

Discrete Data Rate Adaptation for Wireless Body Area Networks

Tibor Szkaliczki

Institute for Computer Science and Control, 1111 Budapest, Hungary; szkaliczki.tibor@sztaki.hu;
Tel.: +36-1-279-6185

Abstract: eHealth services require continuous data streaming and a stable level of quality of service. However, wireless network connections can be characterized by variable bandwidths. This requires continuous adaptation of systems, including adapting the bit rates of data streamed by sensors. Assigning appropriate rates to the data represents a main task in congestion control. Most of the current methods look for proper sensor data rates within continuous domains. We examine the case when sensors can generate data streams with several different qualities (e.g., sampling rates, sampling accuracies, etc.). For this reason, the domain of the data rate values can be restricted to the discrete values representing the data rates of the possible quality variations. This paper examines the optimization of the utility of the delivered data under resource constraints by selecting an appropriate variation of the provided data from a discrete set. We provide a formal model for delivering data streams in WBANs and recommend an optimization algorithm to solve the problem. Our recommended solutions are related to the multiple-choice multidimensional knapsack problem. By comparing the proposed algorithms, we found that the greedy method closely approximates the optimum in a short running time.

Keywords: WBAN; quality of service; adaptation; multiple-choice multidimensional knapsack problem



Citation: Szkaliczki, T. Discrete Data Rate Adaptation for Wireless Body Area Networks. *Appl. Sci.* **2023**, *13*, 8529. <https://doi.org/10.3390/app13148529>

Academic Editor: Alessandro Lo Schiavo

Received: 10 June 2023

Revised: 18 July 2023

Accepted: 19 July 2023

Published: 24 July 2023



Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, the quick spread and evolution of wireless body area networks (WBANs) can be seen in the healthcare industry [1,2]. WBANs can monitor the health data of patients in real time at low cost. They can be considered as a special type of wireless sensor network (WSN) where the connected sensor devices (e.g., blood pressure sensor, temperature sensor, motion sensor, ECG, EEG, etc.) can be implanted under the skin, fixed on the body, or carried as portable devices. The sensor data are transferred to the WBAN coordinator (also called the sink) through wireless links. The sink transmits the collected sensor data from the WBAN for evaluation and further processing on a local terminal or a distant server.

Limited computation power, power supply, and communication bandwidth make the data transfer challenging in wireless sensor networks. The data should arrive with a short delay and high reliability in eHealth services to provide real-time observation and urgent response in case of emergencies. Monitoring human physiological activities requires continuous data streaming and a stable level of quality of service (QoS) [3,4]. However, wireless networks can be characterized by frequent packet loss and variable delays and capacities. For this reason, continuous adaptation of systems is required. In the case of a shortage of network resources, the quality of the data stream may be reduced. In this case, how to optimize the utility of the delivered data under the current resource constraints is a key question.

Figure 1 presents an example of WBAN. The various sensor data in WBANS have different QoS requirements (e.g., delay, reliability, energy efficiency, bandwidth utilization, and throughput). The data streams require different data rates depending on the type of measured data [5]. For example, data generated by medical sensors typically have a lower rate than audio and video streams. The importance of the data streams may also

vary, and they can be prioritized accordingly [6], e.g., heart rate has a higher priority than temperature.

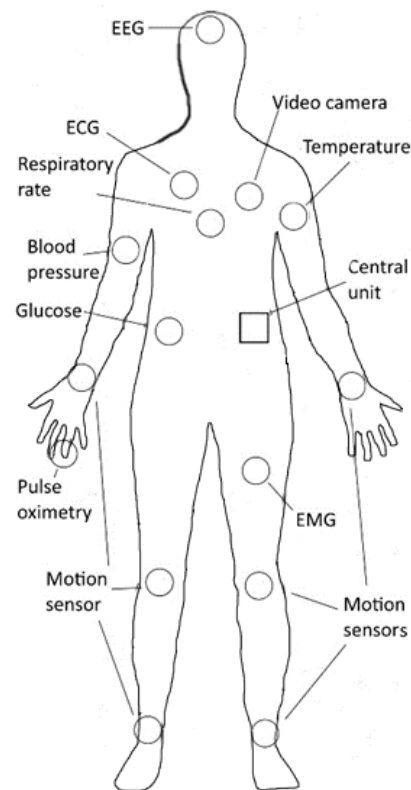


Figure 1. An example of a WBAN.

Our model exploits the fact that the sensors can be configured from software to provide a data stream in a few variations with different qualities. This is typical in the case of multimedia streams, e.g., a video stream can be generated with various spatial resolutions, frame rates, and quantization parameters. Commercial biosensors can be also programmed for different sampling frequencies or accuracy levels [7–9]. For example, an ECG (electrocardiogram) or EMG (electromyography) sensor node may produce data with a base rate and the rate of each subsequent variation is doubled (e.g., 125, 250, 500, 1000, . . . , 8000 SPSs). Furthermore, the number of bits in a sample can also be configured, e.g., an EEG (electroencephalography) sensor can be configured to sample signals with an accuracy of 24 or 32 bits. The programmable sensors enable the dynamic adaptation of the sampling rate in response to the changes in the system including the network, the sensor state, the monitored phenomenon, or the requirements.

Rate adaptation methods in the literature usually consider the data rate domain to be continuous. However, the sensors can generate data streams with a limited number of quality variants. Furthermore, the sensor transmitters can often send data with rates from a discrete set. For example, in IEEE 802.15.6-based WBANs, at most, eight different data rates can be defined in a frequency band [10]. For this reason, discrete rate adaptation models have high practical significance where the data rates are selected from a pre-determined, discrete set of rates coming from hardware and software constraints in WSNs [11,12].

We aim to optimize the delivered data under the current resource constraints by selecting an appropriate variation of the transferred data. The sensor data of the best quality should be transferred if there are enough resources in the network. However, if this is not possible because of the lack of network and node resources (e.g., throughput, energy, etc.) the data can still be useful in reduced quality for monitoring and evaluation purposes.

Data rate adaptation can be considered a complex multicriteria optimization problem where the aim is to maximize the overall utility of the delivered data streams. The utility

is defined as a metric expressing the overall satisfaction of the user consuming the data, who is typically a member of the medical staff in the case of a WBAN. The utility function defines the relationship between the usefulness of the delivered data and their parameters (e.g., data rate). The main reason for optimizing utility instead of throughput is that the same amount of data may have different contributions to the correct evaluation of the streamed information. The utility is typically a monotone concave function of the data rate, i.e., increasing the data rate results in a smaller or equal utility increase if the data rate is larger. For example, if the sampling rate of an ECG sensor is doubled, the utility of the stream increases by less than double.

Various utility models have been developed, especially in multimedia [13–15]. The utility functions are used for rate control and resource allocation in communication networks [16]. Network utility maximization (NUM) deals with allocating traffic rates to maximize the users' total utility subject to network resource constraints [17–19]. Utility models can be used to combine multiple optimization targets into one metric by applying weights. They are widely used in resource allocation in WSNs to ensure the effectiveness and fairness of the system [20]. The utility function can model the trade-off between QoS and energy efficiency [21]. Game theoretic approaches use the concept of utility to quantify the degree of a player's preference across a set of available alternatives [22].

The rate adaptation can be performed in a distributed or centralized manner. In the distributed case, each node locally determines a proper transmission rate to avoid network congestion. In the centralized case, the rate adjustment algorithm running on the sink node tries to optimize the data rate assignment using global network information and the sink node sends the calculated data rates to all sensor nodes. The main advantages of the centralized approach include that it can find near-optimal rate assignments and the complexity of software on sensor nodes except the sink node can be reduced.

The principal contributions of our work are defined as follows:

- A utility-based formal optimization model for discrete data rate adaptation in WBANs with a tree topology.
- Optimization algorithms to solve the discrete data rate adaptation based on the multiple-choice multidimensional knapsack problem.
- Evaluation of the proposed algorithms, where the greedy method closely approximated the optimum in a short running time and proved to be suitable for real-time usage.
- Further results of the evaluation show that solving the proposed model has much higher accuracy than adapting the solutions of more common models.

The remainder of this paper is organized as follows. In Section 2, related work is discussed. Section 3 introduces the proposed content adaptation module. Section 4 provides a formal model for delivering data streams in WBANs. In Section 4, a detailed description of the proposed algorithms is provided. In Section 5, the presented algorithms are evaluated and compared to each other, while Section 6 concludes the paper.

2. Related Work

This section provides an overview of rate adaptation methods applied in WBANs. Rate adaptation is widely used in resource allocation and congestion control in WBANs [23,24]. We selected some papers to compare with our proposed model and to show the special features of our model.

Ababneh et al. [25] presented an adaptive routing and bandwidth allocation protocol for traffic streaming in WBAN that selects routes based on the residual energy first and then provides adaptive bandwidth allocation for data streams. The sources can continuously adjust the rates of the provided data streams. The rate assignment optimizes the utility of the data streams under resource constraints on the amount of traffic that could pass through the links and nodes. The utility of a stream is considered proportional to its priority, the selected data rate, and the available energy level. Mixed integer linear programming is applied to solve the optimization problem, which can provide an accurate optimum but requires a long running time.

Baek et al. [26] proposed an adaptive rate control method for congestion avoidance in WBANs. The network model applies tree topology. The adaptive rate control mechanism calculates the risk degree of congestion occurrence in each node using the queue occupancy of the node, the number of upstream nodes on a path toward the sink, and the count of the child nodes. A valuation function is introduced to express the average effects of the difference in the transmission rates between the node and its parent. The optimal transmission rate is calculated by using the exponential weighted moving average formula, which depends on the risk degree at the node and the valuation function.

Ghanavati et al. [27] proposed an algorithm to detect and control congestion in a WBAN with a tree topology. The proposed congestion control scheme applies the Type-2 Fuzzy Logic Controller (T2-FLC) to estimate the severity of the congestion. If congestion is detected in a node, a rate adjustment unit (RAU) is called to allocate the sharing rate to each child node based on their priorities by using formulas, and the new transmission rate is calculated. The congestion notification spreads in the network hop by hop and the changes in transmission rates are sent to the child nodes. When a node receives a new rate assignment message, it adjusts its traffic rate accordingly.

Anwar et al. [28] proposed a protocol to avoid and mitigate congestion in WBANs. First, it identifies the congestion level of a node by using the current queue occupancy. Sensor data are transmitted in a way depending on the calculated congestion level. If the congestion level is low, the data are forwarded normally. If the congestion level is between two predefined thresholds, the protocol redirects data packets from congested nodes to their neighbors to avoid congestion. In the case of a high congestion level, the packet sending rate is reduced to control the congestion. The reduced packet sending rate of a node is computed by the available free queue size of the next upstream node in WBAN.

Manfredi [29] proposed a method to avoid congestion by distributing the available bandwidth among the individual data streams in a healthcare system. A rate control law is applied to calculate the amounts of bandwidth allocated to sensors depending on their priorities. The proposed formula maximizes the sum of a parametric concave utility function for all nodes. It can be tuned by a parameter to follow various fairness criteria, including the proportional fairness allocation in a special case, and it can converge to the max–min fairness as well. The proposed mechanism can deal with differentiated and dynamic healthcare scenarios by changing the priorities of the nodes and the parameter of the formula.

Liu et al. [30] formulated another QoS optimization problem for the allocation of the transmission power and time slots for each sensor in a WBAN using a common wireless channel to minimize energy consumption subject to the QoS constraints. A transmission rate adaption policy (TRAP) was proposed to adjust the transmission rate at each sensor when the link quality is poor to assist the QoS optimization problem. Rate adaptation can be tuned for a trade-off between energy consumption and the packet loss rate requirements.

Liu et al. [31] proposed a resource allocation scheme to optimize the allocation of the transmission power, the source rate, and the time slots to improve the QoS performances of a WBAN with a single wireless channel for all sensors. They formulated a throughput maximization problem whose optimal numerical solution was found efficiently by applying geometric programming. The source rates are allocated based on the optimal throughput results.

Goyal et al. [32] proposed the GABAT-TRAP algorithm to achieve QoS metrics in a WBAN with a star topology, which applies an optimized hybrid technique of the genetic algorithm (GA) with the nature-inspired metaheuristic optimization algorithm, called the bat algorithm (BAT). The transmission rates are initially optimized in TRAP, proposed in [30]. Then, the transmission rates of the sensors are further optimized using GABAT to optimize the energy efficiency of the network.

Kiran and Nithya [33] proposed the network status aware congestion control (NSACC) algorithm to predict the congestion level and regulate the packet sending rate. A fuzzy controller estimates the severity of the congestion. The rate regulation (RR) module applies

rules for each congestion level to change packet sending rates. It controls the congestion by adopting the logarithmic growth for the transmission rate to better utilize the bandwidth available, enhances the throughput, and reduces the number of retransmissions.

Kathuria and Sapna [34] proposed the following two approaches to improve QoS in WBANs: the reliable packet transmission with dynamic QoS (RPT-DQ) and the lion cooperative hunt optimization (LCHO) techniques for the optimization of multi-objective QoS. RPT-DQ consists of a series of steps from the priority assignment to the smart delay mitigation. The intelligent rate assignment step applies fast start and stabilized rate adjustment strategies. The fast start strategy initializes the packet transmission with a high data rate to utilize maximum bandwidth. The stabilized rate adjustment strategy increases or decreases the transmission rate of a sensor node by a stable amount when congestion occurs in the system to increase the packet transmission rate and throughput while reducing loss and delays.

Table 1 compares the problem models applied in the above papers with our proposed model. Our model is most similar to the model in [25]. Like in our proposed model, the multi-hop network and utility optimization are used in a centralized manner, but the optimum data rate is searched for in a continuous domain. The models in [26–28] also use multi-hop network topology but throughput is maximized instead of utility. Furthermore, they follow a distributed approach in rate adaptation. The method in [29] maximizes the utility similarly to our model, but it applies the simple star topology. Like in our proposed model [30–32], search for the data rate in the discrete domain. However, all nodes are directly connected to the sink in a star topology in these models. The methods in [23,33,34] adapt the data rate in the continuous domain in a WBAN with a star topology, which differs from our approach.

Table 1. Comparison of rate adaptation techniques.

Paper	Data Rate Domain	Optimization Goal	Constraints	Network Topology	Method for Rate Adaptation
Ababneh et al. [25]	continuous	utility	link and relay node capacity	multi-hop network	energy-balanced rate assignment and routing protocol, integer linear programming
Baek et al. [26]	continuous	throughput	buffer capacity	multi-hop network, tree	exponential weighted moving average using congestion risk degree and valuation function
Ghanavati et al. [27]	continuous	throughput	buffer capacity	multi-hop network, tree	fuzzy logic and formula for sharing data rate, hop-by-hop propagation of the congestion information
Anwar et al. [28]	continuous	throughput	buffer capacity	multi-hop network	formula using the queue occupancy of the upstream node
Manfredi [29]	continuous	utility	sink bandwidth	star	rate control law for bandwidth allocation
Liu et al. [30]	discrete	energy, packet loss rate	packet loss rate, throughput delay, packet loss rate, energy harvesting rate, transmission power	star	transmission rate adaption policy (TRAP), heuristics
Liu et al. [31]	discrete	throughput	packet loss rate, throughput delay, packet loss rate, energy harvesting rate, transmission power	star	joint power-rate control scheme, geometric programming
Goyal et al. [32]	discrete	energy	packet loss rate, throughput, delay	star	GABAT TRAP, TRAP followed by genetic algorithm (GA) with BAT algorithm
Kiran and Nithya [33]	continuous	throughput, packet transmission rate	buffer occupancy level, BSN priority, packet arrival rate	star	fuzzy logic and logarithmic growth for the rate
Kathuria and Sapna [34]	continuous	packet transmission rate	queue size	star	fast start and stabilized rate adjustment strategies
current paper	discrete	utility	bandwidth	multi-hop network, tree	knapsack problem-based algorithms

3. Optimization Problem Model

This section develops the optimization problem model for deciding on content adaptation. The network structure of WBANs can be described as a graph $G(V, E)$, where V denotes the nodes within the network and the edges E represent the wireless connection between two nodes. The number of nodes and edges are denoted by n and e , respectively ($n = |V|$, $e = |E|$). The network may consist of nodes of various types. Most of them are sensor nodes that generate and forward data streams. Let S denote the subset of the vertices consisting of sensor nodes. The number of sensor nodes is denoted by n_s ($n_s = |S|$). The sink nodes represent the target nodes of streaming, which may also belong to the sensor nodes. The network may include relay nodes as well, which only forward the data. Two nodes are connected by an edge if they are within each other's communication range. Each edge e_{ij} and node v_i may have a capacity C_{ij} and C_i , respectively, which represent the maximum amount of traffic that could pass through them. The capacity at the sink is denoted by C_s , which can be considered as an upper bound for all capacities in the network.

The data rate adaptation follows the routing phase. Let the set of the edges of the routing tree be denoted by E_r , which is a subset of E . The routing tree consists of edges used for data transmission; therefore, it is enough to check capacity constraints only on the edges in E_r . The routing tree consists of the same nodes as the network graph. Therefore, the routing tree can be denoted by $G_r(V, E_r)$. Let P_i and Q_{ij} denote the set of nodes whose routing path to the sink passes through node n_i and edge e_{ij} , respectively. These sets can be easily determined based on the routing tree G_r . Let R_i denote the unique route from node v_i to the sink in the routing tree, which can be determined in linear time in the length of the route from routing tree G_r . Let l_{tree} denote the maximum of the route lengths.

The total number of variants is denoted by n_v . Let D_i represent the set of the variations of specific data generated in the network node $v_i \in S$ with respect to the sensor capabilities (see Figure 2). l_i denotes the number of possible data variations at node v_i ($l_i = |D_i|$). Let max_v denote the maximum number of variants for the nodes, i.e., the maximum of l_i . Let $r(d_{ik})$ denote the required data rate of a data variation ($d_{ik} \in D_i$, $v_i \in S$, $1 \leq k \leq l_i$). Furthermore, we suppose that the utilities of the data variations are given. The utility is denoted by $u(d_{ik})$ for the data variation d_{ik} . The data variations with zero utility and the ones which do not provide a further increase in the utility can be omitted. We introduce d_{i0} for each $v_i \in S$ to denote the case when the data stream is stopped. This special variation neither requires resources nor has utility ($u(d_{i0}) = 0$, $r(d_{i0}) = 0$).

We aim to select a data variation (d_i) for each sensor to maximize the total utility ($\sum_{v_i \in S} u(d_i)$) under the constraints. The data variation is selected from the ones the sensor can generate. In addition, d_i can be 0 as well if streaming from the sensor should stall. The capacity constraints must be fulfilled both on the network edges and the nodes.

Input:

- The routing tree: $G_r(V, E_r)$;
- The set of possible data variations for each sensor node: D_i for $v_i \in S$;
- Utility for each data variation: $u(d_{ik})$ for $v_i \in S$, $1 \leq k \leq l_i$;
- Bit rate for each data variation: $r(d_{ik})$ for $v_i \in S$, $1 \leq k \leq l_i$;
- Bandwidth of each edge: C_{ij} for $e_{ij} \in E_r$;
- Limit on the bit rate for each node: C_i for $v_i \in V$.

Output:

- Selected data variation for each sensor: $d_i \in D_i \cup \{\emptyset\}$ for $v_i \in S$;

Maximize

- the total utility of the selected data variations:

$$\sum_{v_i \in S} u(d_i)$$

Subject to

- The capacity constraint for each network node $v_j \in V$:

$$\sum_{v_k \in P_j} r(d_k) \leq C_j$$

- The capacity constraint for each network edge used for routing $e_{ij} \in E_r$:

$$\sum_{v_k \in Q_{ij}} r(d_k) \leq C_{ij}$$

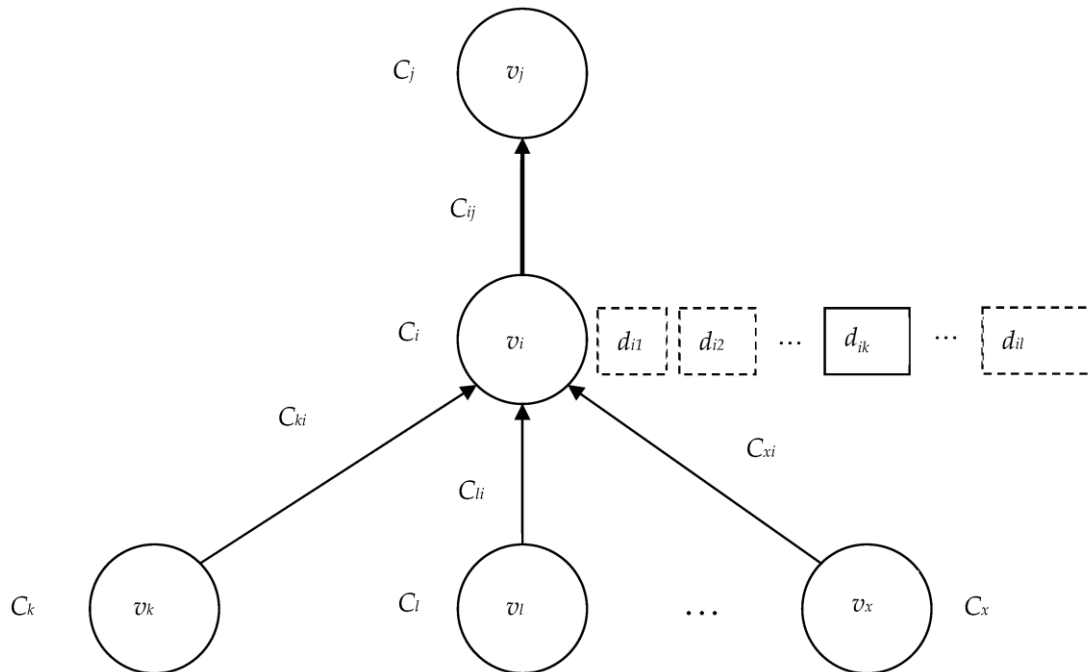


Figure 2. A sensor (v_i) with its neighbors in the proposed model. v_j represents its outgoing neighbor and $v_k, v_l \dots v_x$ represent its incoming neighbors in the routing tree. d_{ik} represents the data variations that can be generated by sensor v_i . Only one of the data variations is transferred. C_i and C_{ij} denote the capacities of nodes and edges, respectively.

4. Optimization Algorithms

4.1. Algorithmic Background

The above problem can be considered as a generalization of the knapsack problem (KP), namely, the multiple-choice multidimensional knapsack problem (MMKP). In our problem, variations of data streams are selected whose utilities and data rates correspond to the values and sizes of items, respectively. In the multiple-choice knapsack problem, there are several classes of items, and it is enough to choose only one item from each class. In our case, one variant can be selected for each of the sensor data. The problem is multidimensional as well, because instead of a single knapsack capacity, a series of capacity constraints along the routing path should be satisfied, which form a vector. In our case, the available capacity for a data stream is equal to the minimum capacity of resources along the routing path to the sink, which may be different for different streams. After selecting an item, the available capacity of node and edge resources should be updated along the route.

Several algorithms have been proposed in the literature [35–37] to solve the MMKP. They include heuristic methods and an exact branch-and-bound method that is useful for checking the validity of the solutions. Khan et al. [38] applied the MMKP to congestion control, but sensors do not provide data streams in different variants in their model. Multiple choice refers to the selection from several possible streaming routes. Khan et al. [39]

applied the MMKP for adaptive multimedia systems. This approach proved to be an efficient method to solve problems like quality adaptation, admission control, and integrated resource management. We have selected some solution methods for the knapsack problem to perform data rate adaptation in WBANs. The algorithms run in a centralized manner on the sink node.

Several algorithms are available to solve knapsack problems, e.g., based on greedy and dynamic programming approaches. The greedy solution of the knapsack problem offers a fast solution approach for data rate adaptation, although it usually fails to find the optimum. The greedy solution algorithm of the KP iteratively selects the item for which the ratio of the value and the size is the highest among the items whose size does not exceed the available space in the knapsack.

Dynamic programming, introduced by Bellman [40], is another widely used solution method for the knapsack problem. This method can be adapted to the multiple-choice knapsack problem (MCKP) as well (see, e.g., [41–43]). Let $U(i, k)$ denote the maximum value that can be attained with a weight less than or equal to k by selecting one item from each class S_1, S_2, \dots, S_i . The dynamic programming algorithm can be formulated by the following equations:

$$U(1, k) = \max_{j \in S_1: r_{1j} \leq k} u_{1j} \quad (1)$$

$$U(i, k) = \max_{j \in S_i: r_{ij} \leq k} (u_{ij} + U(i-1, k - r_{ij})) \quad (2)$$

For the processing the first class, the achievable value is the largest value of the item in the class with weight below the capacity limit (Equation (1)). In the general case described in Equation (2), the partial solutions belonging to S_1, S_2, \dots, S_i can be regarded as extensions of partial solutions belonging to S_1, S_2, \dots, S_{i-1} . Each $U(i, k)$ can be calculated by using the recursive formula. After completing the calculations, $U(n, C)$ contains the optimal value for the problem instance, where n is the number of classes and C is the capacity of the knapsack.

4.2. Greedy Algorithm

We adapted the greedy method to the rate assignment problem in the following way. The proposed method starts from the initial solution when no item is selected and incrementally improves the solution (see Algorithm 1 and Figure 3). It iteratively selects the node for which the ratio of the utility and the data rate is the highest among the nodes whose data rate is not larger than the available bandwidth on any of the links along the routing path, even after increase. The capacity constraints are determined in each iteration for each node. The data variations are sorted in decreasing order of the ratio of the utility and the data rate for each sensor.

The data rate adaptation method is called periodically, and the above method can be adapted accordingly. After the first run, the method can be started from the solution of the last run, which corresponds to the data rates currently streamed in the real system. The network bandwidth can be continuously monitored in WBAN, and the data rate adaptation algorithm can run periodically with updated network bandwidths.

At most, the while loop of the algorithm may run as many times as the number of variants: $O(n_v) = O(n_s \cdot \max_v)$. Sorting requires $O(n_s \cdot \max_v \cdot \log \max_v)$. The iteration through L_2 runs $O(n_s)$ times in each iteration of the while loop. The capacity calculation through the route takes $O(l_{tree})$ steps in each iteration. For this reason, the time complexity of the algorithm is $O(n_v \cdot n_s \cdot l_{tree} + n_s \cdot \max_v \cdot \log \max_v) = O(n_s \cdot \max_v \cdot (n_s \cdot l_{tree} + \log \max_v))$.

Algorithm 1: Greedy method

```

1 initialization:
2   Let the index of the current data variation for each sensor be zero:  $k_i = 0$ ,
3   and the current data variation for each sensor be empty:  $d_i = d_{i0}$  for  $v_i \in S$ 
4   Create a list  $L_2$  from all sensors  $v_i \in S$ 
5   for each sensor  $v_i \in S$ 
6     Sort the data variations in decreasing order of ratio  $u(d_{ik})/r(d_{ik})$ 
7 while  $L_2$  is not empty
8   Let  $max$  denote the highest ratio of utility and data rate found in the iteration so far.
   Let its initial value be 0:  $max = 0$ 
9   Let  $m$  denote the sensor node where the highest ratio was found. Let it be null
   initially:  $m = null$ 
10  Iterate through  $L_2$ 
11    Take the next item ( $v_i$ ) from  $L_2$ 
12    Calculate the minimum available capacity ( $C_i$ ) along the route  $R_i$  from node
    $v_i$  to the sink in the routing tree  $Gr(V, Er)$ 
13    Let  $k = k_i$ ,  $d_i = d_{ik}$ 
14    while  $r(d_i) > C_i$  and  $k \leq l_i$ 
15       $k = k + 1$ ,  $d_i = d_{ik}$ 
16    if  $k \leq l_i$ 
17      if  $u(d_i)/r(d_i) > max$ 
18         $max = u(d_i)/r(d_i)$ 
19         $m = v_i$ 
20      else
21        Remove  $v_i$  from  $L_2$ 
22  if  $m \neq null$ 
23    Move the current data variation  $d_m$  for sensor  $m$  to its next data variation:  $k_m$ :
    $= k_m + 1$ ,  $d_m = d_{m(k_m)}$ 
24    Decrease the available capacities along the route  $R_m$  with  $\Delta r(d_m)$ 

```

4.3. Dynamic Programming

We could not directly apply the dynamic programming solution for the multiple-choice knapsack problem to the sensor data rate adaptation. The problem of rate adaptation is more complex than the MCKP because it contains not a single capacity but multiple capacities. We applied the dynamic programming method to our problem in the following way. We defined an MCKP instance for each node of the routing tree, where the problem inputs include the possible variations of the data streams coming to the node. The possible variations of the data stream can be determined from the output of the problem instance at the source node of the incoming edge. For this reason, the optimal data rate adaptation can be found by solving a series of MCKP instances while traversing the routing tree.

The output of the MCKP is usually one solution with the highest value under the given capacity constraint ($U(n, C)$). We extend the output at the nodes to a list of the highest values for all capacity values in the array at class $S_n(U(n, k), 0 \leq k \leq C)$. At the leaves of the tree, the problem is simplified to the special case when there is only one class. This case can be solved by simply omitting the items with larger weights than the current capacity of the leaf. For the other nodes, the streams through the incoming edges should be optimized by considering the capacity of the node. This can also be considered an MCKP instance in the following way. Each incoming stream can be considered as a class, where the items are the optimal solution belonging to the different resource constraints. These items are stored in the output of the MCKP solved at the source node of the incoming edge of the stream. For this reason, the sensor data rate values can be found by recursively solving MCKP instances starting from the sink of the WBAN. After solving the knapsack problem for all vertices, the utility of the proposed solution can be found in $U_s(m_s, C_s)$ of the dynamic programming solution at the sink, where C_s denotes the available capacity at the sink and m_s is the number of incoming edges at the sink. Algorithm 2 provides the pseudocode for the dynamic programming on the routing tree. See Figure 4 for the flow chart of the algorithm.

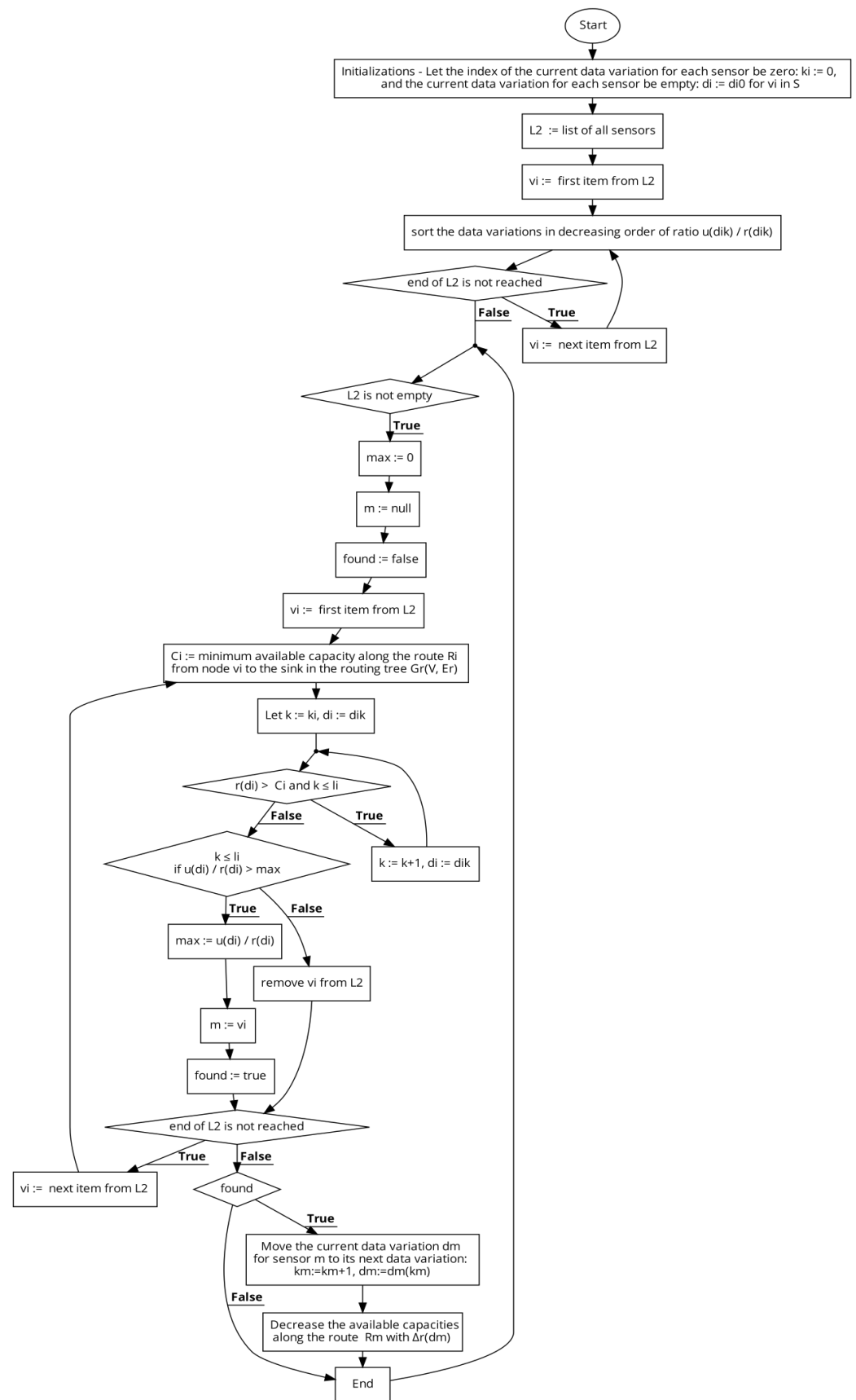


Figure 3. Flowchart of the greedy method.

The algorithm calls the $DP()$ function in Line 13, which is a dynamic programming method to solve the MCKP. See [43] for an example of the implementation of the algorithm. The input parameters of $DP()$ include the list of classes S_1, S_2, \dots, S_n , where each class contains items with weights and values and the size of the knapsack C . The algorithm adapted to our problem returns a list of value and weight pairs.

Algorithm 2: Dynamic programming on the routing tree (Traverse)

```

1 input: node  $n$ 
2 output:  $L$ : list of the possible data variations that can be transmitted by node  $n$  if  $n$  is not the sink
3      $max$ : the maximum utility if  $n$  is the sink
4 if  $n$  is a leaf of the routing tree  $G_r(V, E_r)$ 
5     Create an empty list  $L$ 
6     Calculate the minimum available capacity ( $C_i$ ) along the route  $R_i$  from node  $v_i$  to the sink in the
        routing tree  $G_r(V, E_r)$ 
7     for each variant  $d_{ni}$  in the set of possible data variations  $D_n$ 
8         if  $r(d_{ni}) \leq C_i$ 
9             add  $d_{ni}$  to  $L$ 
10    return  $L$ 
11 else
12    for each node  $j$  where  $e_{nj} \in E_r$ 
13         $L_j = \text{call } \text{Traverse}(j)$ 
14     $L = \text{call } \text{DP}(\{L_j: e_{nj} \in E_r\}, C_i)$ 
15    if  $n$  is not the sink
16        return  $L$ 
17    else
18         $max = 0$ 
19        for each variant  $d_j$  in list  $L$ 
20            if  $r(d_j) \leq C_i$ 
21                if  $u(d_j) > max$ 
22                     $max = u(d_j)$ 
23    return  $max$ 

```

The method runs a dynamic programming solution algorithm of the MCKP for each node of the routing tree. The time complexity of the algorithm for the MCKP is $O(n_i \cdot C_i \cdot max_v)$, where n_i and C_i denote the number of incoming edges and the capacity at node v_i . $O(n_i \cdot C_i)$ is the size of the dynamic programming table and an element is calculated in the $O(max_v)$ step in the worst case due to Equation (2). The time complexity of the algorithm for the whole network is $\sum_{v_i \in V} O(n_i \cdot C_i \cdot max_v) = O(n \cdot C_s \cdot max_v)$. Although the algorithm finds the exact optimum, its running time is because the value of the maximal capacity can be a large number. The value of the maximal capacity is exponential in its input length. The running time can be reduced if the capacity requirements of the sensor streams have a large common divisor. Otherwise, the capacities can be rounded to reduce the number of steps in the dynamic programming, but in this case, the solution usually differs from the exact optimum.

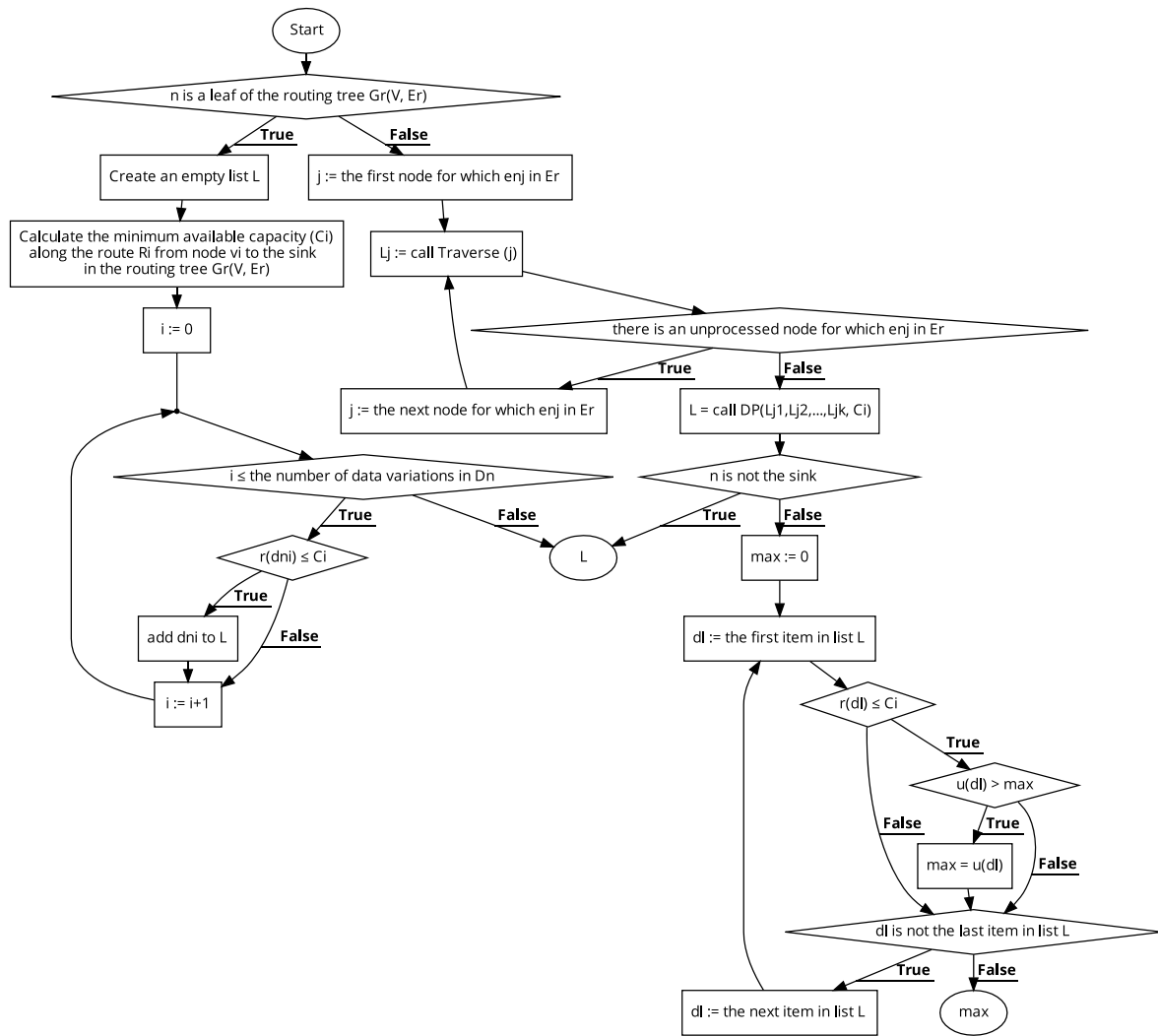


Figure 4. Flowchart of the dynamic-programming-based method.

5. Evaluation

This section presents an evaluation of the algorithms presented in the previous section. We implemented the proposed algorithms in C++ in Visual Studio Community Edition and tested them on simulated environments. We created an integer linear program (ILP) based on the formal model to find the exact optimum. ILP is a general method to solve hard combinatorial problems, and it is used for rate adaptation as well in the literature [25]. We solved it by using CPLEX. The implemented algorithms were compared with each other.

We also examined the accuracy of adapting solutions of common models to our case, namely the model with continuous data rate and the model maximizing throughput. The continuous data rate adaptation can be applied to the discrete case by rounding the results to the feasible discrete data rates. Furthermore, since the utility is typically a monotone increasing function of the data rate, increasing throughput usually increases the utility as well. For this reason, throughput maximization is often used to maximize the utility. We compared the optimal utility of the solution to our proposed model with the utility achieved by rounding down the optimal solutions of the continuous model and the utility of the solution with maximum throughput. In the continuous model, a piecewise linear utility function was maximized, which connects the discrete utility values. The exact optimum of the utility was found through linear programming in the continuous case. A discrete data rate model was applied in the case of throughput maximization, and the solution with maximum throughput was found by applying ILP.

5.1. Multiple Runs

The proposed algorithms were tested on generated networks. Five test series were generated; each of them consisted of ten cases. The number of nodes varied in the different series; there were 5, 10, 15, 20, and 25 nodes in the different series. We examined the utilities of the solutions as a function of the number of nodes. Figure 5 shows the average utilities of the solutions as a function of the number of nodes. The figures compare the results of the proposed algorithms, namely the greedy algorithm and the dynamic programming on the routing tree. Dynamic programming and integer linear programming methods always find the exact optimum. According to the chart, the greedy algorithm produces similar results to dynamic programming. The accuracy of the approach rounding the solution of the continuous model is low even if the continuous model is solved optimally. We obtained similar results for applying the solution with maximum throughput.

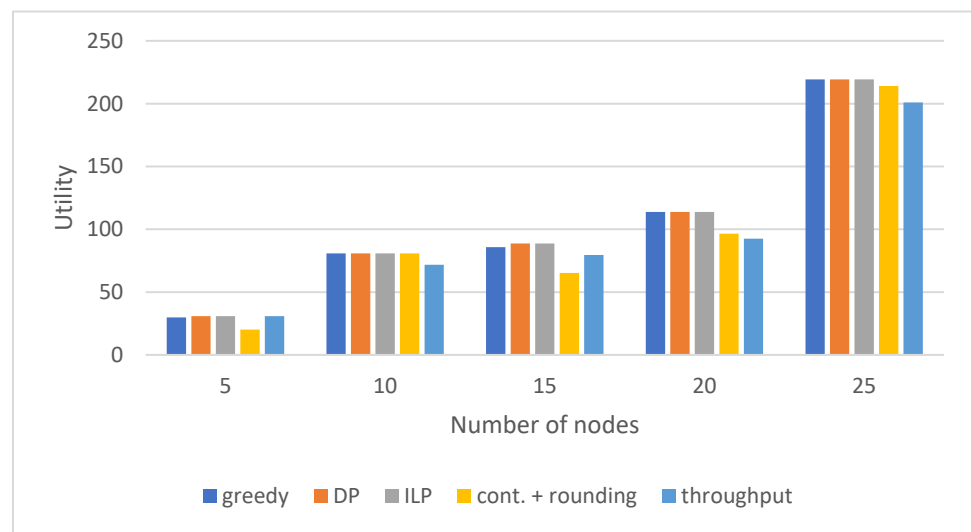


Figure 5. Utility values of the proposed data rate adaptation algorithms.

Table 2 shows the accuracy of the proposed algorithms. The accuracy of a method in a test series is defined as the ratio of the measured and optimal overall utilities. Although the greedy method does not guarantee the optimum, it approximates well it in the test series.

Table 2. Accuracy (in %) of the proposed data rate adaptation algorithms.

Number of Nodes	5	10	15	20	25
greedy	96.8	100	96.7	100	100
DP	100	100	100	100	100
ILP	100	100	100	100	100
cont. + rounding	65.5	100	73.5	84.8	97.6
throughput	100	88.7	89.6	81.3	91.7

Figure 6 shows the ratio of the solutions with the exact optimum to all test cases. It shows that the proposed greedy method finds the exact solutions in most cases. Although rounding the optimal continuous solution and the throughput optimization often find the exact optimum, they fail to find the optimum in many cases.

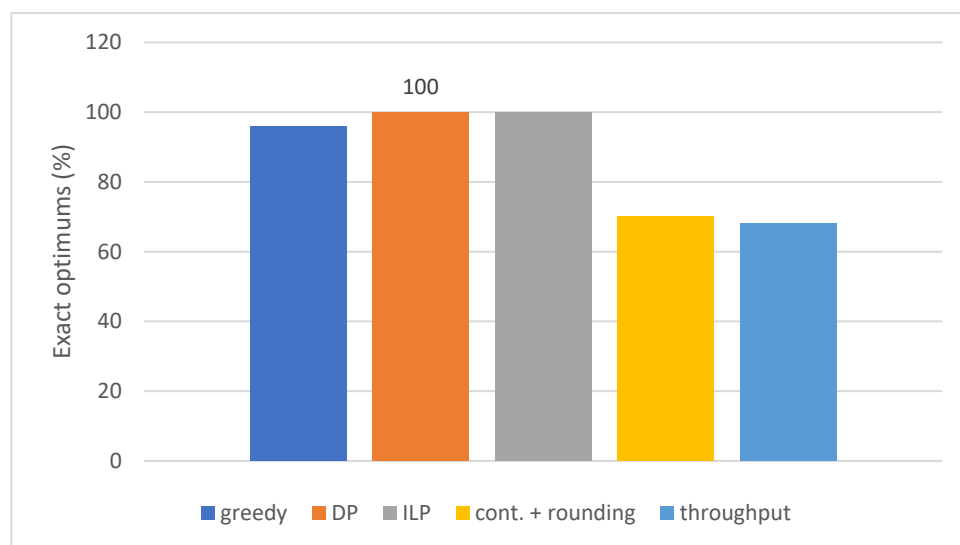


Figure 6. The ratio of test cases with the exact optimum.

5.2. Running Times

We also examined the running time of the proposed algorithms. The algorithms were implemented in C++. The tests were run on a 2.4 GHz processor with 8 GB memory. Table 3 shows the average running times in microseconds for different test series. As shown in the table, the greedy algorithm can produce results much faster than the dynamic programming approach. The dynamic programming algorithm required 50–150 times more time than the greedy method. The ILP-based method was clearly the slowest.

Table 3. Running times (in μs) of the proposed data rate adaptation algorithms.

Number of Nodes	5	10	15	20	25
1	8	14	26	32	30
DP	485	1893	2395	2870	3198
ILP	28,206	14,709	14,877	14,201	44,595

Our dynamic programming-based approach has a rather high complexity because it traverses the routing tree and runs dynamic programming for each node. The greedy algorithm has a faster running time than the dynamic programming algorithms and can provide similar results.

6. Conclusions

We proposed a model assigning discrete data rate values of the possible quality variations to multi-hop WBAN sensors. The discrete data rates can be dynamically adapted to changing network conditions to maximize the total utility of the data streams. The proposed approach can continuously stream data even in a shortage of resources, and it can also exploit the full capabilities of the devices if there are enough resources. We defined the formal model of the problem. We introduced two knapsack-problem-based algorithms to solve the problem of the sensor data rate adaptation. Our test results show that although the dynamic programming method can provide the best utility, the greedy method can also approximate the optimum well: it achieved at least 96% of the optimum. The running time of the greedy method was much shorter (1–2% of the running time of DP), which may represent a significant advantage in WSNs with limited computing capacity. The results show the importance of applying the discrete model. Although the solution of the continuous model can be adapted to the discrete case by rounding, the accuracy of this approach is low, even if the continuous model is solved optimally. The error of rounding the optimal continuous solution was, at most, 35%.

The proposed model has some limitations. For example, the utility function is hard to define exactly. An extensive analysis would be required to better understand the utility of the data streams. In the future, the model could be extended with energy constraints, which may represent a crucial limitation in WSNs.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Yaghoubi, M.; Ahmed, K.; Miao, Y. Wireless Body Area Network (WBAN): A Survey on Architecture, Technologies, Energy Consumption, and Security Challenges. *J. Sens. Actuator Netw.* **2022**, *11*, 67. [[CrossRef](#)]
2. Taleb, H.; Nasser, A.; Andrieux, G.; Charara, N.; Motta Cruz, E. Wireless technologies, medical applications and future challenges in WBAN: A survey. *Wirel. Netw.* **2021**, *27*, 5271–5295. [[CrossRef](#)]
3. Kathuria, M.; Gambhir, S. Quality of service provisioning transport layer protocol for WBAN system. In Proceedings of the 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), Faridabad, India, 6–8 February 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 222–228.
4. Rodrigues, V.F.; Righi, R.D.; da Costa, C.A.; Antunes, R.S. Smart Hospitals and IoT Sensors: Why Is QoS Essential Here? *J. Sens. Actuator Netw.* **2022**, *11*, 33. [[CrossRef](#)]
5. Alam, M.M.; Hamida, E. Ben Surveying wearable human assistive technology for life and safety critical applications: Standards, challenges and opportunities. *Sensors* **2014**, *14*, 9153–9209. [[CrossRef](#)]
6. Ben Elhadj, H.; Elias, J.; Chaari, L.; Kamoun, L. A Priority based Cross Layer Routing Protocol for healthcare applications. *Ad Hoc Netw.* **2016**, *42*, 1–18. [[CrossRef](#)]
7. Liu, B.; Yan, Z.; Chen, C.W. Medium Access Control for Wireless Body Area Networks with QoS Provisioning and Energy Efficient Design. *IEEE Trans. Mob. Comput.* **2017**, *16*, 422–434. [[CrossRef](#)]
8. Algabroun, H. Dynamic sampling rate algorithm (DSRA) implemented in self-adaptive software architecture: A way to reduce the energy consumption of wireless sensors through event-based sampling. *Microsyst. Technol.* **2020**, *26*, 1067–1074. [[CrossRef](#)]
9. Bregar, K.; Kristofelc, T.; Depolli, M.; Avbelj, V.; Rashkovska, A. Power Autonomy Estimation of Low-Power Sensor for Long-Term ECG Monitoring. *Sensors* **2022**, *22*, 5070. [[CrossRef](#)]
10. IEEE. IEEE Standard for Local and Metropolitan Area Networks—Part 15.6: Wireless Body Area Networks. *IEEE Stand.* **2014**, *2014*, 74.
11. Shan, F.; Luo, J.; Wu, W.; Li, M.; Shen, X. Discrete Rate Scheduling for Packets with Individual Deadlines in Energy Harvesting Systems. *IEEE J. Sel. Areas Commun.* **2015**, *33*, 438–451. [[CrossRef](#)]
12. Khairnar, P.S.; Mehta, N.B. Discrete-Rate Adaptation and Selection in Energy Harvesting Wireless Systems. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 219–229. [[CrossRef](#)]
13. Skorin-Kapov, L.; Ivesic, K.; Aristomenopoulos, G.; Papavassiliou, S. Approaches for Utility-Based QoE-Driven Optimization of Network Resource Allocation for Multimedia Services. In *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7754, pp. 337–358. ISBN 9783642367830.
14. Gomes, R.L.; Bittencourt, L.; Madeira, E.; Cerqueira, E.; Gerla, M. Management of virtual network resources for multimedia applications. *Multimed. Syst.* **2017**, *23*, 405–419. [[CrossRef](#)]
15. Gómez, G.; de Torres, E.; Lorca, J.; García, R.; Pérez, Q.; Arias, E. Assessment of Multimedia Services QoS/QoE over LTE Networks. In *Communications in Computer and Information Science*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 455, pp. 257–272.
16. La, R.J.; Anantharam, V. Utility-based rate control in the Internet for elastic traffic. *IEEE/ACM Trans. Netw.* **2002**, *10*, 272–286. [[CrossRef](#)]
17. Fu, X.; Modiano, E. Learning-NUM: Network Utility Maximization with Unknown Utility Functions and Queuing Delay. *IEEE/ACM Trans. Netw.* **2022**, *30*, 2788–2803. [[CrossRef](#)]
18. Cao, Y.; Sun, B.; Tsang, D.H.K. Online Network Utility Maximization: Algorithm, Competitive Analysis, and Applications. *IEEE Trans. Control Netw. Syst.* **2022**, *10*, 274–284. [[CrossRef](#)]
19. Li, L.; Long, J.; Zhou, W.; Jolfaei, A.; Haghighi, M.S. Joint Optimization of Energy Consumption and Data Transmission in Smart Body Area Networks. *Sensors* **2022**, *22*, 9023. [[CrossRef](#)]
20. Song, G.; Li, Y. Cross-layer optimization for OFDM wireless networks-part I: Theoretical framework. *IEEE Trans. Wirel. Commun.* **2005**, *4*, 614–624. [[CrossRef](#)]

21. Wang, C.; Guo, K.; Hu, X. The QoS and Energy Consumption Efficiency Trade-off Model Based on Utility Function in WBAN. In Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering, Xiamen, China, 22–24 October 2021; ACM: New York, NY, USA; pp. 1356–1360.
22. Misra, S.; Moulik, S.; Chao, H.C. A cooperative Bargaining solution for priority-based data-rate tuning in a Wireless Body Area Network. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 2769–2777. [[CrossRef](#)]
23. Mekathoti, V.; Nithya, B. A Survey on Congestion Control Algorithms of Wireless Body Area Network. In *Lecture Notes in Electrical Engineering*; Springer: Singapore, 2021; Volume 735, pp. 373–387. ISBN 9789813369764.
24. Kim, B.-S.; Shah, B.; He, T.; Kim, K.-I. A survey on analytical models for dynamic resource management in wireless body area networks. *Ad Hoc Netw.* **2022**, *135*, 102936. [[CrossRef](#)]
25. Ababneh, N.; Timmons, N.; Morrison, J.; Tracey, D. Energy-balanced rate assignment and routing protocol for body area networks. In Proceedings of the 2012 26th International Conference on Advanced Information Networking and Applications Workshops, Fukuoka, Japan, 26–29 March 2012; pp. 466–467. [[CrossRef](#)]
26. Baek, Y.; Lee, B.; Li, J.; Shu, Q.; Han, J.; Han, K. An adaptive rate control for congestion avoidance in wireless body area networks. In Proceedings of the 2009 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Zhangjiajie, China, 10–11 October 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1–4.
27. Ghanavati, S.; Abawaji, J.; Izadi, D. A Congestion Control Scheme Based on Fuzzy Logic in Wireless Body Area Networks. In Proceedings of the 2015 IEEE 14th International Symposium on Network Computing and Applications, Cambridge, MA, USA, 28–30 September 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 235–242.
28. Anwar, M.; Abdullah, A.H.; Saedudin, R.R.; Masud, F.; Ullah, F. CAMP: Congestion Avoidance and Mitigation Protocol for Wireless Body Area Networks. *Int. J. Integr. Eng.* **2018**, *10*, 59–65. [[CrossRef](#)]
29. Manfredi, S. Congestion control for differentiated healthcare service delivery in emerging heterogeneous wireless body area networks. *IEEE Trans. Wirel. Commun.* **2014**, *21*, 80–90. [[CrossRef](#)]
30. Liu, Z.; Liu, B.; Chen, C.W. Transmission-Rate-Adaption Assisted Energy-efficient Resource Allocation with QoS Support in WBANs. *IEEE Sens. J.* **2017**, *17*, 5767–5780. [[CrossRef](#)]
31. Liu, Z.; Liu, B.; Chen, C.W. Joint Power-Rate-Slot Resource Allocation in Energy Harvesting-Powered Wireless Body Area Networks. *IEEE Trans. Veh. Technol.* **2018**, *67*, 12152–12164. [[CrossRef](#)]
32. Goyal, R.; Patel, R.B.; Bhaduria, H.S.; Prasad, D. An Energy Efficient QoS Supported Optimized Transmission Rate Technique in WBANs. *Wirel. Pers. Commun.* **2021**, *117*, 235–260. [[CrossRef](#)]
33. Kiran, M.V.; Nithya, B. Network Status Aware Congestion Control (NSACC) Algorithm for Wireless Body Area Network. *Procedia Comput. Sci.* **2020**, *171*, 42–51. [[CrossRef](#)]
34. Kathuria, M.; Gambhir, S. Reliable packet transmission in WBAN with dynamic and optimized QoS using multi-objective lion cooperative hunt optimizer. *Multimed. Tools Appl.* **2021**, *80*, 10533–10576. [[CrossRef](#)]
35. Han, B.; Leblet, J.; Simon, G. Hard multidimensional multiple choice knapsack problems, an empirical study. *Comput. Oper. Res.* **2010**, *37*, 172–181. [[CrossRef](#)]
36. Yang, J.; Kim, Y.H.; Yoon, Y. A Memetic Algorithm with a Novel Repair Heuristic for the Multiple-Choice Multidimensional Knapsack Problem. *Mathematics* **2022**, *10*, 602. [[CrossRef](#)]
37. Mansini, R.; Zanotti, R. A core-based exact algorithm for the multidimensional multiple choice knapsack problem. *INFORMS J. Comput.* **2020**, *32*, 1061–1079. [[CrossRef](#)]
38. Khan, S.; Li, K.F.; Manning, E.G.; Watson, R.; Shoja, G.C. Optimal Quality of Service routing and admission control using the Utility Model. *Futur. Gener. Comput. Syst.* **2003**, *19*, 1063–1073. [[CrossRef](#)]
39. Khan, S.; Li, K.F.; Manning, E.G.; Akbar, M.M. Solving the Knapsack Problem for Adaptive Multimedia Systems. *Stud. Inform. Universalis* **2002**, *2*, 157–178.
40. Bellman, R. *Dynamic Programming*; Press Princeton: Princeton, NJ, USA, 1957.
41. Dudziński, K.; Walukiewicz, S. Exact methods for the knapsack problem and its generalizations. *Eur. J. Oper. Res.* **1987**, *28*, 3–21. [[CrossRef](#)]
42. Szkaliczki, T.; Eberhard, M.; Hellwagner, H.; Szobonya, L. Piece selection algorithms for layered video streaming in P2P networks. *Discret. Appl. Math.* **2014**, *167*, 269–279. [[CrossRef](#)]
43. Gavalas, N. Solving the Multiple Choice Knapsack Problem. 2019. Available online: <https://nickgavalas.com/solving-the-multiple-choice-knapsack-problem/> (accessed on 15 January 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.