Exploiting Sleep-and-Wake Strategies in the Gnutella Network

Salvatore Corigliano and Paolo Trunfio

DIMES - University of Calabria, Italy

CTS 2014
15th International Conference on
Collaboration Technologies and Systems
May 21, 2014 - Minneapolis, MN, USA

Motivations and goal

- P2P architectures are widely used to implement large-scale collaborative networks, including file sharing systems
- Given the large sets of computing resources involved in P2P file sharing networks, their aggregate energy consumption is an important problem to be addressed
- The *sleep-and-wake* approach has been proposed as a general approach to reduce energy consumption in P2P systems
- **Goal:** evaluating how the sleep-and-wake energy-saving approach can be used to reduce energy consumption in the *Gnutella* network

Main contribution

- We introduce a general sleep-and-wake algorithm for Gnutella networks in which
  - All leaf-peers cyclically switch between wake and sleep mode
  - Each leaf-peer autonomously decides the time passed in sleep mode
- We define different strategies that a leaf-peer may employ to decide the duration of its sleep periods
- Such strategies have been evaluated through simulation using the general sleep-and-wake algorithm in different network scenarios

Outline

- Energy-efficient peer-to-peer systems
- Network assumptions
- General sleep-and-wake algorithm
- Sleep duration strategies
  - VAR_HR: duration depends on the *hit rate*
  - VAR_FS: duration depends on the *number of files shared*
  - VAR_QR: duration depends on the *query rate*
  - FIX_*n*WD: duration fixed to *n* times the *wake duration*
- Performance evaluation
- Conclusions

Outline

- **Energy-efficient peer-to-peer systems**
- Network assumptions
- General sleep-and-wake algorithm
- Sleep duration strategies
  - VAR_HR: duration depends on the *hit rate*
  - VAR_FS: duration depends on the *number of files shared*
  - VAR_QR: duration depends on the *query rate*
  - FIX_*n*WD: duration fixed to *n* times the *wake duration*
- Performance evaluation
- Conclusions

Energy-efficient peer-to-peer systems

- Existing systems can be classified under six categories*:
  - **Proxying**: peers can go offline to save energy by delegating some of their activities (e.g. download tasks) to proxies
  - **Task allocation optimization**: energy savings is achieved by deciding which peer will satisfy the request of another peer
  - **Message reduction:** energy consumption is reduced by minimizing the number of messages and the associated processing times
  - **Location-based:** reduces the energy consumed by multi-hop re-transmissions by improving the match between overlay and network
  - **Overlay structure optimization**: improves energy efficiency by controlling overlay topology or introducing new layers to the overlay
  - **Sleep-and-wake:** reduces energy consumption by letting peers cyclically switch between wake and sleep mode

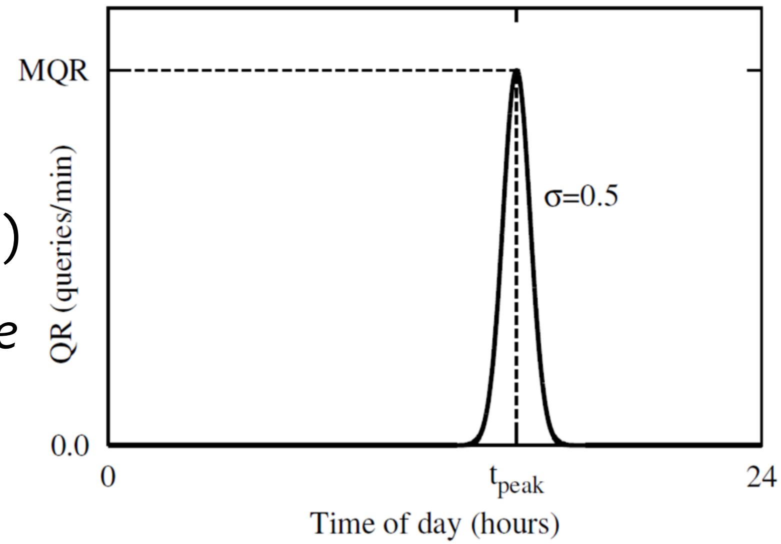
* A. Malatras, F. Peng, B. Hirsbrunner "**Energy-efficient peer-to-peer networking and overlays**" in: M. S. Obaidat, A. Anpalagan, and I. Woungang (Eds.), Handbook of Green Information and Communication Systems, Elsevier, 2013

Outline

- Energy-efficient peer-to-peer systems
- **Network assumptions**
- General sleep-and-wake algorithm
- Sleep duration strategies
  - VAR_HR: duration depends on the *hit rate*
  - VAR_FS: duration depends on the *number of files shared*
  - VAR_QR: duration depends on the *query rate*
  - FIX_*n*WD: duration fixed to *n* times the *wake duration*
- Performance evaluation
- Conclusions

Network assumptions

- Two-layer overlay (Gnutella 0.6):
  - **Top layer** composed of a number of **ultra-peers**
  - **Bottom layer** comprises a higher number of **leaf-peers**
  - Each leaf-peer is connected to a few ultra-peers, while each ultra-peer is connected to several other ultra-peers
  - A leaf-peer submits a query to its ultra-peers, which in turn forward the query to other ultra-peers using a TTL-limited flooding search
- Query submission rate:
  - The inter-generation times are independent and obey an exponential distribution with a given *query rate* (*QR*)
  - The *QR* reaches a *maximum query rate* (*MQR*) at a given time and distributes around it following a Gaussian

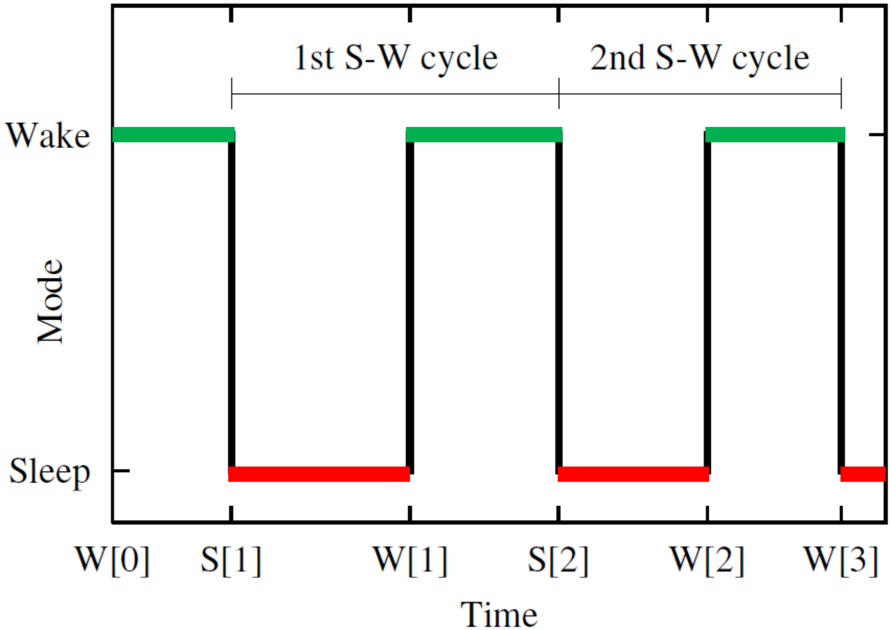


Outline

- Energy-efficient peer-to-peer systems
- Network assumptions
- **General sleep-and-wake algorithm**
- Sleep duration strategies
  - VAR_HR: duration depends on the *hit rate*
  - VAR_FS: duration depends on the *number of files shared*
  - VAR_QR: duration depends on the *query rate*
  - FIX_*n*WD: duration fixed to *n* times the *wake duration*
- Performance evaluation
- Conclusions

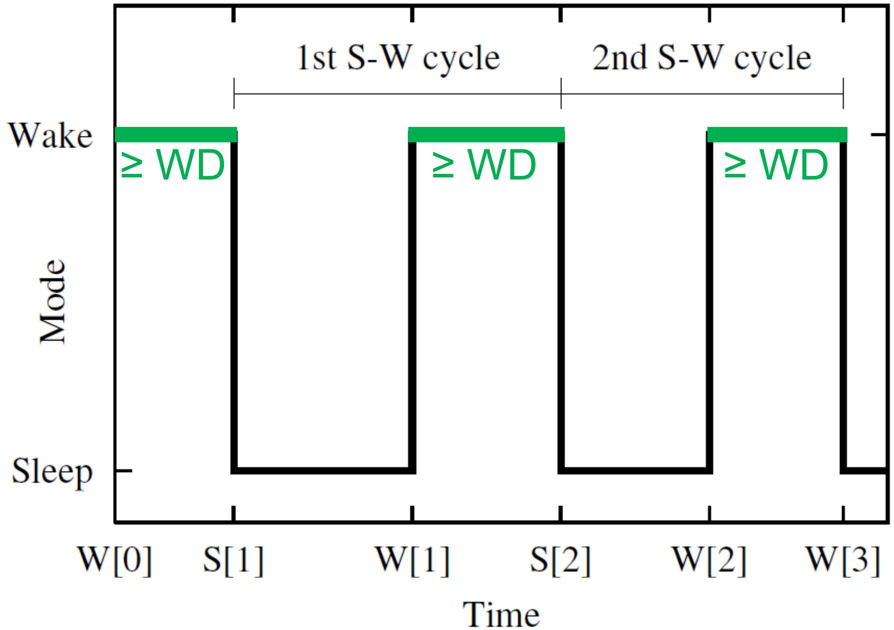
General sleep-and-wake algorithm (1/3)

- Leaf-peers can switch between *wake* and *sleep* mode over the time to reduce energy consumption
  - **Wake mode**: the leaf-peer it is available for download requests and works at normal power level
  - **Sleep mode**: the leaf-peer is unavailable and works at reduced power level



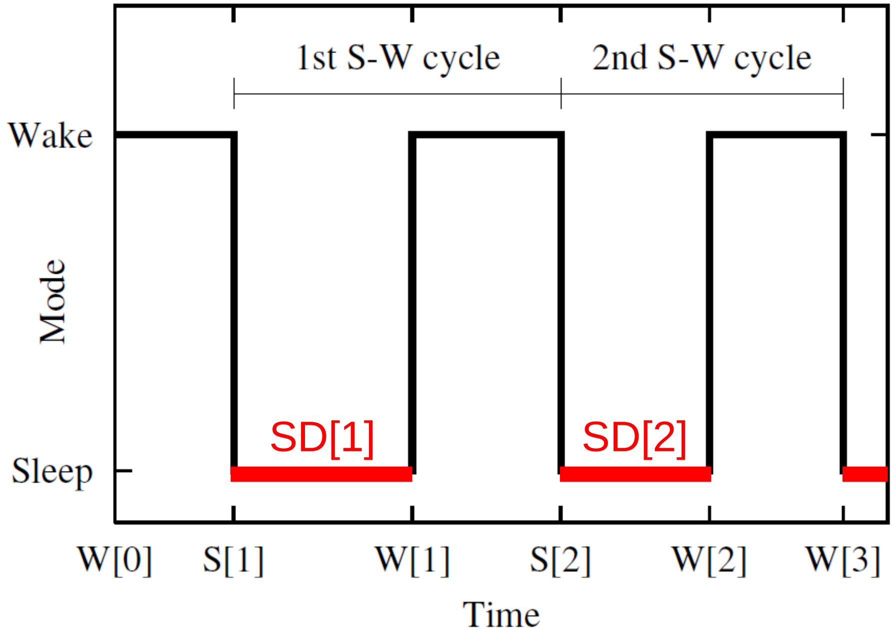
General sleep-and-wake algorithm (2/3)

- The duration of the i-th wake period, i.e. S[i+1]-W[i], is greater than or equal to a constant WD:
  - It is equal to WD if at time W[i] + WD the leaf-peer is not busy with any query processing or file transfer activity
  - Otherwise, the beginning of the next sleep period is deferred and so the i-th wake period will be longer than WD



General sleep-and-wake algorithm (3/3)

- The duration of the i-th sleep period, SD[i], is calculated by the leaf-peer at end of the (i-1)-th wake period based on the specific strategy adopted
  - Variable duration
  - Fixed duration



Outline

- Energy-efficient peer-to-peer systems
- Network assumptions
- General sleep-and-wake algorithm
- **Sleep duration strategies**
  - VAR_HR: duration depends on the *hit rate*
  - VAR_FS: duration depends on the *number of files shared*
  - VAR_QR: duration depends on the *query rate*
  - FIX_*n*WD: duration fixed to *n* times the *wake duration*
- Performance evaluation
- Conclusions

Sleep duration strategies

- Given the general *sleep-and-wake* algorithm, it is possible to define different strategies for deciding the duration of the next sleep period
- We defined and evaluated the following strategies:
  - **VAR_HR**: variable sleep duration depending on the *hit rate*
  - **VAR_FS**: variable sleep duration depending on the number *of files shared*
  - **VAR_QR**: variable sleep duration depending on the *query rate*
  - **FIX_*n*WD**: fixed sleep duration equal to *n* times WD

VAR_HR: Variable with Hit Rate

- **Hit rate** of the i-th wake period of a leaf-peer p, **HR[i]**, is the number of query hits generated by p during the time interval [W[i], t] divided by t - W[i], where t is the ending time of the i-th wake period

- The duration of the i-th sleep period of a leaf-peer p, denoted **SD[i]**, depends on HR[i-1] as follows:

$$SD[i] = \begin{cases} 0\ m & \textbf{if } HR[i-1] > 1\ h/m \\ 5\ m + \frac{10}{HR[i-1]} & \textbf{if } 0.1\ h/m < HR[i-1] \leq 1\ h/m \\ 120\ m & \textbf{otherwise} \end{cases}$$

- Using VAR_HR, the leaf-peers with a high hit rate will not sleep at all or will sleep for a short amount of time, while those with a lower hit rate will sleep longer

VAR_FS: Variable with Files Shared

- With VAR_FS, the duration of the *i*-th sleep period of a leaf-peer *p*, *SD*[*i*], depends on ***FS*[*i*-1]**, which represents the number of files shared by *p* at the end of the (*i* - 1)-th wake period:

$$SD[i] = \begin{cases} 5\ m & \textbf{if } FS[i-1] > 100 \\ 15\ m + \frac{100\ m}{FS[i-1]} & \textbf{if } 1 \leq FS[i-1] \leq 100 \\ 120\ m & \textbf{otherwise} \end{cases}$$

- Using this strategy, the leaf-peers with a high number of files will sleep for a short amount of time, while those with a lower number of files will sleep longer

VAR_QR: Variable with Query Rate

- Differently from the previous strategies, VAR_QR links the sleep duration of a leaf-peer to its client-side behavior, i.e. the *query rate* of the leaf-peer during the previous wake period

- **Query Rate** of the *i*-th wake period of a leaf-peer *p*, denoted ***QR*[*i*]**, is the number of queries submitted by *p* during the time interval [*W*[*i*], *t*] divided by *t* - *W*[*i*]

- Specifically, **SD[i]** in VAR_QR depends on QR[i-1] as follows:

$$SD[i] = \begin{cases} 0\ m & \textbf{if } QR[i-1] > 1\ q/m \\ 5\ m + \frac{10}{QR[i-1]} & \textbf{if } 0.1\ q/m < QR[i-1] \leq 1\ q/m \\ 120\ m & \textbf{otherwise} \end{cases}$$

FIX_1WD and FIX_3WD

- FIX_1WD and FIX_3WD are two blind strategies with which all the sleeps have the same fixed duration (introduced mostly for comparison with the previous strategies).

- Specifically, with FIX 1WD (Fixed to *WD*) the sleep duration is equal to *WD*:

$$SD[i] = WD$$

while with FIX 3WD (Fixed to 3*WD*), the sleep duration is equal to three times *WD*:

$$SD[i] = 3WD$$

Outline

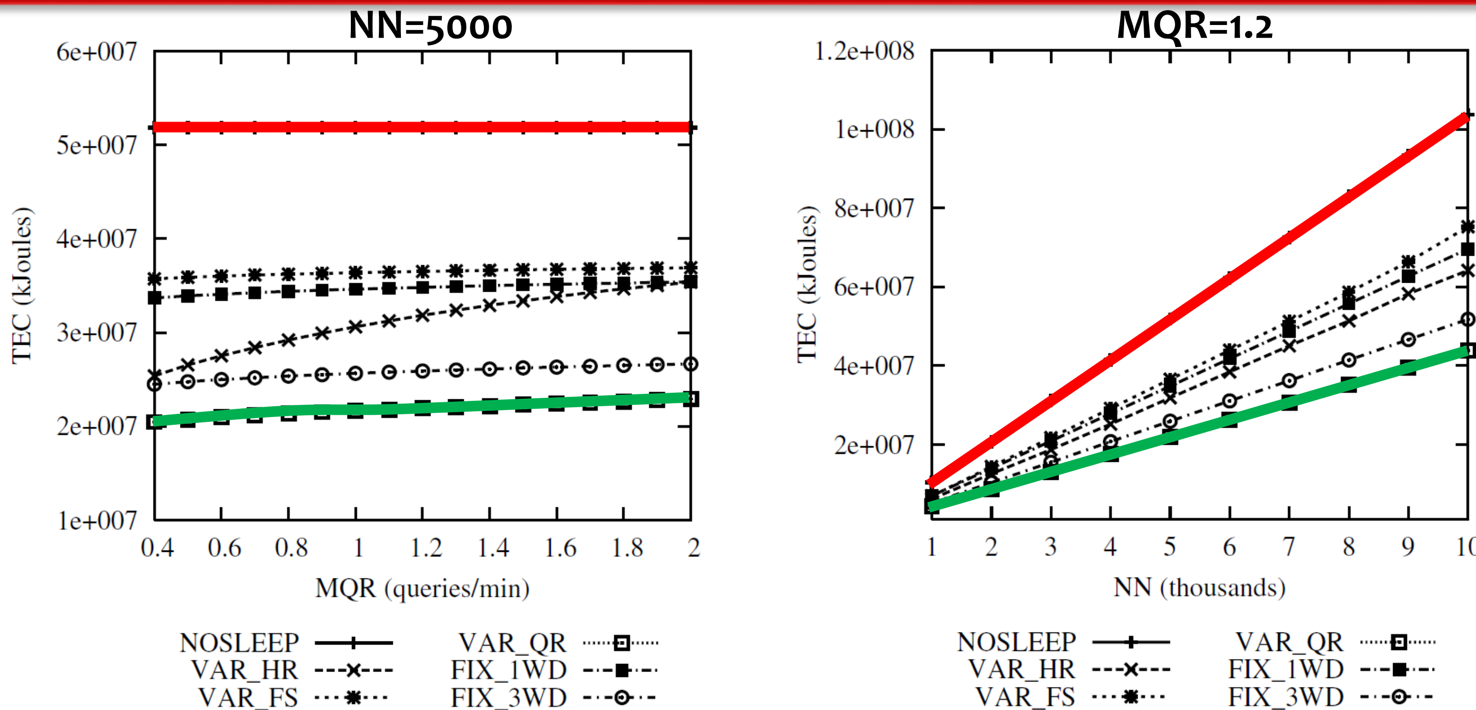
- Energy-efficient peer-to-peer systems
- Network assumptions
- General sleep-and-wake algorithm
- Sleep duration strategies
  - VAR_HR: duration depends on the *hit rate*
  - VAR_FS: duration depends on the *number of files shared*
  - VAR_QR: duration depends on the *query rate*
  - FIX_*n*WD: duration fixed to *n* times the *wake duration*
- **Performance evaluation**
- Conclusions

Performance Evaluation

- The five strategies will be compared with a sixth strategy, referred to as NOSLEEP, in which all nodes are assumed to be always in wake mode
- Performance parameters:
  - *Total Energy Consumption* (TEC) of the network
  - *Hit Rate* (HR), i.e., the fraction of successful queries

Parameter	**Description**	**Values**
NN	Number of nodes	1000-10000
MQR	Max. query rate (queries/min.)	0.4-2.0
SRD	Simulation run duration	24 h
$WMPC$	Wake-mode power consumption of every peer [32]	120 W
$SMPC$	Sleep-mode power consumption of every peer [32]	5 W
$WSTT$	Wake-to-sleep transition time [33]	9 s
$SWTT$	Sleep-to-wake transition time [33]	4 s
TTL	Time-to-live [34]	3
MTT	Message transfer time [35]	20 ms
QPT	Query processing time [36]	2 ms
$QHPT$	Query hit processing time [36]	1 ms
WP	Min. duration of a wake period	20 min.

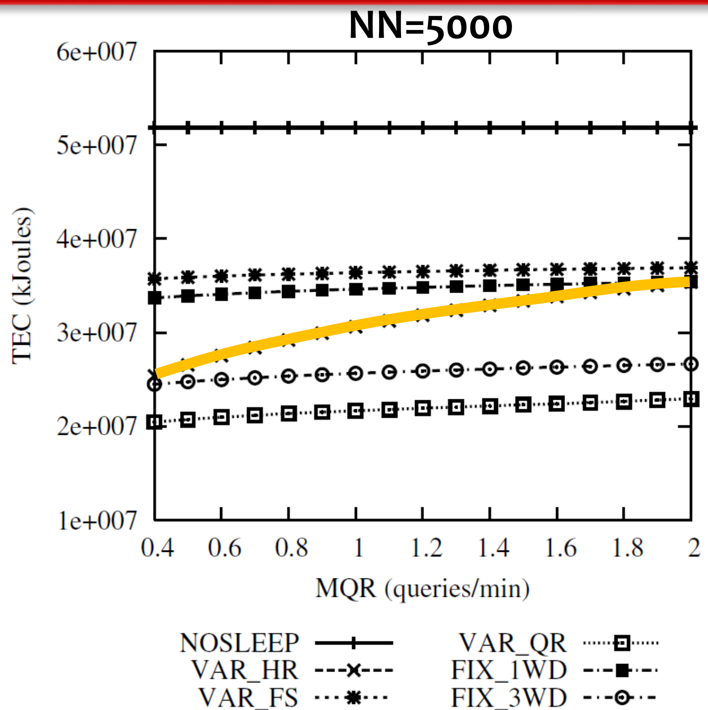
Total Energy Consumption (TEC) (1/2)

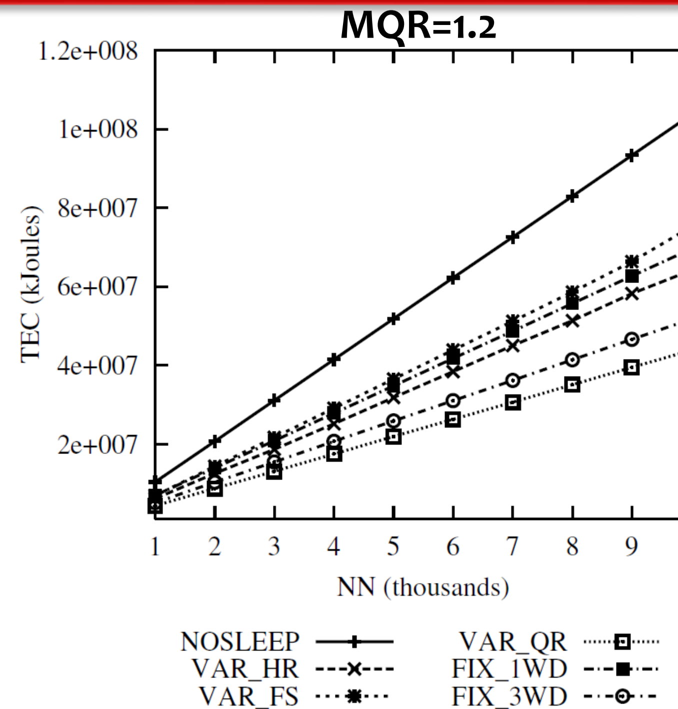


- The best result is obtained with VAR_QR, whose *TEC* is on average 42% of that obtained with NOSLEEP

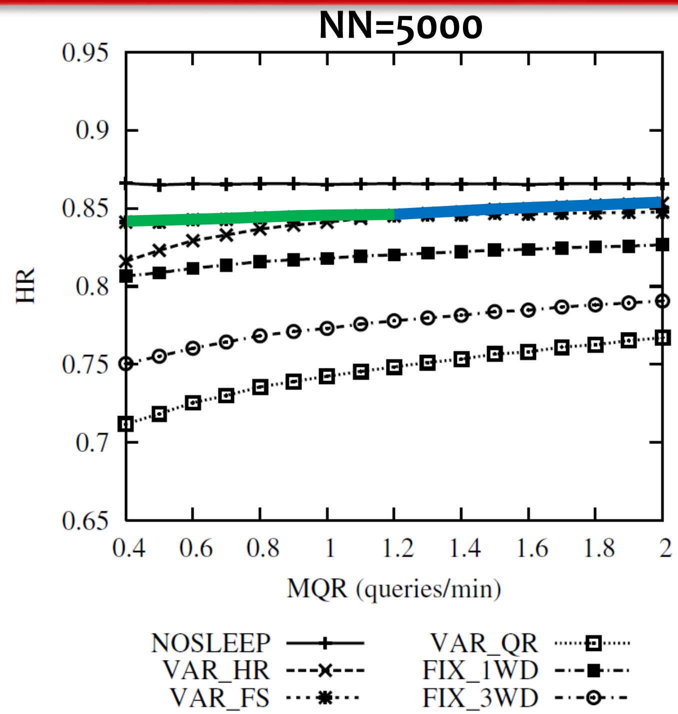
- *TEC* increases linearly with the network size → the absolute amount of energy saved increases significantly as the network grows in size

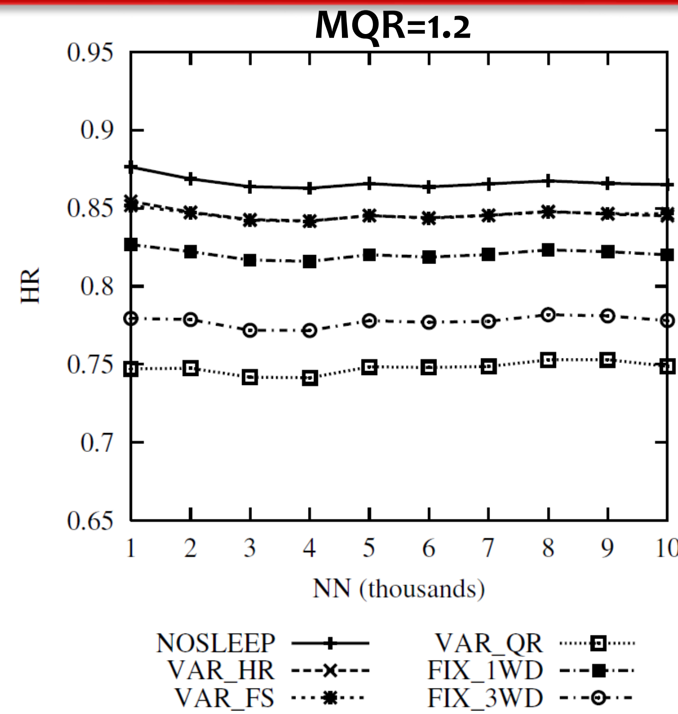
Total Energy Consumption (TEC) (2/2)





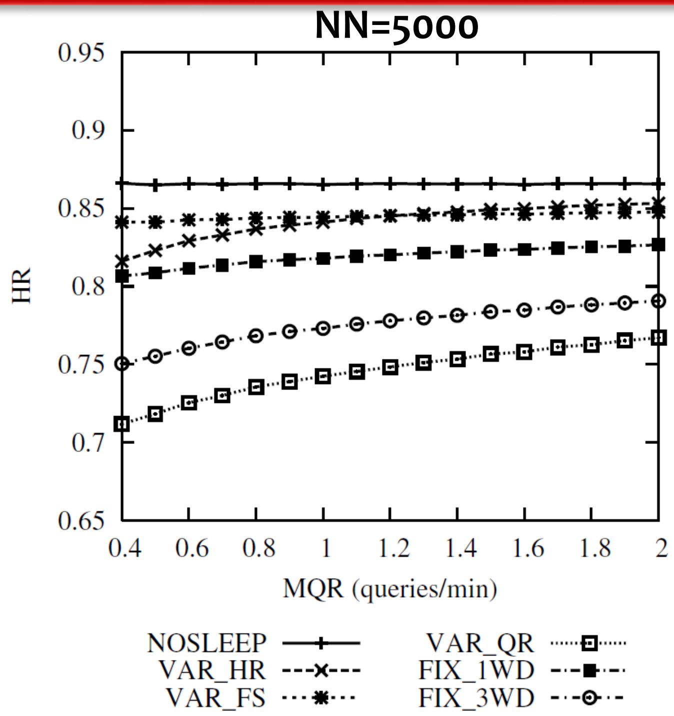
- All the strategies increase their *TEC* as *MQR* increases:
  - High *MQR* values → peers submit more queries → increases the possibility that a sleep will be deferred due to the consequent client- or server-side activity
  - More evident with VAR_HR, because the hit rate grows proportionally with the number of queries submitted to the network

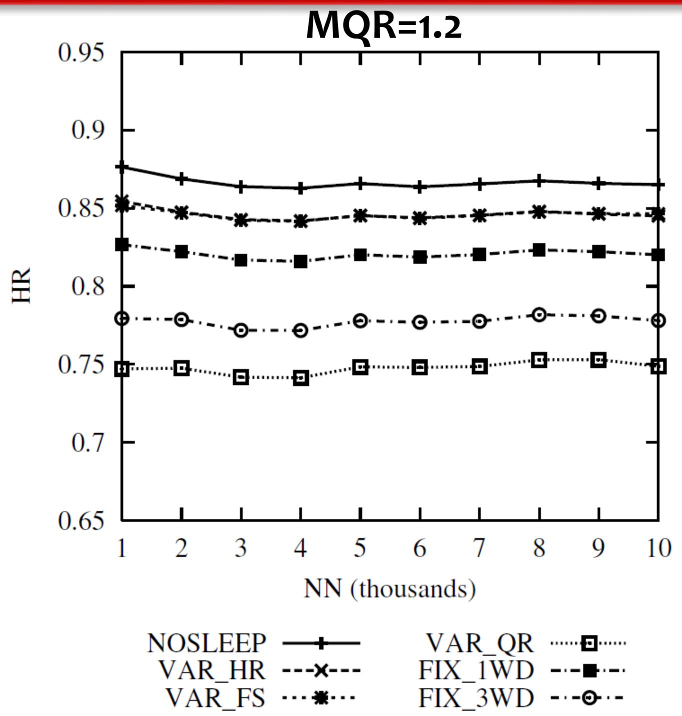




- The highest *HR* is obtained with VAR_FS and VAR_HR: the former with *MQR* < 1.2, the latter with *MQR* > 1.2
- *HR* does not depend on the number of nodes: minor changes of *HR* are due to stochastic variations while the network is created by the simulator

Hit Rate (HR) (2/2)





- All the strategies increase their *HR* as *MQR* increases.
  - As for the case of *TEC*, high *MQR* values increases the possibility that a sleep will be deferred due to the consequent client- or server-side activity
  - This increases the overall time passed in wake mode by the peers and consequently the possibility that a file is available when it is searched

Energy-Search Performance Index (ESPI)

- ESPI is an aggregate performance indicator providing an overall evaluation of a strategy X:

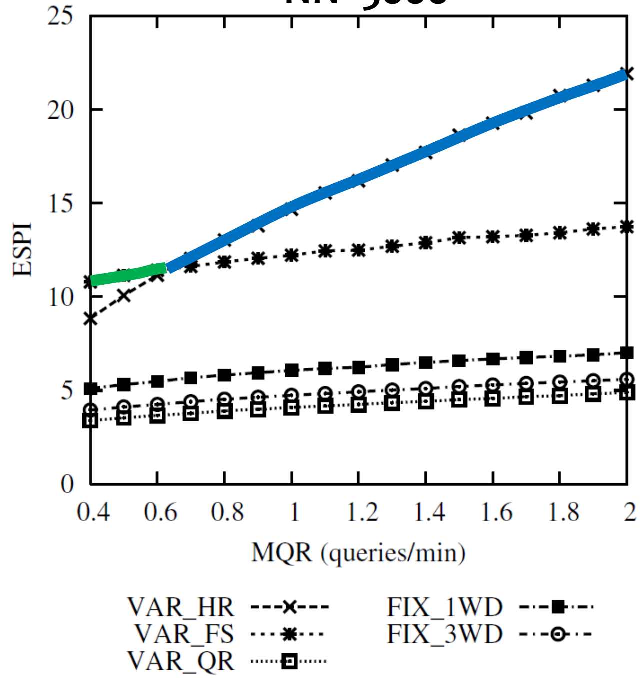
$$ESPI(X) = \frac{\frac{TEC(\text{NOSLEEP}) - TEC(\text{X})}{TEC(\text{NOSLEEP})}}{\frac{HR(\text{NOSLEEP}) - HR(\text{X})}{HR(\text{NOSLEEP})}}$$

  where:

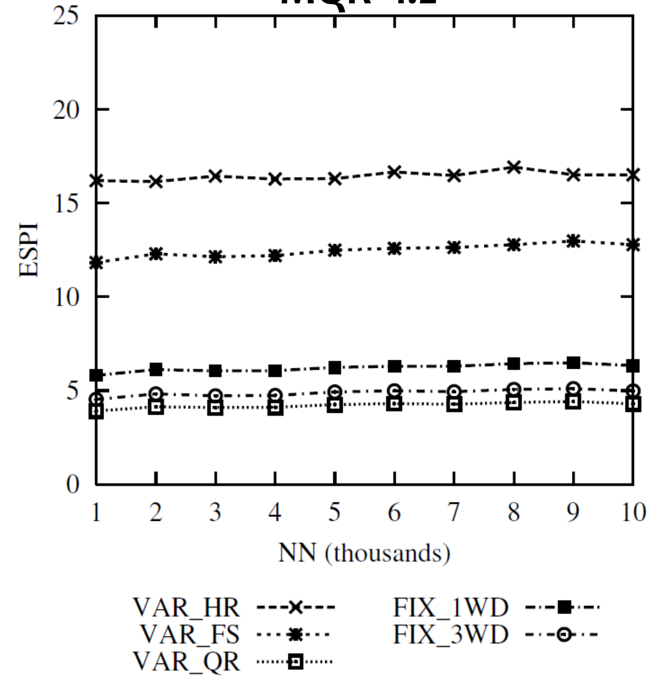
  - TEC(NOSLEEP): is the TEC in the NOSLEEP case
  - HR(NOSLEEP): is the HR in the NOSLEEP case
  - TEC(X): is the TEC of strategy X
  - HR(X): is the HR of strategy X
  - X is in {VAR_HR, VAR_FS, VAR_QR, FIX_1WD, FIX_3WD}

ESPI

NN=5000



MQR=1.2



- The best strategies, based on their *ESPI* values, are VAR_FS and VAR_HR: the former with *MQR* < 0.8, the latter with *MQR* > 0.8.
- As expected, the result is independent from network size.

Conclusions

- Use of the sleep-and-wake energy-saving approach for reducing reduce energy consumption in Gnutella
- A general sleep-and-wake algorithm in which
  - All leaf-peers cyclically switch between wake and sleep mode
  - Each leaf-peer autonomously decides the time passed in sleep mode
- Different strategies that a leaf-peer may employ to decide the duration of its sleep periods have been tested.
- Simulation results have shown that the best performing strategy (VAR_QR) consumes on average 42% of the energy consumed in a network where leaf-peers are always online