Enhanced PCF protocol for Control Area Networks

Goutam Das
Sathya Murthy

Global research Center, Bangalore, India
goutam.das@geind.ge.com
sathya.murthy@geind.ge.com

ABSTRACT
Most Sense and Control applications have unique traffic characteristics with small bursts of data and long intervals of silence. They also have unique requirements for small and predictable value of latency. These applications can exploit the advantages of wireless networks in terms of mobility, quick and easy installation etc. Current wireless networks do not meet the latency and traffic requirements efficiently as most of the current wireless network protocols are random access protocols. Even Point Coordination Function protocol defined for polling applications fail to meet the requirements of latency both in predictability and in low values needed, as shown in our simulation. This paper defines an enhanced PCF protocol suitable for use in such applications creating a good robust Wireless Control Area Network.

1.INTRODUCTION
Many Industrial applications use Control Area networks. These will be in all cases where there is a central controller controlling operations of multiple entities. They can extend from
- Manufacturing (Central controller controlling multiple machines),
- Monitoring and control applications (Monitoring and Control of Turbines, Aircraft Engines, Medical instruments, etc, where a central controller monitors performance of the equipment using sensors and controls them using actuators)
- Environment control equipment (where sensors report the status at regular intervals to the central controller, which takes decision based on inputs) etc.
In all these applications, the network topology will be either Star or Tree networks and the traffic is almost deterministic and repetitive in nature. The general architecture consists of one central controller, which polls the nodes in a round robin fashion according to the requirements. Individual node respond whenever polled by the central controller. The time sequence used by the controller is shown in figure 1:

![Control Frame](image)

Figure 1. Control Frame.

In control network the controller informs all the nodes before start polling and individual nodes (Sensors, monitoring elements etc.) prepare data to indicate the status of the system health for that particular instant. The figure 1 shows the burst arrival period in all the nodes invoked by the “start of cycle” call from the controller. In the next time slot T1 the controller start polling the nodes according to the polling table it maintains. This is the data transfer period of the communication cycle, and the channel is occupied heavily by the bursts present in all the nodes. The nature of this traffic is summarized below
- The traffic is repetitive in nature, which means information flow from the nodes to the controller (both the number of packets sent and their sequence of transmission) does not vary much from cycle to cycle.
- The traffic is highly bursty in nature.
- For all the nodes, the burst comes more or less at the same time instant. Hence, the bursts in individual nodes are almost synchronized.
- The entire cycle period depends upon the time controller takes to process data and send command, which is again random in nature. Therefore, the burst arrival time also maintains randomness.
To handle this kind of traffic the challenge here is to optimally poll the nodes and efficiently
share the bandwidth. In short the requirements are:

- The data transfer delay should be minimized
- The delay for all the nodes should be as deterministic as possible

Recently wireless solution for control has become highly demanding. The wireless technology is having its own advantage of mobility, low installation and maintenance cost. But majority of the wireless protocols uses random access mechanism, which does not suits well for control requirements. Among all the existing wireless protocols, 802.11b PCF uses polling technique, which makes it favorable for control application. In the next section, we will briefly describe the PCF protocol and its disadvantage. In section 3 we will introduce a new protocol to enhance PCF for control traffic requirements.

2. POINT COORDINATION FUNCTION

PCF is a recommendation in 802.11 b and the basic access method is polling [1,2]. PCF architecture has a central controller called Point coordinator to supervise the media access control (MAC) operation. Typically, PCF starts after a command from the Point coordinator (Beacon) in the system to all nodes and remains in polling state until a command CF End is issued by the point coordinator.

![PCF Timing Diagrams](image)

The point Coordinator looks for activity in channel and if channel is free for a period of PIFS, transmit the poll to the next node in the polling list. Two implementations are possible [3]. A node in the polling list can be polled repeatedly until its queue becomes empty and null data packets are sent for a predefined number of consecutive polls. Alternatively, each node is polled sequentially. In the second implementation, the node with null data for predefined number of consecutive polls will be dropped from the next poll and only other nodes are polled in the Contention free period. After being dropped from the polling table, In case if any data is received by a node, node will have to wait till the next cycle for contention free period (next beacon).

At the end of Contention free period, the point coordinator will send a CF End broadcast. The point coordinator sends out a beacon again after the beacon repetition time.

2.1. Problem with PCF protocol:

PCF ensures no collisions and makes data go across the network in the least possible time. However, it does not address the time data waits at the nodes before polled. An example can make the problem clearer.

![Additional delay due to unsynchronized data arrival and beacon issuing](image)

The figure 3 describes the case for 2 nodes having bursty traffic. It is assumed that the controller will poll for 2 successive times before dropping a node from its polling list. The rest of the diagram shows the PCF operation mode. Note that the operation of PCF (set of Beacon, Poll, and CF End) has timing relations within itself, but has no relation to the timing of arrival of data in the nodes. As the two events are uncorrelated and no timing relationship can be exercised over them, the positioning of the Data received can be anywhere from previous beacon to the current beacon and there is every possibility of burst coming after the CF end has been issued.
This problem of lack of synchronization between beacon arrival and burst arrival has not been addressed in PCF protocol description. However, one can try to adjust PCF parameters to resolve this problem. Let us look at a few possibilities.

**Very short PCF cycles:** This causes delays as the total data available at the nodes may not be completely transferred in one CF cycle and additional CF cycle overheads cause unnecessary delays.

**Very long PCF cycle:** Here as no node is dropped from the polling list, it causes added delays due to unnecessary polling. Additionally if the data arrival cycle and beacon intervals do not match, successive cycles will have differing delays. For example, if the beacon interval is 3 ms and data arrival cycle is 10 ms waiting time in successive cycles would be different as shown in the figure below.

![Fig 4: Effect of mismatch between beacon interval and data arrival period](image)

3. **ENHANCED PCF**

The new method employs the principle of creation of timing relationship between the Beacon and the data arrival. This can be achieved by the higher layer (control application) in the controller. The application layer in the controller knows when exactly the “start of frame” happens. It can always invoke a trigger to its MAC to indicate the burst arrivals in all the sensor or controlled nodes. MAC then can issue beacon immediately to synchronize the burst arrival in the nodes and the beacon arrival in the shared media. But this adds dependency among the layers and protocol implementation will be a costly and complicated task.

Instead, we propose to enhance the MAC functionality to create the relationship between the two independent events: beacon arrival and data arrival. A special packet called WTS (Waiting to Send) is used to create this relationship.

The typical steps for communication in the proposed method are as follows:

1. The Point Coordinator sends a beacon but does not issue any polling packets. This prevents any node from transmitting information across the network.
2. All nodes listen to check if channel is busy.
3. As soon as a node receives a packet for transmission in its MAC, it generates a new type of packet called “WTS”. The MAC layer considers this as a control packet and allows this packet to go through even in PCF mode without polling.
4. The “WTS” packet is sent across to the point coordinator. All other nodes sense the channel busy state since last beacon.
5. The Control point receives this packet, sends beacon and starts polling. The beacon becomes the ack for the node. If node does not get beacon, retries till it gets it.
6. All nodes identify the beacon and discard their “WTS” packets.
7. Control point completes the polling and issues CF end. It then waits for another “WTS” or beacon interval time to be elapsed the next polling cycle. It issues a beacon after any one of this above event occurs.
8. Nodes will change their state to create “WTS” packet and wait for data.
9. Back to step 2

![Fig 5: Timing diagram with WTS](image)

Figure 5 shows how in a two-node environment the burst and beacon synchronization can be achieved. Here node 1 after receiving data sends WTS, while node 2 defers sensing the channel is busy. The controller on receiving WTS issues beacon to declare the “start of frame” and start polling nodes one by one. The figure 5 clearly shows that the amount of delay has been reduced from the unsynchronized case discussed earlier (Figure 3).

With this new protocol implemented, the polling list immediately after the successful transmission of WTS may also be modified accordingly. The sender of the WTS may come in the top of the polling list from its original position and the rest remains the same.
If a collision occurs while transmitting “WTS” the following steps are followed:
1. The nodes, which experience collision, will see no beacon and retry sending “WTS” after a backoff period.
2. This back off period is large even after first collision to avoid collision in the next attempt.
3. In the next cycle, if collision occurs again, the non-collided nodes will stop attempting, start a counter called “deference counter” and allow the collided nodes to attempt for “WTS”. This is done to ensure lesser number of nodes attempt the transmission.
4. All other nodes sense the channel and stop attempt to send “WTS” even when they have data to send. They also start a timer.
5. After a defined “retry count”, the colliding or unsuccessful nodes stop attempting to send the “WTS”.
6. After elapse of defined time, if poll does not start, the other nodes (non-collided) also attempt to transmit the “WTS”.
7. On successful transmission of “WTS” all nodes go back to their original state.

The backoff mechanism in this protocol is different from Binary Backoff algorithm. Here only one WTS transmission is sufficient for the initiation of the polling process. Hence, nodes become considerate instead of being greedy. At the first instant, all the nodes prepare WTS packet and try to send it. But once they sense the channel busy before transmitting WTS, they assume somebody has already started the process of WTS transmission (in case of WTS collision also), stop attempting and restart a timer (The timer elapsed either when it becomes 0 or the controller sends beacon). This mechanism is called “Sense and deny”. This will help in resolving collision faster by reducing the collision domain every time.

If collision occurs only the nodes collided contend for transmission of WTS. After the collision, the contending nodes use “rapid backoff” mechanism to quickly resolve collision. In the “rapid backoff” method the nodes use extended backoff window to have sufficient number of slots available for avoiding further collision.

The two things here influences the collision resolution process:

- “Sense and deny” mechanism which reduces the collision domain after each collision.
- “rapid backoff” which increase the contention window in a more rapid fashion than Binary Backoff Algorithm.

All the nodes contending for WTS will maintain one “retry counter” to restrict themselves from retrial of WTS transmission even after exceeding that many number of collisions. Then the other denied nodes start trying once their “deference timer” is elapsed. The figure 8 shows the new collision avoidance mechanism with an example of 3 nodes in the network.

4. RESULTS AND DISCUSSION

We have simulated DCF, PCF and proposed enhanced PCF in OPNET to compare the results. The simulation assumes 10-controlled node along with a controller in a wireless channel of 11 Mbps transmission rate. The PCF beacon interval time of 10 ms.. The traffic is defined as follows:

- Burst repetition interval: Exponential (mean 10 ms.)
- Burst on time: 1 ms.
- Packet/burst: 1 packet/burst
- Packet Size: 100 Bytes

The OPNET results for delays with simulation time are shown below:
It is quite clear from the results that although PCF delays are less compared to DCF delays, the delay variance is still high in both the cases. Where as enhance PCF has almost deterministic delays for different nodes. Figure 9 shows the delays are different for different nodes as the nodes are polled in different times in the sequence. Here in the simulation polling table is not modified after every successful WTS to achieve a consistent delay for all the controlled nodes.

We have also done the simulation with varying no. of nodes for all three protocols, results are compiled as average delay vs. number of nodes. The PCF without WTS has higher delays with lesser number of nodes as the synchronization between beacon and packet arrival is poor, and nodes get polled in the next beacon cycle in most of the observation. The DCF delays become higher with large number of nodes because of high rate of collision and retransmission. The graph clearly shows enhance PCF/PCF with WTC gives the same predictable performance in terms of delay with any number of nodes.

5. CONCLUSIONS

Our invention suggests a noble way to enhance PCF protocol for control area networking. It helps polling-protocol to work with bursty traffic with minimal delay of deterministic nature. It provides a means of synchronization between the polling-sequence-start and the availability of data. The nodes here notify the controller about the data arrival in the node and make a relationship between events happening in the controller with the events in the nodes. It also proposes a new collision avoidance technique to resolve the collision faster in the nodes for the burst arrival notification packets.

6. REFERENCES