


Tesis de Grado de Ingeniería en Informática

Tráfico determinístico en Internet de las Cosas (IoT)

Alumna: Ana Czarnitzki Estrin

Tutor: J. Ignacio Alvarez-Hamelin

Cotutor: Georgios Z. Papadopoulos



El objetivo de esta tesis fue el de proponer formas de obtener tráfico determinístico en redes inalámbricas de dispositivos IoT en ambientes industriales.



(Más sobre esto en los próximos minutos)

Índice

1. Introducción
2. Estado del arte
3. Protocolos y software
4. Algoritmos propuestos
5. Análisis de los algoritmos
6. Conclusiones

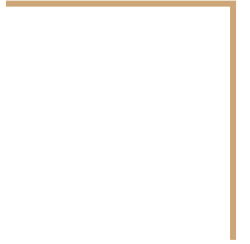


Varian que puede tener unos maullidos que acotar

Introducción

- A. Internet
 - B. Internet de las cosas (IoT) e Industria 4.0
 - C. Determinismo en redes industriales
-

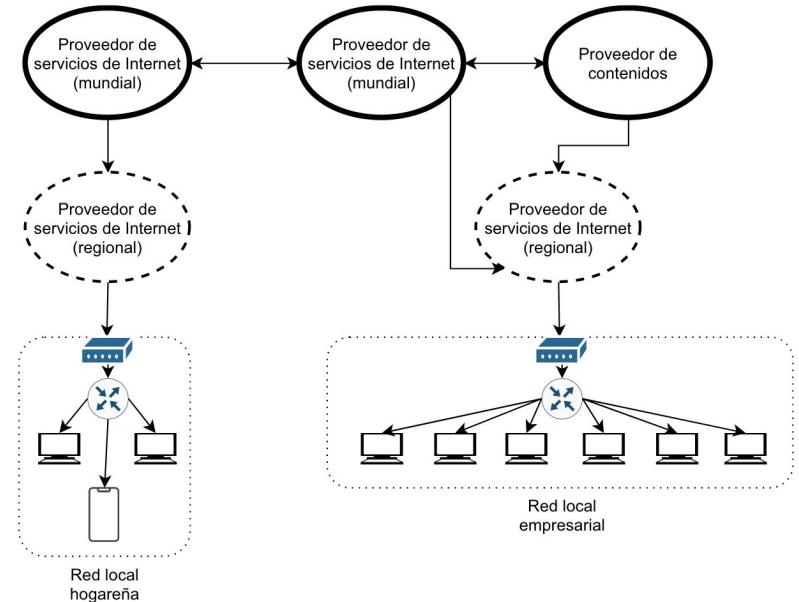
Internet



Qué es Internet?

Internet es una *red de redes de computadoras* que *conecta* millones de dispositivos, llamados hosts o anfitriones, permitiendo que los mismos *intercambien paquetes de comunicación*.

Internet no fue diseñado para un uso en particular, sino como una *infraestructura* sobre la cual se puedan desarrollar *múltiples usos*.



Protocolos y capas

Internet se basa en diferentes *protocolos de comunicación* que determinan cómo, cuándo y de qué formas los diferentes *hosts* van a enviar y recibir mensajes.

Estos protocolos se organizan en capas, donde cada capa le provee servicios a su capa superior.

Aplicación - aplicaciones de internet

Transporte - servicios de comunicación para las aplicaciones

Red - servicios de comunicación para los *hosts*

Enlace - transporte de información de un nodo adyacente a otro

Física - transporte de bits individuales de un nodo a otro



Internet de las cosas e Industria 4.0



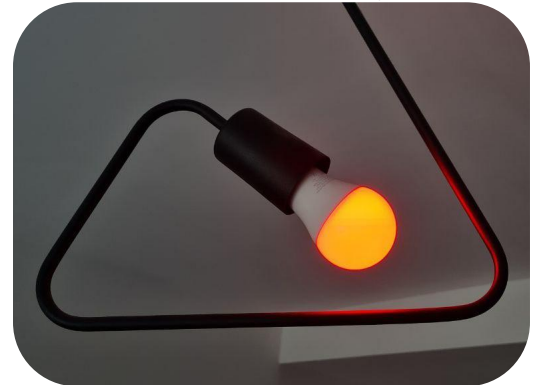
Internet de las Cosas

Internet de las Cosas (Internet of Things, *IoT*) refiere a la *conectividad de los objetos cotidianos* y tangibles del mundo material *entre ellos mismos*.

“Una red abierta de objetos inteligentes que tienen la capacidad de *auto-organizarse, compartir información, datos y recursos*, reaccionando y actuando debido a *diversas situaciones y cambios* en su ambiente”



One-one, mi aspiradora



Lámpara smart, cortesía de Juanma

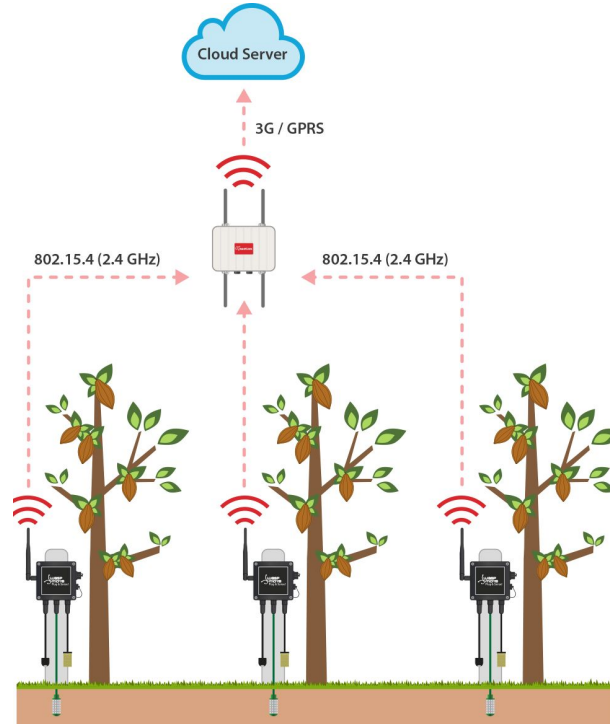
Industria 4.0

La industria 4.0 refiere a una nueva revolución industrial marcada por la *automatización* total y *digitalización* de los procesos productivos y el uso de maquinaria electrónica y *tecnologías de la información* en estos.

La industria 4.0 aún se encuentra en un *estadio conceptual* y se basa en *tecnologías muy dinámicas* que involucran a *múltiples industrias*.

En general, en este fenómeno se tiene un flujo constante de información generado desde los consumidores y desde los procesos productivos para que se puedan tomar decisiones en tiempo real.

Industria 4.0 en la práctica





Determinismo en redes industriales

El objetivo de esta tesis



De qué hablamos cuando hablamos de *determinismo*

El determinismo refiere a que ante una causa estable y conocida, la consecuencia será estable y conocida. En otras palabras, que una entrada siempre generará la misma salida.

En el contexto de este trabajo, hablar de una *red determinística* (y *confiable*) es hablar de una red donde siempre que se envía un mensaje, el mismo se recibe y, además, que los mensajes siempre tardan lo mismo en llegar a sus destinos.

Determinismo en redes industriales

Las redes de dispositivos IoT inalámbricas otorgan mejoras en eficiencia y productividad a las industrias manufactureras, sin embargo, debido a la alta necesidad de comunicaciones determinísticas y confiables que tienen estas industrias, su adopción al día de hoy es baja.

Ventajas del uso de redes de dispositivos IoT

- Auto-organización, rápido despliegue y flexibilidad.
- Abaratamiento de costos al no necesitar cables.
- Posibilitan casos de usos más amplios.

Desventajas del uso de redes de dispositivos IoT

- Recursos limitados de los dispositivos IoT.
- Presencia de topologías dinámicas y condiciones ambientales duras.
- *Requerimientos en la calidad de servicios.*

Estado del arte

(Lo que existe hoy)

- A. Caminos múltiples
 - B. Scheduling
 - C. Otras alternativas
-

Estado del arte

Caminos múltiples

Generan copias de los paquetes, donde cada copia usa un camino distinto de la red.

De esta manera, se agregan oportunidades de que un mensaje llegue de forma correcta a su destino.

Scheduling

Se construyen *schedules* bajo ciertos criterios para que los nodos de la red transmitan y reciban sus mensajes de formas predecibles.

En general, estas estrategias se combinan con algún uso simple de caminos múltiples.

Otras alternativas

Detección de errores y corrección de los mismos.

Uso de retransmisiones.

Y más.

Estado del arte - Desafíos

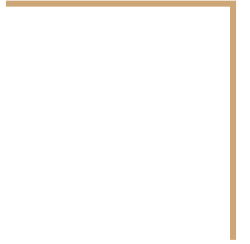
Las soluciones existentes el problema en análisis suelen tener como problemas su *alto consumo de energía* y/o *una alta variabilidad en los valores de demora en la entrega de paquetes*.

Tener una alta variabilidad en los valores de demora implican que una red no es determinística.

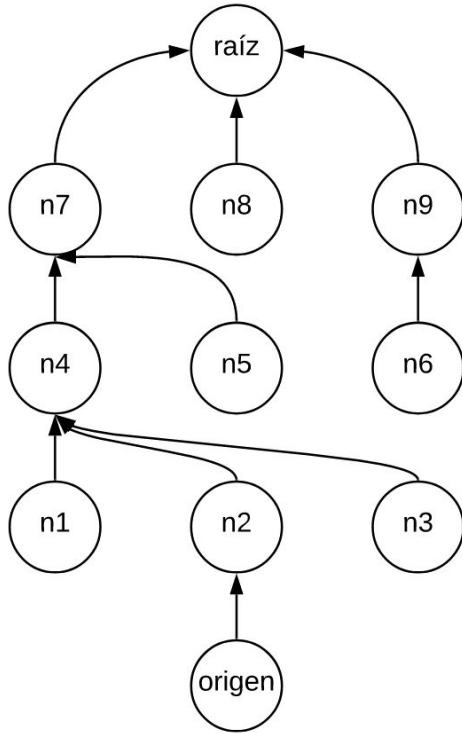
Protocolos y software

- A. RPL
 - B. TSCH
 - C. Contiki y COOJA
-

Protocollo RPL



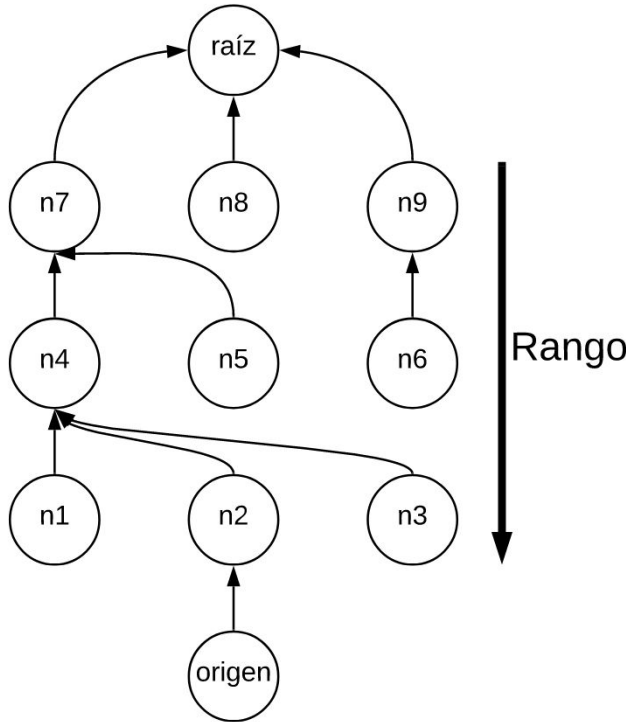
IPv6 Routing Protocol for Low-Power and Lossy Networks



RPL es un *protocolo de capa de red* diseñado para acoplar redes de dispositivos IoT a la *red IPv6* de Internet.

Este protocolo genera uno o varios *DODAGs* (grafos acíclicos dirigidos con orientación a una destinación), donde *las hojas del grafo son dispositivos IoT* que quieren conectarse a Internet y *la raíz el border router* que los conecta a la misma.

Construcción del DODAG



Para generar al DODAG, *cada nodo elige un padre preferido*.

Esta elección de padres se realiza gracias a una **Objective Function (OF)** que *toma varias métricas* como parámetros y *devuelve un valor único denominado rango*.

Cada nodo elegirá como padre preferido (PP) a aquel nodo del que reciba el valor de rango menor. Todos los nodos de los que se reciba un rango menor al propio formarán el **set de padres (PS)** de cada nodo.

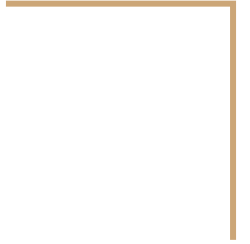
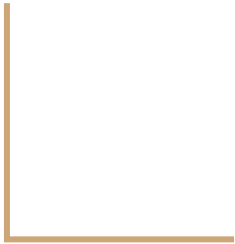
Mantenimiento y reparaciones de los DODAGs

Para mantener los DODAGs y actualizarlos en caso de que parte de los enlaces resulten inaccesibles o se tenga información incorrecta, *todos los nodos envían de forma periódica paquetes de control* según un temporizador.

La frecuencia de este temporizador aumenta al detectarse inconsistencias en la información recibida de estos paquetes *y disminuye en caso contrario*.

RPL cuenta con mecanismos de *reparación global* (se rearma toda la estructura) y *local* (se elige un camino alternativo temporal para un nodo) pero estos mecanismos son lentos y no solucionan todas las posibles pérdidas de la red.

Protocollo TSCH

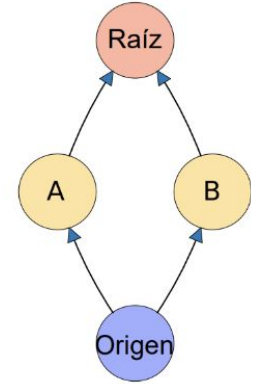
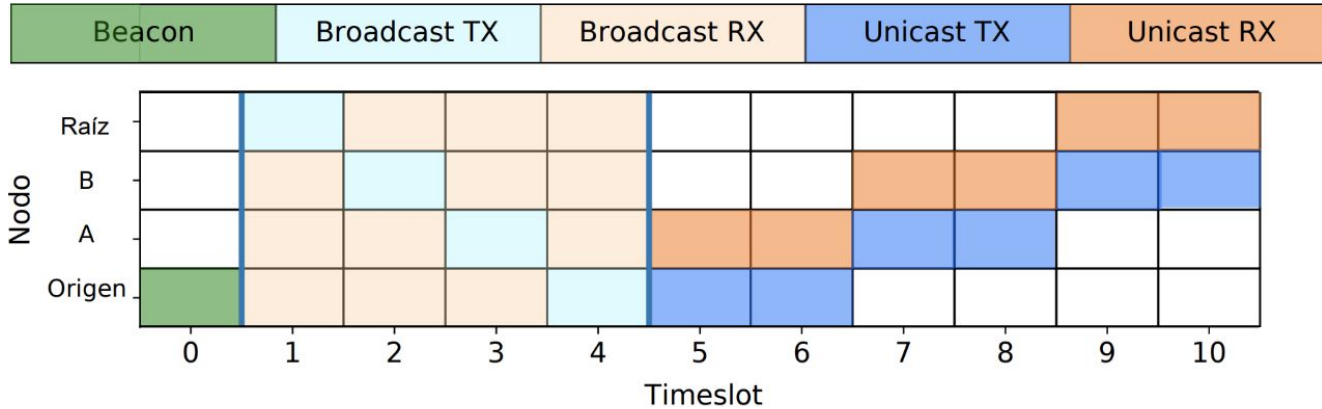


IEEE Std 802.15.4-2015 Time Slot Channel Hopping

TSCH es un *protocolo de capa de enlace* que combina *Time-Division Multiple Access (TDMA)* y *Frequency-Division Multiple Access (FDMA)*.

TSCH permite el acceso a un medio de transmisión compartido dándole rangos de tiempo específicos a cada transmisor y dividiendo el espectro disponible en canales de manera de que *dos transmisores no utilicen el mismo canal al mismo tiempo*.

Registro de TSCH



El protocolo *divide el tiempo en intervalos que se agrupan en un timeslot* (registro de tiempos) que se repetirá indefinidamente mientras la red exista.

Las transmisiones y recepciones entre nodos tendrán asignado un intervalo y un canal. A este *par intervalo-canal* se le llama *celda*. Las *celdas* pueden ser *dedicadas o compartidas*.

Sistema operativo Contiki

Contiki

Contiki OS es un *sistema operativo de código abierto* diseñado *para dispositivos IoT*.

Está hecho en *C* e *implementa de forma nativa* muchos de los protocolos estándar de Internet para dispositivos IoT, incluidos *RPL* y *TSCH*.

Simulador COOJA

The screenshot displays the COOJA network simulator interface. The main window, titled "My simulation - Cooja: The Contiki Network Simulator", features a menu bar with "File", "Simulation", "Motes", "Tools", "Settings", and "Help". The central area shows a network diagram with 32 motes (nodes) arranged in a grid, connected by red and blue lines representing network links. A "Simulation control" window is overlaid on the network, showing "Run", "Speed limit", "Start", "Pause", "Step", and "Reload" buttons, along with a "Time: 04:36.452" and "Speed: ---" display. Below the network diagram, a code editor window shows the following JavaScript code:

```
294 var allMoteIds = getAllMoteIds();
295 for(otherMoteIdIndex in allMoteIds) {
296   var otherMoteId = allMoteIds[otherMoteIdIndex];
297   if(otherMoteId != moteId){
298     unsetEdge(moteId, otherMoteId, priority);
299     unsetEdge(otherMoteId, moteId, priority);
300   }
301 };
302 };
303 };
304 };
305 radioRatioRandomSetter = function(){
---
```

At the bottom of the interface, a timeline window shows the execution of 32 motes, with a "Timeline showing 32 motes" label. The timeline displays various events and messages for each mote, represented by small colored squares and lines.

Contiki provee un simulador de redes llamado COOJA realizado en Java.

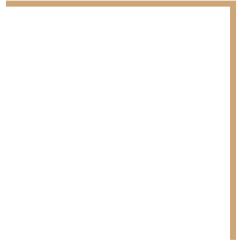
Este es el simulador que se utilizó para hacer el análisis de los algoritmos propuestos para esta tesis.

Algoritmos propuestos

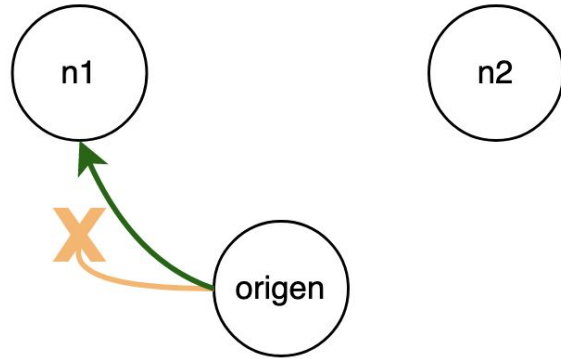
Algoritmos n-Disjuntos

- A. Conceptos previos
 - a. Retransmisiones y réplicas
 - B. Algoritmos n-Disjuntos
 - a. n-Disjunto: por defecto
 - b. n-Disjunto: controlado
-

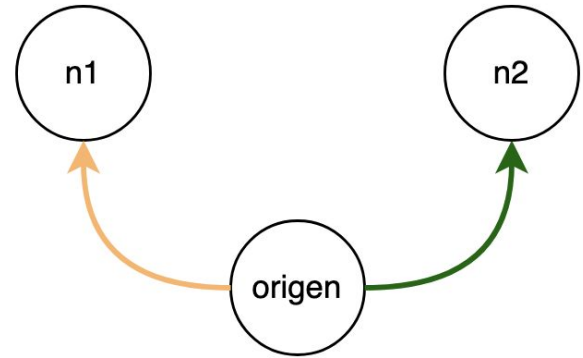
Conceptos previos



Retransmisiones y réplicas



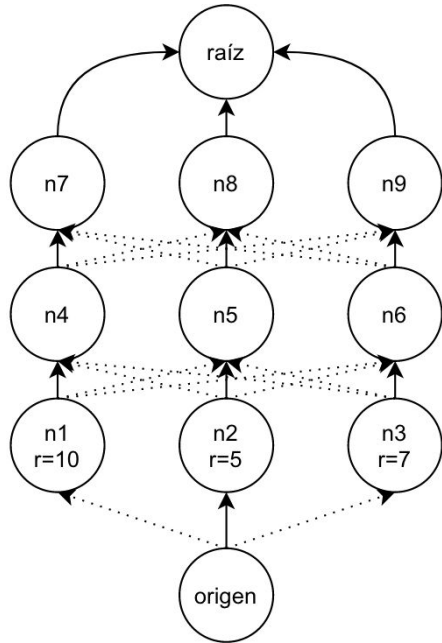
Las **retransmisiones** son oportunidades adicionales para enviar un paquete. Si el paquete está perdido o se considera perdido, se realizará una nueva transmisión que recorra el mismo camino que la original.



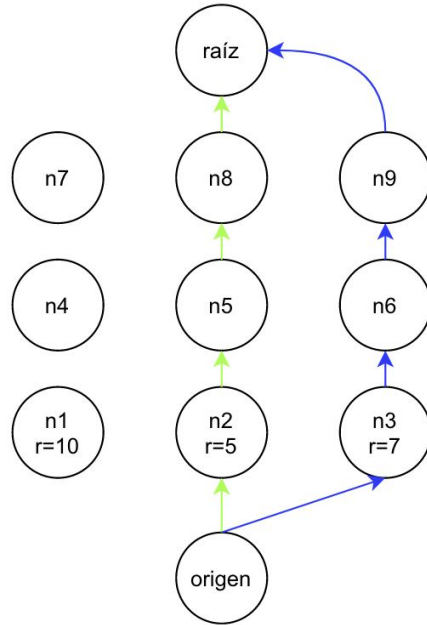
Las **réplicas** son copias de un paquete que se envían para anticiparse a pérdidas potenciales. Cada réplica puede utilizar un camino distinto y todas se transmiten a la vez.

Algoritmos n-Disjuntos

n-Disjunto: por defecto - Réplicas



DODAG



Caminos generados por el algoritmo

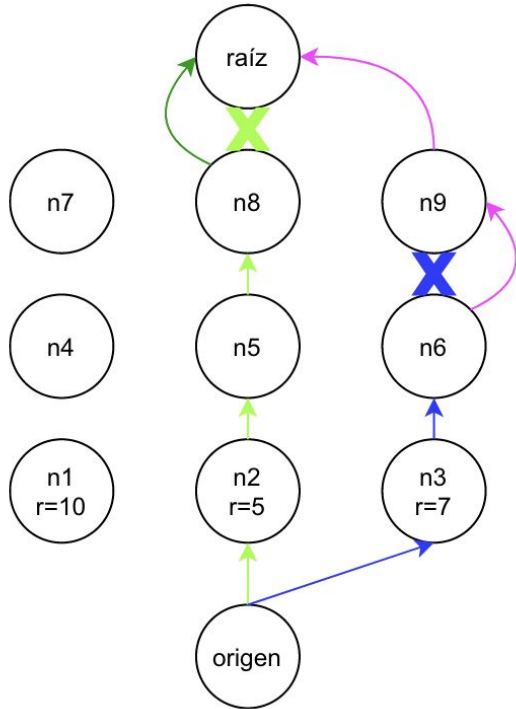
Usando como base a RPL, *el nodo origen selecciona $n + 1$ padres* de su set de padres usando la métrica por defecto de RPL.

El origen envía una copia a cada uno de sus padres seleccionados (n réplicas más el paquete original).

El resto de los nodos reenvían los paquetes a su padre preferido.

El valor de n debe cumplir: $n < |PS|$.

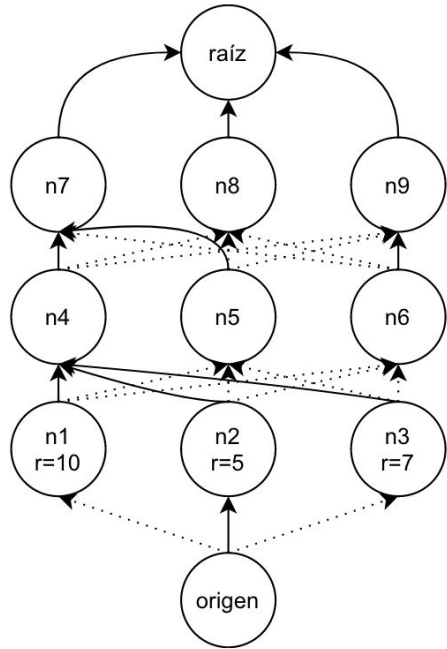
n-Disjunto: por defecto - Retransmisiones



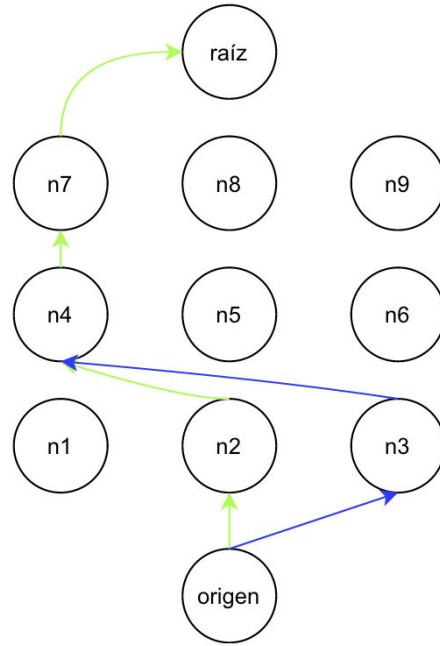
Se permiten retransmisiones, tanto del paquete original como de las copias, a nivel capa de enlace, *implementadas por TSCH*.

La máxima cantidad de retransmisiones a realizar antes de asumir a un paquete como perdido definitivamente *es configurable*.

n-Disjunto: por defecto - Caso *no* ideal



DODAG

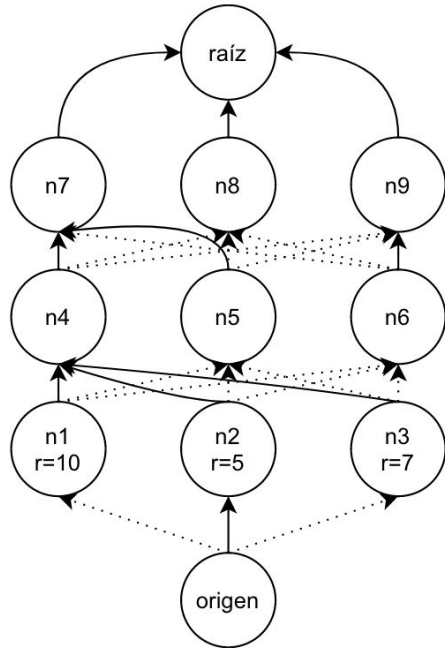


Caminos generados por el algoritmo

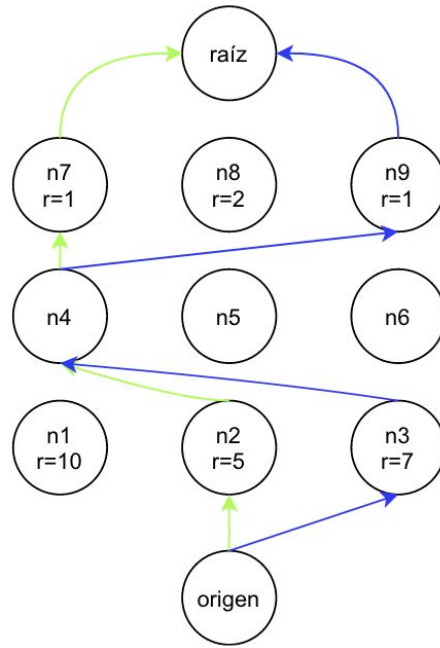
En la versión por defecto, *si dos o más nodos tienen el mismo PP, sus caminos convergerán.*

Si dos o más copias llegan al mismo nodo, *sólo una será reenviada* al padre de este nodo.

n-Disjunto: controlado



DODAG



Caminos generados por el algoritmo

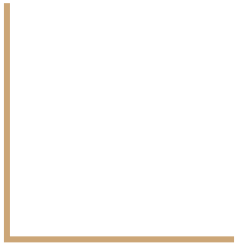
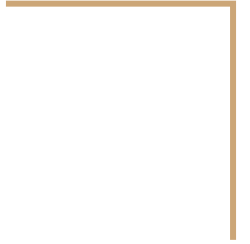
En este algoritmo, *si un nodo detecta que ocurrió una convergencia* (recibe dos o más copias del mismo paquete de distintos nodos) *reenvía las copias a diferentes padres de nuevo.*

El resto de los nodos selecciona $n + 1$ padres de su PS usando la métrica por defecto. Si se recibe una réplica, el mejor padre aún no utilizado es seleccionado.

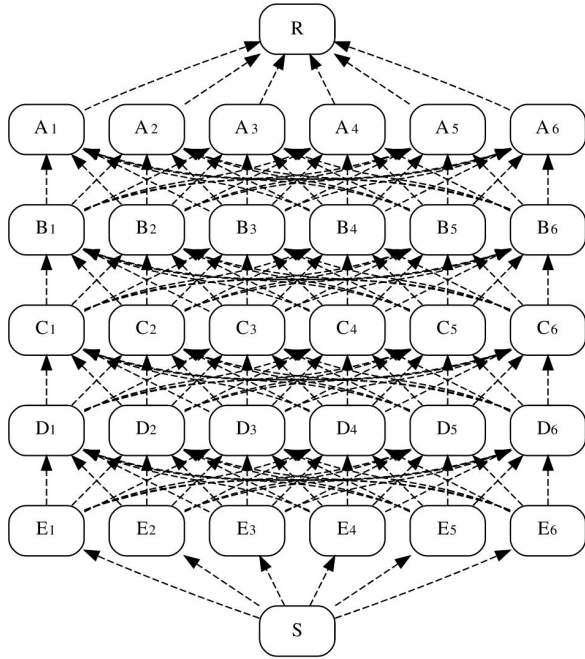
Análisis de los algoritmos

- A. Configuración de las simulaciones
 - a. Topología de la red y configuración de TSCN
 - b. Escenarios
 - B. Resultados
 - a. PDR
 - b. Demora y su variabilidad
 - c. Cantidad de copias por paquete
 - C. Comparación de los algoritmos
-

Configuración de las simulaciones



Topología de la red y configuración de TSCH



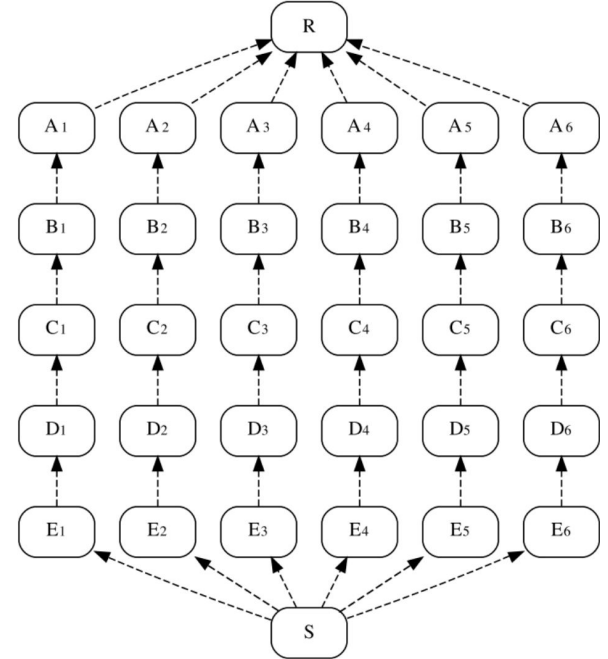
Topología

32 nodos, S es el origen y R la raíz

Los nodos se encuentran en rango de comunicación con los nodos de sus niveles superiores e inferiores

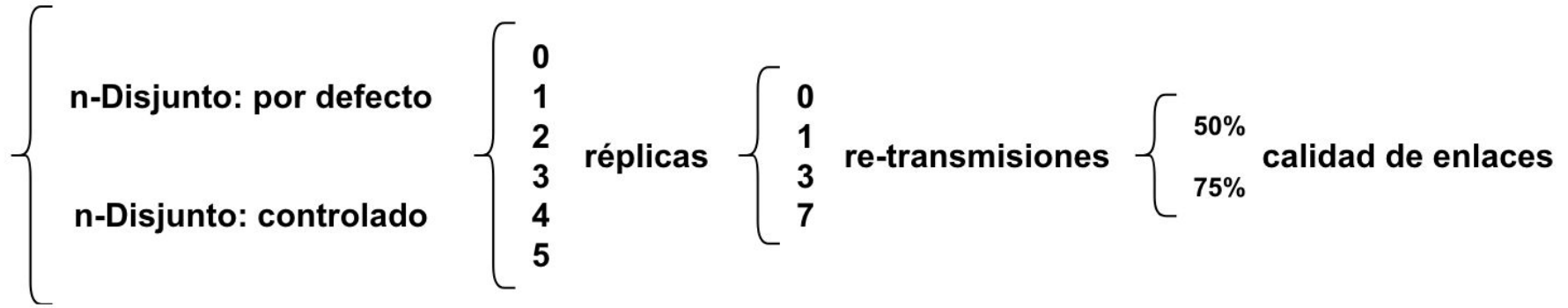
Único canal

Cada nodo tiene dos celdas para transmitir a cada uno de sus posibles padres



Caminos disjuntos posibles

Escenarios

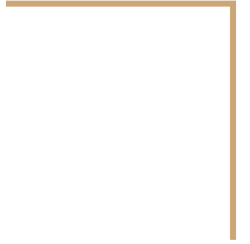


Se tienen 96 escenarios en total

Cada escenario tuvo 20 simulaciones

Cada simulación envió 250 paquetes

Resultados



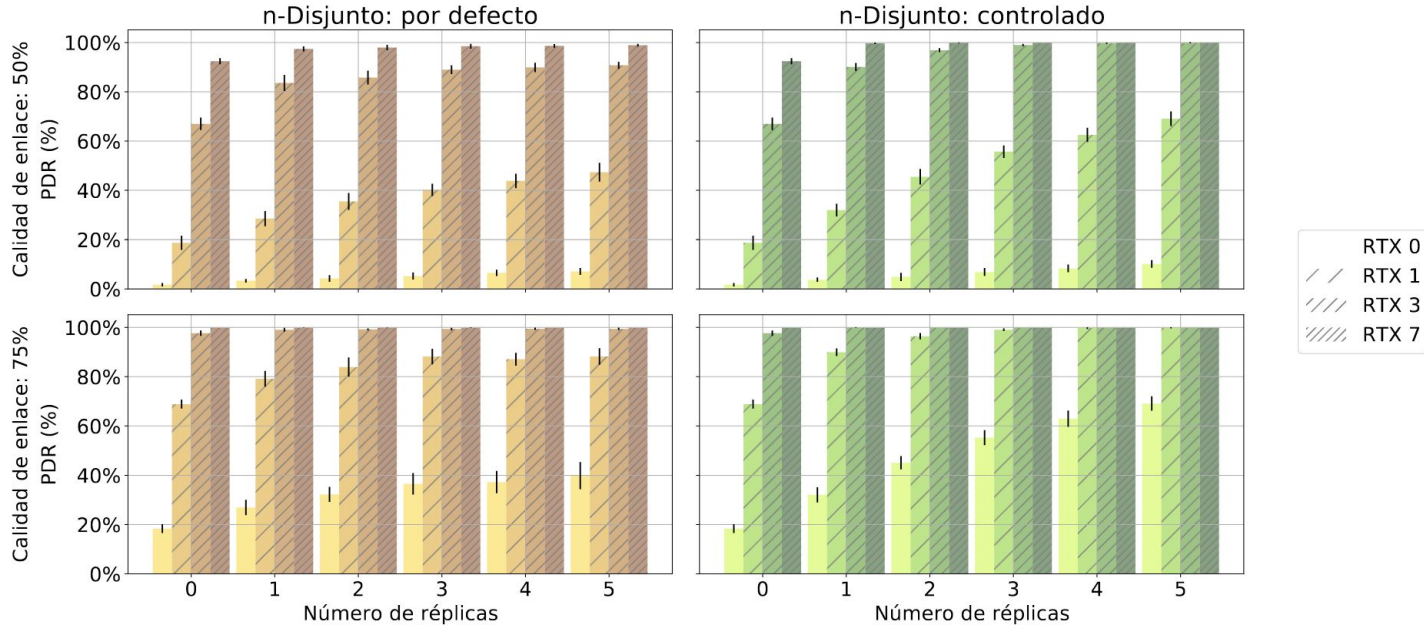
Todos los gráficos mostrados a continuación siguen el mismo esquema:

- *A la izquierda, en naranja*, los resultados para el algoritmo *por defecto*.
- *A la derecha, en verde*, los resultado para el algoritmo *controlado*.
 - *El valor de n* utilizado se encuentra en el eje *X*.
 - *La métrica* en análisis se encuentra en el eje *Y*.
 - *Las líneas negras* muestran *la variabilidad* de cada métrica.
- La *cantidad máxima de retransmisiones* está indicada *según color y según rallado*.
- Se tienen los valores para calidades de enlace de 50% (arriba) y de 75% (abajo) en el eje *Y*.

El *Packet Delivery Ratio (PDR)* mide el *porcentaje de paquetes distintos recibidos exitosamente* frente a la cantidad total de paquetes distintos enviados.

En una red determinística y confiable, *esperamos un valor de PDR alto.*

PDR - Packet Delivery Ratio

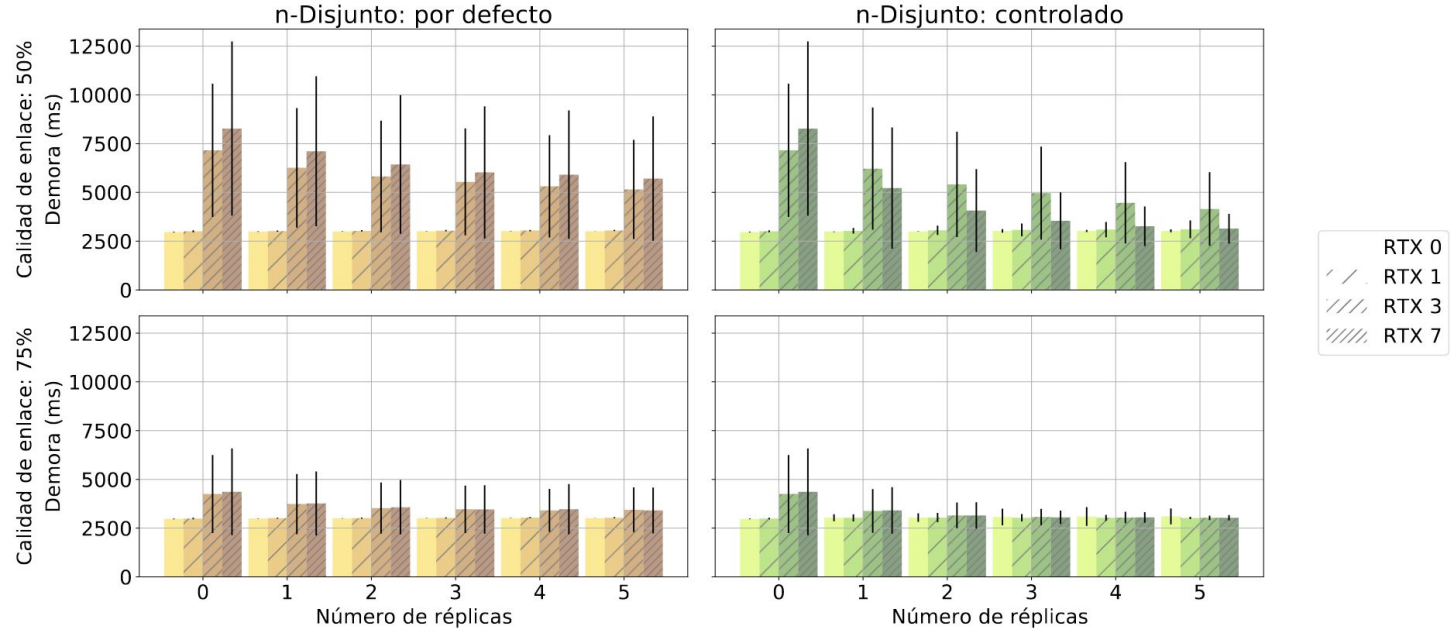


El PDR aumenta de forma proporcional con la cantidad de retransmisiones y réplicas.

La *demora* refleja *cuánto tarda cada paquete* en ser entregado. La *variabilidad de la demora es la diferencia máxima entre las demoras* de los paquetes en una red.

En una red determinística y confiable, *esperamos un valor de variabilidad en la demora bajo.*

Demora y su variabilidad

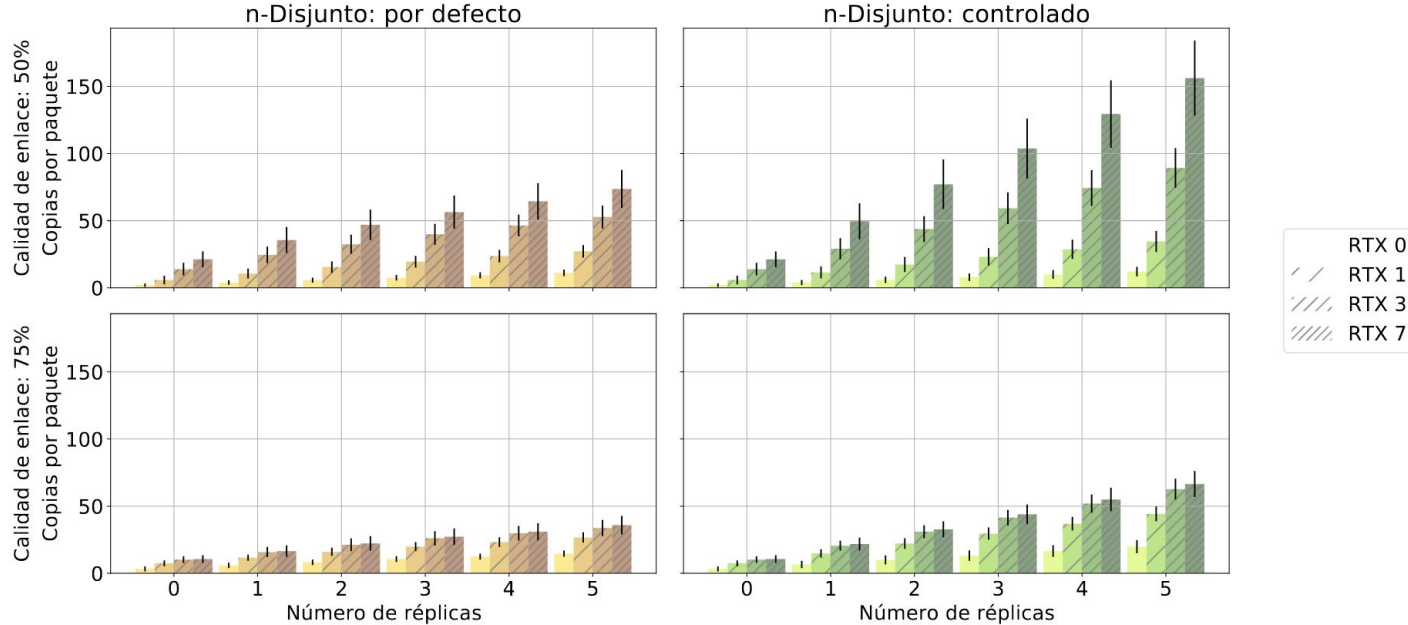


La demora y su variabilidad aumentan al agregar retransmisiones, pero se mantienen al usar réplicas.

La *cantidad de copias* por paquete es la *sumatoria de la transmisión original más todas las réplicas y retransmisiones* de este mismo paquete.

Este valor *será proporcional a la* cantidad de *energía de la red* utilizada, *en una red IoT este valor debería ser bajo.*

Cantidad de copias por paquete



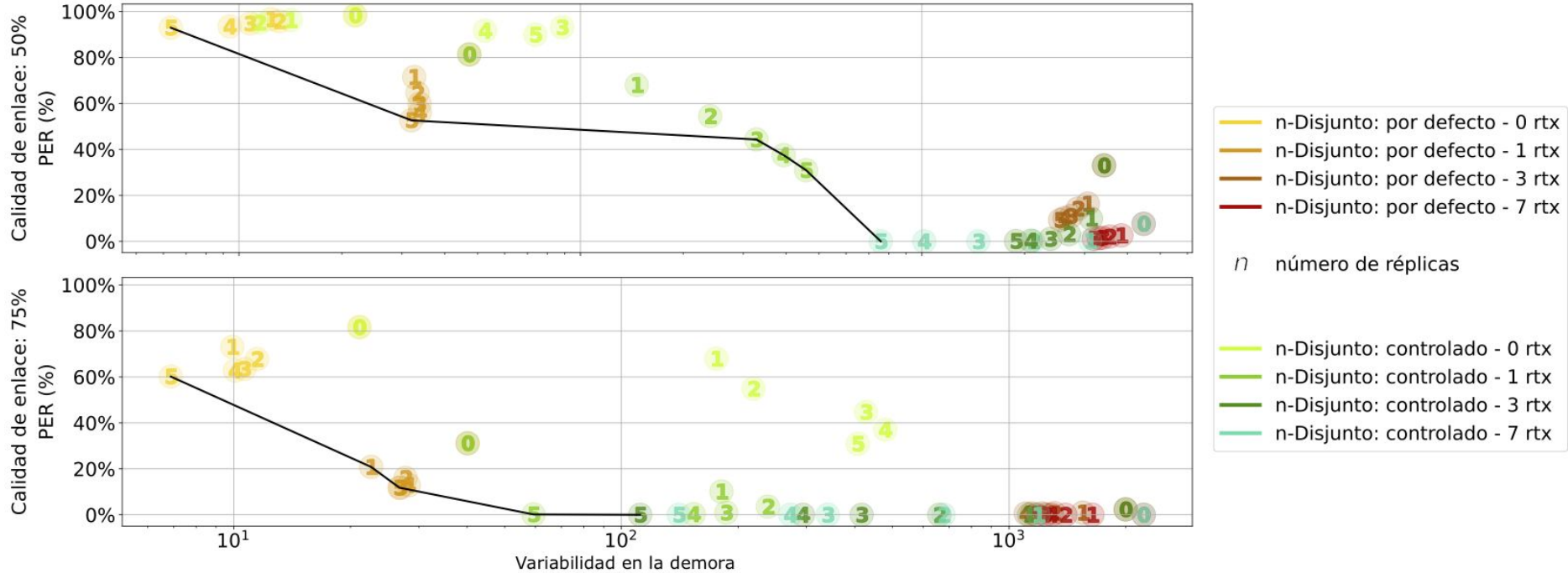
La cantidad de copias aumenta con la cantidad de RTX y réplicas, especialmente en la calidad de enlace más baja.

Comparación de los algoritmos

El *Packet Error Rate (PER)* mide el *porcentaje de paquetes distintos no recibidos exitosamente* frente a la cantidad total de paquetes distintos enviados.

En una red determinística y confiable, *esperamos un valor de PER bajo.*

Evaluación de performance



Conclusiones

Conclusiones

- Se abordó la necesidad de tráfico determinístico en redes industriales inalámbricas de dispositivos IoT.
- Se presentaron los algoritmos n-Disjoint.
- Se logró:
 - Una rendimiento de demora y de su variabilidad acotado (a costo de un uso mayor de energía).
 - Valores de PDR mayores al 99.96% para las calidades de enlace estudiadas.



Gracias!

