Programming Approach

• Programming abstraction has been a major focus of research in WSN
• With IoT, heterogeneous devices with different capabilities brings in new issues.
• Essential features for systems to support these changes and user to write applications are as following:
  • Abstraction, Mobility and Modularity

T-Res

• T-Res attempts to provide support for IoT devices
• Tasks are divided into 4 parts: Input Source(is), Output Device(od), Processing Function(pf) and Last Output(lo)
• It uses CoAP and IPv6 addresses to assign tasks to resources
  • Put, Post, Get, Observe
• User inputs via CoAP agent for Firefox, Copper.

mT-Res: Mobility in T-Res

• mT-Res extends T-Res with help of automated CoAP operations
• Simple applications such as
  • node failure
  • Change of host node
  • New application for each node

Application Manager

• Web framework in Django
• For user to submit tasks in 4 parts: Input Type, Output Type, Host (Fixed or Any), and Code.
• Wraps T-Res code with small functions for conditional flags

Resource Administrator

• Python Scripts enabling CoAP functions
• Always active and updating resources
• Provides a table to Application Manager
• Works along the Application manager
• After tasks are submitted, allocates resources
• For any change detected in Resources
  • Restarts the resource allocation

Example

In this demo, we extend capabilities of T-Res to provide autonomous resource allocations for IoT applications. In addition, mT-Res provides a web-interface for user(s) to input applications independent of specific resources. This extension is an effort to support context-aware IoT[3]

Conclusion

References