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 layers. On the top layer, 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 bottom layer, 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 simply 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 results are shown in the figures XXX. Compared to CEN, LaSeC can reduce the cost by around 20% while only introducing delay of a couple of milliseconds.

|  |  |
| --- | --- |
|  |  |
|  |  |