
Uso de middleware de distribución en sistemas particionados

Héctor Pérez (perezh@unican.es)
J. Javier Gutiérrez (gutierjj@unican.es)

Grupo de Computadores y Tiempo Real, Universidad de Cantabria
<http://www.ctr.unican.es>

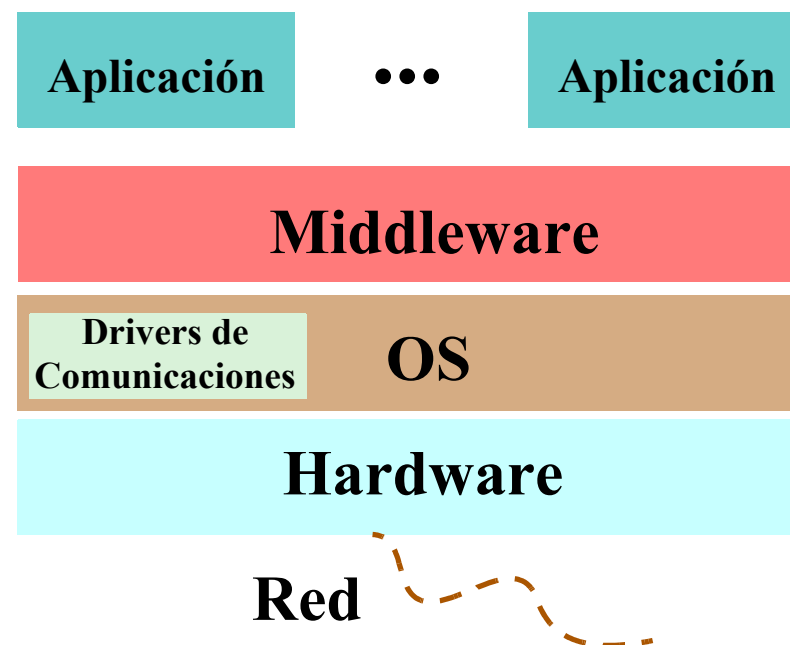
Financiado por el Gobierno de España y fondos FEDER
([TIN2011-28567-C03-02-HI-PARTES](#))

XVIII Jornada Técnica de Ada-Spain, Madrid, 30 de abril de 2013

Middleware de distribución

Capa de abstracción de sistema operativo, comunicaciones y hardware:

- Dependiente del lenguaje de programación
 - Ada (DSA): RPCs y distribución de objetos
 - Java RMI: distribución de objetos
- Independiente del lenguaje de programación
 - CORBA: distribución de objetos
 - DDS (Data Distribution Service): orientado a los datos



Middleware de distribución (cont.)

Generación de aplicaciones:

- **Ada:** configuración + categorización de partes de la aplicación
- **Java RMI:** configuración + uso de clases especiales
- **CORBA:** configuración + diseño de objetos CORBA que se agrupan en servicios y aplicaciones clientes que los usan
- **DDS:** configuración implícita (fase de descubrimiento) + diseño de datos sobre los que se puede leer/escribir

Los middleware que separan mejor la funcionalidad de la aplicación de la distribución (más peso en la configuración) son más fácilmente integrables en procesos MDE

Middleware de distribución y tiempo real



Restricciones en el uso de servicios y en aspectos dinámicos

Nuevos elementos de configuración: parámetros de planificación, plazos, QoS

Las implementaciones de middleware para tiempo real deben permitir

- **realizar una planificación adecuada (evitando inversiones de prioridad no acotadas, correcta gestión de threads)**
- **acceso a comunicaciones de tiempo real**
- **extraer un modelo compatible con el resto del sistema (aplicación, sistema operativo, red de comunicaciones ...)**
- **realizar el análisis de planificabilidad de todo el sistema**

Implementaciones de estándares evaluados

Ada DSA:

- **GLADE:** sobrecarga baja

CORBA:

- **PolyORB:** integra CORBA y DSA, sobrecarga muy alta
- **TAO:** sobrecarga media

DDS:

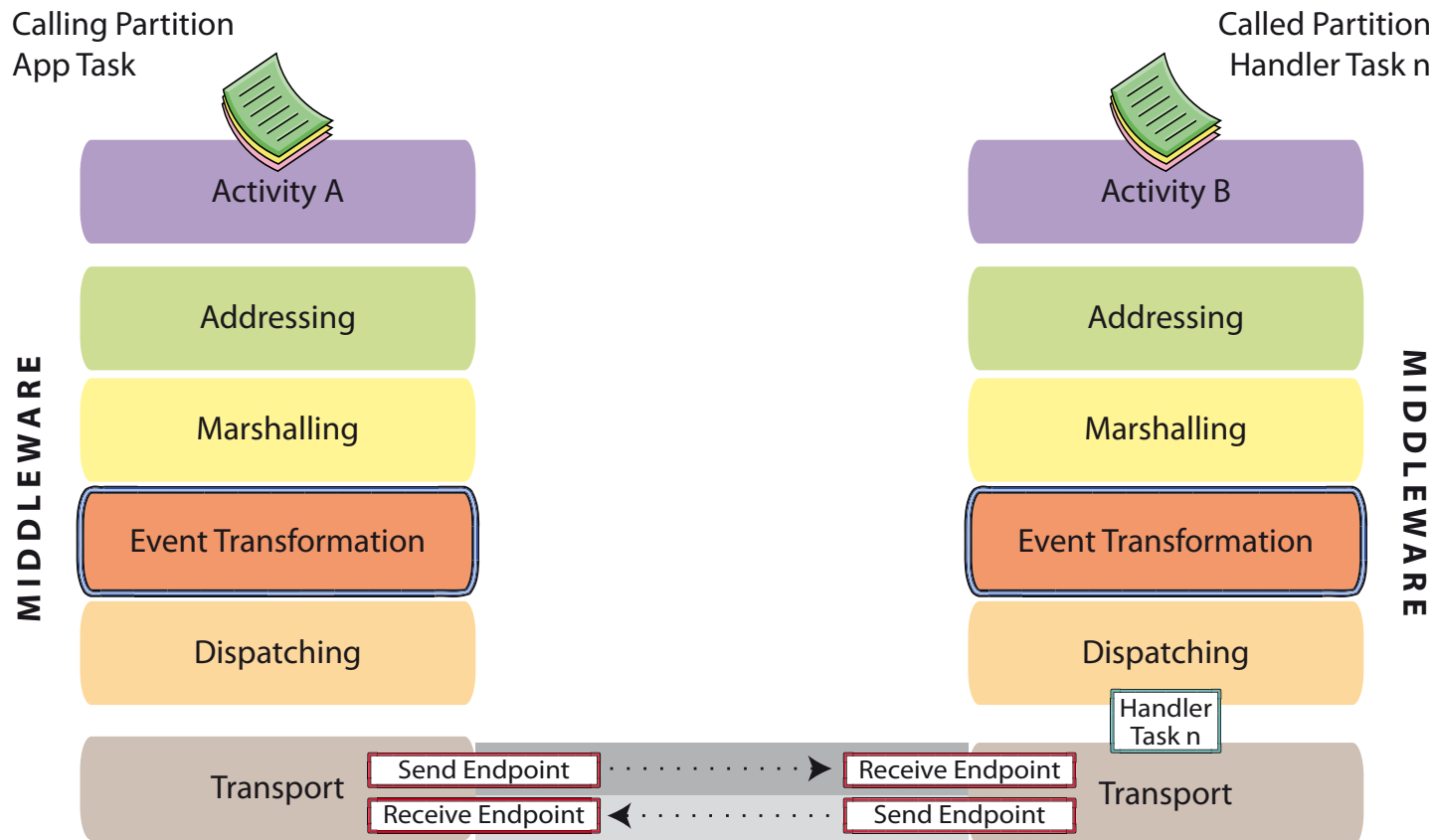
- **RTI:** sobrecarga muy baja

Todas presentan problemas para aplicaciones de tiempo real:

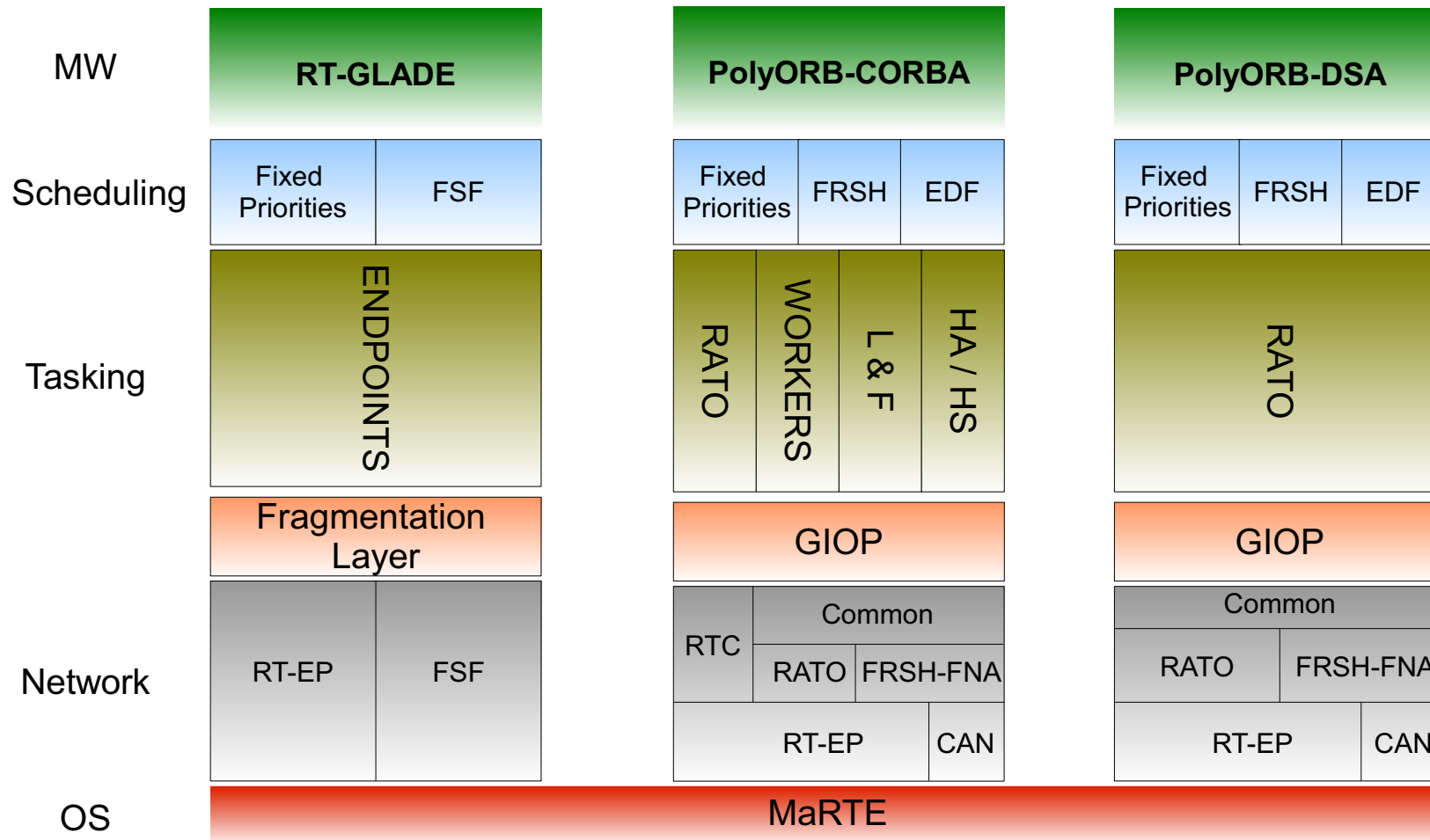
- por la especificación del estándar que implementan
- decisiones de la propia implementación

Solución para middleware de tiempo real: patrón de endpoints

End-to-end flows definidos en la fase configuración



Arquitecturas de middleware de tiempo real para Ada



Sistemas particionados

Utilizados para sistemas de alta integridad (ej. ARINC 653):

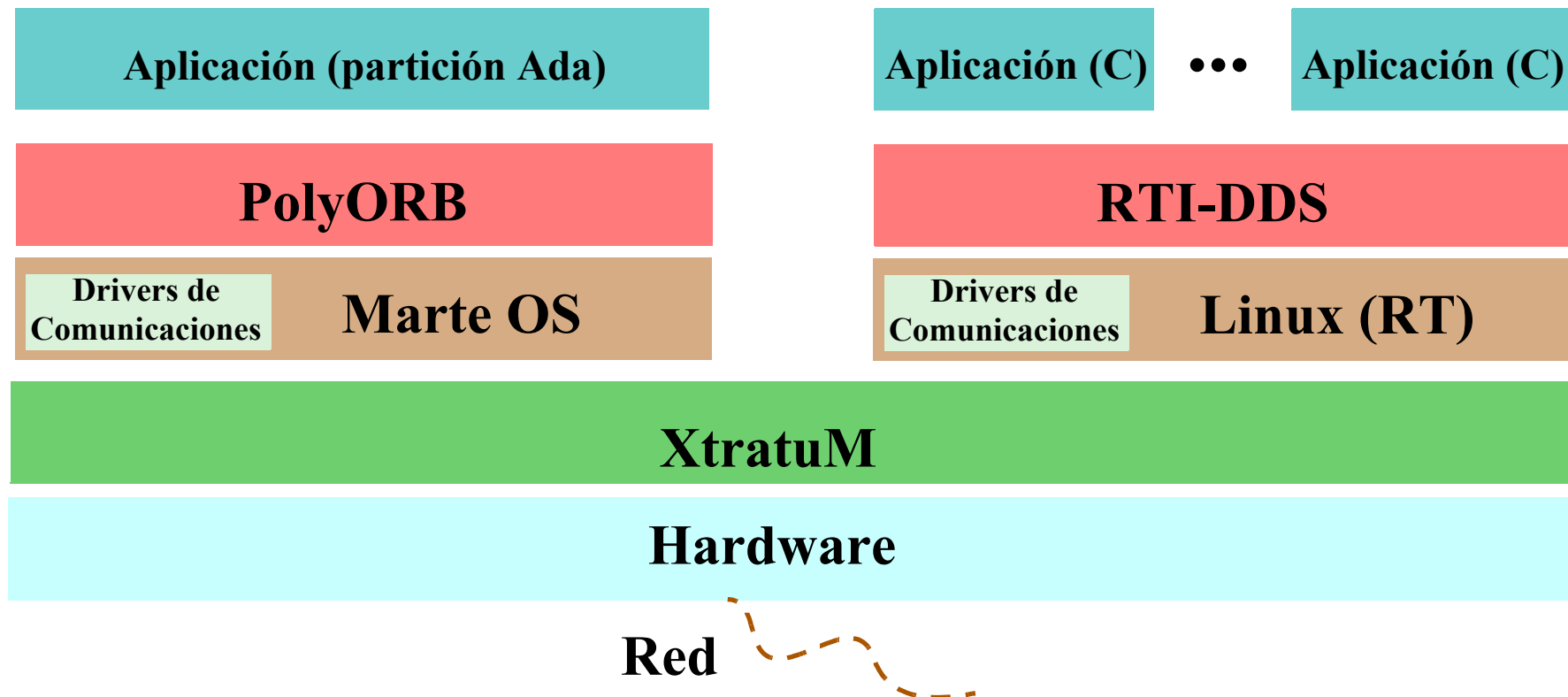
- aislamiento temporal y espacial de particiones
- reticentes a la complicación del software por la certificación
- encaja con el concepto de partición definido en Ada

El particionamiento lo puede soportar el sistema operativo o se puede basar en el uso de un hipervisor

- permite la ejecución de particiones con diferentes OS
- las particiones pueden tener diferente nivel de criticidad
- se pueden considerar sistemas multicore como distribuidos
 - las particiones se asignan estáticamente a un núcleo con parámetros de planificación fijos

Experiencias de uso de middleware para sistemas particionados

El aspecto más relevante es la gestión de las comunicaciones



Servicios de comunicaciones de XtratuM

Virtualización de I/O de XtratuM (XMIO)

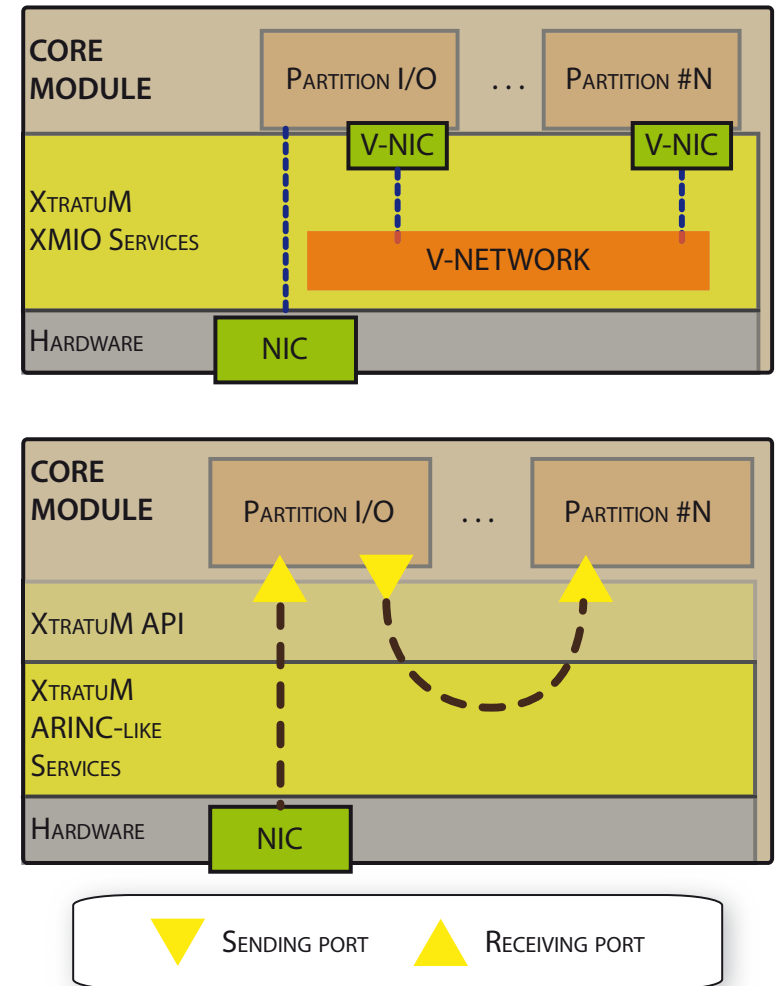
- para particiones Linux

Puertos de comunicaciones ARINC

- configuración off-line
- puertos *sampling* y *queuing*
- I/O no bloqueante y unidireccional
- un puerto de recepción puede tener una sola fuente

Las particiones gestionan los dispositivos

- configuración off-line



Soluciones para las particiones de I/O



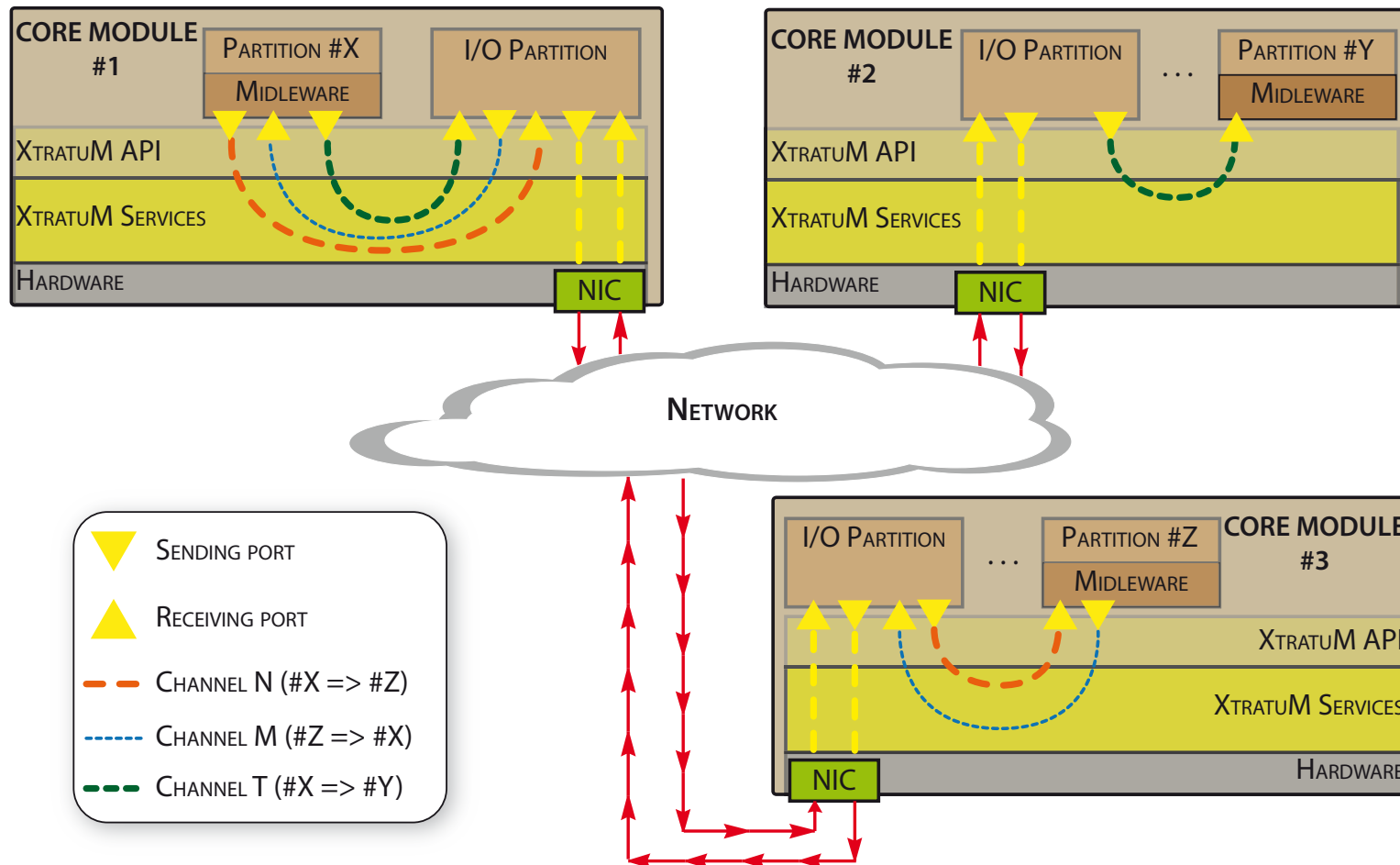
PolyORB sobre MaRTE OS/XtratuM

- soporta el patrón de *endpoints*
- partición de I/O para el acceso a las comunicaciones
 - se puede montar directamente sobre los puertos ARINC
 - o como un servicio que usa middleware
- las demás particiones se comunican con la de I/O por medio de puertos ARINC

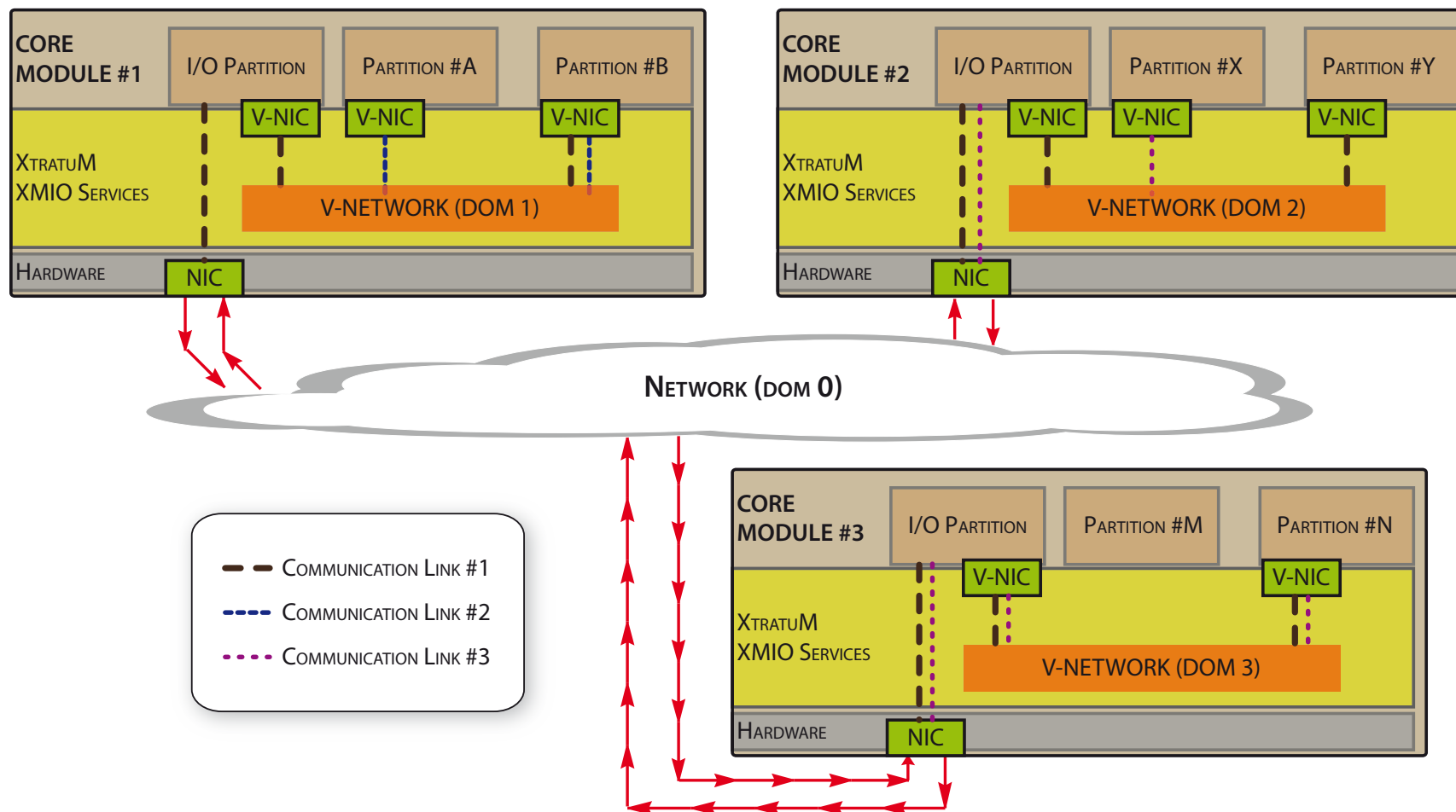
RTI-DDS sobre Linux (RT)/XtratuM

- partición de I/O para el acceso a las comunicaciones
- las demás particiones se comunican con la de I/O por medio de Virtio (virtualización de las comunicaciones)

PolyORB para MaRTE OS/Xtratum



RTI-DDS para Linux(RT)/XtratuM



Algunas medidas de tiempos

- Para PolyORB/MaRTE OS/XtratuM,

	MAX (μ S)	MED (μ S)	MIN (μ S)
RED	321	305	298
SIN PARTICIONADO	5728	5691	5689
PARTICIONADO	7944	7334	7130

- incremento de sobrecarga 29% en promedio (25% debido a la configuración de particiones)

- Para RTI-DDS/Linux (RT)/XtratuM

	MAX (μ S)	MED (μ S)	MIN (μ S)
RED	262	206	154
SIN PARTICIONADO	415	286	218
PARTICIONADO	4876	2159	1071

- sobrecarga debida a la configuración y uso de middleware en I/O

Es posible usar middleware de distribución en sistemas particionados:

- **aprovechar el nivel de abstracción en el desarrollo de aplicaciones**
- **inicialmente en aplicaciones menos críticas pero con requisitos de tiempo real o QoS**

El uso de un hipervisor que gestione multicore aún abre más posibilidades al uso de middleware de distribución:

- **uso de una sola partición (incluyendo la I/O) por núcleo**
- **habría que resolver el acceso a las comunicaciones desde diferentes particiones/núcleos**