BLOOGO
BLOOm filter based GOssip algorithm for wireless NDN

ABSTRACT
Modern ad-hoc networks focus on data rather than obsolete end-to-end communications. This paper describes BlooGo, a gossip algorithm that disseminates messages throughout the network with a minimum number of transmissions. Its peculiarity lies on the fact that without any knowledge of the network the receiver of a packet can decide autonomously whether to forward it or not. All the information that BlooGo uses to decide the utility of a transmission is enclosed in the packet as a bloom filter; this way the parties, i.e. sender and receiver, never have to communicate directly. This approach makes BlooGo stateless; lightweight; power-efficient; ideal for sensors or embedded devices that use the NDN philosophy.

Categories and Subject Descriptors
H.1.1 [Models and Principles]: Systems and Information Theory—General systems theory

General Terms
Gossip Algorithms, Network Modeling, Ad-hoc Networks

Keywords
Ad-hoc Networks, Content Centric Networking, Distributed Algorithms, Gossip Algorithm, Modelling, Named Data Networking, Network modelling

1. INTRODUCTION
Undoubtedly during the last twenty years IP has served the greater cause of a world-wide network but, as a "killer application", it has also forced the emerging technologies to follow its lead. So did mobile ad-hoc networks (MANET) pursuing machine to machine communication in a scenario where connectivity is typically poor and intermittent. However, now that Named Data Networking (NDN) has finally broken the taboo of dismissing IP, ad-hoc wireless networks can finally take shape. Firstly, NDN oversteps the issues of TCP with wireless and mobility. Secondly, NDN relaxes the problematics of routing packets from a source to a destination, i.e. a common client-server interaction will be dismissed [?, ?]. Future data-centric networks will retrieve content from any of its replicas using multiple independent paths. This work stems from the recent efforts of substituting the obsolete routing protocols with gossip protocols [?] for locating data within the network. Gossip protocols are known to guarantee robust data dissemination at the cost of a more aggressive bandwidth usage. In a wireless network the medium is a precious resource thus a proper heuristic should decide whether a packet should be forwarded or not. BlooGo discriminates the utility of a transmission taking advantage of the properties of the wireless medium and achieving quasi-optimal performance. The specification of NDN call Interest the packet used to request content. Instead, throughout the paper the more general terms of query or message will be intentionally used as BlooGo embraces the whole Content-Centric-Networking philosophy without a particular focus on CCNx (its most noticeable implementation). This choice is due to the fact that many projects already envisioned future improvements for data-networking such as: data queries, content advertising and preemptive caching; BlooGo can adapt to all of them. The purpose of BlooGo is to efficiently deliver messages from one host to an unknown set of hosts for which these are of interest. Through the whole paper, it will be assumed that responses can travel backwards on the same path of the original message. A bidirectional extension of BlooGo is currently being tested but it will not be discussed in this paper. The breakthrough of BlooGo consists in being completely decentralized and stateless—nodes only know their neighbors from the beacons they receive. The underlying idea of BlooGo consists in piggy-backing the neighborhood of a sender into each message. A relay node forwards the message only if its neighborhood is not strictly included into the one of the sender. The packet is then relayed substituting the neighborhood of the previous sender with the one of the relay. Efficiency is guaranteed by compressing this information in a bloom filter.

2. RELATED WORK

2.1 NDN and Ad-Hoc Networks
A first step toward data oriented MANETs was taken by Delay Tolerant Networks (DTN) [?]. At the same way of NDN, they substitute routing schemes with opportunistic
forwarding. Unfortunately, DTNs lacked of formalization because (1) data were not named thus could not be cached efficiently and (2) the network was still dependent on IP whereas NDN can work aside of it. Robust packet delivery was achieved by sacrificing performance. A concrete project for mobile content centric networking arrived from Haggle proposed by Crowcroft et al in [?]. For the sake of precision, Haggle still gives names to machines but the intent of a content centric network is clear. For instance Haggle allows to request content given its meta-data. To the best of our knowledge the only concrete proposal for a complete NDN MANET is from Meisel et al [?]. This work is based on a forwarding scheme named LFBL [?]. With LFBL a node \( N \) floods a request for data to the network. Any node \( R \) that has the content sends back an offer packet that is free to acknowledge or refuse with a third packet. If \( N \) acknowledges the data exchange can start. With both BlooGo and LFBL the receiver decides autonomously whether to forward or not. Differently than BlooGo, LFBL uses proximity to the content as heuristic to take a decision. BlooGo, instead, infers the utility of another transmission by (1) controlling if this would deliver the message to a new node and (2) checking how far from the sender this message would be forwarded.

### 2.2 Gossip Algorithms

Gossip algorithms, or epidemic protocols, were invented as a robust way to deliver data in a disruptive network. These algorithms work exactly as gossip in social relationships. Packets are forwarded from node to node until they reach their destination. Since replicating each packet on every single node can cause a cumbersome overhead packets are usually forwarded by a node with a given probability chosen in relation to the network density. This probabilistic approach is easy to implement and relatively efficient, although it is myopic compared with BlooGo since it does not discriminate about the utility of an additional transmission. Other gossip algorithms use a stateful approach, i.e. node share information about the network and coordinate their transmissions - some examples are ExOr [?], Spray and Wait [?], MaxProp [?]. These protocols maintain the legacy of end-to-end communication whereas BlooGo does not.

A valuable insight to the performance of gossip algorithms have been given by Boyd et al in [?], [?].

![Figure 1](image)

Figure 1: When \( A \) requests a content its interest should not be forwarded if any between \( B,C \) and \( D \) has it. If that is not the case then only \( B \) and \( D \) should forward \( A \)'s request since they are the only ones that can reach a part of the network that \( A \) cannot reach.

A node \( h_i \) receives a packet from a node \( h_j \), \( p \) is forwarded only if:

\[
h_i \not\subset T_m \quad \text{The node is not one of the targets}
\]

\[
NBF_i \not\subset NBF_j \quad \text{There is at least one host that did not receive it with the previous transmission. An example is represented in figure ??.
}

Ultimately, a host controls if the packet has been received before. In which case this is dropped because it must be a duplicate; otherwise the packet is forwarded.

A real world example can help understand the algorithm, let's assume that we are standing in the middle of huge Hall. Someone from the back wall starts shouting the names of the people who can hear what he is saying then he shouts a message. Assuming that each person in the hall knows the name of everybody else around him, each one is able to verify if these people heard what the previous person shouted. If there is at least one that did not, he will do the exact same thing, shouting first the names of the people who can hear him and then then shouting the original message. If every one in the room does this, it is easy to prove that everybody in the room will receive the message. BlooGo does exactly the same thing with the only difference that (1) the name of the nodes (people who can hear the person who shouts) are known by listening to their beacons and (2) they travel compressed into a bloom filter of 16 Bytes. The reader is owed of an explanation for the following assumptions (1) The wireless channel is assumed symmetric (2) Bloom Filters are prone to false positives (3) In case of mobility the NBF can contain stale information (4) Multiple nodes can decide to forward the same message to the same part of the network.

### 3. OVERVIEW

The shared nature of the wireless medium can help avoiding redundant transmissions. Since data "on-air" are received by any host in range multiple transmissions are useless if nobody else requested the data. At the same way, when a node sends a request message (named Interest by the NDN specification) this should be forwarded only if the node in the transmission range do not have the content. This scenario is shown in figure ??.. BlooGo uses this same conservative approach to deliver a message \( m \) from a sender \( S \) to a set of target nodes \( T_m \). A target node \( t_i \in T_m \) is any node for which \( m \) is of interest. The message travels in a packet \( p = \{ Id_m, NBF_S, m \} \); where \( Id_m \) is a random identifier and \( NBF_S \) contains the hosts in the transmission range of the sender. Each time \( p \) is relayed by a peer \( h_j \) \( NBF_S \) is changed with \( NBF_j \). Without loss of generality lets assume that all the nodes in the network can act as relays. When
4. USE SCENARIOS

4.1 Sensor Networks

Current research [?, ?] looks at sensor networks as distributed databases. BlooGo can be used as career for the queries sent from the base station to the network; or, vice versa, it can carry alarms autonomously sent by the sensors to the base stations. Since the algorithm uses basic arithmetic operations on bloom filters it is easy to hardwire; beside being lightweight, BlooGo is also fast to deploy as the nodes do not need a unique identifier or any knowledge about the network—whereas IP based nodes need unique identifiers and some knowledge about the network topology.

4.2 Content Retrieval and Content Indexing

Locating content is one of the problems of NDN over ad-hoc networks. BlooGo can transport NDN’s Interest packets to where the content has replicas. As shown in the result section, BlooGo would deliver the Interests along approximated shortest paths therefore content can also efficiently travel backwards on them. However, content retrieval is not the only feature that a content-based network should have. Future NDNs could also integrate search engines and content indexing as well. For this purpose, it would be useful to be able to query the characteristics, or the description, of a content without downloading all of it. Once created, indexes could be distributed and cached as any other content in the network.

4.3 Generic messages for many application

In a thorough NDN the content of the message m can be anything, such as: (1) “I want content Ci” (2) “What is the author of the song in Ci?” (3) “I want the environment temperature at coordinates X Y Z”. Respectively, the answer would be: (1) the content Ci, (2) the name of the author Ci, or (3) the value read from a sensor nearby X Y Z. It is important to point out how this does not violate the named data pattern. Case 1 is self explanatory i.e. the sender asks content C. Case 2 and 3 are less obvious because the name of the content is the query itself, although the reader should notice how the content remains a singleton and can be downloaded and cached at the same way of any other information in the network. With sensors this is true only under the assumption that their sensed values are not noisy; which is unlikely but also outside the scope of this paper.

5. BLOOGO DESCRIPTION

Adopting BlooGo an application can efficiently send messages from one host to many. This is valuable whenever the message does not have a known destination, or when it is not known who is able to answer to the message. The focus of this paper is on how efficiently messages can be delivered rather than how response can be sent back to the sender. As previously stated, for this purpose there is the breadcrumbs technique. The nodes keep track of the messages they have forwarded by storing their identifiers; an answer to the message Id will travel inside a packet p that has the same identifier Id, and the flag Answer set to 1. This approach has been shown by Van Jacobson et al to be loop-free in [?] with the only difference that in CCNs the response can only contain data.

5.1 Beaconing and Nodes id

Nodes pick a random NodeID before joining the network. Differently than with IP this does not have to be unique in the network. Periodically, each host hj sends a beacon containing its identifier NodeID and its NBF. When a node hj receives the beacon it reacts as follows: (1) it controls if its NodeID is included into the NBF. This way is possible to understand if the channel is symmetric or not. (2) if this is the case the node increases a counter for the beacons received from hj. Otherwise hj just drops the beacon. (3) if the counter reached a certain threshold α the NodeID is added to the current NBF. α must be dimensioned with respect to mobility and the expected packet loss of the network. Periodically each node rebuilds its NBF adding only the NodeIDs that are counted more than α times because some of them might not be in range of transmission anymore. All the counters are also reset to 0.

5.2 Bloom Filters

The NBF has to be rebuilt from scratch because bloom filters do not allow to remove elements. A bloom filter consists of (1) a bit array B of m bits all initialized to zero and (2) a set of k hash functions hj : N → {1...m}, j ∈ {1...k}. Given a set of values L = {x1, x2, ... , xn} each xi with 1 ≤ i ≤ n feeds the k hash functions so that each xi maps into k positions of the array B, respectively hj(xi). All these positions are then set to 1 in the array B. In order to query if an element y was part of the original set this must feed the hash functions in order to obtain k positions in the array B. If all these positions are set to 1 we accept y as part of the original set with a known probability. A peculiarity of bloom filters is that they allow false positives but they do not allow false negatives. If an element is not included in a bloom filter then it was surely not part of the original set.

5.3 Message Forwarding

Each time a host receives a packet it only has to compare two bloom filters. This operation consists in verifying if the local NBFi is completely included into the one inside the incoming packet NBFi. This can be done by testing if (NBFi AND NBFi) XOR NBFi > 0. Where the AND operator finds which bits the two bloom filters have in common and the final XOR checks if there is at least one position with different values. This solution, is an approximation prone to error since bloom filters allow false positive a node can mistakenly decide to not transmit, the quality of this approximation will be further discussed. It remains to say how the number of relayed messages can be reduced, i.e. how to avoid that two or more peers relay a packet to the same part of the network. This problem is reduced to the minimum by delaying the transmission proportionally to the number of bits shared by NBFi and NBFj, this because there is an high probability that a relay node far from sender can cover a bigger area of the network, as can be seen in ??.
6. ANALYSIS

The analysis is conducted assuming a lossless network with nodes uniformly distributed in a square of side $l = 5$ and density $\delta$. Nodes have a circular transmission range of radius $r = 1$ and area $\Phi$. The average number of nodes in an area of dimension $\Delta$ is equal to $\bar{X}(\Delta) = \Delta \delta$; accordingly the probability mass function of the $X$ number of nodes within the area is:

$$f_X(x, \Delta, \delta) \sim \text{Poisson}(x, \Delta \delta)$$

where $x$ is the number of peers and $\Delta \delta$ is the rate of the Poisson distribution. This scenario ensures connectivity with high probability when $r = \Theta(\sqrt{\log(n)}/n)$, as result of [7].

Given two nodes $h_A$ and $h_B$ distant $d$ from each other, the dimension of the overlapping area of their range of transmission is written $S(d)$; instead the portion of the range of transmission that the two nodes do not share is written $I(d)$.

Table ?? shows how this value can be calculated given the parameters and figure ?? shows the graphic.

### Error with Bloom filters

The utility function of BloomGo is prone to error because of the probabilistic nature of bloom filters, i.e. a node can mistakenly decide to not forward a packet when it should. This can be proven by the following claims:

**CLAIM 1.** A forward decision can never be wrong.

**PROOF 1.** The claim is proven recalling that bloom filters do not allow false negatives and this is the only condition that can make this happen.

**CLAIM 2.** Given two nodes $h_A$ and $h_B$. If $h_B$ receives a packet from $h_A$ the packet will not be forwarded if the NodeIDs of ALL the nodes that are exclusively in range of $h_B$ map into positions of $NBF_A$ that are already set to 1. 

**PROOF 2.** Considering the expression:

$$(\text{NBF}_A \text{ AND } \text{NBF}_B) \text{ XOR } \text{NBF}_B > 0$$

The expression between the parenthesis is equal to the bits that $\text{NBF}_A$ shares with $\text{NBF}_B$. Thus XORing the result with $\text{NBF}_B$ if this contains at least a bit that is not set to 1 in $\text{NBF}_A$ the total must be greater than 0.

In order to derive the quality of the approximation lets consider a node $h_B$ that receives a packet from $h_A$. On average $\text{NBF}_A$ contains $\bar{X}(\Phi)$ NodeIDs. Recalling that a bloom filter with $n$ elements has a false positive probability equal to: $f_p(k, n, m) = (e^{-kn})^k$, then $f_p(k, \bar{X}(\Phi), m)$ is the average probability of having one node in range of $h_B$ that maps into positions of $NBF_A$ that are all set to 1, where $k$ is the number of hash functions and $m$ is the dimension of the bloom filter in bits. Although, the packet will not be forwarded only if ALL the nodes that are in range of $h_B$ map into bits of $NBF_A$ that are set to 1, which leads to:

$$e_{NBF} = \left( e^{-\frac{k \bar{X}(\Phi)}{m}} \right)^{\bar{X}(I(d))}$$

Note that, while $\bar{X}(R)$ is function of $\delta$ and $\bar{X}(I(d))$ depends on both delta and $d$, $e_{NBF}$ depends only on the distance $d$ between $h_B$ from $h_A$. The statement can be proven by simplifying the exponent of $e_{NBF}$ given the parameters $\delta$ and $d$, the algebra is omitted because trivial. This property does not hold whenever the network has not a uniform distribution of the nodes, future work will comprise the analysis of more artificial topologies.
Thus the probability that a node \( h \) is the furthest candidate relay is equal to \( f \) of \( h \) the furthest candidate relay. In fact, as shown in figure ?? any node at distance \( h \) node \( h \) the radius of the transmission range. Any node outside the range of transmission of the sender \( h \) node \( h \) that is at distance \( r + d \) from the sender, where \( r \) is the radius of the transmission range. Any node \( h \) that is at distance \( r < d \), \( r + d \) overlaps with the transmission range of \( h \) with an area equal to \( S(d) \). The probability of this area being empty can be bounded by Bonferroni’s inequality (aka Union Bound) to:

\[
 f_B(d, \delta) = f_X(0, S(d_i), \delta) \leq \int_{r+d}^{r+d} f_X(0, S(k), \delta)
\]

Thus the probability that a node \( h_j \) is the furthest from a sender \( h_S \) given its distance \( d \), is equal to

\[
 f_s(d, \delta) \cdot f_B(d, \delta)
\]

Note that the two probabilities can be multiplied together because given the network model they are independent. The results in figure ?? show how the probability of being a relay, and the probability of being the furthest relay for a given area, increases accordingly with the distance from the sender.

### 6.2 Pseudo-Geographic forward

Throughout the paper, and in the previous section, we claimed that BlooGo benefits from implicitly taking into consideration the distance between the sender and the relay. To find a close form for the probability that a node \( h_i \) is the perfect relay for a message that arrives from \( h_j \) is overly complicated, but it is possible to bound the quality of the assumption. A candidate relay for a message is any node \( h_j \) that can forward the message to a node \( h_i \) outside the range of transmission of the sender \( h \) S. Given the distance \( d \) between \( h_j \) and \( h \) \( h \) and the density \( \delta \), the probability of being a candidate relay is equal to \( f_s(d, \delta) = 1 - f_X(0, I(d), \delta) \). The best relays \( h_j \) are the ones that cover the biggest area outside the range of transmission of the sender \( h \) S. This probability cannot be easily found. Instead, it is possible to bound the probability that a node at distance \( d \) from \( h \) S is the furthest candidate relay. In fact, as shown in figure ??, any node at distance \( d \) from the sender can reach at most a node \( h_1 \) that is at distance \( r + d \) from the sender, where \( r \) is the radius of the transmission range. Any node \( h_1 \) that is at distance \( r < d \), \( r + d \) overlaps with the transmission range of \( h \) with an area equal to \( S(d) \). The probability of this area being empty can be bounded by Bonferroni’s inequality (aka Union Bound) to:

\[
 f_B(d, \delta) = f_X(0, S(d_i), \delta) \leq \int_{r+d}^{r+d} f_X(0, S(k), \delta)
\]

Thus the probability that a node \( h_j \) is the furthest from a sender \( h \) S given its distance \( d \), is equal to

\[
 f_s(d, \delta) \cdot f_B(d, \delta)
\]

### 6.3 Simulation

The last results consist in comparing BlooGo’s performance against probabilistic forwarding. The experiments are conducted simulating random topologies faithful with the network model previously discussed. For each set of parameters the simulator generates 100 topologies then for each pair of nodes \( (h_1, h_2) \) a message is routed from \( h_1 \) to the network until this reaches its destination \( h_2 \). The results in figure ?? show how BlooGo outperforms probabilistic forwarding reaching the target nodes in a quasi-optimal number of transmissions. Probabilistic forwarding performs poorly, especially when the density of the network is low, because it does not have a real heuristic to forward the packets. BlooGo utility function, instead, tries to cover the network in the fastest way possible.

### 7. CONCLUSIONS AND FUTURE WORK

BlooGo allows to efficiently send messages from one to many in a named-data ad-hoc network. BlooGo needs a
negligible amount of memory; it uses only basic arithmetic
operations; and it is power-efficient because it minimizes the
number of transmissions; this makes it ideal for sensors or
embedded devices in general. At last, as shown in the
results, BlooGo uses a pseudo geo-routing scheme by consid-
ering the proximity of sender and relay. This functionality is
usually aided by GPS device \cite{1} whereas BlooGo uses only
the wireless medium. There is a working implementation
of BlooGo that is being used to test its performance with
mobility and packet loss. Preliminary results are promis-
ing even though they are not presented in this paper due to
space constraints. A thorough presentation of BlooGo that
includes mobility and experiments in a real environment will
be presented soon. In the mean time, the source code of the
implementation is available upon request to whoever wants
to use it for research purposes.

8. ACKNOWLEDGEMENTS
Many thanks to Inchara Shivalingaiah and Ruolin Fan.

9. REFERENCES
multi-hop routing for wireless networks. In in
[2] Stephen Boyd, Arpita Ghosh, Student Member, Balaji
Prabhakar, and Devavrat Shah. Randomized gossip
[3] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and
Devavrat Shah. Randomized gossip algorithms.
[4] John Burgess, Brian Gallagher, David Jensen, and
Brian Neil Levine. Maxprop: Routing for
[5] Kevin Fall. A delay-tolerant network architecture for
conference on Applications, technologies, architectures,
and protocols for computer communications,
SIGCOMM ’03, pages 27–34, New York, NY, USA,
2003. ACM.
[6] Ramesh Govindan, Joseph M. Hellerstein, Wei Hong,
Samuel Madden, Michael Franklin, and Scott Shenker.
The sensor network as a database, 2002.
[7] Van Jacobson, Diana K. Smetters, Nicholas H. Briggs,
Michael F. Plass, Paul Stewart, James D. Thornton,
and Rebecca L. Braynard. Vocnc: voice-over
workshop on Re-architecting the internet, ReArch ’09,
pages 1–6, New York, NY, USA, 2009. ACM.
[8] Van Jacobson, Diana K. Smetters, James D.
Thornton, Michael F. Plass, Nicholas H. Briggs,
and Rebecca L. Braynard. Networking named content. In
Proceedings of the 5th international conference on
Emerging networking experiments and technologies,
ACM.
gecasting protocols for mobile ad hoc networks. Mob.
deterministic approach to throughput scaling in
wireless networks. IEEE Trans. on Information
[11] Samuel R. Madden, Michael J. Franklin, Joseph M.
Hellerstein, and Wei Hong. Tinydb: An acquisitional
query processing system for sensor networks. ACM
Listen First, Broadcast Later: Topology-Agnostic
Forwarding under High Dynamics. In Technical
Report.
Hoc Networking via Named Data. In Proceedings of
the Fifth ACM Workshop on Mobility in the Evolving
Internet Architecture (MobiArch), September 2010.
[14] M. Mitzenmacher. Digital fountains: a survey and
[15] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and
Cauligi S. Raghavendra. Spray and wait: An efficient

Figure 6: (a) shows the average number of transmission used to reach a node from another node. (b) shows the average
maximum number of transmission used to reach a node from another. From the plots it is clear how BlooGo outperforms
probabilistic routing, especially when the density of the network is low BlooGo is quasi-optimal.

Figure 6: (a) shows the average number of transmission used to reach a node from another node. (b) shows the average
maximum number of transmission used to reach a node from another. From the plots it is clear how BlooGo outperforms
probabilistic routing, especially when the density of the network is low BlooGo is quasi-optimal.
routing scheme for intermittently connected mobile networks, 2005.
