Prototype Implementation for Service Composition

*System Architecture*

We develop a prototype for our service composition approach to study its performance in the real environments. As shown in Figure XXX, we used 40 MicaZ [] nodes with a MIB600 gateway for the hardware and deployed them in an indoor environment.

|  |  |  |
| --- | --- | --- |
|  | IMG_1035 | PHM037 |
| Fig. 1(a): MicaZ Node | Fig. 1(b): MIB600 Gateway | Fig. 1(c): Part of the Testbed |

The system architecture for the prototype is shown in Figure XXX. In order to support flexible interaction between the application programmer and our service composition middleware, our prototype mainly consists of two major components. On the application side, the application user will provide an XML file which defines service composition. Then the file will be parsed and packetized for dissemination into the pervasive computing environment. On the pervasive device side, after the devices receive the service specification they will start executing the service composition algorithm and notify the user when the composition process is complete.



*.Experiments*

We conducted experiments based on our prototype. We pre-defined a number of services for each sensor node in our testbed. Similar to our simulation, we use CEN as a reference for performance comparison. We implemented CEN based on existing CTP in TinyOS where the nodes will send their available services to sink for composition. We use message cost and delay as metrics for the experiments and study the performance according to of SCP. The experimental results for message cost are shown in the Figures XXX. The data in Sub-figure (a) are obtained from experiments on 10 pre-defined services while the data in Sub-figure (b) are obtained from experiments on 20 pre-defined services. Compared to CEN, LaSeC can reduce the message cost by over 50%. This is primarily because of LaSeC’s decentralized service composition. Unlike CTP where all the messages must be transmitted to the sink, a lot of messages in LaSeC are processed in-network. In addition, LaSeC also scale better when the number of services increases. This is because when the number of services increases, the number of messages to be sent to the sink in CEN will increase accordingly. This inevitably causes a lot of packet collisions and message re-transmission. We believe such cost saving is of particular significance when considering the fact that many devices in pervasive computing environment are battery-powered.

|  |  |
| --- | --- |
|  |  |
| (a) | (b) |

The experimental results for message delay are illustrated in Figure YYY. Similarly, the data in Sub-figure (a) are obtained from experiments on 10 pre-defined services while the data in Sub-figure (b) are obtained from experiments on 20 pre-defined services. We can see that on average, LaSeC achieves smaller delay, especially when there are more services in the environment. This is because in CEN, the delay is primarily introduced by message collision and packet retransmission when the devices try to deliver messages to the sink. On the other hand, the decentralized service composition in LaSeC allows more data to be processed in-network and hence, reduces the chance of packet collision.

|  |  |
| --- | --- |
|  |  |
| (a) | (b) |