Universidad Nacional de Córdoba

Congestión y Escalabilidad en Redes Espaciales

IEEE ComSoc
IEEE Communications Society

Dr. Ing. Pablo G. Madoery

CONICET

IDIT
CONICET

UNC LCD

Laboratorio de
Comunicaciones
Digitales

Universidad
Nacional de
Córdoba

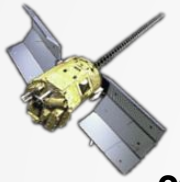
Laboratorio de Comunicaciones Digitales

Temario

- **Contexto**
  - *Arquitectura Segmentada y Constelaciones*
- **Problema**
- ***Redes DTN***
- **Modelado**
- **Contribuciones**
  - *Congestión: LPA-CGR, GPA-CGR, EGPA-CGR*
  - *Escalabilidad: GRK, LRK*
- **Conclusiones**

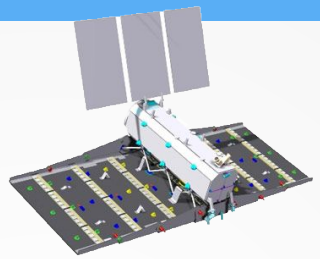
Contexto

1998
SAC-A
68Kg



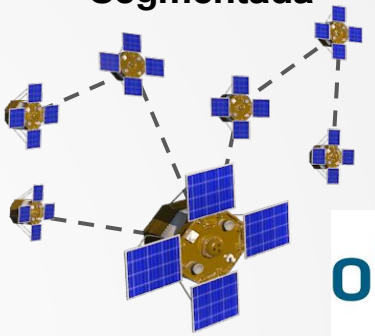
2000
SAC-C
485Kg

2011
SAC-D
1600Kg



2018 & 2020
SAOCOM
3000Kg

Arq. Segmentada



STARLINK

OneWeb

SATELLOGIC

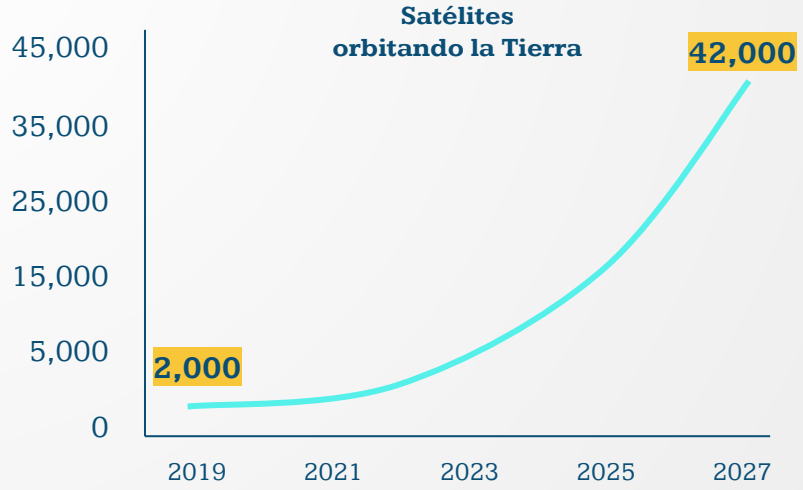
Skyloom

1996
SAC-B
190Kg

1996
Victor 1
30Kg

1990
Lusat1
10Kg

INVAP SI VENG CONAE



Problema

Problema

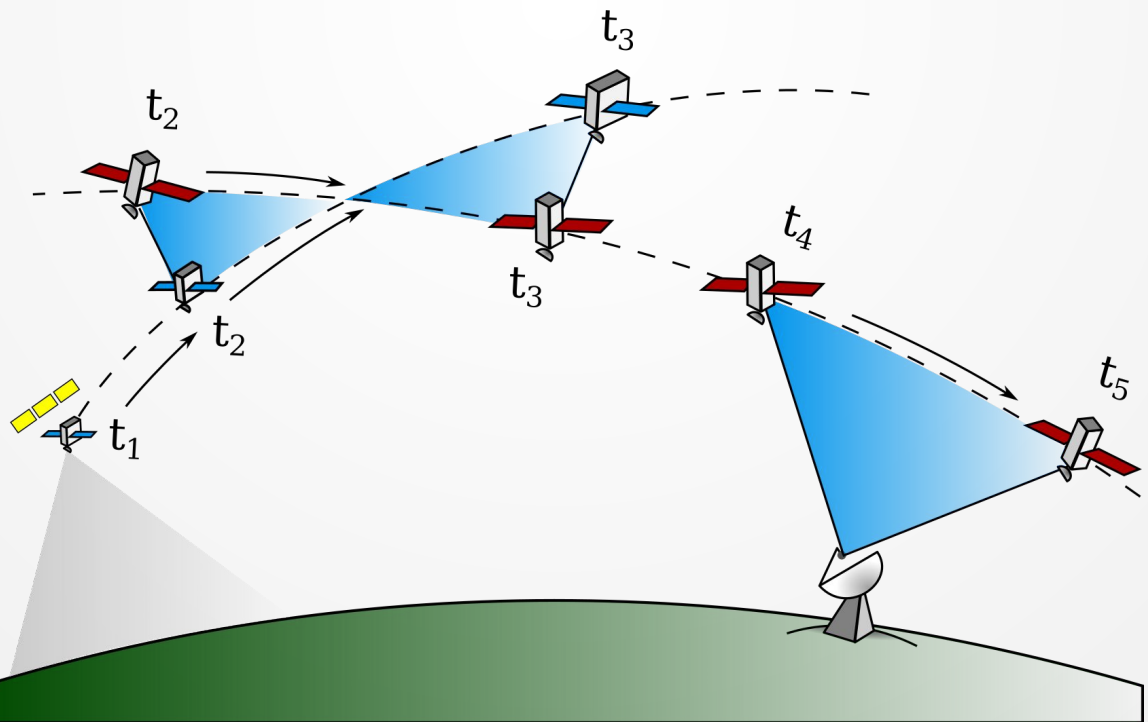
Utilizando esquemas de comunicación tradicionales los satélites descargan el **1%** de datos de lo que pueden generar

Hipótesis

Utilizando un esquema **DTN** es posible aumentar la capacidad de transporte de datos de las nuevas redes satelitales

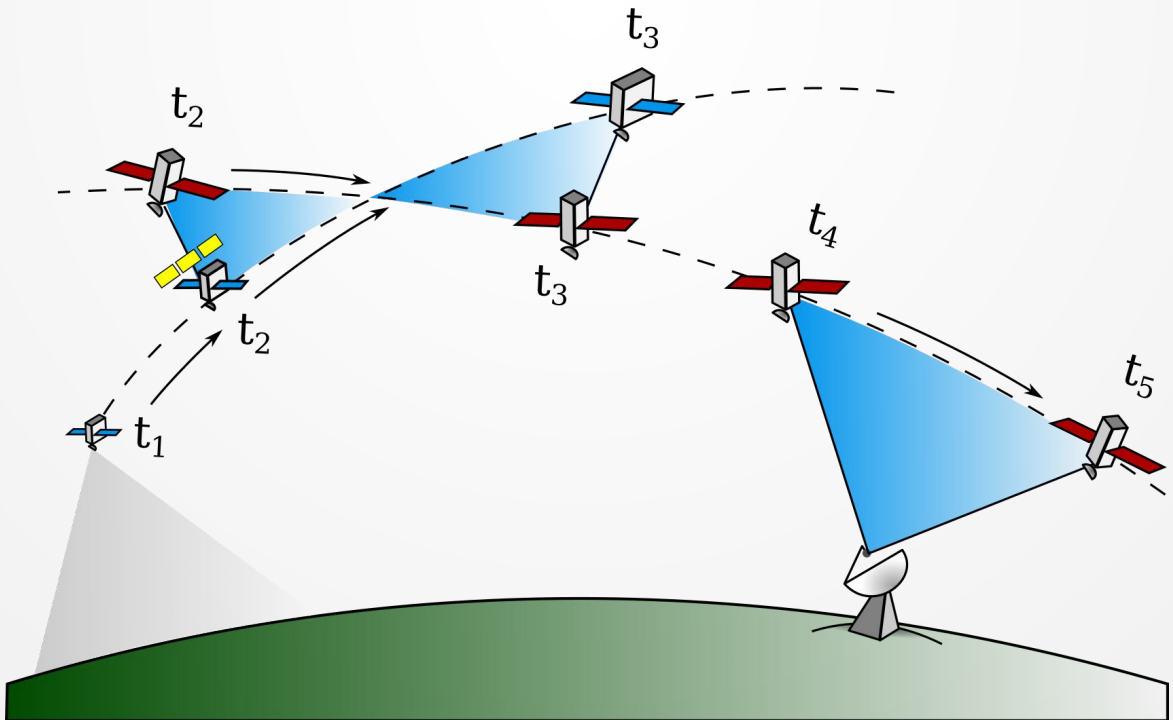
DTN: Delay Tolerant Networks

- En las DTN espaciales los nodos tienen trayectorias **predecibles**.
- Utilizan el principio ***store-carry-forward*** para enviar datos siguiendo rutas temporales.



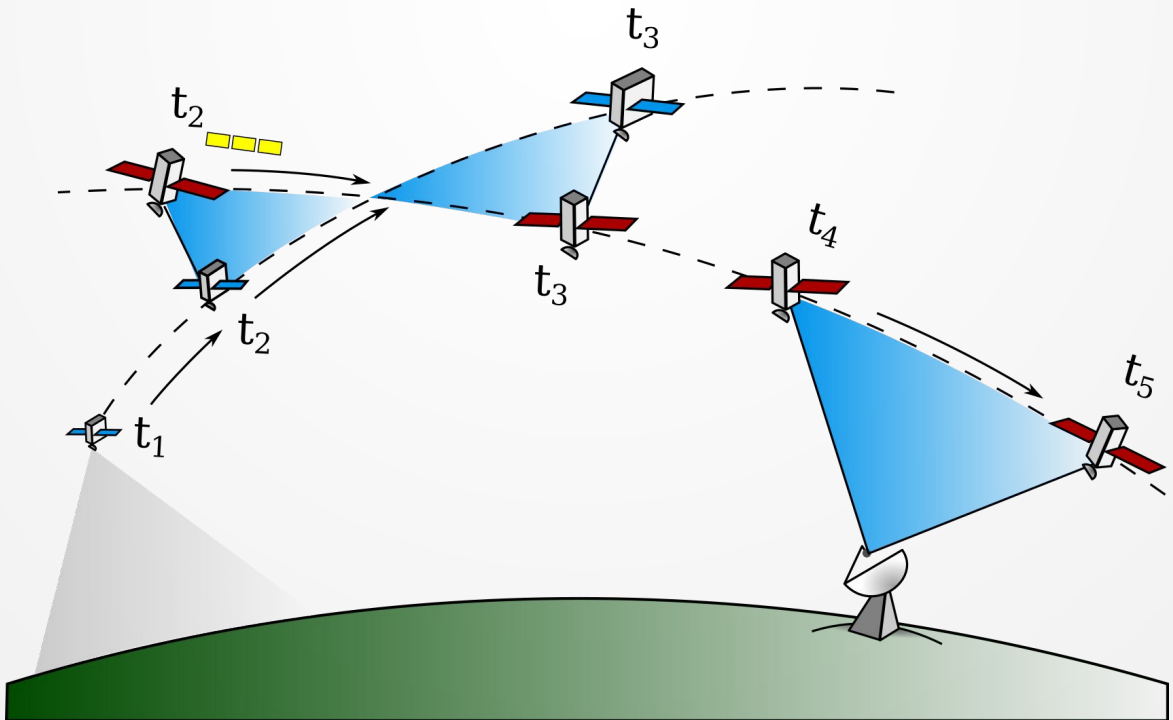
DTN: Delay Tolerant Networks

- En las DTN espaciales los nodos tienen trayectorias **predecibles**.
- Utilizan el principio ***store-carry-forward*** para enviar datos siguiendo rutas temporales.



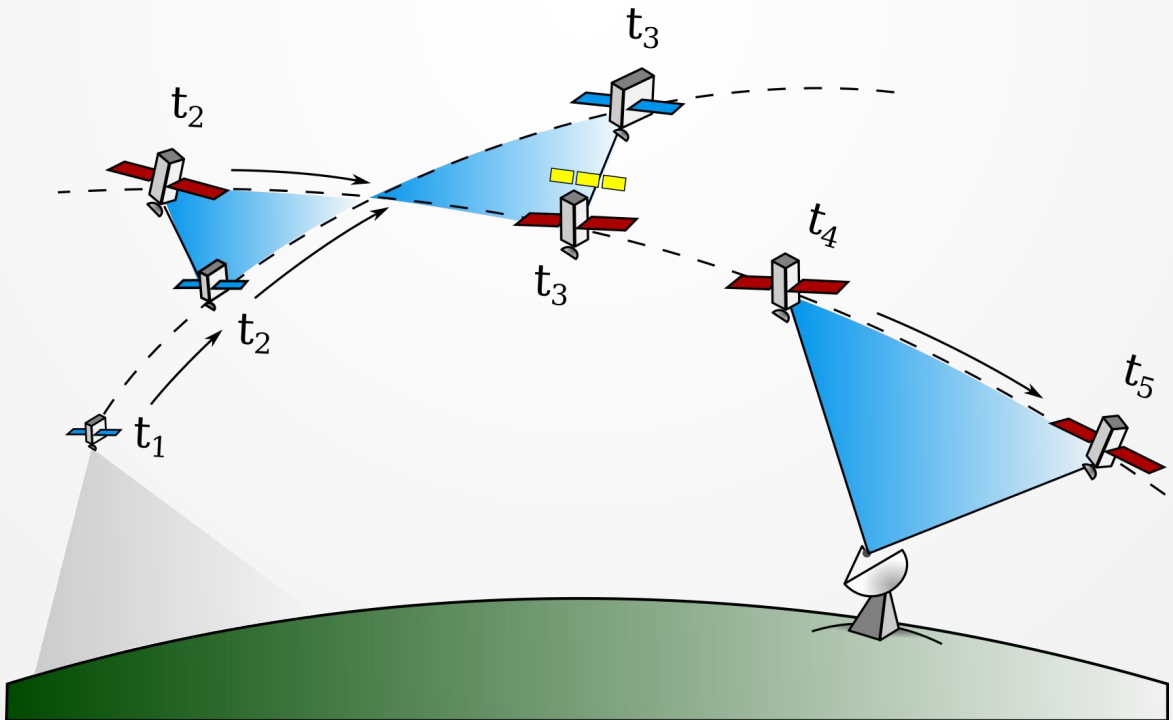
DTN: Delay Tolerant Networks

- En las DTN espaciales los nodos tienen trayectorias **predecibles**.
- Utilizan el principio ***store-carry-forward*** para enviar datos siguiendo rutas temporales.



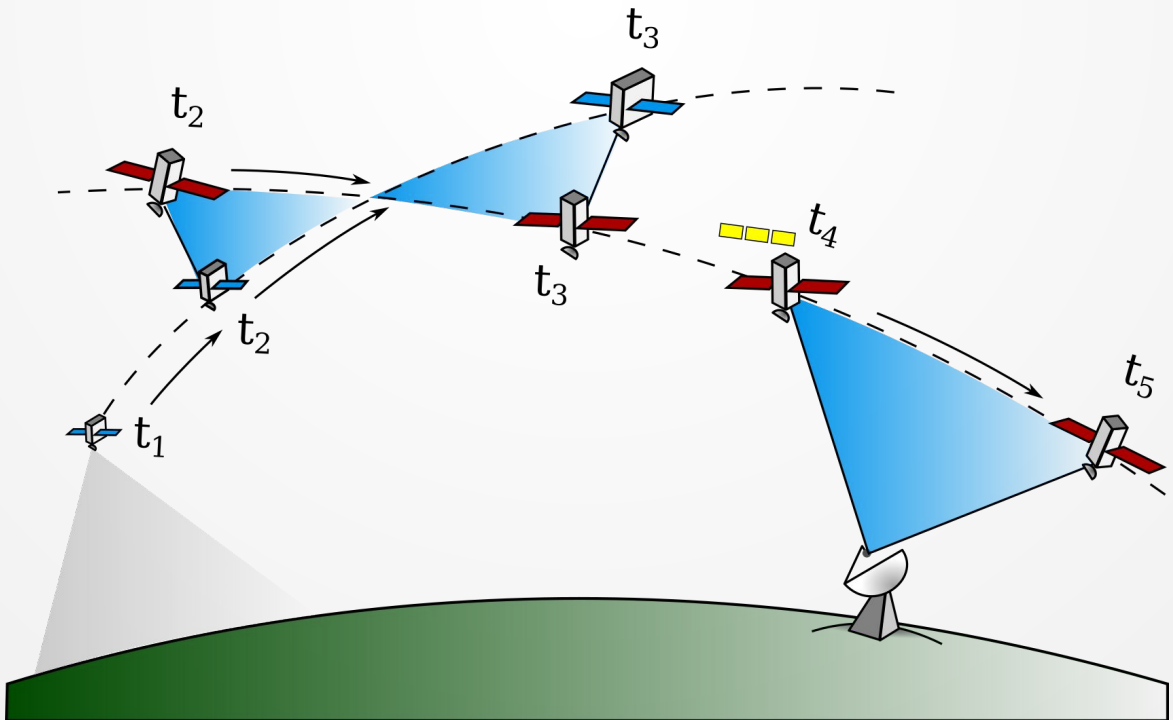
DTN: Delay Tolerant Networks

- En las DTN espaciales los nodos tienen trayectorias **predecibles**.
- Utilizan el principio ***store-carry-forward*** para enviar datos siguiendo rutas temporales.



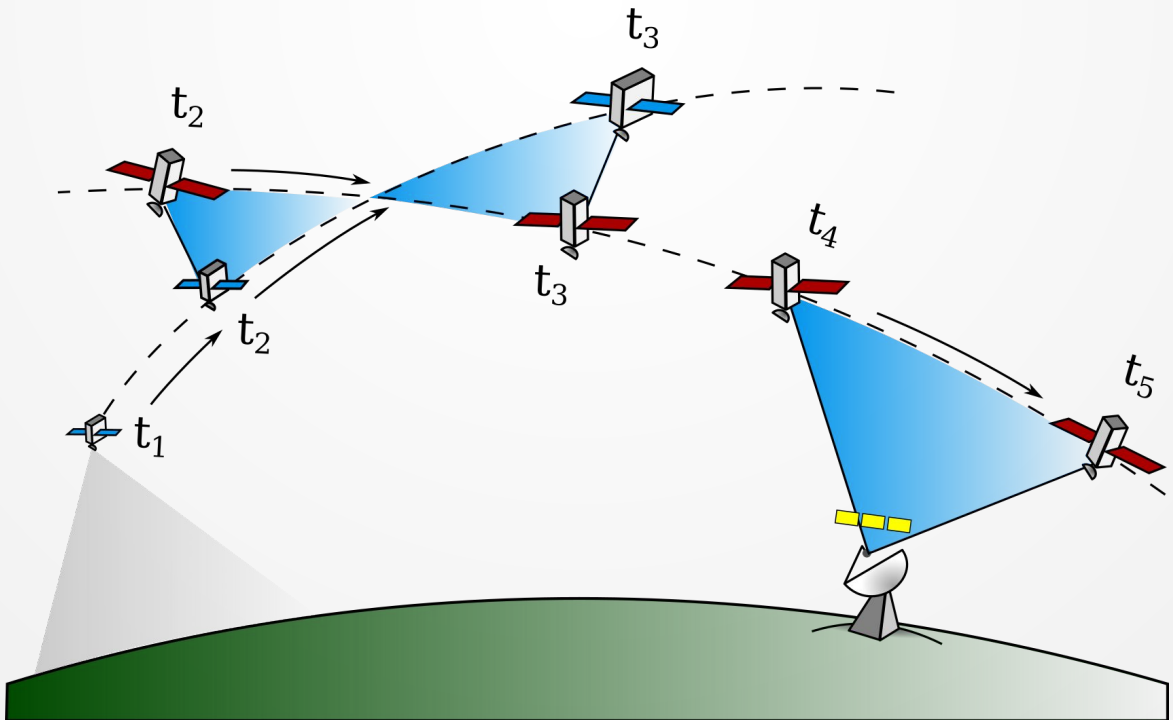
DTN: Delay Tolerant Networks

- En las DTN espaciales los nodos tienen trayectorias **predecibles**.
- Utilizan el principio ***store-carry-forward*** para enviar datos siguiendo rutas temporales.



DTN: Delay Tolerant Networks

- En las DTN espaciales los nodos tienen trayectorias **predecibles**.
- Utilizan el principio ***store-carry-forward*** para enviar datos siguiendo rutas temporales.



DTN: Delay Tolerant Networks

- En las DTN espaciales los nodos tienen trayectorias **predecibles**.
- Utilizan el principio ***store-carry-forward*** para enviar datos siguiendo rutas temporales.

Ruta

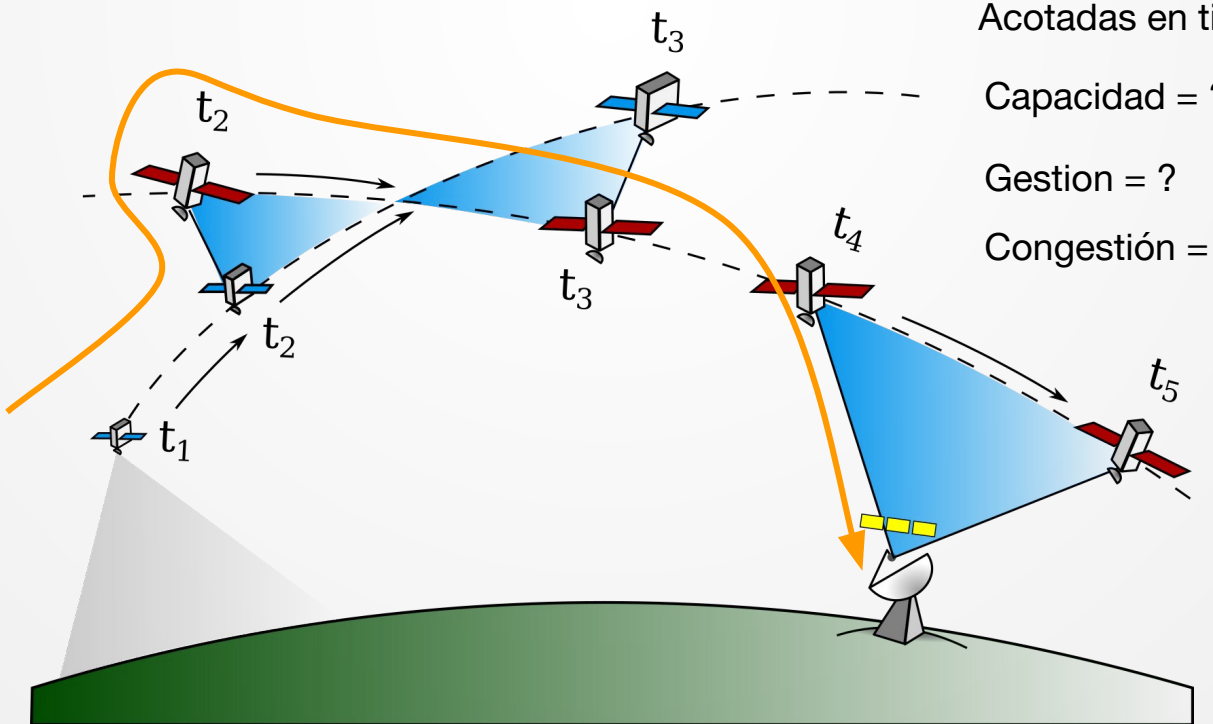
Secuencia de contactos

Acotadas en tiempo

Capacidad = ?

Gestion = ?

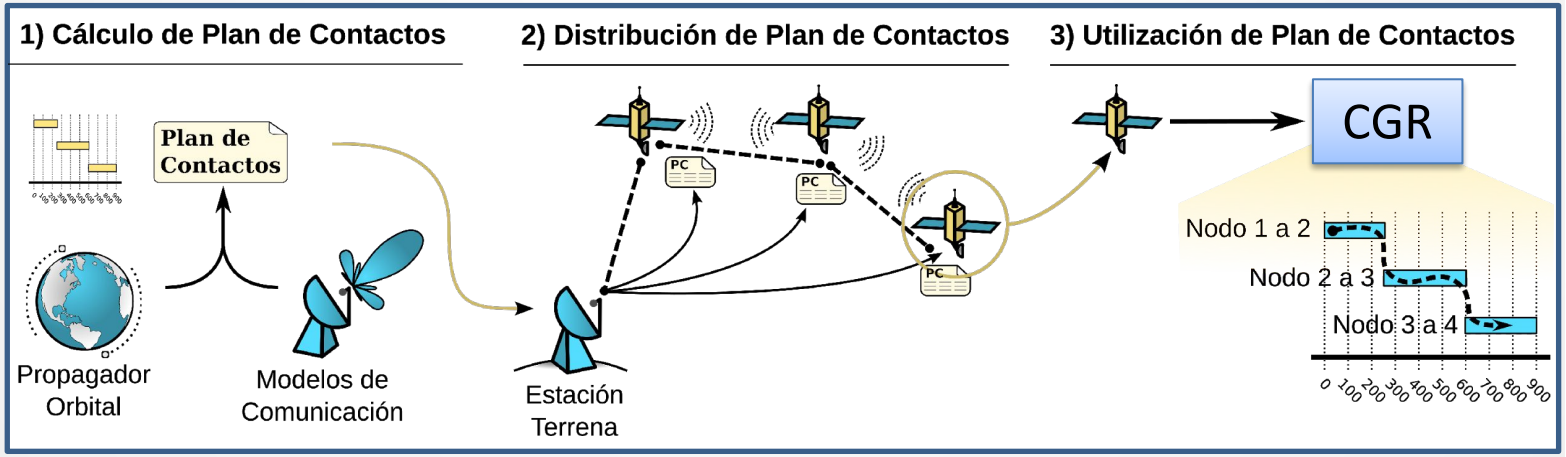
Congestión = ?



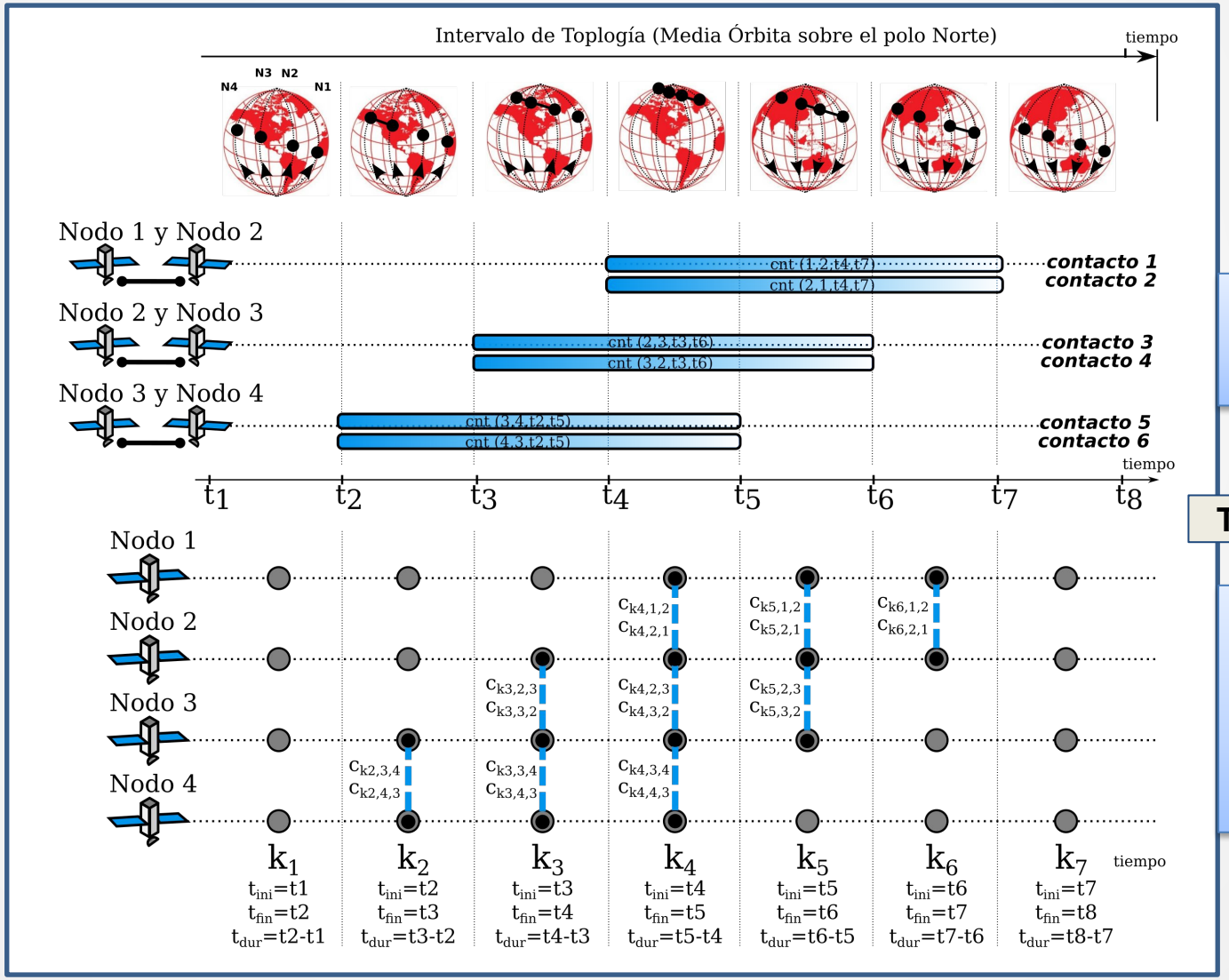
Plan de Contactos

- Combinando la dinámica orbital con modelos de comunicaciones se puede determinar un **Plan de Contactos**

- Este plan puede **distribuirse** anticipadamente para que los nodos tomen decisiones de enrutamiento de tráfico



Modelado



Lista de Contactos

Tiempo

Grafos evolutivos en el tiempo

Hipótesis

«Es posible mejorar el rendimiento de las comunicaciones disruptivas

mediante técnicas que actúen sobre

la planificación de los contactos

y/o el algoritmo de enrutamiento»

Contribuciones

Congestión

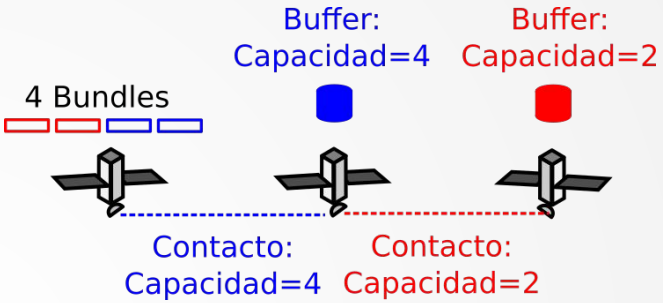
- **P. Madoery**, J. Fraire, and J. Finochietto, "*Congestion management techniques for disruption-tolerant satellite networks*," **International Journal of Satellite Communications and Networking**, May 2017.
- J. Fraire, **P. Madoery**, J. Finochietto, "Traffic-Aware Contact Plan Design for Disruption-Tolerant Space Sensor Networks," **Ad Hoc Networks**, April 2016.
- **P. Madoery**, J. Fraire, and J. Finochietto, "*Analysis of communication strategies for earth observation satellite constellations*," **IEEE Latin America Transactions**, June 2016.

Escalabilidad

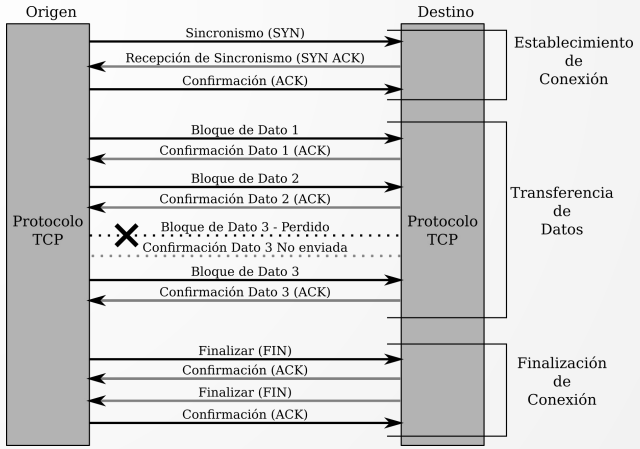
- **P. Madoery**, J. Fraire, F. Raverta, J. Finochietto, and S. Burleigh, "*Managing Routing Scalability in Space DTNs*," In **5th IEEE International Workshop on Space-Terrestrial Internetworking (STINT)**, Huntsville, AL, USA, 2018.
- J. Fraire, **P. Madoery**, Amir Charif, J. Finochietto, "On route table computation strategies in Delay-Tolerant Satellite Networks," **Ad Hoc Networks**, November 2018.
- **P. Madoery**, C. Zerbini, J. Fraire, and J. Finochietto, "*Enhancing contact graph routing forwarding performance for segmented satellites architectures*," In **1st IAA Latin American Symposium on Small Satellites: Advanced Technologies and Distributed Systems**, Buenos Aires, Argentina, March 2017.

Congestión

*La **Congestión** se produce cuando a la red, o a parte de ella, se le ofrece mas tráfico del que puede cursar*

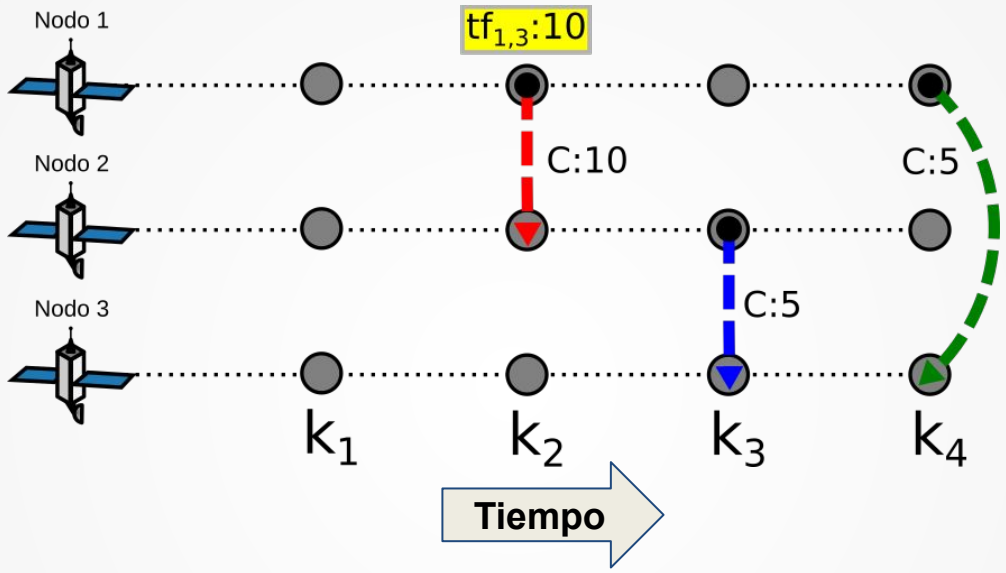


*En **Internet**, esto se soluciona con una constante realimentación (TCP) imposible de garantizar En DTN*

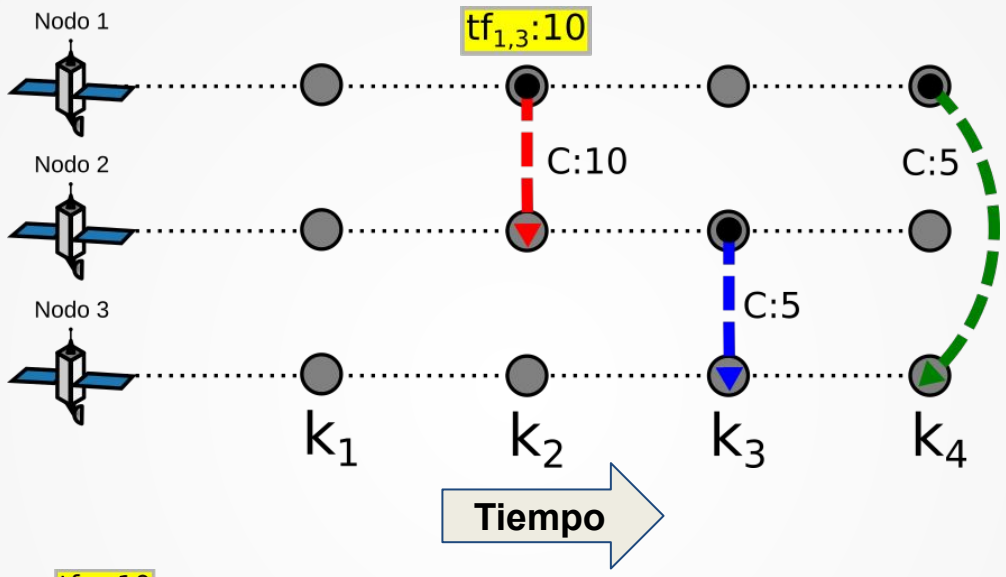


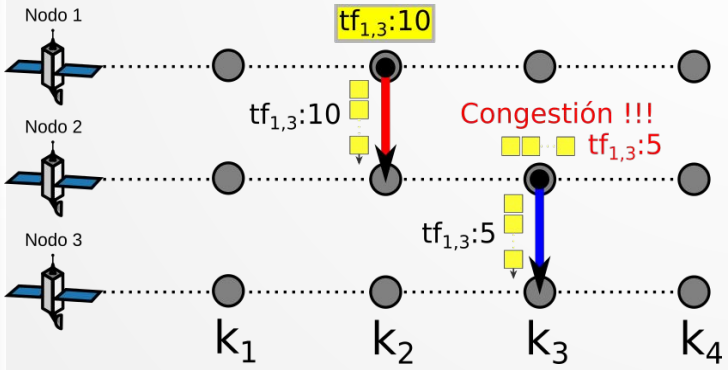
*Actualmente, **CGR** aporta cierto grado de mitigación de la congestión local, que resulta insuficiente en muchos casos*

Congestión Local



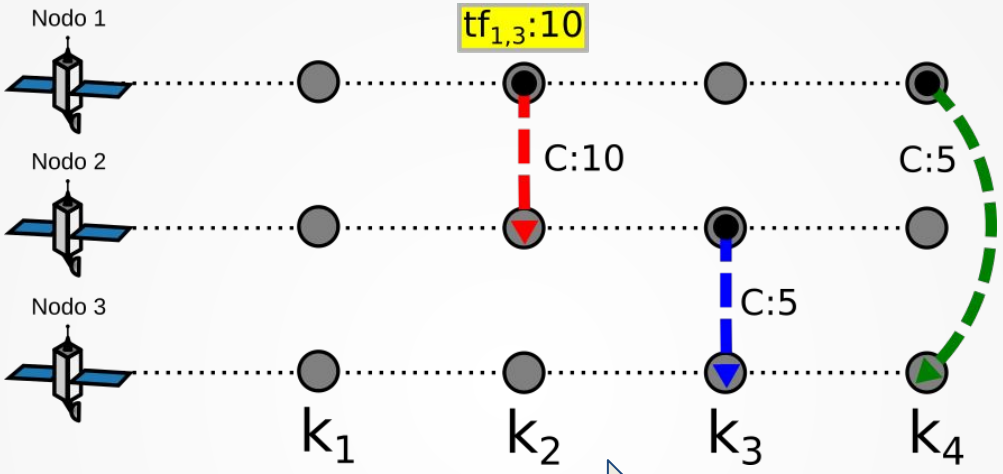
Congestión Local



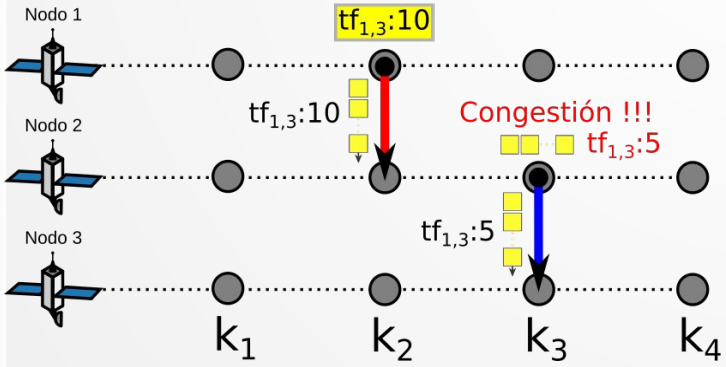


Flujo con Congestión

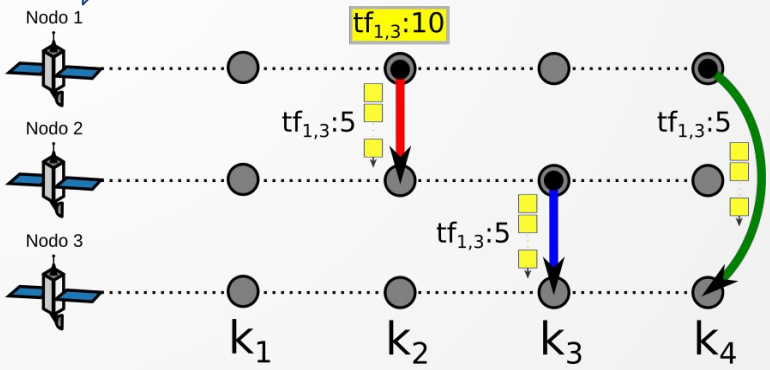
Congestión Local



Tiempo



Flujo con Congestión



Flujo sin Congestión

Local Path Aware CGR (LPA-CGR)

Algoritmo 4: LPA-CGR Forward

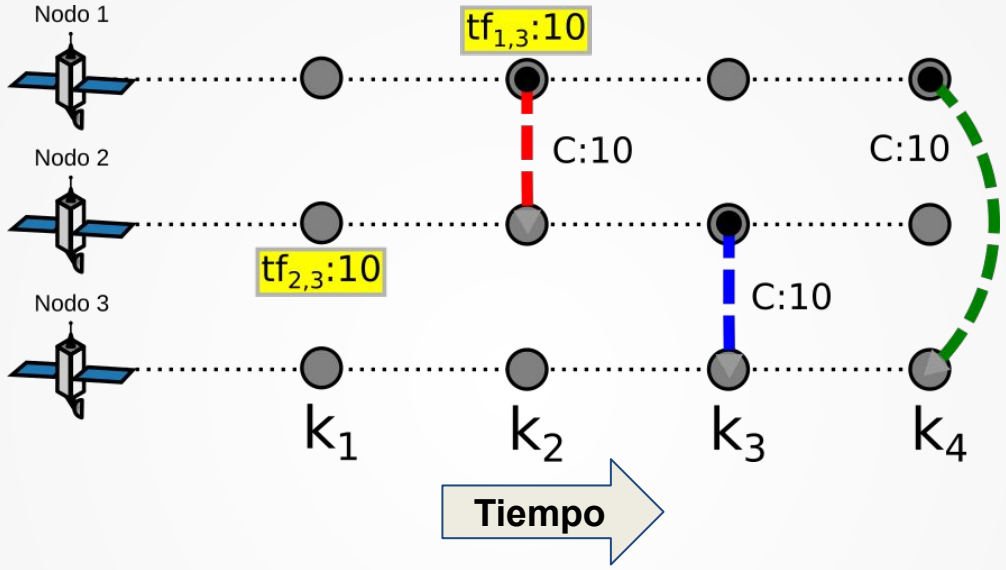
```
input  : bundle to forward B, contact plan Cp, route list Rl, excluded nodes En
output: bundle B is enqueued in the corresponding queue
global : BufferCap
1  if Cp changed since last Rl calculation then
2  |  Rl ⟵ ∅;
3  if B forbids return to sender then
4  |  En ⟵ B sender node;
5  Pn ⟵ identifyProxNodes (B, Cp, Rl, En);
6  if B is critical then
7  |  enqueue a copy of B to each node in Pn;
8  |  updateBuffersResidualCapacity;
9  |  decreaseContactsResidualCapacity;
10 |  decreaseRoutesResidualCapacity;
11 |  return
12 set nextHop to empty;
13 for pn ∈ Pn do
14 |  if nextHop is empty then
15 |  |  nextHop = pn
16 |  else if pn.arrivalTime< nextHop.arrivalTime then
17 |  |  nextHop = pn
18 |  else if pn.arrivalTime> nextHop.arrivalTime then
19 |  |  continue
20 |  else if pn.hops< nextHop.hops then
21 |  |  nextHop = pn
22 |  else if pn.hops> nextHop.hops then
23 |  |  continue
24 |  else if pn.id< nextHop.id then
25 |  |  nextHop = pn
26 if nextHop is not empty then
27 |  enqueue B to nextHop;
28 |  updateBuffersResidualCapacity;
29 |  decreaseContactsResidualCapacity;
30 |  decreaseRoutesResidualCapacity;
31 |  manageOverbook (B,nextHop)
32 else
33 |  enqueue B to limbo
34 return
```

Algoritmo 5: LPA-CGR Identify Proximate Node List

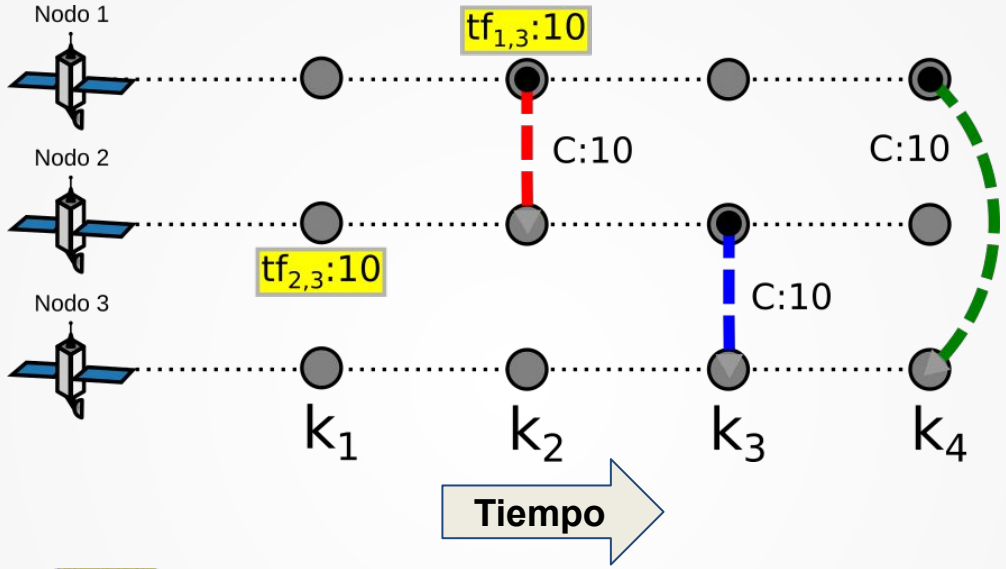
```
input  : bundle to forward B, contact plan Cp, route list Rl, excluded nodes En,
output: proximate nodes list Pn
1  if Rl is empty then
2  |  Rl ⟵ loadRouteList (B,Cp);
3  Pn ⟵ ∅;
4  for route ∈ Rl do
5  |  if route.toTime <= currentTime then
6  |  |  continue (ignore past route)
7  |  if route.arrivalTime >= B.deadline then
8  |  |  continue (route arrives late)
9  |  if route.residualCapacity < B.bitLenght then
10 |  |  continue (not enough capacity)
11 |  if route.nextHop ∈ En then
12 |  |  continue (next hop is excluded)
13 |  for pn ∈ Pn do
14 |  |  if pn = route.nextHop then
15 |  |  |  if pn.arrTime> route.arrTime then
16 |  |  |  |  replace pn with route.nextHop
17 |  |  |  else if pn.arrTime< route.arrTime then
18 |  |  |  |  continue (previous route was better)
19 |  |  |  else if pn.hops> route.hops then
20 |  |  |  |  replace pn with route.nextHop
21 |  |  |  else if pn.hops< route.hops then
22 |  |  |  |  continue (previous route was better)
23 |  |  |  break
24 |  if route.nextHop ∉ Pn then
25 |  |  pn ⟵ route.nextHop;
26 |  |  Pn ⟵ pn;
27 return Pn
```

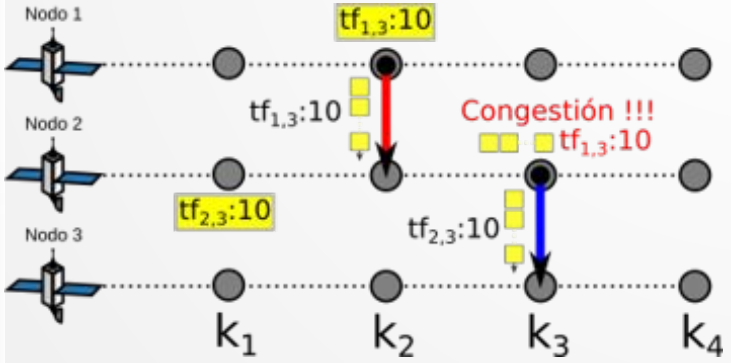
LPA-CGR considera la capacidad residual de la ruta completa

Congestión Global



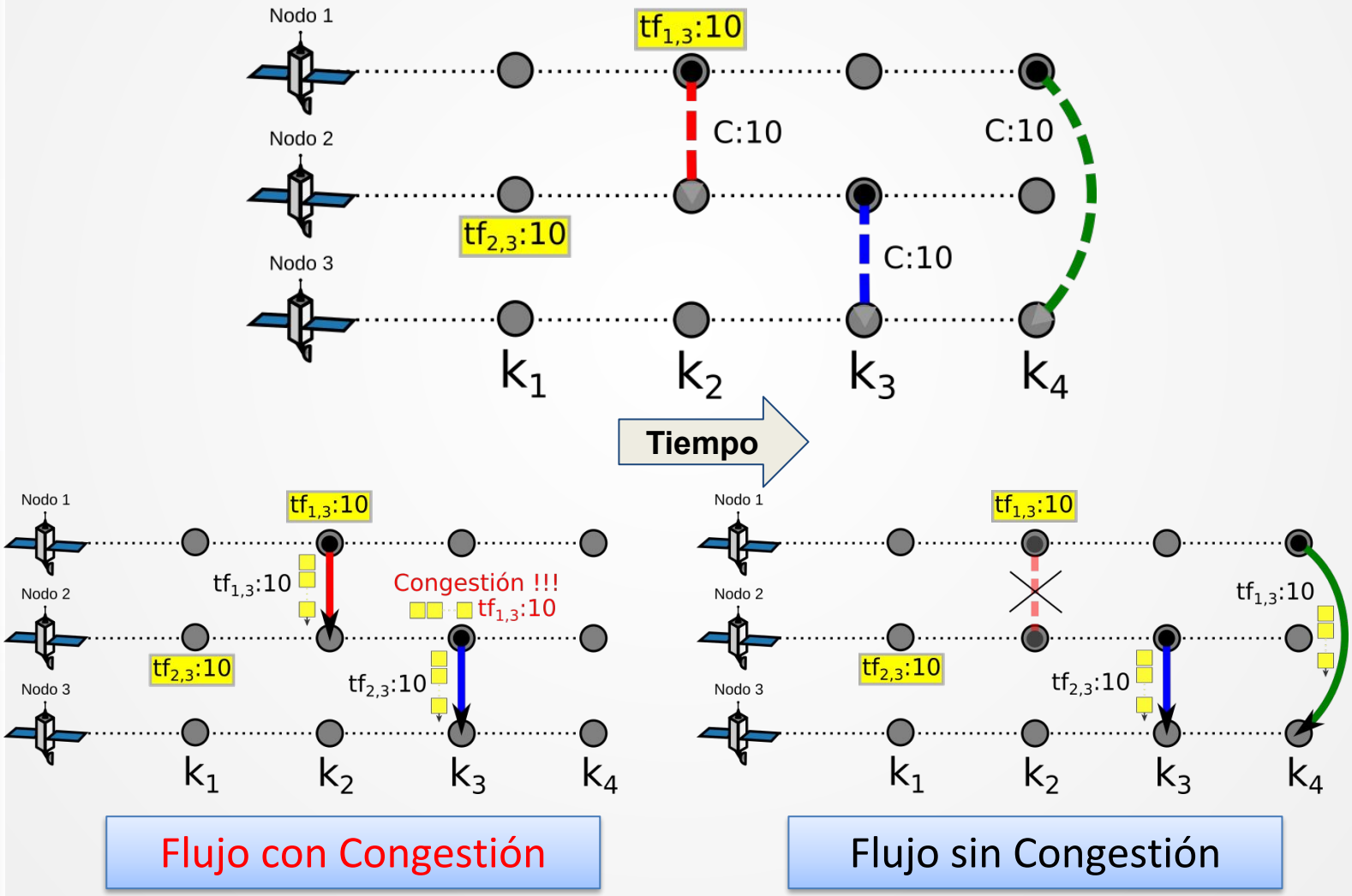
Congestión Global





Flujo con Congestión

Congestión Global



Flujo de Tráfico Óptimo

contactos

Modelo LP

tráfico

$$\text{minimizar:} \quad \sum_{k_q \in K} \sum_{e \in E_{k_q}} \sum_{y \in V} \sum_{z \in V} w(k_q) * X_e^{y,z} \tag{3.1}$$

Sujeto a:

$$B_{t_q,v}^{y,z} = \begin{cases} B_{t_{q-1},v}^{y,z} + \sum_{e \in I^v} X_e^{y,z} - \sum_{e \in O^v} X_e^{y,z} + d_{t_q}^{y,z} & \text{if } y = v \\ B_{t_{q-1},v}^{y,z} + \sum_{e \in I^v} X_e^{y,z} - \sum_{e \in O^v} X_e^{y,z} & \text{if } y \neq v \end{cases} \quad \forall\, t_q, v, y, z \tag{3.2}$$

$$\sum_{y \in V} \sum_{z \in V} B_{t_q,v}^{y,z} <= b_v \quad \forall\, t_q, v \tag{3.3}$$

$$\sum_{y \in V} \sum_{z \in V} X_e^{y,z} <= c_e \quad \forall\, k_q, e \tag{3.4}$$

$$B_{t_0,v}^{y,z} = \begin{cases} d_{t0}^{y,z} & \text{if } y = v \\ 0 & \text{if } y \neq v \end{cases} \quad \forall\, v, y, z \tag{3.5}$$

$$B_{t_q,v}^{y,z} >= \begin{cases} d_{t_q}^{y,z} & \text{if } y = v \\ 0 & \text{if } y \neq v \end{cases} \quad \forall\, t_q > t_0, v, y, z \tag{3.6}$$

$$B_{t_f,v}^{y,z} = \begin{cases} \sum_{t_q \in T} \sum_{y \in V} d_{t_q}^{y,z} & \text{if } z = v \\ 0 & \text{if } z \neq v \end{cases} \quad \forall\, v, y, z \tag{3.7}$$

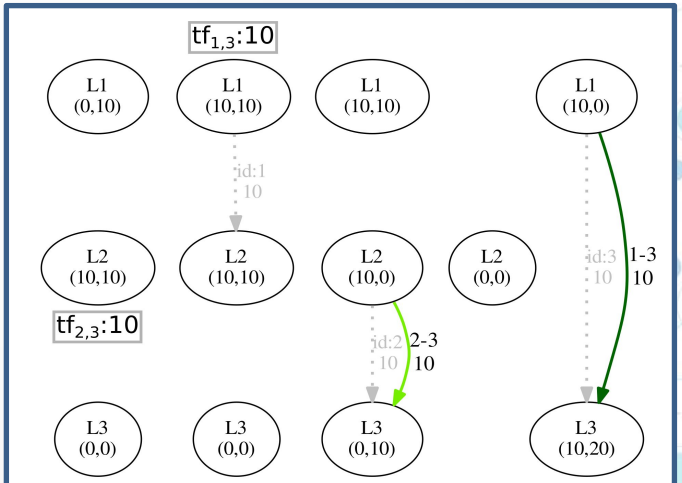
Coeficientes

Coeficientes de Entrada

$t_q \in T$	Instantes de tiempo
$k_q = [t_{q-1}, t_q] \in K$	Estados / Intervalos de tiempo
$v \in V$	Nodos de la red
$e \in E_{k_q}$	Arcos de un grafo en el estado k_q
c_e	Capacidad del arco e en el estado k_q correspondiente
b_v	Capacidad de almacenamiento del nodo v
$d_{t_q}^{y,z} \in D$	Tráfico desde el nodo y al nodo z originado en el tiempo t_q

Variables de Salida

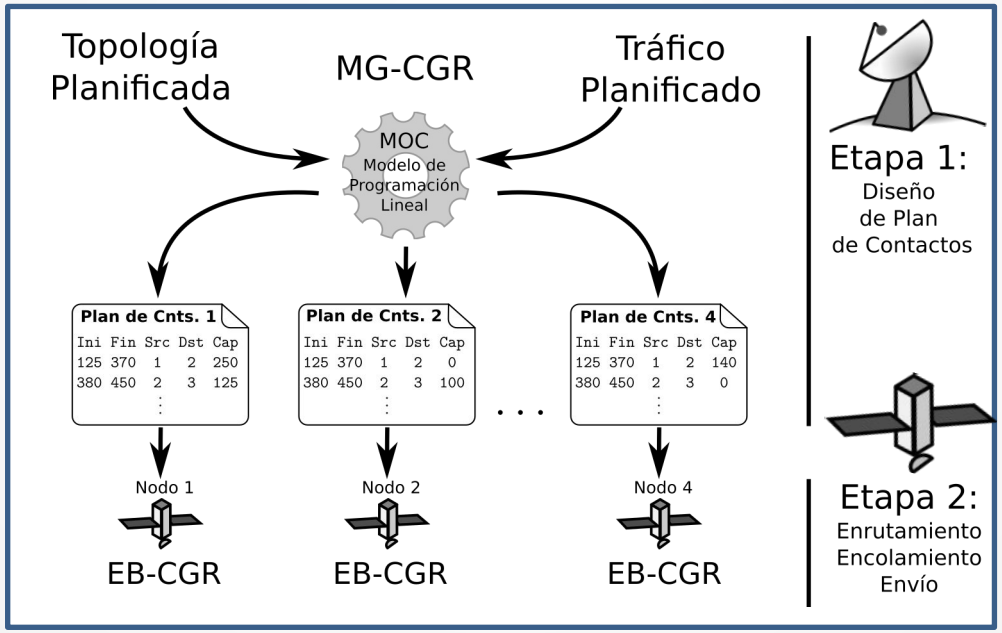
$\{X_e^{y,z}\}$	Tráfico desde y a z enviado en el arco e
$\{B_{t_q,v}^{y,z}\}$	Ocupación del buffer del nodo v en el tiempo t_q por el tráfico y, z



Resultado:
Flujo Óptimo sin Congestión

Multi Graph CGR (MG-CGR)

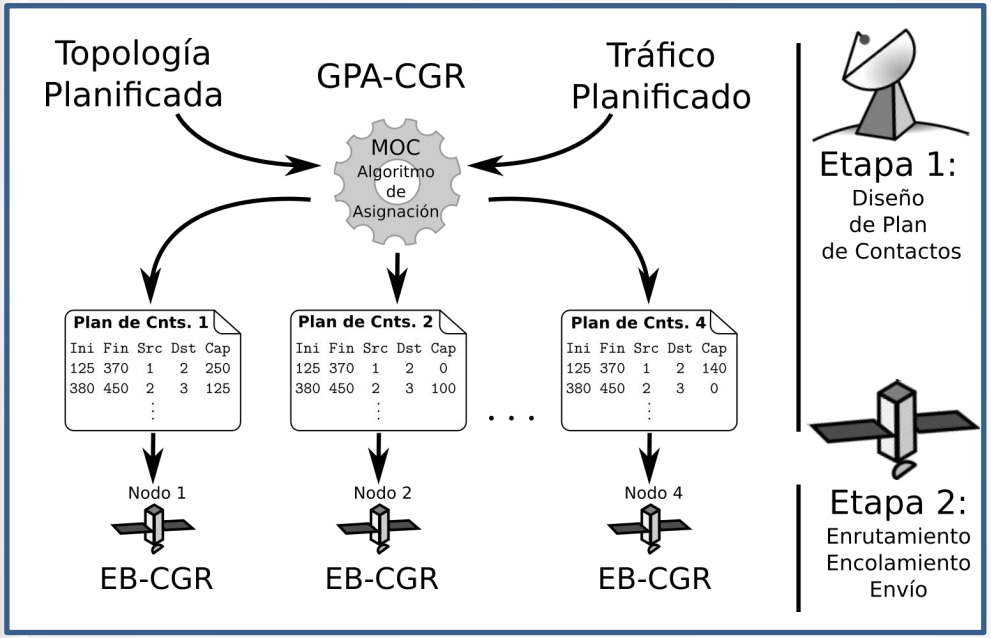
MG-CGR aprovecha la previsibilidad del tráfico para evitar la congestión global mediante la reserva de recursos de comunicación por adelantado



No soporta mas de un Tráfico por Nodo,
Utiliza un modelo LP que no escala a escenarios grandes

Global Path Aware CGR (GPA-CGR)

GPA-CGR soluciona estos problemas mediante un *Algoritmo de Asignación*



Algoritmo 6: GPA-CGR

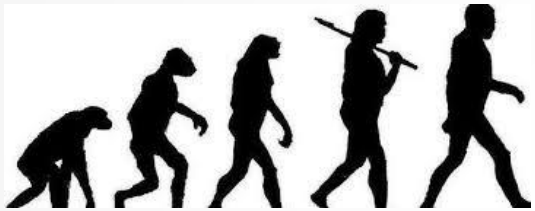
input : $Topology, Traffic$
output: $Cplans[\,], RoutedTraffic$
global : $RoutedTraffic, BufferCap$

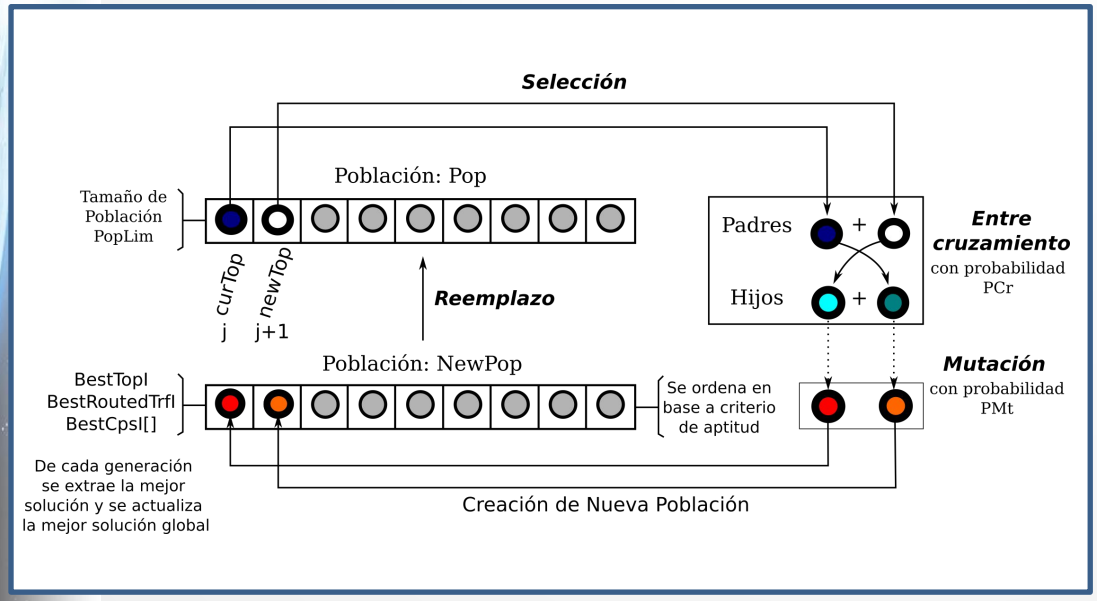
```
1  Generators ← GetSortedGenerators(Traffic);
2  RouteTable ← ComputeRouteTable(Topology, Generators);
3  for g_i ∈ Generators do
4      while g_i.GetCurrentDataLength() > 0 do
5          Route ← getBestRoute(g_i, RouteTable);
6          if Route ≠ ∅ then
7              RouteGenerator(g_i, Route);
8              UpdateBuffersCapacity();
9              DecreaseContactsCapacity();
10             UpdateRouteTable();
11         else
12             skip g_i;
13 for n_i ∈ Topology.Nodes do
14     Cplans.At(n_i) ← GetContactPlan(RoutedTraffic, n_i);
15 return [Cplans[ ], RoutedTraffic]
```

La solución no es necesariamente óptima

Optimización Evolutiva de GPA-CGR

EGPA-CGR proporciona soluciones que se acercan al óptimo y una mayor flexibilidad en la definición de la función objetivo a optimizar



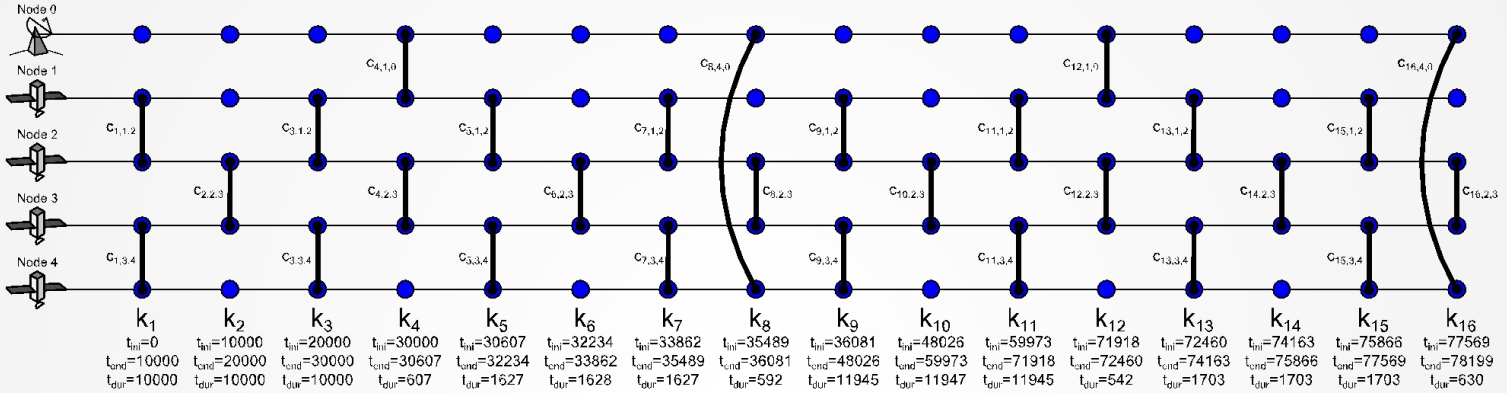


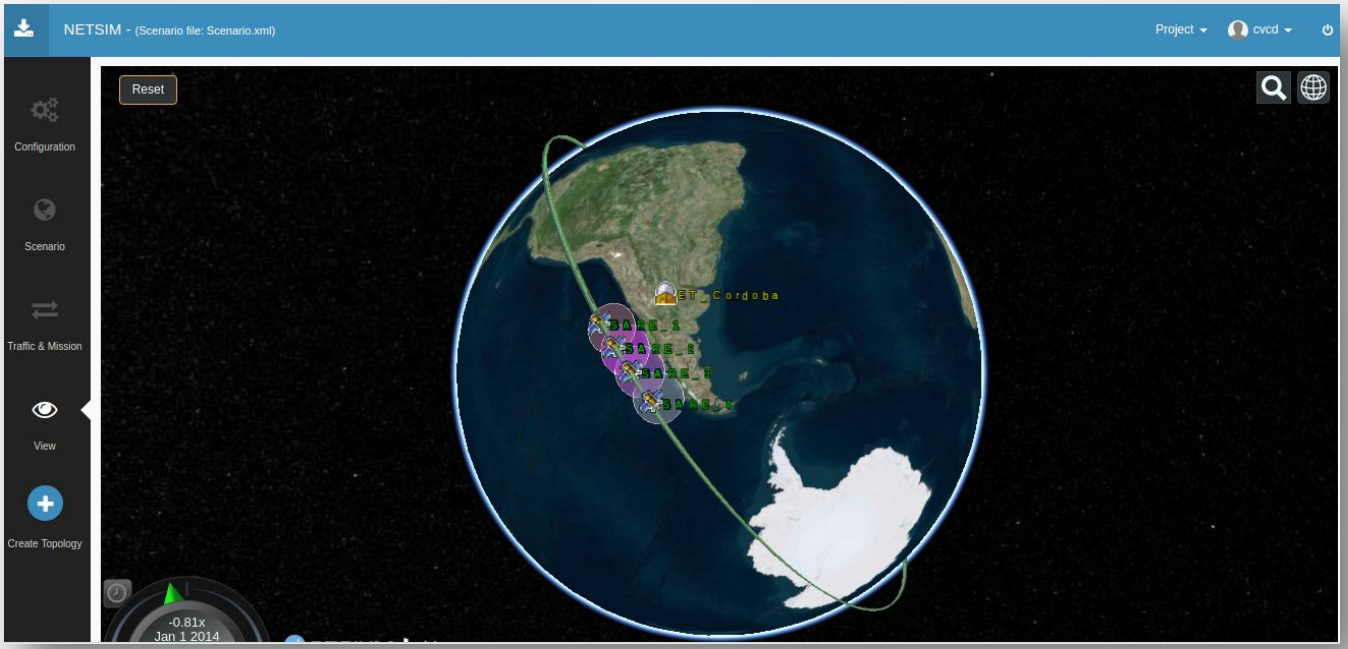
Algoritmo 7: EGPA-CGR

input : $Topology, Traffic, PCr, PMt, Iters, PopLim$
output: $Cps[\,], RoutedTrf$
global : $Pop, NewPop$
global : $BestTop, BestTopI$
global : $BestRoutedTrf, BestRoutedTrfI$
global : $BestCps[\,], BestCpsI[\,]$

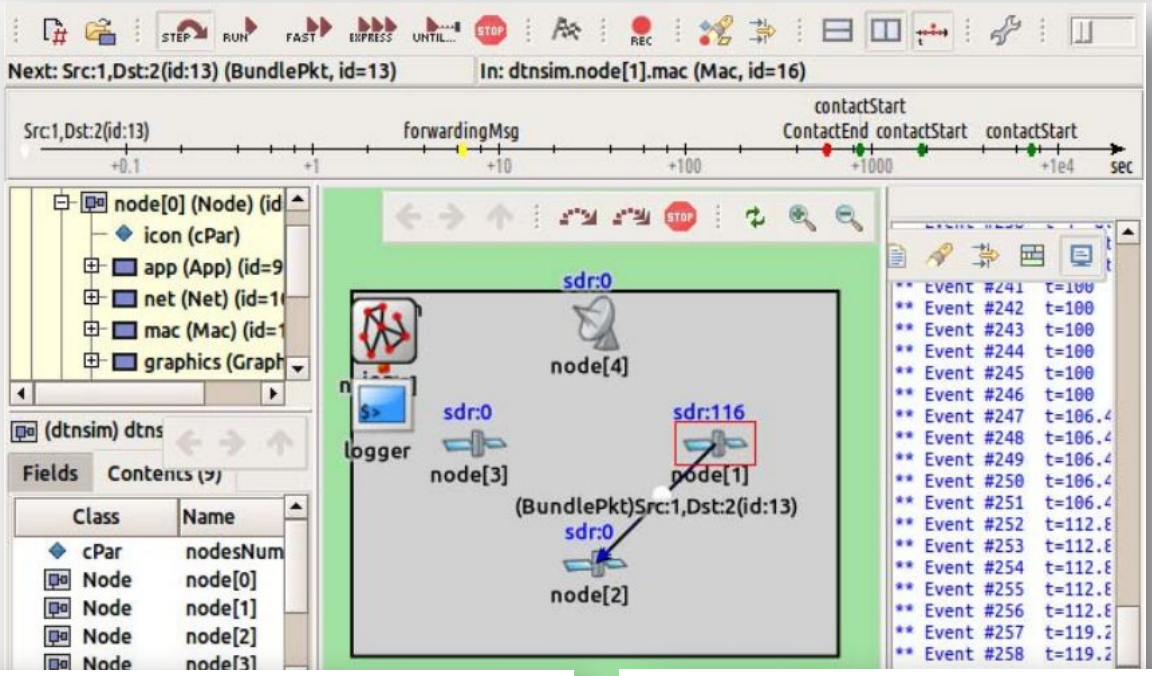
```
1  InitializePopulation(PopLim);
2  SortPopulation(Pop);
3  for i ← 1 to Iters do
4      j ← Pop.Begin();
5      CurTop ← Pop.At(j);
6      j ← j + 1;
7      while j ≠ Pop.End() do
8          NewTop ← Pop.At(j);
9          if Random(0,1) ≤ PCr then
10             TopA, TopB =CrossOver(CurTop, NewTop);
11             Mutate(TopA, PMt);
12             Mutate(TopB, PMt);
13             NewPop.PushBack(TopA);
14             NewPop.PushBack(TopB);
15             if NewPop.Size()≥ PopLim then
16                 break;
17             Pop.PopFront();
18             j ← Pop.Begin();
19             if j ≠ Pop.End() then
20                 curTop = Pop.At(j);
21                 j ← j + 1;
22         else
23             j ← j + 1;
24             if j = Pop.End() then
25                 Pop.PopFront();
26                 j ← Pop.Begin();
27                 CurTop ← Pop.At(j);
28                 j ← j + 1;
29     SortPopulation(NewPop);
30     BestTopI = NewPop.Front();
31     [BestCpsI[ ], BestRoutedTrfI] = GPA-CGR(BestTopI, Traffic);
32     if IsBetter(BestRoutedTrfI, BestRoutedTrf) then
33         BestRoutedTrf = BestRoutedTrfI;
34         BestCps[ ] = BestCpsI[ ];
35     Pop = NewPop;
36     NewPop.Clear();
37 return [BestCps[ ], BestRoutedTrf]
```

Evaluación


Node 0
Node 1
Node 2
Node 3
Node 4
k1 tini=0 tcnd=10000 tdur=10000
k2 tini=10000 tcnd=20000 tdur=10000
k3 tini=20000 tcnd=30000 tdur=10000
k4 tini=30000 tcnd=30607 tdur=607
k5 tini=30607 tcnd=32234 tdur=1627
k6 tini=32234 tcnd=33862 tdur=1628
k7 tini=33862 tcnd=35489 tdur=1627
k8 tini=35489 tcnd=36081 tdur=592
k9 tini=36081 tcnd=48026 tdur=11945
k10 tini=48026 tcnd=59973 tdur=11947
k11 tini=59973 tcnd=71918 tdur=11945
k12 tini=71918 tcnd=72460 tdur=542
k13 tini=72460 tcnd=74163 tdur=1703
k14 tini=74163 tcnd=75866 tdur=1703
k15 tini=75866 tcnd=77569 tdur=1703
k16 tini=77569 tcnd=78199 tdur=630

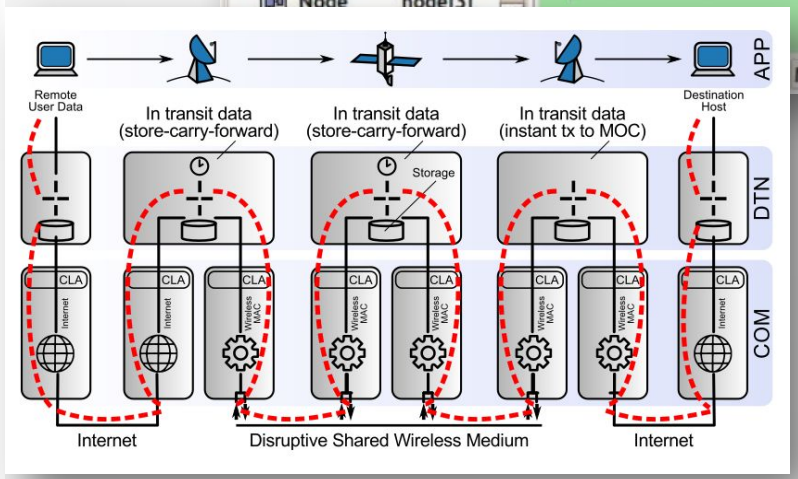

NETSIM - (Scenario file: Scenario.xml)
Project
cvcd
Reset
Configuration
Scenario
Traffic & Mission
View
Create Topology
-0.81x
Jan 1 2014

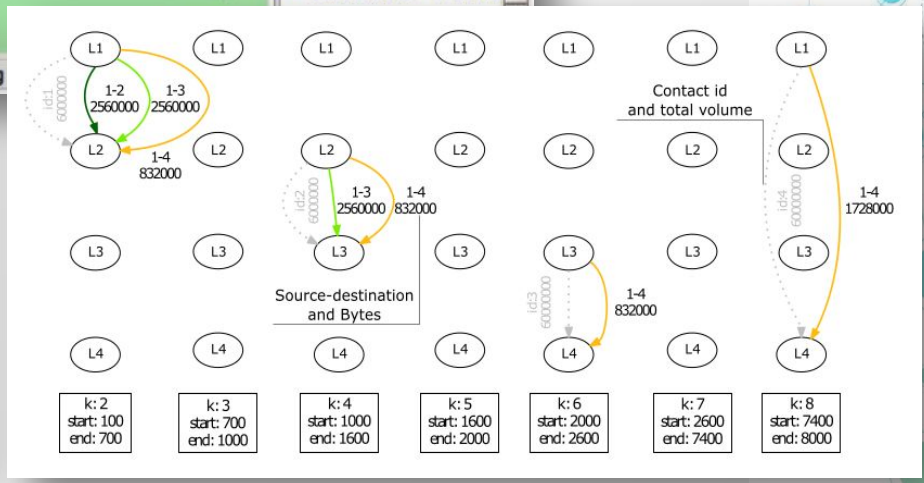
Evaluación



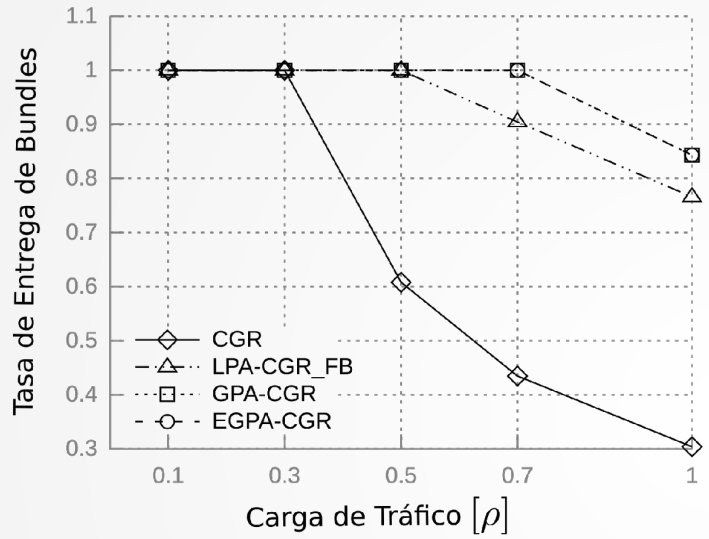
https://bitbucket.org/lcd-unc-ar/dtnsim/src/master/

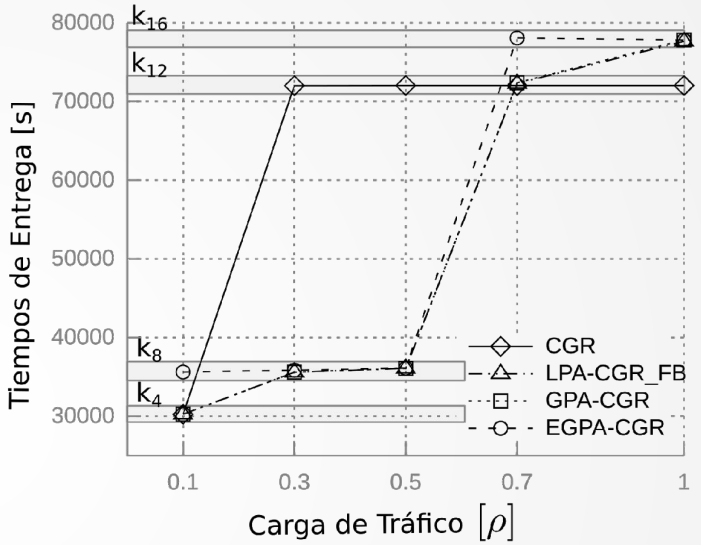
DtnSim: Bridging the Gap between Simulation and Implementation of Space-Terrestrial DTNs

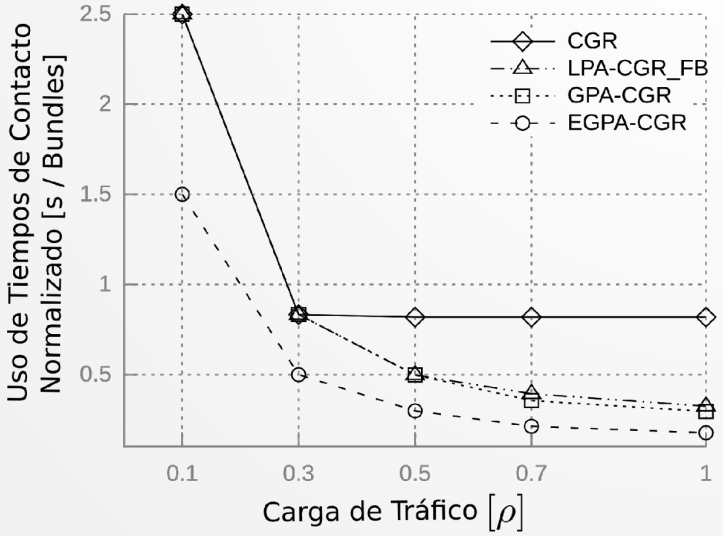


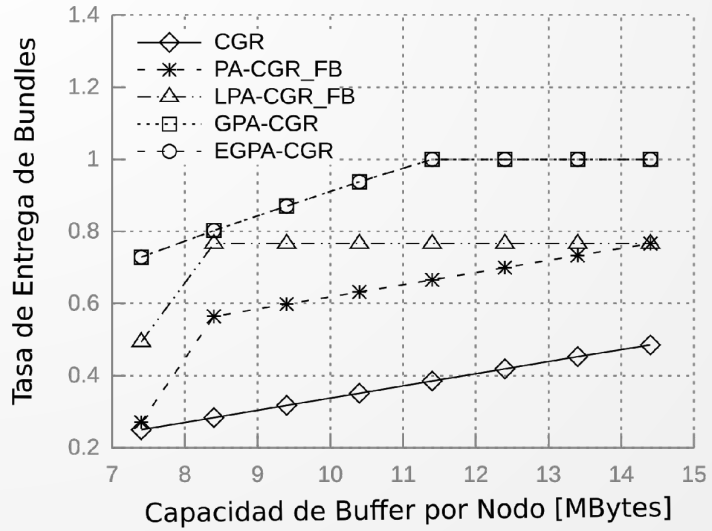


Evaluación

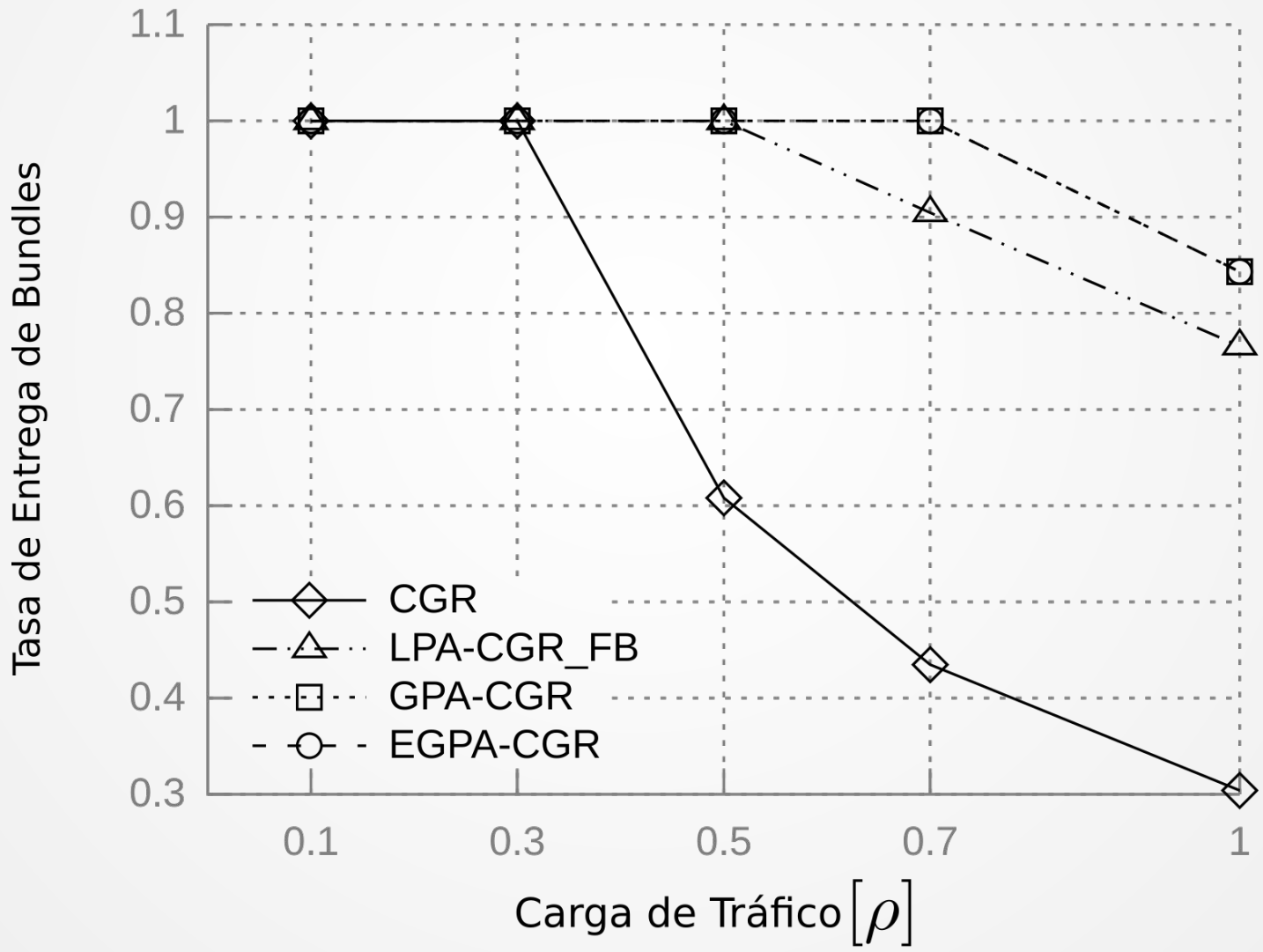

Tasa de Entrega de Bundles
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.1
0.3
0.5
0.7
1
Carga de Tráfico [ρ]
CGR
LPA-CGR_FB
GPA-CGR
EGPA-CGR


Tiempos de Entrega [s]
80000
70000
60000
50000
40000
30000
k16
k12
k8
k4
0.1
0.3
0.5
0.7
1
Carga de Tráfico [ρ]
CGR
LPA-CGR_FB
GPA-CGR
EGPA-CGR

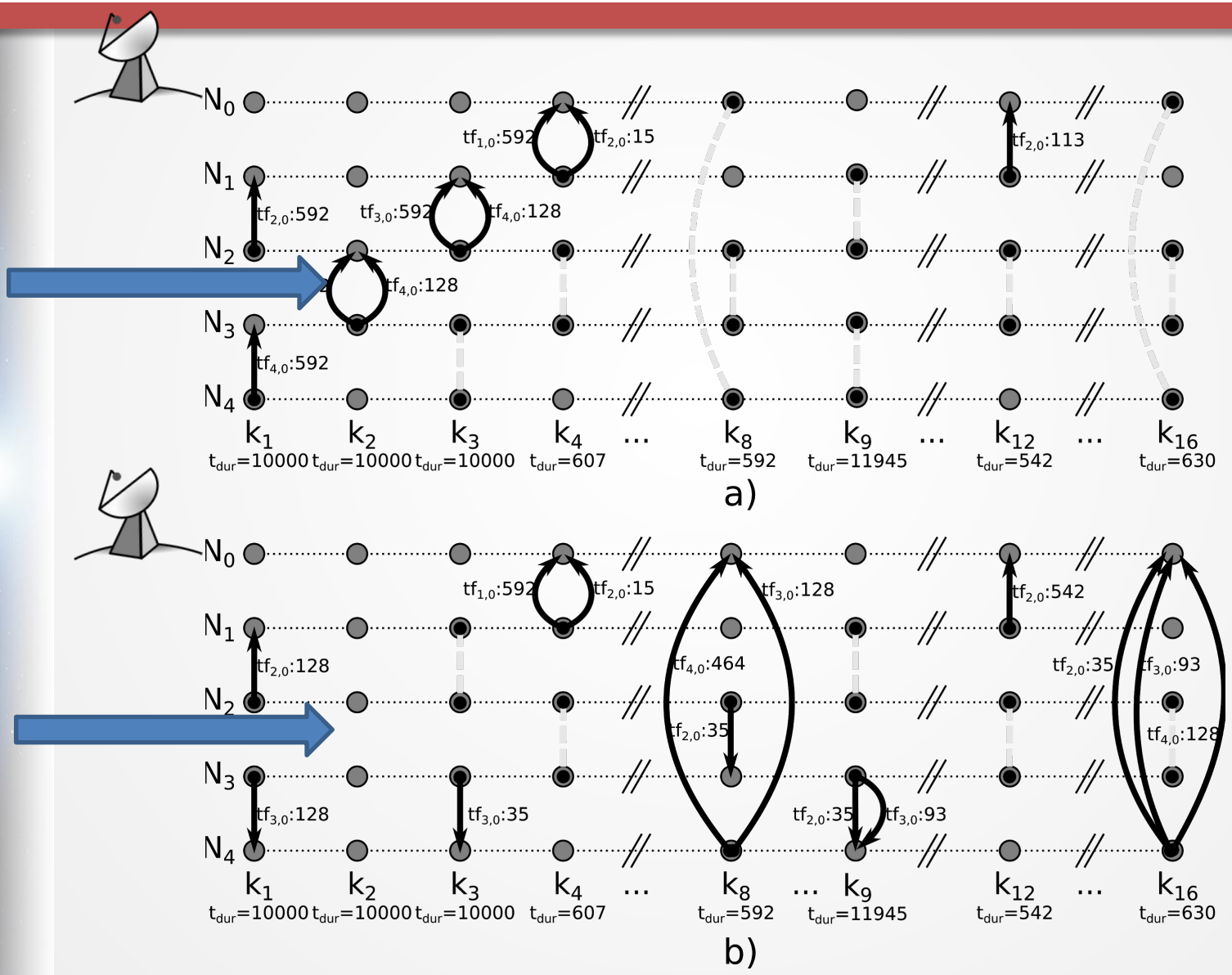

Uso de Tiempos de Contacto Normalizado [s / Bundles]
2.5
2
1.5
1
0.5
0.1
0.3
0.5
0.7
1
Carga de Tráfico [ρ]
CGR
LPA-CGR_FB
GPA-CGR
EGPA-CGR


Tasa de Entrega de Bundles
1.4
1.2
1
0.8
0.6
0.4
0.2
7
8
9
10
11
12
13
14
15
Capacidad de Buffer por Nodo [MBytes]
CGR
PA-CGR_FB
LPA-CGR_FB
GPA-CGR
EGPA-CGR

Evaluación


Tasa de Entrega de Bundles
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
CGR
LPA-CGR_FB
GPA-CGR
EGPA-CGR
0.1
0.3
0.5
0.7
1
Carga de Tráfico [ρ]

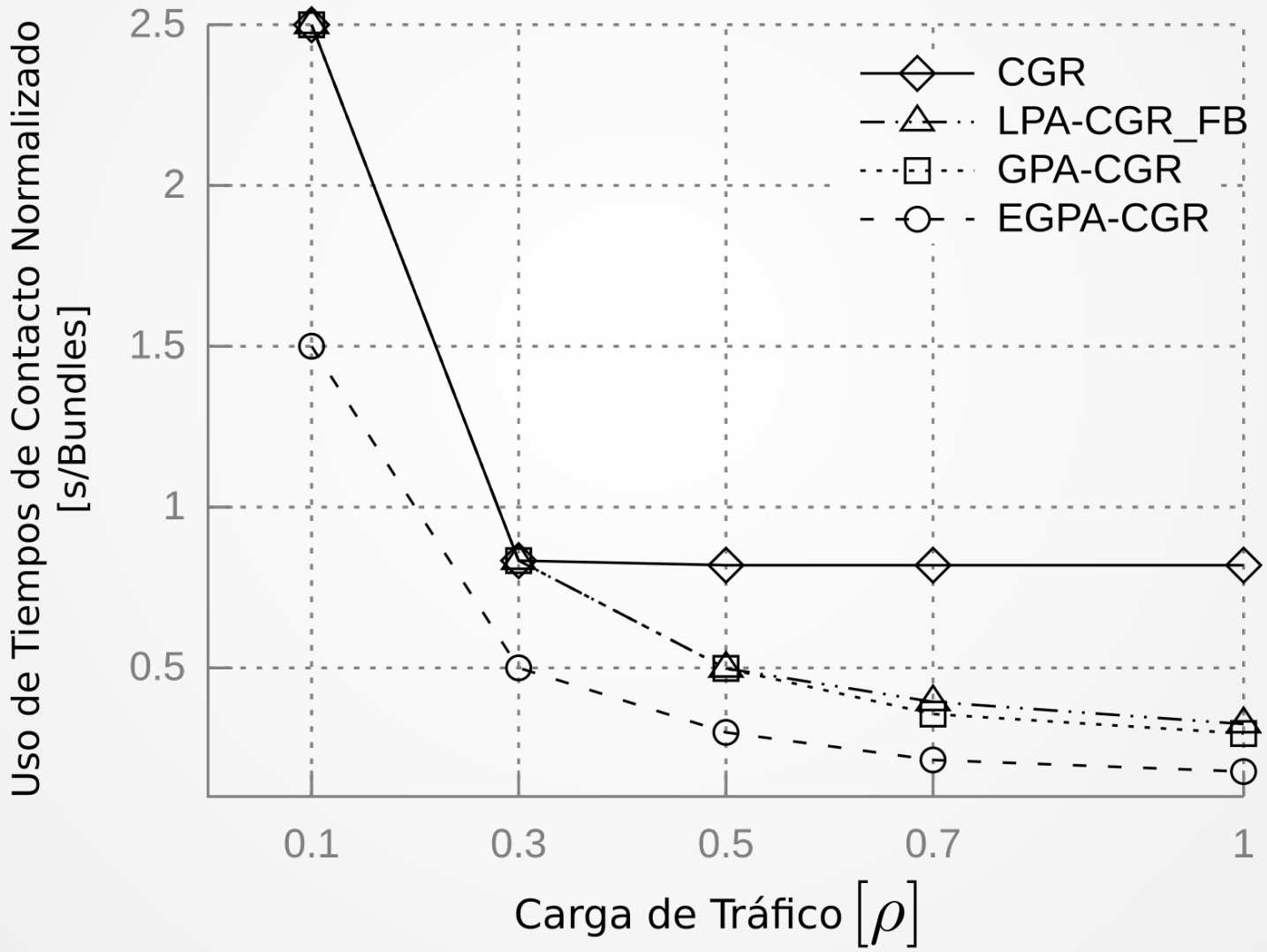
Evaluación



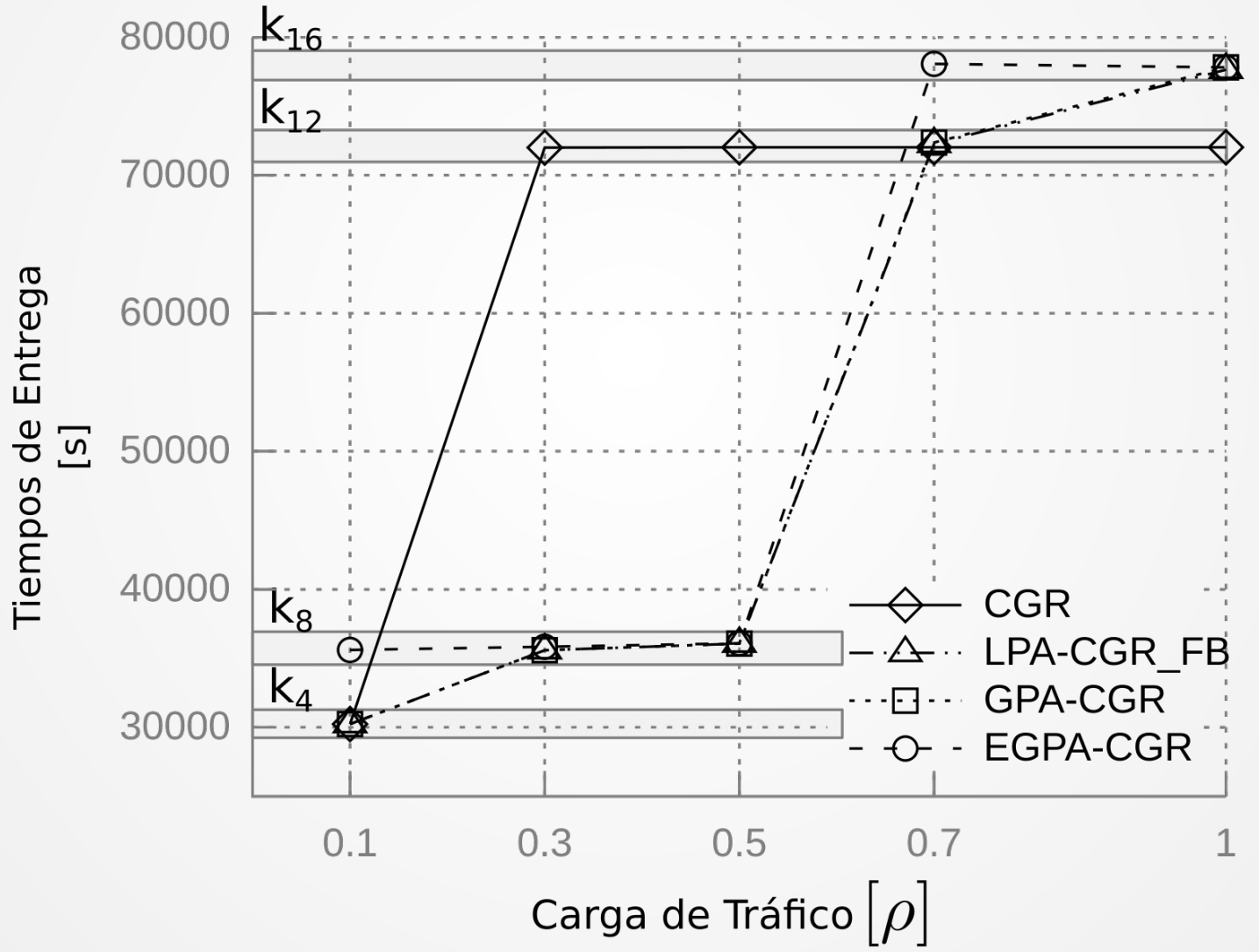
CGR

EGPA-CGR

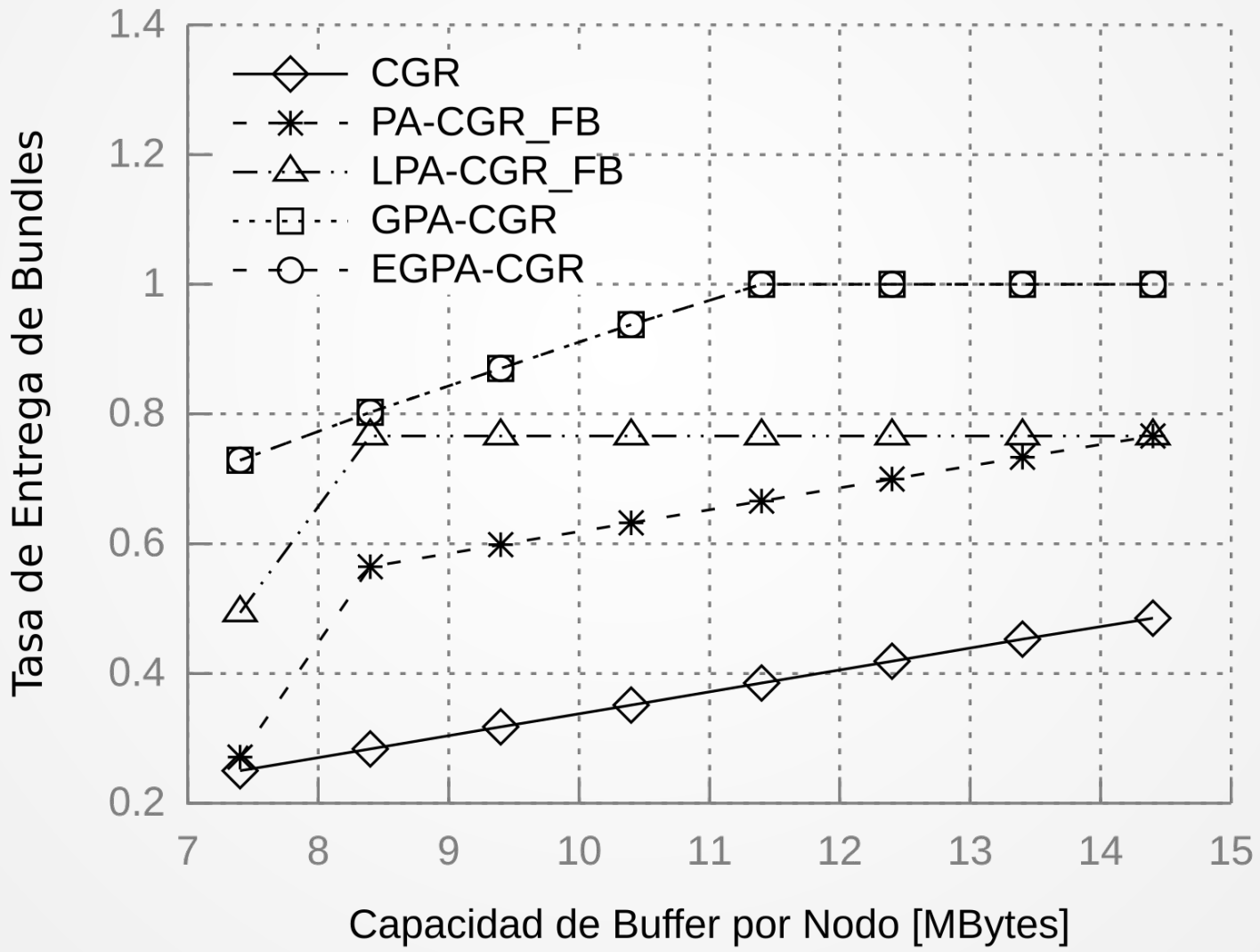
Evaluación


Uso de Tiempos de Contacto Normalizado
[s/Bundles]
2.5
2
1.5
1
0.5
0.1
0.3
0.5
0.7
1
Carga de Tráfico $[\rho]$
CGR
LPA-CGR_FB
GPA-CGR
EGPA-CGR

Evaluación


80000
70000
60000
50000
40000
30000
k_{16}
k_{12}
k_8
k_4
Tiempos de Entrega [s]
0.1
0.3
0.5
0.7
1
Carga de Tráfico $[\rho]$
CGR
LPA-CGR_FB
GPA-CGR
EGPA-CGR

Evaluación


CGR
PA-CGR_FB
LPA-CGR_FB
GPA-CGR
EGPA-CGR
Tasa de Entrega de Bundles
1.4
1.2
1
0.8
0.6
0.4
0.2
7
8
9
10
11
12
13
14
15
Capacidad de Buffer por Nodo [MBytes]

Contribuciones

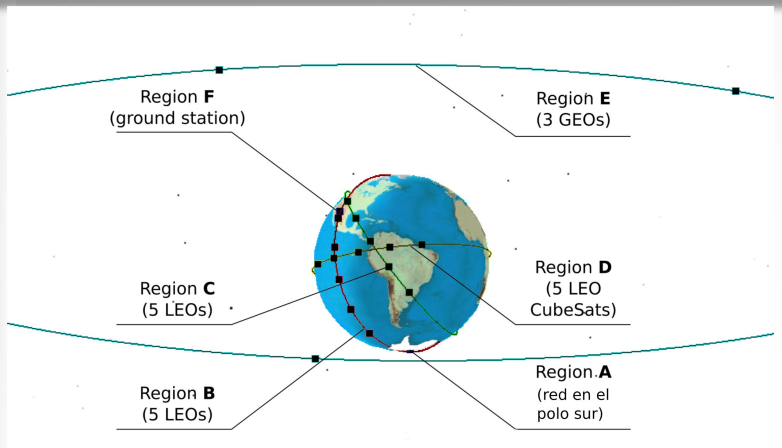
Congestión

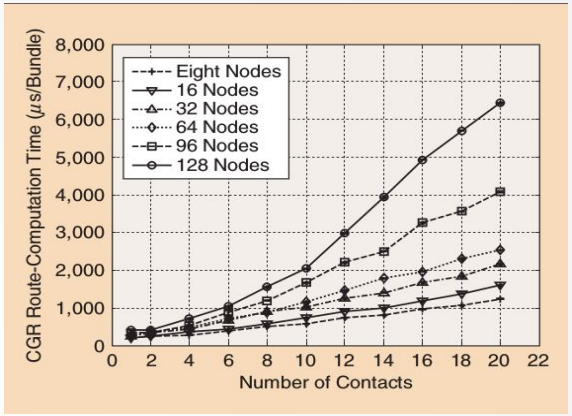
- **P. Madoery**, J. Fraire, and J. Finochietto, "*Congestion management techniques for disruption-tolerant satellite networks*," **International Journal of Satellite Communications and Networking**, May 2017.
- J. Fraire, **P. Madoery**, J. Finochietto, "Traffic-Aware Contact Plan Design for Disruption-Tolerant Space Sensor Networks," **Ad Hoc Networks**, April 2016.
- **P. Madoery**, J. Fraire, and J. Finochietto, "*Analysis of communication strategies for earth observation satellite constellations*," **IEEE Latin America Transactions**, June 2016.

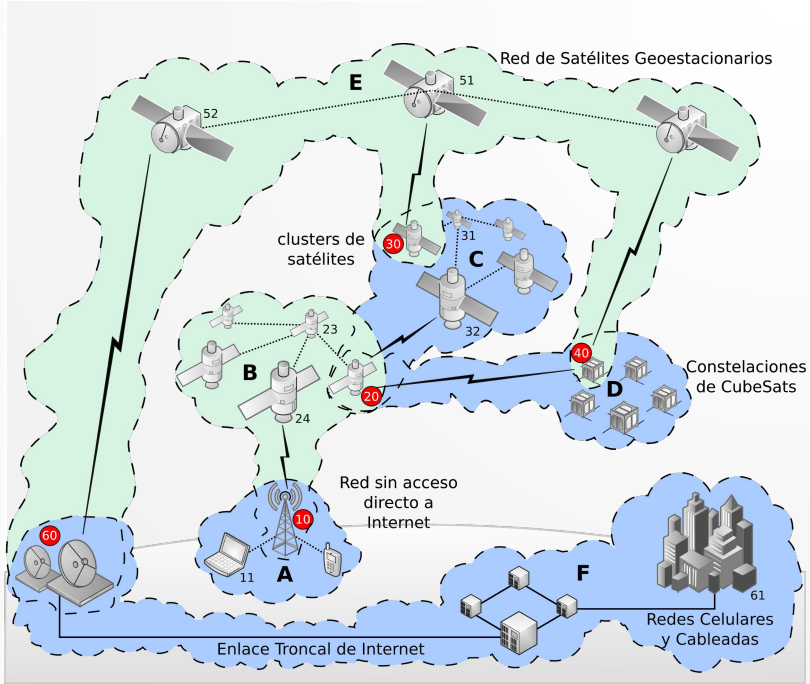
Escalabilidad

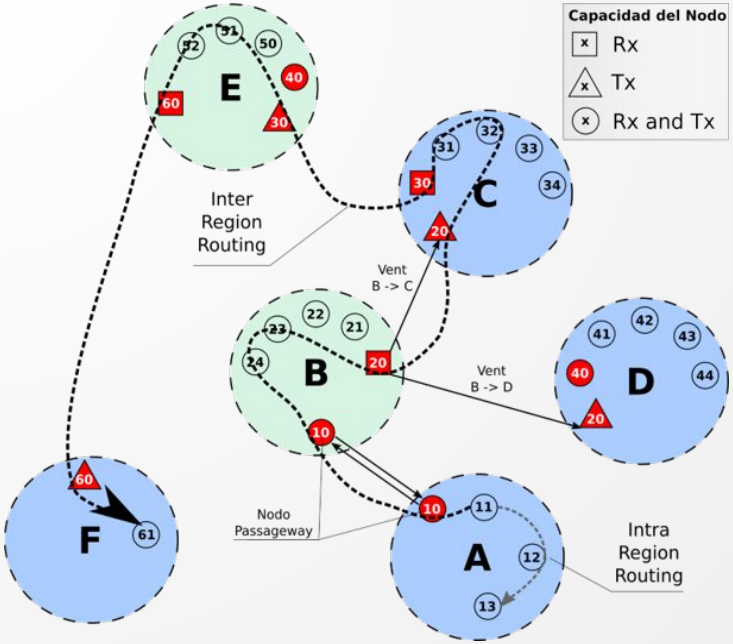
- **P. Madoery**, J. Fraire, F. Raverta, J. Finochietto, and S. Burleigh, "*Managing Routing Scalability in Space DTNs*," In **5th IEEE International Workshop on Space-Terrestrial Internetworking (STINT)**, Huntsville, AL, USA, 2018.
- J. Fraire, **P. Madoery**, Amir Charif, J. Finochietto, "On route table computation strategies in Delay-Tolerant Satellite Networks," **Ad Hoc Networks**, November 2018.
- **P. Madoery**, C. Zerbini, J. Fraire, and J. Finochietto, "*Enhancing contact graph routing forwarding performance for segmented satellites architectures*," In **1st IAA Latin American Symposium on Small Satellites: Advanced Technologies and Distributed Systems**, Buenos Aires, Argentina, March 2017.

Escalabilidad







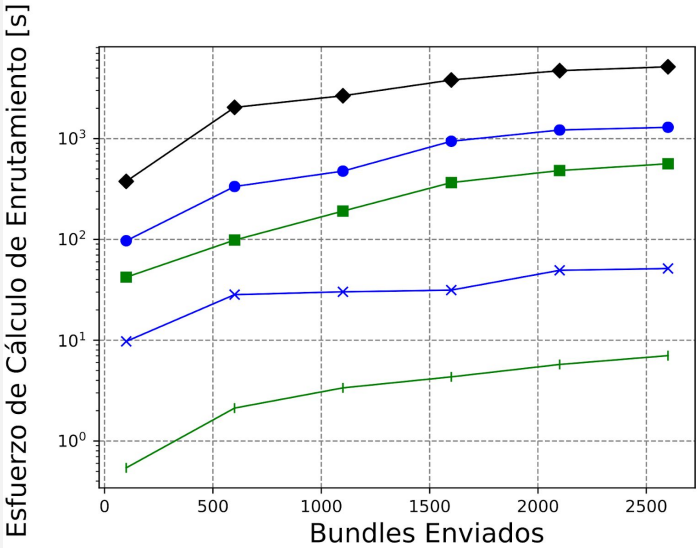


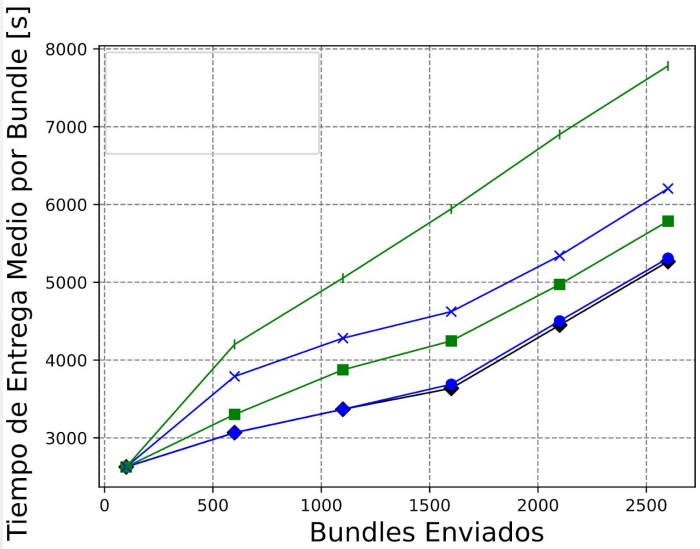
Escalabilidad

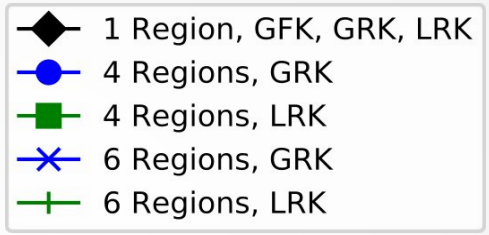
Escalabilidad

Trade-off

Rendimiento







Global Full Knowledge (GFK):
todos los nodos reciben el plan de contactos completo

Global Regionalized Knowledge (GRK):
se filtran los contactos de acuerdo a las regiones. Luego, todos los nodos reciben el mismo plan de contactos

Local Regionalized Knowledge (GLK):
se filtran los contactos de acuerdo a las regiones. Luego, los nodos reciben planes de contactos regionales

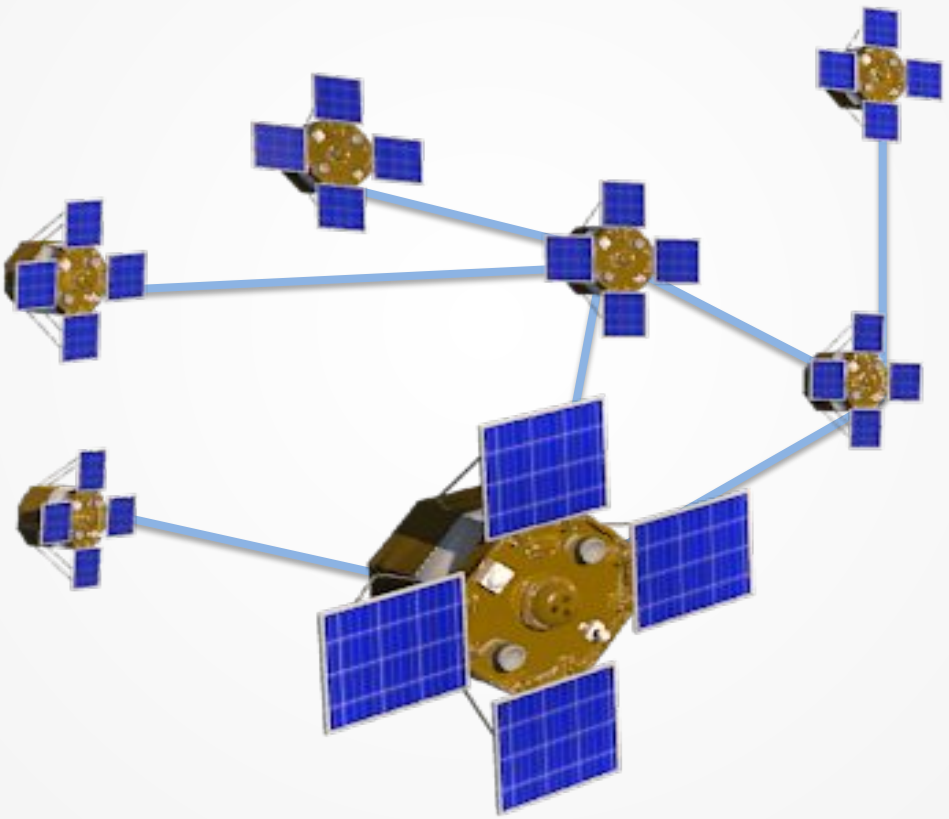
Conclusiones

Se validó la hipótesis de trabajo, lo cual derivó en aportes útiles para la comunidad académica

*Se desarrollaron los mecanismos **LPA/GPA/EGPA-CGR** que se enfocan en el problema de la congestión; y **GRK** y **LRK** que ponen el foco en la escalabilidad.*

Como trabajo futuro, estamos explorando el problema de fallas e incertidumbre y también esquemas de optimización de la planificación y el enrutamiento a nivel de misiones.

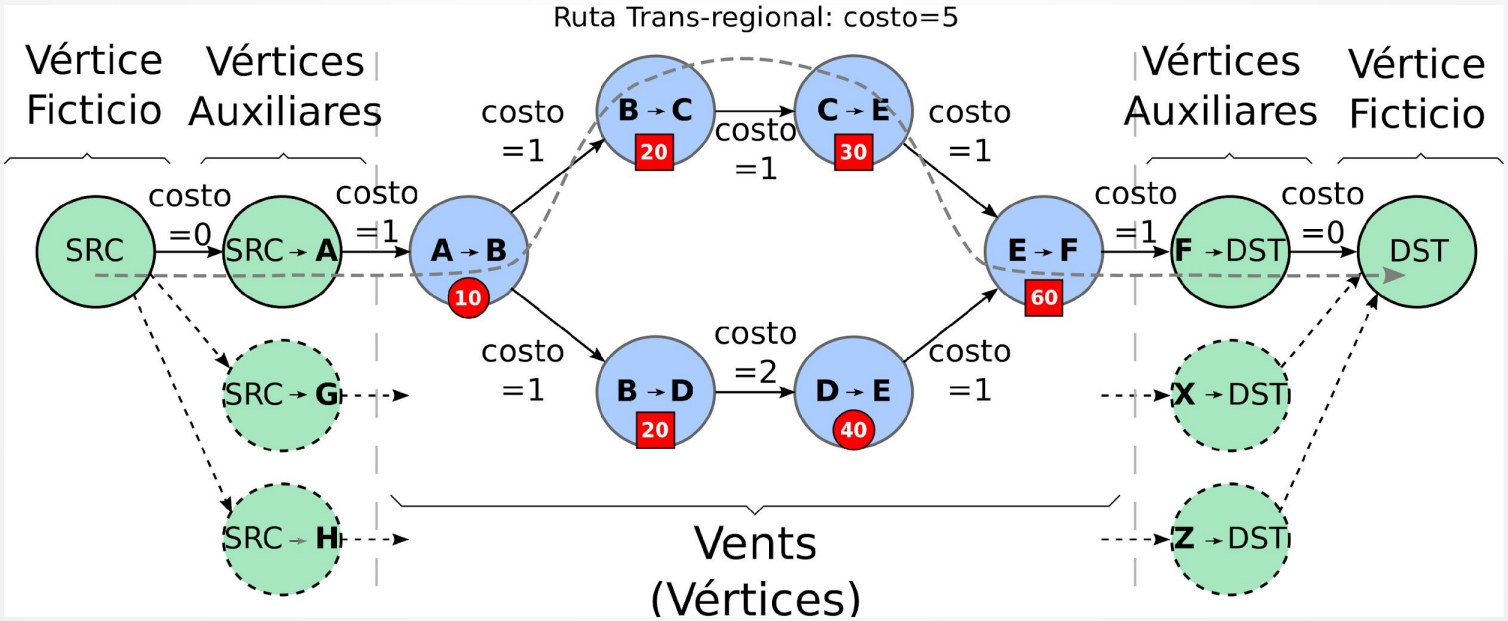
Gracias!



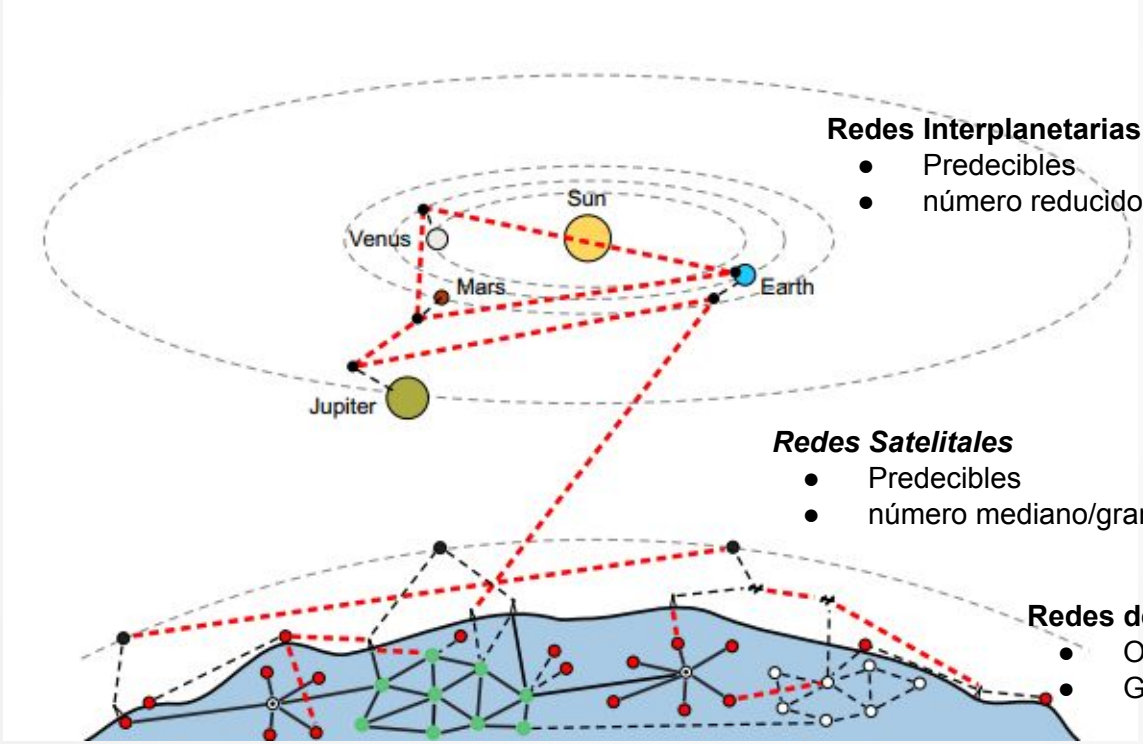
Filminas de Respaldo

Escalabilidad

Inter-Region Routing



DTN: Delay Tolerant Networks



Redes Interplanetarias

- Predecibles
- número reducido de nodos

Redes Satelitales

- Predecibles
- número mediano/grande de nodos

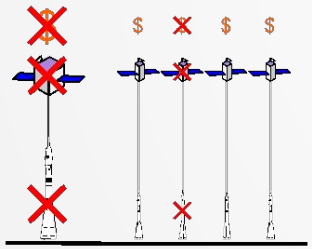
Redes de sensores

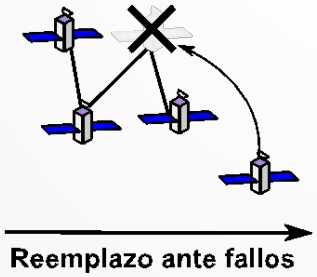
- Oportunísticas/Probabilísticas
- Gran número de nodos

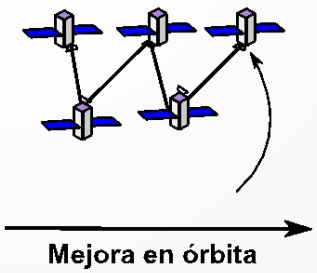
Contexto

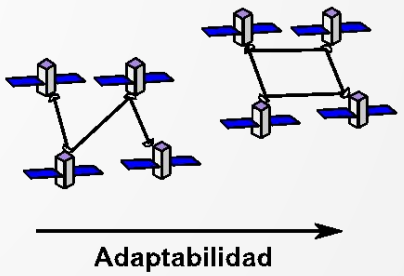
Uso Militar
Arquitectura Fragmentada

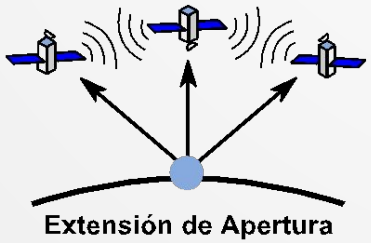
Uso Socio-Económico
Arquitectura Segmentada

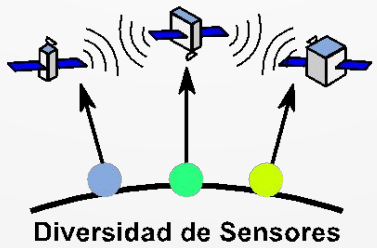


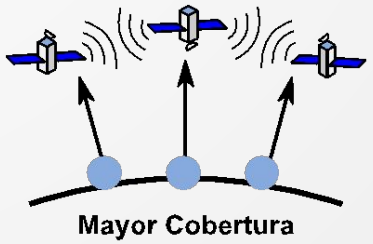




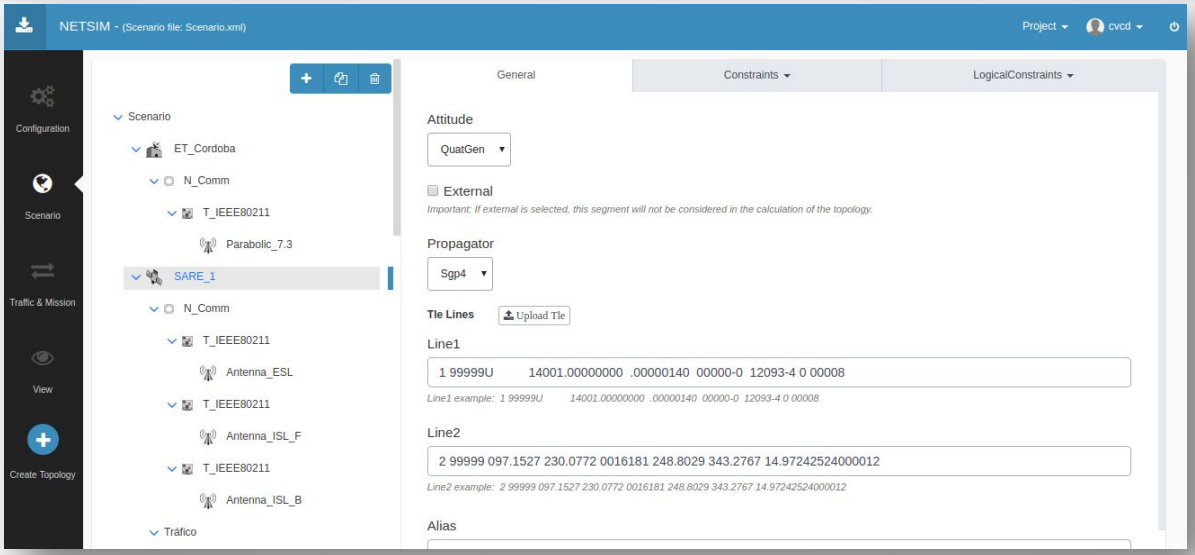


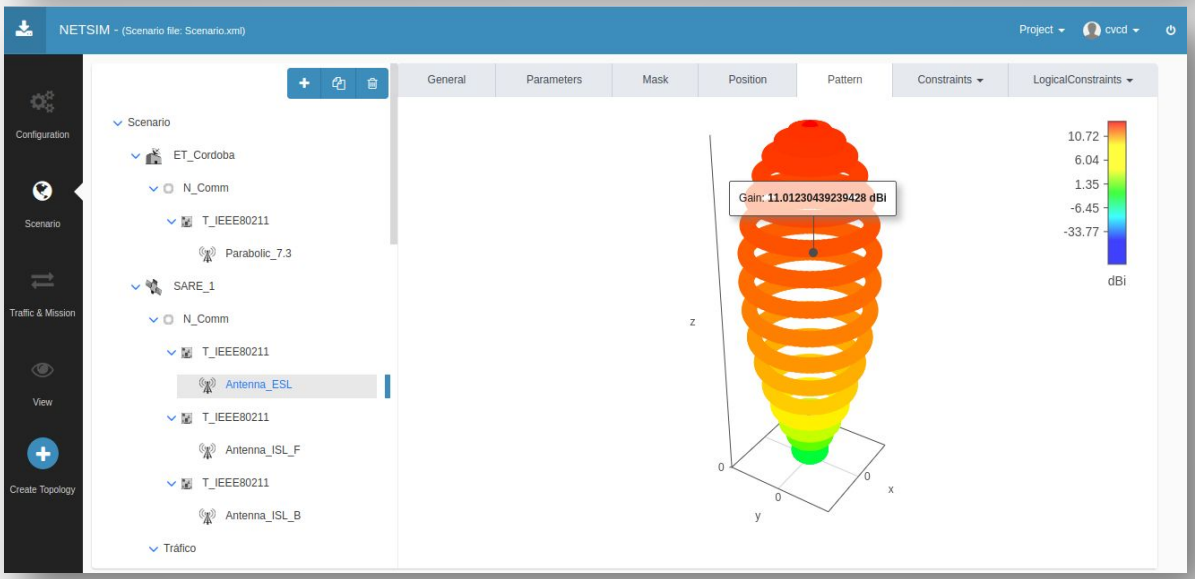




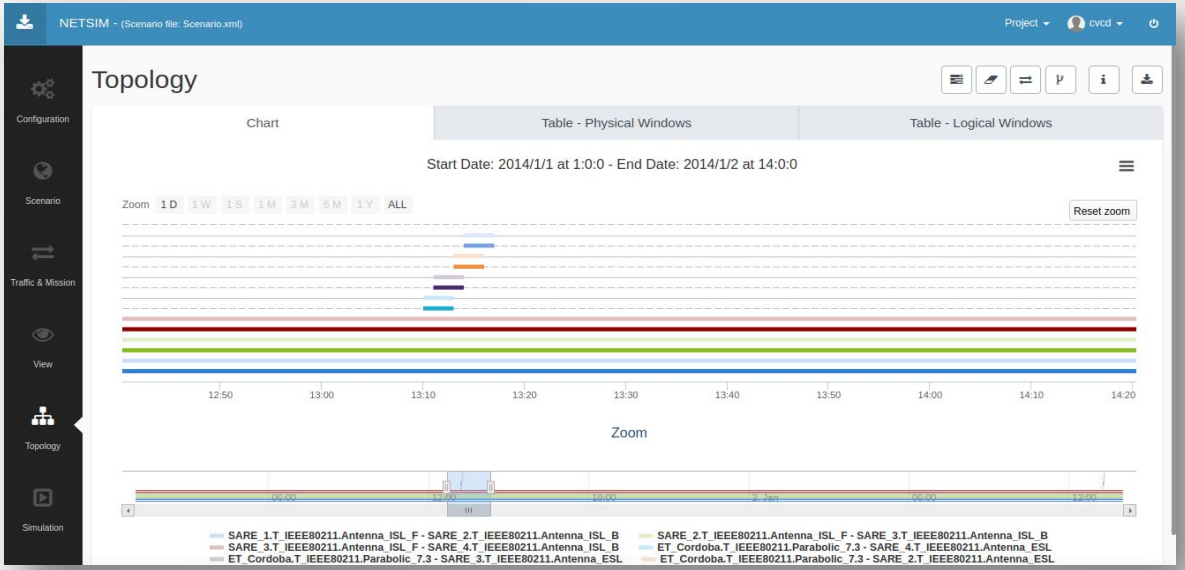


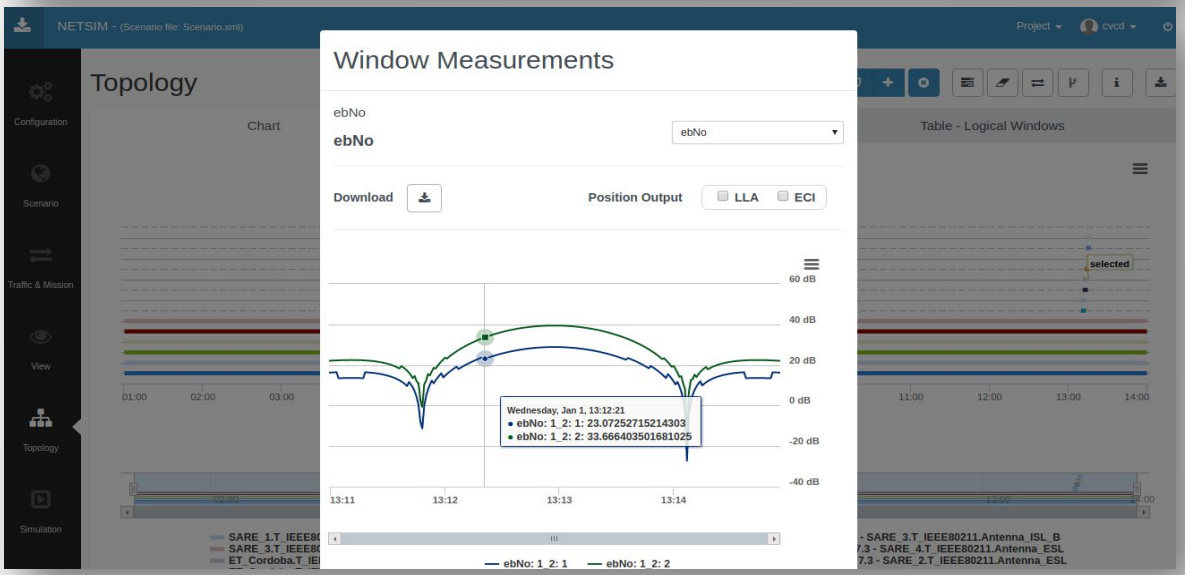
Herramientas Desarrolladas





Herramientas Desarrolladas





Herramientas Desarrolladas

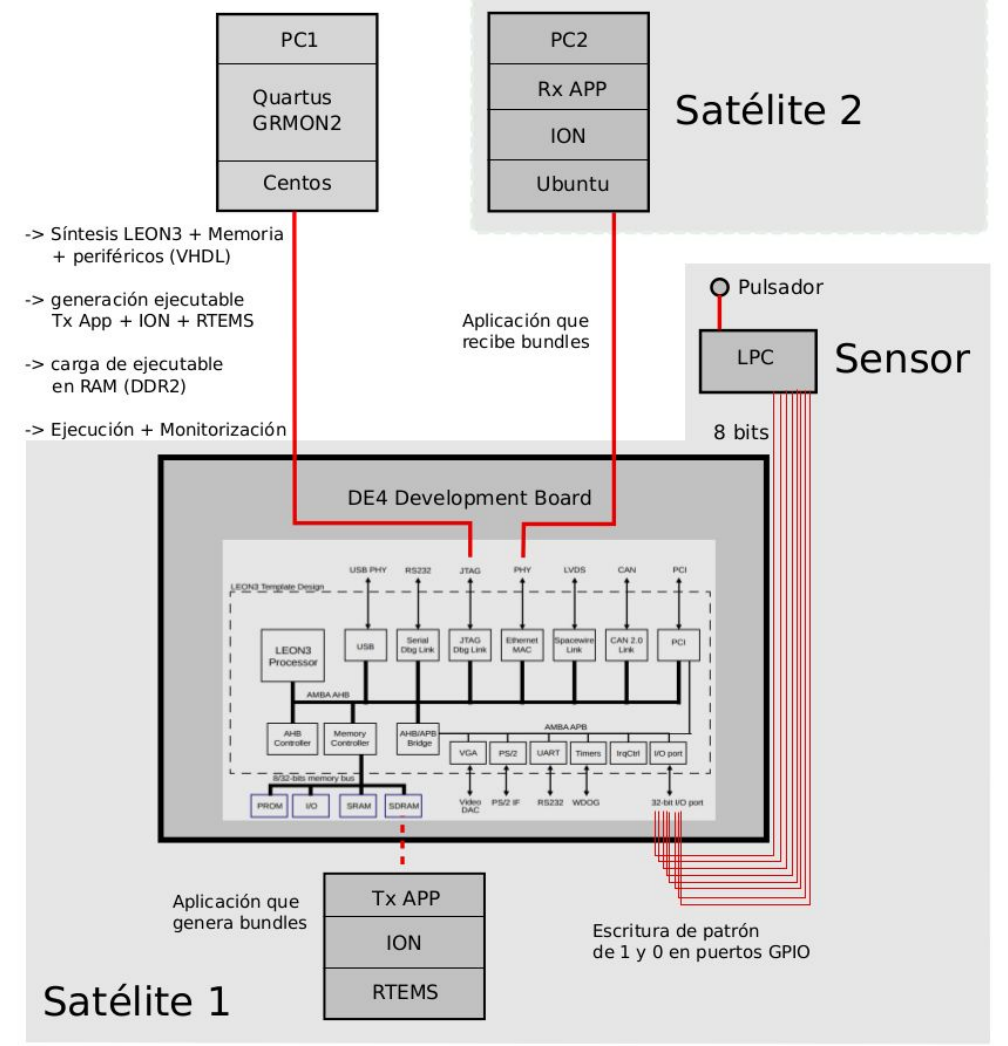


Figura 3: Diagrama en Bloques detallado del Banco de Pruebas

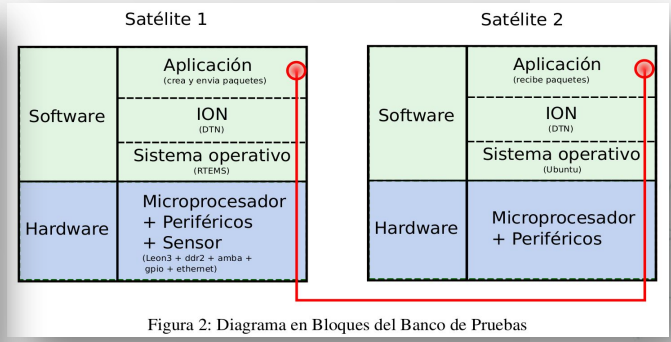
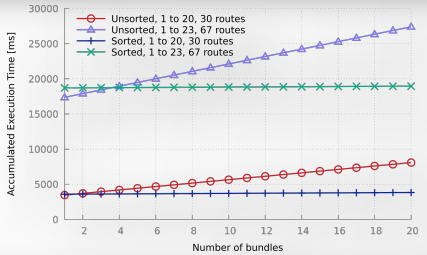
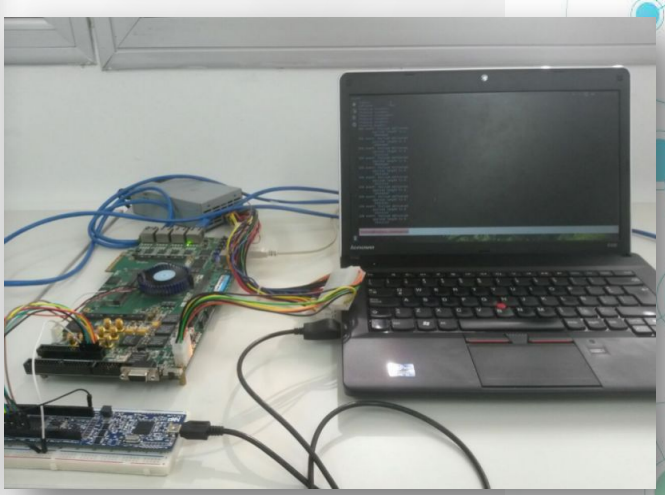


Figura 2: Diagrama en Bloques del Banco de Pruebas



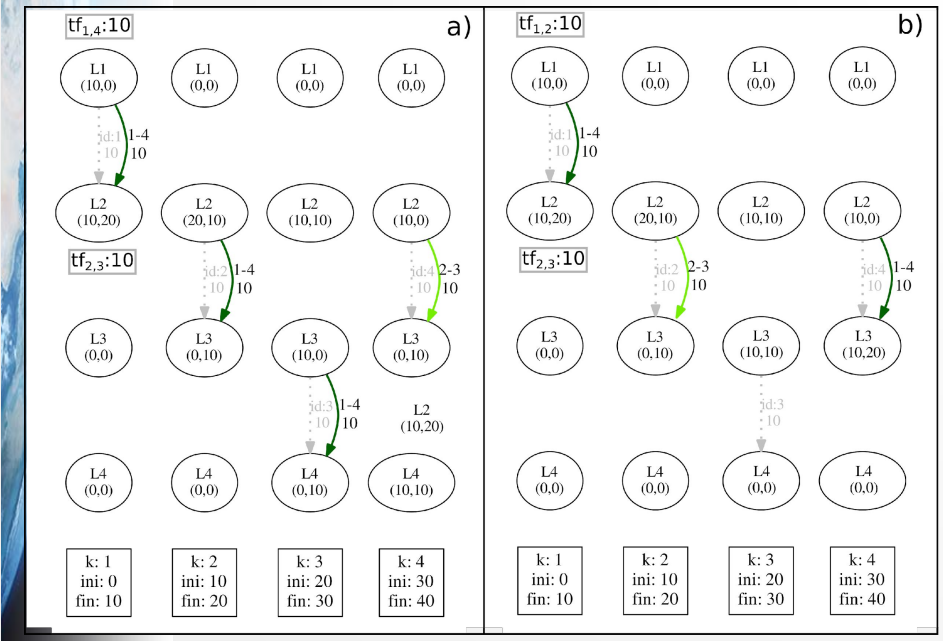


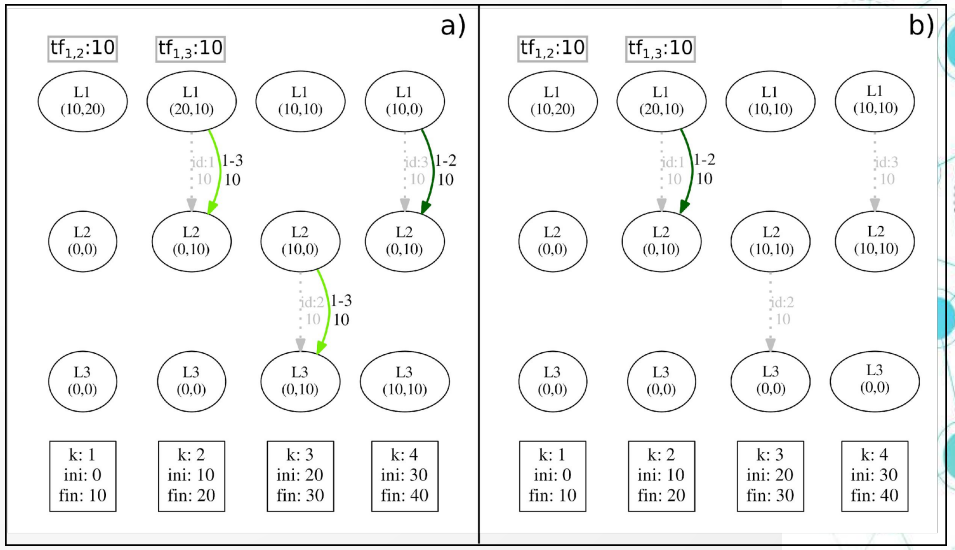
Multi Graph CGR (MG-CGR)

Problemas de Implementabilidad

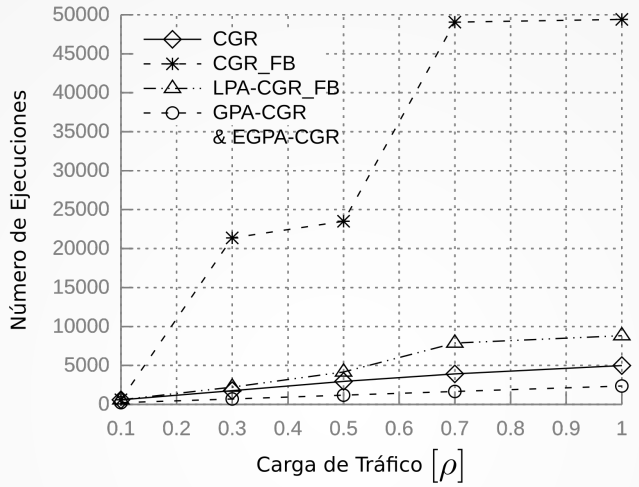
Encolamiento por Nodo

Múltiples Tráficos por Nodo



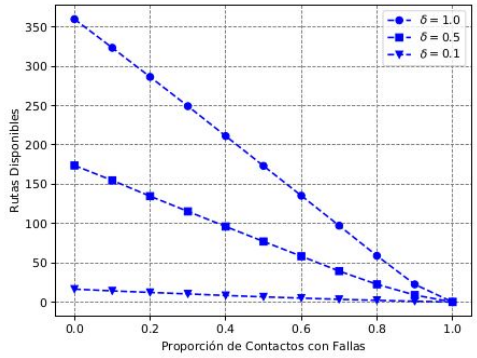


Evaluación

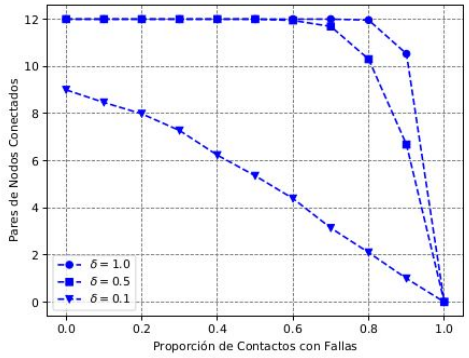


Características \ Algoritmos	CGR	LPA-CGR	GPA-CGR/EGPA-CGR
Nivel de Congestión	Local (1° contacto)	Local (toda la ruta)	Global
Información Requerida	Topología	Topología	Topología + Trafico
Plan de Contactos	El mismo para todos los nodos	El mismo para todos los nodos	Uno diferente para cada nodo
Sobre Carga (*Overhead*)	Ninguna	Ninguna	Ruta calculada
Número de Ejecuciones	Una por bundle en cada nodo	Una por bundle en cada nodo	Una en el nodo central + una por bundle solo en el nodo origen

Incertidumbre

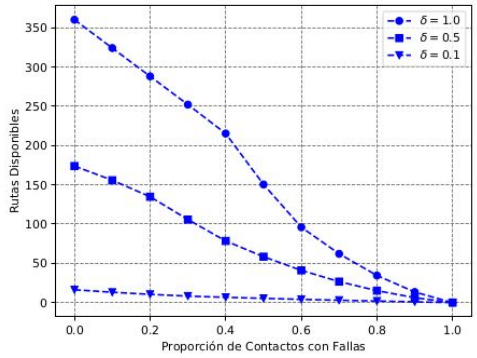


(A)

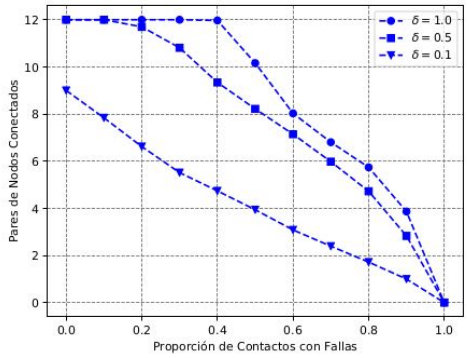


(B)

FIGURA 4.3: Disponibilidad de rutas con inyección de fallas de forma aleatoria



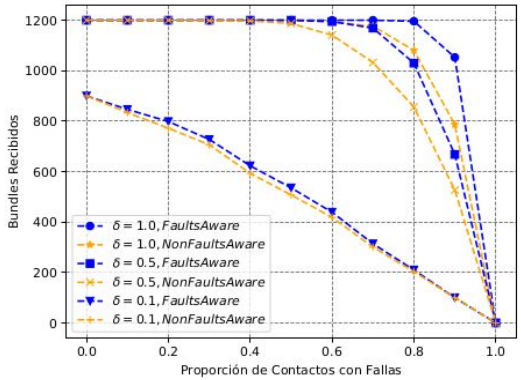
(A)



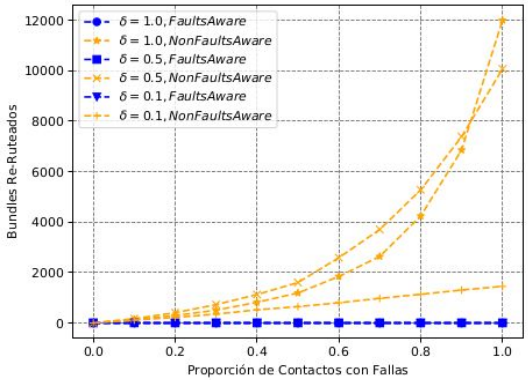
(B)

FIGURA 4.4: Disponibilidad de rutas con inyección de fallas de forma dirigida

Incertidumbre

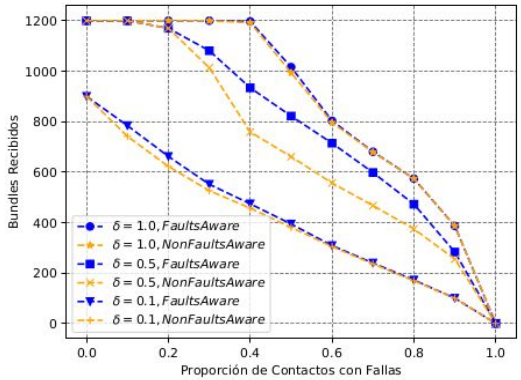


(A)

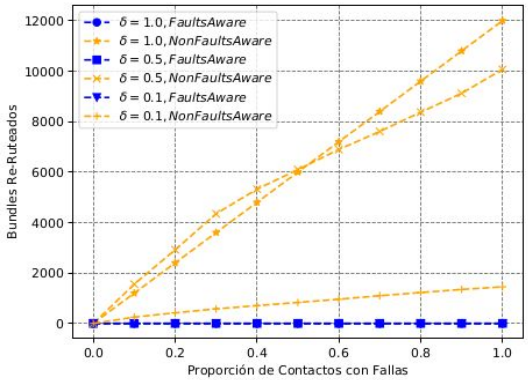


(B)

FIGURA 4.5: Rendimiento con inyección de fallas de forma aleatoria



(A)



(B)

FIGURA 4.6: Rendimiento con inyección de fallas de forma dirigida

Incertidumbre

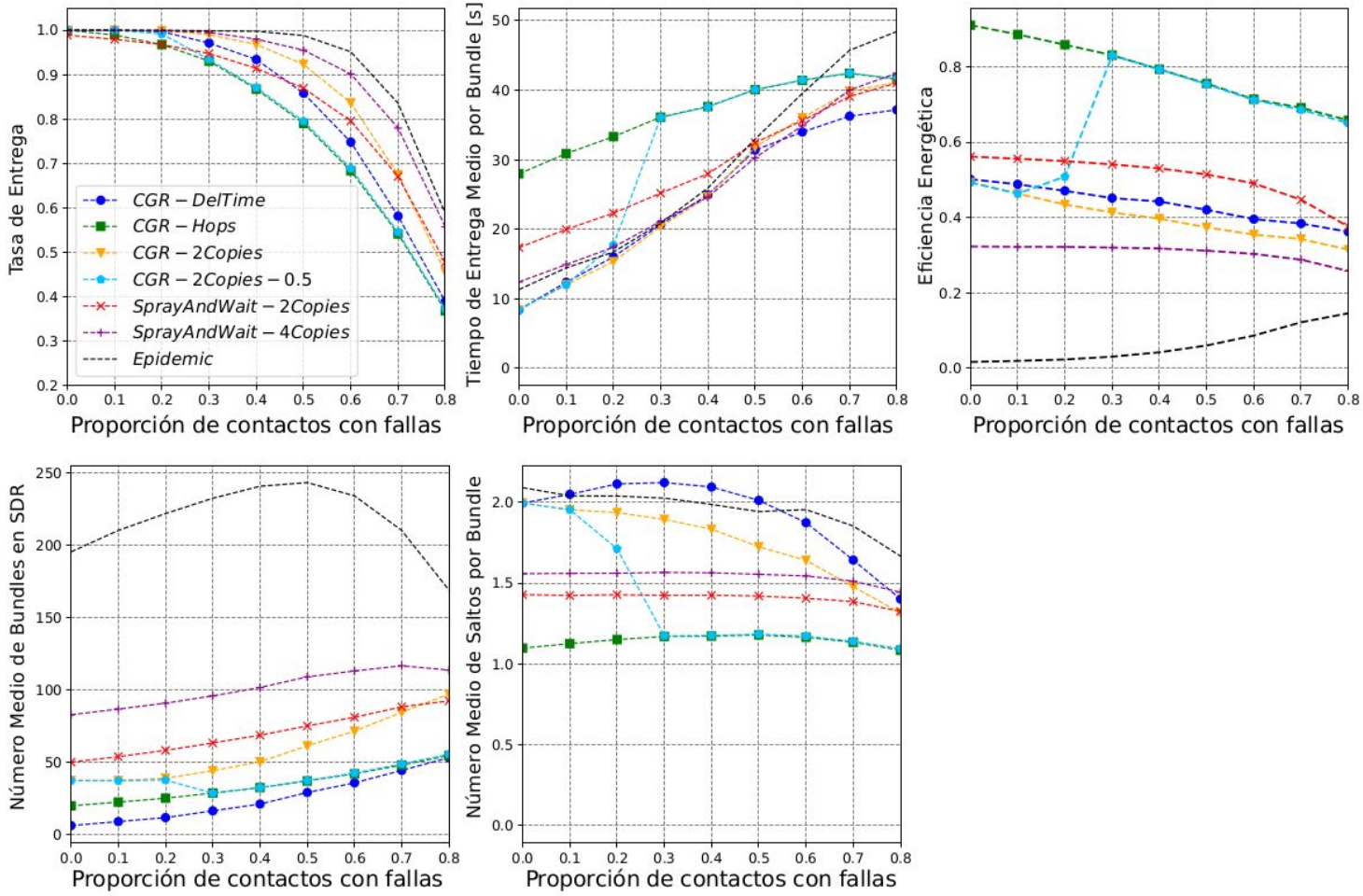


FIGURA 4.8: Resultados de caso de estudio I) redes aleatorias con baja congestión

Incertidumbre

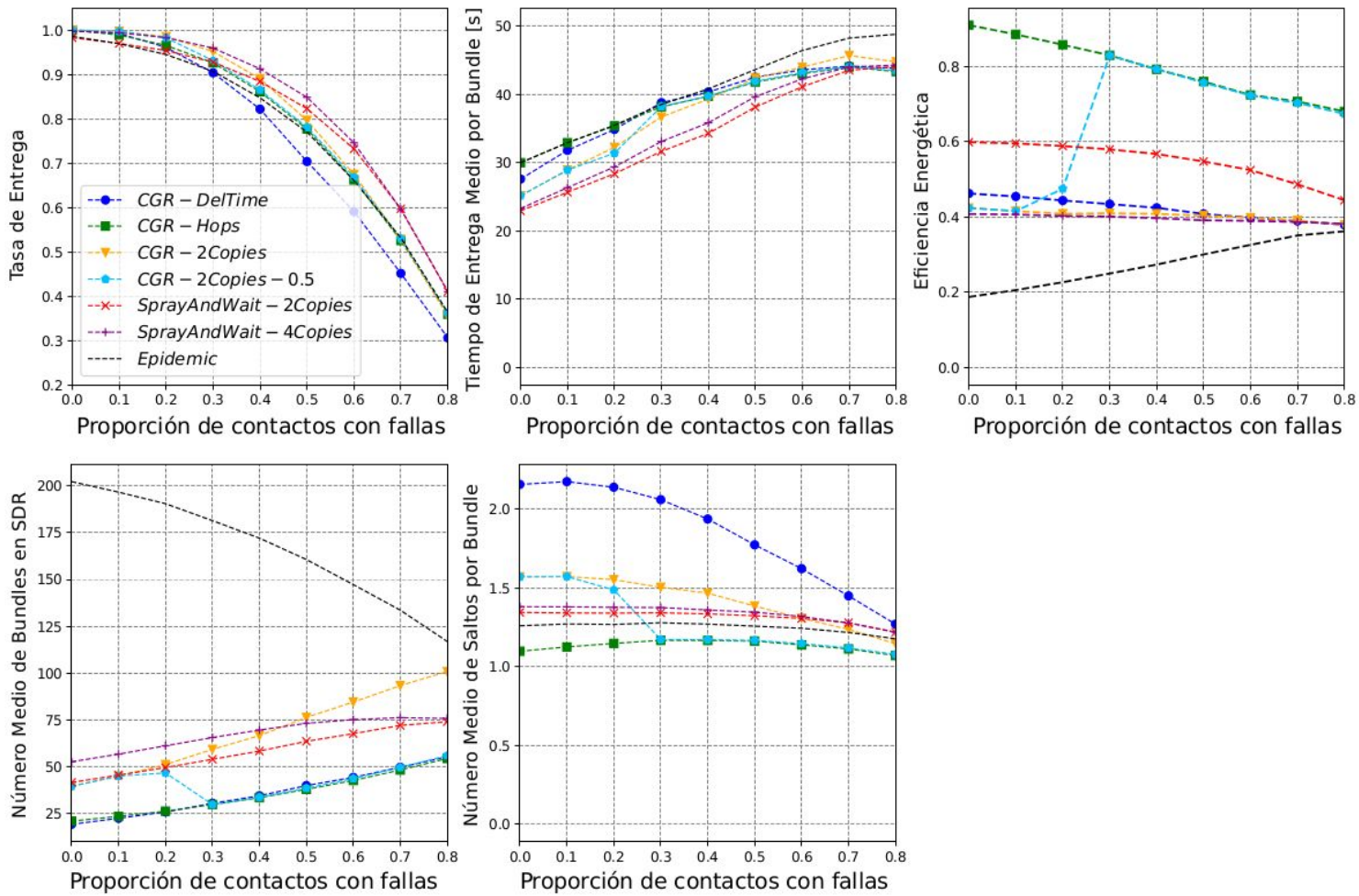


FIGURA 4.9: Resultados de caso de estudio II) redes aleatorias con alta congestión

Incertidumbre

Proporción de Fallas	Recursos Abundantes	Recursos Limitados
Sin Fallas (0,0)	1) *CGR-DelTime*	1) *CGR-Hops*
Baja (0.0 to 0.2)	1) *CGR-2Copies* 2) *CGR-DelTime*	1) *Spray-and-wait-2Copies* 2) *CGR-Hops*
Media (0.2 to 0.4)	1) *Epidemico* 2) *CGR-2Copies*	1) *Spray-and-wait-2Copies* 2) *CGR-Hops*
Alta (0.4 to 0.6)	1) *Epidemico* 2) *Spray-and-wait-4Copies*	1) *Spray-and-wait-2Copies* 2) *CGR-2Copies-0.5*

TABLA 4.1: Preferencia de esquemas de enrutamiento en constelación Walker

Escalabilidad

Algoritmo 8: Inter-RR

input : bundle to forward B, region database Rdb, current node ID $OwnID$
output: passageway node inside local region Pn

```
1  Vents ← getVents(Rdb);
2  SrcRegions ← getTxRegions(OwnID, Rdb);
3  DstRegions ← getRxRegions(B.dstID, Rdb);
4  Pn ← ∅;
5  FoundVent ← false;
6  for v ∈ Vents do
7  |  if v.toRegID ∈ DstRegions then
8  |  |  FoundVent ← true;
9  if FoundVent == false then
10 |  return Pn;
11 Set Vertices contains all created Vertex
12 Struct Vertex contains
13 |  fromRegID;
14 |  toRegID;
15 |  pwayID;
16 |  distFromOrigin;
17 |  visited;
18 |  Predecessor;
19 Vertex NVS("SRC", "SRC");
20 Vertex NVD("DST", "DST");
21 for s ∈ SrcRegions do
22 |  Vertex VS("SRC", s);
23 for d ∈ DstRegions do
24 |  Vertex VD(d, "DST");
25 for v ∈ Vents do
26 |  Vertex Vrt(v.fromRegID, v.toRegID, v.pwayID);
27 for v ∈ Vertices do
28 |  if v == any VS then
29 |  |  v.distFromOrigin ← 0;
30 |  else
31 |  |  v.distFromOrigin ← Inf;
32 |  v.visited ← false;
33 |  v.predecessor ← ∅;
34 regionsRoute ← dijkstra(NVS, NVD, Vertices);
35 Pn ← regionsRoute.firstVertex.pwayID;
36 return Pn;
```