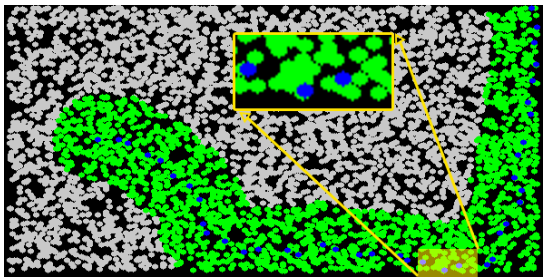
Time-Based Ray Tracing Forwarding in Dense Nanonetworks



Eugen Dedu, Masoud Asghari

Université de Franche-Comté, CNRS, institut FEMTO-ST
Montbéliard, France
http://eugen.dedu.free.fr

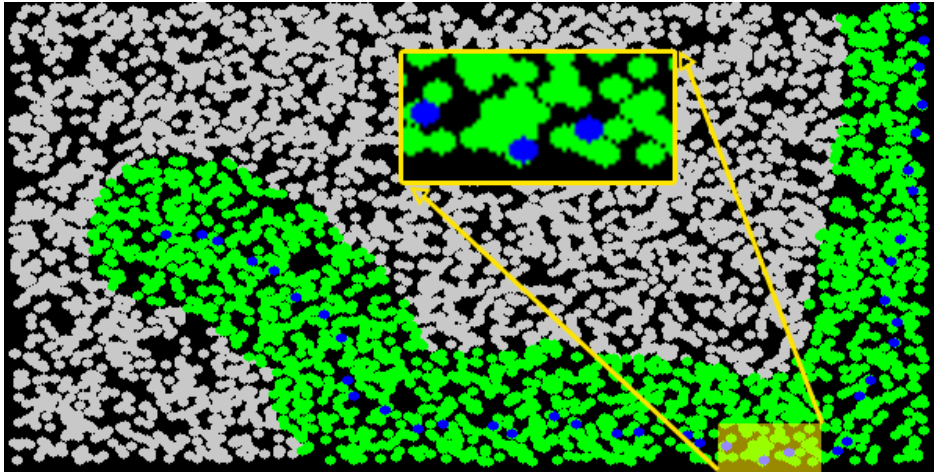
AINA conference, 29–31/03/2023

Context: nanonetworks, TS-OOK modulation

- Nanonetworks are networks whose nodes are nanometric (<10 µm); can have a **high node density**
- The TS-OOK modulation used is very peculiar, as it does not use a signal carrier, but **pulses**
- Bits are sent at T_s interval, e.g. bits of packet 1 are sent at x, $x+T_s$, $x+2T_s$, $x+3T_s$ etc., i.e. at the same time slot x (modulo T_s)
- Times are extremely short, e.g. time slot T_p=100 fs, and T_s=1000 T_p = 100 pc
- Receiver reads at the same T_s interval (and with a short *delay* compared to sender), allowing it to group incoming bits into packets, e.g. bits received at y, $y+T_s$, $y+2T_s$ etc. (i.e. the same time slot y) are grouped in one packet

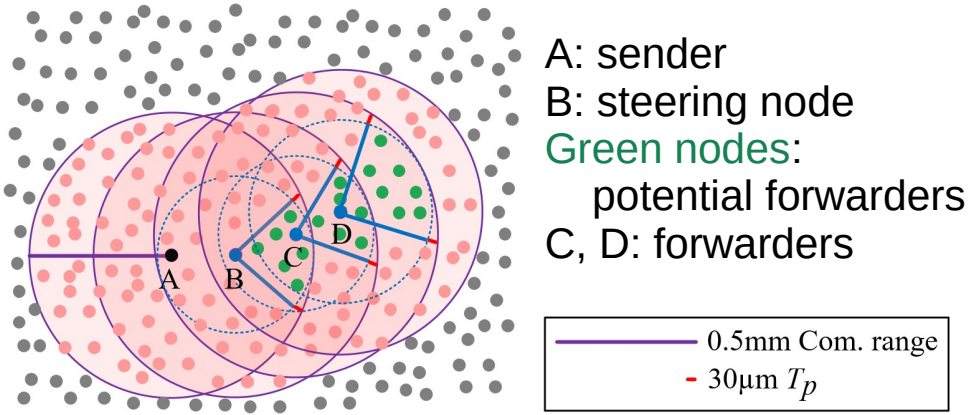
Problem and contributions

- Can we improve multi-hop routing by using bit reception times?
- It is the first time bit reception time and signal propagation duration are used to construct a quasistraight multi-hop path
- Based on this path, we propose a ray tracing forwarding to reduce the number of forwarders
- We implement it, evaluate it, and compare it with related protocols



Ray tracing forwarding principle

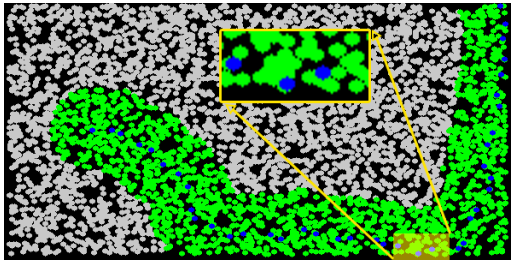
- B has just received the bits from a packet, in time slot x (modulo T_s): x, $x+T_s$, $x+2T_s$ etc.
- B forwards it **at the same time slot** x, i.e. at $y+x$, $y+x+T_s$, $y+x+2T_s$ etc., where y is a delay
- => nodes on right of B (collinear) receive both packets **at the same time slot**
- Because the time slot is non null, the two packets are received in the same time slot **also** by nodes close to collinear nodes (shown in green)



- Nodes having received the first two copies of the same packet in the same time slot are potential forwarders
- Potential forwarders choose a random backoff, and the one with the smallest backoff forwards the packet
- Nodes having received a third copy of the same packet unselect themselves as forwarders (because packet has propagated)
- Note: the ray tracing method does not make any assumption!

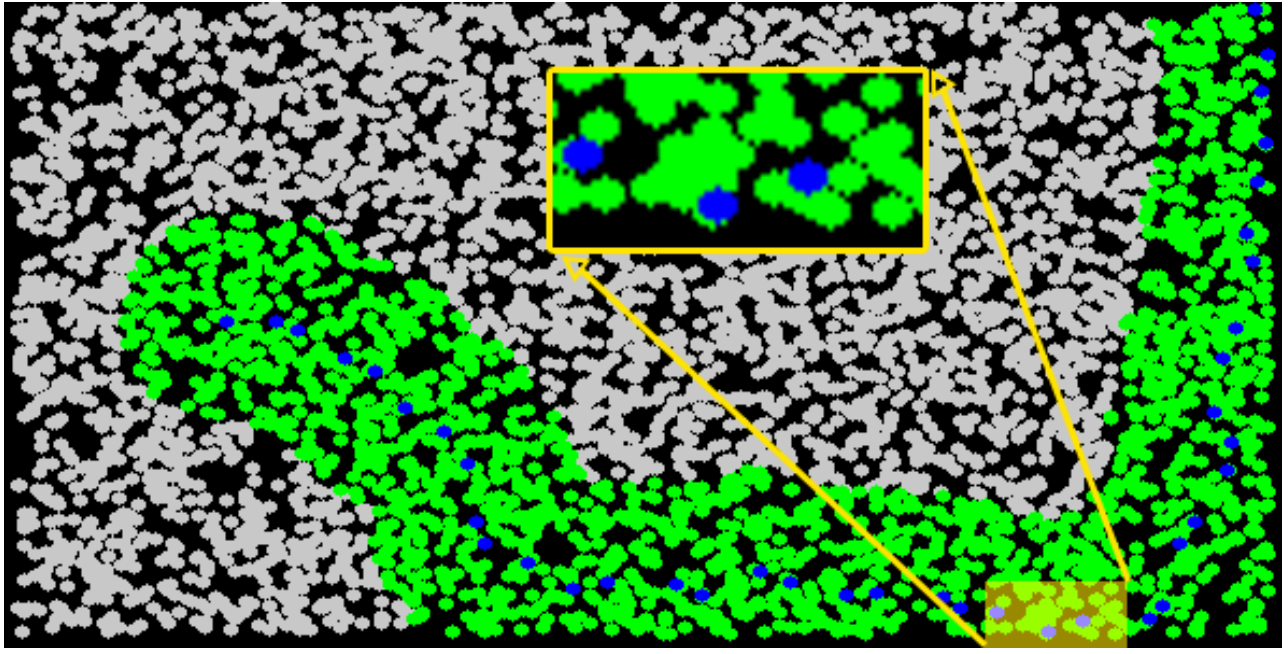
Evaluation

- BitSimulator:
  - implements nanonetworks' peculiarities and TS-OOK
  - highly scalable (e.g. 20 000 nodes)
  - has a visualiser software
- Scenario: network is a rectangular strip, 5002 nodes, 218 neighbours/node
- Full reproducibility Web page

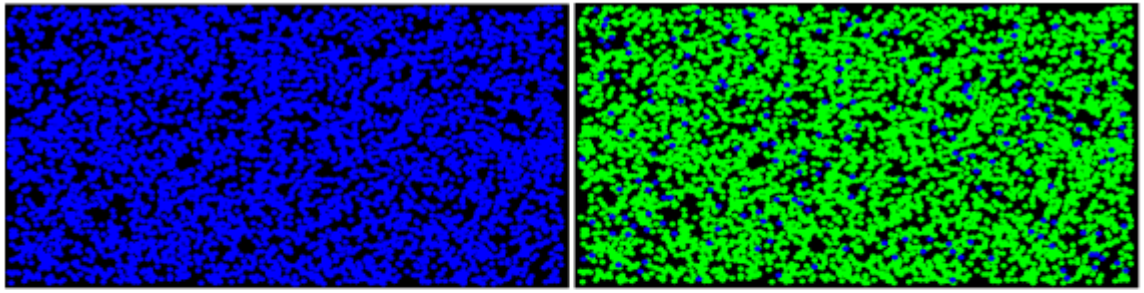


Evaluation – features

- Quasilinear forwarding
- Auto-deviation at borders

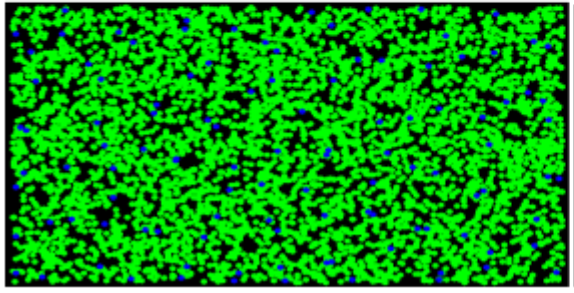


Evaluation – comparison with related coordinate-free methods

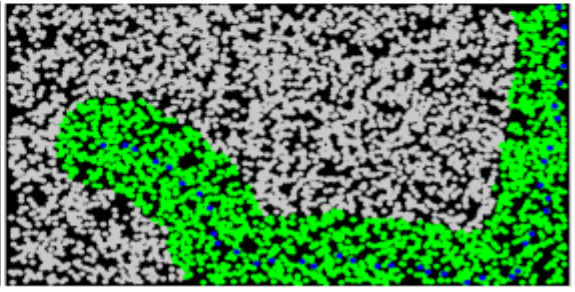


(a) Pure flooding (5002 blue points)

(b) Probabilistic flooding (218 blue points)



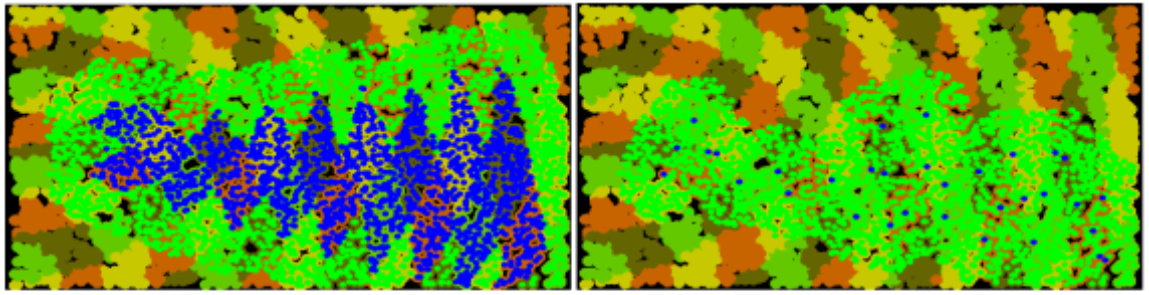
(c) Backoff flooding (110 blue points)



(d) Ray tracing (40 blue points)

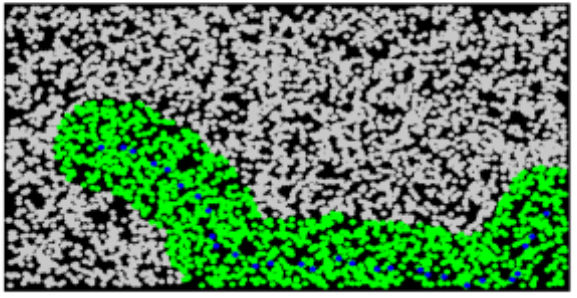
Conclusion: ray tracing method needs fewer forwarders than the other methods

Evaluation – comparison with related coordinate-based methods



(a) SLR (1654 blue points)

(b) Counter-based SLR (33 blue points)



(c) Ray tracing (29 blue points)

Conclusion: ray tracing method needs fewer forwarders than the other methods

Comparison with related work

- Geoforwarding protocols using GPS or triangulation – unusable at this tiny node size (µm)
- *Straight-line routing* protocol assumes that nodes are able to determine the distance to transmitter using signal strength – inappropriate for the short distances between nanonodes, and for nanonodes' basic receivers (pulse either received, or not)
- *Stateless Linear-path Routing* (SLR) needs a coordinate system and has a high redundancy

Conclusions and perspectives

- Bit reception time *can* be used for routing
- Ray tracing forwarding features: quasilinear forwarding and auto-deviation at borders
- It uses fewer forwarders than the other related methods
- Perspectives:
  - less dense and denser scenarios
  - choose steering node automatically based on desired direction
  - improve deviation