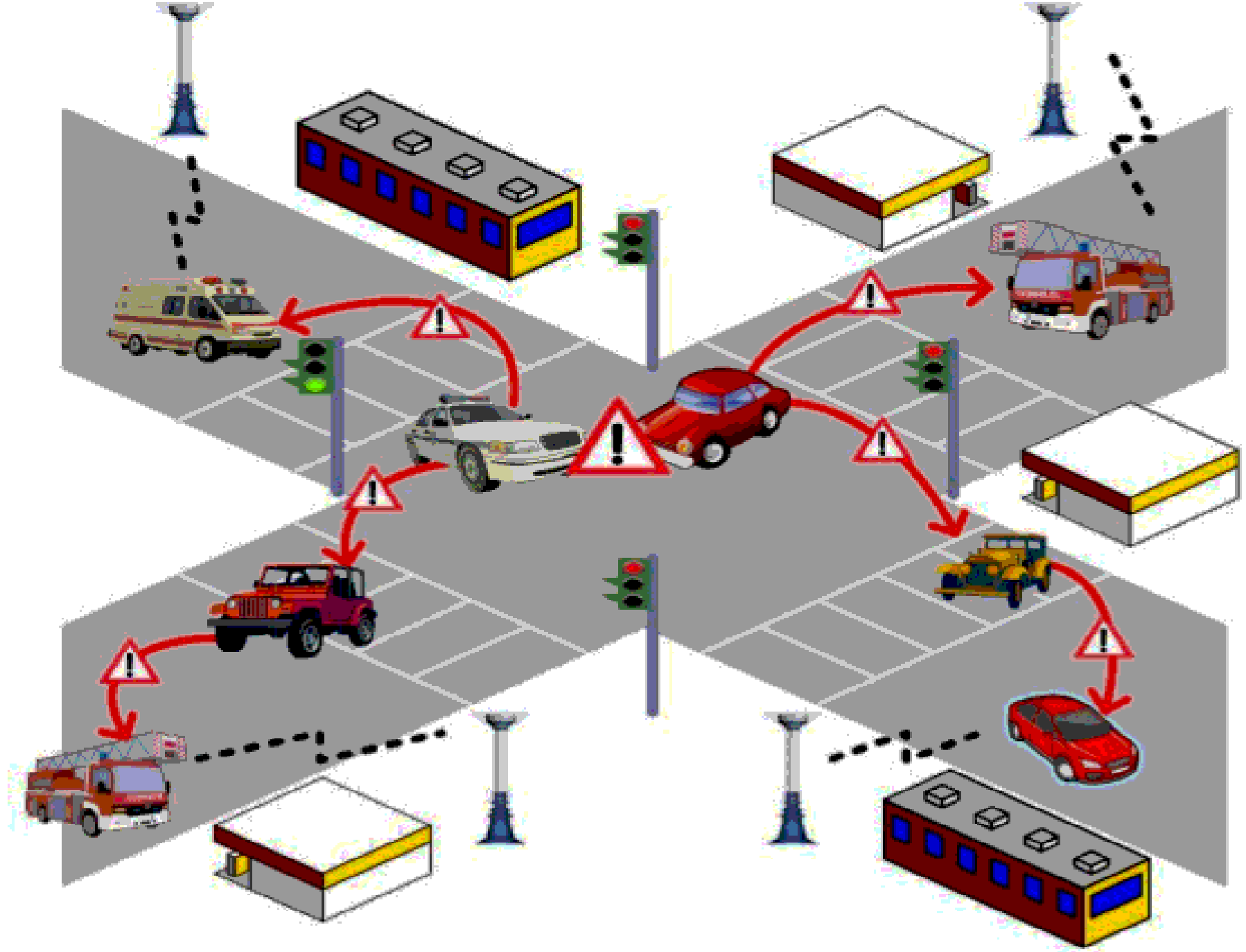
Efficient communication protocols for vehicular networks

Kevin XIAO

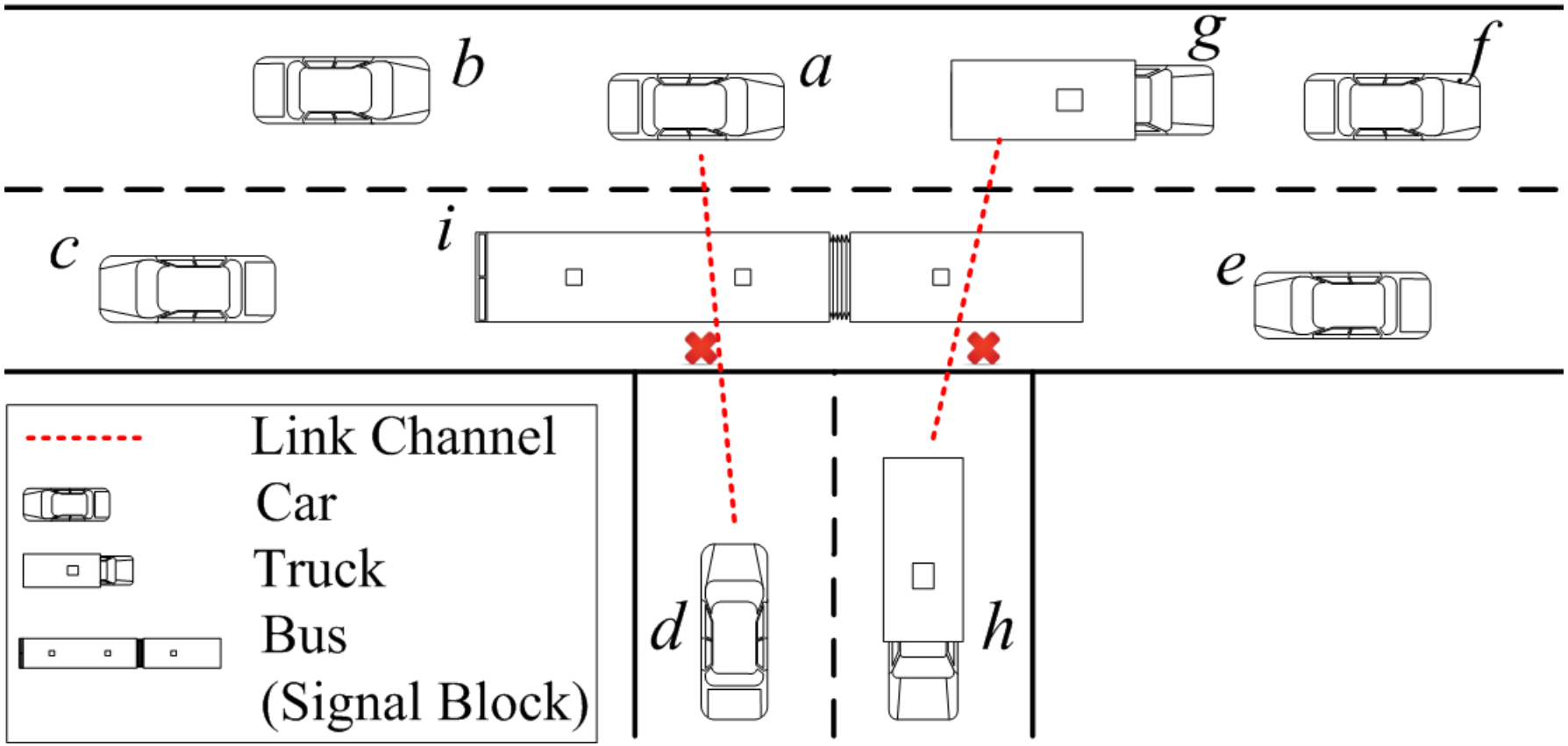
VANET

1 control channel

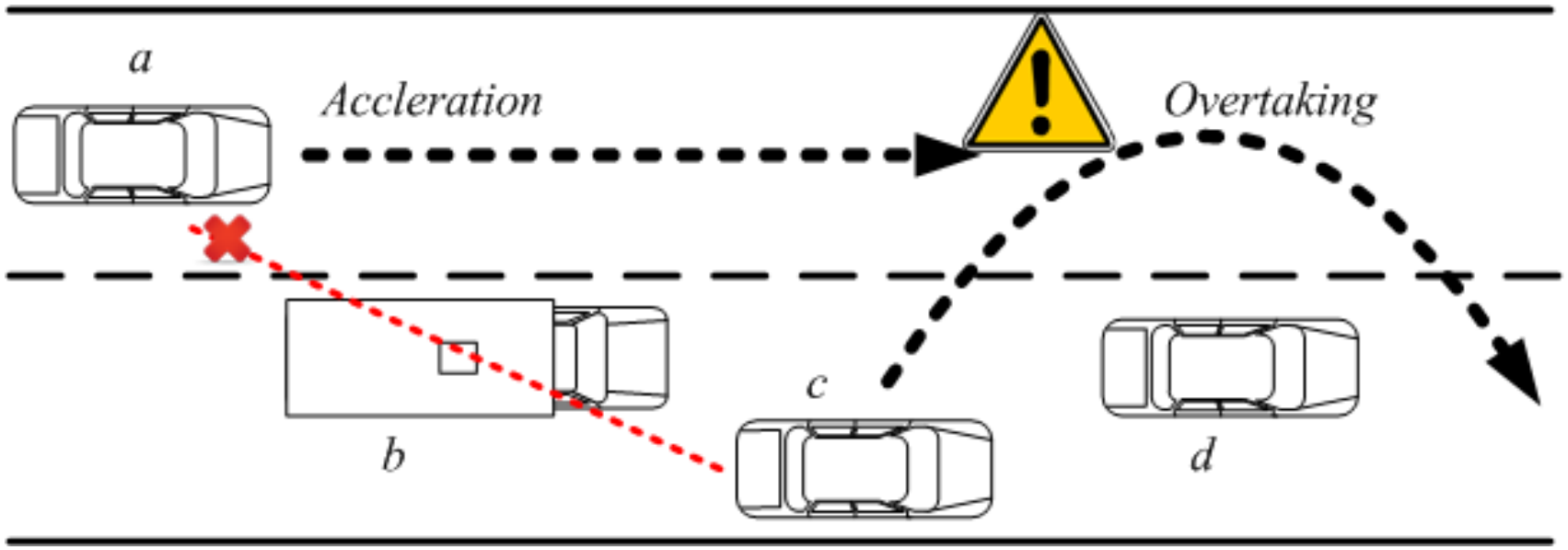
6 service channels



VANET



Reliable Broadcast Problem



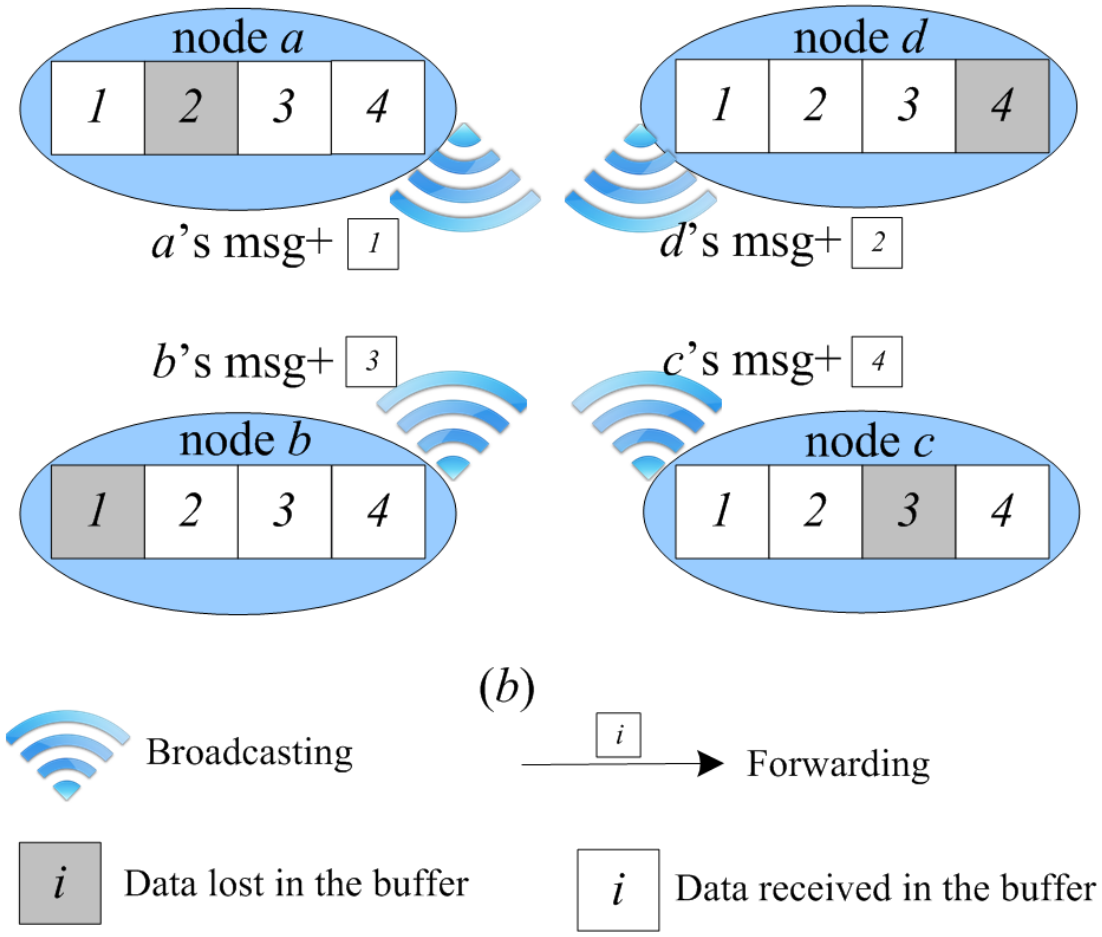
Existing Solutions:

1. **Retransmit Forwarding.**
   --- Can be blocked again with high probability;

2. **Flood Forwarding.**
   --- Prone to broadcast storms

3. **Cooperative Forwarding.**
   --- Call for a global view of the dynamic network to select the optima forwarder.

Cooperative Piggybacking



Greedy Piggybacking

1. Each node sends out a request of its lost messages.

2. Each node counts its effective neighbour’s requests.

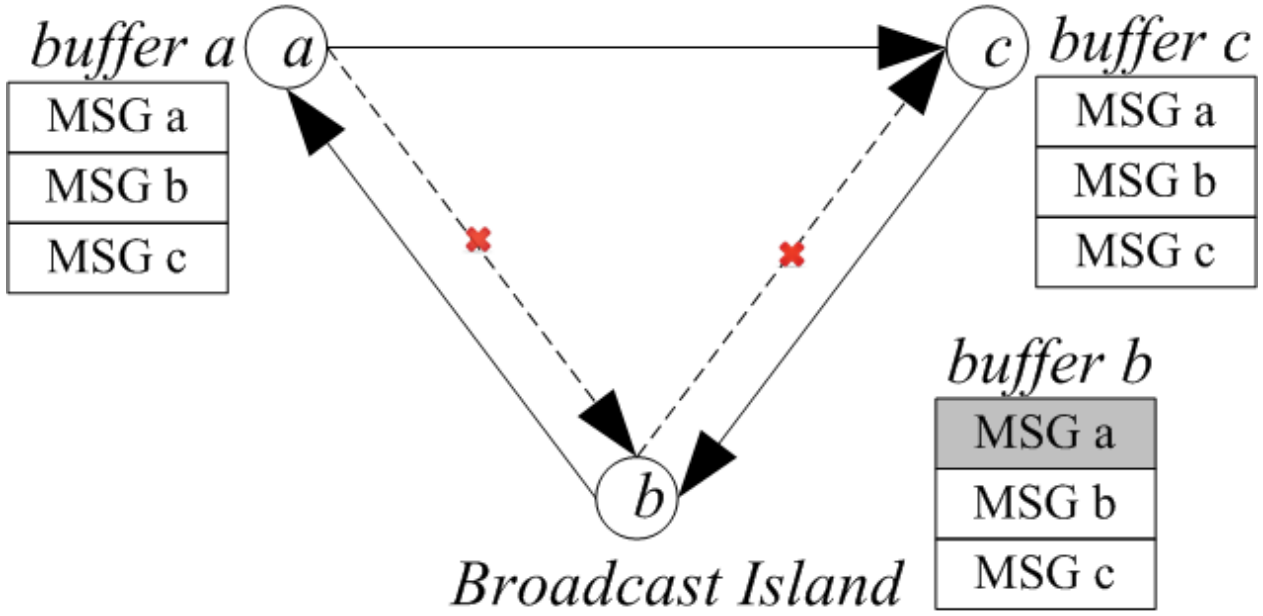
3. Each node selects the message which is requested by its neighbours with the max count .

4. Each node piggybacks the selected message.

5. After the piggybacking, each updates its local buffers and sends out a new request if available.

Broadcast Island

Node b broadcasts its request when it lost the message from node a. Node a can hear node b's request, but its piggybacking cannot be heard by node b; Node c's piggybacking can be heard by node b, but unfortunately node c cannot hear node b's request at all.



why GP cannot solve the broadcast island problem :

1. asymmetric channel links due to the antenna's imperfect disk propagation;

2. different broadcast coverage in the VANET;

3. each node makes its piggybacking too much greedy, which merely based on the local partial information .

Improved Greedy Piggybacking

1. Each node sends out a suggestion for all the other nodes based on its lost messages.

2. Each node collects its effective neighbour's suggestions.

3. Each node i updates its suggestion as:
   (1) counts the suggestions for node j;
   (2) selects the suggestion with the max count as the new suggestion;
   (3) piggybacks the message according to the new suggestion.

4. After the piggybacking, each updates its local buffers and generates a new suggestion.

Simulation

1. *NodeNum* is defined as the total number of nodes in the VANET, which changes from 4 to 10;

2. *LostNum* is defined as the number of lost data in each node, which changes from 1 to 4;

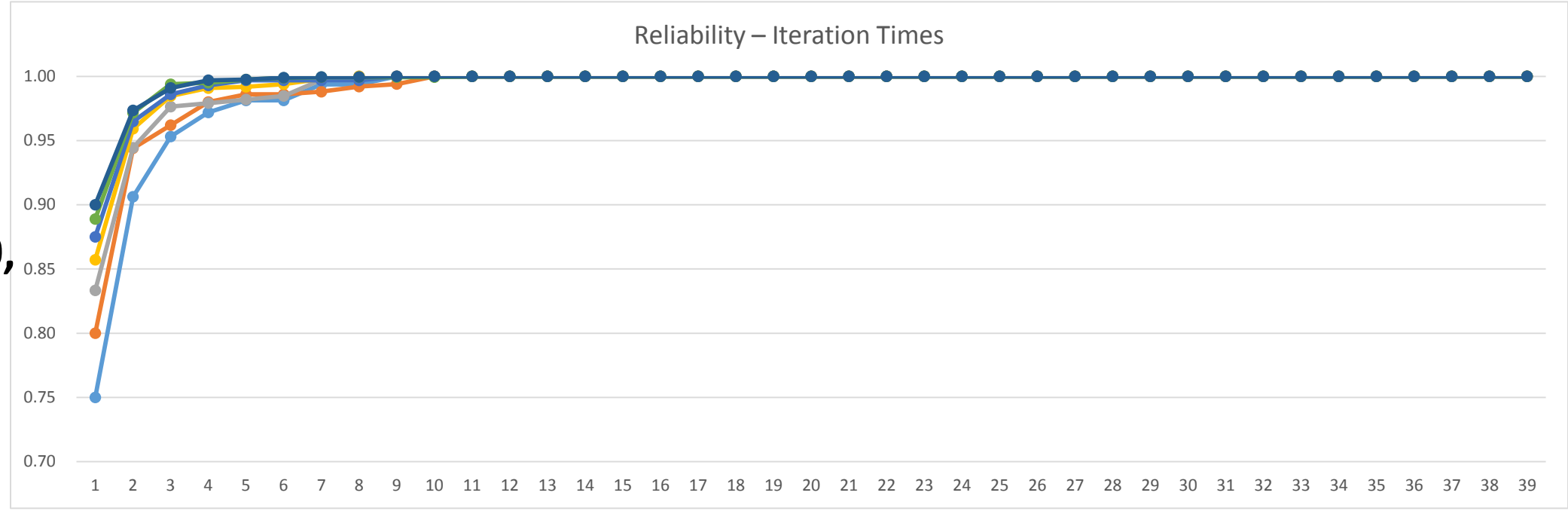
3. The buffer state for each node is generated randomly.

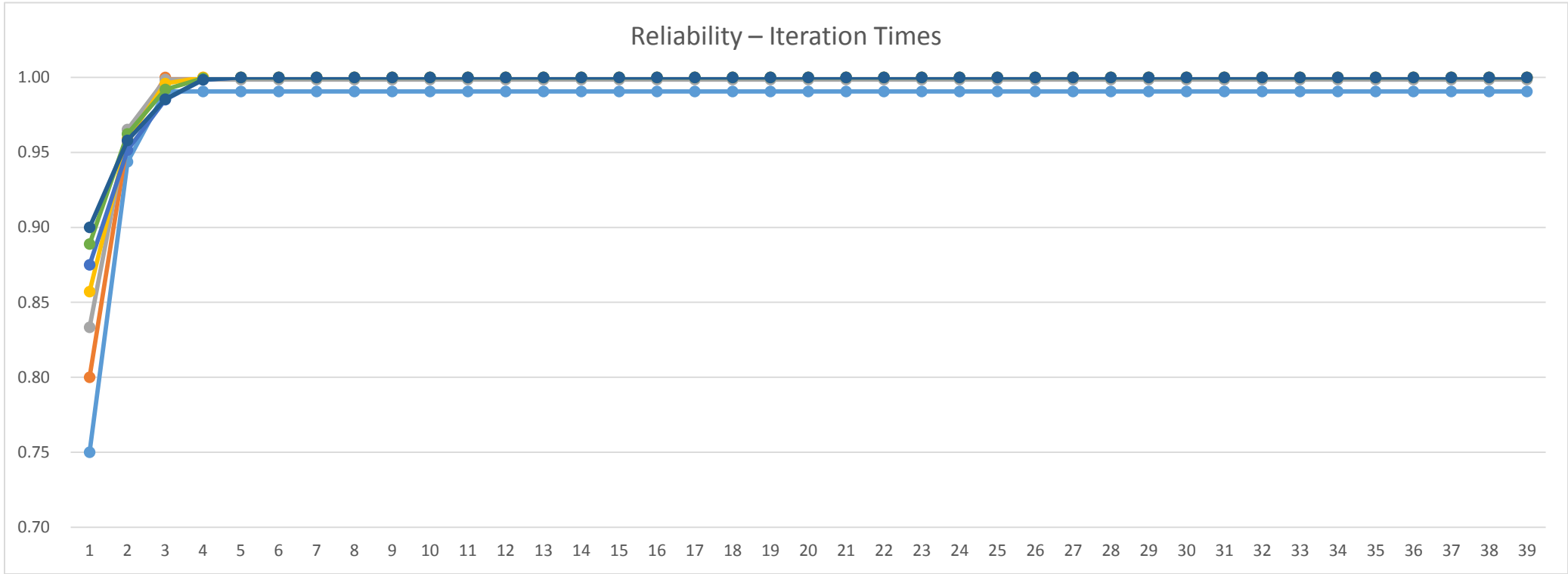
4. The max iteration time is set to 40.

5. Each simulation is repeated 20 times and then set the average as the final result.

NodeNum=4:10,

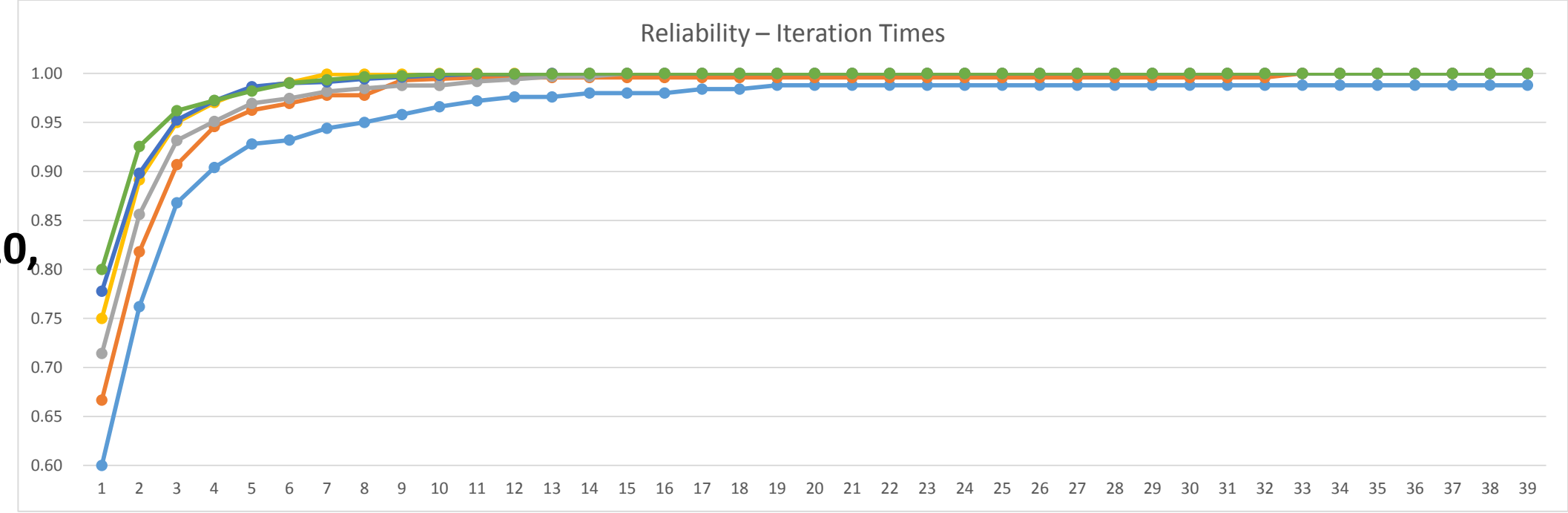
LostNum=1;

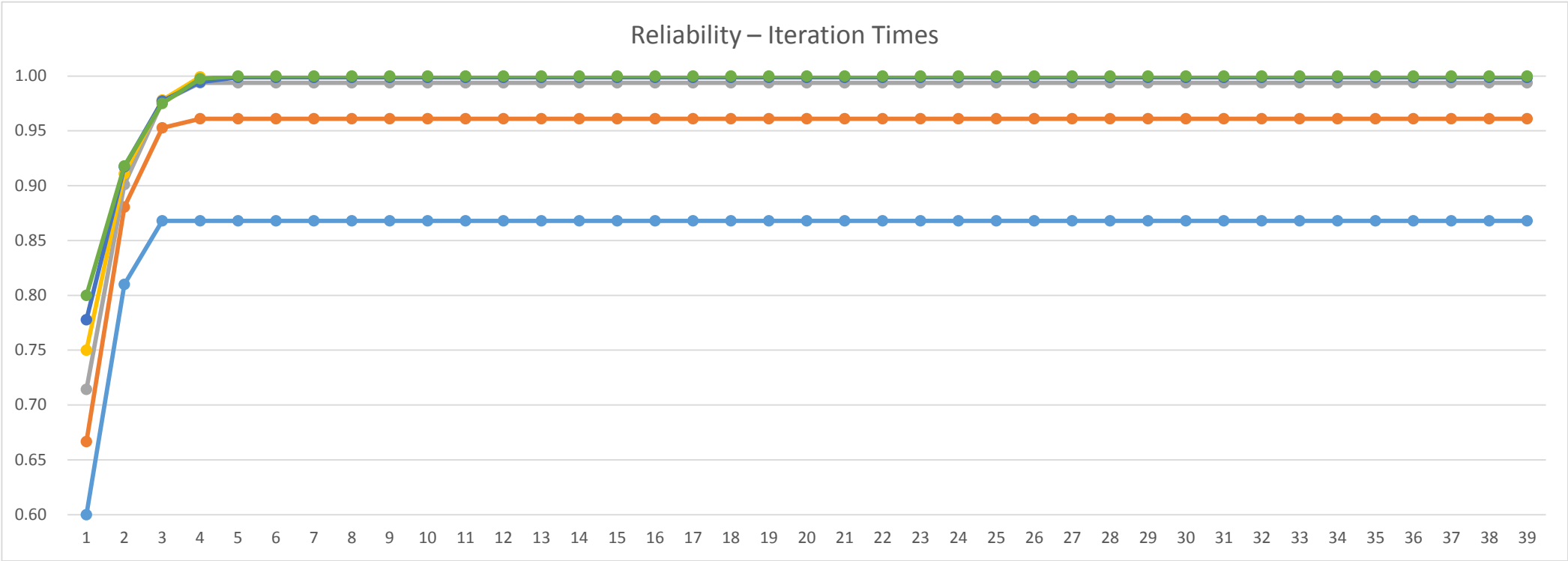




NodeNum=5:10,

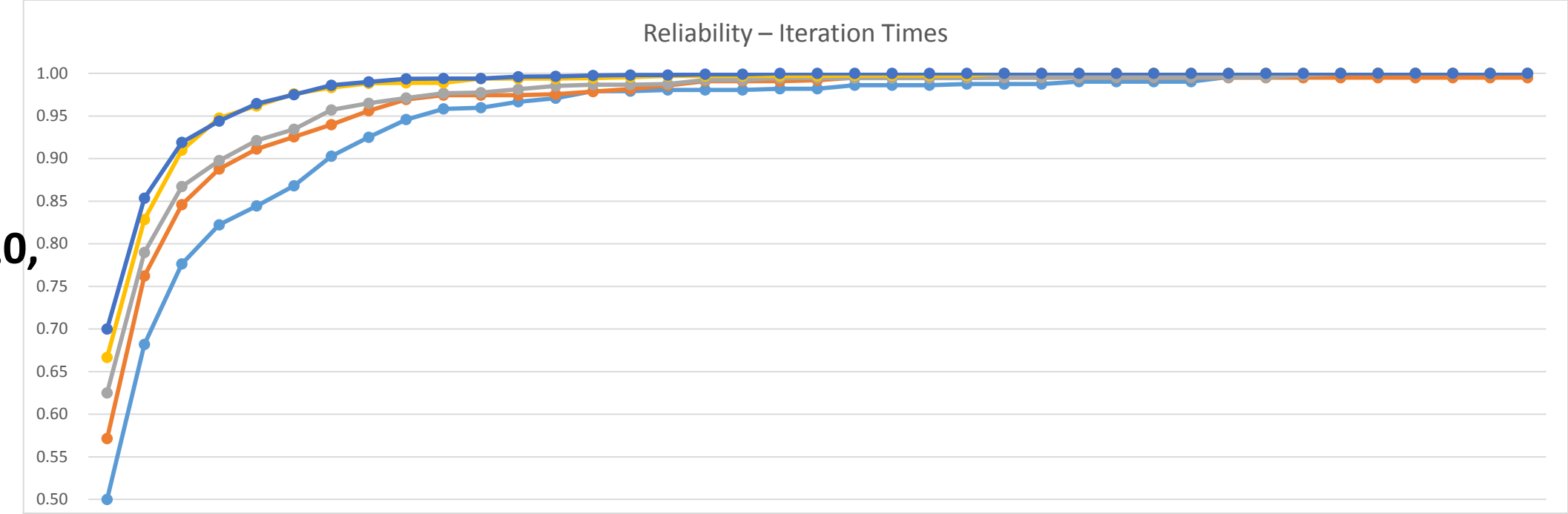
LostNum=2;

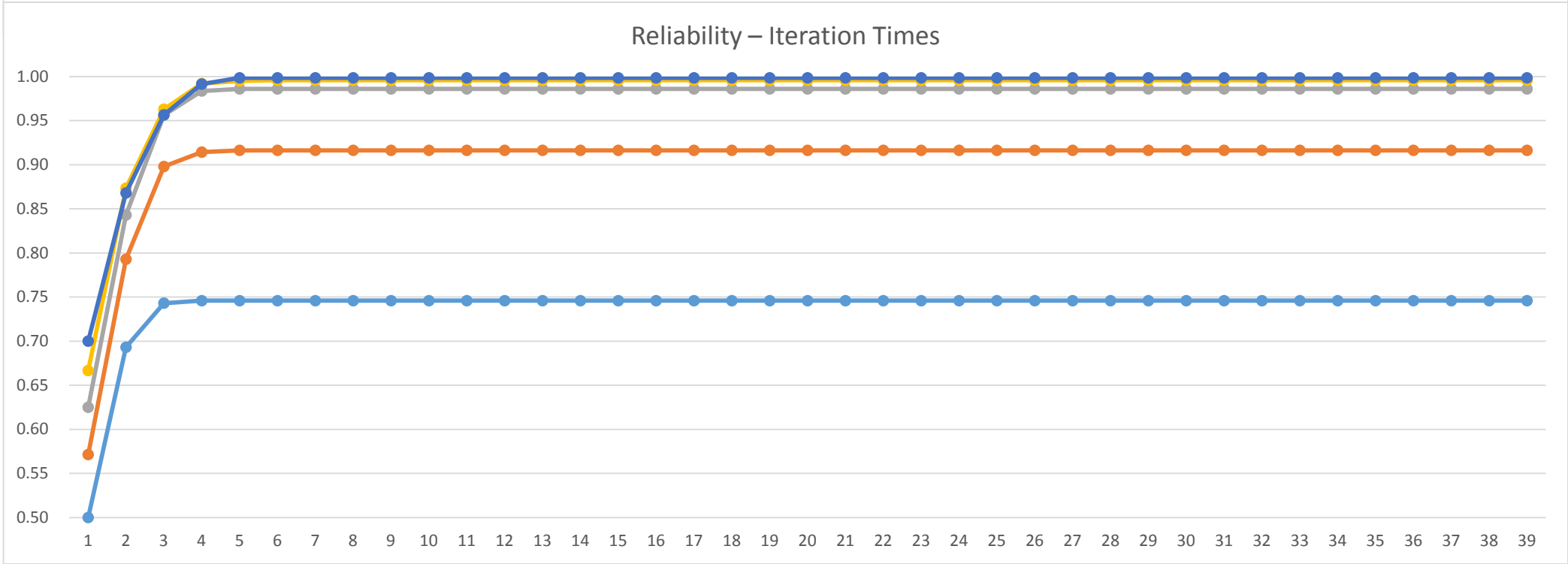




NodeNum=6:10,

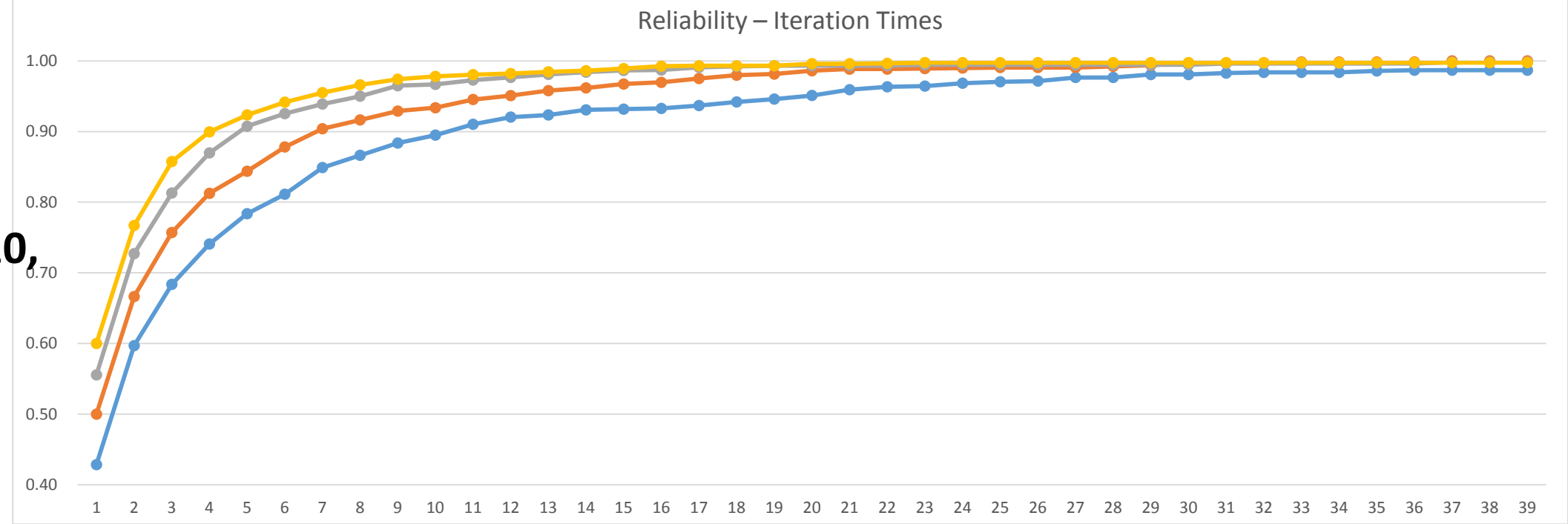
LostNum=3;

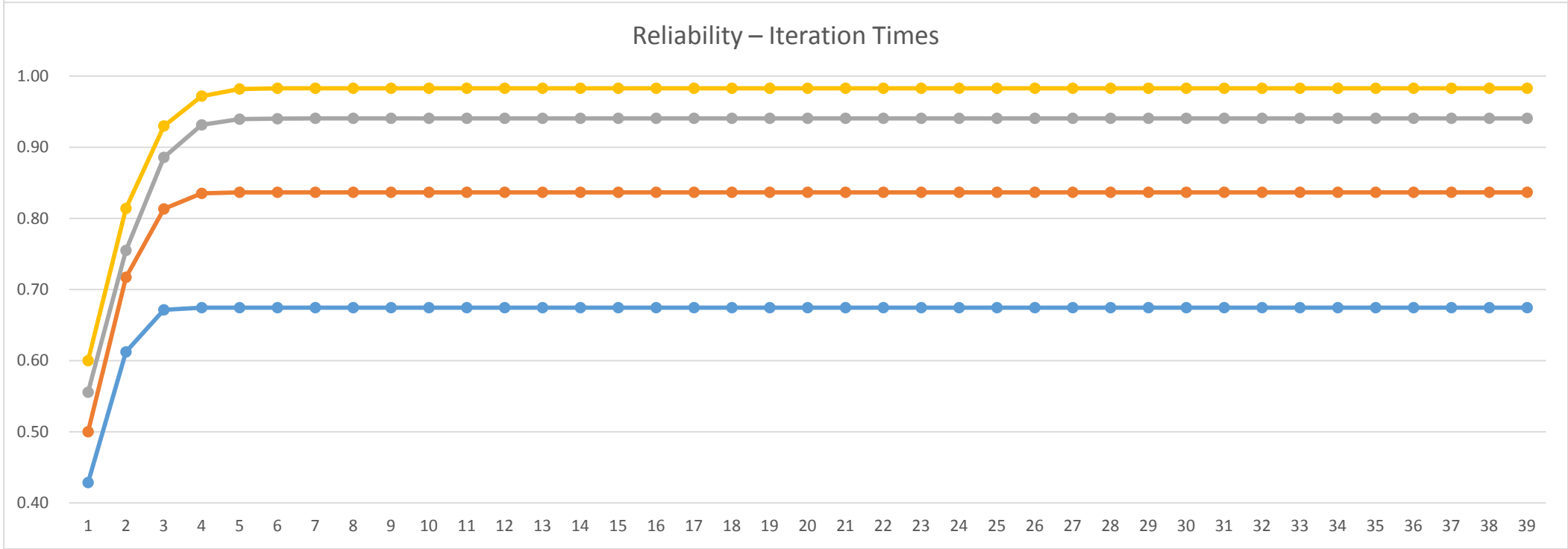




NodeNum=7:10,

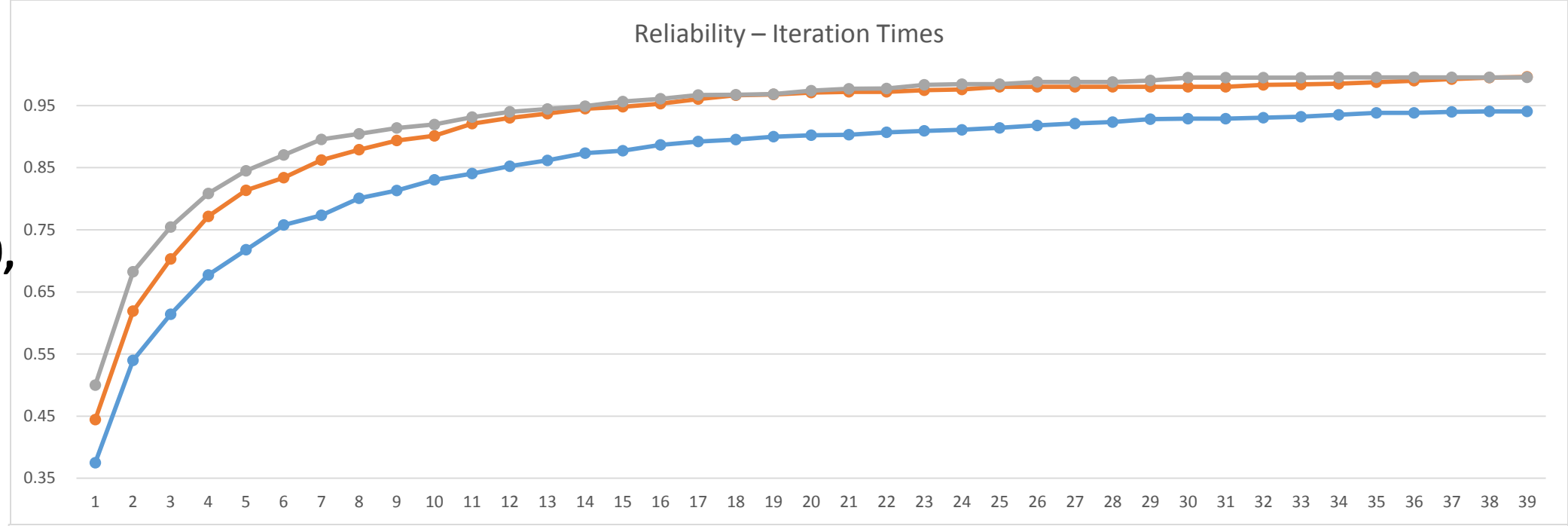
LostNum=4;

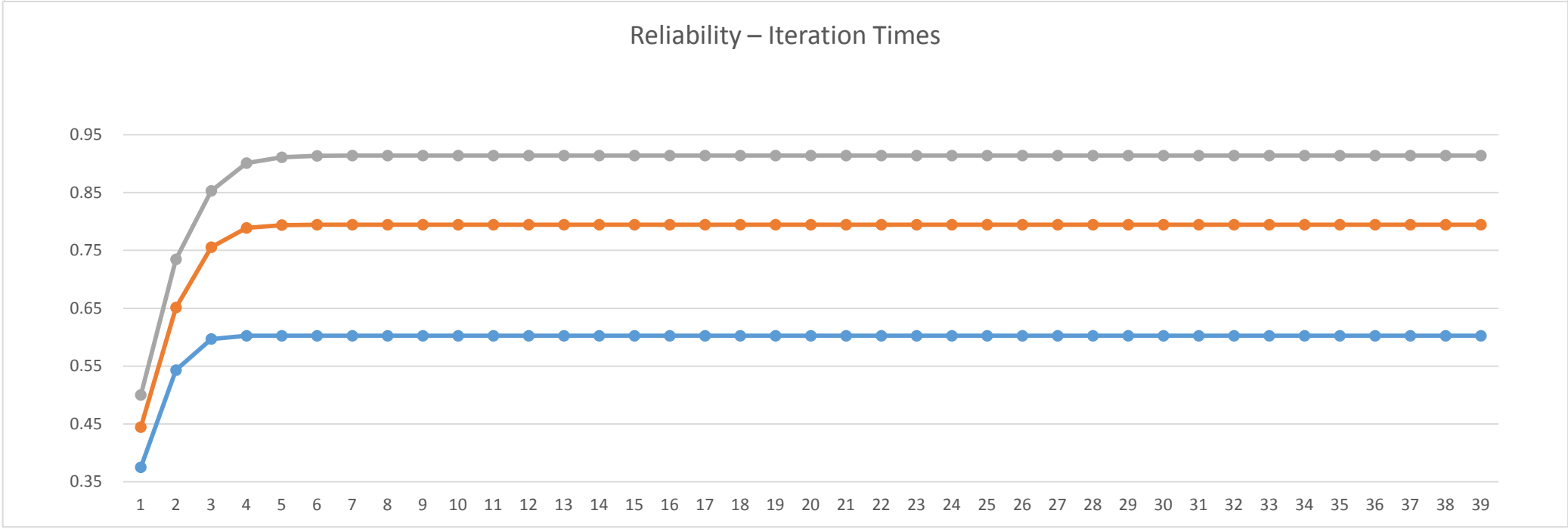




NodeNum=7:10,

LostNum=4;





Conclusion

1. When the ratio of (NodeNum/LostNum) is high, the two solutions have a similar performance (iteration times).

2. When the ratio of (NodeNum/LostNum) is low, the solution of Proposal with Suggestion has a better performance.