Comparison of Policies for Epidemic Broadcast in DTNs under Different Mobility Models

Francesco Giudici, Elena Pagani, Gian Paolo Rossi

Information Science and Communication Department,

Università degli Studi di Milano, Italy,

{fgiudici, pagani, rossi}@dico.unimi.it

Abstract

The broadcast diffusion of messages in Delay Tolerant Networks (DTNs) is heavily dependent on nodes mobility, since protocols must rely on contact opportunities among devices to diffuse data. This work is a first effort to study how the dynamics of nodes affect both the effectiveness of the broadcast protocols in diffusing data, and their efficiency in using the network resources. The paper describes three control mechanisms. The mechanisms characterize a family of protocols able to achieve some awareness about the surrounding environment, and to use this knowledge in order to keep the broadcast overhead low, while ensuring high node coverage. Simulation results allow to identify the winning mechanisms to diffuse messages in DTNs under different conditions.

KEYWORDS: wireless technologies; networking; distributed systems; delay tolerant networks; opportunistic networks; broadcast; performance evaluation

Introduction

The mobile nodes of a delay tolerant network, or DTN (DTN Research Group, 2008), experiment intermittent connectivity and network partitions. So far, in such a critical scenario, the research mainly focused on the problem of providing unicast communications (e.g. (Burgess, Gallagher, Jensen, & Levine, 2006; Davis, Fagg, & Levine, 2001; Jones, Li, & Ward, 2005; Juang, Oki, Wang, Martonosi, Peh, & Rubenstein, 2002; Spyropoulos, Psounis, & Raghavendra, 2004)). By contrast, the one-to-all communication scheme has not received the same attention despite the fact that its service is strategic to support protocols at both application and routing levels. For instance, a broadcast service is required to diffuse scoped advertisements – e.g. about available services or events – and summaries (Lee, Magistretti, Zhou, Gerla, Bellavista, & Corradi, 2006), to support podcasting (Lenders, Karlsson, & May, 2007), or to diffuse acknowledgments, or cure, packets (Harras, Almeroth, & Belding-Royer, 2005). In a DTN, broadcast can be designed by adopting one of the gossip-based mechanisms that have been proposed in the literature in a few slightly different alternatives by starting from the following *PUSH-based* scheme: when a node has a message *m*, it forwards *m* to one or more (possibly all) neighbors it happens to encounter while moving. The forwarding, elsewhere called infection or epidemic, can be either performed periodically (Montresor, Jelasity, & Babaoglu, 2005) or whenever a contact occurs (Vahdat, & Becker, 2000). Infection can continue up to either the message lifetime or a given number of transmissions. The PUSH-based algorithm is effective in achieving a node coverage arbitrarily close to 1 with low latency, but fails in doing this efficiently. Indeed, nodes perform epidemic forwarding with no knowledge about the state of the encountered nodes and, as a consequence, they often happen to forward a message to already infected nodes.

The primary focus in the design of a – both efficient and effective – broadcast protocol is to increase the node likelihood of delivering a message only to uninfected nodes. There are several, growing levels of knowledge a node could achieve about the neighbors state and to approximate the global system state. Whatever is the followed approach to achieve efficiency, the performance of the algorithms is greatly influenced by the mobility patterns that nodes follow (Camp, Boleng, & Davies, 2002). The comparative analysis of the effects of mobility on protocols deserves more attention and, as far as we know, has never been applied to broadcast. This paper moves into this research track and provides some interesting contributions to understand broadcast delivery over DTNs under different mobility conditions. Firstly, the paper defines a family of broadcast protocols, obtained from the PUSH-based approach through the introduction of adaptive mechanisms whose purpose is the improvement of the node awareness about the level of infection in the neighborhood. Secondly, the performances of the different mechanisms are analyzed in three basic mobility model: the classical random waypoint, the swarm mobility, and the aggregation model where nodes move throughout aggregation points. The main contribution of the paper is twofold: *(i)* it identifies and characterizes the winning mechanisms to diffuse messages in DTNs under different conditions, and *(ii)* it is a first attempt to move toward the design of an autonomic and situational algorithm able of autonomously adapting its parameters according to the mobility context the node is moving through, in order to optimize performances.

System and Mobility Models

System Model

The scenario we consider in this paper includes people walking in a limited urban area, such as a *campus area*, and equipped with wireless portable devices. No base stations are assumed and the communication between a source *s* and a destination *d* may eventually occur through either direct contact, when, for instance, node *d* moves into the range of *s*, or indirect contact, when one or more relaying nodes help to create the multi-hop path toward the destination and the last of them finally enters the range of *d*. The devices have a unique identifier, are not required to have positioning capabilities on board and, to meet resource saving requirements, are supposed to adopt a short radio range to communicate. This latter point, together with the fact that devices can be sparsely distributed over a large area, makes high the probability of network partitions and link disruption. Throughout the paper we only assume that each mobile device, or node, periodically broadcasts a *beacon* message in its radio cell. Beacons are used to discover other devices in the neighborhood and their content is limited to the device identifier. In such a scenario, people mobility might follow either a Random Waypoint (RWP) model (Camp et al., 2002) or a more structured motion.

Mobility Models

A great deal of research is currently ongoing in order to characterize mobility models suitable for DTNs. Mobility could be extracted from traces of nodes contacts in real settings; several traces are for instance provided by the CRAWDAD community (CRAWDAD, 2008). Yet, the use of traces with simulators creates some problems. Their timescale is hardly scaled to the simulation time and they generally model the specific behavior of a given mobility scenario, thus loosing the generality required during the protocol design phase. Traces are more likely useful during the validation process than during the design and performance analysis phase.

In this work, we analyze how three basic mobility models affect the performance of the epidemic protocols described in the next section. The classical *Random Waypoint* (RWP) model is not realistic, but it is simple, often provided within network simulators, and the most commonly used in the literature. In the *aggregation* (AGG) model, a node moves toward an aggregation point (*ap*), chosen from a set according to a certain probability distribution P_{AGG}, and once there it pauses for a time t_{AGG} before selecting the next *ap*. This model reproduces the mobility of users who may group in interest points according to some spatial or functional rule. Parameters of this model are also the number and position of *ap*s over the area, the speed range $[v_{min}, v_{max}]$ and the radius of the *ap*s. The distance of a node from the center of an *ap* is determined according to a Rayleigh distribution with standard deviation σ_{AGG}. In the *swarm* (SWR) model, nodes move in a coordinated way. Each swarm has a logical center, which moves toward a destination chosen randomly. Once there, the nodes in the swarm stop for a pause time t_{SWR} before moving to a new destination. Each swarm has a number of nodes determined by a probability distribution with mean μ_{SWR} and standard deviation σ_{SWR}. Nodes in a swarm move randomly around its logical center, within maximum distance d_{SWR}. If the regions of two swarms overlap, nodes in the intersection may choose to migrate to the other swarm according to a probability M_{SWR}.

The RWP model reproduces sporadic encounters of two (or a few) nodes. The AGG model reproduces the encounters of many nodes, with nodes experiencing relevant neighborhood changes every time they enter an *ap*. The SWR model reproduces nodes maintaining the same neighborhood for a long time, with sporadic encounters with other (groups of) nodes. Analyzing these three basic models allows to bring into evidence the effects each of them separately has on message diffusion. However, mobility of people in a DTN is more likely modeled by the combination of the above three patterns. A current research topic is producing a synthetic model that addresses these issues or adopts statistical distributions inferred from real traces

(CRAWDAD, 2008). A first attempt in this direction is represented by (Pedersini, Grossi, Gaito, & Rossi, 2008); as a future work we intend to conduct performance measures with this model.

Whatever is the adopted mobility model, it should be assumed that the following *mobility assumption* applies: when a contact occurs, the reciprocal speed is such that the two nodes can set up a communication channel and a significant amount of data is exchanged before they become disconnected. This assumption is reasonable according to results achieved by observations reported in (Hui, Chaintreau, Scott, Gass, Crowcroft, & Diot, 2005; Su, Chin, Popinova, Goely, & de Lara, 2004), but can be occasionally violated in our simulations.

Family of Broadcast Protocols

Given a general DTN scenario as described in the previous section, purpose of this paper is the comparative analysis of a family of topology-independent broadcast protocols under different mobility conditions. Protocols are obtained by the incremental introduction of adaptive and situational mechanisms that progressively augment the protocol capability to adapt to changing conditions. Purpose of the paper is the evaluation of the role played by the different mechanisms and parameter settings to ensure broadcast *effectiveness*, i.e. the capability of the protocol of eventually achieving node coverage arbitrarily close to 1, and *efficiency*, i.e. the capability of the protocol of keeping the generated broadcasts-per-message as close as possible to $O(n \ln n)$ (Cooper, Ezhilchelvan, & Mitrani, 2004).

Each broadcast message *m* is supposed to have a *"scope"* that is defined by the source and specified through a *lifetime*; when this time expires, a node deletes its copy of *m* and stops its diffusion. In the following, we assume long lived messages to better understand the broadcast behavior independently of other constraints.

The basic broadcast protocol to start with is represented by the PUSH-based algorithm:

P-BCAST: a node *p*, holding a message *m*, starts a forwarding of *m* with probability $Prob_p=1$ whenever a node enters its radio range.

This algorithm guarantees node coverage approximately close to 1 with low latency, but generates a redundant load of duplicates. This is motivated by the fact that nodes perform epidemic forwarding with a very limited knowledge about the state of the encountered nodes and, as a consequence, they often happen to forward the message to already infected nodes. It is possible to improve P-BCAST behavior by adding some autonomic capabilities that extract an approximation of global state from locally observed data. We characterize a taxonomy of approaches, dividing them in two classes according to the amount of knowledge about the system state the nodes maintain.

Zero-Knowledge Approaches

In order to reduce duplicate diffusions, the protocol design must respond to the following questions: *(i)* how to provide a node with information about the state of its neighbors to enable the forwarding control? *(ii)* how to ensure the termination of the forwarding algorithm (*stop condition*) when the message has been delivered to the entire population of nodes? A viable approach to simply answer to point *(ii)* is to stop the forwarding when a message copy count has been reached:

CC-BCAST: a node *p*, holding a message *m*, starts a forwarding of *m* with probability $Prob_p=1$ whenever a node enters its radio range. Every time *m* is forwarded, a local copy count is incremented; when it reaches a threshold τ, forwarding is stopped.

The choice of τ influences the capability of achieving high coverage; it can be obtained either from the analysis of mobility traces in the target environment (Chaintreau, Mtibaa, Massoulie, & Diot, 2007), or – under restrictive assumptions – derived analytically. Cooper et al. (2004)

showed that, under the condition of *a well known node cardinality n and RWP*, each node can stop forwarding when a copy count of $\tau = 2 \lceil ln\ n + \gamma \rceil$ broadcasts is reached, with γ the Eulero-Mascheroni's number.

The drawback of the previous approach is that nodes do not keep trace of the infected nodes: if *p* contacts *q* multiple times, it forwards the message every time, thus generating duplicates and wasting its budget of transmissions. As an alternative, a *self-adaptive mechanism* could be adopted such that a node *p* could send *m* only when the percentage of neighborhood changes with respect to the previous diffusion exceeds a threshold *Nth*, and it should be able to tune the value of $Prob_p$ according to the delivery status of *m* in its current area: when most nodes in the neighborhood have delivered *m*, *p* should reduce $Prob_p$ and vice versa. In a zero-knowledge paradigm, *p* is unable to obtain this information; however, *p* can derive the symptoms of the delivery status from sensing the events generated by its encounters, i.e. counting the received duplicates. Each node maintains a local view of its neighborhood at the time it performs a forwarding, by exploiting the underlying beaconing. The mechanism works as follows:

SA-BCAST: a node *p*, holding a message *m*, starts a forwarding of *m* with probability $Prob_p$ whenever the number of neighborhood changes w.r.t. the previous forwarding is greater than a threshold *Nth*. $Prob_p$ is dynamically adapted within a range [*MINP, MAXP*] according to a function $\mathcal{F}$ that decreases $Prob_p$ every time a duplicate is received, and increases $Prob_p$ when *m* is received from a process that has been infected less than *t* time units before.

In (Giudici, Pagani, & Rossi, 2009), it has been shown that this simple mechanism works properly to limit the number of duplicates under high coverage conditions. However, SA-BCAST might to not reach a *stop condition* (in case *MINP*>0), when the approach has to ensure a drip feed of message transmissions to manage late node joins and temporary partitions. A non-terminating protocol has severe effects on the efficiency, which can be only mitigated by adding more knowledge to the nodes about their neighborhood.

Local-Knowledge Approaches

The introduction of an encounters' history can provide the extra information required. A history mechanism has been recently adopted either to compute utility functions to control unicast forwarding (Boldrini, Conti, Iacopini, & Passarella, 2007; Burns, Brock, & Levine, 2005; Dubois-Ferriere, Grossglauser, & Vetterli, 2003), or as the log of infection events to control broadcast forwarding. In this work, we assume that the node cardinality n is known (we discuss this assumption in the Performance Evaluation section). The basic history-based mechanism is obtained from P-BCAST by adding the following statement:

HP-BCAST: whenever a node p forwards or receives the message m, it adds to a local data structure $history_p$ the list of its current neighbors or m's sender respectively. Initially, $history_p$ only contains p itself. p broadcasts a message m if its current neighborhood has nodes other than those held in $history_p$. It skips forwarding otherwise. p terminates the epidemic algorithm when $history_p$ contains n entries.

In the basic HP-BCAST (indicated as HP-BCAST$_0$), each node maintains a local history. The simple evolution of this algorithm is obtained by enabling a node to exchange its local history with its neighbors. This policy accelerates the node's awareness about the epidemic diffusion and should provide a more effective forwarding control. We obtain this new algorithm from HP-BCAST$_0$ by adding the following statement:

HP-BCAST$_{100}$: when p forwards m then it piggybacks (the 100% of) its $history_p$ on m. When p receives m from q, it merges $history_p$ with $history_q$.

It is worth to notice that the described HP-BCAST$_{100}$ algorithm is not as aggressive as it could in its attempts of propagating the infection knowledge. In fact, nodes could achieve a much quicker awareness about the evolution of the infection by piggybacking the history on the beacons.

Several aspects have yet to be explored. There is the need of defining the amount of history a node should maintain and/or exchange, of evaluating the real performance advantages it guarantees, of verifying its validity under mobility models other than RWP, and of exploring the scalability issues the history involves. Part of these arguments are considered in the Performance Evaluation section.

Local Suppression. The described mechanisms have been mainly designed to properly work under both RWP and AGG mobility models. By contrast, the SWR model generates some new issues that deserve specific attention. In fact, with this setting, we will show in the next section that 100% coverage is hard to reach, while a high number of duplicates might be generated. In order to achieve high effectiveness while maintaining an acceptable efficiency, a mechanism of *local duplicate suppression* (LS) can been added to HP-BCAST$_{100}$. When a beacon from a new neighbor is received, a node *p* sets a timer slightly larger than the beacon period. When the timer expires, the message is sent, with the history *including all the nodes that are new neighbors*. If another infected node with the timer set receives this message, it schedules its own transmission only if it has new neighbors not included in the history. This way, diffusion is not prevented for nodes that connect two swarms or two different groups in a swarm, but multiple infections of the same nodes are avoided.

Merging the Approaches

All the described basic mechanisms can be combined, thus leading to the complete family of broadcast protocols shown in fig.1, where unlabeled vertices correspond to combinations not analyzed in the next section. In particular, we did not test the combination of copy count with the self-adaptive mechanism, because the latter starts working when a high number of duplicates is generated, i.e. when many nodes are already infected. At this time, the copy count stops diffusions; hence, the combined effects of the two mechanisms cannot be seen. The impact of the LS mechanism is shown only with HSA-BCAST; its effects are comparable for the other history-based algorithms. The characteristics of the basic mechanisms are summarized in Table 1. Both the self-adaptive mechanism and the history aim at controlling the number of duplicates, while history and copy-count aim at achieving a *stop condition*; SA-BCAST can stop when *MINP*=0, but in this case infection of late joining nodes is not guaranteed. The history requires additional bandwidth only in case of sharing. All mechanisms require some additional memory. Only the copy-count mechanism – when used according to (Cooper et al., 2004) – assumes that nodes move according to a known model and their number is known. In the next section, we compare the above algorithms to identify their contribution to achieve the control on forwarding.

Performance Evaluation

Simulation Environment

We implemented the described protocols in the framework of the GloMoSim 2.03 (UCLA, 2008) simulation environment. The simulation setting considers a system of 50 nodes sparsely distributed over a 1000 × 1000 m. area. Nodes move at a speed in [1, 2] m/s, thus reproducing a pedestrian environment. They are equipped with a low power 802.11 radio device with 10 m. communication range and DCF at the MAC layer. Beaconing is performed every 1 sec.; after 3 missing beacons, the corresponding neighbor is removed from the neighbor list.

For SA-BCAST, the simulations run different values of *Nth* and two different functions $\mathcal{F}$: a linearly decreasing function (or *Lin10*) and an inverse exponential function (or *InvExp*).

When an infected node *p* receives a duplicate from a node that is infected from less than 3 min., then *p* sets $Prob_p$ to *MAXP*=1. Otherwise, $Prob_p$ is either decremented of 0.1 or halved in line with functions *Lin10* or *InvExp*, respectively. $Prob_p$ has a lower bound defined by *MINP* = 0.01. For CC-BCAST, according to (Cooper et al., 2004), τ=10 is adopted.

We consider long lived broadcasts, with simulations lasting up to 6 hours. All simulation results are averaged over 50 simulations performed with variable random seed. The mobility models are provided by BonnMotion (de Waal, & Gerharz, 2008); in order to allow movements to reach a steady state, the first 1000 sec. of the traces are not considered for the measures. The parameters of the different mobility models are presented in the next subsections.

Main performance indexes are the *coverage*, i.e. the percentage of nodes infected, and the *duplicate messages* (a message is a duplicate if it is received by an already infected node).

Simulation Results

Random Waypoint model. In the measurements presented in this work, the pause time for RWP is 0. In fig.2, the basic P-BCAST is compared with the zero-knowledge approaches. SA-BCAST effectively reduces the number of duplicates with respect to P-BCAST (fig.2(b)); yet, when coverage is high and several duplicates are generated, the self-adaptive mechanism decreases the infection aggressiveness and, as a side effect, a higher latency is observed in reaching 100% coverage (fig.2(a)). CC-BCAST has the same latency as SA-BCAST with $\mathcal{F}$=*Lin10*, because nodes that exhaust their budget stop diffusions, thus reducing the number of relays. CC-BCAST stops diffusions 1.5 hours after full coverage is achieved, and from this point on no more packets are generated.

HP-BCAST (fig.3) provides a twofold advantage over P-BCAST: it does not affect coverage (the white circle (Rc 100%) evidences the point where full coverage is reached in the worst case), and it provides an effective mechanism to control the forwarding. The higher is the global state awareness (HP-BCAST$_{100}$ vs. HP-BCAST$_0$), the lower the number of duplicates. However, HP-BCAST is very slow in reaching the stop condition: our simulations show that full coverage is achieved in less than 1 hour of simulated time. At this point, each node on average knows the 15% of infected nodes with HP-BCAST$_0$ and 65% with HP-BCAST$_{100}$ (fig.8(b)). With HP-BCAST$_0$ all nodes stop within 47 hours (although only 2 packets/hour overall are sent after 24 hours); with HP-BCAST$_{100}$, only 1 packet/hour is sent after 24 hours and all nodes stop within 41 hours. In (Cooper, Ezhilchelvan, Mitrani, & Vollset, 2005), the CC-BCAST approach has been enhanced by adding a history either locally maintained by nodes or completely shared; let us indicate it with HC-BCAST$^{\tau}_{\alpha}$ with τ the copy count threshold and α the amount of history exchanged. When a node *p* encounters a node *q* already in its history, *p* suppresses the transmission but increments its copy count. HC-BCAST (fig.3) achieves full coverage, although with a slightly higher latency than HP-BCAST, due to nodes that exhaust their available copy count before all nodes have been infected, thus decreasing the number of relays and slowing down the diffusion. Yet, thanks to the bound on the number of diffusions, HC-BCAST is optimal in RWP and is thus much more effective than HP-BCAST in limiting useless traffic: full coverage is reached after 3672 sec., and message diffusion stops after 6696 sec. with HC-BCAST$^{10}_{100}$, and after 7560 sec. with HC-BCAST$^{10}_{0}$.

We studied how the algorithm behavior is affected by the broadcast nature of the radio channel where, to correctly deliver *m* to an uninfected node *q*, *m* may be duplicated in a node *p* that happens to be in range. A good forwarding control should identify the presence of uninfected nodes and refrain from forwarding otherwise. The *broadcast success rate bsr* properly captures this ability; it is defined as the ratio *hitting broadcasts/total broadcasts*, where a “hitting broadcast” is a broadcast that delivered *m* to at least one uninfected node. Of course, *bsr=1* indicates that all the broadcasts hit the mark. The index drops to 0 when full coverage is reached. The behavior of *bsr* (fig.4(a)) confirms the remark above. The *bsr* index before full coverage is the same for HP-BCAST and HC-BCAST, confirming the effectiveness of the history in suppressing useless transmissions. However, the node’s knowledge does not grow as quickly as the nodes infection and this influences the efficiency of the forwarding control. With HP-BCAST nodes continue performing transmissions till their histories are full; with HC-BCAST no useless transmission is anymore generated after all nodes halt. With HC-BCAST$^{10}_{0}$, though, the last infected nodes could issue sporadic transmissions before halting.

Lesson learnt: SA-BCAST is more effective than HP-BCAST in reducing duplicates. HC-BCAST is more effective than HP-BCAST in both implementing a stop condition and decreasing duplicates.

Aggregation model. With AGG, we performed experiments with a number of *ap*s variable from 3 to 10, uniformly distributed in the area. A node stops in an *ap* for 10 minutes. The next *ap* is chosen according to a uniform probability distribution. The distance of a node from the *ap* center is determined by σ_{AGG} in [0,15]; for σ_{AGG} =10 it follows the probability distribution shown in fig.5. When switching to the AGG model, SA-BCAST fruitfully uses the contact opportunities in *ap*s to speed up the infection (fig.4(b)). The collateral effect is that SA-BCAST, although able of smoothing down the generated traffic as soon as a high coverage is reached, is too aggressive in diffusing when nodes are in an *ap*, thus generating a high number of duplicates (fig.6(a)). Some improvements can be achieved with a more stringent $\mathcal{F}$, as shown. With *Nth*>100 some contact opportunities may be missed, thus further reducing traffic; coverage is anyway achieved, although with a higher latency, thanks to the existence of multiple relays.

The number of *ap*s also has impact: when it tends to ∞, the RWP and AGG models converge. By contrast, with 3 *ap*s we observed more duplicates independently of $\mathcal{F}$ and *Nth*, because *(i)* nodes are more dense in *ap*s, and *(ii)* there is a higher probability of encounters during movements between two *ap*s, which are used for dissemination. For larger *ap*s (higher σ_{AGG}), nodes in an *ap* are not all in mutual communication range. When a node enters an *ap*, the messages exchanged do not affect all nodes and, at the same time, reduce $Prob_p$ thus preventing excessive diffusions; as a result, we observed a lower number of duplicates. An interesting aspect is shown in fig.6(b), where the network coverage is shown for a single simulation; in order to emphasize the behavior, with $\mathcal{F}$ =*InvExp* the $Prob_p$ is decreased by dividing the current value by 20. With 10 *ap*s, where encounters during movements are more sporadic, coverage increases in steps, which correspond to relevant membership changes in the *ap*s. Steps become less high with the progress of the simulation, because the probability of entering an *ap* with already infected nodes increases. This behavior is much more evident for $Prob_p$ decreasing more quickly. The broadcast nature of the channel, although nodes are unaware of being either in an *ap* or on the road, allows to effectively exploit node density to increase the coverage. However, this is achieved at the expenses of efficiency.

The weakness of the copy-count approach as used by (Cooper et al., 2004) is its dependence on the uniform distribution of contacts as in RWP. We simulated HC-BCAST in AGG with 10 *ap*s and σ_{AGG}=10. In this mobility scenario, HC-BCAST suffers multiple contacts between the same pair of nodes (fig.7(a)); this does not lead to duplicate generation but forces the nodes to waste their broadcast budget every time they re-encounter a node seen in the past. As a consequence, the algorithm is too conservative and full coverage is not achieved: in the conditions shown in the figure, all nodes terminate the algorithm when, on average, the 2% of them is still uninfected; the minimum coverage observed is 94%. By exasperating the non-uniformity of contacts till considering a swarm mobility, the coverage drops to 17% in the worst case. By contrast, under the same conditions the HP-BCAST algorithm guarantees full coverage independently of the mobility model, although this result is paid with a higher number of duplicates. The above arguments lead to say that – when nodes are unaware of the node cardinality and the mobility model – the copy count mechanism as used according to (Cooper et al., 2004) fails.

We then measured the performance achieved with HSA-BCAST: a node can schedule a diffusion only when has one or more neighbors not in its history; the diffusion is performed or suppressed according to $Prob_p$. Merging history and self-adaptive mechanism favors the control of the number of generated duplicates (fig.7(b)), because useless transmissions are suppressed both towards already known nodes and when duplicates indicate that coverage in the region is already high. The latency obtained by HSA-BCAST is intermediate between HP-BCAST and SA-BCAST, and higher for higher *Nth*. Yet, the number of duplicates diverges for all algorithms, indicating that HSA-BCAST too is still far from reaching a stop condition within the 6 hours of simulated time.

In order to compare the efficiency of the approaches, we consider as performance index the *target ratio* $T = (msgrecv - dups)/msgrecv$, with *msgrecv* the total number of messages received, and *dups* the total number of duplicates among them. T is a measure of efficiency in using the network resources. Of course, T is optimized by $dups=0$ and is affected by the number of the encounter nodes and by the progress of the infection in the neighborhood. In fact, packets are broadcast to the nodes in range, let us say k; so that, for any message sent, we count $msgrecv=k$ and the *dups* value depends on the level of infection among the k neighbors. The measure in fig.8(a) confirms that HSA-BCAST in the AGG model behaves even better than SA-BCAST in the RWP model.

Since history proves to be an efficient mechanism, a trade-off could be characterized between the amount of history exchanged and the bandwidth saved by suppressing duplicates. The comparison among different percentages of history exchanged and different policies to extract the history entries to be exchanged (Gamberini, Giudici, Pagani, & Rossi, 2008) yields that – by indicating with 100% the full history condition and exchanging the most recent entries – the knowledge growth by sharing 20% of the history (HP-$BCAST_{20}$) well approximates the behavior achievable with full sharing (fig.8(b)); this effect is obviously reproduced by the number of generated duplicates. However, in the following measures the nodes exchange the whole history.

Lesson learnt: the copy count mechanism fails in case of non-uniform mobility model. History and self-adaptive mechanisms are efficient in suppressing duplicates. However, the stop condition is reached too late.

Swarm model. The SWR model used in the experiments has pause time of 10 minutes, σ_{SWR} =1.73, μ_{SWR} of 4 or 15, d_{SWR} =15 m., and M_{SWR} =0.2. With this model, coverage can be incremented when two swarms partially overlap, one of which has already been infected. On the other hand, once a swarm has been infected, the nodes belonging to it should refrain from transmitting again till the swarm membership does not change. Better performance is achieved with *Nth* low, which promptly detects swarm overlapping. In fig.9(a), the performance of SA-BCAST is reported for $\mathcal{F}$ =*Lin10*; with $\mathcal{F}$ =*InvExp* the coverage achieved is worse. Yet, in the latter case a lower number of duplicates is generated (fig.9(b)). Hence, the *InvExp* function has been adopted for experiments with HSA-BCAST and LSA-BCAST. The local suppression mechanism does not produce benefits (nor drawbacks) in the aggregation model. However, in the SWR model it is able to improve both coverage and – above all – efficiency. This derives from the small delay before diffusing: if two swarms A and B are overlapping, such that nodes in A own *m* while nodes in B do not, an infected node in swarm A is likely to observe a sequence of new neighbors appearing at a short interval one after another. One “late” transmission allows to infect more new neighbors at one time. At the same time, the history mechanism allows infection propagation in swarm B: the newly infected nodes in the intersection have empty histories. They see all their neighbors in swarm B as not being in their histories, thus starting message diffusion. This repeats recursively till the whole swarm B is infected.

Lesson learnt: with the LS mechanism, HSA-BCAST seems to achieve the best performance in all basic mobility models. However, it is unable to stop promptly.

Conclusions

In this work, three mechanisms for message broadcasting in DTNs have been analyzed, together with their combinations, in different mobility models. The most promising approach seems to be the adoption of both a self-adaptive policy and a history of past encounters, optimized with a local duplicate suppression mechanism. This combination is independent of the mobility model, but is slow in reaching a stop condition thus being possibly inefficient. On the other hand, with a copy count approach nodes risk of being too prompt in halting diffusions, thus not reaching a satisfying coverage nor infecting late joining nodes. Moreover, estimating an appropriate copy count threshold could be a hard task.

Future research work can proceed along different directions. We intend to analyze real mobility traces in order to develop a synthetic model that captures real human behaviors. This model will be useful for both more accurate performance evaluation of communication protocols for DTNs via simulations, and analysis aiming at optimizing the protocols. The LSA-BCAST protocol must be improved by including an adaptive stop condition. A promising approach could be exploiting the *bsr* index: if nodes could (quite accurately) estimate *bsr* using local observations, they could stop when *bsr* approximates 0, which should be the time at which full coverage is achieved. Finally, the analytical model proposed in (Cooper et al., 2004) could be extended in order to compute a value for τ more appropriate for realistic mobility, to then re-evaluate the performances obtained.

Acknowledgments. This work has been partially funded by the Italian Ministry of University and Research in the framework of the "Context-Aware RouTing Over Opportunistic Networks (CARTOON)" PRIN Project.

References

Boldrini, C., Conti, M., Iacopini, I., & Passarella, A. (2007). HiBOp: a history based routing protocol for opportunistic networks. *Proc. IEEE WoWMoM*.

Burgess, J., Gallagher, B., Jensen, D., & Levine, B.N. (2006). MaxProp: routing for vehicle-based disruption tolerant networks. *Proc. IEEE INFOCOM*.

Burns, B., Brock, O., & Levine, B.N. (2005). MV routing and capacity building in disruption tolerant networks. *Proc. IEEE Infocom*.

Camp, T., Boleng, J., & Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless Communication and Mobile Computing, 2*(5), 483-502.

Chaintreau, A., Mtibaa, A., Massoulie, L., & Diot, C. (2007). The diameter of opportunistic mobile networks. *Proc. ACM CoNEXT Conference*, (pp. 1-12).

Cooper, D.E., Ezhilchelvan, P., & Mitrani, I. (2004). High coverage broadcasting for mobile ad-hoc networks. In *Lecture Notes in Computer Science, 3042*, (pp. 100-111).

Cooper, D.E., Ezhilchelvan, P., Mitrani, I., & Vollset, E. (2005). Optimization of encounter gossip propagation in mobile ad-hoc networks. *Proc. 13th MASCOTS*, (pp. 529-532).

CRAWDAD (2008). *CRAWDAD – A community resource for archiving wireless data at Dartmouth*. Retrieved from http://www.crawdad.org.

Davis, J.A., Fagg, A.H., & Levine, B.N. (2001). Wearable computers as packet transport mechanisms in highly partitioned ad-hoc networks. *Proc. 5th IEEE Intl. Symp. on Wearable Computers*.

de Waal, C., & Gerharz, M. (2008). *BonnMotion – A mobility scenario generation and analysis tool*. Retrieved from http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/.

DTN Research Group. (2008). *Delay Tolerant Networking Research Group*. Retrieved from http://www.dtnrg.org/wiki.

Dubois-Ferriere, H., Grossglauser, M., & Vetterli, M. (2003). Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. *Proc. 4th ACM Intl. Symp. MobiHoc*, (pp. 257-266).

Gamberini, G., Giudici, F., Pagani, E., & Rossi, G.P. (2008). Impact of history on epidemic broadcast in DTNs. *Proc. 1st IFIP Wireless Days Conference*.

Giudici, F., Pagani, E., & Rossi, G.P. (2009). Self-adaptive and stateless broadcast in delay and disruption-tolerant networks. *Proc. Italian Networking Workshop*.

Harras, K.A., Almeroth, K.C., & Belding-Royer, E.M. (2005). Delay tolerant mobile networks (DMTNs): controlled flooding in sparse mobile networks. *Proc. IFIP Networking*.

Hui, P., Chaintreau, A., Scott, J., Gass, R., Crowcroft, J., & Diot, C. (2005). Pocket switched networks and human mobility in conference environments. *Proc. ACM SIGCOMM Workshop on Delay Tolerant Networking*.

Jones, E.P.C., Li, L., & Ward, P.A.S. (2005). Practical routing in delay tolerant networks. *Proc. ACM SIGCOMM Workshop on Delay Tolerant Networking*, (pp. 237-243).

Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L., & Rubenstein, D. (2002). Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with Zebranet. *Proc. ASPLOS*.

Lee, U., Magistretti, E., Zhou, B., Gerla, M., Bellavista, P., & Corradi, A. (2006). MobEyes: smart mobs for urban monitoring with a vehicular sensor network. *IEEE Wireless Communications*, *13*(5).

Lenders, V., Karlsson, G., & May, M. (2007). Wireless ad hoc podcasting. *Proc. 4th IEEE Conf. SECON*, (pp. 273-283).

Montresor, A., Jelasity, M., & Babaoglu, O. (2005). Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems, 23*(3), 219-252.

Pedersini, F., Grossi, G., Gaito, S., & Rossi, G.P. (2008). Experimental validation of a 2-level social mobility model in opportunistic networks. *Proc. 1st IFIP Wireless Days Conference.*

Spyropoulos, T., Psounis, K., & Raghavendra, C.S. (2004). Single-copy routing in intermittently connected mobile networks. *Proc. 1st IEEE SECON*, (pp. 235-244).

Su, J., Chin, A., Popinova, A., Goely, A., & de Lara, E. (2004). User mobility for opportunistic ad-hoc networking. *Proc. 6th IEEE Workshop on Mobile Computing Systems and Applications*.

UCLA Parallel Computing Laboratory (2008). *GloMoSim – Global Mobile Information Systems Simulation Library*. Retrieved from http://pcl.cs.ucla.edu/projects/glomosim.

Vahdat, A., & Becker, D. (2000). Epidemic routing for partially connected ad-hoc networks. Technical Report CS-200006, Duke University.

Table 1

Characteristics of the Broadcast Protocols

	memory	bwth	stop condition	duplicates	mobility model	node cardinality
P-BCAST	–	no	no	Yes	no	no
CC-BCAST	counter	no	Yes	Yes	RWP	known
SA-BCAST	neighbors	no	if MINP=0	decreased	no	no
HP-BCAST	encounters	if sharing	Yes	decreased	no	no

Figure Caption

Figure 1. Family of broadcast protocols.

Figure 2. (a) Coverage and (b) cumulative number of duplicates vs. time for P-BCAST, CC-BCAST and SA-BCAST in the RWP model.

Figure 3. (a) Coverage and (b) cumulative number of duplicates vs. time for HP-BCAST and HC-BCAST in the RWP model.

Figure 4. (a) Broadcast success rate for HP-BCAST and HC-BCAST in the RWP model. (b) Coverage and number of generated packets vs. time for SA-BCAST in the RWP and AGG model with 10 *ap*s and σ_{AGG}=10.

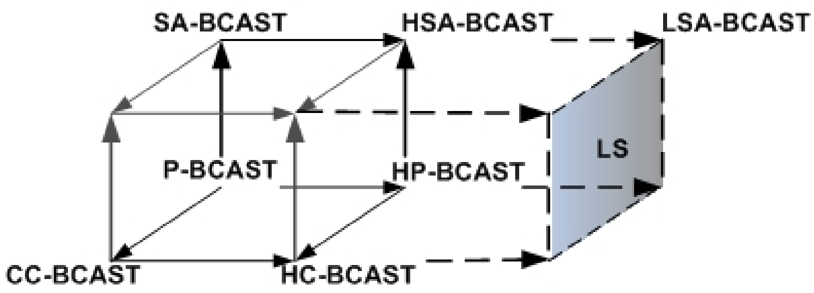
Figure 5. Rayleigh distribution for σ_{AGG}=10.

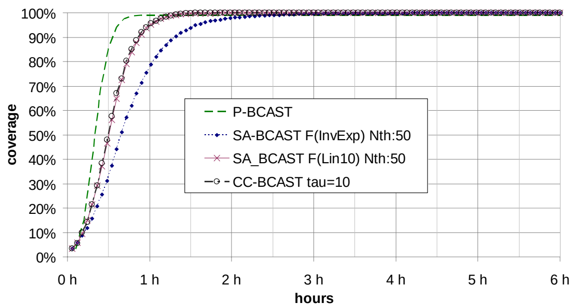
Figure 6. (a) Cumulative number of duplicates after 6 hours of simulated time for SA-BCAST in the AGG model. (b) Progress of coverage vs. time in a single simulation, with σ_{AGG}=10.

Figure 7. (a) Coverage for HP-BCAST and HC-BCAST, and (b) cumulative number of duplicates for HP-BCAST and HSA-BCAST, in the AGG model with 10 *ap*s and σ_{AGG}=10.

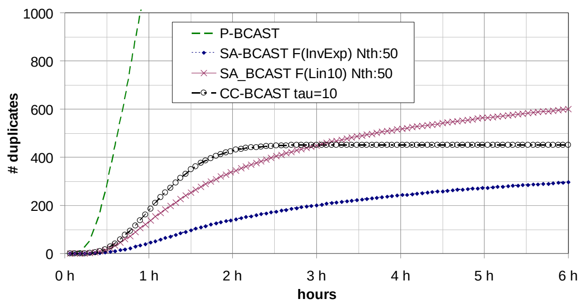
Figure 8. (a) Target ratio vs. coverage with and without history in RWP and AGG model with 10 *ap*s and σ_{AGG}=10. (b) Cumulative knowledge about the coverage status.

Figure 9. (a) Coverage and (b) cumulative number of duplicates vs. time for different algorithms in the SWR model.

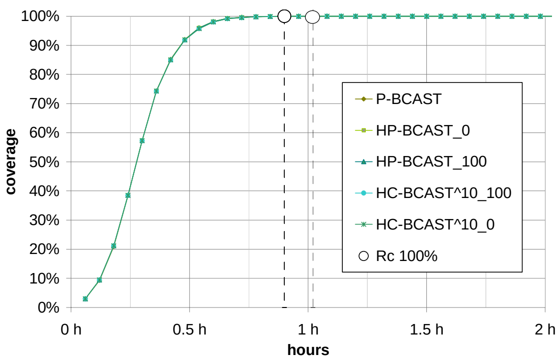

SA-BCAST
HSA-BCAST
LSA-BCAST
P-BCAST
HP-BCAST
LS
CC-BCAST
HC-BCAST


coverage
hours
P-BCAST
SA-BCAST F(InvExp) Nth:50
SA_BCAST F(Lin10) Nth:50
CC-BCAST tau=10

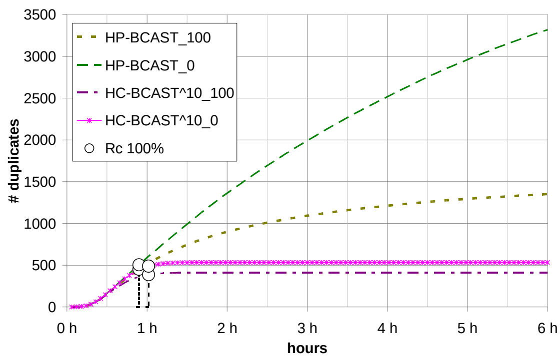
2(a)


duplicates
hours
P-BCAST
SA-BCAST F(InvExp) Nth:50
SA_BCAST F(Lin10) Nth:50
CC-BCAST tau=10

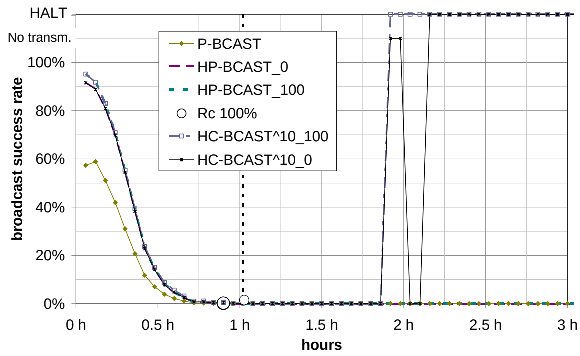
2(b)


coverage
hours
P-BCAST
HP-BCAST_0
HP-BCAST_100
HC-BCAST^10_100
HC-BCAST^10_0
Rc 100%

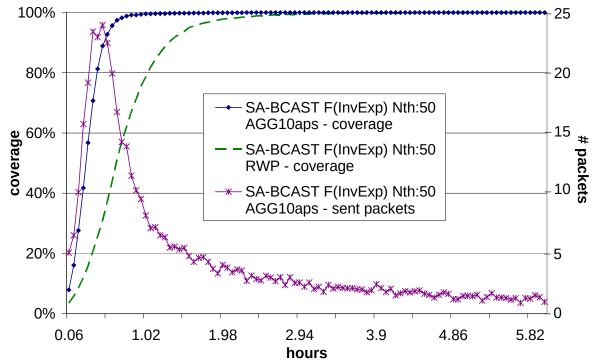
3(a)


duplicates
hours
HP-BCAST_100
HP-BCAST_0
HC-BCAST^10_100
HC-BCAST^10_0
Rc 100%

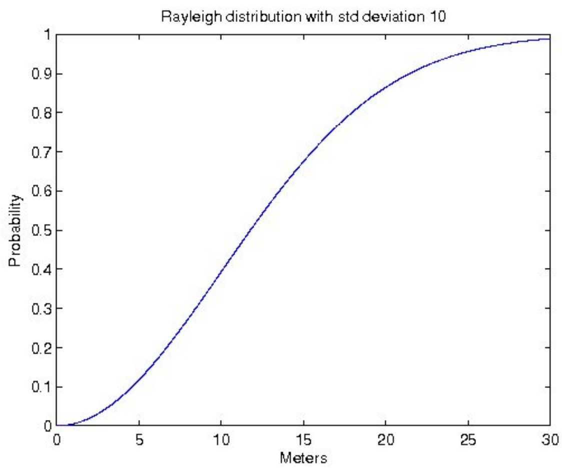
3(b)

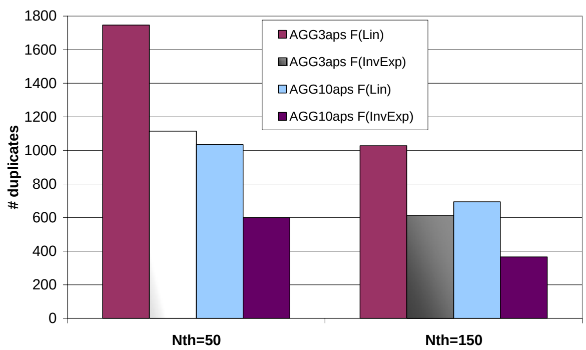

HALT
No transm.
100%
80%
60%
40%
20%
0%
broadcast success rate
0 h
0.5 h
1 h
1.5 h
2 h
2.5 h
3 h
hours
P-BCAST
HP-BCAST_0
HP-BCAST_100
Rc 100%
HC-BCAST^10_100
HC-BCAST^10_0

4(a)

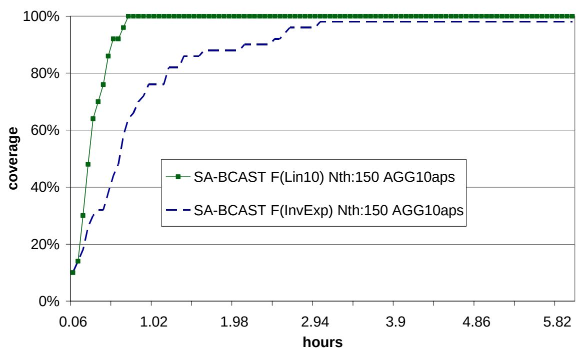

100%
80%
60%
40%
20%
0%
coverage
25
20
15
10
5
0
packets
0.06
1.02
1.98
2.94
3.9
4.86
5.82
hours
SA-BCAST F(InvExp) Nth:50 AGG10aps - coverage
SA-BCAST F(InvExp) Nth:50 RWP - coverage
SA-BCAST F(InvExp) Nth:50 AGG10aps - sent packets

4(b)

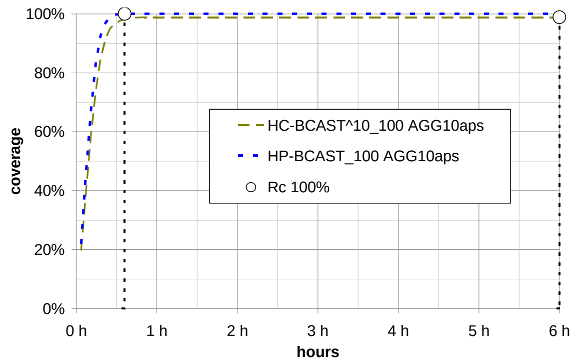

Rayleigh distribution with std deviation 10
Probability
Meters
0
5
10
15
20
25
30
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


1800
1600
1400
1200
1000
800
600
400
200
0
# duplicates
Nth=50
Nth=150
AGG3aps F(Lin)
AGG3aps F(InvExp)
AGG10aps F(Lin)
AGG10aps F(InvExp)

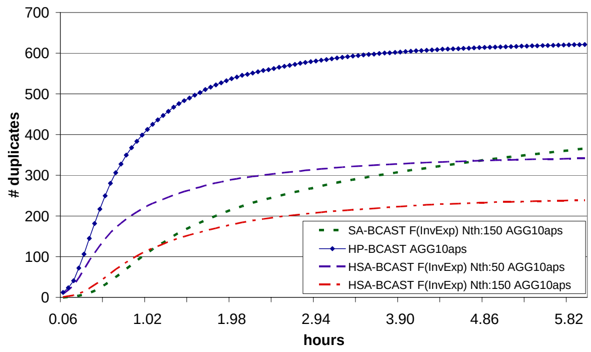
6(a)


100%
80%
60%
40%
20%
0%
coverage
0.06
1.02
1.98
2.94
3.9
4.86
5.82
hours
SA-BCAST F(Lin10) Nth:150 AGG10aps
SA-BCAST F(InvExp) Nth:150 AGG10aps

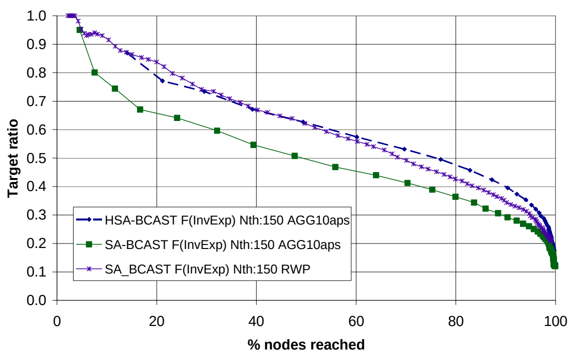
6(b)



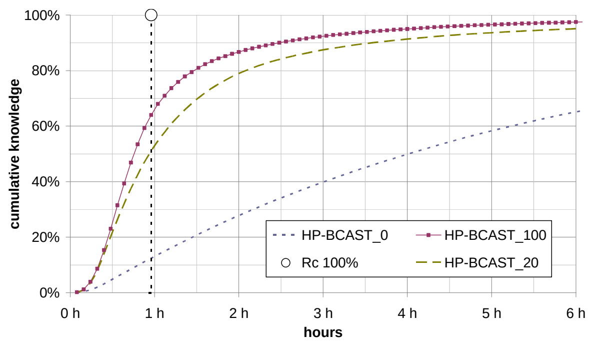
7(a)



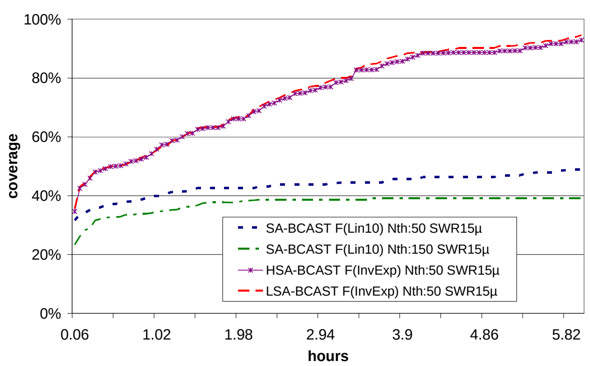
7(b)



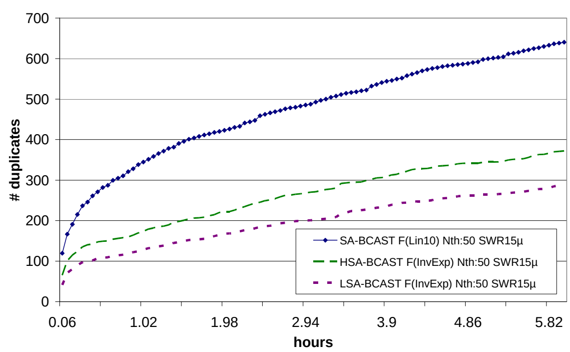
8(a)



8(b)



9(a)



9(b)