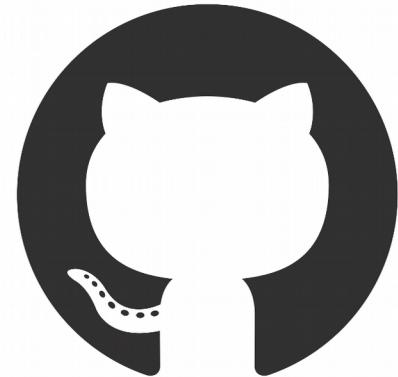
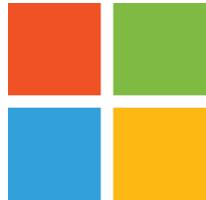


# PyDTN e a comunicação Reptiliana



Matheus Monte –[https://twitter.com/montematheus\\_](https://twitter.com/montematheus_)  
<https://github.com/matheusmonte>

# Who f\*\* are you?



clever**tech**

 BUSCAR

 TECMUNDO



NOTÍCIAS ▾

COMPARADOR DE CELULARES

GAMES

TECMUNDO TV

CUPONS DE DESCONTO

TECMUNDO PRO

## Já existe conexão sem fio na lua – e pode ser mais rápida do que a nossa

POR [BRUNO MICALI](#) | [@micali](#) - EM [INTERNET](#) -  28 MAI 2014 – 17H45

 COMPARTILHAR



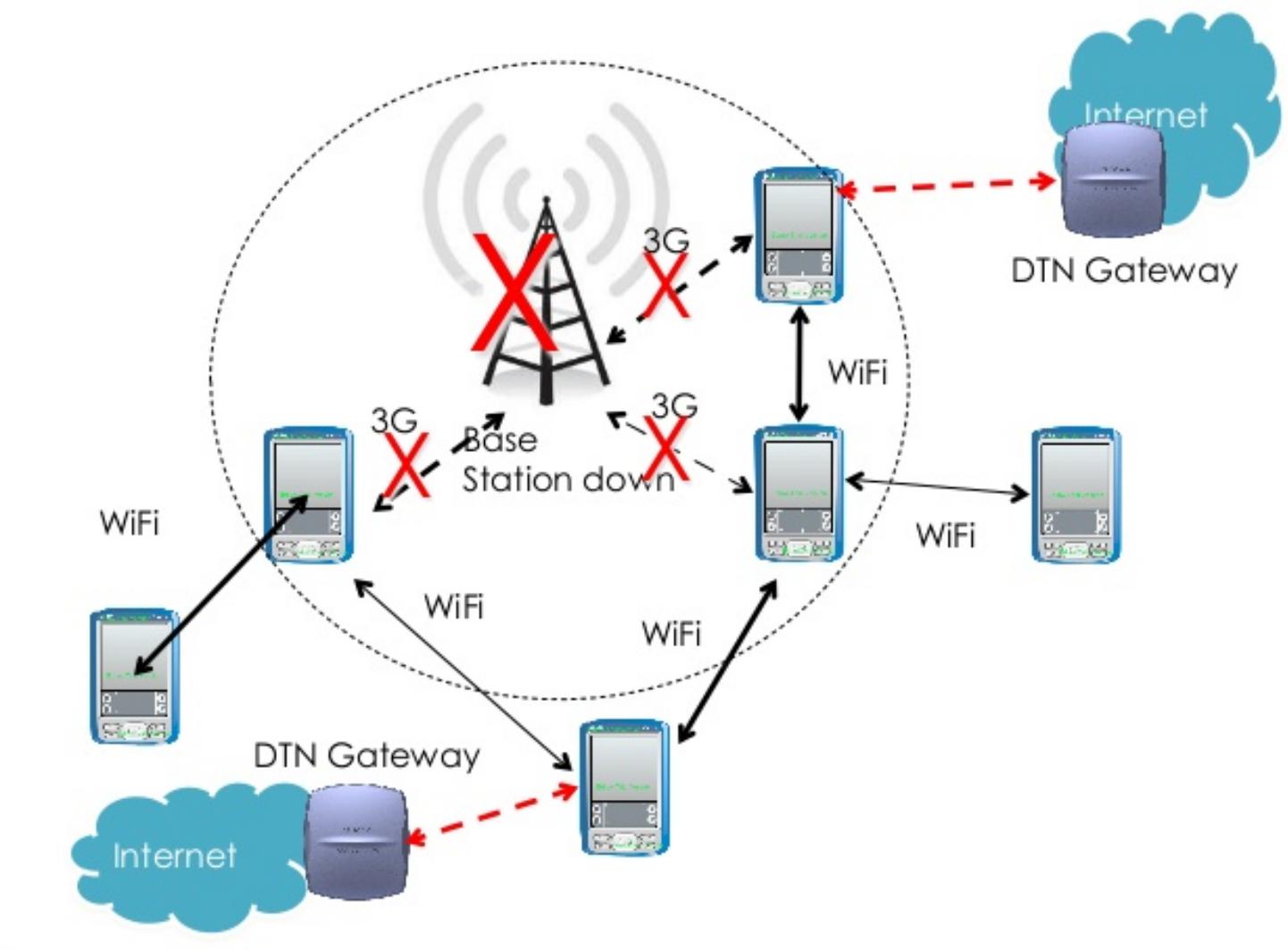
111

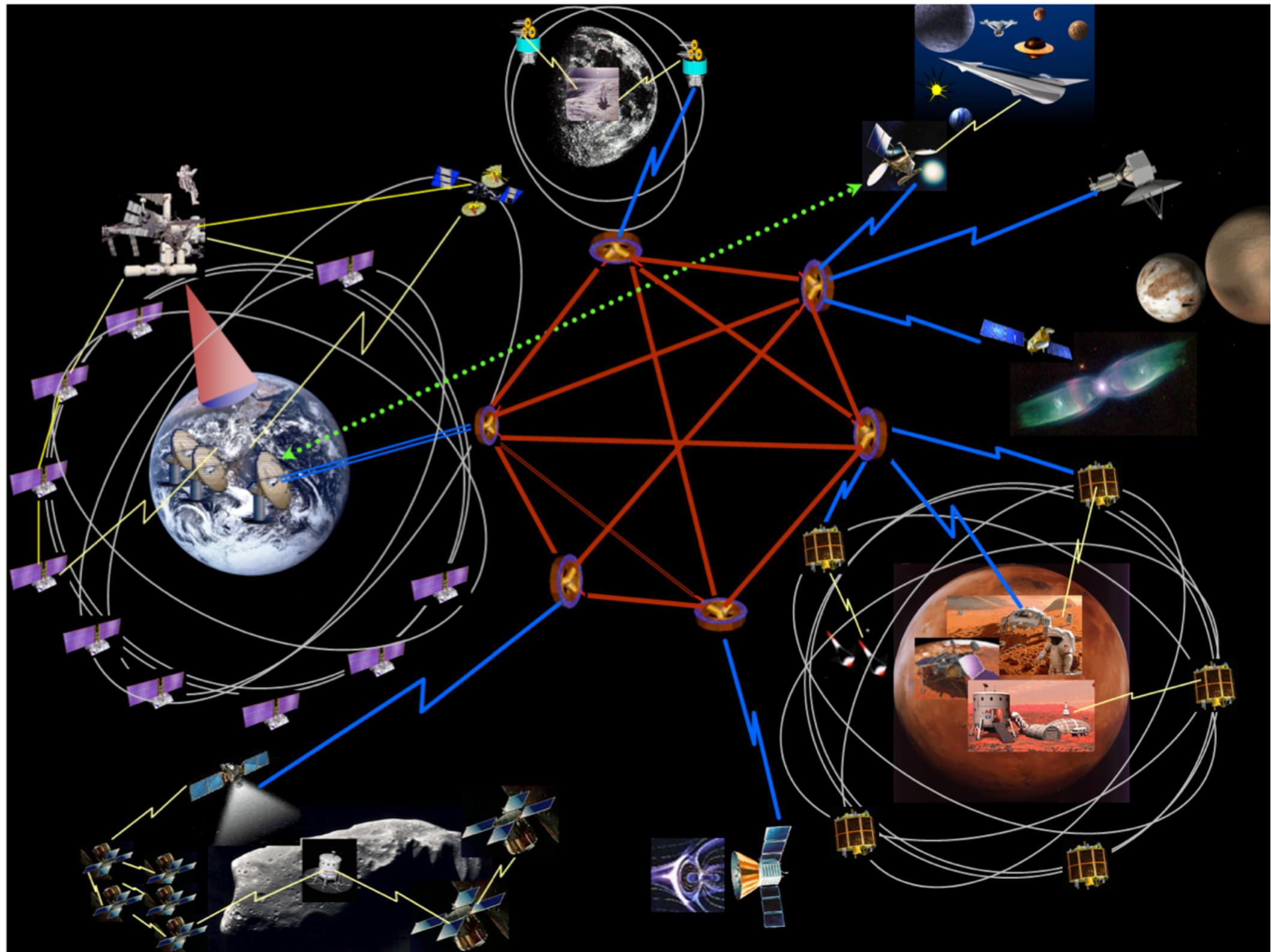
0 compartilhamentos

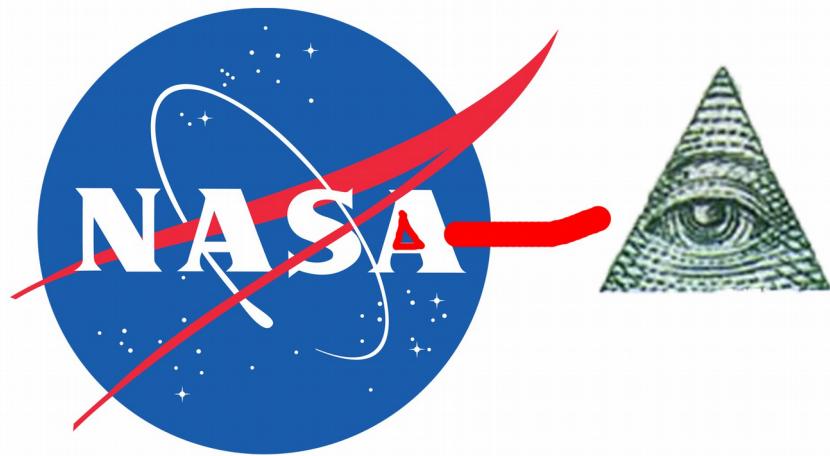




DELAY/DISRUPTION TOLERANT NETWORKING









# IBR-DTN

- Completo
- C++
- Código de engenharia
- Falta de Padrões
- Difícil configuração

# PyDTN

- Baseado em Python <3
  - Padrão de Projeto
- Fácil configuração(Ubuntu)
  - Fácil simulação
- Documentação acessível

- import pydtn
- import pydtn.namtrace
- import pydtn.network
- import pydtn.flowtrace
- import pydtn.storetrace
- import pydtn.storeprofile
- import pydtn.sampleapp
- 
- pydtn.namtrace.setup("example.nam")
- pydtn.flowtrace.setup("example.flow")
- pydtn.storetrace.setup("example.stor")
- pydtn.storeprofile.setup("example.storprof")
- pydtn.config("bundle\_lifetime",[0,9000]);
- pydtn.stats( "stats.out", 1000 )
- pydtn.network.algorithm("forwarding:routed:explicit") # this is the default
- 
- # Create three nodes.
- node1 = pydtn.network.addNode("\0\1",capacity=4096,stable=10000)
- node2 = pydtn.network.addNode("\1\0",capacity=4096,stable=10000)
- node3 = pydtn.network.addNode("\1\1",capacity=4096)
- node2.config("custody","spaceavail")
- node2.config("resend",[0,5000])

- 
- # Create links 1 <--> 2 <--> 3
- link12 = pydtn.network.addLink( node1, node2, latency=0.003, bandwidth=100 )
- link21 = pydtn.network.addLink( node2, node1, latency=0.003, bandwidth=100 )
- link23 = pydtn.network.addLink( node2, node3, latency=0.003, bandwidth=100 )
- link32 = pydtn.network.addLink( node3, node2, latency=0.003, bandwidth=100 )
- 
- # Set up routing.
- pydtn.network.route()
- 
- pydtn.network.applyToNodes(pydtn.storeprofile.collect)
- pydtn.network.applyToNodes(pydtn.sampleapp.attach)
- 
- # Schedule a data event.
- node1.emit([0,1],"data",node3,"hello world")
- node1.emit([0,2],"data",node3,"hello world")
- node1.emit([0,3],"data",node3,"hello world")
- 
-

```
• # Send a message from an application.  
• pydtn.sampleapp.send(node1,node3,"app data",[0,50])  
•  
• for i in range(1000):  
•     node1.emit([0,100],"data",node3,"hello world")  
•  
• for i in range(20):  
•     node1.emit([0,100000],"datafake",node3,10240)  
•  
• def td( now ):  
•     t = now  
•     t[0] += 0  
•     t[1] += 100000  
•     return t  
•  
• def gen():  
•     node1.emit(sim.time(),"datafake",node3,10)  
•  
• tg = sim.Entity("trafficgen")  
• tg.config("time_dist",td)  
• tg.config("generator",gen)  
• tg.emit([0,5000])  
•  
• sim.stopAt([0,600000])  
•  
• # Start the simulation.  
• sim.run()  
•  
• # Stop the interpreter.  
• sys.exit()
```

# Pydtn – A Python/C++ Delay Tolerant Networking Simulator

The 1.0 release of the source code is available for [download](#), and is licensed according to the [GNU General Public License version 3.0 or later](#). If you have doxygen installed on your system, you can generate the source code documentation locally. Otherwise, it is available [here](#).

The [user guide](#) is available under the [Creative Commons Attribution-Share Alike 3.0 Unported License](#).

[Michael Marsh](#), mmarsh at umiacs dot umd dot edu, 2008.

Main Page Modules Namespaces Classes Files Directories Related Pages

## Pydtn Simulator Modules

Here is a list of all modules:

- [pydtn.mobility.accelnode](#)
- [libaodv](#)
- [pydtn.mobility.basicepidemic](#)
- [libdtnaodv](#)
- [pydtn.mobility.epidemic](#)
- [pydtn.flowtrace](#)
- [libmobileforwarding](#)
- [pydtn.mobility](#)
- [pydtn.namtrace](#)
- [pydtn](#)
- [pydtn.aodv](#)
- [pydtn.mobility.rf](#)
- [pydtn.sampleapp](#)
- [sim](#)
- [pydtn.storeprofile](#)
- [pydtn.storetrace](#)

<https://github.com/matheusmonte/AmazoniaPyCon2017Sample>

The screenshot shows a GitHub repository page. At the top, the URL https://github.com/matheusmonte/AmazoniaPyCon2017Sample is displayed in the address bar, along with a search bar and navigation icons. Below the header, there are links for 'Mais visitados', 'System Dashboard', 'Mail', 'iTRIAD Slack', 'React App', 'WhatsApp', 'React App', and 'New Tab'. The main navigation bar includes 'This repository' (highlighted), 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Gist'. On the right side of the header, there are buttons for 'Unwatch', 'Star', 'Fork', and 'Edit'. The repository name 'matheusmonte / AmazoniaPyCon2017Sample' is shown in bold blue text. Below it, there are tabs for 'Code' (selected), 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Settings', and 'Insights'. A summary box displays '4 commits', '1 branch', '0 releases', and '1 contributor'. Below this, a 'Branch: master' dropdown and a 'New pull request' button are visible. To the right are buttons for 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The main content area lists recent commits:

Author	Commit Message	Time Ago
Matheus Clark	Add Samples and PyDTN Simulator	Latest commit ec3e1ba 4 minutes ago
DjangoCOM Azure	/AmazoniaPyC... Add Samples and PyDTN Simulator	4 minutes ago
PyDTN	Add Samples and PyDTN Simulator	4 minutes ago
.gitattributes	Add sample code to be used on Amazonia PyCon 2017 Speak - Django with...	2 months ago
.gitignore	Add sample code to be used on Amazonia PyCon 2017 Speak - Django with...	2 months ago

At the bottom, a call-to-action box encourages adding a README with a 'Add a README' button.



Neymar Jr   
@neymarjr



Seguindo

Deus é TOP

RETWEETS

157.563

CURTIDAS

53.137



14:34 - 23 de jul de 2010



160 mil



53 mil

•••