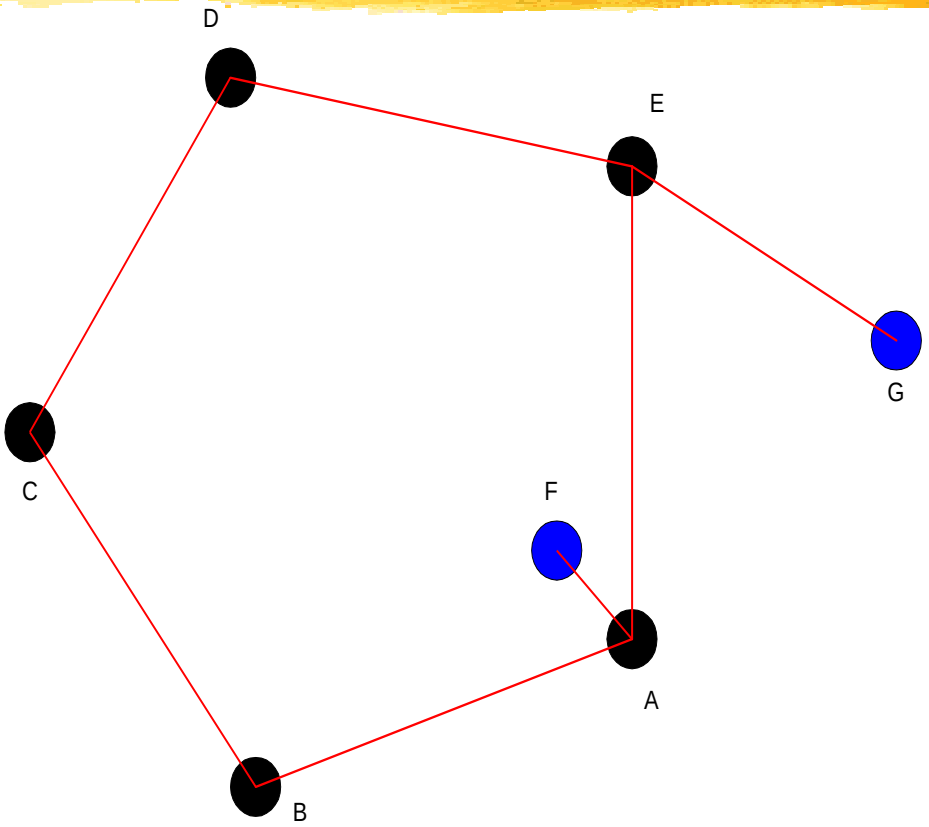
Scenario 2



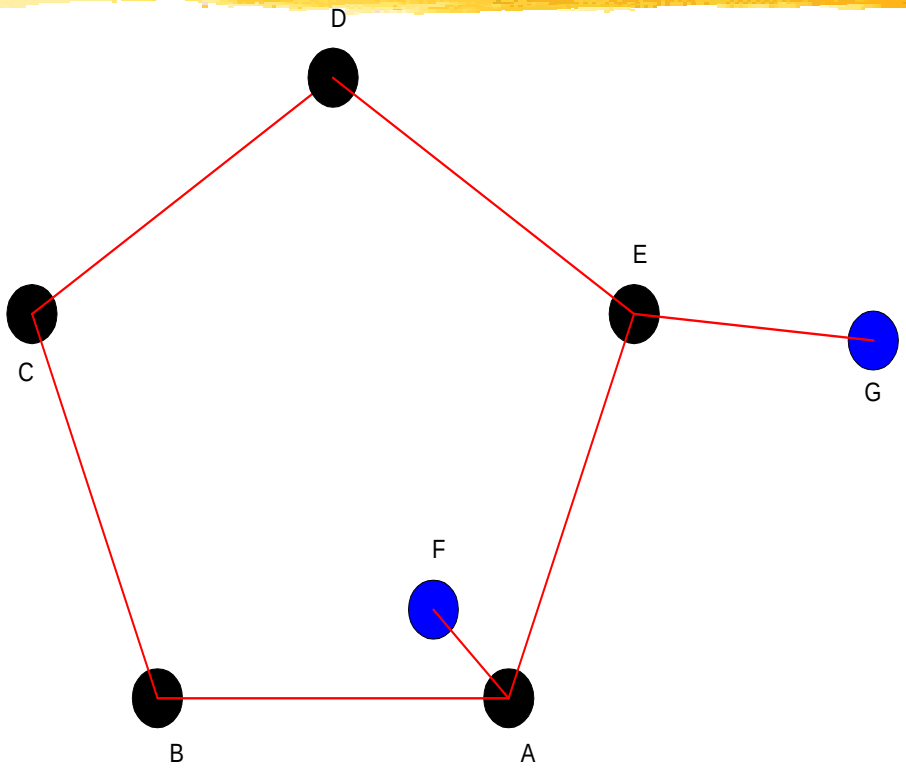
Scenario 2



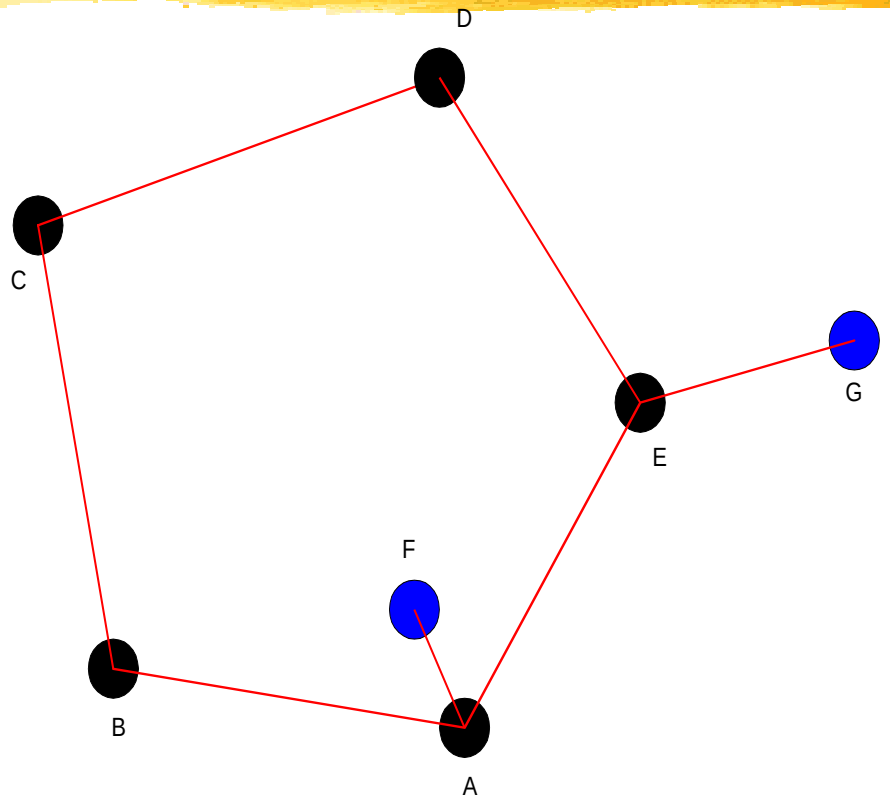
Scenario 2



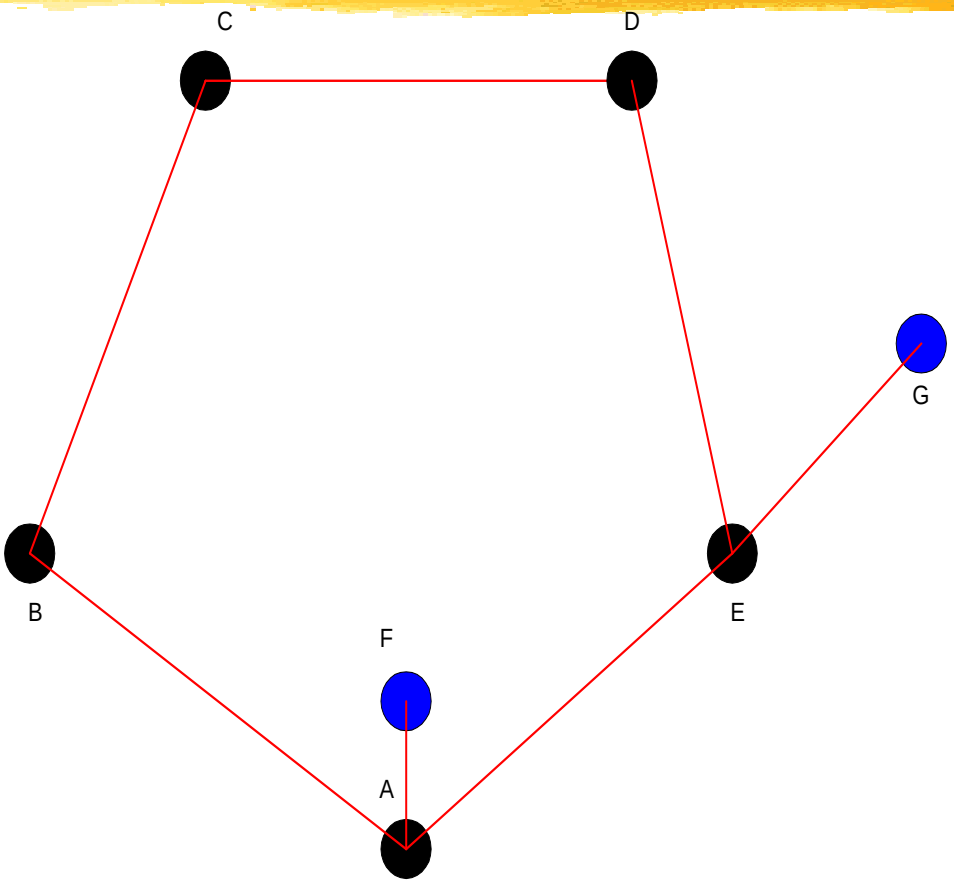
Scenario 2



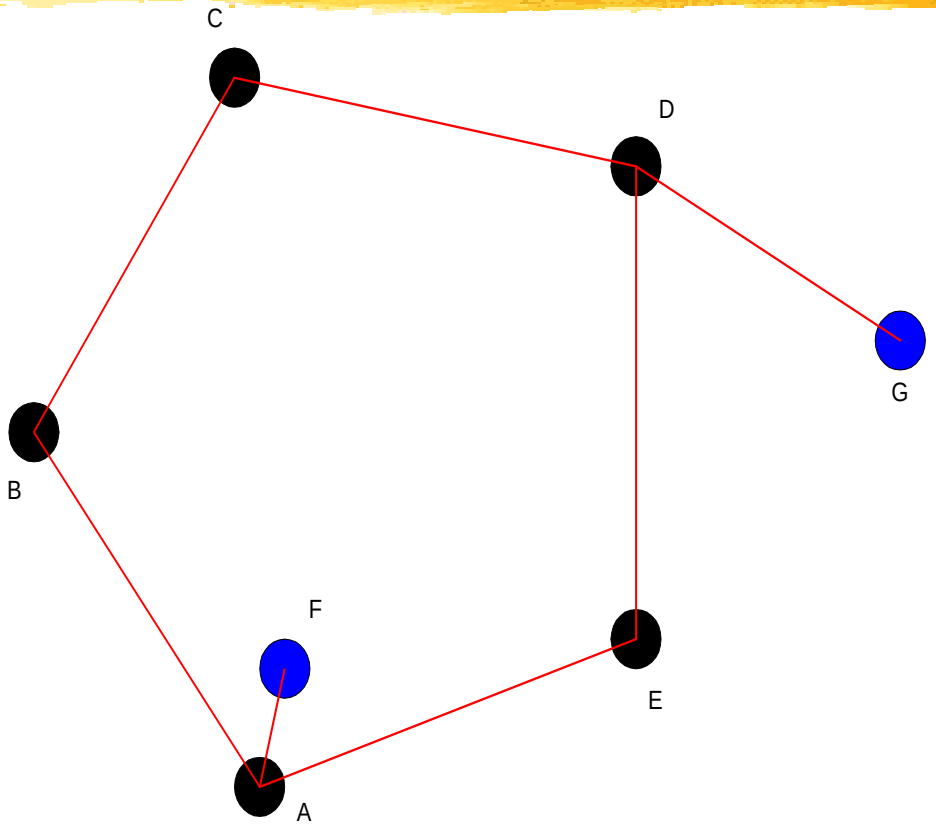
Scenario 2



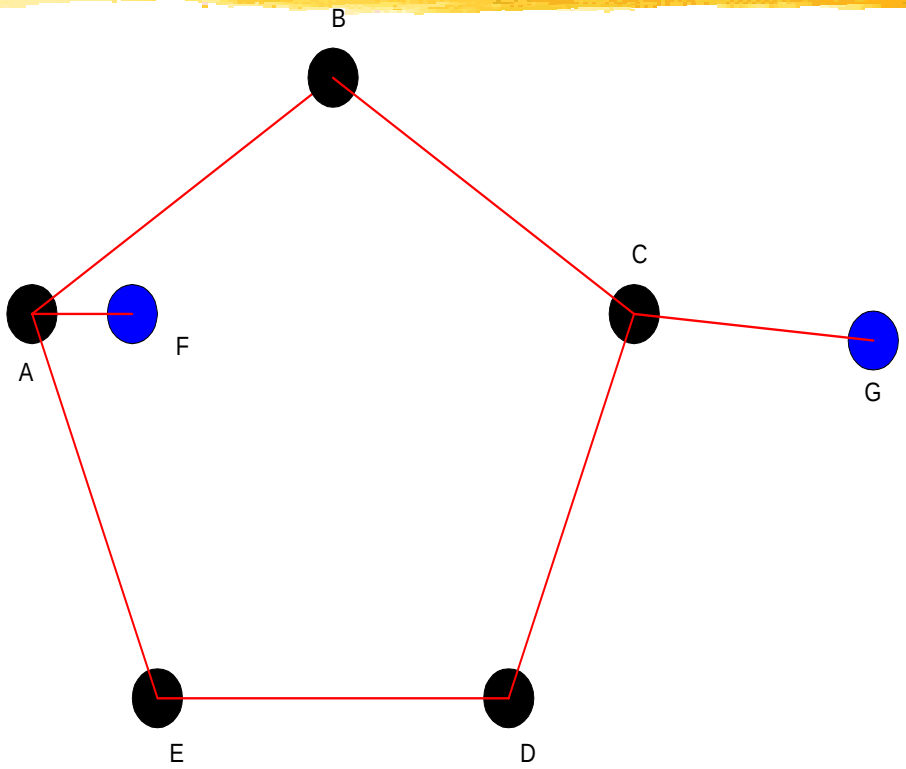
Scenario 2



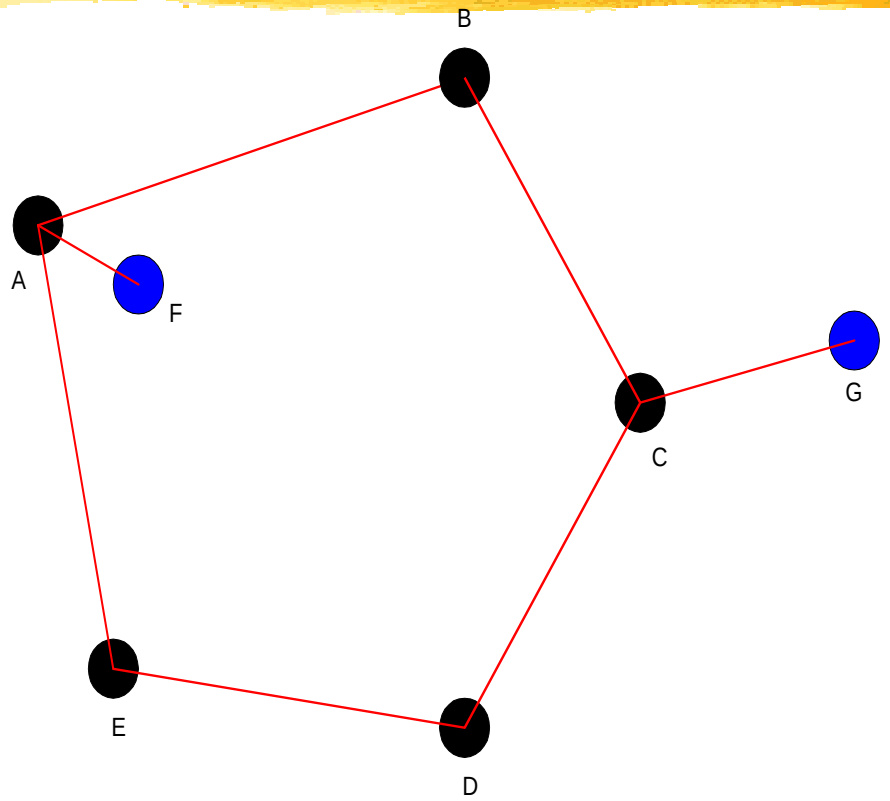
Scenario 2



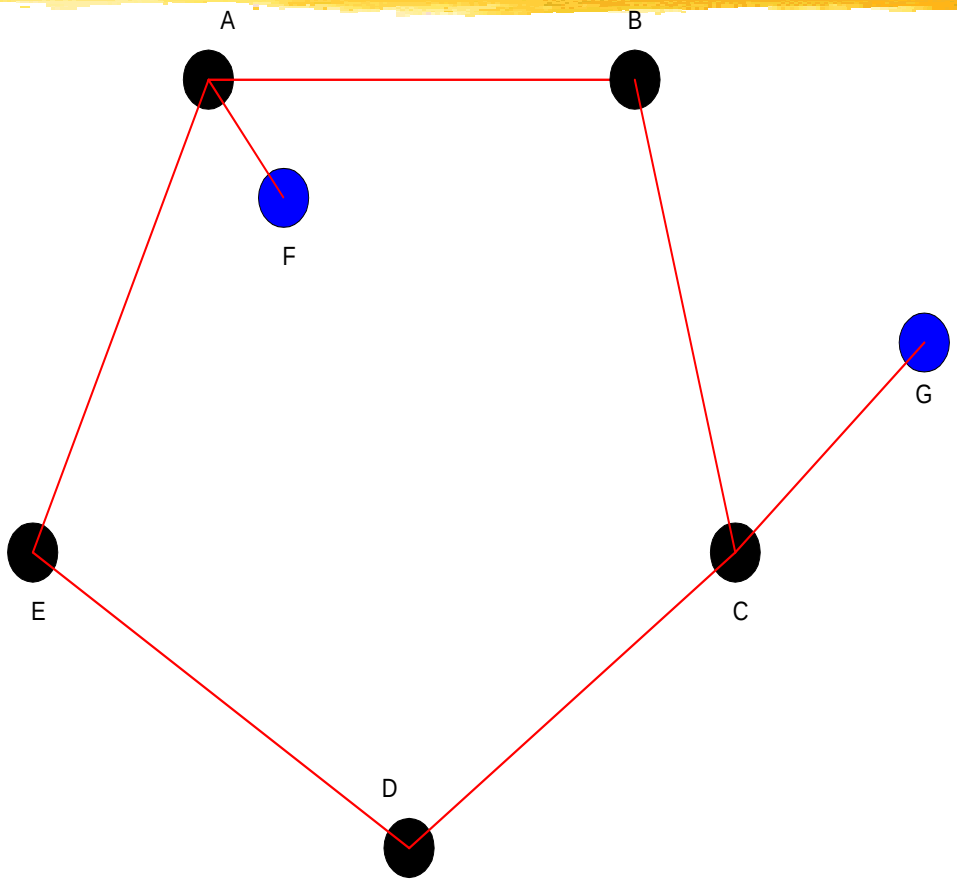
Scenario 2



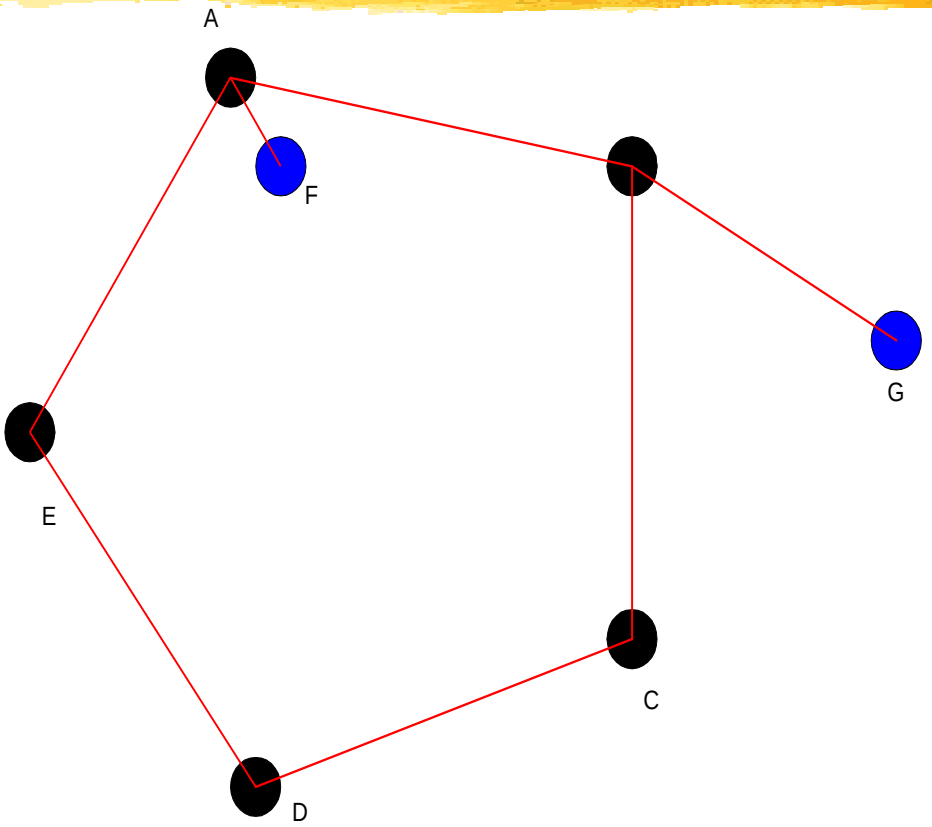
Scenario 2



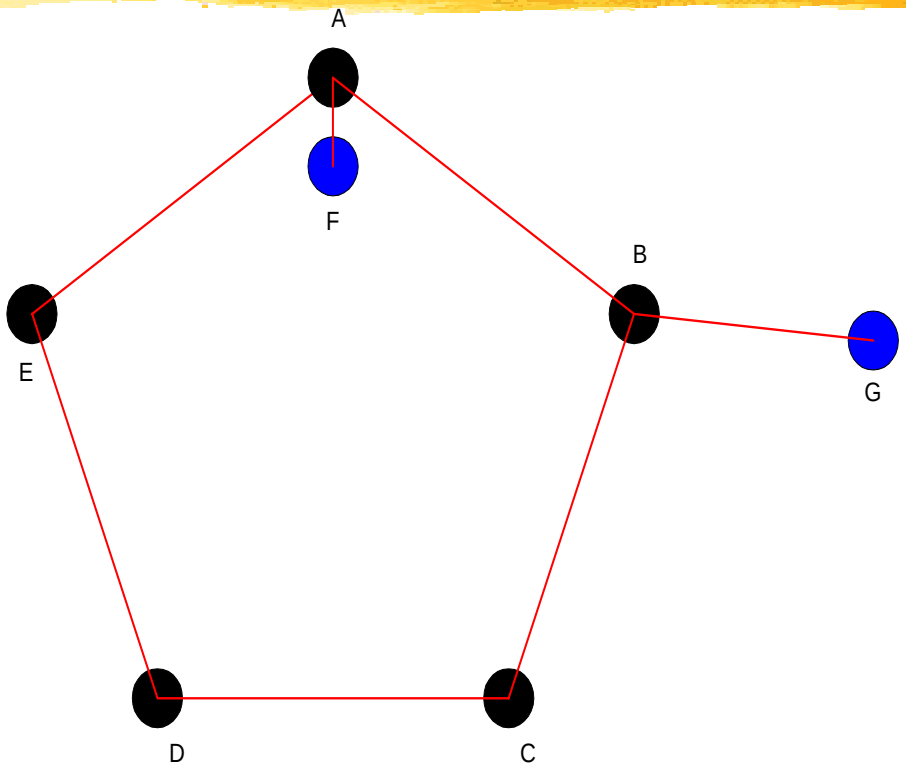
Scenario 2



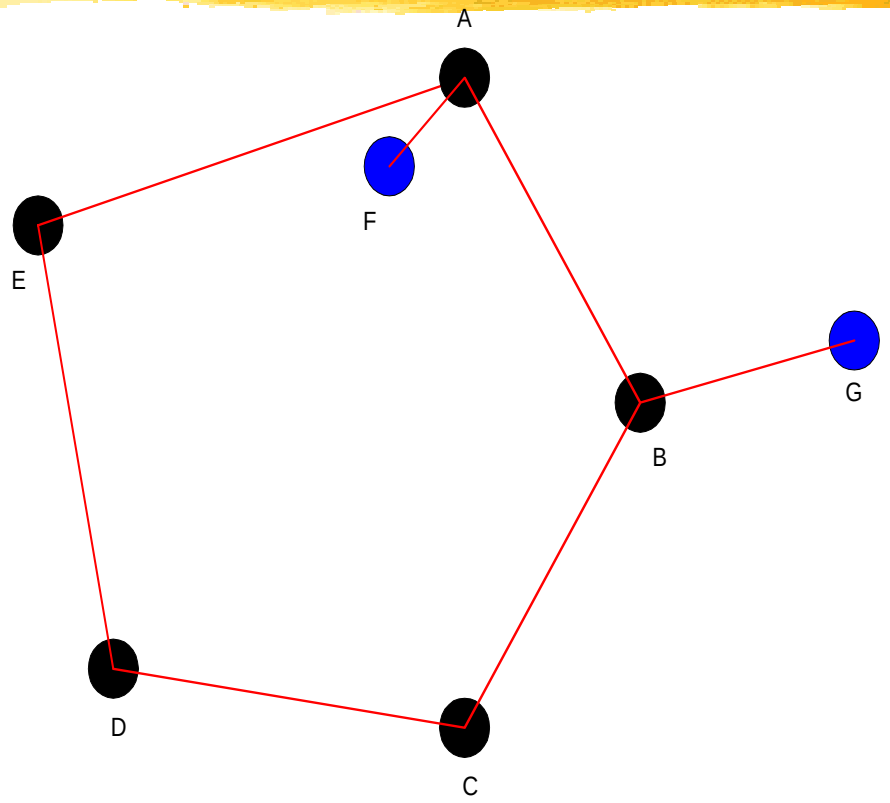
Scenario 2



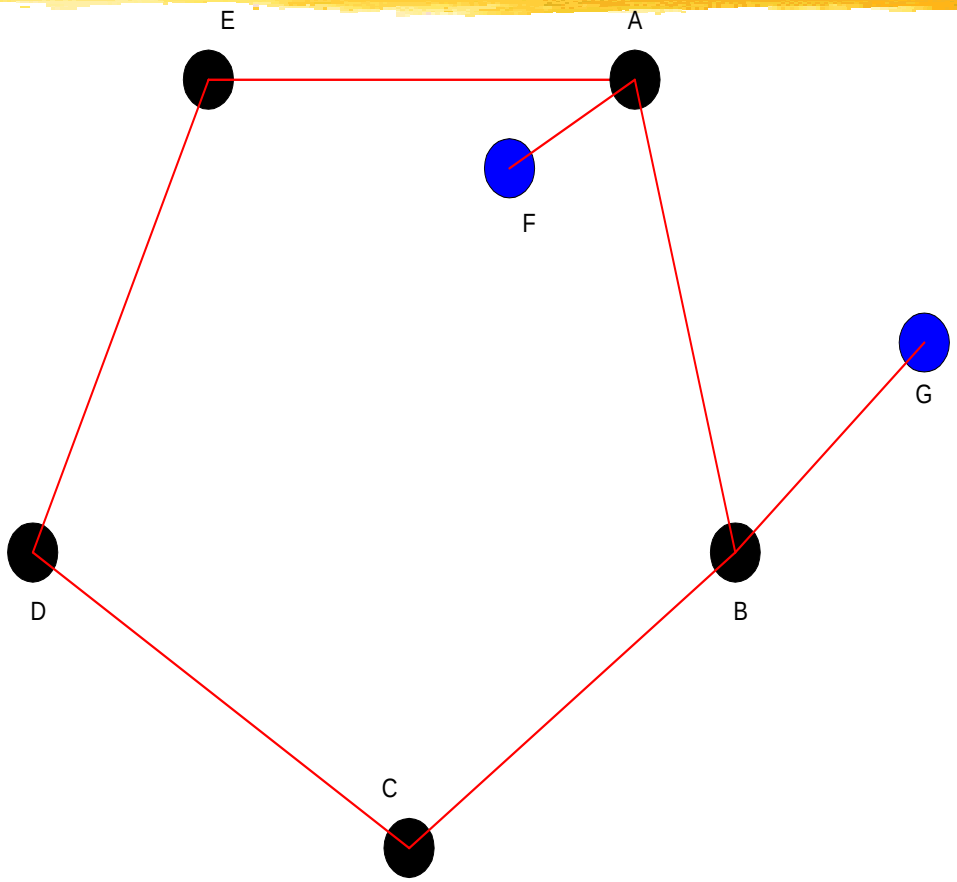
Scenario 2



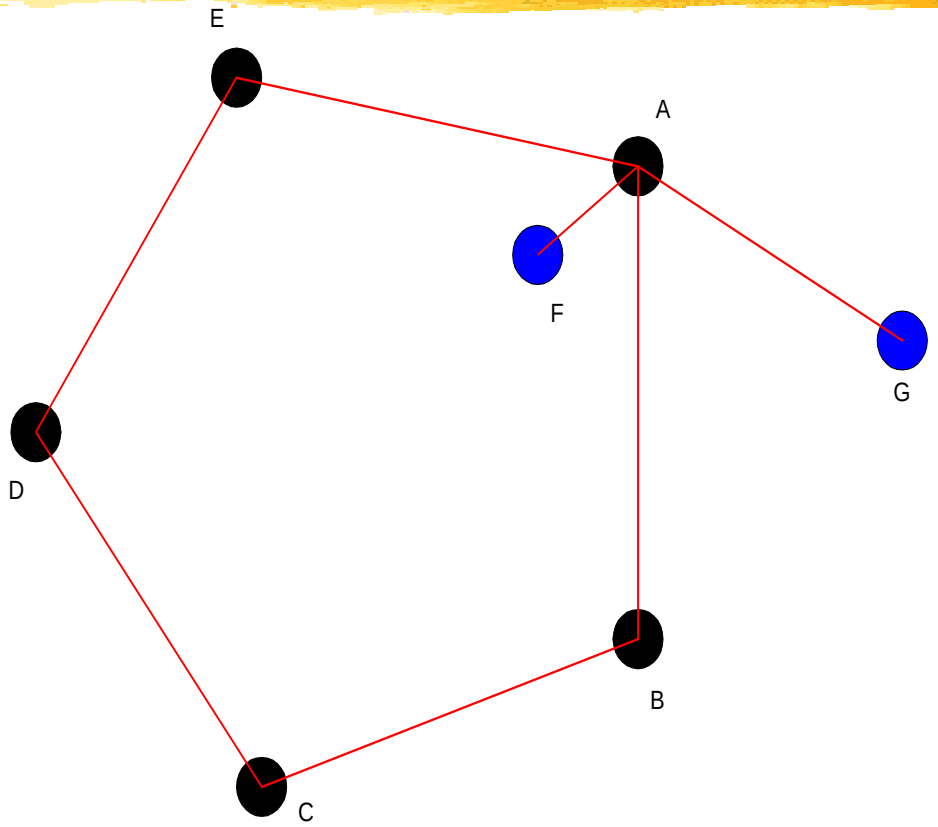
Scenario 2



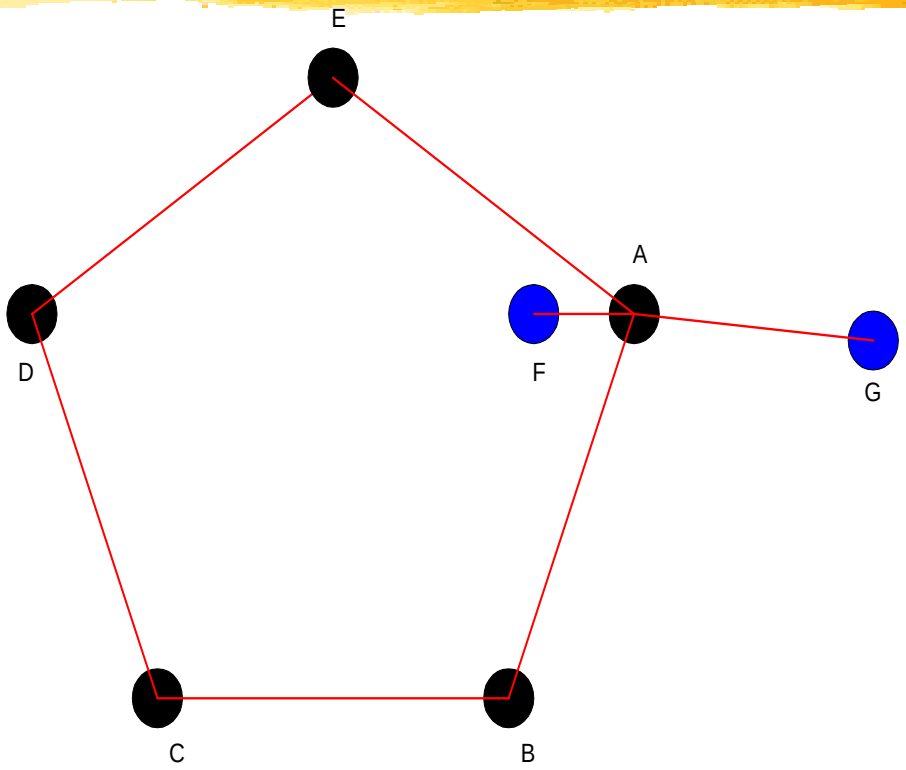
Scenario 2



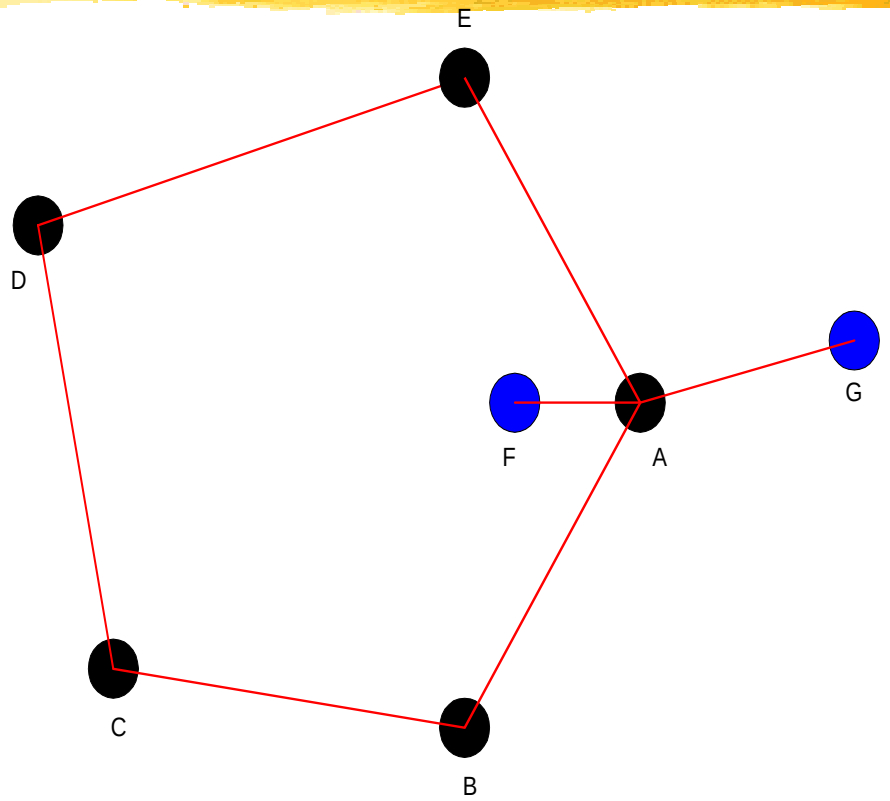
Scenario 2



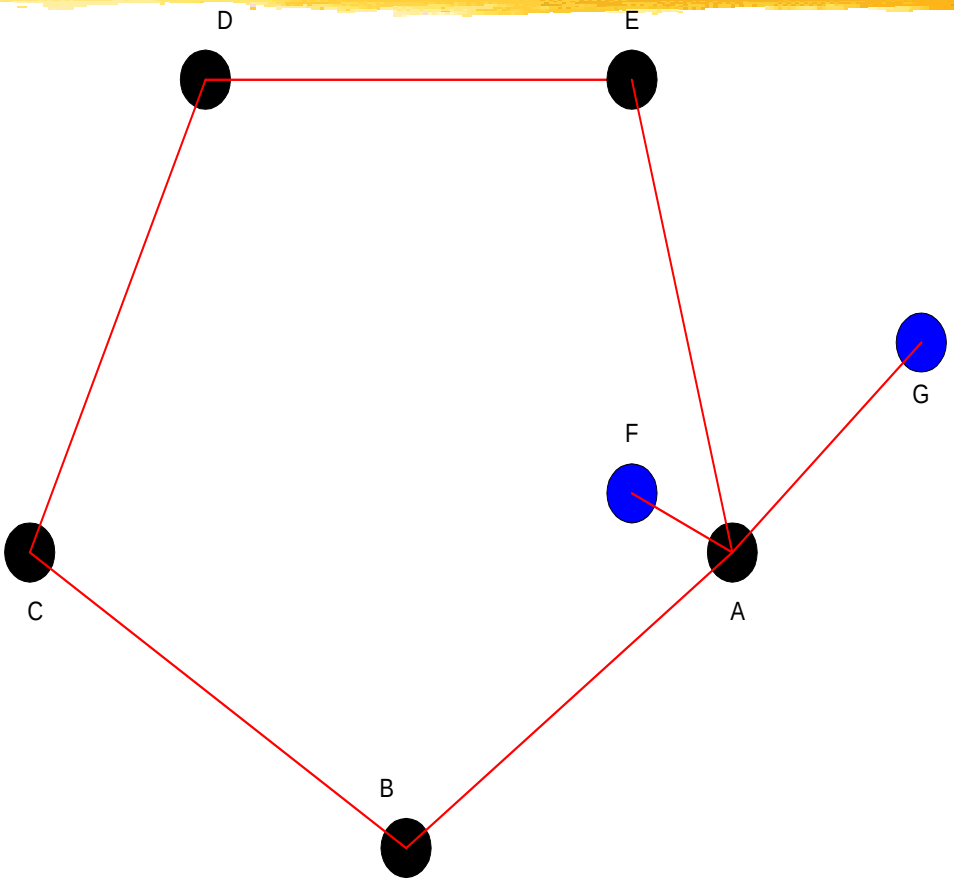
Scenario 2



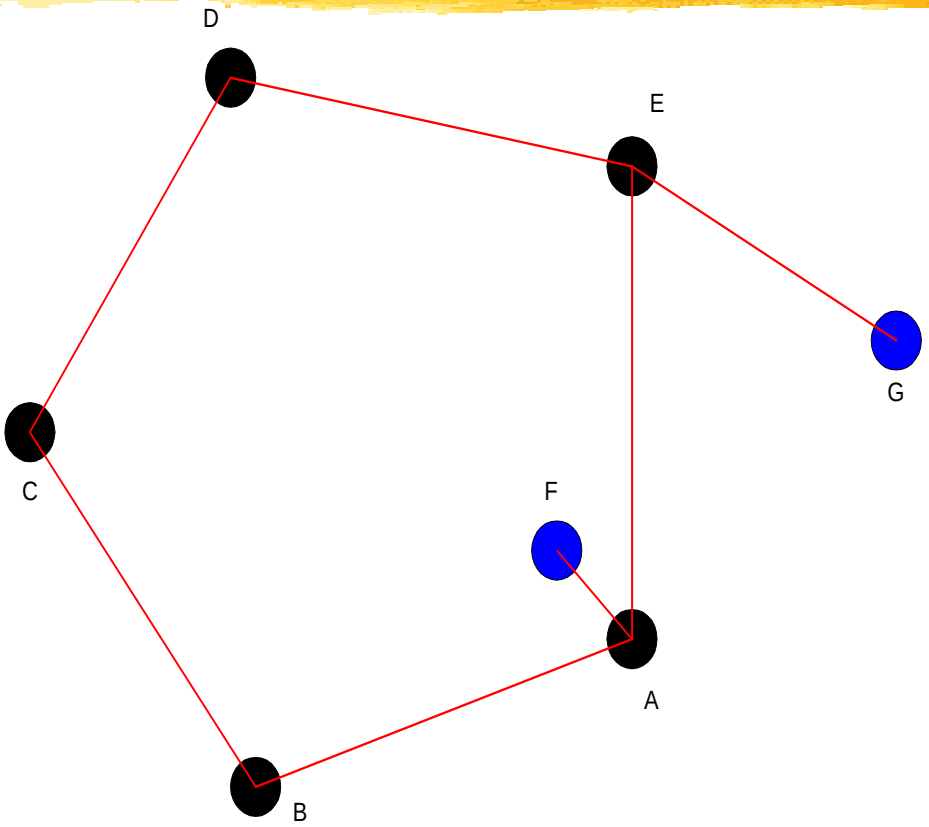
Scenario 2



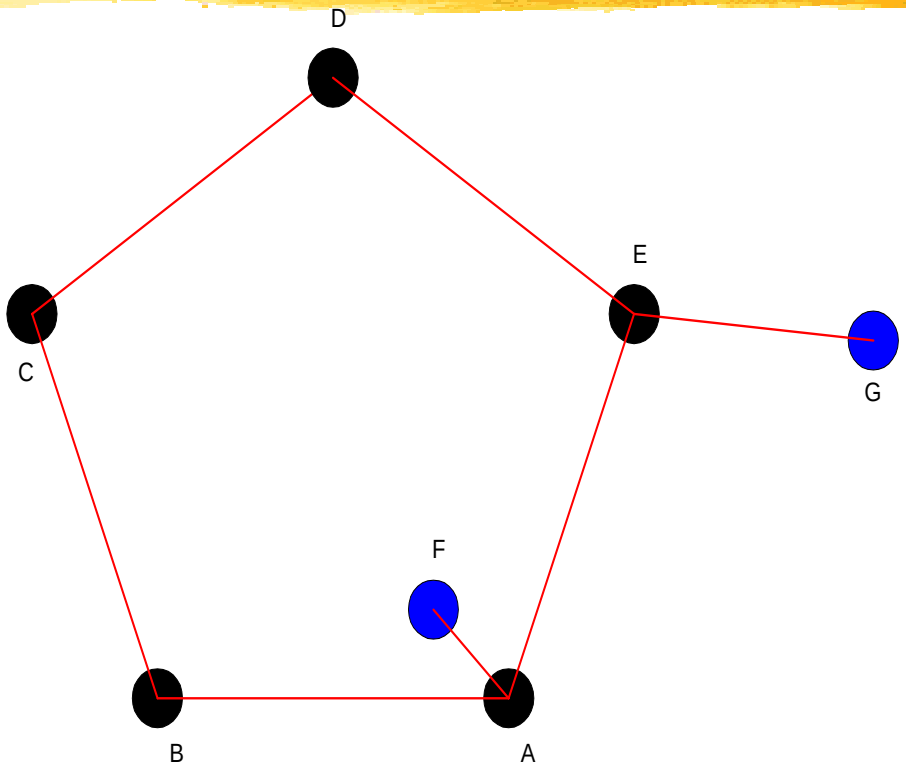
Scenario 2



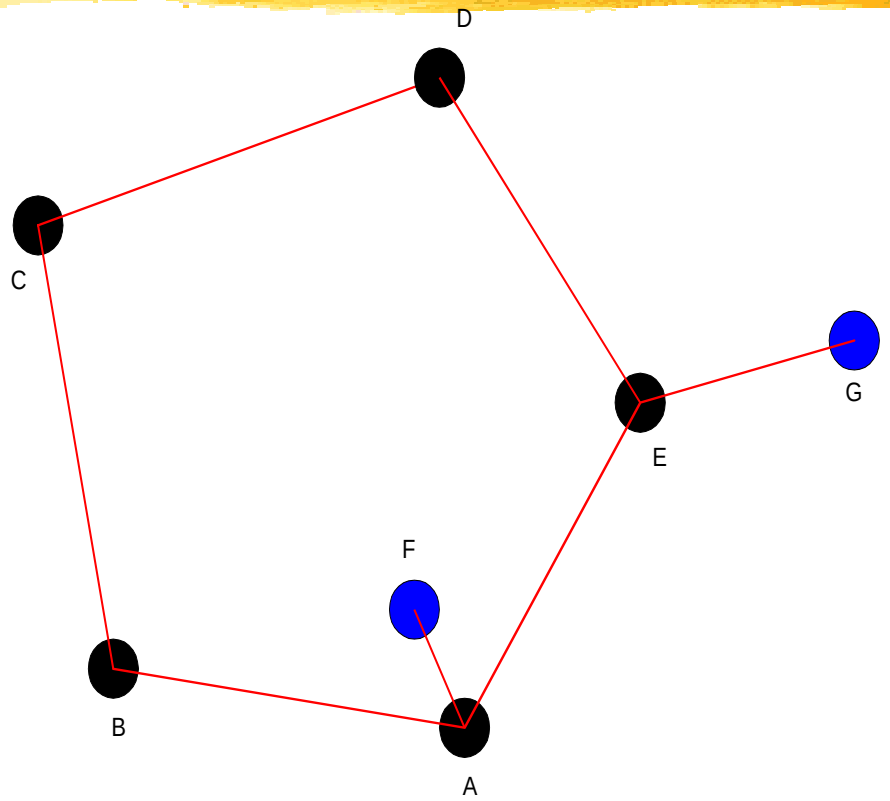
Scenario 2



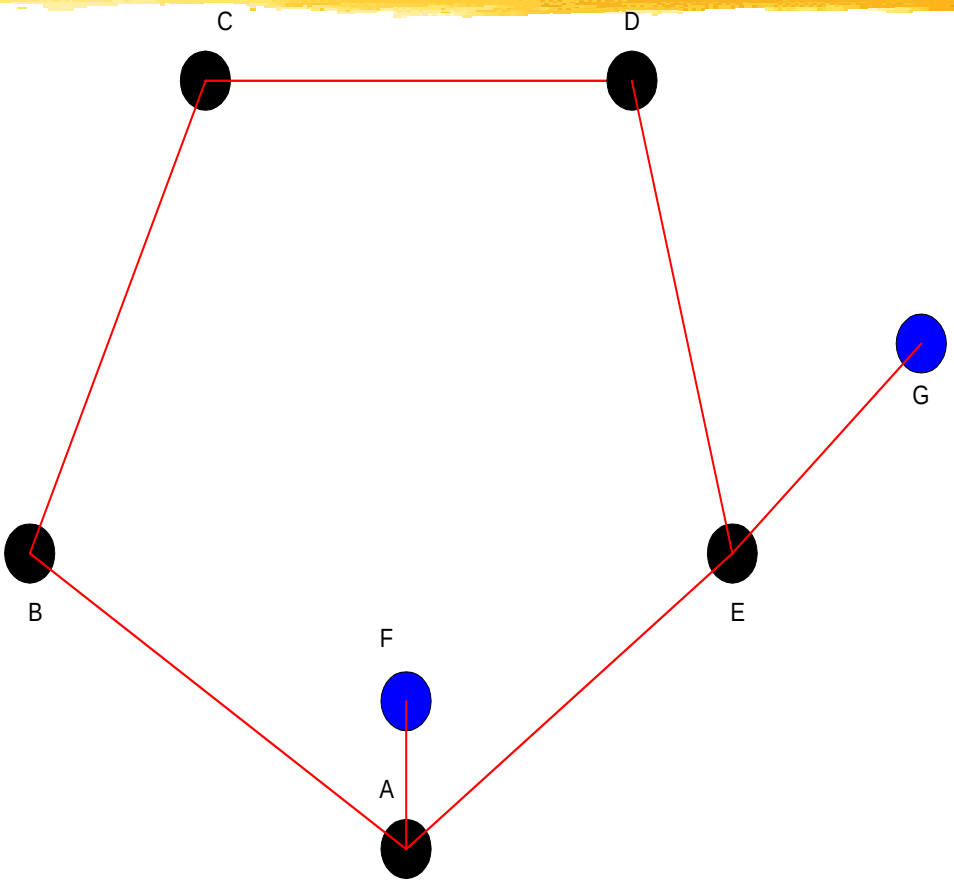
Scenario 2



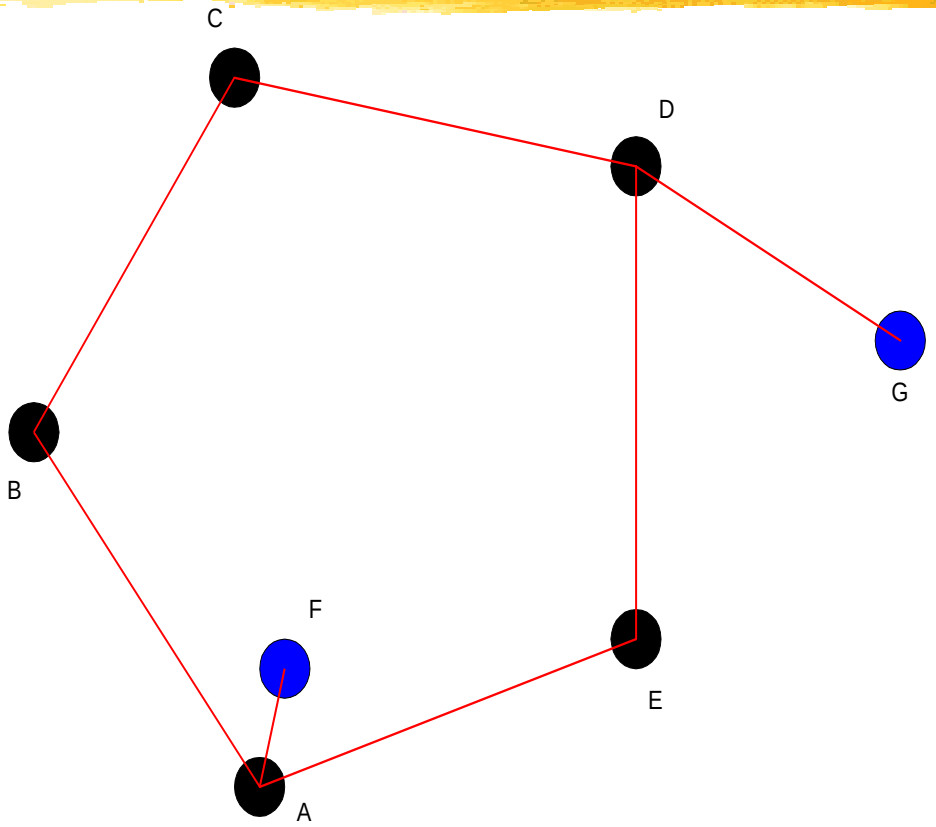
Scenario 2



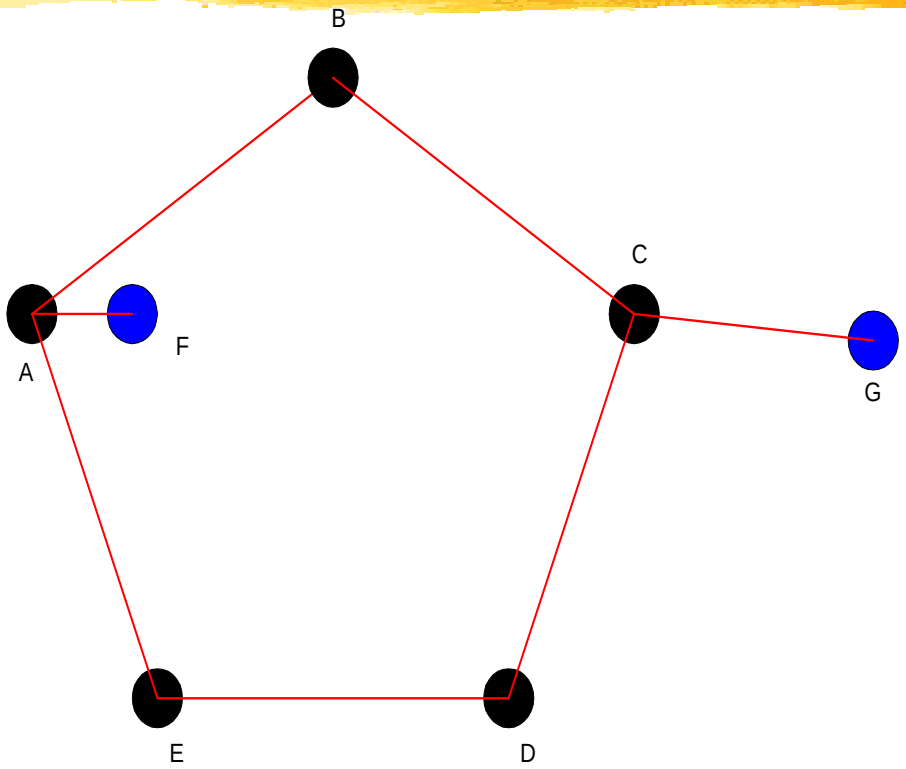
Scenario 2



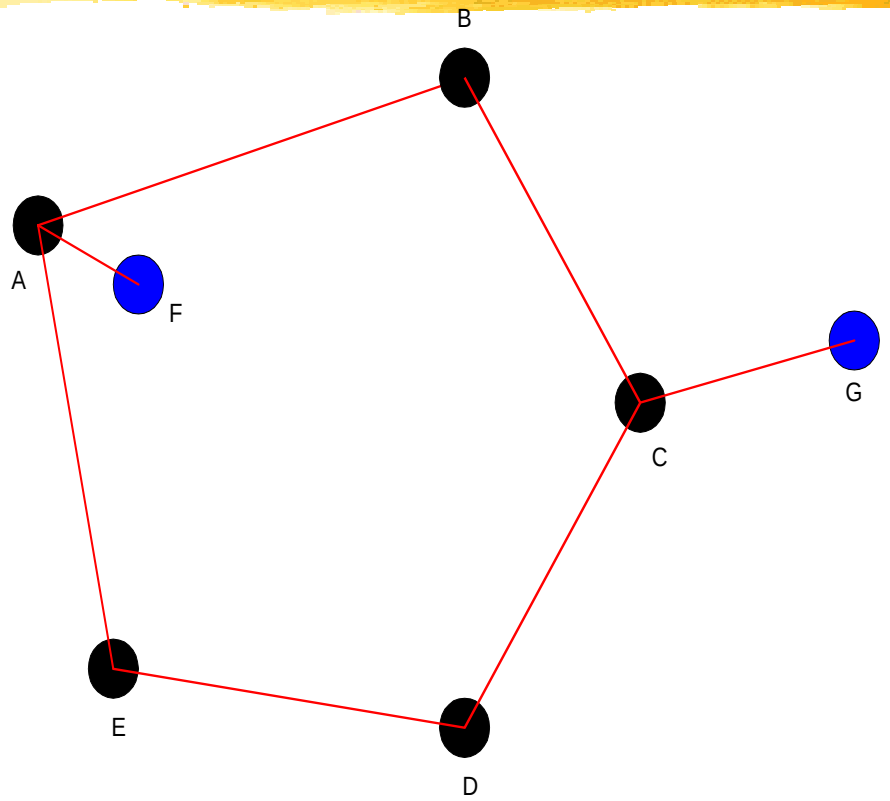
Scenario 2



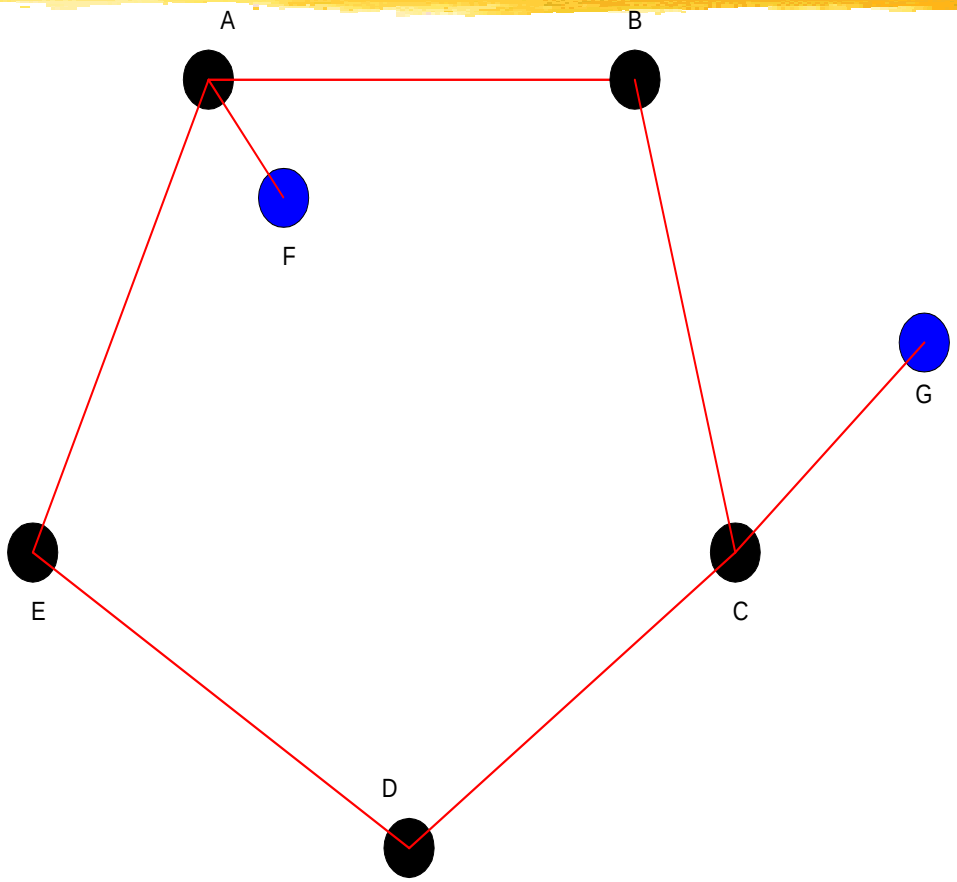
Scenario 2



Scenario 2

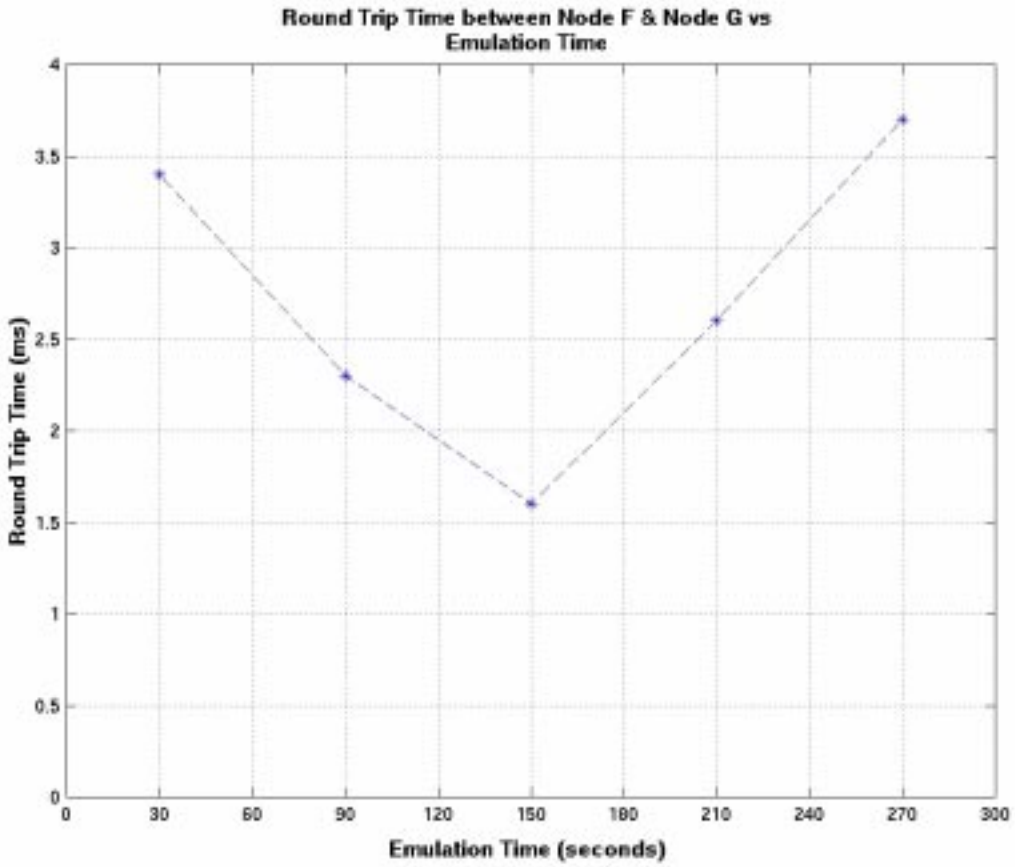


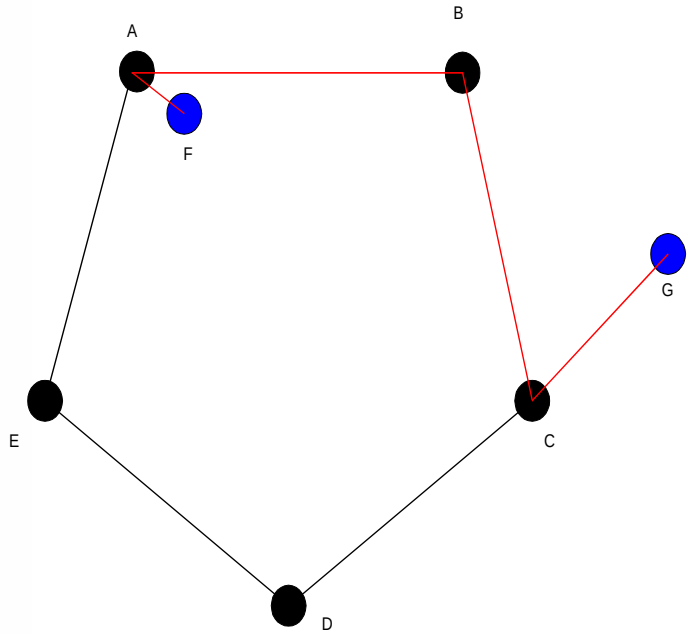
Scenario 2



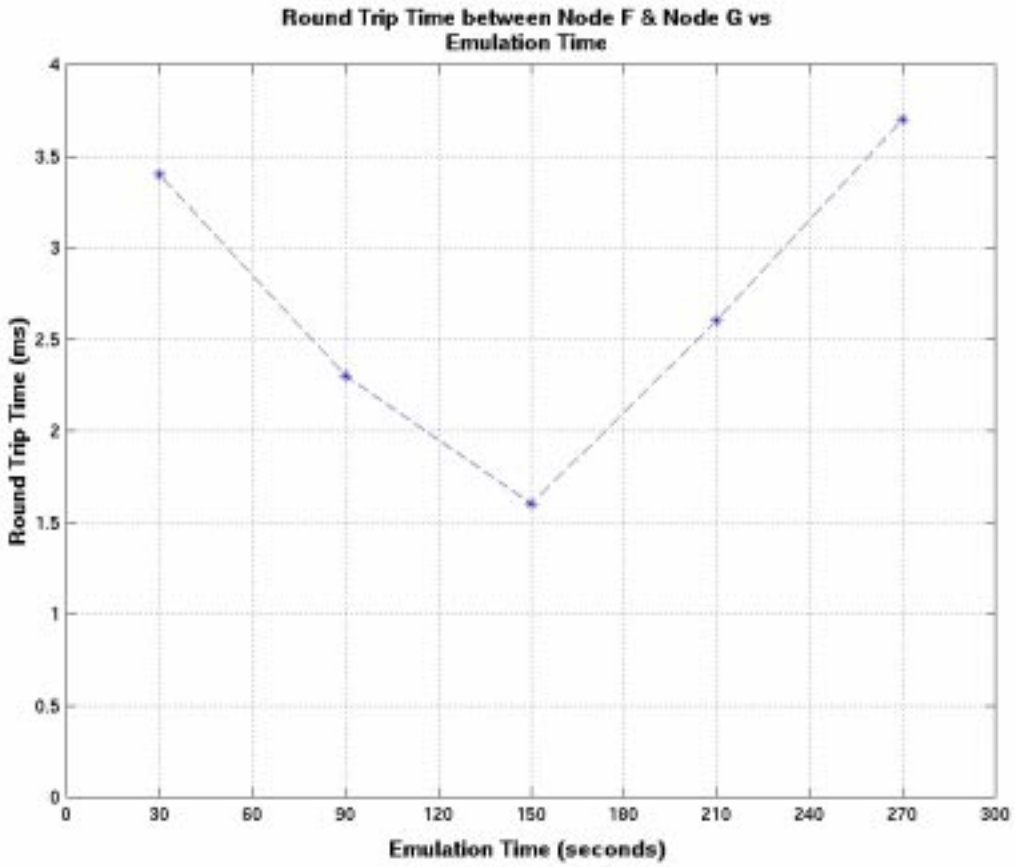
View Again

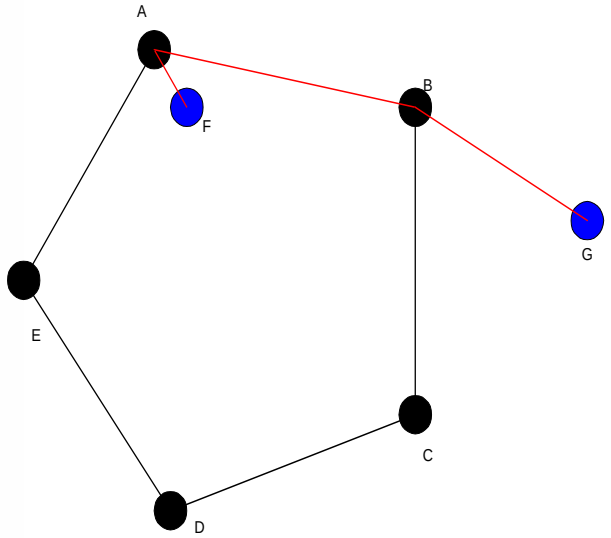
Results from Scenario 2



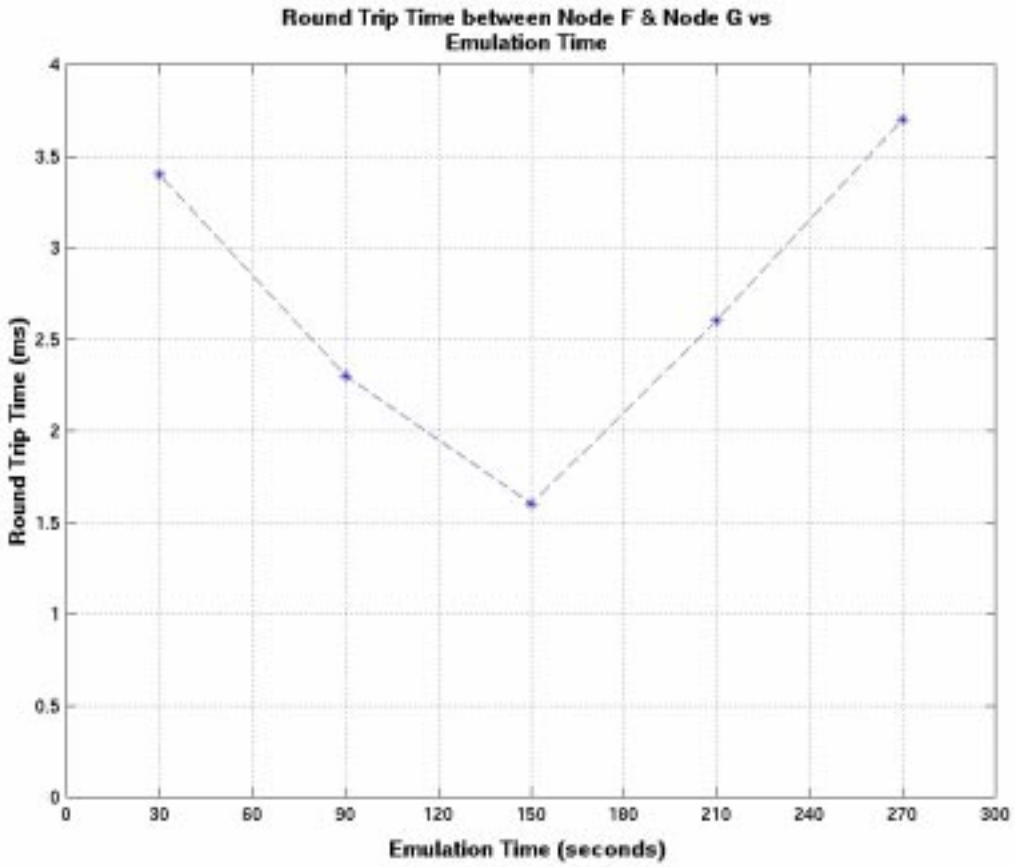


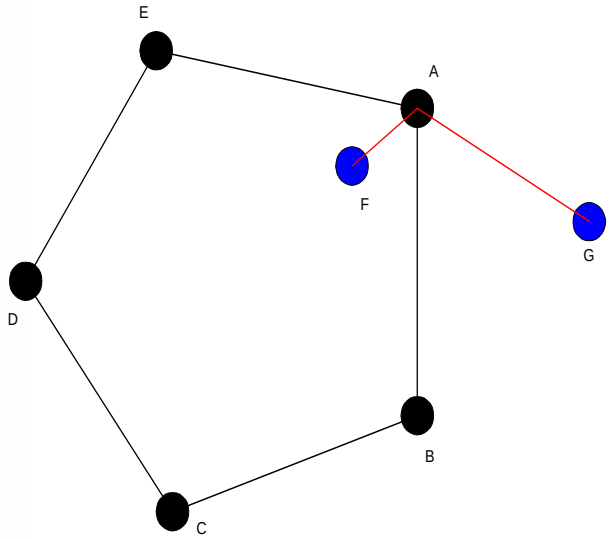
Results from Scenario 2



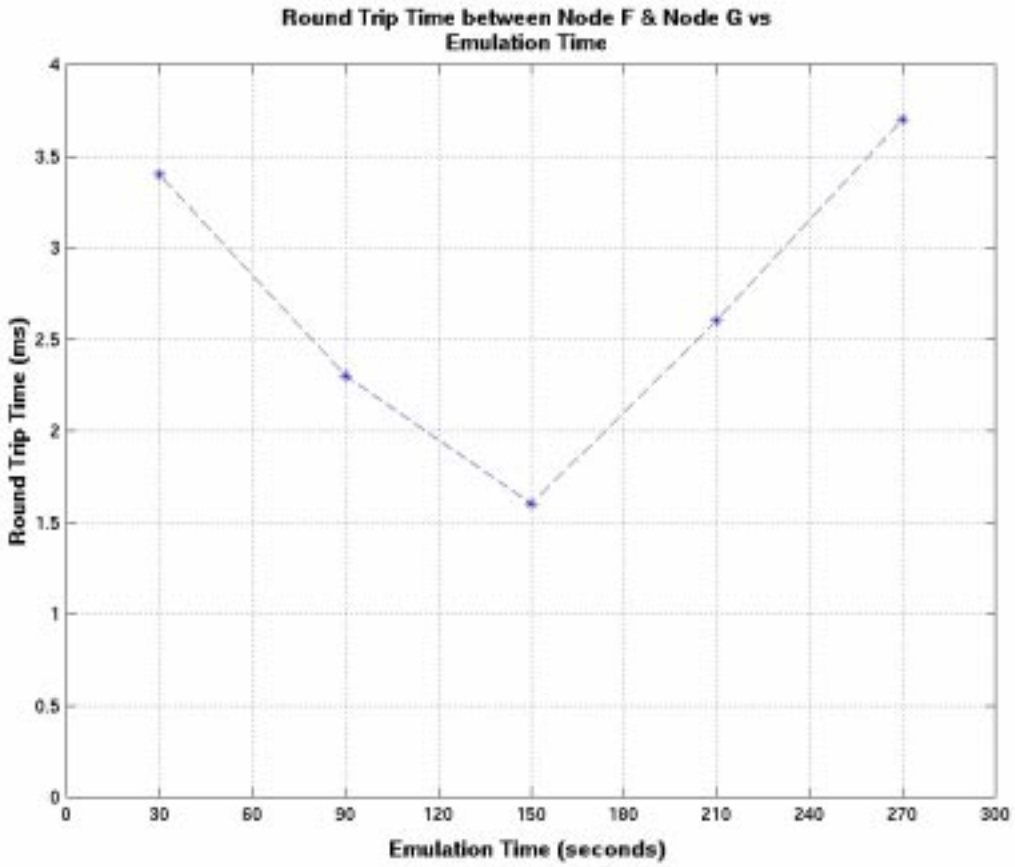


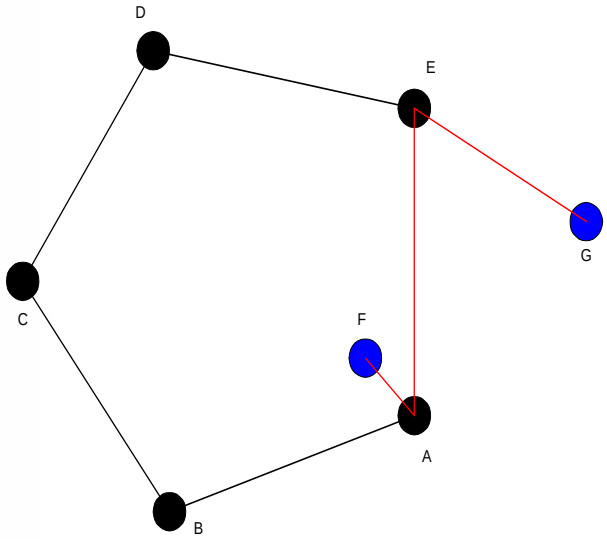
Results from Scenario 2



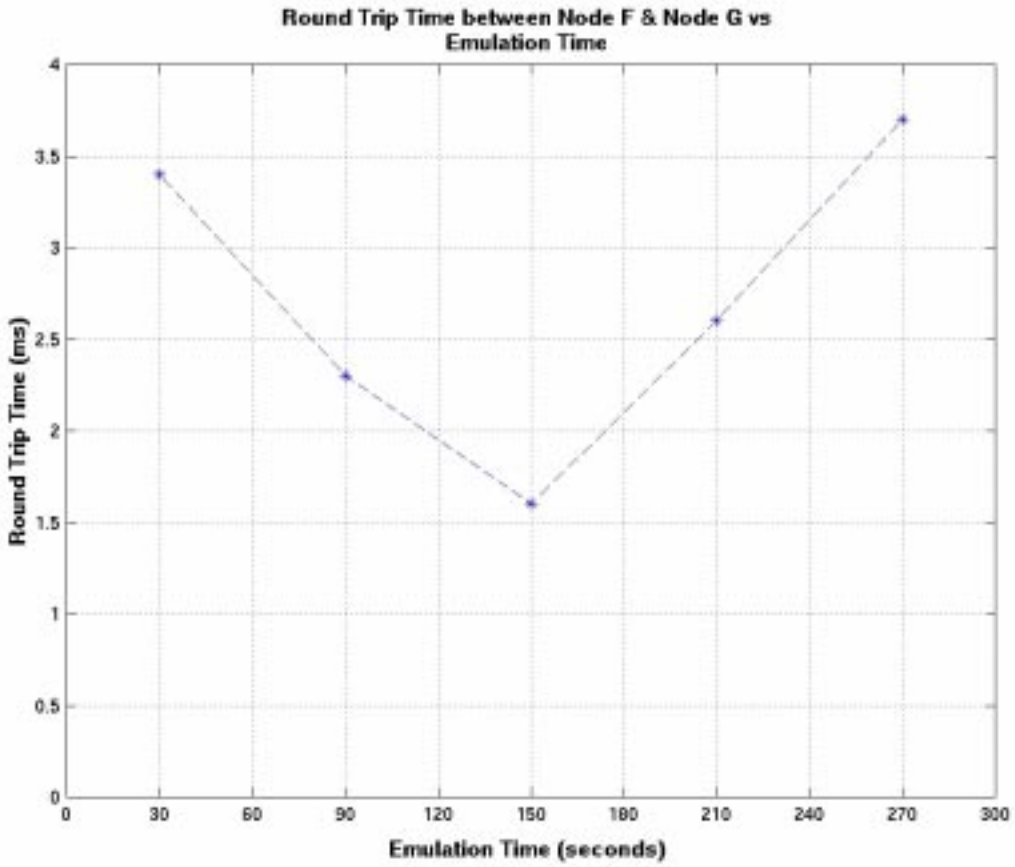


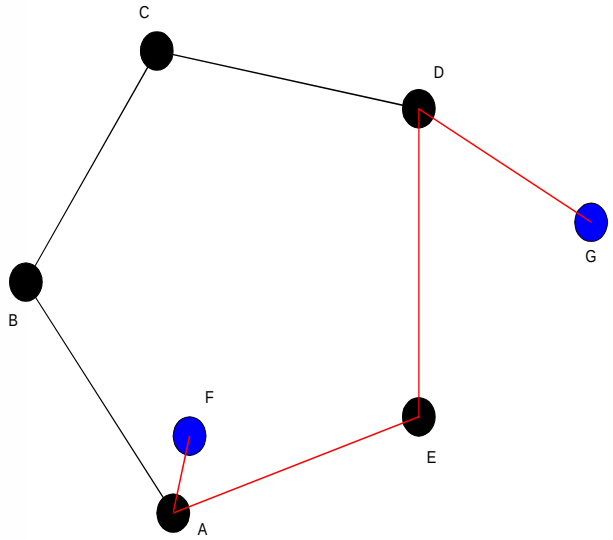
Results from Scenario 2





Results from Scenario 2



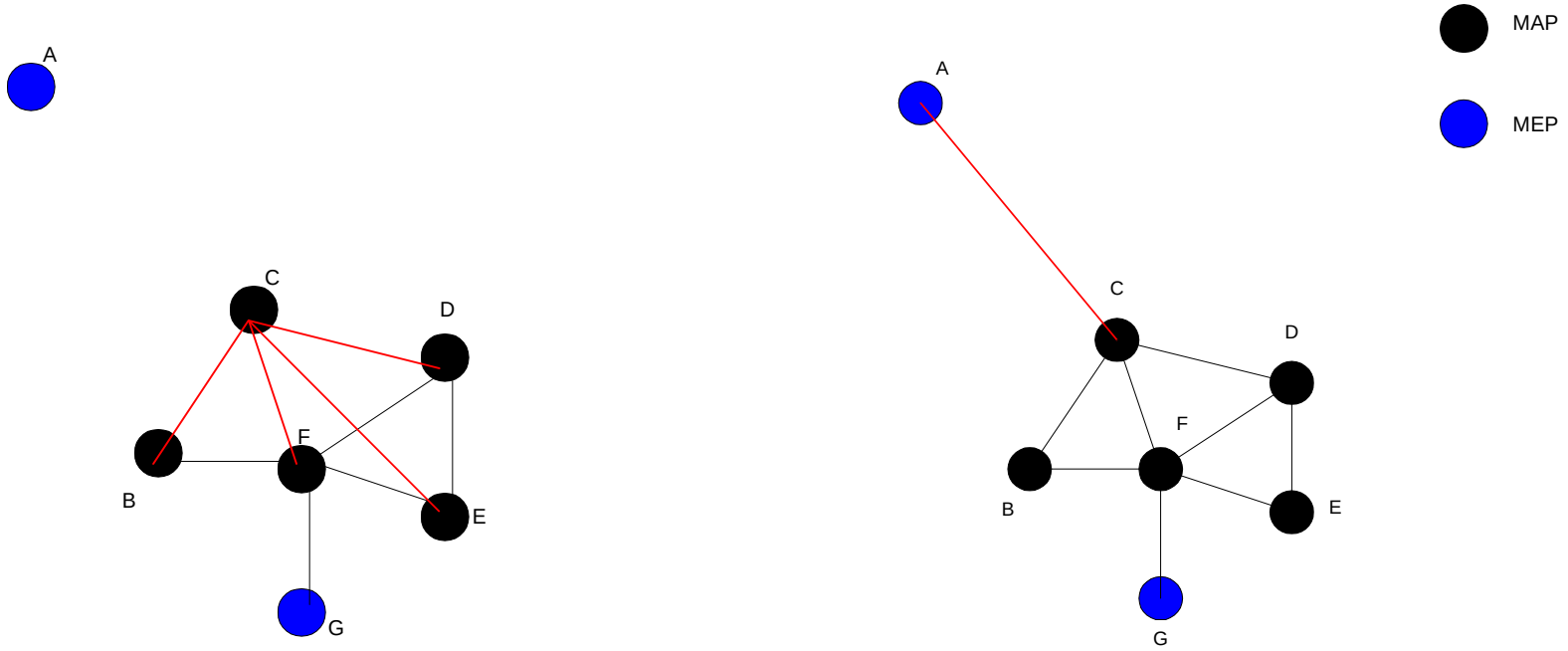


Conclusion

- Emulation Environment
  - Successfully implemented a repeatable, a controlled and a scalable emulation environment
- Scalability of the Network Controller
  - Before this work the network controller had been tested only for a 3-node scenario. We tested it for 7-node scenarios. Hence, the Network Controller does scale up
- IP vs ATM
  - For smaller packet size, the throughput achieved for end-to-end IP connectivity was greater than that for ATM. However, the difference was *not* appreciable
  - For larger packet size, the throughput achieved for end-to-end ATM connectivity was greater than that for IP. However, the difference was appreciable

Future Work

- Topology Algorithm



Future Work

- Wireless Channel Model
  - Current Emulation Environment does not include a model which emulates the channel characteristics
  - The model could be included as a layer in the VATM
  - Provide a handle to control the characteristics of the layer at run-time
- Performance Metrics for Larger Scenarios
  - Larger and more richer networks need to be tested under the emulation environment