|  |  |  |
| --- | --- | --- |
| **Módulo 3 – Tema 1** | Hari Balakrishnan |  |

**Video 1:**

IoT = Despertar de las cosas – autonomía.

Objetos: conectarse, captar realidad con sensores, procesar datos, y lidiar con limitaciones (pobreza captación, baja precisión, ruido, defectos.

Las cosas deberán ubicarse (geolocalización) -> cómputos -> acciones. Protegerse de los que quieren controlar, controlar, desvirtuar, destruir. Comunicarse con humanos, escucharlos, repreguntar, y mostrarles el resultado del trabajo.

Que impulsa IoT

* Programas incrustados. En realidad embebidos.
* Sensores / actuadores miniatura. Crecimiento tecnológico en este sentido.
* Redes sin cable. **Pegamento en IoT**

Sensores a servidores -> Comunicación (celular, wifi) -> Rango + energía + flujo de datos ->

Aumentar rango vía mulas de datos.

Aplicaciones sensibles a la latencia???

Diferencia entre Internet e IoT: valor de los datos???

Tecnologías actuales como servidores computadoras, pcs, notes, cels -> tecnologías +/- estándares. Si bien hay evolución e innovación. **La evolución es predecible. En redes de dispositivos y sensores de IoT no así. Campo reciente. Hay confusión, muchos estándares, perspectivas.**

Maraña de conectividad posible: bluetooth, zigbee -> **No hay formula universal para conexión IoT. DEPENDE DE LA APLICACIÓN.** Hay que pensar en principios organizativos clave e ideas.

Cricket: es como un GPS de interiores. Funciona como combinación RFs + ultrasonido. Balizas sobre paredes, techos, transmiten msgs baliza en RF y US. Receptores hard similar miden dif. tiempo entre la RF y la SN. Sonido viaja mucho más lento que la luz. Esa dif se puede saber la distancia entre balizas. Aún se usa en hospitales.

CARTEL: uno de los primeros sistemas móviles de sensores. Sensores en los vehículos recolectando datos interesantes. Lo que hoy se conoce como vehículo conectado. Nuevas formas de conectar vehículos con WIFI y utilizar WIFI vehicular para conectar a internet. Nuevas formas de pensar tecnologías de satélite + map matching.. Sistemas de tránsito, enrutamiento, pot holes peligrosos (baches).

DRIVEWELL: aún disponible en mercado. Artefacto IoT con sensores en parabrisas + sistema en Smartphone que proporciona info. de cuan prudente es el conductor. Además detectar accidentes en tiempo real.

**Video 2:**

Arquitectura simplificada:

* Dispositivos, sensores a la web vía puertas de enlace (mediar y traducir entre dispositivos e Internet).
* Estáticos en la casa, oficina, calles, rutas.
* Móviles como el celulares, tablets.

La manera de **conectar dispositivos con las puertas de enlace** son muchas pero la mejor respuesta es que **depende de la aplicación!!!**.

Podemos utilizar wifi y red celular salvo que no se pueda.

* Disponibles.
* Ancho de banda.
* **Pero consumen mucha energía** -> no es bueno para escenarios que operen con baterías.
* **CELULAR mayor desventaja alto costo.** Sobre todo si se transfieren pocos datos y se paga por mes demasiado. *Buena opción para vehículos conectados (cosa de alto valor y motor). No hay problemas de energía.*
* **WIFI** no tiene costo pero **problemas de alcance. *Buena opción para lavarropas, heladeras.***

Análisis del consumo energía celular.

* Izquierda y derecha del uso en reposo.
* Mucha energía con pico al transferir datos (uso).
* Luego de transferir datos queda aun consumiendo energía. Ese período en 2 partes. La primera por “residuos” de la comunicación y la **segunda consume energía aunque esté en reposo.**

**Video 3**

Espacio de diseño de redes IoT en 3 dimensiones (VER IMG 3 niveles red IoT.png):

* Duración de batería.
* Cantidad de datos.
* Distancia a la puerta de enlace.

WIFI: GB por día + pocas hs. de almacenamiento con esa transferencia + alance red doméstica/campus.

LTE (celular): se logra un poco menos que WIFI: GB + unas horas + alcance superior pero a mayor costo.

Hay **muchas redes** porque nos gusta inventar tecnologías pero porque se puede **elegir regiones de diseño dentro de este espacio**.

* Estos 3 ejes no son independientes.
* Evolución rápida de tecnologías.
* Integración de tecnologías en aparatos populares. +adquisición -> impacto en economía. Ejemplo WIFI (se incorporó a laptops). Bluetooth. **Bluetooth de baja energía BLE muy usado en red doméstica (todos los celulares lo tienen por lo que puede ser puertas de enlaces directas)**.

**Video 4**

BLE (B baja energía): corporal y hogareña.

* Hasta 10 metros. 50 alta transmisión en buenas condiciones.
* Pico de 100KB/s.
* Cientos KB/día dura desde meses hasta años. GB/día reduce considerablemente.
* Nació como Wibree (BLE) por Nokia y aún es dominante por haberse incorporado en los celulares ->Puertas de enlace naturales.

Como funciona BLE:

* Etapa de device discovery (**emisión**): los aparatos emiten avisos (balizas), tienen datos por enviar.
* Segunda etapa (**conexión**): puertas de enlace y otros aparatos pueden conectarse intercambiar datos.

Arquitectura BLE:

* Periférico: dispositivo con datos.
* Central: puerta de enlace.

Avisos:

* Los avisos son periódicos por ejemplo típico de 100 milisegundos donde se comunica la existencia. Se puede aumentar ese período para +batería y –detección por parte de la central. DEPENDE DE LA APLICACIÓN.
* Se puede programar. Por ejemplo un dispositivo en el parabrisas solo emite avisos cuando el auto se mueve.

Comunicación:

* Establecida la misma la central identifica cosas del dispositivo. PROTOCOLO PARA ELLO. MAC.
* De manera jerárquica. Servicios luego características por servicio.
* Características solo lectura como ritmo cardíaco.
* Características de lectura/escritura.
* Características de notificación. Si los datos cambian, se notifica al central.
* Algunos soportan OTA over the air upgrades (programación).

Protocolo MAC (media access):

* Central orquesta la conn.
* **Idea clave: programar en el tiempo para disminuir consumo energía.**
* Cuando central y periférico se conectan intercambian parámetros:
  + Conjunto de frecuencias para comunicarse.
  + Periodicidad de conn.
* Cada T segundos, pueden intercambiar 1-4 paquetes de datos, alternándose.
* No hay sobrecarga una vez establecida la conn. y hayan intercambiado parámetros.
* El resto del tiempo, periférico en reposo.

Cálculo de tiempo de vida de la batería:

* Teniendo en cuenta pila de botón con capacidad: 250 mAh/h; 3 volts. Potencia = voltaje \* corriente. Corriente = potencia \* tiempo.
* => 3 \* 0.250 AH/h = 0.75 amp-h-volts => 0.75 \* 3600 jules = **2.7 KJules Energy.**

Ejemplo:

* Consumo de corriente en dispositivos BLE aprox. 1-10 mAh. Tomamos 1 mAh. Standby
* El hardw recibe a 3.3 mAh.
* Al transmitir sube a 4 mAh.
* Si suponemos que el dispositivo transmite cada 1 milisegundo. Sería una app. pequeña de transmisión de 12 bytes (1000 bit)s.
* **VER Cálculo batería.png**

**Video 5**

Sistemas IoT edificios => Redes en malla.

Saltos inalámbricos entre dispositivos para llegar a la puerta de enlace y conectarse a Internet.

Historia previa:

* 70s 80s DARPA red de radio paquetes.
* 90s MANETs: malla de nodos móviles. Comunicación multi salto cuando los dispositivos están en movimiento.
* 90s 2000s redes de sensores. Adoptar las ideas de los 80s 90s y dar nuevas soluciones, teniendo en cuenta bajo consumo de energía.

Mediados de 2000s nacen las redes en malla para Internet. Saltos entre dispositivos BLE.

* Zigbee: malla multi salto IoT.
* 6LoWPAN: IPV6 para redes de área personal de bajo consumo.
* Ambas protocolo de enlace 802.15.4

VER IMG.

Algunos dispositivos tendrán **más trabajo que otros**. Aquellos cerca de la puerta de enlace tienen más tráfico en ambas direcciones => Consumirán +energía.

Reglas generales:

* Estos pueden ser nodos conectados a la red.
* Fácil reemplazo de batería.
* Tecnología solar.
* Analizar los nodos más activos.

Ruta a seguir de los paquetes:

* Cada enlace lleva 1 paquete con cierta probabilidad.
* El enlace sufre de ruido, etc. => No son confiables => Pérdida de datos.
* **“Red en malla – routing.png”:**
  + ETX: tiempo esperado de transmisiones. Si la velocidad de enlaces es fija.
  + 1/probabilidad de enviar satisfactoriamente el paquete.
  + Calcular eso por cada salto y sumarlos.
  + Elegir el menor.

**Video 6**

Escala de 1 ciudad.

Red tolerante al retraso (DTN):

* Acareo de datos (data mules) por ejemplo proyecto CARTEL, en el que cada auto los almacenaba y al tener conexión los transmite. Inclusive entre autos.
* En aplicaciones que acepten retraso de trasmisión de datos (delay). No real time.
* **“Redes de kilómetros de distancia.png”**

Largo alcance y bajo retraso (nuevas saliendo al mercado):

* Varias tecnologías: Sigfox, LoRaWan, propuestas IoT celular.
* Algunas abiertas otras más cerradas.
* Diseño que requiera muy poca energía para que con baterías los dispositivos duren meses.
* En las 3 dimensiones se logra largo alcance con baja batería, pero ciclos de trabajo lentos.
* **Por lo tanto se está dando en aplicaciones de baja-ultra baja productividad.** Algunos bytes por día.
* LoRaWAN agrega funcionalidades de localización. Es muy utilizado para evitar GPS por ejemplo que consumen mucha batería.

Segunda opción, utilizar el celular:

* LTE, 3G, 4G.
* Alto consumo de energía.
* Igualmente tienen un retraso entre servidores y sensores sobre todo en latencia de datos.

**Video 7**

Aplicaciones IoT sensibles de latencia o intensivo real-time.

Ejemplo: reconocimiento continúo (basado en imagen). **“Ejemplo reconocimiento apps.png”**

* Asistencia al conductor.
* Aprendizaje y reconocimiento de objetos con el celular.
* Con anteojos de realidad aumentada.

Glimpse (MIT development):

* Reconocimiento continúo de objetos. Reconocimiento facial: no solo que hay una persona sino quién es.
* Alta tasa de fotogramas (HFR): 24 a 30 cuadros por segundo.
* En video stream del celular.
* También para reconocer señales de tránsito.

Funciona en 3 etapas:

* Captura imágenes en una trasmisión de video, detección de todos los objetos **HFR**, donde están – características de ellos.
* Extracción: tomar los objetos de interés. Dividirlos en características para saber que son.
* Clasificar: en una de las categorías.

Es una tarea costosa. Los servidores son mucho más rápido que los cels. Capturar imágenes, enviar las imágenes al servidor, que sean reconocidas y respuesta tarda segundas cuando se espera 30 cuadros por segundo en realtime.

Desafío!!! **“Solución liberación al server.png”**:

* Ceder computación a los servidor, genera latencia.
* Hay que tomar medidas de eficiencia en el uso del ancho de banda y energía.

Idea de caché extra activa:

* Se envía el primer cuadro al servidor y el cel sigue capturando.
* Una vez que el servidor responde con la identificación, se retrasa solo eso porque el celular luego puede seguir los objetos etiquetados mientras avanza el video.
* **Toda la extracción de características y el reconocimiento de objetos en el servidor mediante etiquetas, rastreo de etiquetas en el celular. El retraso solo se da en el reconocimiento de nuevos objetos.**
* Se puede no enviar siempre sino cuando la imagen cambiar sustancialmente. **Cuadros disparados.**

Desempeño. “Desempeño Glimpse reconocimiento facial.png”:

* Precisión: nro. de imágenes detectadas y etiquetadas correctamente / total que reportamos.
* Memoria: nro. total que reportamos / nro. real de objetos.
* En reconocimiento de señales de tránsito es aún mejor **“Desempeño Glimpse reconocimiento símbolos.png”**. En el servidor es muy bajo debido a que en facial se utiliza cuestiones de hard del teléfono y en este esas funcionalidades no están aún y se tiene que desplegar en el server.

**Video 8**

Conclusión

* Diseño variado para conectar cosas con puertas de enlace.+
* Tener en cuenta las 3 dimensiones.
* Vimos los 3 casos de estudio BLE + extensión de alcance + reconocimiento.

No estamos siquiera al final de inicio de IoT:

* Se ven aplicaciones absurdas como… quien quiere encender la luz yendo a una app del celular.
* Que pasa a nivel ciudad con alta tasa y baja batería. Ejemplo sensores de vibración de alta fidelidad monitoreo de terremotos, sensores de clima de alta fidelidad, sensores de imágenes. O hay respuesta.
* Sistemas actuales están casi siempre en bajo cónsul. Trabajos recientes lograron reducir microWatts a nanoWatts el consumo de reserva.
* Las apps están aisladas una de otras.

Forma de arreglárselas:

* Utilizar sensores / dispositivos IoT. O construirlos.
* Montar app usualmente en cel. O corriendo en la puerta de enlace.
* Desplegar un webservice para procesar datos. Inferencia de datos.
* Estás apps. Están integradas verticalmente. Difícil e hacerlo e innovar.
* Las funciones de la puerta de enlace se reinventan una y otra vez.
* **El valor real está en los datos no en los dispositivos**, ni siquiera machine to machine.

Soluciones (combinar):

* Coordinar accesos a servidores vía APIS en la nube.
* Acceso a los datos en apps celulares vía “kits” (HomeKit, HealthKit, Google Fit).
* Desarrollo de puertas de enlaces generales con varias tecnologías.

Predicciones:

* Cambio en estándares. Múltiples ganadores. Van a dividir el espacio de 3 dimensiones y quedarse con un pedazo cada uno.
* Ajetreo en IoT de muy bajo consumo.
* Ajetreo en IoT computación intensiva. Como el reconocimiento.
* Arquitecturas no aisladas. APIs- KITs.
* Segmentación centrados en celulares y ocultos (oblicuos) computacionales. Actualmente el cel es la puerta de enlace + UI.

**FORO 2**