

Bisque: A Benchmark for In-Network Sensor Query Processing

Specification

Version 1.0

June 23, 2005

Qiong Luo Hejun Wu Wenwei Xue Bingsheng He
Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong, China
{luo, whjnn, wwxue, saven}@cs.ust.hk
<http://www.cs.ust.hk/bisque>

Table of Contents

- 1 Introduction 2**
- 2 System Under Test (SUT) 2**
- 3 Network Topology 3**
- 4 Database Schema and Population 3**
 - 4.1 Database Schema 3
 - 4.2 Database Population 4
- 5 Query Workload 6**
 - 5.1 Data Acquisition Query Workload 6
 - 5.2 Aggregation Query Workload 8
- 6 Performance Metrics 9**
- 7 Scaling Factors 9**
- 8 Other System Parameters 10**
 - 8.1 Transmission Range 10
 - 8.2 Noise Level 10

1 Introduction

Bisque is a benchmark for query processing systems in Wireless Sensor Networks (WSNs). The purpose of this benchmark is to reveal the performance characteristics of a complex in-network query processing system under realistic workloads while keeping the major performance factors of the system controllable. The workload is a set of continuous queries that are common in real-time environmental monitoring applications of current-generation WSNs. The database is virtual and dynamic in that it consists of data flowing out of each sensor node at a point in time. This specification defines the System Under Test (SUT), the network topology, the database schema and population, the query workload, the performance metrics, the scaling factors, and other system parameters.

2 System Under Test (SUT)

The SUT in Bisque is an *in-network sensor query processing system*. A layered illustration of a typical SUT is shown in Figure 1.

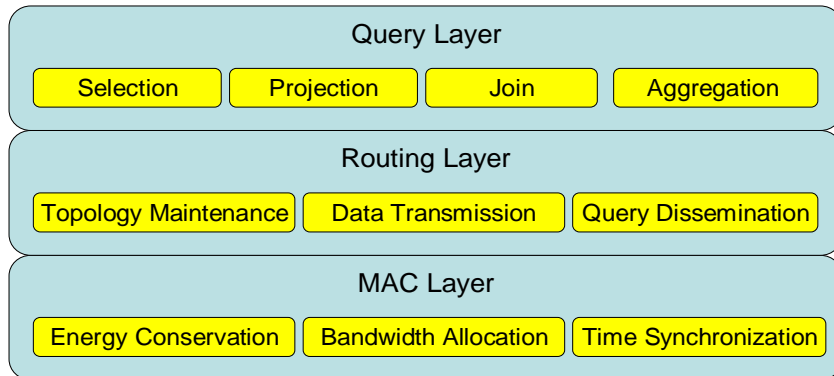


Figure 1. Three-layer illustration of an SUT in Bisque

The MAC (Medium Access Control) layer manages the wireless, multi-hop radio communication channel of a WSN. Its tasks include bandwidth allocation, time synchronization, and energy conservation. The routing layer maintains the network topology to disseminate queries from the sink node into the network and to transmit sensor readings in the network. The sink node is usually connected to a resource-abundant PC and has external power supply. Finally, the query layer executes declarative, database-style queries on each sensor node. In this layer, application-level filtering and aggregation schemes are applied to the sensor readings before the data are transmitted by the routing layer. This layer is also in charge of metadata management and power-aware query optimization.

This layered SUT architecture in Figure 1 is an abstraction of current in-network sensor query processing systems, but not a requirement of Bisque. A specific SUT may have its components mixed across layers.

3 Network Topology

As in common practice, the network topology in Bisque is defined with three factors: the area, the node positions, and the network links. The sensor nodes are deployed at the specified positions in a two-dimensional area of a specified shape and size. Given the wireless transmission range of each node, a network link exists between two nodes if their distance is within the transmission range of each node and no obstacles block the wireless signal between them.

To simplify performance analysis, similar to the previously adopted topologies, Bisque defines the area as a square of width w , in which $m*m$ nodes (including a sink) are deployed at the cross points of an $(m-1)*(m-1)$ grid with the cell width $a = w/(m-1)$. Without causing confusion, we call this grid an $m*m$ grid in the remainder of this document. The position of each node is denoted as coordinate (x,y) , with the node at the lower left corner being $(0,0)$ and the node at the upper right corner (w,w) . The sink resides at or near the center of the area: if m is odd, it is at $(w/2,w/2)$; otherwise, it is at $((m-2)a/2, (m-2)a/2)$.

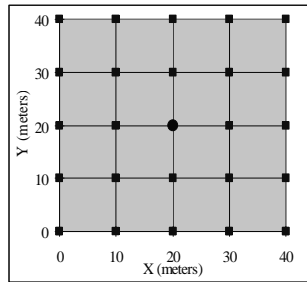


Figure 2 A 5*5 grid deployment

Figure 2 shows an example of a 5*5 grid deployment. The sink is shown as the dot in the center. If the transmission range of all nodes is 15 meters, the maximum number of hops from a node to the sink is 2 (e.g., from $(0,0)$ to the sink $(20,20)$).

4 Database Schema and Population

4.1 Database Schema

The Bisque sensor database schema (shown in Table 1) consists of a single, virtual relational table named *sensors*. At a point in time, the table consists of a number of tuples, each of which represents data from some sensor node in the network. An attribute in the table is either *sensory* that represents the real-time readings of a sensor attached to a node, or *non-sensory*, e.g., nodeid and location information.

Table 1. Schema of the sensors table

Attribute	Type	Description
nodeid	uint_16	Unique ID of the node, non-sensory
timestamp	datetime	The time when the tuple is produced, non-sensory
light	uint_16	Reading of the light sensor, sensory
temp	uint_16	Reading of the temperature sensor, sensory
humidity	uint_16	Reading of the humidity sensor, sensory
pressure	uint_16	Reading of the barometric pressure sensor, sensory
accel_x	uint_16	X-axis reading of the 2-axis accelerometer, sensory
accel_y	uint_16	Y-axis reading of the 2-axis accelerometer, sensory
mag_x	uint_16	X-axis reading of the 2-axis magnetometer, sensory
mag_y	uint_16	Y-axis reading of the 2-axis magnetometer, sensory
loc_x	uint_16	X-axis Cartesian coordinate of the node, non-sensory
loc_y	uint_16	Y-axis Cartesian coordinate of the node, non-sensory
voltage	uint_16	Remaining battery voltage of the node, non-sensory
freeram	uint_16	Amount of available RAM in the node, non-sensory
depth	uint_8	Minimum hop count of the node from the sink, non-sensory

4.2 Database Population

Sensory data collected from a real-world network deployment usually reflects the spatio-temporal trends and correlations of the physical phenomena being monitored in the environment. Therefore, we use real-world data sets as candidates for the Bisque database population. In order to fit an original real-world data set for different sample intervals in the queries, network topologies and test durations, Bisque provides a data generator to generate a specific database population out of the original data set that preserves the spatio-temporal characteristics of the original data set as well as satisfies the benchmark setup. The tool together with its documentation is available at the Bisque Web site.

A number of general requirements for a Bisque database population (denoted as D) are listed as follows:

- The schema of D is a subset of the Bisque database schema in Table 1. The two attributes *nodeid* and *timestamp* must be included in the schema of D .
- D preserves the spatio-temporal characteristics of a real-world data set.
- The nodes recorded in D have a one-to-one correspondence with the nodes in the benchmark network topology.
- The temporal granularity and duration of D is sufficient for the sample intervals in the queries and the test duration.

After analyzing several real-world data sets, we use the Intel Lab data set (<http://berkeley.intel-research.net/labdata>) to generate three synthetic data sets with different scales

(small, medium and large) as the default Bisque data populations. The three data sets contain sensory data for a 9-node (3*3), 25-node (5*5), and 49-node (7*7) grid network topology, respectively. Each reading contains four fields: *nodeid*, *timestamp*, *light* and *temp*. The first reading of each node in a data set has the same timestamp. In a data set, there are 1000 readings for each node (except for the sink) with a fixed interval of one second. Therefore, there are 8K, 24K and 48K sensor readings in the three data sets, respectively. Each data set is sufficient for testing queries with various multi-second sample intervals for a long test duration.

To give a rough idea about the spatial and temporal characteristics of the default data sets, Figure 3 shows the mean values of the light readings of each node in the 49-node data set, and Figure 4 the light readings of five representative nodes in the 49-node data set over time.

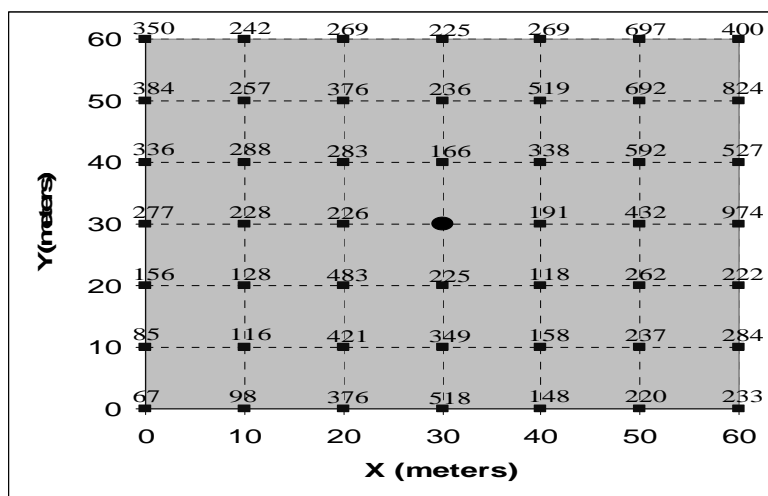


Figure 3. Mean light readings of nodes in the 49-node data set

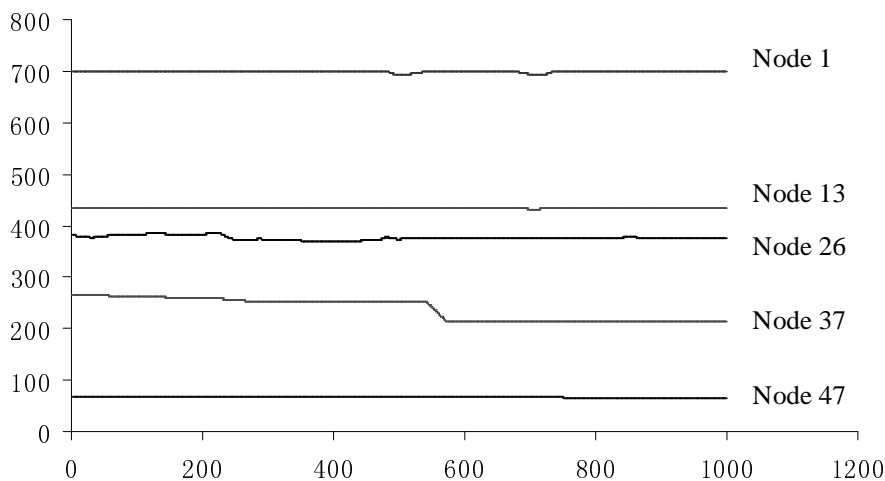


Figure 4. Light readings of five nodes in the 49-node data set over time

For the convenience of Bisque users, these three data sets are available at the Bisque web site for

public access. Nevertheless, Bisque users have the freedom to use other real-world data sets.

5 Query Workload

The Bisque query workload consists of both selection-projection queries and aggregation queries. We call them *data acquisition queries* and *aggregation queries*, respectively. As joins are uncommon in current systems, they are not included in the current version of Bisque.

The design considerations of the query workload include:

- Reveal and compare the performance characteristics of the major components in different SUTs, including the MAC protocols, the routing protocols, and the aggregation schemes.
- Choose queries that are most common and useful for real-world sensor network applications.
- Choose simple queries that can serve as basic building blocks of more complex queries.

Based on these considerations, we selected twelve queries to be included in the workload, with Q1-Q7 being data acquisition queries and Q8-Q12 aggregation queries. All queries in the workload are continuous queries with a fixed sample interval. The length of the sample interval is a scaling factor in Bisque and is discussed separately in Section 7. For the simplicity of presentation, the `SAMPLE INTERVAL` clause of each query is omitted in the following workload description.

5.1 Data Acquisition Query Workload

Q1: Single Sensory Attribute Projection

```
SELECT nodeid, light
FROM sensors
```

Q1 is the simplest query in the workload, which projects a single sensory attribute from all nodes in the network. Because which sensory attribute to project does not affect the relative performance of different SUTs, an arbitrary attribute *light* is projected in the query. The non-sensory attribute *nodeid* is also projected in order to identify what node a tuple is from and to enable the performance analysis on individual nodes for the query.

Q2: Projection of Multiple Sensory Attributes

```
SELECT nodeid, light, temp
FROM sensors
```

Q2 projects multiple sensory attributes for all nodes in the network. The purpose of this query is to investigate the effect of number of sensory attributes projected and transmitted. The number of attributes projected in the query can be increased as necessary.

Q3: Single Sensory Attribute Projection and Selection

```
SELECT nodeid, light
FROM sensors
WHERE light > C
```

Q3 studies the performance of selection queries on a sensory attribute. In comparison with Q1, this query adds a WHERE clause with a selection predicate on the projected *light* sensory attribute.

In each epoch (sample interval) of the query, only those nodes whose current *light* readings satisfy the predicate will send out their data towards the sink even though all nodes in the network acquire their own *light* readings. The set of nodes that satisfy the predicate may vary from epoch to epoch depending on the data distribution. The user-specified constant *C* in the predicate can be changed to achieve different selectivities of the predicate.

Q4: Conjunctive Selection with Multiple Sensory Attributes

```
SELECT  nodeid, light, temp
FROM    sensors
WHERE   light > C1
AND     temp > C2
```

The query condition of Q4 is the conjunction of multiple selection predicates on the sensory attributes projected. This query is used to investigate the predicate ordering issue in query evaluation. The number of predicates involved in the query condition can be increased as necessary. The selectivity of the predicates can be tuned by setting the values of the two constants *C*₁ and *C*₂.

Q5: Disjunctive Selection with Multiple Sensory Attributes

```
SELECT  nodeid, light, temp
FROM    sensors
WHERE   light > C1
OR      temp > C2
```

Q5 differs from Q4 in that its query condition is disjunctive. The techniques of handling disjunctive conditions, if present, are evaluated by this query.

Q6: Spatial Selection on Nodes near the Sink

```
SELECT  nodeid, light
FROM    sensors
WHERE   loc_x - $sinkX <= C
AND     loc_x - $sinkX >= -C
AND     loc_y - $sinkY <= C
AND     loc_y - $sinkY >= -C
```

This query involves range selections on the location attributes. It requires the nodes near the sink to send their sensory data to the sink node every epoch. For a stationary network, the set of nodes that satisfy the query is fixed for all epochs. Moreover, nodes that do not satisfy the predicates (i.e., outside the region) neither need to do sampling for the query, nor need to send data to the sink node.

The two system built-in variables *\$sinkX* and *\$sinkY* in the query condition represent the x and y axis coordinates of the sink node in the target network topology. The user-specified constant *C* is a positive real number and can be modified to change the area of the covered spatial region.

The query tests how well, if any, an SUT optimizes queries on geographical attributes. For instance, if each node in the network knows about the locations of its neighboring nodes, the query

can be optimized by not forwarding and installing the query to the nodes that reside far away from the sink node. Consequently, the sampling and communication cost can be saved for these nodes.

Q7: Spatial Selection on Nodes far away from the Sink

```
SELECT  nodeid, light
FROM    sensors
WHERE   loc_x - $sinkX > C
OR      loc_x - $sinkX < -C
OR      loc_y - $sinkY > C
OR      loc_y - $sinkY < -C
```

In contrast to Q6, Q7 requires those nodes that reside far away from the sink node to send their data to the sink node. However, due to the multi-hop nature of sensor networks, some nodes near the sink node still need to help forward data from the distant nodes to the sink node.

5.2 Aggregation Query Workload

Q8: Duplicate-Insensitive Simple Aggregation

```
SELECT  MAX(light)
FROM    sensors
```

Q8 tests the performance of the aggregation schemes for duplicate-insensitive aggregates. All nodes in the network participate in the aggregation process.

Q9: Duplicate-Sensitive Simple Aggregation

```
SELECT  SUM(light)
FROM    sensors
```

Q9 tests the performance of the aggregation schemes for duplicate-sensitive aggregates. The duplicate-sensitivity of the aggregate requires extra effort in multi-path routing in order to ensure the correctness of query results.

Q10: Aggregation with Sensory Attribute Selection

```
SELECT  AVG(light)
FROM    sensors
WHERE   light > C
```

In comparison with Q8 and Q9, Q10 adds a selection predicate on the aggregation sensory attribute. The predicate selects a subset of the nodes in the network to participate in the aggregation and this subset may change over epochs of the query depending on the data.

Q11: Aggregation with GroupBy Clause

```
SELECT  AVG(light), loc_x
FROM    sensors
GROUP BY loc_x
```

Q11 adds a GROUP BY clause to an aggregation query, which results in more communication cost in the network. The increased communication cost is because multiple partial aggregates, each of which is from one group, need to be transmitted in each epoch.

Q12: Aggregation with GroupBy and Having Clauses

```

SELECT    SUM(light), loc_x
FROM      sensors
GROUP BY  loc_x
HAVING    SUM(light) < C

```

Q12 further adds to Q11 a HAVING clause with a predicate on the aggregation sensory attribute. In each epoch of the query, a partial aggregate for a group may be dropped before arriving at the sink if it does not satisfy the HAVING clause.

6 Performance Metrics

We select the following three performance metrics for Bisque to evaluate the expected battery life, the query response time, and the result data quality.

(1) Power Consumption within an Interval (PT). It is the average accumulated power consumption of a node that processes the query within a time interval. This metric determines the battery life of sensor nodes.

(2) Response Time (RT). For a given query, the response time is the average time interval between two consecutive query results arriving at the sink. Note that, given a fixed time duration, the number of results generated by different SUTs may be different for the same query on the same data set and the same network topology. This difference is due to the differences in the routing and query processing techniques of the SUTs, and it affects the response time inversely.

(3) Relative Error Rate (RE). Assume at time t , the returned result is V_t and the correct sensory result is C_t . If there are M results produced between time t_1 and t_2 , we compute the average result error rate RE in Equation 1. The relative error rate measures the accuracy of the query results.

$$RE = \frac{1}{M} \sum_{t=t_1}^{t_2} abs\left(\frac{V_t - C_t}{C_t}\right) * 100 \% \dots\dots\dots Equation 1$$

7 Scaling Factors

We define three scaling factors in Bisque:

(1) Number of Nodes, or network size. It is the number of nodes in the WSN, including the sink. Three representative scales are 3*3, 5*5, and 7*7 nodes.

(2) Sample Interval of Continuous Queries. The sample interval is specified in a continuous query. Three representative scales are 10 seconds, 20 seconds, and 30 seconds.

(3) Test Duration. The test duration of each query is set based on the sample interval of the query. Bisque specifies the duration to be 10 times long as the sample interval, which is sufficient to get stable performance results.

8 Other System Parameters

8.1 Transmission Range

The transmission range is the maximum distance between a source and a destination node within which the radio signal of the source node can successfully reach the destination node. It determines the link between two nodes in a network. The default is 15 meters.

8.2 Noise Level

The noise level of an environment describes the noise strength in the wireless channel of a WSN. As it can be measured in an environment and can be classified to describe typical application environments, it is often used to estimate the Bit Error Rate (BER) of the wireless communication. The default is -13dB.